

Integrated Software for a Macromolecular Crystallography Synchrotron Beamline

JOHN M. SKINNER AND ROBERT M. SWEET*

Biology Department, Brookhaven National Laboratory, Upton, New York 11973, USA. E-mail: sweet@bnl.gov

(Received 1 May 1997; accepted 22 September 1997)

Abstract

A package of software has been produced for the operation of the synchrotron beamline X12-C at the National Synchrotron Light Source at Brookhaven National Laboratory. Years of observing common user mistakes has enabled the production of software that reduces user errors significantly. Users of the beamline communicate with all the experimental apparatus, including both the data-collection equipment and the beamline components, including the monochromator, through an easy-to-use graphical user interface (GUI). Important features of the system are (1) its modularity, so that different underlying programs or different apparatus can be incorporated easily; (2) its ease of use, minimizing both user errors and training effort; and (3) that most of the experimental operations and parameters are logged automatically, again minimizing errors and facilitating more-or-less automatic reduction of the data. Features of the software are useful enough to have been incorporated into operations at other synchrotron beamlines.

1. Introduction

Within a complete macromolecular crystallography project, the work carried out at synchrotron beamlines often is minor in time, but crucial in importance. Typical experimenters come to a synchrotron-radiation source only for brief periods, say three to four days, and rarely more than two or three times each year. During visits, they are expected to keep the apparatus running around the clock, and they try to use their precious beamtime allocation efficiently.

There are several consequences of these visits being short intense and widely spaced. One is that the experimenters rarely become completely familiar with the apparatus and the software that drives it. Secondly, they often are working under stressful conditions with little sleep. Finally, a team of three or four experimenters often will include one or two who are really not experienced crystallographers. They simply are not able to make all the decisions that might need to be made during data collection.

To give users of these facilities as much help as possible, we work continually to improve the interface between the users and the instruments. This interface includes both the physical presentation of the apparatus,

and the appearance and functioning of the software that controls it. There are several aspects to this work. Firstly, one must meet continuing advances in the quality of the experimental hardware. Of course computers must be upgraded continually, but the measurement apparatus itself seems to have an obsolescence lifetime of about three years. When apparatus is replaced, the control mechanisms and software also must be upgraded or replaced, along with the documentation and training program.

We take very seriously the role that good instrument-control and data-reduction software can play in making use of the facilities reliable and as easy as possible. Software development and maintenance should be able to respond easily to new experimental situations (new equipment or software) and to users' suggestions or complaints. The beamline software we present to our users now has several aspects of the experiment integrated into one package, is fairly easy for the users to learn, and is filled with checks to prevent mis-steps and to guide the user in making measurements.

2. The MARMAD/OptiX software package

A fundamental structural feature of this software is modularity. The two major underlying programs were

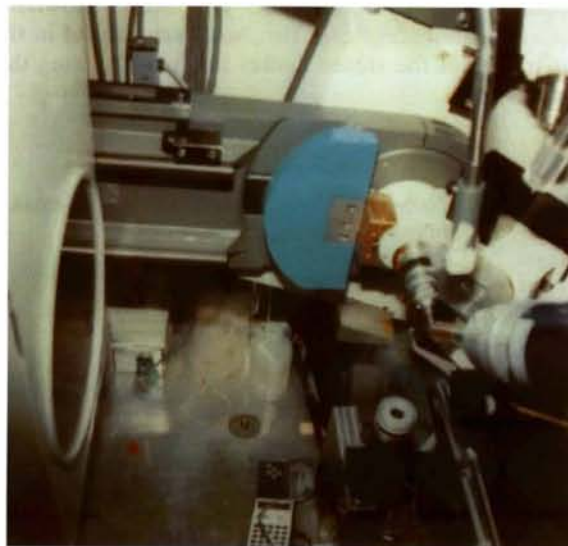


Fig. 1. The hybrid diffractometer at NSLS beamline X12-C with the FAST diffractometer carrying the MAR 300 mm IP-based detector.

command-line driven; we have produced graphical user interfaces (GUI's) that simply construct text commands and pipe them to the programs. The system we have constructed (Skinner *et al.*, 1993) allows communication among the components of the program and convenient modification through the use of popular GUI scripting languages like Tcl/Tk (Skinner *et al.*, 1996a).

