44444444444444444444444444444444444444444444444444444444444444444444444

NIRSPEC

UCLA Astrophysics Program **U.C. Berkeley** **W.M.Keck Observatory**

4444444444444444444444444444444444444444444444444444444444444444

Tim Liu and Harry Teplitz **August 25, 1995**

NIRSPEC Software Design Note 4.00
Quick Look

1 Introduction

Quick-look facilities are necessary for a quick assessment of data quality. The current Keck instruments use a variety of standard image display and data reduction tools for quick-look purposes, such as SAOimage and figdisp. Unfortunately, these packages either primarily serve as image display tools and don't have all the functions needed for data manipulation or provide data processing functions too comprehensive to be easily used as a quick-look tool. Therefore, it is necessary to develop our own quick-look routines which integrate image display with data manipulation function specific to our needs. In addition to the quick-look facilities, NIRSPEC will provide at least one on-line data reduction package like IRAF or IDL for more complete data reduction.

2 Requirements

The NIRSPEC quick-look will consist of an image display tool used for 2-D images from both spectrometer and slit-viewing camera and a spectral plot tool for manipulating extracted 1-D spctra. The quick-look functions provided by the tools are discussed separately below:

2.1 Image Display Tool

Image Format and I/O:

! Read and write image data files in FITS format. The tool may operate images in its own format, but the data conversion from/to FITS must be efficient.
! New keywords can be added, existing keywords renamed, keyword values changed, and header can be listed.
! Data processing information should be recorded in the FITS header when the process file is saved.
! Provide hardcopy function of images and plots.

Basic Display Features:

! Zoom and pan an image.
! Linear and logarithmic mapping of images.

! Contrast and brightness can be specified by the user or computed by the tool using graylevel autoscaling algorithm.

! Provide both grayscale and color palettes with a selection of color tables; change contrast and brightness continuously for the current palette using mouse.

Computation Features:

! Pixel value is continuously read and displayed from the image at the current pixel location of the cursor.

! Add, subtract, multiply, or divide an image by a constant value or another image.

! Combine a set of images in median and with mode scaling.

! Compute the minimum, maximum, mean, median, mode, and standard deviation of any portion of an image.

! Compute centroid and distance of a point source.

! Box or aperture photometry. Background fitting and subtraction.

Plotting Functions:

! Plot radial profile at the image cursor position.

! Plot 2D line-cut; plot a cut along an arbitary line; plot horizontal/vertical cut and an average or a median collapsed cut.

! Plot histogram and contour of a portion of an image.

! Provide surface plot.

2.2 Spectral Plot Tool

Image Format and I/O:

The image format and data I/O requirements for the spectral plot tool are the same as for the image display tool.

Plot Functions:

! Plot multiple traces of a spectrum.

! Automatic or manual xy scaling; Zoom and pan.

! Provide automatic converion between various x-axis units, including pixel,wavelength, frequency, wavenumber, and velocity.

! Overplot.

Fitting Functions:

! Continuum fitting using polynomial or spline function.

! Line profile fitting using gaussian or polynomial.

Computation Functions:

- Arithmetic functions
- Caculate line center, integrated flux, FWHM, sigma, and equivalent width.
- Compute statistics of a spectrum
- Smooth spectrum using different filters

3 Programming Tool - IDL

It's not economical and necessary to develop a complete quick-look from scratch. Consider that quick-look routines are computing-intensive and that comercial software packages have provided efficent and robust math functions for various applications, we would like to develop the NIRSPEC quick-look around a computing environment which is suited to our requirements and also popular among astronomers. The choice is IDL, a powerful data analysis and visualization package from Research Systems, Inc. IDL integrates an array-oriented language with numerous mathematical analysis and graphical display techniques. Its main advantages which are relevant to our application include:

- IDL is a complete and structured language that can be used to create sophisticated functions, procedures, and applications.
- It provides rapid 2D and multi-dimensional plotting, volume visualization, image display, and animation.
- It has many numerical and statistical analysis routines and can therefore save us a lot of time by skipping writing these math functions ourselves.
- It allows easy construction of multi-platform graphical user interfaces using IDL widgets which is one of the best features IDL provides.
- IDL programs run the same across all supported platforms with little or no modification which will make our program portable.

