Stacking Star Photographs

by
W. Smith

Introduction

A collection of digital astronomical photographs of the same star field, made with a camera that tracks the motion of the stars by means of a driven equatorial mount, will each differ in terms of the noise inherent in the imaging process. The noise may arise from atmospheric fluctuation which affects the light transmission, or from the electrical characteristics of the camera. Whatever the source, the noise degrades the quality of the image. However, making a composite of all the photographs, using a process known as stacking, can reduce the noise and improve the image quality. So if the number of photographs is N, the noise in the final stacked photograph will be reduced by a factor $1/\sqrt{N}$. For example, a stack of 100 photographs will have $1/10th$ of the noise of a single image and will therefore be considerably improved.

However, for the stacking to work, all the photographs must be fully *in register* with each other, which means the stars in every image must be in precisely the same position on the camera sensor when the image is taken. Unfortunately, errors in tracking, though perhaps negligible for a single image, can accumulate over a sequence of shots and result in an apparent shift in the star positions from image to image. It is necessary to correct for this shift to obtain a successfully stacked image. Achieving a proper registration of the images in a stack is the subject of this article.

The Stacking Function

We begin by defining an image stack as the average of a sequence of N images as follows

$$S_N(\vec{r})=\frac{1}{N}\sum_{n=1}^{N} I_n(\vec{r}), \quad (1)$$

in which $S_N(\vec{r})$ is the final stacked image and $I_n(\vec{r})$ is the $n'th$ image in a stack of N images. The vector $\vec{r}$ appearing as the argument in both these functions shows that they are functions of two variables, which we will designate as x and y. Together these define the location of individual pixels in the images on the XY grid that constitutes the two dimensional digital image. (We assume here that the centre of the image is the origin of the coordinates, indicated as $(0,0)$.) Variables x and y are formally the *components* of the vector $\vec{r}$ as shown below.

$$\vec{r}=\begin{pmatrix} x \\ y \end{pmatrix} \quad (2)$$

Since the images represent a star field we can write the function $I_n(\vec{r})$ as follows:

$$I_n(\vec{r})=\sum_{j=1}^{M} g_{n,j}(\vec{r}), \quad (3)$$

where the function $g_{n,j}(\vec{r})$ represents an individual star j in the $n'th$ image and M is the number of stars in the image.

The form of the star function can be represented in various ways, for example the stars may be regarded as points of light, which we could represent by

$$g_{n,j}(\vec{r})=b_j\delta(\vec{r}-\vec{r}_{n,j}), \tag{4}$$

where b_j is a scalar factor representing the star's brightness and $\delta(\vec{r})$ is a so-called *delta* function which is zero everywhere, except at locations infinitesimally close to the position $\vec{r}_{n,j}$, which is the position of the star in the photograph. The position vector $\vec{r}_{n,j}$ has two components: $x_{n,j}$ and $y_{n,j}$, as in

$$\vec{r}_{n,j}=\begin{pmatrix} x_{n,j} \\ y_{n,j} \end{pmatrix}. \tag{5}$$

An alternative star function is a Gaussian function of the form

$$g_{n,j}(\vec{r})=b_j\left(\frac{1}{2\pi\sigma^2}\right)\exp\left(\frac{-|\vec{r}-\vec{r}_{n,j}|^2}{2\sigma^2}\right), \tag{6}$$

in which b_j is again the star's brightness and σ is a measure of the width of the star's image in the photograph. The vector $\vec{r}_{n,j}$ in this case marks the location of the centre of the star in the image. The Gaussian form better represents how a star actually looks in a photographic image, while the delta function better describes a star in the absence of atmospheric light scattering. Fortunately, we do not need to worry about the actual form of the star function as such, as it is the brightness b_j and the star's location $\vec{r}_{n,j}$ that are of most concern in stacking.