The data-collection hardware that is driven by this software package includes interchangeable detectors (for example a MAR Research 300 mm imaging-plate scanner, or a CCD-based electronic imager) mounted on the θ arm of a Nonius FAST (CAD-4-style) diffractometer (Fig. 1), and standard beamline equipment – alignment table, counters, and monochromator drive. The diffractometer is controlled by a multi-process system.

(a) Diffractometer drive: *MADNES* (Messerschmidt & Pflugrath, 1987). This is the same *MADNES* that was written by the EEC workshop and ran at least three different diffractometers: a multiwire chamber at LURE, the FAST™, and the Argonne diffractometer at NSLS beamline X8-C.

(b) MAR detector drive: *MARdc*. From MAR Research.

(c) Image-transformation process: *IPXform*. Different processes with similar functions would be substituted for a CCD-based detector. All of these are under control of a GUI. Our GUI forks off *MADNES*, communicates through pipes that link standard I/O channels, then starts and communicates with other programs (Fig. 2). Here, and in the case of the beamline-control program *OptiX* described below, the original program *MADNES* still can be run on its own – with no

GUI! This particular feature is useful for debugging or maintenance, and the modularity leads to substantial flexibility. We have been able to patch in other detectors (Brandeis and ADSC CCD-based detectors) with few modifications of either their or our software.

The layout of the GUI matches the tree structure of *MADNES*, pruned to match the particular needs of our facility. Each level relates to different parts of data collection and analysis. We follow the *Motif Style Guide* (Heller, 1991; Open Software Foundation, 1991): there is a multi-pane screen with a 'standard' menu bar to give file control and to direct control to other levels. One must push a button to begin execution of any process. Numerical values always come from filled-in blanks. We eschew dials, up/down arrows, or slider bars for input of numerical values because usually a particular value is wanted and these 'analogue' devices make this difficult. Other features include the following.

- A command line for conventional text input.
- A scrolled text window for text output, including an adjustable paned window.
- Full frame adjustment to expand or decrease the size of the output window.
- Extensive color-coding of actions.
- A Files menu that includes I/O of conventional *MADNES* files, unix-style file commands *etc.* From the Files → Write button one may generate *Denzo* and *MOSFLM* processing scripts. In the case of *MOSFLM*, an orientation matrix also is generated. The files generated in this manner make use of the current parameters in *MARMAD*: specimen-to-detector distance, wavelength, file-name templates, ω , κ , φ , and θ . Since most of these parameters are read from the

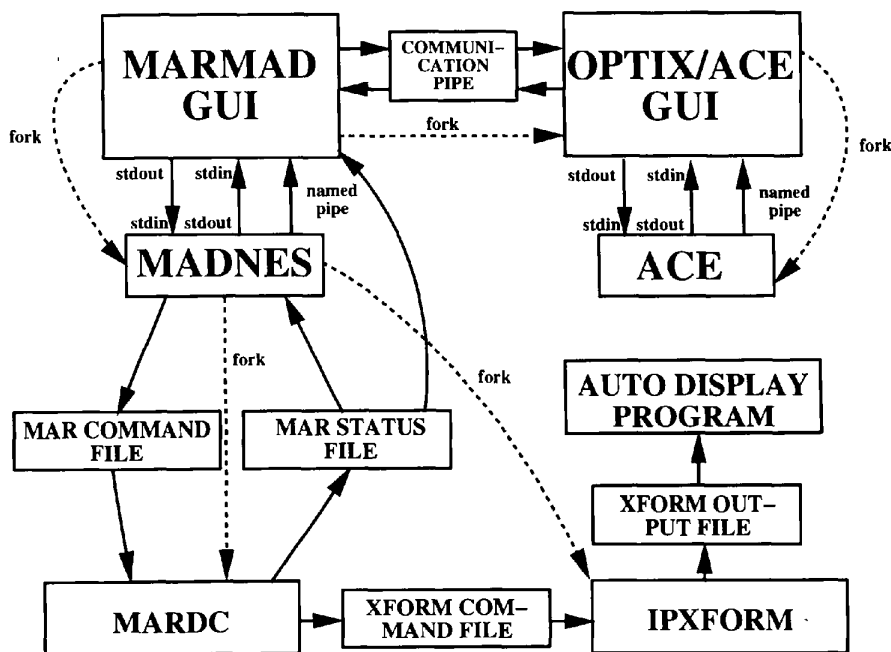


Fig. 2. Logical connections among the beamline-control programs. (Used with permission from Sweet & Skinner, 1997.)

hardware, the result of generating the script is usually that the script is ready to be run without any editing. The user typically needs only to edit the range of images to be processed.

