A Genetic Algorithm For Stellar Photometry

J.R. Parker
Laboratory for Computer Vision
Department of Computer Science
Univeristy of Calgary
Calgary, Alberta, Canada

Abstract

A class of scientific images, which we will call blot images, contains information in the form of relationships between grey level pixels. One way to extract this information is to fit model functions to the objects in the image. We have explored the use of a Moffat function as a data model, and use a genetic algorithm to fit many instances of this function to the data in the image. The example of stellar photometry is used, a natural application for the Moffat function.

1. Introduction

In general, scientific images can be classified into two classes: images in which position, lines, and shape are the key elements (EG graphs, chart recordings) which can be referred to as *line images*, and images in which grey levels or colors and their relationships are the most important items (EG astronomical images, DNA gels) which will be referred to as *blot images*.

The information content of a blot image is contained in the grey level values and relationships of the pixels in the image. It is generally true that mathematical functions can be used to model the objects in such an image. One specific example of a blot image to be explored is the astronomical stellar image, which consists of bright spots on a non-uniform 'dark' background (although astronomers often use the negative image). Researchers are often interested in the brightness and color of stars, especially when studying star clusters in general and globular clusters in particular. A problem is that stars are often too close together to obtain an accurate measurement of brightness. Moreover, there may be many thousands of stars in an image, a fact that makes computer assisted analysis very important.

The problems to be encountered in new blot image types involves the structure of the data to be extracted. The work appearing in this paper involves modeling the data objects in a blot image, so that the relevant scientific information may be extracted more easily. Mathematical functions, perhaps different for each image type, will be fit to the image objects, and the properties of the objects will be inferred from the parameters of the functions. For example, measuring the brightness of a stars is equivalent to fitting a function to the data and measuring the height of the fitted function. When star images overlap, many stars would have to be fitted simultaneously. Optimizing this fit requires of a method able to explore the variable space involved and not get caught in local minima, since the best result is that corresponding to the global minimum. Because of this normal downhill methods may not yield the best results. It is proposed here to ap-

ply the techniques of genetic algorithms, a biased random walk method, to the optimization of this fit.

The major example to be explored is that of crowded field stellar photometry. Section 2 of this paper describes the use of a genetic algorithm in this context. Section 3 describes the problems encountered in computer assisted photometry of crowded star fields and shows why this is interesting. Section 5 summarizes the work and gives some hints about future efforts.

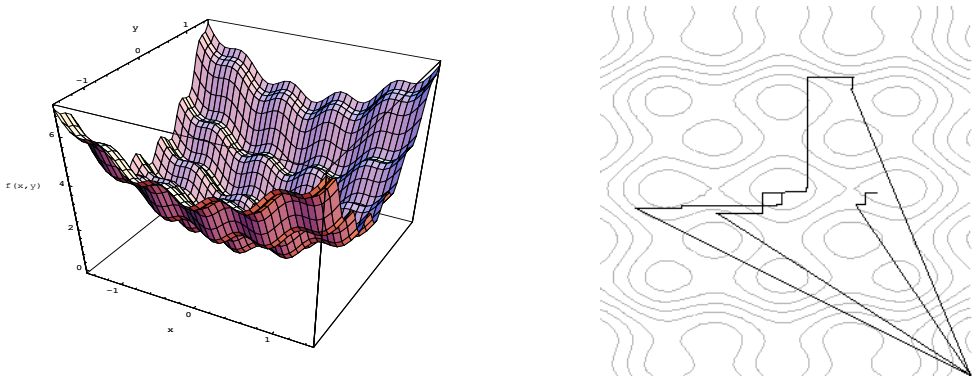
2. Genetic Algorithms

Many methods have been devised to optimize the value of a function in one of more parameters. These methods usually employ a figure of merit that determines how good the optimization is. They then optimize this value by changing the parameters repeatedly. The most straight forward approach is to choose new values of the parameters by changing them in the direction that reduces the value of the figure of merit. Though this would work fine for functions with a single minimum, it has the unpleasant problem of trapping the optimization process in local minima. To improve the chances of locating the one global minimum these methods are usually run several times from several different starting points and the best result obtained is taken as the global minimum.

