FUTURE DEVELOPMENT OF THE DAOPHOT CROWDED-FIELD PHOTOMETRY PACKAGE

PETER B. STETSON
Dominion Astrophysical Observatory, 5071 W. Saanich Rd., Victoria, BC V8X 4M6, Canada

LINDSEY E. DAVIS
National Optical Astronomy Observatories, 950 N. Cherry Ave., P.O. Box 26732, Tucson, AZ 85726, USA

DENNIS R. CRABTREE
Space Telescope Sciences Institute, 3700 San Martin Drive, Baltimore MD 21218, USA

ABSTRACT The DAOPHOT package has been widely used by the astronomical community for obtaining stellar photometry in crowded fields. The first section of this paper discusses the port of DAOPHOT to the IRAF image processing system which we have undertaken in order to increase its availability to the astronomical community and enhance its scientific capabilities. The second section discusses some modifications to DAOPHOT's profile-fitting algorithms, intended to improve the program's accuracy when reducing undersampled data.

INTRODUCTION

DAOPHOT (Stetson 1987) is a software package for obtaining stellar photometry in crowded fields, which has been widely used in the astronomical community. The version of DAOPHOT which has been distributed by its author, Peter B. Stetson (PBS), is a stand-alone program written in Fortran which runs on VAX/VMS machines. Over the past few years stand-alone DAOPHOT has been ported by PBS and others to several machines, including a VAX running BSD Unix, Sun workstations, Crays, and an Alliant. It has also been integrated to varying degrees into existing astronomical image-processing systems including MIDAS and STSDAS. An early design decision to keep DAOPHOT independent of all graphics and display devices facilitated these ports; however, it also made it necessary for each recipient DAOPHOT site to re-develop such display and graphics capablities as manpower and hardware permitted. The recent explosion of workstations into the scientific market and the highly interactive nature of those workstations suggests that the stand-alone approach to software ports of major packages may not now be the best approach. Astronomers wishing both to keep up with new hardware developments and to make use of all the attractive interactive capabilities of the new machines may find the work of porting many individual software packages to new machines time-consuming and tedious. With these considerations in mind, two of us (LED and DRC) discuss in the first part of this paper the port of DAOPHOT into the IRAF image processing system, with emphasis both on how such a port can both increase the availability of DAOPHOT (and, by analogy, other astronomical software) to the astronomical community, and on how its scientific capabilities can thereby be enhanced.

Although DAOPHOT has been used successfully in numerous studies by various researchers, its current algorithms have some long-recognized limitations. Its technique of modeling the point-spread function (PSF) by means of the combination of a two-dimensional Gaussian and a look-up table of residuals can be a poor one for severely undersampled images, due to unavoidable interpolation errors. In addition, the DAOPHOT variable-PSF fitting algorithm, which employs a two-dimensional Gaussian and three look-up tables, is suitable only for removing simple linear variations in the PSF due, for example, to detector tilt relative to the telescope's optical axis. As larger and larger CCD's become available it will become increasingly necessary for DAOPHOT to track higher-order variations in the PSF due to optical aberrations and, perhaps, non-flatness of the detector. In the second section of this paper, one of us (PBS), discusses initial experiments with new PSF models for DAOPHOT.

IRAF/DAOPHOT

Why IRAF ?

The goals of the IRAF/DAOPHOT project are the following: to make the DAOPHOT software package more widely available to the astronomical community; to enhance the scientific capabilities of DAOPHOT and make it both more flexible and easier to use; and to reduce the package maintenance and installation overhead required at each site. The following discussion briefly outlines the reasons why the IRAF environment is particularly suitable for meeting these goals. For a full discussion of the IRAF system the reader is referred to the IRAF design paper (Tody 1986).

The IRAF system is portable. Only a small number of routines, known collectively as the host system interface (HSI), talk directly to the host computer's operating system. Only routines in the HSI must be rewritten or modified during a port to a new machine or operating system. All the applications programs and the command language (CL), which itself can be thought of as an application program, talk to the HSI through a well defined set of interfaces, known as the IRAF virtual operating system, or IRAF/VOS. Any application such as DAOPHOT, once ported to IRAF, will run on any machine which runs IRAF without changes to the applications level code. To date IRAF has been successfully ported to VAX machines running VMS, BSD Unix, and Ultrix, the DECstation (MIPS architecture) running Ultrix, Convex and Alliant running Unix, an MV10000 and MV8000 running AOS/VS, the HP9000 800 and 300 series running HP-UX and Sun 386i, Sun3, Sun4 and Sparc workstations running Unix. Ports to other machines are in progress or planned. The choice of new IRAF machines is dictated by community interest, the potential of the technology and the availability of manpower. The portability of IRAF allows the scientific user a wide choice of hardware, removing the necessity of being locked into any one vendor, while minimizing the installation and maintenance overhead.