Levels that are most important are DET ('detector' or diffractometer immediate control) (Fig. 3) and COLLECT. Both show relative zero ('datum') parameters and status of the MAR scanner: a sliding bar to show the progress of exposures, erasures, etc., and color coding to show the step in the process. In the DET level, axes of the diffractometer can be positioned either in native (ω , κ , φ) or in Eulerian (ω , χ , φ) settings. Exposures can be taken either as rotation images or still photographs. Also, one can choose detector size – 180 or 300 mm – or exert primitive control over the detector (scan, erase).

Control of the beamline comes from another combination of a GUI with an existing algorithmic program. The underlying program is ACE (Feng *et al.*, 1994), a program that allows control of stepping motors, counters, multi-channel analyzers, etc., mostly with command-line instructions. With this program and the GUI we have constructed, called *OptiX* (Skinner *et al.*, 1996b), we have reduced most beamline-control operations to a few button pushes. For example, ordinary beamline alignment requires only a button push to launch a few linked scans of the motorized table that carries the diffractometer, with optimization logic providing the final position (Fig. 4). A graph appears during the scan to show the profile of the X-ray beam. Color coding shows when the program is active – panels turn green when the counter is running or motors are moving. Pressing the *SetMono...* button activates a

monochromator-control panel (Fig. 5). One can drive to a selected wavelength (or photon energy), select a heavy-atom absorption edge from a selection box of K and L_{III} absorption edges (*SelectWave...* button), and perform a scan and plot a spectrum. A feature that we employ for 'automatic' MAD data collection (see below) is analysis of heavy-atom absorption spectra in terms of particular features, *i.e.* the rising edge or inflection point, the maximum at the 'white line', and the falling edge after the white line. We use an algorithm that performs Fourier smoothing of the spectrum with analytical calculation of the maximum and then maximum and minimum of the first derivatives (Press *et al.*, 1986).

Sweeps of data are recorded with the COLLECT level of *MADNES* (Fig. 5). There is a chart to set up multiple sweeps of data collection. This is useful for multi-wavelength data collection, to measure data for use of the anomalous effect by the 'Friedel Flip' method, or to distribute data to several disks.

Whenever possible we try to relieve users of responsibility for critical decision making, especially when these decisions can be made by carefully crafted algorithms. An example is the choice of wavelength for MAD data collection where that wavelength is usually a strong feature in an atomic absorption spectrum. Therefore, for precise control of the monochromator we have linked control of the data sweeps with control of the monochromator. The monochromator setting in the COLLECT level is coded with a flag (R, P, F, A) to indicate, respectively, whether the rising edge, the peak, the falling edge of the absorption spectrum of the heavy atom, or some absolute wavelength setting is to be used.

