

CCS-lite does not require HP RTAP and therefore does not allow to access the workstation based real-time database. It allows though to easily implement user interfaces to LCU software, providing access to LCU databases and support at the workstation level for passing messages, errors and logs. It represents also an easily portable platform to any users, who will be able to prepare an Observing programme ahead of time on a generic UNIX workstation.

Some developments, like the Real time display, although part of the VLT control software, have been implemented in a fully portable way and can run completely independent of the VLT software (see following article).

Documentation and Standards

A complete set of specifications and user manuals exists for the VLT software

and for the VLT common software in particular.

In spite of the large number of documents, there exist obvious "access roads" to documentation. So if you are interested in instrumentation software, you should simply start reading one document: VLT Instrumentation software specification. (Please refer to latest issue of 12/4/95.)

If instead the point of interest is telescope subsystems, the access specification is: VLT software – Telescope Control system functional specification.

These two documents can also be used as references to any other documentation, if deeper understanding is required.

The general requirements, standards and rules for this software have been fixed some time ago (although there is a constant evolution in the project) by the VLT software group.

All documents are available from the VLT Archive or directly via the ESO home page of the World Wide Web (main documents).

Meeting with Consortia and Contractors

A two-day meeting on the VLT control software was held at ESO on June 19–20, 1995. Software representatives of about 10 external Consortia and Contractors were present. The meeting was organised around a number of tutorials and demonstrations, as most of the software comes with a set of auxiliary tools and user interfaces, to facilitate development.

For more information on the VLT software please use the ESO home page in the WWW. Should you have additional questions, you can contact by e-mail either graffi@eso.org or other members of the VLT software group.

The VLT Real Time Display Software

T. HERLIN, A. BRIGHTON, P. BIEREICHEL, ESO

Introduction

The VLT will contain a large number of technical CCD's, which will be used for guiding and slit-viewing cameras requiring real-time display facilities. The frequency of image display from these image sources can be several Hz, for example, while performing tracking. For scientific detectors, real-time display capability is equally important. In particular, requirements for infrared detectors have been a driving force in specifying the functionality for a real-time display system.

The Real Time Display (RTD) software was developed in order to support image display in real-time on the VLT. The RTD software provides a tool for users to display video like images from a camera or detector as fast as possible on a workstation or X-terminal. The RTD is implemented as a package providing a widget and library. It is designed to be a building block, adding display capabilities to dedicated VLT applications in areas such as telescope and instrument control.

The intention of RTD is *not* to provide the image processing functionality already existing with image processing packages such as MIDAS. Functional overlapping with image processing systems is kept to a minimum and confined to the area of on-line operations.

Although it is part of the VLT control software, the RTD is an independent software package and does not require any other VLT software components, external packages, such as MIDAS or any commercial products.

Functionality

The core of RTD is an image display widget which supports two image sources: cameras and FITS files. The *camera* is either a technical CCD or scientific detector, which provides the image data in shared memory. The RTD widget is notified via an image event mechanism by the *camera* that a new image is available. See below for more about real-time images.

Figure 1 shows a typical screen layout when working with the RTD widget. The application shown was designed for demonstration purposes and shows an image loaded from a FITS file.

The user can change the magnification of the displayed image, zooming in to get a close-up view of a section of the image or zooming out to get an overall view of a large image. A panning window supports navigation on the image and a zoom window enables the targeting of a single pixel.

The RTD widget supports a basic set of colour scaling algorithms: linear,

square-root, logarithmic and histogram equalisation. MIDAS compatible colour maps are supported by the RTD widget as well as the MIDAS intensity transfer tables.

The colours can be manipulated by a colour bar rotating the colour map or changing the slope of the intensity via mouse interaction. The colour distribution can be changed by either manually setting the cut-levels or using automatic cut-level calculation.

The widget supports line graphics so that the user can overlay markers and text on the image. This might be used, for example, to identify interesting areas around a star. Standard line graphic components such as line, circle and text are available and can be drawn interactively on the image. Line graphic attributes such as line width, colour, filling, font, etc. can be set by the user via buttons and menus. In addition to the interactive line graphics, a programmatic interface is available to support overlaying of more complex line graphics, such as star maps taken from a catalogue.