Photographic Registration

In the set of N images we must choose one of them as the *reference image*, for which the image of the star field is accepted as the ideal (though it does not matter mathematically which image is chosen for this purpose). We designate this as image $I_s(\vec{r})$, where s is the index number of the image. In all the other images, $I_n(\vec{r})$ where $n\neq s$, the positions of the stars are assumed to be different i.e. $\vec{r}_{n,j}\neq\vec{r}_{s,j}$, because of differences between the photographs.

The purpose of this article is to determine how to bring a set of images into register with the reference image so they can all be merged into a final image with reduced noise. For this purpose a digital photograph editing program that supports a `Layers' feature, such as GIMP or Photoshop© is required. With the images handled as layers in a stack it is possible to manually translate or rotate the layered images until they are in register and then combine them into a single image. While this is not particularly difficult, it can be quite tedious if more than two or three photographs are involved. For larger image stacks some automation is called for, which means additional software (so called 'plug-ins') is required. But before such software can be written, a mathematical theory of the process is required, which is supplied here.

As mentioned above, the registration procedure requires two operations to be performed on the image. Each image must be translated (i.e. moved laterally) and/or rotated until the stars in the image are in register with the stars in the reference image.

The translation operation applies a *linear shift* of the image in a direction described by a displacement vector $\vec{d}_n$, which is defined below with components d_n^x and d_n^y.

$$\vec{d}_n = \begin{pmatrix} d_n^x \\ d_n^y \end{pmatrix}. \qquad (7)$$

The displacement vector $\vec{d}_n$ is initially an unknown quantity and has to be calculated somehow. Note that, since $\vec{d}_n$ is a vector, this means we have two unknowns: d_n^x and d_n^y.

The rotation operation is a uniform rotation of the image through some angle about a fixed point, which we may take to be the centre of the image. It is defined by a rotation matrix $\boldsymbol{R}_n$, which has the form

$$\boldsymbol{R}_n = \begin{pmatrix} C_n & -S_n \\ S_n & C_n \end{pmatrix}, \qquad (8)$$

in which

$$C_n = \cos(\theta_n) \text{ and } S_n = \sin(\theta_n) \qquad (9)$$

and θ_n is the angle through which the image must be turned. Once again θ_n is an unknown quantity which must be determined. So our registration procedure requires us to determine, for each individual image, three unknown quantities: θ_n, d_n^x and d_n^y. To enable this determination, we consider the effects of the translation and rotation operations on the stars in each image.

The effect on a star position $\vec{r}_{n,j}$ of a translation *without* an accompanying rotation is described by the equation

$$\vec{r}\,'_{n,j} = \vec{r}_{n,j} + \vec{d}_n, \qquad (n \neq s) \qquad (10)$$

where $\vec{r}\,'_{n,j}$ is the new position of the j'th star.

The effect of a rotation *without* an accompanying translation is given by

$$\vec{r}\,'_{n,j} = \boldsymbol{R}_n \cdot \vec{r}_{n,j}, \qquad (n \neq s). \qquad (11)$$

Equations (10) and (11) are simple enough, but the effect of applying both a rotation and a translation together raises a complication concerning which is applied first – the rotation or the translation.

If the rotation is applied first, the effect of applying both is given by

$$\vec{r}\,'_{n,j}=\boldsymbol{R}_n\cdot\vec{r}_{n,j}+\vec{d}_n, \qquad (n\neq s). \tag{12}$$

In this equation, the rotation is first applied to the star position vector $\vec{r}_{n,j}$ and then the translation $\vec{d}_n$ is added to the result. On the other hand, if the translation is applied first, followed by the rotation, the effect is described by

$$\vec{r}\,'_{n,j}=\boldsymbol{R}_n\cdot(\vec{r}_{n,j}+\vec{d}_n), \qquad (n\neq s). \tag{13}$$

Clearly, both equation (12) and (13) are valid descriptions of the effect of combined rotation and translation operations, but it is easy to see that for a given rotation matrix $\boldsymbol{R}_n$ and translation vector $\vec{d}_n$, the two equations will not give the same result[1] for $\vec{r}\,'_{n,j}$. It follows that when determining the unknowns θ_n and $\vec{d}_n$ we must be consistent in choosing either (12) or (13) as the description of the combined operations. In this article we shall hold to equation (12) as a convention.

