#

NAME

ximtool-alt – interactive image display program for the X Window System with experimental GUI

SYNOPSIS

ximtool-alt [–*toolkitoption* **...] [** *options* **...]**

OPTIONS

-basePixel *N*
The base colormap cell used by the colormap. This essentially allows you to reserve *basePixel* colors in the global colormap for other applications. The default is 64, if changed you'll need to also specify the *-cmapInitialize* option or resource.

-cmap1 *file*
User colormap 1. This flag allows you to specify a colormap to be made available at task startup.

-cmap2 *file*
User colormap 2. This flag allows you to specify a second colormap to be made available at task startup.

-cmapDir1 *dir*
User colormap directory 1. Specifies a directory to be searched for colormaps.

-cmapDir2 *dir*
User colormap directory 2. Specifies a directory to be searched for colormaps. By default this points to the system directory /usr/local/lib/imtoolcmap, allowing a set of site default colormaps to be defined here.

-cmapInitialize *bool*
Initialize the ximtool colormap at startup. When setting the *basePixel* option or resource this is required in order to force the Gterm widget to update its global colormap resource in the X server. The default is *false*.

-cmapName *name*
Name used for private colormap. The default for all IRAF imaging applications is *image*. Gterm widget based imaging applications which have the same value of cmapName will share the same colormap, minimizing colormap flashing and allowing multiple applications to be run at the same time.

-config *N*
Initial frame buffer configuration number. The default value is 1, indicating a 512x512 frame buffer with 2 frames. See below for information on the frame buffers.

-defgui
Print the default GUI to the stdout. The GUI is a Tcl program that may be customized by the user and reloaded using the *-gui* option or the *gui* resource parameter.

-displayPanner *bool*
Display panner marker window at startup. If set, a panner window showing the full frame buffer will appear in the upper-right side of the main display window.

-displayMagnifier *bool*
Display magnifier marker window at startup. If set, a magnifier window showing a zoomed section around the cursor will appear in the upper-left side of the mail display window.

-displayCoords *bool*
Display WCS coordinate marker window at startup. If set, a coordinate readout text marker showing will appear in the lower-right side of the main display window.

-fifo *pipe*
Specifies the name of the fifo pipe to be used, the *i* and *o* suffixes will be added automatically. The default pipe names will be /dev/imt1i (input pipe) and /dev/imt1o (output pipe).

-fifo_only
If set, only fifo pipes will be used for communication with a client program, sockets will be disabled.

-gui *file*
Specifies the GUI file to be used.

-help Print a summary of command line options to the screen.

-imtoolrc *file*
Specifies the frame buffer configuration file to be used. See below for information on frame buffers.

-inet_only
If set, only inet sockets will be used for communication with a client program, fifo pipes and unix sockets will be disabled.

-invert
Start XImtool using inverted colormaps. When set, a "normalized" display will always be the inverse of the selected colormap.

-maxColors *N*
Specify the max number of colors to be used for the display.

-memModel *type*
Determines how ximtool uses memory in the ximtool client and the X server. The options are *fast*, *beNiceToServer*, and *small*. The default is *fast*, which uses server pixmaps to make frame blink fast. This is recommended unless server memory is very limited. Note that even in fast mode, the server pixmap is only the size of the display window, so memory usage is reasonable even if the frame buffer is very large.

-nframes *N*
Specifies the number of frame buffers to configure at startup. By default there will be 2 frames available, a maximum of 4 frames are allowed.

-port *N*
Specifies the port number to use when connecting through an inet socket.

-port_only
Same as *-inet_only* option. If set, only inet sockets will be used for communication with a client program.

-printConfig *name*
Specifies the printer configuration file to use. By default this will be /usr/local/lib/ximprint.cfg. See below for more information on configuring output devices.

-showToolBar
Show the Toolbox menubar at startup.

-showPanelBar
Show the Panels menubar at startup.

-tile The default display mode is to view one frame at a time. In tile frames mode, 2 or 4 frames may be viewed simultaneously in the display window. All the usual operations (zoom and pan, colortable enhancement, cursor readback, etc.) still work for each frame even when in tile frames mode.

-unix *name*
Specifies the unix domain socket name to use. A "%d" in the filename will be replaced with the user id.