Pixel query operations are supported at various levels:

- get and/or display the value of the pixel directly under the mouse.
- get and/or display a table of pixel values surrounding the pixel directly under the mouse.

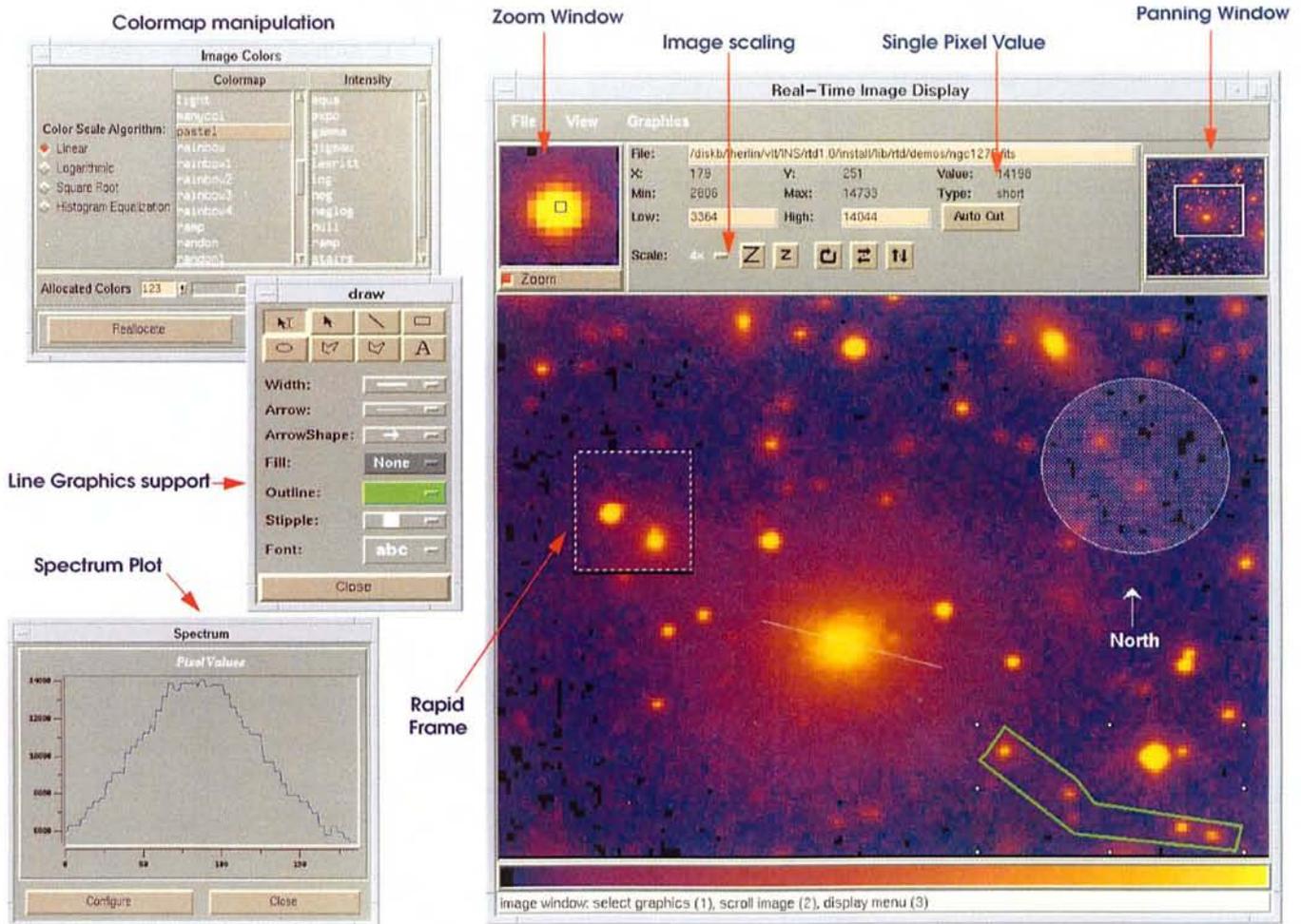


Figure 1.