IRAF has a fully integrated graphics and image display environment. The applications-level programs can make full use of the graphics and image-display capablities of supported devices without change to the applications level code. In the original DAOPHOT, graphics and display capabilities were deliberately left out of the package to avoid device dependence, and each site was left to implement those capabilites they thought necessary and could support. IRAF offers the opportunity to integrate those capabilities into DAOPHOT in a device- and hardware-independent way, making some tasks fully interactive. At present only the graphics environment is fully integrated into IRAF. A complete graphics interface is available to the applications programmer, and a large number of graphics devices are supported. Interfacing a new graphics device to IRAF is usually a simple matter of defining a 'graphcap' entry for each new device added. At present the only image display devices currently fully supported by IRAF are the IMTOOL server on Sun workstations and IIS Models 70 and 75, and the presently available image-display functions are particularly on the IIS are limited in scope. Display tasks for other machines and devices are available from outside IRAF sites. The display interfaces will undergo major development over the coming year.

A wide variety of reduction and analysis tasks already exist in IRAF and more are continually being developed. For example, some users might choose to reduce their raw CCD frames with CCDRED in the IRAF image reductions package (IMRED) before going on to analyze the data with IRAF/DAOPHOT. A wide range of spectroscopic reduction and analysis tasks as well as basic imaging tasks also exists inside IRAF. The common interface to all these tasks is the IRAF command language (CL), which looks identical to the user on any IRAF host machine. In fact since the CL provides many of the facilities of the host machine, it is possible for the IRAF user to be totally ignorant of the host operating system. In effect IRAF provides an interface to the host operating machine while maintaining the ability to escape to that operating system any time the user desires.

The IRAF system is expected to have a very long lifetime. New and existing software once ported to IRAF will be available to the user for long period of time, and will move automatically to new hardware as IRAF is ported to new machines, justifying the effort spent to develop new or port existing packages.

IRAF has a very rich programming environment. Currently three levels of programming interfaces are available to the prospective IRAF programmer. The CL itself is programmable. Existing tasks can be combined into simple scripts or have their parameter sets tailored for a particular application. Most users programming in IRAF program at the CL level. IRAF also supports a host Fortran interface IMFORT, which is a set of Fortran-callable routines providing access to the image pixels and the CL. Users wishing to create or modify small applications tasks commonly use this interface. Finally IRAF itself and all its major applications are written in IRAF subset preprocessor language (SPP) which provides full access to the IRAF/VOS. Writing applications in SPP is the method of choice for developing large, portable

IRAF applications. A package of software development tools, SOFTOOLS, is also provided for the software developer. Finally, the new layered external package facility available in IRAF 2.8 now makes it trivial to install locally developed packages into IRAF.

Porting DAOPHOT to IRAF offers a chance to standardize the software, and track changes in an orderly manner. There are now many versions of DAOPHOT in the community, and there are sometimes minor differences between results from one site's DAOPHOT as compared with those of another. Finally the IRAF group supplies a high degree of site support, often including remote debugging sessions.

For all the above reasons as well as high user demand, the IRAF group decided that an IRAF port of DAOPHOT was justified. The IRAF group also felt strongly that a collaboration with the DAO was highly desirable and could be crucial to the success of the project. The history of this collaboration is briefly discussed in the next section.

History of the IRAF/DAOPHOT Project

In January of 1988 the Canadian Astronomy Data Center (CADC) located at the Dominion Astrophysical Obervatory (DAO) and the IRAF group at the National Optical Astronomy Observatories (NOAO) agreed to collaborate on porting DAOPHOT to IRAF. A collaboration was felt to be in order as the port both required experience with DAOPHOT and involved translating DAOPHOT from its native Fortran to the IRAF SPP language. One of us, Dennis R. Crabtree, (DRC) then of the CADC and now at Space Telescope (ST), agreed to do the intial translation of DAOPHOT into SPP, since he was already familiar with DAOPHOT and had spent some time visiting the IRAF group and learning SPP. Lindsey E. Davis (LED) of the NOAO IRAF group agreed to do the system integration after the initial port.

The IRAF DAOPHOT package was demonstrated at the January 1989 meeting of the AAS in Boston, and officially delivered to NOAO in February of 1989, at which time integration into IRAF began. The IRAF system integration involved restructuring sections of the code to take better advantage of some of the SPP features such as dynamic memory allocation, streamlining the image i/o, and integrating the graphics and display facilities more fully into the package, and making the usual bug fixes. One of the principal accomplishments of this phase of the project was adding support for text file input and output. (The original version of IRAF/DAOPHOT supplied by DRC supported only binary ST Tables format.) Much of the system integration work was done by LED in the spring of 1989 while a guest of Cerro Tololo Inter-American Observatory.

On return to NOAO LED collaborated with staff scientist Phil Massey to complete the numerical verification of the package by running IRAF/DAOPHOT on a Sun4 versus DAOPHOT on a VAX/VMS machine. The DAOPHOT ALLSTAR task and the variable PSF fitting code, which were not supplied with the original version of IRAF/DAOPHOT, were ported to the package during the summer. In midsummer all the numerical pieces of the package were in place and serious user testing began. In August 1989 LED spent a week at the DAO collaborating with PBS. This visit provided an opportunity for LED to get updated on planned DAOPHOT algorithm development, and resulted in significant improvements to the ALLSTAR routine within IRAF/DAOPHOT.