With this issue decided, the method we will employ for determining the unknowns is the least-squares method described below.

The Least-Squares Method

To begin with we define a function G_N as follows

$$G_M=\frac{1}{M}\sum_{j=1}^{M}B_j\left|\vec{r}\,'_{n,j}-\vec{r}_{s,j}\right|^2. \tag{14}$$

This function is composed of the sum of the squares of the differences in position of each star j in the images n and s, weighted by the (squared) brightness term B_j where

$$B_j=b_{j,k}b_{s,j}, \quad \text{or alternatively} \quad B_j=b_{s,j}^2, \tag{15}$$

which is included to help ensure that the brightest stars will dominate in the registration procedure. (To that extent, the choice of weighting is arbitrary, so long as it biases in favour of the brightest stars.) Note that we use $\vec{r}\,'_{n,j}$ in (14)

---

1 The two results can be made equal if $\vec{d}_n$ in (12) is the same vector as $\boldsymbol{R}_n\cdot\vec{d}_n$ in (13), which means that $\vec{d}_n$ must be different in the two equations, though $\boldsymbol{R}_n$ is the same for both.

and not $\vec{r}_{n,j}$, since the former represents the star position *after* correction, and it is *that* vector we wish to be closest to the reference vector $\vec{r}_{s,j}$.

We may expand (14) as

$$G_M = \frac{1}{M}\sum_{j=1}^{M} B_j\left((x'_{n,j} - x_{s,j})^2 + (y'_{n,j} - y_{s,j})^2\right), \tag{16}$$

where it is understood that the components $x'_{n,j}$ and $y'_{n,j}$ of $\vec{r}'_{n,j}$ differ from the corresponding components of $\vec{r}_{s,j}$ on account of them being derived from a different image from the reference image. Only when $\vec{r}'_{n,j} - \vec{r}_{s,j} = \vec{0}$ for all stars in the image, do we get the result $G_M = 0$ (i.e. perfect registration). However, the function G_M in (16) is a quadratic function, which is always positive (i.e. larger than zero) when $\vec{r}'_{n,j}$ and $\vec{r}_{s,j}$ are unequal for any star j. Therefore when G_M achieves the minimum value possible, the differences $\vec{r}'_{n,j} - \vec{r}_{s,j}$ correspond to the best possible solution for the image registration. The minimum value of the function G_M occurs when the derivatives of G_M with respect to the parameters d_n^x, d_n^y and θ_n are zero. The derivatives of G_M are as follows.

$$\begin{aligned}
\frac{\partial G_M}{\partial d_n^x} &= \frac{1}{M}\sum_{j=1}^{M} 2B_j (x'_{n,j} - x_{s,j}) \frac{\partial x'_{n,j}}{\partial d_n^x} = 0 \\
\frac{\partial G_M}{\partial d_n^y} &= \frac{1}{M}\sum_{j=1}^{M} 2B_j (y'_{n,j} - y_{s,j}) \frac{\partial y'_{n,j}}{\partial d_n^y} = 0 \\
\frac{\partial G_M}{\partial \theta_n} &= \frac{1}{M}\sum_{j=1}^{M} 2B_j \left((x'_{n,j} - x_{s,j}) \frac{\partial x'_{n,j}}{\partial \theta_n} + (y'_{n,j} - y_{s,j}) \frac{\partial y'_{n,j}}{\partial \theta_n}\right) = 0.
\end{aligned} \tag{17}$$