By using IDL, we can focus on developing user-interface code and control flow of the quick-look, while the image display and plotting functions and various computation tasks are performed from executing IDL commands directly.

IDL version 4.0 provides a new feature of callable library, i.e. it allows other programs to call IDL as a subroutine. We were interested in using this feature for developing the NIRSPEC quick-look in our initial host software design. However, we have changed our idea after our investigation of callable IDL. The reason is it's difficult or impossible to write applications using callable IDL in which they create GUI and then have IDL draw graphics into windows it didn't create, because IDL's X-window graphics driver requires it to open its own display connection and run its own event loop which may not be possible. A side isssue is that in order to use the callable IDL, we need to purchase a X-window manager and development kit called the IXI premier Motif package which costs $600. Programs written in callable IDL cannot link to the bundled Motif-library under Solaris.

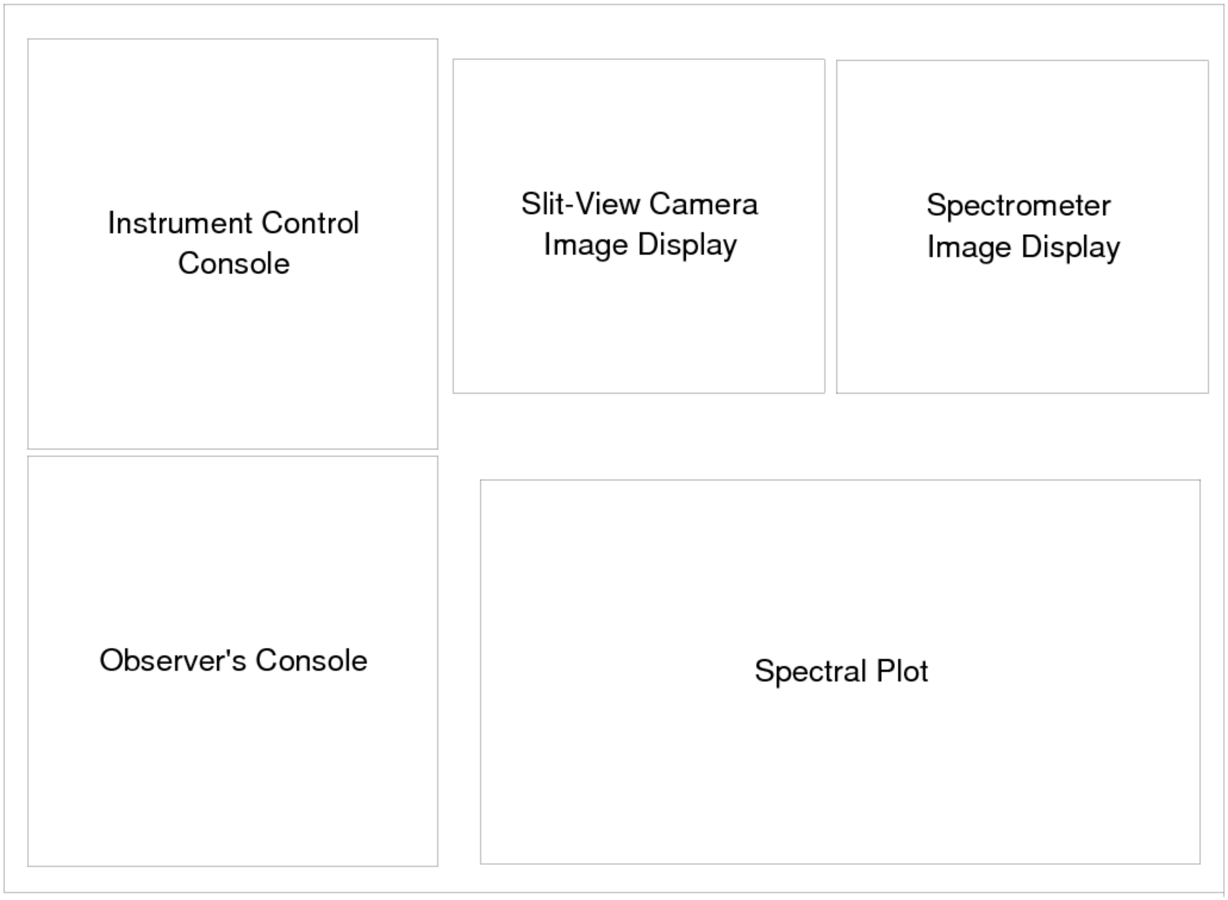
The alternative is to run IDL in a separate process and use an interprocess communication mechanism like RPC (Remote Procedure Calls) to control. If there's no need to talk to the instrument control program, the NIRSPEC quick-look can be developed as a completely stand-alone program.