The IRAF/DAOPHOT port is a significant event in the history of IRAF. Its completion marks the first time that the IRAF group has collaborated with an outside institution to port a major, externally developed applications package into IRAF. As IRAF becomes more widespread in the community it is hoped that such collaborations will become more common. IRAF/DAOPHOT is expected to be available to users as an external add-on package in late fall of 1989.

IRAF/DAOPHOT and DAOPHOT

Porting a well known package such as DAOPHOT into another image processing system presents both opportunities and dilemmas for the programmer. On the one hand a major port presents an opportunity to add desirable features and fix perceived deficiencies; on the other hand there exists a community of users who are familiar with DAOPHOT in its original form and may object to changes. In this section IRAF/DAOPHOT is compared to DAOPHOT, and the similarities and differences between the two packages are discussed. This discussion assumes familiarity with DAOPHOT on the part of the reader. Readers unfamiliar with DAOPHOT should consult Stetson (1987).

A block diagram of IRAF/DAOPHOT is shown in Figure 1. The user already familiar with DAOPHOT can see at a glance that the basic package structure defined by the individual tasks is identical to that of the original DAOPHOT. The path followed by traditional DAOPHOT users of defining parameters (OPTIONS), finding stars (FIND), fitting sky values and estimating starting magnitudes (PHOT), modelling the point-spread function (PSF), grouping the stars into physically meaningful associations (GROUP), simultaneously fitting the stars in each group (NSTAR), and subtracting the fitted stars (SUBSTAR) is represented by the left hand side of the flow chart. The OPTIONS task is replaced by five parameter sets (PSETS) defined logically by function: DATAPARS, the set of data-dependent parameters used by all the package tasks; CENTERPARS, FITSKYPARS, and PHOTPARS, which are centering-algorithm parameters, sky-fitting-algorithm parameters, and photometry parameters used by PHOT; and DAOPARS, the set of DAOPHOT fitting parameters used by PSF and all the tasks below it in Figure 1. The task for fitting individual stars (PEAK) and for grouping and fitting all the stars dynamically (ALLSTAR) are also included in the package, as well as a number of tasks for manipulating the input and output files, including RENUMBER, APPEND, SORT, and SELECT. The default algorithms used to carry out each of these tasks in IRAF/DAOPHOT are identical with those of DAOPHOT. In short, the IRAF/DAOPHOT tasks and default algorithms are the same as those of DAOPHOT. The differences between the two packages are discussed below.

IRAF/DAOPHOT has no image-size restriction. All the tasks work on any two-dimensional image. Similarly the PSF can be any size. IRAF/DAOPHOT decides at run time what portion of the input image it requires and dynamically allocates the necessary memory. An option to cache the entire image in memory is available in the ALLSTAR task.

The setup tasks FIND, PHOT, and PSF can be run either in their default batch mode, or interactively. In interactive mode the image cursor, image