-unix_only
If set, only unix domain sockets will be used for communication with a client program, inet sockets and fifos will be disabled.

RESOURCES

XImtool is implemented as a client program which is responsible for loading the frame buffers/colormaps, communicating with clients, etc, and a user-modifiable GUI file written as a Tcl script which handles all the user interface details. The *client resources* described below will be common to any user-defined GUI, the *gui resources* may change depending on how extensively the GUI has been modified by the user. Each of these components has its own set of resources, but to the user setting them is the same as with any other application.

Gterm widget resources (i.e. those for the main image window or colorbar) may be set as either client or GUI resources. See the *xgterm(1)* man page for a complete description of *Gterm* widget resources.

CLIENT RESOURCES

The client resources generally define the initial state of the application or set configuration parameters.

Resource Name	Default Value
defConfig	1
defNFrames	0
tileBorderWidth	3
tileBorderColor	9
autoscale	false
antialias	false
antialiasType	boxcar
tileFrames	false
highlightFrames	true
gui	default
imtoolrc	/usr/local/lib/imtoolrc
invert	false
memModel	fast
basePixel:	64
maxColors:	216
cmapInitialize:	false
cmap1	none
cmap2	none
cmapDir1	none
cmapDir2	/usr/local/lib/imtoolcmap
input_fifo	/dev/imt1i
output_fifo	/dev/imt1o
unixaddr	/tmp/.IMT%d
port	5137

Description of ximtool client resources:

defConfig Default frame buffer configuration number on startup. See below for more information on frame buffers.

defNFrames Default number of frames on startup. Set to zero to use the value from the frame buffer configuration (*imtoolrc*) file.

tileBorderWidth
tileBorderColor Used by the tile frames option. Specifies how far apart to space the frames in tile frames mode. Color "9" refers to the Gterm widget resource color9, which is assigned a color with its own resource.

autoscale Enable/disable the autoscale option.

antialias Enable/disable the antialias option.

antialiasType Type of antialiasing.

tileFrames Enable/disable the tile frames option.

highlightFrames Determines whether the current frame is highlighted when in tile frames mode.

gui The GUI to be executed. "default" refers to the default, builtin ximtool GUI. You can replace this with your own GUI file if you are bold enough, and completely change the look and functionality of the GUI if desired.

imtoolrc Where to find the imtoolrc file. This defines the recognized frame buffer configurations.

invert Start Ximtool using an inverted colormap. When set, a "normalized" display will always be the inverse of the selected colormap.

memModel Determines how ximtool uses memory in the ximtool client and the X server. The options are "fast", "beNiceToServer", and "small". The default is fast, which uses server pixmaps to make frame blink fast. This is recommended unless server memory is very limited. Note that even in fast mode, the server pixmap is only the size of the display window, so memory usage is reasonable even if the frame buffer is very large.

basePixel
maxColors These two resources determine the region of colormap space used to render image pixels.

cmapInitialize Initialize the ximtool colormap at startup. This is sometimes necessary to clear a previous ximtool colormap allowing a new basePixel and maxColors to take effect.

cmap1
cmap2 User colormap files. The intent here is to allow individual colormaps to be conveniently specified as a resource.

cmapDir1
cmapDir2 User or system colormap directories. By default cmapDir2 points to the system directory /usr/local/lib/imtoolcmap, allowing a set of site default colormaps to be defined here. This leaves cmapDir1 available to a user colormap directory.

input_fifo
output_fifo The input and output fifos for fifo i/o. "Input" and "output" are from the client's point of view. Note that only one display server can use a fifo-pair at one time.

unixaddr Template address for unix domain socket. The user must have write permission on this directory, or the file must already exist. %d, if given, is replaced by the user's UID.

port TCP/IP port for the server. Note that only one server can listen on a port at one time, so if multiple ximtool servers are desired on the same machine, they should be given different ports.

GUI RESOURCES

In principle ximtool can have any number of different GUIs, each of which defines its own set of resources. GUIs typically define a great many resources, but most of these are not really intended for modification by the user (although one can modify them if desired).

The following are some of the more useful resources used by the default ximtool GUI. The *imagewin* resources are Gterm widget resources.