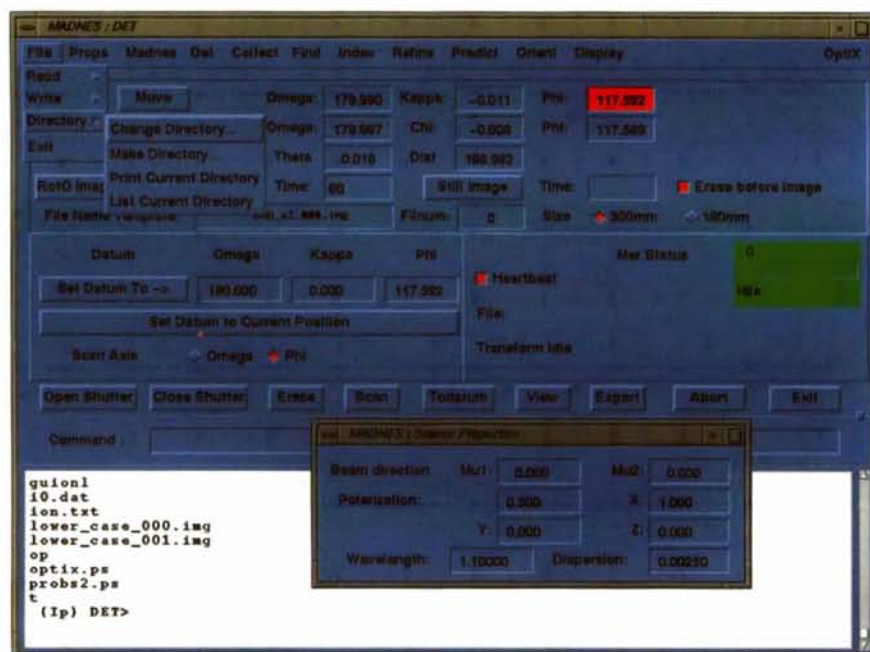


Fig. 3. The immediate-control, or DET level of the *MADNES* GUI. (Used with permission from Sweet & Skinner, 1997.)

A monochromator scan is made before each data sweep, our analysis algorithm selects the proper feature in the spectrum, the monochromator is set to that wavelength, and finally the sweep of data is recorded.

There are other ways we help users with decisions about how to take their data; these topics are discussed in detail elsewhere in this paper. Our Distancehints tool allows one quickly to place the detector in the right place for the data one wants to measure. The ability to

index and integrate images directly on-line from the data-collection program makes it straightforward for one to run 'strategy' software to establish the most efficient data-collection orientations. During data collection we provide a plot of the incident X-ray intensity, I_0 , so that users will see easily when realignment of the diffractometer would be useful; this process requires no decisions, just the push of a button. Also, users don't need to decide whether they need to realign

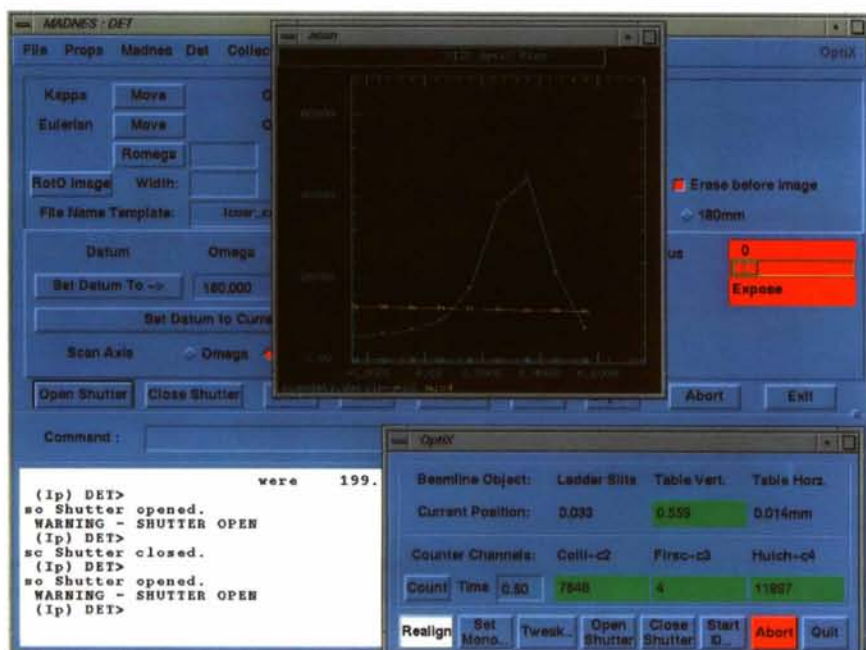


Fig. 4. View of the *OptiX* GUI during realignment of the diffractometer to the X-ray beam. (Used with permission from Sweet & Skinner, 1997.)



Fig. 5. Automatic MAD data collection in progress with the monochromator-control box and an absorption edge for a heavy atom displayed. (Used with permission from Sweet & Skinner, 1997.)

Table 1. *Possible user mistakes and their remedies*

To 'issue a warning' requires that a text window pop up in the foreground of the monitor.