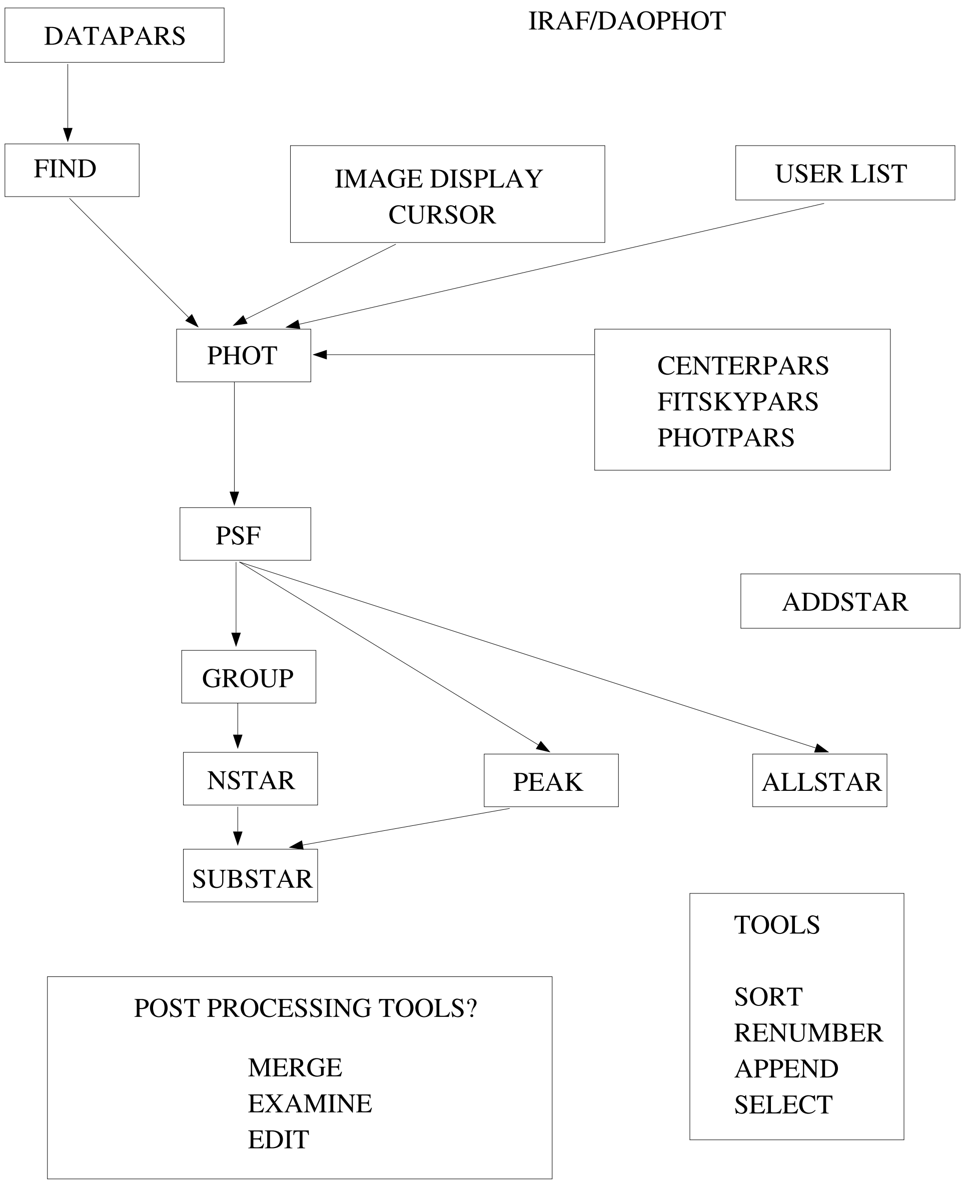

Fig. 1. Block diagram of the IRAF/DAOPHOT package.

display, graphics cursor and graphics window are used to select setup stars, display their radial profiles, interactively set the algorithm parameters and store the parameters. In batch mode the user simply edits the parameter files and runs the task. The split between interactive and batch modes permits users at sites with supported devices to make use of display and graphics capabilites while other sites can still run the tasks in batch mode. Users can bypass the FIND task and use PHOT in interactive mode to define and center a list of objects and compute initial magnitudes. Alternatively users can supply their own text files of (x, y) coordinates for input to PHOT.

The IRAF/DAOPHOT algorithm parameter sets can be stored in files in the same directory as the data and recalled for later use. A user processing several images with different characteristics and different optimal values for the parameters can change parameter sets simply by setting the appropriate parameter set filename.

The input/output files produced by IRAF/DAOPHOT in both text and binary ST Table format are self-describing: all the information required to decode any file is present in the file itself. Although the IRAF/DAOPHOT numerical tasks do expect to see certain columns, the tools used to process these files are independent of such quantities as the number, name and datatype of the columns. The IRAF/DAOPHOT output may then be modified or extended to meet user needs without impacting the list processing tools at all. The DAOPHOT tasks automatically sense on input whether a file is text or ST table. On output the user selects text or ST table. In IRAF/DAOPHOT the PSF is stored as an IRAF image rather than as a text file as in DAOPHOT. All the IRAF tools which can display and examine images can be used to examine the PSF.

Finally, IRAF/DAOPHOT will provide a number of post-processing tools. Some already exist in the form of the ST TTOOLS package for manipulating ST tables, which will be provided with IRAF/DAOPHOT. For example, crude CM diagrams can be produced using the TTOOLS software. Simple tasks for interactively examining and editing the output catalogues using the image display and graphics device and for merging output catalogues are planned for the near future. User input is particularly valuable in this area.

IRAF/DAOPHOT Meets the NOAO TEK2048 CCD

Although large-format chips have only recently become available to the astronomical community, it is clear that their availability and quality is only going to increase in the future (cf. papers in these proceedings). It is also clear that the large size of these new devices and the expected image shape variations across their field of view will challenge the capabilities of IRAF/DAOPHOT.

To estimate just how severe these problems are likely to be, B and V images of M92 were obtained at the NOAO 4m during an engineering run with the newly acquired TEK2048 CCD camera (Jacoby, *et al.* 1989) and run through IRAF/DAOPHOT. Each image is equivalent to 16 (512×512) images. Unfortunately, engineering problems resulted in severe charge-transfer inefficiency and poor image quality over the frames. The following conclusions, however, could still be drawn.

The large image size itself is not a fundamental problem for IRAF/DAOPHOT, as the necessity for holding the whole image in memory has been removed from all the package tasks. However some tuning of the internal IRAF image buffer sizes is necessary to increase the image i/o efficiency. The required changes are transparent to the user. Modern desktop workstations are sufficiently fast to handle these reductions in a finite period of time. The FIND and PHOT steps which detect and perform aperture photometry took 13 minutes of CPU time to run on 10,502 stars in the M92 V frame. The GROUP, NSTAR, and SUBSTAR steps took 85 minutes of CPU time to compute magnitudes for 6000 of those stars in the outer parts of the globular cluster. Although these timings are highly dependent on such variables as the distribution of stellar group sizes, they do indicate that an astronomer can both expect to take advantage of the large format chips to increase the data collection rate and decrease the bookkeeping, and to reduce the data in a finite period of time.

Unfortunately the poor image quality of our 2K images made it impossible to address the PSF variability issue at this time. However, preliminary inspection of the images and the results of PSF fitting suggest that simple linear variation in the PSF is not adequate to model the data. More experimentation on better quality data is planned. The whole issue of PSF fitting and higher order PSF models is discussed in detail in the second section of this paper.