Resource Name	Default Value
.geometry:	
*controlShell.geometry:	
*info.geometry:	420x240
*load_panel.geometry:	
*save_panel.geometry:	
*print_panel.geometry:	

*help_panel.geometry:	
*cmapName:	image
*basePixel:	64
*imagewin.warpCursor:	true
*imagewin.raiseWindow:	true
*imagewin.deiconifyWindow:	true
*imagewin.ginmodeCursor:	circle
*imagewin.ginmodeBlinkInterval:	500
*imagewin.color0:	black
*imagewin.color1:	white
*imagewin.color8:	#7c8498
*imagewin.color9:	steelblue
*imagewin.width:	512
*imagewin.height:	512
*autoscale:	True
*zoomfactors:	1 2 4 8
*displayCoords:	True
*displayPanner:	True
*displayMagnifier:	False
*showToolBar:	False
*showPanelBar:	False

As a display server, XImtool is started as a separate process from client software such as IRAF. Once it is running it will accept client connections simultaneously on fifo pipes, unix domain sockets, or inet sockets. A display client like the IRAF *DISPLAY* task makes a connection and sends the image across using an IIS protocol. Once the image is loaded in the display buffer it may be enhanced, saved to a disk file in a number of different formats, or printed as Encapsulated Postscript to a printer or disk file. Up to four frame buffers are allowed, these may be displayed simultaneously in a tiled mode, or blinked frame-to-frame. Each frame may have its own colormap or brightness/contrast enhancement. Pan/Zoom and cursor readout are permitted using *markers*, on-line help is also available.

When run in standalone mode, images (currently IRAF OIF, GIF, Sun Rasterfiles or simple FITS formats are permitted) may be loaded on the command line or by using the Load Panel. This allows you to browse images and perform the same manipulations as if they had been displayed by a client.

GUI OVERVIEW

The GUI consists of a large image display window and a number of smaller pannels that control various specific functions such as image Load, Save and Print as well as a general purpose Control Panel. The main window menubar has several menu buttons to the left: the *Files* menu is used to load/save/print an image as well as quit the task. The *View* menu let's you select the image orientation, zoom, colormap or frame. The *Options* menu allows you to call up control panels, toggle markers or blinking etc. Some of this functionality is duplicated elsewhere in the GUI.

The right side of the menubar contains command buttons to flip the image as well as buttons for frame selection. The *Toolbox Button* is labelled with a 'T', when this is enabled a second menubar appears below the main one containing a number of command buttons providing quick access to functions otherwise found elsewhere in the GUI. From the left these buttons include:

+ symbol	- zoom in
Magnify	- set zoom factor 1
- symbol	- zoom out
Inv	- Invert contrast
Norm	- Normalize colormap
Match	- Match LUTs
Reg	- Register
Cntr	- Center frame
< arrow	- decrease blink interval

Blink	- Toggle frame blink
> arrow	- inrease blink interval
<-> symbol	- X-flip and Y-flip
|+| symbol	- Tile Frame toggle
< arrow	- previous frame
<number>	- select frame
> arrow	- next frame

The image flip and frame selection buttons are also moved from the main menubar to provide more space for the image title. Next to the toolbox toggle is the *Control Panels* button which operates in a similar manner. When enabled a second menubar appears with more buttons: on the left side are two icons used as accelerators for a disk save (the floppy icon) and print function (the printer icon), the parameters used for these operations are those which have seen set through their respective control panel or the task resources. The middle two sections of buttons are toggles which manage the control panels for each function or the main imagewindow markers. Finally a help and a quit button for the task. By default these two extra menubars are disabled to provide more screen space for the image, they are controlled by the task ∗*showToolBar* and ∗*showPanelBar* resources or the *-showToolBar* and *-showPanelBar* command line flags.

For more detailed information on the operation of the control panels please see the on-line help (i.e. use the '?' button or Alt-h keystroke in the main image window).

MOUSE OPERATIONS

Clicking and dragging MB1 (mouse button 1) in the main image window creates a rectangular region marker, used to select a region of the image. If you do this accidentally and don't want the marker, put the pointer in the marker and type DELETE or BACKSPACE to delete the marker. With the pointer in the marker, MB3 will call up a marker menu listing some things you can do with the marker, like zoom the outlined region. MB1 can be used to drag or resize the marker. See below for more information on markers.