Mistake	When to test	Remedy
Don't hit return after entering a value	When input-field contents change	Turn background red
Realign with no beam	Before every alignment or monochromator scan	Issue warning if test for a threshold of counts (~100) fails
Scan φ instead of ω when $\kappa \neq 0$	Just before start of data sweep	Issue stern warning that data reduction may fail
Start data collection with insufficient disk space	Ditto	Issue warning
Fail to change file names so that data will be overwritten	Ditto	Ditto
Wrong plate size with MAR	Ditto	Ditto
Forget to write 'sav' file	Ditto	Program writes it with unique name
Forget to write processing file	Ditto	Ditto
Forget to set relative zero to current position	Ditto	Issue warning for user to check
Forget to have correct monochromator-scan parameters	Just before automatic MAD-data collection	Ditto
Continue to take data with burnt-out MAR erase lamps	During data sweep	Aborts data collection
Don't pay attention to beam alignment	Ditto	Incident beam intensity is plotted
Forget what they did	Anytime	The scrolled output of <i>MADNES</i> is written in a file; a condensed logfile is written to record exposures
Wrong step size for monochromator scan	Just before a monochromator scan	Have default 'small' and 'large' choices
Choice of wavelength that would cause a collision	Ditto	Software knows the safe limits and refuses to scan
Don't know wavelength for heavy atom	Ditto	There's a table of many useful absorption edges
Forget to close shutter after realignment	After beam alignments	Software closes it
Uninterpretable characters input	Parsing of all input	Restore original value
Continue to take data after beam dump	During ordinary synchrotron fills or accidental dump	Beam monitoring routine restarts data collection

the system after a refill of the synchrotron if they use our restart option. Finally, the writing of control files for data reduction, and graphic diagnostic tools, allow users immediately to reduce their data to get rapid feedback on the quality, thereby assisting them in decisions about whether to stop or extend work on a particular crystal.

3. Status and safety

The real-time status of the experiment is displayed in several ways. Changing parameter values are updated constantly in the text fields of the GUI. The 'Time Remaining' field of the Mar Status Window indicates the amount of time until the current data sweep is complete. In the 'Mar Status' window in Figs. 3–5, the sliding bar is used to indicate what percentage of the current operation is complete. Different colors are used to represent the states of the detector and diffractometer shutter so that a user needs only to glance at the computer monitor to know what the current state of the system is. During an exposure the scale is red. It is white during an erase, pink while scanning, and green when idle. In addition, a text field that indicates the status of the spiral-to-Cartesian transform process turns yellow when the transform process is active. In the multi-row table of the COLLECT window (Fig. 5), the row that is in operation is colored green against the sky-blue background.

We make it straightforward for users to see their raw image data as they are being recorded to confirm that the quality of the data is satisfactory. To do this, we provide a separate program called *Autodisp* [a specially tailored version of the *Ipdisp* program supplied with the CCP4 package of software (Collaborative Computational Project, Number 4, 1994)] that can be left running on an X-terminal next to the workstation used for data collection. It will monitor data collection continuously, displaying the most recent image. It is fully automatic and requires no input.

Years of observing common user mistakes has enabled us to produce software that reduces user errors significantly (Sweet & Skinner, 1997). By starting with existing programs [*MADNES* (Messerschmidt & Pflugrath, 1987), *ACE* (Feng *et al.*, 1994)] that already possess most of the functionality we require, we have been able to focus our effort into the development of user-friendly GUI's for these programs. These front ends not only make the underlying programs easier to use, but are also intelligent in that they often evaluate user input in the context of the current state of the experiment to determine whether or not the user input 'makes sense'. This is approached in three ways: preventing an operation, forcing the user to acknowledge a potential mistake before proceeding, and reminding the user of suggested operations. We provide a complete but concise catalog of mistakes that might be made and our

corrections for them in Table 1. In addition, the checks made by the software before one takes a sweep of data provide a complete example.

(a) When one pushes the start button in the data collection screen, the program checks to see if $\kappa = 0$. If it's not, and the scan axis is not ω , then the user will be notified that data reduction may not be possible if the images are collected under these conditions. The user may then proceed, or cancel the request.

(b) If the start button is pressed and there is insufficient disk space to write all of the images in the data sweep the user will be told this and be given the option to cancel or continue.

(c) Users often take test exposures with the detector in 180 mm mode to save time, but may forget to place it in 300 mm mode for full data collection. Therefore, we prompt the user for confirmation if data collection is requested while the program is in 180 mm mode.

(d) If the above mentioned conditions are resolved, the final step before allowing data collection to start is to remind the user to be sure that the relative zero setting is correct and to write a **save** file.

There are other safety features.