Genetic algorithms have been used to minimize continuous real functions of many parameters, and, like simulated annealing, tend not to get trapped in local minima. In order to fit a function to a sampled surface a measure of goodness of fit is minimized. The figure of merit employed here to determine the goodness of the fit is the χ^2 value, which gets smaller as the fit improves. An example, introduced by Bohachevsky [1], can be used to illustrate these how to fit a function using a genetic algorithm. The function to be minimized is:

$$f(x, y) = x^2 + 2y^2 - 0.3\cos(3\pi x) - 0.4\cos(4\pi y) + 0.7$$

This function is an effort to model the sort of convoluted surface that a fitness function might achieve as a worst case. As can be seen in Figure 1a this function has many local minima and one well defined global minimum at (0,0). Any downhill method could get caught in one of these local valleys and never reach (0,0). To test the genetic algorithm we start at (1,1) and let the algorithm proceed. For this specific purpose we use a very simple implementation. Figure 1b



(a) (b)

Figure 1 - (a) Bohachevsky's function; it is hard to find a global minimum because of all o fthe local ones. (b) The paths that the genetic algorithm takes to find the minimum.

shows some example paths followed by the algorithm, where the dark lines indicate the steps that it followed. This plot illustrates why the genetic algorithm can be considered a random walk.

For every pixel in the region being fitted it is necessary to locate all those objects near enough to contribute some overlap. This is called *grouping*, and is done from a list of all located objects and their positions. One object is selected and then those which lie close enough are located and removed from the list. The process is recursively repeated for all these objects until no other overlapping object can be found. At this point a new object is selected from the list and the whole process repeated until no more objects remain in the list.

The genetic algorithm that is used for the function fit uses no bit strings. The operators, in this case single bit mutation and one point crossover, are implemented directly on the floating point parameters. The difference between this implementation and the traditional is numerically very small - the two approaches agree to five decimal places. What is more, the ranges associated with each parameter can be changed dynamically with no computational cost, which permits very accurate fits. The method used is described in [PARK95].

3. Stellar Photometry

One common activity in astronomy is to measure the brightness of stars. If the image of a star as seen through a telescope were just a dot of light, one would be tempted to just measure one pixel in our image and consider that to be the brightness of the star. Unfortunately, the light has been spread over a region of the image according to the transfer function of the telescope. Instead of just measuring one pixel what should be done is to integrate all this light back into one value. This is equivalent to undoing the spreading of the transfer function. If no other stars are present a simple way of doing this integration is simply to select a region of the image and add up the light in all those pixels. This is known as *synthetic aperture* photometry [2] because the region simulates how the aperture of a single detector would have measured the signal.

A second approach to photometry uses the transfer function of the device. Here we use a function to account for the fact that the star was a point source that was spread into the resulting image. Because of this it is common to refer to this function as the point spread function (PSF). If we can come up with an analytic expression for the PSF then given the image it should be possible to adjust this expression until the function best fits the data. We can now compute the brightness of the star by integrating this function. For clear isolated stars synthetic aperture offers the advantage of being very fast. Astronomers are however more interested in clusters of stars, and these result in what are called crowded fields or images. If two stars are close enough their PSFs may overlap. It is then impossible to use synthetic aperture reliably. This is because it is not possible to know how much of the light in the overlap region corresponds to each star.