Clicking on MB2 in the main image window pans (one click) or zooms (two clicks) the image. Further clicks cycle through the builtin zoom factors. Moving the pointer to a new location and clicking moves the feature under the pointer to the center of the display window. Holding down the Shift key while clicking MB2 will cause a full-screen crosshair cursor to appear until the button is released, this can be useful for fine positioning of the cursor.

MB3 is used to adjust the contrast and brightness of the displayed image. The position of the pointer within the display window determines the contrast and brightness values. Click once to set the values corresponding to the pointer location, or click and drag to continuously adjust the display.

KEYSTROKE ACCELERATORS

The following keystrokes are currently defined in the GUI:

Ctrl-b	Backward frame
Ctrl-c	Center frame
Ctrl-f	Forward frame
Ctrl-i	Invert
Ctrl-m	Toggle magnifier
Ctrl-n	Normalize
Ctrl-p	Toggle panner
Ctrl-r	Register
Ctrl-s	Match LUT scaling
Ctrl-t	Tile frames toggle
Ctrl-u	Unzoom (zoom=1)

Ctrl-x Flip X
Ctrl-y Flip Y

Alt-b Blink frames (toggle)
Alt-c Control panel (toggle)
Alt-h Help popup (toggle)
Alt-i Info box popup (toggle)
Alt-l Load file popup (toggle)
Alt-p Print popup (toggle)
Alt-s Save popup (toggle)
Alt-t TclShell popup (toggle)

Ctrl-Alt-q Quit
Ctrl-Alt-f Fitframe

Ctrl-= Print using current setup
Ctrl-< Decrease blink rate (blink faster)
Ctrl-> Increase blink rate (blink slower)
Ctrl-+ Zoom in
Ctrl-- Zoom out

Ctrl-[hjkl] or <arrow_key>
Move cursor one pixel in each direction
Ctrl-Shift-[hjkl] or Shift-<arrow_key>
Move cursor ten pixels in each direction
Ctrl-<arrow_key>
Move one full panner frame in each direction
Ctrl-Alt-<arrow_key>
Move one half panner frame in each direction
Alt-1 thru Alt-4
Set frame displayed
Ctrl-1 thru Ctrl-9
Set integer zoom factor

NOTE: These keystrokes only work with the cursor in the main image window, not on the subwindows or in markers since they are implemented as *imagewin* translations. If a command does not work, check the cursor location.

CLIENT CONNECTIONS

XImtool allows clients to connect in any of the following ways:

fifo pipes

The traditional approach. The default global /dev/imt1[io] pipes may be used, or a private set of fifos can be specified using the *-fifo* command line argument or ∗*fifo* resource. Values should be specified as the root pathname to a pair of fifo pipes whose last character is ’i’ or ’o’, these characters will be added automatically when opening the pipes. For example, to use the default pipes the path would be specified as simply "/dev/imt1". A value of "none" disables this connection.

tcp/ip sockets

Clients connect via a tcp/ip socket. The default port is *5137*, or a custom port may be specified using the *-port* command line switch or a ∗*port* resource. This permits connecting to the server over a remote network connection anywhere on the Internet. A port number of 0 (zero) disables this connection.

unix domain sockets

Like a tcp/ip socket, but limited to a single host system. Usually faster than a tcp/ip socket, and comparable to a fifo. By default each user gets their own unix domain socket, so this option allows multiple users to run ximtools on the same host without having to customize things. The default value is "/tmp/.IMT%d", other sockets may be defined using the *-unix* command line switch or the ∗*unixaddr* resource. Legal values should be specified as a filename to be used for the socket, up to two "%d" fields are allowed and will be replaced by the userid. An empty string value disables this connection.

By default ximtool listens simultaneously for client connections on all three types of ports. Clients may connect simultaneously by different means allowing up to three different displays to be loading at the same time into different frames.

COMMUNICATIONS PROTOCOL

Clients communicate with XImtool using a protocol developed originally for IIS (International Imaging Systems) Frame Buffer hardware, the so-called "IIS protocol"; other more modern protocols will likely be supported in the future. The IIS protocol is basically a command packet stream with a header describing the operation to be performed (select frame, load display, read cursor, etc), and an optional data packet containing e.g. pixels. It is beyond the scope of this document to describe fully the details of the protocol; interested users should contact *iraf@noao.edu* for further information.