Test Images for IRAF/DAOPHOT

The astronomical community has long recognized (*e.g.*, most recently, Murtagh and Warmels 1989, MW) that a series of standard test images would provide a useful toolkit for both testing new software and new ports of existing software and for evaluating results in the literature obtained with different software packages. Ideally such a set of test images would include a set of artifical frames modeling a wide range of physical conditions, but where the input data are exactly understood, and also a set of real images taken under various conditions.

It is planned in the case of IRAF/DAOPHOT to provide such a suite of test images and results, and to include them as part of the distributed package. These results would be used by the programmer to evaluate new algorithms and to quickly catch bugs introduced into the program during any future development, and by the user to quickly validate new releases or evaulate floating-point hardware differences between machines. Shortly before this meeting IRAF/DAOPHOT was sufficiently advanced to begin reductions of a subset of the MW (1989) test images. This exercise has already resulted in significant improvements to the ALLSTAR routine in IRAF/DAOPHOT and, when complete, will provide the first step towards producing an archive of test images.

Future Development of IRAF/DAOPHOT

Crowded field photometry is an evolving field and the software must evolve with it. The package structure of the original DAOPHOT clearly shows such evolution. The ALLSTAR task which both groups and fits the entire starlist dynamically was developed after the GROUP and NSTAR tasks, which were in turn developed after the single-star fitting task PEAK. IRAF/DAOPHOT is also expected to evolve. The three areas targeted for future IRAF/DAOPHOT development are the user interfaces, post processing tools, and algorithm development.

Users will see two approachs in the area of user interface development. Firstly, the package will be more fully integrated into the IRAF interactive graphics and display device enviroment in those areas where increased interactivity makes sense, such as PSF fitting, interactively subtracting stars from an image, adding artificial stars to an image, and grouping the stars. These changes will be added as programmer time and system development permit. Secondly, simple scripts will be provided to automate common procedures. Here again the example of PSF fitting can be used. Many users simply wish to select their PSF stars, identify their neighbours, fit the group, subtract out the neighbours, refit and iterate on this procedure until they are satisfied with the result, all without significant user interaction. This kind of automation can easily be done with IRAF scripts.

Several post-processing tools are planned for IRAF/DAOPHOT. It is hoped that the initial release will include an interactive tool for examining and editing the star catalogues which uses both the display and the graphics window if available, and performs simple catalog operations if it is not. Similarly, a robust tool for merging catalogs which can deal with the multiple hit problem is also planned for a later version of the package.

The whole area of algorithms research in DAOPHOT is also moving ahead. Currently a major effort is directed towards providing the user with a choice of point spread functions and increasing the complexity of PSF variations which DAOPHOT can handle. Developments in this area will be discussed in detail by one of us (PBS) in the next section of this paper.

It is the intent of the IRAF group to track major algorithm developments and incorporate them into IRAF/DAOPHOT in a timely manner.

MODIFICATIONS TO DAOPHOT'S POINT-SPREAD FUNCTION

In this section I (PBS) will be discussing various ways in which the detailed morphology of a star image can be encoded and stored in a computer, so that stellar positions and brightnesses may be derived by profile-fitting techniques.

Some years ago, King (1971) published an empirical determination of the radial intensity profile of a star image. He showed that the central part of the image is approximately Gaussian in form, due primarily to atmospheric seeing. The outer part of the profile appears to be dominated by scattering from dust and aerosols in the atmosphere, and from dirt and scratches in the optical system; it is well represented by a power law, $I \sim r^{-\beta}$ with $\beta \approx 2$ (but the outer profile cannot be exactly an inverse square law, because the integral to infinity would diverge). In between the Gaussian and power-law parts of the profile there is a transition region, which is as well represented by an exponential as by anything. Thus, a star image is inherently a complex thing. When you further realize that in practise telescope aberrations, tracking errors, and the spatial sampling of the detector itself impress their own signatures on the stellar images recorded in CCD frames, you will see that the best way to describe such an image in arithmetic terms — so that a computer program can process it — is far from clear.

In the 1980's a number of computer programs were developed more or less independently to perform profile-fitting photometry and astrometry from digital images, and each of them has adopted a different method for encoding and storing the point-spread function (PSF). Three mathematical functions in particular have been found to be useful descriptions of at least some parts of a stellar brightness profile. They are the Gaussian function, the Lorentz function, and the Moffat function; in their simplest forms they look like this:

$$\begin{aligned} \text{Gaussian:} \quad & G(r;\alpha) & \propto \quad & e^{-r^2/2\alpha^2} \\ \text{Modified Lorentzian:} \quad & L(r;\alpha,\beta) & \propto \quad & \frac{1}{1+(r^2/\alpha^2)^\beta} \\ \text{Moffat:} \quad & M(r;\alpha,\beta) & \propto \quad & \frac{1}{(1+r^2/\alpha^2)^\beta}. \end{aligned}$$

In addition, it is possible simply to store the *observed* profile of one or the sum of several bright stars — with the actual detector sampling — as a data array $O(i,j)$, and to estimate the brightness that would be observed at a different sampling by some interpolation technique:

$$\text{Empirical:} \qquad E(x_i,y_j) \quad \propto \quad O(i,j).$$

All of these methods have been tried in various combinations. For instance, ROMAFOT (Buonanno, *et al.* 1983, 1989) uses an analytic Moffat function; STARMAN (Penny and Dickens 1986) uses the sum of a Gaussian and a Lorentzian; WOLF (Lupton and Gunn 1986) uses an empirical look-up table ("L.U.T.") and *sinc* interpolation; DAOPHOT (Stetson 1987) uses the sum of a Gaussian and a look-up table with bicubic interpolation; and HAOPHOT (Gilliland and Brown 1988) uses the sum of a Moffat function and a look-up table with Gaussian-weighted bilinear interpolation.