4 GUI Design

As mentioned in the above section, one of the best features in IDL is its widgets which are simple graphical objects such as sliders, buttons, pulldown menus, etc. that allow you to create complete graphical user interfaces very easily. The NIRSPEC quick-look GUI will be developed with IDL widgets.

The quick-look requires two 2-D image display windows for images from both spectrometer and slit-viewing camera and a 1-D spectral plot window for reduced spectra from the data reduction pipeline, as shown in Figure 1. Figures 2 and 3 show a first-cut design of the image display and spectral plot implemented with IDL WIDGET tool.

Figure 1 GUI screen layout


Instrument Control Console
Slit-View Camera Image Display
Spectrometer Image Display
Observer's Console
Spectral Plot

4.1 Structure of IDL Widget Programming

IDL provides a programming language for creating WIDGET based user interfaces. WIDGET's are "ojbects" ranging from display windows to buttons to pull-down menus. Each WIDGET is treated as a separate object and can be addressed and tracked individually. User input to a WIDGET program causes an "event" to occur in the WIDGET. IDL provides a control program called XMANAGER to connect user events with the appropriate IDL subroutine response.

A WIDGET program consists of many WIDGET's acting together to form the user interface. Each window will consist of one "parent" WIDGET and many "children". The children provide the user interaction, while the parent keeps track of common information (particularly, when the entire window is dismissed). The entire GUI may consist of many windows, and thus many parent WIDGETS.

In the case of the Image Display tool, the parent WIDGET is the base for all the other WIDGETs. Its children are the pulldown menu bar, the image display window, and the pixel value fields. When the user interacts with any of the children, IDL informs the XMANAGER where the event occured, what the event was, and also that the event occured in a child of the display base. The XMANAGER then passes this information to the subroutine that interprets what to do about that event; usually that mean calling yet another subroutine. Dialogue boxes are their own parent WIDGETs, but they may be assigned to be in a "group" with the display base as the "group-leader".

Each IDL WIDGET may have associated with it a structure called its UVALUE. This structure is not used by IDL for anything, but may be used by the programmer to keep track of information relating to that WIDGET (or anything else, in principal). In the case of the display tool, its UVALUE is used to keep track of all information that must be shared between it and its children. For example, the name of the current image is stored in the UVALUE. With all information in this single structure, a subroutine need only be passed the name of the display base, and then it will have access to everything it needs to know. Rob Pina suggested this method of passing infromation as superior to "common blocks", which are another programming option.

IDL controls most of the characteristics of each WIDGET. Each of these can be set when the WIDGET is created in the program. A few characteristics, however, are controlled by the Window Manager for operating system, Open-Widnows in the case of the NIRSPEC computer, CRAB. Each window (or WIDGET) consists of several "resources" that can be controlled through an Open-Windows resources file, ".OWdefaults" or ".Xdefaults" (these files are effectively interchangable). IDL allows each WIDGET to be named, so that it may be identified in the resources file. Characteristics such as color and placement of WIDGETs, and hot-key assignment of menu items are controlled in the resources file.