To take matters further we must describe $x'_{n,j}$ and $y'_{n,j}$ explicitly in terms of d_n^x, d_n^y and θ_n. From equations (7), (8), (9) and (12) we obtain

$$\begin{aligned}
x'_{n,j} &= C_n x_{n,j} - S_n y_{n,j} + d_n^x \\
y'_{n,j} &= S_n x_{n,j} + C_n y_{n,j} + d_n^y,
\end{aligned} \tag{18}$$

and differentiating these equations we get

$$\begin{aligned}
\frac{\partial x'_{n,j}}{\partial d_n^x} &= 1 \\
\frac{\partial y'_{n,j}}{\partial d_n^y} &= 1 \\
\frac{\partial x'_{n,j}}{\partial \theta_n} &= -S_n x_{n,j} - C_n y_{n,j} \\
\frac{\partial y'_{n,j}}{\partial \theta_n} &= C_n x_{n,j} - S_n y_{n,j}.
\end{aligned} \tag{19}$$

Substituting (18) and (19) into (17) leads to

$$\begin{aligned}
&\frac{1}{M}\sum_{j=1}^{M} B_j\left(C_n x_{n,j} - S_n y_{n,j} + d_n^x - x_{s,j}\right) = 0\\
&\frac{1}{M}\sum_{j=1}^{M} B_j\left(S_n x_{n,j} + C_n y_{n,j} + d_n^y - y_{s,j}\right) = 0\\
&\frac{1}{M}\sum_{j=1}^{M} B_j\Big(-\left(C_n x_{n,j} - S_n y_{n,j} + d_n^x - x_{s,j}\right)\left(S_n x_{n,j} + C_n y_{n,j}\right) +\\
&\qquad \left(S_n x_{n,j} + C_n y_{n,j} + d_n^y - y_{s,j}\right)\left(C_n x_{n,j} - S_n y_{n,j}\right)\Big) = 0.
\end{aligned} \tag{20}$$

By expanding the brackets into separate terms and gathering them into separate summations, equations (20) can be written as

$$\begin{aligned}
&C_n \frac{1}{M}\sum_{j=1}^{M} B_j x_{n,j} - S_n \frac{1}{M}\sum_{j=1}^{M} B_j y_{n,j} - \frac{1}{M}\sum_{j=1}^{M} B_j x_{s,j} + d_n^x \frac{1}{M}\sum_{j=1}^{M} B_j = 0\\
&S_n \frac{1}{M}\sum_{j=1}^{M} B_j x_{n,j} + C_n \frac{1}{M}\sum_{j=1}^{M} B_j y_{n,j} - \frac{1}{M}\sum_{j=1}^{M} B_j y_{s,j} + d_n^y \frac{1}{M}\sum_{j=1}^{M} B_j = 0\\
&C_n\left(\frac{1}{M}\sum_{j=1}^{M} B_j x_{s,j} y_{n,j} - \frac{1}{M}\sum_{j=1}^{M} B_j y_{s,j} x_{n,j} + d_n^y \frac{1}{M}\sum_{j=1}^{M} B_j x_{n,j} - d_n^x \frac{1}{M}\sum_{j=1}^{M} B_j y_{n,j}\right) +\\
&S_n\left(\frac{1}{M}\sum_{j=1}^{M} B_j x_{s,j} x_{n,j} + \frac{1}{M}\sum_{j=1}^{M} B_j y_{s,j} y_{n,j} - d_n^x \frac{1}{M}\sum_{j=1}^{M} B_j x_{n,j} - d_n^y \frac{1}{M}\sum_{j=1}^{M} B_j y_{n,j}\right) = 0.
\end{aligned} \tag{21}$$