When an analytic function is used either to represent the star image in its entirety or to provide a first guess at the brightness profile (which will subsequently be corrected by a look-up table of residuals), it is usually convenient to generalize that function by introducing additional parameters which allow it to resemble non-circular images produced by tracking errors or telescope aberrations. As two elementary examples,

$$\begin{aligned} G \quad & \Rightarrow \quad \exp\left[-\frac{1}{2}\left(\frac{x^2}{\alpha_x^2}+\frac{y^2}{\alpha_y^2}\right)\right] \quad \Rightarrow \quad \exp\left(-\frac{x^2}{2\alpha_x^2}\right)\exp\left(-\frac{y^2}{2\alpha_y^2}\right) \\ M \quad & \Rightarrow \quad \left(1+\frac{x^2}{\alpha_x^2}+\frac{xy}{\alpha_{xy}}+\frac{y^2}{\alpha_y^2}\right)^{-\beta}. \end{aligned}$$

There is a special advantage to *not* generalizing the Gaussian function any further: as long as the Gaussian is elliptical with the principal axes parallel to the rows and columns of the data array, then the two-dimensional integral over the area of each pixel can be decomposed into two one-dimensional integrals, which can be evaluated separately and then multiplied together. This allows a non-negligible speeding up of the computations. When other analytic functions than the Gaussian are employed, they cannot in general be transformed into one-dimensional integrals, and the full two-dimensional integrals must be computed. In this case, there is no particular speed penalty in making the analytic functions more elaborate — for instance, allowing the elliptical Moffat function to be inclined with respect to the rows and columns of the detector, as in the example above. However, the programmer must realize that when these non-linear functions are made increasingly complex, rapid convergence of the model profile is not always guaranteed. For example, when I attempt to fit an elliptical Moffat function to star images allowing all three α's *and* β to be determined simultaneously, I often have the devil's own time getting the solution to converge. I interpret this as meaning that β is usually so strongly correlated with the α's that β does not *need* to be known with great precision: the user may arbitrarily give β any reasonable value, and then the α's will adjust themselves to take up any slack in the fit.

In the case of undersampled images (FWHM $< 2 - 2.5$ pixels), the finite sampling of the detector and the stochastic way in which stellar centroids can land in the pixel grid will cause photometric and astrometric scatter. Let $F(x,y)$ represent the "model" point-spread function — either some analytic function, some look-up table plus interpolation scheme, or some combination of the above. Let $T(x,y)$ represent the "true" PSF in the CCD image, of which $F(x,y)$ is a working approximation. The best fit of F to T (in the least-squares sense) is given by

$$\int_{x=-\infty}^{\infty} \int_{y=-\infty}^{\infty} w(x,y)\,(F-T)^2\,dx\,dy = \mathrm{Min!}$$

(where $w(x,y)$ is a weighting function). However, the only information we have on the "true" PSF is obtained by discrete sampling with finite pixels,

$$O(i,j) = \int\!\!\int T(x,y)\,dx\,dy$$

where, in this case, the double integral is taken over the area of a pixel: $i - \frac{1}{2} \le x \le i + \frac{1}{2}$, and $j - \frac{1}{2} \le y \le j + \frac{1}{2}$. Therefore we cannot do the correct least-squares fit. The best we can do is

$$\sum_i \sum_j w_{i,j} \left[\int\!\!\int F\,dx\,dy - O(i,j) \right]^2 = \mathrm{Min!}$$

If F and T do not have *exactly* the same functional form (and they won't), these two statements of the least-squares problem will not in general have the same solution. Furthermore, the difference between the right answer and the almost right answer will depend upon just where the star's centroid falls with respect to the sample grid, upon the degree of over- or undersampling, and upon just how different F and T really are.