When deciding on a function to approximate the PSF of a telescope the best choice is fairly obvious. Moffat [3] studied the way in which an image forms in a telescope, and he showed that a simple function with 2 free parameters was enough to account for most of the distortions. The function suggested has come to be known as the Moffat function and is of the form

$$I(r) = I_0\left(1 + \left(\frac{r}{\rho}\right)^2\right)^{-\beta}$$

where I_0 is the intensity of the star, I(r) is the intensity observed at a distance r from the center of the image and ρ and β are shape parameters. If the PSF is space invariant then so are the shape parameters. For example, given a value for peak intensity and full width at half maximum, only

one possible shape exists for both the Gaussian and Lorentzian curves. For the Moffat curve there is an infinite combination of values of ρ and β that still give the same full width at half maximum, as can be seen in Figure 2a. It is this flexibility that makes this function a more suitable choice.

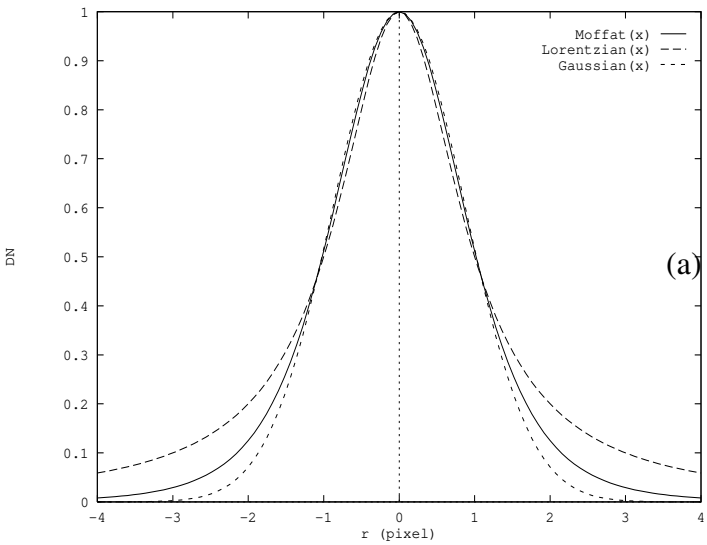
The photometry system was tested in two ways: first on an extensive suite of simulated images, and then on real images for which good photometry has been published. Testing on synthetic images was done to explore the ability of the algorithm to solve the different problems crowded-field photometry imposes in a controlled environment. Several sets of test data were created and different types of noise added to evaluate performance under conditions that went from ideal to realistic.

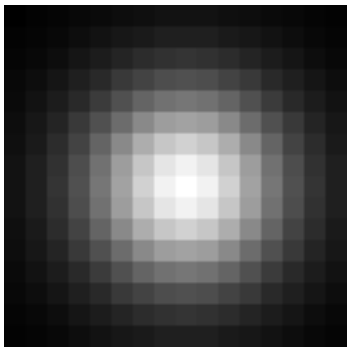
In a test image a star is represented by a two dimensional peak. A pair of *x* and *y* coordinates determine the center of the star, and a value of *I* in DN its brightness. The PSF of the image is controlled through the shape of the star function. The genetic algorithm attempts to minimuize the χ^2 value as measured against the star image, which is discrete. This means that the objective function is quite time consuming, requiring the comparison of the integral of the Moffat function over each pixel with the pixel value at the same point. The results from the genetic algorithm will be compared against the fit obtained by simulated annealing on the same star.

Figure 2b shows an example of a Moffat star, which is simply a two-dimensional Moffat function. An attempt was made to fit a Moffat function to this type of image using both the genetic algorithm and simulated annealing. The comparison between the two solutions is:

Method	Beta	Rho	X	Y	I	χ^2	Evaluations
Genetic	5.006	9.824	8.000	8.000	101.563	0.0040	14000
Annealing	4.658	9.431	8.003	7.999	101.332	0.0039	3366

After extensive testing on images containing a single simulated star, a series of tests were run using a set of test images having two test stars each. At first the test images corresponded to pairs of stars of equal intensity. One star is kept centered on a pixel while the other one is placed at different pixel offsets. Then experiments were performed using stars of variable brightness. Another test uses a sequence of two star images where the two stars are moved progressively





(b)

Figure 2 - (a) The Moffat function compared with a Gaussian and a Lorentzian.
(b) A Moffat star.

closer to each other. They correspond to a separation of between 1 and 10 pixels between the centers. Even at a distance of only 2 pixels the genetic algorithm does a good job of estimating both the center coordinates and intensities of the stars (Figure 3).