It is helpful at this stage to reduce the notational complexity of our equations by defining some simple averages:

$$\begin{aligned}
&\langle B_j x_{n,j}\rangle = \frac{1}{M}\sum_{i=1}^{M} B_j x_{n,j}, && \langle B_j y_{n,j}\rangle = \frac{1}{M}\sum_{i=1}^{M} B_j y_{n,j},\\
&\langle B_j x_{s,j}\rangle = \frac{1}{M}\sum_{i=1}^{M} B_j x_{s,j}, && \langle B_j y_{s,j}\rangle = \frac{1}{M}\sum_{i=1}^{M} B_j y_{s,j},\\
&\langle B_j x_{s,j} y_{n,j}\rangle = \frac{1}{M}\sum_{i=1}^{M} B_j x_{s,j} y_{n,j}, && \langle B_j y_{s,j} x_{n,j}\rangle = \frac{1}{M}\sum_{i=1}^{M} B_j y_{s,j} x_{n,j},\\
&\langle B_j x_{s,j} x_{n,j}\rangle = \frac{1}{M}\sum_{i=1}^{M} B_j x_{s,j} x_{n,j}, && \langle B_j y_{s,j} y_{n,j}\rangle = \frac{1}{M}\sum_{i=1}^{M} B_j y_{s,j} y_{n,j},\\
&\langle B_j\rangle = \frac{1}{M}\sum_{i=1}^{M} B_j.
\end{aligned} \tag{22}$$

Using these identities equations (21) become

$$\begin{aligned}
&C_n\langle B_j x_{n,j}\rangle - S_n\langle B_j y_{n,j}\rangle - \langle B_j x_{s,j}\rangle + d_n^x\langle B_j\rangle = 0\\
&S_n\langle B_j x_{n,j}\rangle + C_n\langle B_j y_{n,j}\rangle - \langle B_j y_{s,j}\rangle + d_n^y\langle B_j\rangle = 0\\
&C_n\left(\langle B_j x_{s,j} y_{n,j}\rangle - \langle B_j y_{s,j} x_{n,j}\rangle + d_n^y\langle B_j x_{n,j}\rangle - d_n^x\langle B_j y_{n,j}\rangle\right) +\\
&S_n\left(\langle B_j x_{s,j} x_{n,j}\rangle + \langle B_j y_{s,j} y_{n,j}\rangle - d_n^x\langle B_j x_{n,j}\rangle - d_n^y\langle B_j y_{n,j}\rangle\right) = 0.
\end{aligned} \tag{23}$$

Formally, these three simultaneous equations can provide the solution to the variables d_n^x, d_n^y and θ_n. Before proceeding to do this however, we can identify some particular circumstances where answers may be obtained more simply.

Case 1: Image translation only

This case applies to cameras on driven equatorial mounts. Quite often there is no apparent rotation of the images - providing the equatorial mount is well set up. Then we may assume that $\theta_n=0$ and consequently $C_n=1$ and $S_n=0$. This means that the equations (23) reduce to

$$\begin{aligned}
&\langle B_j x_{n,j}\rangle - \langle B_j x_{s,j}\rangle + d_n^x \langle B_j\rangle = 0\\
&\langle B_j y_{n,j}\rangle - \langle B_j y_{s,j}\rangle + d_n^y \langle B_j\rangle = 0\\
&\langle B_j x_{s,j} y_{n,j}\rangle - \langle B_j y_{s,j} x_{n,j}\rangle + d_n^y \langle B_j x_{n,j}\rangle - d_n^x \langle B_j y_{n,j}\rangle = 0.
\end{aligned} \tag{24}$$

Clearly, the third equation in (24) is redundant, since we already have two equations for the two unknowns d_k^x and d_k^y. The first two equations of (24) can be rearranged into

$$\begin{aligned}
d_n^x &= \frac{\langle B_j x_{s,j}\rangle - \langle B_j x_{n,j}\rangle}{\langle B_j\rangle},\\
d_n^y &= \frac{\langle B_j y_{s,j}\rangle - \langle B_j y_{n,j}\rangle}{\langle B_j\rangle}.
\end{aligned} \tag{25}$$

Equations (25) are therefore the solution for d_n^x and d_n^y when $\theta_n=0$. The redundant third equation of (24) holds for the displacements d_n^x and d_n^y and can be used to verify the result.