To provide an illustration of this effect, I produced a synthetic CCD image wherein I added 36 identical analytic Moffat functions ($\beta = 2$) into a frame full of zeroes (of course, the analytic Moffat functions were numerically integrated over the area of each pixel). The 36 stars were assigned centroids as follows:

$$x =, \ y = \begin{cases} \text{integer} \\ \text{integer} + 0.17 \\ \text{integer} + 0.33 \\ \text{integer} + 0.49 \\ \text{integer} + 0.67 \\ \text{integer} + 0.83 \end{cases}$$

I then fitted these artificial stars with an analytic Gaussian function, and with a Gaussian plus an empirical look-up table of corrections, *à la* DAOPHOT. The root-mean-square brightness variations which I obtained for these identical, infinite signal-to-noise profiles with different, finite sampling, are listed in Table I. It is seen that when the stellar images are oversampled,

TABLE I Sampling-induced photometric scatter in synthetic star images

FWHM (pixels)	σ(mag) Gaussian	Gaussian + L.U.T.
3.0	0.0000	0.0000
2.5	0.0001	0.0001
2.0	0.0008	0.0009
1.7	0.0026	0.0027
1.5	0.0056	0.0052
1.3	0.0104	0.0093
1.0	0.0196	0.0177
0.8	0.0223	0.0210

critically sampled, or only mildly undersampled (FWHM $\geq$ 1.7 pixels, say), the method which DAOPHOT uses to encode the point-spread function does not introduce photometric scatter large enough to be important in most applications. If one wishes to obtain the smallest possible scatter with more severely undersampled data, however, changes to DAOPHOT must be made. For instance, if instead of using an analytic Gaussian function to fit these artificial star images, I had used an analytic Moffat function with $\beta = 2$, clearly the photometric scatter in this test would have been identically zero for all sampling intervals (since if $F \equiv T$ the right answer will be obtained for any sampling). This suggests that the first line of defence against sampling-induced scatter is to *start* with an analytic function which resembles an actual star image as closely as possible.

I am therefore writing and testing a new version of DAOPHOT which will allow the user a choice of ways to encode the point-spread function. The lines of code which define each of the various possible analytic first approximations to the PSF are segregated into a single subroutine where a programmer can easily add or delete fitting parameters, or can custom-design an analytic function for a specific set of images. The options which I currently have working include a simple Gaussian function, a Moffat function, and a "Penny" function (the sum of a Gaussian plus a modified Lorentzian):

$$\begin{aligned}
G &\propto \exp(-\alpha_1 x^2/2) \cdot \exp(-\alpha_2 y^2/2) \\
M &\propto (1 + \alpha_1 x^2 + \alpha_2 y^2 + \alpha_3 xy)^{-5/2} \\
P &\propto (1 - \alpha_4) \exp\left[-\frac{1}{2}(\alpha_1 x^2 + \alpha_2 y^2 + \alpha_3 xy)\right] + \\
& \qquad \frac{\alpha_4}{1 + (\alpha_1 x^2 + \alpha_2 y^2 + \alpha_3 xy)^2}.
\end{aligned}$$

Any of these analytic first approximations can now be used alone, with a look-up table of corrections from the analytic first approximation to the "actual" stellar profile, or with either three or six look-up tables, which allow the point spread function to vary linearly or quadratically with position in the frame. Thus, the user will have at least the following options:

(1) Analytic function

$$O(x_i, y_i) \doteq A(x, y; \{\alpha\}), \qquad A = G,\ M, \text{ or } P$$

(2) Analytic function plus empirical look-up table

$$O \doteq A + E(x_i, y_i)$$

(3) Analytic function plus three look-up tables

$$O \doteq A + E_1 + (X \cdot E_2) + (Y \cdot E_3)$$

(4) Analytic function plus six look-up tables

$$O \doteq A + E_1 + (X \cdot E_2) + (Y \cdot E_3) + (X^2 \cdot E_4) + (X \cdot Y \cdot E_5) + (Y^2 \cdot E_6)$$

(where (X, Y) represents the star's position within the image). This may be compared with the version of DAOPHOT which has been around for a few years, which provided only a Gaussian plus one look-up table, or a Gaussian plus three look-up tables. I hope also to try out, at some time in the future,

(5) Analytic function with variable parameters

$$O(x_i, y_i) \doteq A(x, y; \{\alpha\}) \qquad \text{where} \qquad \alpha_i = \alpha_i(X, Y),$$

which may do a better job of dealing with a variable PSF within a severely undersampled image. The subtlety here will be to ensure that the *volume* of the model PSF does not change with position in the frame, which would introduce gradients in the photometric zero points.

In the two working days which have elapsed since I first got the new software running, I have had time to perform only one realistic experiment. I took an old CCD frame of the globular cluster E3 (McClure *et al.* 1985) which was obtained with a thinned RCA CCD at the prime focus of the CTIO 4-m telescope, and reduced it with each of the three analytic PSF's plus look-up table; the size of this frame was 284×492 pixels, and the stellar images had a FWHM of 2.54 pixels. Then I compressed the image by a factor of two in each dimension by a 2×2 block average, producing a 142×246 frame with an effective FWHM of 1.27 pixels, and reduced this new image, again with each of the three PSF's. Finally, I re-compressed the original image by a factor of three in each dimension, producing a 94×164 frame with a FWHM of 0.85 pixels. This last image was once again reduced with each of the three analytic PSF's, only this time I tried it both with and without the look-up table of empirical corrections from the analytic PSF to the "true" PSF. In contrast with the previous test on synthetic star images, this test will include some other effects of real data besides the PSF mismatch: notably, difficulty in estimating the correct sky brightness when the total number of independent uncontaminated pixels is reduced, increased confusion between near neighbors, and additional photometric scatter caused by the difficulty of correctly estimating the centroid positions of faint stars against a noisy background. It should also be noted that in compressing the data, the percentage of pixels affected by cosmetic flaws and cosmic rays will increase, since if even one of the pixels involved in a block average is bad, the resulting pixel will be bad.