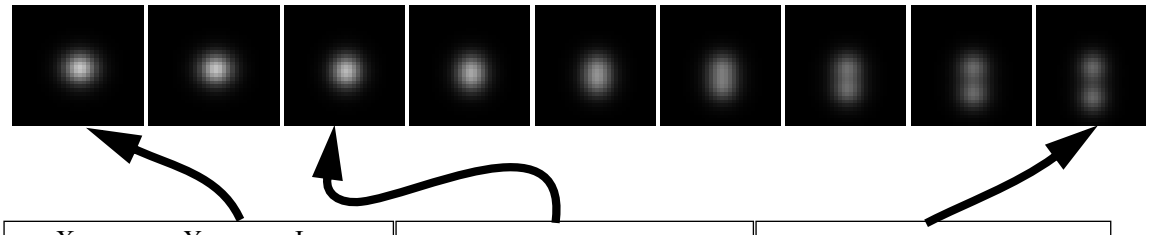
Tests were also performed to verify the ability to fit stars by iterating when a star is missed in a first pass, and later added to the fitting list. For example, an image was created with three stars in it. Two were large and separated enough to make sure they would be detected in the first pass, and a third smaller star was placed between these two so it would not be possible to see it in a first pass. An example of this sort of image can be seen in Figure 4, which shows both the residual image and a three dimensional representation of the residuals. When only one or two stars are fitted the median filtered residuals look like those in Figure 4a and b. Based on this a third star is added and all three fitted again. The resulting median filtered residuals can be seen in Figure 4c, which also has the best χ^2 value. Adding another star (4d) makes matters worse.

4. Conclusions and Further Work

While synthetic test images provide control situations were the performance of the algorithm can be tested, the final test is the application to real data. For this purpose several CCD images of the globular cluster NGC6397 have been obtained but have yet to be reduced[4,5]. In astronomical work the independent values obtained for each star in the frame are rarely considered individually. Usually two frames are available for reduction, one taken with a V filter (meaning visual, a yellowish color) and another one with a B (blue) filter. Once the stars are measured in both frames it is possible to establish their color from the two sets of results. A normal way of describing the color is by expressing the difference between these values, for example B-V. This is time consuming, but is under way.

However, it can be concluded that genetic algorithms may offer a viable way of optimizing function fits to data, in particular as it applies to astronomical photometry. The main drawback remains speed. By its very nature the method is slow, randomly exploring regions of parameter space that other methods may never visit. But it is also here were its strength lies, since in doing so it is able to come up with results other methods may miss.

This work has been supported by the Natural Sciences and Engineering Research Council of Canada.



Xo	Yo	I
15.7 (16)	15.6 (16)	102.65 (100)
16.3 (16)	16.83(16.5)	100.2(100)

Xo	Yo	I
15.9 (16)	17.8 (18)	103.0 (100)
16.1 (16)	16.1 (16)	95.52(100)

Xo	Yo	I
16.0 (16)	23.9 (24)	100.5 (100)
16.0 (16)	15.93(16)	98.72(100)

Figure 3 - Model Moffat star images. Each successive frame moves the lower star a little farther away fro the upper one. The overlap of two stars make them hard to identify.

5. References

[1] Bohachevsky, I.O., Johnson, M.E. and Stein, M.L., **Generalized simulated annealing for function optimization**, *Technometrics*,Vol. 28, Pp. 209-217, 1986.

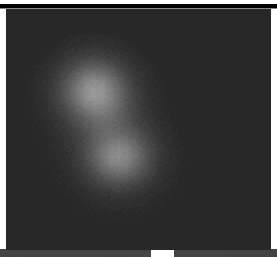
[2] Stetson, P.B., **DAOPHOT: A computer program for crowded-field stellar photometry**, *Pub. A. S. P.* , Vol. 191, 1987. Pp. 191-222,

[3] Moffat, A.F.J., **A theoretical investigation of focal stellar images in the photographic emulsion and application to photographic photometry**, *Astron. & Astophys*, Vol. 3, 1969. Pp. 455-461.