Case 2: Image rotation only

This case applies to cameras on a driven alt-azimuth mount that is properly North-South aligned, so that the image centre undergoes no shift in position (i.e. $d_n^x=d_n^y=0$), but there is evident rotation. In this case the equations (23) reduce to

$$\begin{aligned}
C_n\langle B_j x_{n,j}\rangle - S_n\langle B_j y_{n,j}\rangle &= \langle B_j x_{s,j}\rangle\\
S_n\langle B_j x_{n,j}\rangle + C_n\langle B_j y_{n,j}\rangle &= \langle B_j y_{s,j}\rangle\\
C_n\left(\langle B_j x_{s,j} y_{n,j}\rangle - \langle B_j y_{s,j} x_{n,j}\rangle\right) + S_n\left(\langle B_j x_{s,j} x_{n,j}\rangle + \langle B_j y_{s,j} y_{n,j}\rangle\right) &= 0.
\end{aligned} \tag{26}$$

As before, the equations (26) contain some redundancy and offer two methods for obtaining the angle θ_n. Solving the first two equations as simultaneous equations for C_n and S_n leads to the two results

$$S_n = \frac{\langle B_j y_{s,j}\rangle\langle B_j x_{n,j}\rangle - \langle B_j x_{s,j}\rangle\langle B_j y_{n,j}\rangle}{\langle B_j x_{n,j}\rangle^2 + \langle B_j y_{n,j}\rangle^2},$$
$$C_n = \frac{\langle B_j x_{s,j}\rangle\langle B_j x_{n,j}\rangle + \langle B_j y_{s,j}\rangle\langle B_j y_{n,j}\rangle}{\langle B_j x_{n,j}\rangle^2 + \langle B_j y_{n,j}\rangle^2}. \qquad (27)$$

From the identities (9) we can see that

$$\tan(\theta_n) = \frac{\sin(\theta_n)}{\cos(\theta_n)} = \frac{S_n}{C_n}. \qquad (28)$$

Combining the two equations (27) as indicated by (28) gives

$$\tan(\theta_n) = \frac{\langle B_j y_{s,j}\rangle\langle B_j x_{n,j}\rangle - \langle B_j x_{s,j}\rangle\langle B_j y_{n,j}\rangle}{\langle B_j x_{s,j}\rangle\langle B_j x_{n,j}\rangle + \langle B_j y_{s,j}\rangle\langle B_j y_{n,j}\rangle}, \qquad (29)$$

from which θ_n may be obtained.

On the other hand, combining the third equation of (26) with the identity (28) gives

$$\tan(\theta_n) = \frac{\left(\langle B_j y_{s,j} x_{n,j}\rangle - \langle B_j x_{s,j} y_{n,j}\rangle\right)}{\left(\langle B_j x_{s,j} x_{n,j}\rangle + \langle B_j y_{s,j} y_{n,j}\rangle\right)}, \qquad (30)$$

from which θ_n may also be obtained. The two equations (29) and (30) thus provide a check on each other.

Case 3: Images with combined rotation and translation

This is the most general case, and we have to solve all equations (23) simultaneously. We begin by rewriting these equations in the form

$$\begin{aligned}
&d_n^x\langle B_j\rangle = -C_n\langle B_j x_{n,j}\rangle + S_n\langle B_j y_{n,j}\rangle + \langle B_j x_{s,j}\rangle\\
&d_n^y\langle B_j\rangle = -S_n\langle B_j x_{n,j}\rangle - C_n\langle B_j y_{n,j}\rangle + \langle B_j y_{s,j}\rangle\\
&C_n\left(\langle B_j\rangle\left[\langle B_j x_{s,j} y_{n,j}\rangle - \langle B_j y_{s,j} x_{n,j}\rangle\right] + d_n^y\langle B_j\rangle\langle B_j x_{n,j}\rangle - d_n^x\langle B_j\rangle\langle B_j y_{n,j}\rangle\right) +\\
&S_n\left(\langle B_j\rangle\left[\langle B_j x_{s,j} x_{n,j}\rangle + \langle B_j y_{s,j} y_{n,j}\rangle\right] - d_n^x\langle B_j\rangle\langle B_j x_{n,j}\rangle - d_n^y\langle B_j\rangle\langle B_j y_{n,j}\rangle\right) = 0
\end{aligned} \qquad (31)$$