Another level of pixel value inspection is supported by a spectrum plot along a *cut* line drawn interactively on the image. The pixel values are displayed in a separate graph window.

The RTD widget provides a library with basic image handling functionality referred to as the Real Time Image (RTI) library. In addition to basic image handling algorithms, the RTI library provides a data access framework for application developers supported by a well-defined application programming interface (API). This enables developers to contribute to and extend the image handling capabilities of the library.

Real Time Images

In order to use the RTD widget for real-time, an image event mechanism is provided by the RTD software. The image event is sent by the subsystem controlling the *camera* device e.g. CCD subsystem or infrared detector control software. The image event contains information about the image, for example, data-type, size and the reference to the shared memory location for the image data. The event is sent to an RTD server process which keeps a list of RTD applications registered for notification of a specific 'camera'. If there is a match, the image event

is dispatched to the corresponding RTD application.

Figure 2. shows a diagram of the image event architecture.

The advantage of using this decoupled approach is that the *camera* subsystems and the RTD applications become independent of each other and several RTD applications can attach to the same camera source.

The RTD server will also (currently not implemented) support eavesdropping which is the multi-casting of images to remote machines, which could be used, for example, for remote observing. For eavesdropping issues, such as network bandwidth and image compression techniques, e.g. H-transform will be investigated.

Updating images in real-time places a

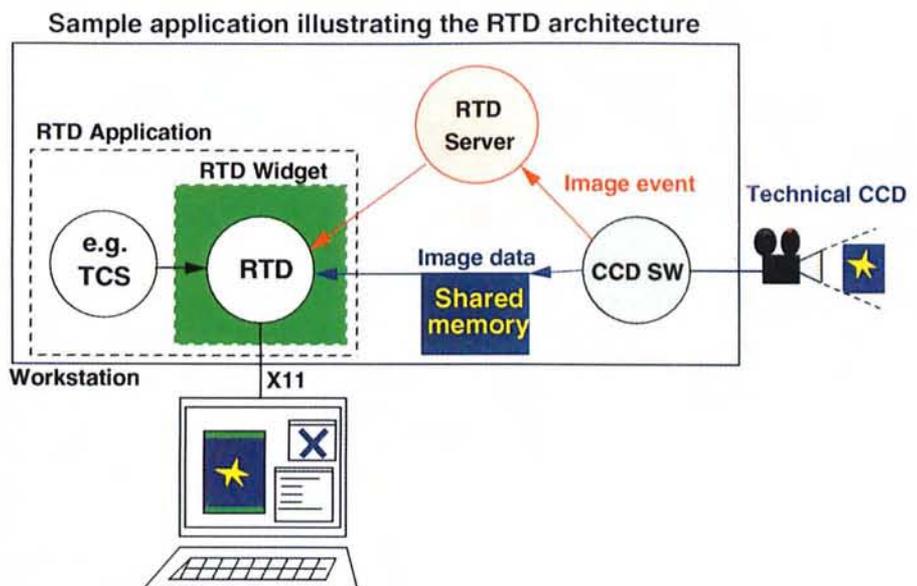


Figure 2.

high demand on the CPU and the network. For some images the variation between exposures is only interesting in a small area. In this case, a *rapid frame* can assist the selection of a specific area, for example around a star. This typically small frame can be updated faster than the rest of the image while the user still sees the complete picture. By using *rapid frames*, an image update frequency of more than 10 Hz is expected to be possible. Detailed performance measurements will be done based on the first release of RTD.

Implementation

The RTD widget is implemented as a Tcl/Tk (Tk 4.0) widget using C++. The RTI library is also implemented in C++ but as an independent library. In order to support fast application development with the RTD widget, a large set of [incr Tcl] widget classes is provided for managing, among other things, a zoom window, a panning window and a line graphics tool-box.