(a) A feature of the underlying *MADNES* program is that no movements of the diffractometer will be allowed that result in a collision.

(b) The MAR detector employs lamps to erase the imaging plate between images. When these lamps burn

out, the MAR software prints a warning message indicating that this has happened, but proceeds with data collection. Experience has shown us that the data collected with burned out erase lamps is poor. We have modified our system so that when the erase lamps fail, the program will pop up a dialog window informing the users of the condition, suggest that they not proceed, and direct them to documentation (on the World Wide Web) describing how to replace the lamps. The user is given the option to continue or to abort the data sweep.

4. Electronic logging of the experiment

Three sorts of data are written 'blindly' by the program to assure that the parameters of the experiment are safe. The output of the scrolled window goes to a file. A special 'logfile' contains a concise textual report of operations made and data taken; an example is shown in Fig. 6. A *MADNES*-style save file, containing all the settable parameters from *MADNES*, is written with a coded name at the end of each sweep. Soon, each image field will contain a header with every possible parameter to describe the experiment stored in it.

5. Tools for the user

The Tools menu contains broadly useful utilities. The DistanceHints... tool solves Bragg's Law in several different ways: one can get a suggested detector distance either to provide the maximum resolution at the edge of the detector or to resolve individual spots; for a given distance and detector tilt one can get the d_{\min} at the top and bottom of the detector; for a given setting of distance, *etc.*, one can get a wavelength that will provide a particular d_{\min} , and finally one can calculate the proper detector tilt to provide a requested d_{\min} . A widely used

```
Data sweep at Tue Feb 20 16:52:52 1996 with following settings:
Phi rotation start = 0.00000, end = 60.00000, increment = 0.50000 time = 10.000
Directory = /usr3/x12cuser/skinner/fresh
File prefix = ppp_1, File Number = 0
Relative Zero settings:
Omega = 180.000, Kappa = 0.000, Phi = 5.990
Crystal to detector distance = 218.435, wavelength = 1.13800 theta = -0.016
```

Fig. 6. Example of text written automatically into the 'logfile' when a sweep of data images is begun.



Fig. 7. Output from *OneButton*, the rudimentary automatic data-reduction link to the *hkl* suite (*XdisplayF*, *Denzo*). (Used with permission from Sweet & Skinner, 1997.)

tool runs the *hkl* suite of programs [*XdisplayF* and *Denzo* (Otwinowski & Minor, 1977)] from within *MARMAD* to index an image and then to transfer the unit-cell and crystal-setting parameters back into the control program. One may then run the *ORIENT* routine to select a particular orientation for the crystal for data collection. Alternatively, a *Strategy* tool allows one to choose the best rotation range(s) for a given crystal setting to obtain nearly complete data in the minimum of time (Ravelli *et al.*, 1997). A logfile editor allows *emacs*-style (or point-and-click if you prefer) editing of the logfile so one can enter comments, time stamps, *etc.*, to provide an electronic textual record of the experiment (Fig. 6).

Employing an algorithm worked out and then suggested to us by Capel (1996), the program monitors the status of the synchrotron during data collection. When a beam dump is detected, the program will suspend data collection. When the beam becomes available, a 5 min wait allows the monochromator temperature to stabilize. Then the shutter is opened, the lift table and beamline slits are realigned, the shutter is closed, the MAR imaging plate is erased, and data collection resumes with the next image. Message windows pop up to keep the users informed about what is happening at each step. Information about when the beam dump occurred also is written to the logfile maintained by the program; this includes indicating whether the beam dumped during an exposure or not. In the worst case, one image may be underexposed if the beam dumped during an exposure. It should be obvious that this allows experimenters to get some sleep, with confidence that little beamtime will be wasted.

We believe that, to a first approximation, the experiment should be well enough defined by the known experimental conditions that data reduction should be automatic – one should need to push only one button to accomplish it. Our first attempt at doing this is shown in Fig. 7, which displays the output of a Tcl/Tk and *perl* script that runs *Denzo*. Here one can see two charts: a color-coded chart of χ^2 values from the image refinement and of $I/\sigma(I)$ values, and the histogram of partiality values from *Denzo*, averaged over the last five images. We also produce a grey-scale image of the averaged spot profiles from *Denzo* (not shown). For these images we enhance the contrast changes at the very edges of the reflections by a novel method: to the values from the tables output by *Denzo* (generally in the range 0–100) we add a small constant, then we take the logarithm of this result, then we add another small constant, and finally we scale this over the range 0–255 to produce a grey scale.