in which the first two equations of (23) have been rearranged to isolate the terms $d_n^x\langle B_j\rangle$ and $d_n^y\langle B_j\rangle$ on the left hand side and the third equation has simply been multiplied throughout by $\langle B_j\rangle$. Then, using the first two equations of (31), we can replace the terms $d_n^x\langle B_j\rangle$ and $d_n^x\langle B_j\rangle$ in the third equation giving

$$
\begin{aligned}
&C_n\Big(\langle B_j\rangle\big[\langle B_j x_{s,j}\, y_{n,j}\rangle-\langle B_j\, y_{s,j} x_{n,j}\rangle\big]+\langle B_j x_{n,j}\rangle\big[-S_n\langle B_j\, x_{n,j}\rangle-C_n\langle B_j y_{n,j}\rangle+\langle B_j\, y_{s,j}\rangle\big]-\\
&\langle B_j\, y_{n,j}\rangle\big[-C_n\langle B_j\, x_{n,j}\rangle+S_n\langle B_j\, y_{n,j}\rangle+\langle B_j\, x_{s,j}\rangle\big]\Big)+S_n\Big(\langle B_j\rangle\big[\langle B_j\, x_{s,j} x_{n,j}\rangle+\langle B_j y_{s,j}\, y_{n,j}\rangle\big]-\\
&\langle B_j x_{n,j}\rangle\big[-C_n\langle B_j\, x_{n,j}\rangle+S_n\langle B_j\, y_{n,j}\rangle+\langle B_j\, x_{s,j}\rangle\big]-\langle B_j\, y_{n,j}\rangle\big[-S_n\langle B_j x_{n,j}\rangle-C_n\langle B_j\, y_{n,j}\rangle+\langle B_j\, y_{s,j}\rangle
\end{aligned}
\qquad (32)
$$

Expanding the brackets in this equation allows the cancellation of many terms, and (32) becomes

$$
\begin{aligned}
&C_n\Big(\langle B_j\rangle\big[\langle B_j\, x_{s,j} y_{n,j}\rangle-\langle B_j\, y_{s,j} x_{n,j}\rangle\big]+\langle B_j\, x_{n,j}\rangle\langle B_j y_{s,j}\rangle-\langle B_j y_{n,j}\rangle\langle B_j x_{s,j}\rangle\Big)+\\
&S_n\Big(\langle B_j\rangle\big[\langle B_j x_{s,j}\, x_{n,j}\rangle+\langle B_j\, y_{s,j}\, y_{n,j}\rangle\big]-\langle B_j x_{n,j}\rangle\langle B_j\, x_{s,j}\rangle-\langle B_j\, y_{n,j}\rangle\langle B_j\, y_{s,j}\rangle\Big)=0.
\end{aligned}
\qquad (33)
$$

Using the identity (28) the equation (33) can be cast into the form

$$
\tan\theta_n=\frac{\Big(\langle B_j\rangle\big[\langle B_j\, y_{s,j} x_{n,j}\rangle-\langle B_j x_{s,j}\, y_{n,j}\rangle\big]-\big[\langle B_j\, x_{n,j}\rangle\langle B_j\, y_{s,j}\rangle-\langle B_j\, y_{n,j}\rangle\langle B_j x_{s,j}\rangle\big]\Big)}{\Big(\langle B_j\rangle\big[\langle B_j\, x_{s,j} x_{n,j}\rangle+\langle B_j y_{s,j}\, y_{n,j}\rangle\big]-\big[\langle B_j\, x_{n,j}\rangle\langle B_j x_{s,j}\rangle+\langle B_j\, y_{n,j}\rangle\langle B_j\, y_{s,j}\rangle\big]\Big)}, \qquad (34)
$$