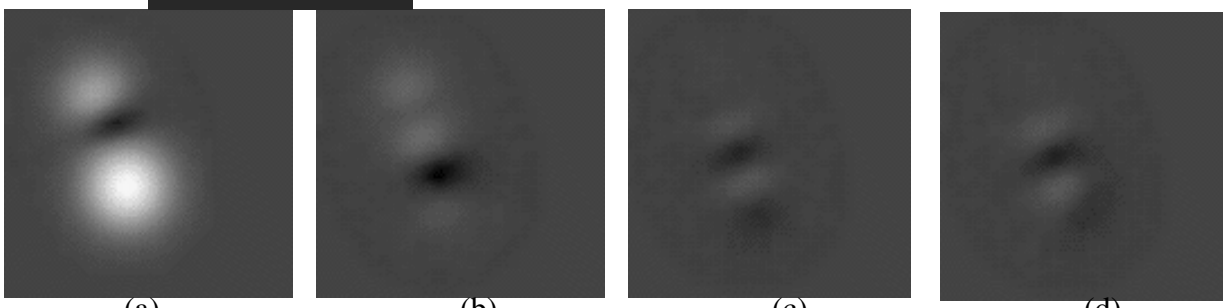
[4] Alcaino, G. et al, **The CM diagram of the nearby globular cluster NGC 6397**, *Astr. J.*, Vol. 94, No. 4, 1987. Pp. 917-1107,

[5] Alcaino, G., **The globular cluster NGC 6397**, *Astron. Astrophys. Suppl.* , Vol. 29, 1977. Pp. 397-405.

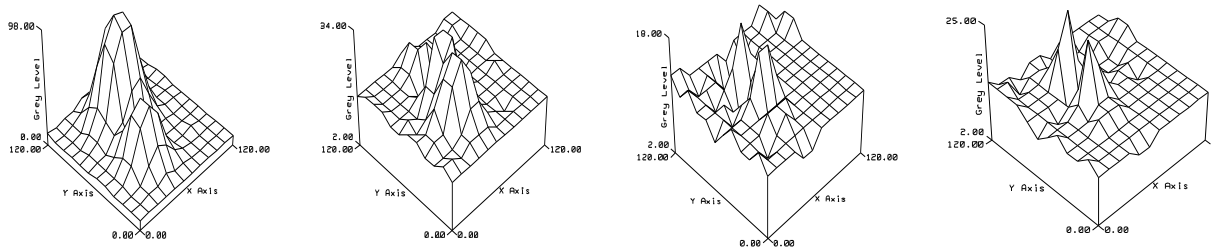
[6] Parker, J.R., **Genetic Operators on Floating Point Parameters**, *Department of Computer Science Research Report #95/564/16.* University of Calgary, 1995.



The original image of a trio of stars. There is a faint star near the upper left bright one that cannot be easily seen or detected by software. We attempt to extract 1, 2, 3, and 4 stars and look for the best χ^2.



(a) (b) (c) (d)



X0	Y0	I
10.63	11.53	109.98

X0	Y0	I
12.97	18.68	107.21
10.15	10.46	110.00

X0	Y0	I
11.71	15.14	31.30
13.13	19.40	88.79
10.04	10.15	109.99

X0	Y0	I
10.03	10.15	110.0
11.88	15.27	31.33
12.54	19.25	41.66
13.75	19.40	46.90

Actual: (10,10,100) (13,19,100) (11,13,30)

Figure 4 - Two bright stars and one faint one. [residuals] (a) Fit of one star (b) Fit of two stars (c) Fit of three stars (d) Fit of four stars. The three star fit is the best, and is correct.