Although Tcl/Tk code is interpreted, the performance of the RTD widget when updating images in real-time is not degraded. This is due to the fact that all image related operations are implemented entirely in C/C++. Only the user interface parts of the software, such as buttons and menus are actually interpreted at run-time.

The image event mechanism uses standard UNIX IPC and image data is passed using shared memory as recommended by the real time extension standard POSIX.4.

The display on X-terminals is supported, although for very fast image display frequencies, workstations are recommended. In addition, on workstations the X11R5 extension for X Shared Memory is supported, giving the final boost for passing the scaled pixmap to the X-Server.

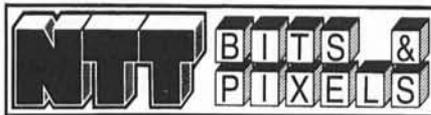
The software has been ported and tested on the following workstation platforms: HP (HP-UX) and Sun (SunOS & Solaris 2.3).

For RTD application development, the VLT Panel Editor can be used to develop control panels using the RTD widget. The VLT Panel Editor supports the interactive placement of GUI components and generates panels that comply with the ESO VLT common conventions for GUI development.

Availability

The RTD software is part of the VLT control software and the first release is available on the July '95 release of the VLT SW. For people interested and not receiving the VLT control software, the RTD user manual is accessible on anonymous ftp via the following WWW reference: ftp://te1.hq.eso.org/vlt/pub/doc/rtd_sum1.0.ps.gz.

For more information on the RTD software please contact: Thomas Herlin – ESO VLT SW group preferably via e-mail: therlin@eso.org.



(With this periodically compiled collection of short notes, the NTT Team intends to keep the community informed about changes in performance, configuration, and operation of the NTT and its subsystems.)

Big Bang Rescheduled

Because of a possible uncertainty in the manpower available in 1996 for the implementation of the new VLT-like control system, the date when the implementation will start has been shifted from April 1 to July 1, 1996. This means that the operation of the NTT will continue unchanged through June 30. Final adjustments will take place after field tests in December 1995 (Telescope Control System) and February 1996 (EMMI control software) of two critical components. Observers who might get time after June 30 will be informed about the allocation of observing time, if any, after the evaluation of the test results. This change in schedule does not seriously compromise the NTT's ability to provide useful feedback to the VLT project.

Further announcements will be made in *The Messenger* and on the World Wide Web (see below).

The NTT on the World Wide Web

As part of ESO's efforts to make more extensive use of the World Wide Web to inform the scientific community (which was strongly encouraged by the Users

Committee), NTT pages are under development. By the time of publication of this issue of *The Messenger*, they should be accessible from the ESO home page (<http://http.hq.eso.org/eso-homepage.html>). In addition to general information about the NTT and the NTT Upgrade Project, any up-to-date announcements will from now on be made on the Web. The previously maintained electronic bulletin board will be discontinued. The Information Desk of the general ESO pages can also be used for inquiries with the NTT Team (e-mail can as before be sent also directly to ntt@eso.org).

Update of EMMI/SUSI Manual

A major revision is currently under way of the EMMI/SUSI manual. In order to let prospective applicants for observing time with the NTT in Period 57 take advantage of the progress made, intermediate versions will be made available electronically. Details will be provided on WWW (see above).

Automatic Guide Star Selection

The Data Handling and Observation Support Group has installed the HST

guide star catalog and a new server on one of the workstations in the NTT building. The performance is for a number of reasons vastly superior to the previous temporary solution which depended on STARGAT in Garching. Guide stars are now on a routine basis automatically supplied for every new preset. First attempts to acquire also the guide star in a fully automatic way with the guide probe have been very successful. Since the light of the guide stars is via a beam splitter also used for parallel image analysis, efforts to adjust the exposure times with the image analyser according to the magnitude of the star are under way.

Graphical Log of Image Analysis Data

With the implementation of the cyclic image analysis in parallel mode, the self-diagnosing capabilities of the NTT can now be much better exploited. In order to enable also Visiting Astronomers to keep track of the performance of the telescope, tools have been developed to plot individual aberrations as measured before and after the setting of the forces versus time. In addition, the level to which aberrations should ideally be corrected is indicated.