from which θ_n may be determined. The unknowns d_n^x and d_n^y can then be obtained from the equations

$$
\begin{aligned}
d_n^x&=\big(-C_n\langle B_j\, x_{n,j}\rangle+S_n\langle B_j\, y_{n,j}\rangle+\langle B_j\, x_{s,j}\rangle\big)/\langle B_j\rangle\\
d_n^y&=\big(-S_n\langle B_j\, x_{n,j}\rangle-C_n\langle B_j\, y_{n,j}\rangle+\langle B_j\, y_{s,j}\rangle\big)/\langle B_j\rangle
\end{aligned}
\qquad (35)
$$

which are derived from (31).

Thus, in order to maximise the registration of the image $I_n(\vec{r})$ with the reference image $I_s(\vec{r})$ prior to completing the stacking defined in equation (1) we must, according to the transformation defined in (12), first rotate the image $I_n(\vec{r})$ through the angle θ_n and then shift the transformed image by the displacement $\vec{d}_n$. Note that θ_n and $\vec{d}_n$ should be determined *before* any corrective transformations have been applied.

Comments on Implementation

The procedure outlined thus far is an idealised algorithm. In reality, there are additional matters to consider when implementing a workable scheme.

The first point to discuss is how to obtain the position vectors $\vec{r}_{n,j}$ that locate the centres of the stars in the images. In the case of a digitized image, a computational procedure would resemble the following.

1. Scan through every pixel in the image and identify all the pixels that have a total brightness that exceeds some pre-defined minimum value and store the brightness value and location (i.e. x and y addresses of the pixel concerned) in suitable arrays. The total brightness can be some combination of the values of the three colour channels of an RGB picture,

such as the sum of the individual channel values. The x and y addresses are defined as the number of pixels horizontally and vertically in the two dimensional grid of pixels making the photograph – actual distances in normal length units are not needed.

2. A star in a digital image is a 'raft' of contiguous pixels in the 2D space of pixels (for a star it will be reasonably circular). The arrays x, y and b from step 1 defining the 'star pixels' in the image are scanned to identify pairs of pixels that are within 1 pixel's distance of each other in the x and y directions and therefore belong to the same star. If neither pixel is already a member of star, both are given the same, new 'star index number' (an array should be used for this purpose, with the same dimension as the x,y, and b arrays). If one pixel already has a star index, but the other not, the unassigned pixel gains the same star index. If both stars already have star indices, but for different stars, it becomes necessary to chose one of the star indices and assign it to all the pixels having the other star index, so they all belong to the same star. Once all pixel pairs have been dealt with, all the stars have been found.

3. All pixels with the same star index belong to the same star. This allows the properties of individual stars to calculated. These properties are: the average star brightness; the average x position; the average y position, the average radius of the set of star pixels. Note that the average x and y positions equate respectively to the coordinates $x_{n,j}$ and $y_{n,j}$ appearing in equation (5). They should be accurate to a fraction of a pixel.

Ideally all images $I_n(\vec{r})$ have the same stars and the same number of stars present, but usually however, this is not the case. Some stars will vary in brightness between images, due perhaps to variations in the sky, and may fail to show up in the pixel scan outlined above. Some stars may be lost because they have moved out of the image field of view. Therefore a knowledge of what stars are present in an image does not mean they have the same star index in in each image. Some means of identifying the same star in different images is therefore required.

If two images are shifted/rotated with respect to each other, it may be that the stars are still in a similar position in each image and may be found and equated with each other by comparing positions in the different images. However, such schemes are not foolproof. A better approach is to determine the distances between pairs of prominent stars in the same image and seek a pair of prominent stars with the same separation in the reference image. For images obtained with the same photographic equipment, such distances should be conserved. By comparing several such pairs of stars, it is possible to uniquely identify corresponding stars in different images. Once this correspondence is established, the calculation of the rotation and shift of an image with respect to the reference image can be undertaken using the identified stars.

W. Smith December 2019.