The results of this test are listed in Table II. Here I have listed the amount of CPU time required to perform the reductions on a μVAX II for each of the three different analytic functions; it is clear from this that the Gaussian has an advantage of order a factor of two in reduction speed over the other two options. I have also listed the number of stars which were successfully reduced by each method, and the photometric scatter resulting from each. This latter is expressed as the standard deviation of the difference between the results for that method, as compared with the average for all twelve methods, for 426 stars in common to the twelve samples. These stars span a brightness range of more than eight magnitudes, from stars which reach near the linearity limit of the detector, to stars which represent $\sim 5\sigma$ detections. This method of estimating the scatter does not necessarily represent the true random photometric error of any given method — the scatter of the three full-frame reductions in particular will have been exaggerated by comparing them to averages which include the compressed-frame reductions. However, I think these figures should allow fair evaluations of the *relative* merits of the various reduction schemes.

TABLE II Sampling-induced photometric scatter in real data, for various model PSF's

Model PSF	CPU min. μVAX II	Number of stars	σ (mag)
Original frame, format = 284×492, FWHM = 2.54			
G + L.U.T.	38.4	463	0.0077
M + L.U.T.	78.4	463	0.0081
P + L.U.T.	70.6	463	0.0065
Compressed 2×2, format = 142×246, FWHM = 1.27			
G + L.U.T.	20.4	462	0.0355
M + L.U.T.	43.7	463	0.0092
P + L.U.T.	41.1	459	0.0139
Compressed 3×3, format = 94×166, FWHM = 0.85			
G + L.U.T.	20.8	452	0.0826
M + L.U.T.	41.4	447	0.0178
P + L.U.T.	36.2	449	0.0301
G	15.5	457	0.0894
M	39.7	440	0.0281
P	33.9	437	0.0414

I do not think that the photometric precisions of the three different reductions of the full frame are significantly different. It is perhaps not surprising that the Penny function seems to give the best results in this case — by a small margin — because it has *both* a Gaussian core and power-law wings (not to mention having four adjustable parameters, compared to two for the Gaussian function, and three for the particular Moffat function which I tried), but I can arrive at different relative rankings of the σ's by reducing the frame with different fitting radii, or by performing the same test with a different frame. When the data are critically- or oversampled, the speed advantage of the Gaussian plus look-up table strikes me as the significant difference. (This advantage may be less important as individual astronomers get their own workstations on their desks; on the other hand, it may become *more* important as 1024×1024 and 2048×2048 CCD's and CCD arrays become common). However, when the data are so severely undersampled that the seeing-induced Gaussian core of the stellar image is not resolved, a Gaussian function is a particularly poor approximation to a stellar profile; the Lorentz function and the Moffat function provide much better matches to the power-law wings which *can* still be perceived, resulting in much less sampling-induced scatter when these model PSF's are used. This is not a new discovery — it was recognized long ago by Roberto Buonanno, by Alan Penny, and by Ron Gilliland and Tim Brown. I do not know why the Moffat function appears to work better on these undersampled data than my "Penny" function; I also do not yet know whether this apparent difference would also be found for images from other detectors and telescopes. However, it is interesting to note that — in this test at least — the inclusion of a look-up table of corrections from the analytic function to the empirical PSF seems to have improved the photometric accuracy significantly even for the most extreme degree of undersampling tested.

Much experimentation remains to be done.

ACKNOWLEDGEMENTS

One of us (LED) would like to thank the directors and staffs of CTIO and DAO for the hospitality extended to her while the IRAF/DAOPHOT project was being worked on, and NOAO staff scientist Phil Massey for the considerable time he invested in doing the numerical comparisons between IRAF/DAOPHOT and DAOPHOT.

REFERENCES

Buonanno, R., Buscema, G., Corsi, C. E., Ferraro, I., and Iannicola, G. 1983 *Astr. Ap.*, **126**, 278.

Buonanno, R., and Iannicola, G. 1989 *Pub. A.S.P.*, **101**, 294.

Gilliland, R. L., and Brown, T. M. 1988 *Pub. A.S.P.*, **100**, 754.

Jacoby, G., Ciardullo, R., Green, R., Reed, R. and Wolfe, T. 1989, *NOAO Newsletter* **18**, 16.

King, I. R. 1971 *Pub. A.S.P.*, **83**, 199.

Lupton, R. H., and Gunn, J. E. 1986 *Astron. J.*, **91**, 317.

McClure, R. D., Hesser, J. E., Stetson, P. B., and Stryker, L. L. (1985) *Pub. A.S.P.*, **97**, 665.

Murtagh, F. and Warmels, R. 1989 *Proceedings of the ESO/ST-ECF Data Analysis Workshop*, in preparation

Penny, A. J., and Dickens, R. J. 1986 *M.N.R.A.S.*, **220**, 845.

Stetson, P. B. 1987 *Pub. A.S.P.*, **99**, 191.

Tody, D. C 1986, in *Instrumentation in Astronomy IV*, ed. D. L. Crawford, Proc. SPIE 627, 733