All our beamline documentation is available on the World-Wide Web (<http://lsx12e.nsls.bnl.gov/x12c/x12cinfo.html>), a strategy we find to be effective for a number of reasons. Since users no longer require a paper copy, we can be sure that everyone viewing the docu-

ment is seeing the current version. Points of interest are found easily in the web documentation with extensive cross-referencing in the document and the web browser's string-searching functions. Also, regardless of where our users come from, they can obtain the document quickly at any time. It's especially useful for users to be able to study the beamline operating instructions before coming to BNL. The document is organized in a layered way that can account for the likelihood that users won't read it all: the summaries of operations come first, with links or pointers to many of the details, which come later. In addition, pictures and links to other important documents (*e.g.* heavy-atom atomic absorption edges or tables of space-group symmetries) are incorporated easily.

6. A summary of our experiences

We have put a huge effort into the design, debugging and continual maintenance of this software. It has involved more than making it fun and easy for the user; there have been several goals. The first is to assure the quality of the data by providing ready feedback to the operator, by having the software document the experiment without user intervention, and by having numerous error checks within the code. A second is to relieve the training effort of the local staff and to extend the usefulness of the facility to relatively inexperienced users by providing an easy-to-learn and easy-to-use interface between the user and the software and hardware, and by providing detailed by layered documentation. Finally, the ease-of-use and reliability of the software, plus the ability of the local staff to make changes and improvements almost in real time, combine to make life easier for the visitors to the beamline. Regular reports of user satisfaction suggest that the effort has been worthwhile.

The suggestion to develop this GUI-driven system is due to James Pflugrath, who also participated in the design of the first *MADNES* GUI (Skinner *et al.*, 1993). Thereafter, nearly all of the features in this software resulted from suggestions (or mistakes) made by our dozens of beamline users, whom we thank warmly for their patience and encouragement. The Brookhaven Biology Department is supported by the Office of Health and Environmental Research of the US Department of Energy. This work was also supported by a biological instrumentation grant from the National Science Foundation.

References

- Capel, M. S. (1996). Personal communication.
- Collaborative Computational Project, Number 4 (1994). *Acta Cryst. D* **51**, 760–763.
- Feng, S. K., Siddons, D. P. & Berman, L. (1994). *Nucl. Instrum. Methods Phys. Res. A*, **347**, 603–606.

- Heller, D. (1991). *Motif Programming Manual*. Sebastol, California: O'Reilly and Associates.
- Messerschmidt, A. & Pflugrath, J. W. (1987). *J. Appl. Cryst.* **20**, 306.
- Open Software Foundation (1991). *OSF/Motif Style Guide*. Englewood Cliffs: Prentice Hall.
- Otwinowski, Z. & Minor, W. (1997). *Methods Enzymol.* **276**, 307–326.
- Press, W. H., Flannery, B. P., Teukolsky, S. A. & Vetterling, W. T. (1986). *Numerical Recipes: the Art of Scientific Computing*. Cambridge University Press.
- Ravelli, R. B. G., Sweet, R. M., Skinner, J. M., Duisenberg, A. J. M. & Kroon, J. (1997). *J. Appl. Cryst.* **30**, 551–554.
- Skinner, J. M., LaBarca, R. S. & Sweet, R. M. (1996a). *Proceedings, Fourth Annual Tcl/Tk Workshop 1996, July 10–13*, pp. 83–89. Berkeley: Usenix.
- Skinner, J. M., LaBarca, R. S. & Sweet, R. M. (1996b). *Nucl. Instrum. Methods Phys. Res. A*, **383**, 627–630.
- Skinner, J. M., Pflugrath, J. W. & Sweet, R. M. (1993). *SHARE 80 Proceedings*, Session I224, San Francisco, California, USA.
- Sweet, R. M. & Skinner, J. M. (1997). *Proceedings of the IUCr Macromolecular Crystallography Computing School*, edited by P. E. Bourne & K. Watenpugh. San Diego: SDSC Inc. <http://www.sdsc.edu/Xtal/IUCr/CC/School96/bs/bs.htm>.