FRAME BUFFERS

XImtool starts up using default frame buffer size of 512x512 pixels, two (of four possible) frames will be created. When loading disk images (i.e. run in standalone mode) the frame buffer configuration file will be searched for a defined frame buffer that is the same size or larger than the current image, if no suitable buffer can be found a custom frame buffer the same size as the image will be created in an unused portion of the configuration table. When used as a display server the frame buffer configuration number is passed in by the client and loaded explicitly even if it means clipping the image. If a new frame buffer is a different size than previously defined frames, all available frames will be initialized and cleared prior to the display. The default frame buffer configuration file is /usr/local/lib/imtoolrc, this can be overridden by defining a IMTOOLRC environment variable naming the file to be used, by creating a .imtoolrc file in your home directory, or a new file may be specified using the *-imtoolrc* command line flag or *imtoolrc* application resource.

The format of the frame buffer configuration file is

configno nframes width height [extra fields]

e.g.

```
1  2  512  512
2  2  800  800
3  1 1024 1024          # comment
:  :   :    :
```

At most 128 frame buffer sizes may be defined, each configuration may define up to 4 frames, configuration numbers need not be sequential.

NOTE: When defining a new frame buffer for use with client software such as IRAF the user must also remember to define those frame buffers in the IRAF *dev$graphcap* file.

MARKERS

Although ximtool doesn't do much with markers currently, they are a general feature of the *Gterm* widget and are used more extensively in other programs (e.g. the prototype IRAF science GUI applications). XImtool uses markers for the marker zoom feature discussed above, and also for the panner, magnifier and the coordinates box. All markers share some of the same characteristics, so it is worthwhile learning basic marker manipulation keystrokes.

- **o** MB1 anywhere inside a marker may be used to drag the marker.
- **o** MB1 near a marker corner or edge, depending on the type of marker, resizes the marker.
- **o** Shift-MB1 on the corner of most markers will rotate the marker.
- **o** Markers stack, if you have several markers and you put one on top of the other. The active marker is highlighted to tell you which of the stacked markers is active. If the markers overlap, this will be marker "on top" in the stacking order.
- **o** MB2 in the body of a marker "lowers" the marker, i.e. moves it to the bottom of the stacking order.
- **o** Delete or backspace in a marker deletes it.
- **o** Markers have their own translation resources and so the default keystroke commands will not be recognized when the cursor is in a marker.

For example, try placing the pointer anywhere in the coords box, then press MB1 and hold it down, and drag the coords box marker somewhere else on the screen. You can also resize the coords box by dragging a corner, or delete it with the delete or backspace key. (The Initialize button will get the original coords box back if you delete it, or you can reset the toggle in the control panel).

PANNER MARKER

The panner window always displays the full frame buffer. Try setting the frame buffer configuration to a nonsquare frame buffer (e.g. imtcryo) and then displaying a square image (e.g. dev$pix) and the panner will show you exactly where the image has been loaded into the frame.

The panner window uses two markers, one for the window border and one to mark the displayed region of the frame. Most of the usual marker keystrokes mentioned below apply to these markers as well, e.g. you can use MB1 to reposition on the panner window within the main image display window, or to drag the region marker within the panner (pan the image). Resizing the region marker zooms the image; this is a non-aspect constrained zoom. The panner window itself can be resized by dragging a corner with MB1. Typing delete or backspace anywhere in the panner window deletes the panner.

A special case is MB2. Hitting MB2 anywhere in the panner window pans the image to that point. This is analogous to hitting MB2 in the main display window to pan the image.

The panner marker can be disabled by defining the *displayPanner* GUI resource, its size and location can be controlled using the *pannerArea* and *pannerGeom* GUI resources respectively.

MAGNIFIER MARKER

The magnifier marker can be used to zoom in on a small area around the cursor. It will be updated as the cursor moves but only for small motions (either mouse movement or with the cursor movement keystrokes) to minimize the impact on the system. The zoom factor is expressed as some fraction of the size of the magnifier marker itself. The default zoom is 4, i.e. the area in the marker represents and area in the image that's one-fourth the size of the marker. Other zoom factors may be selected using the popup menu created by hitting MB1 in the marker.

By default the magnifier marker is not visible, to toggle it select the *Magnifier* option from the *Options* menubar button. Alternatively, for just a quick look holding down the Shift and MB1 buttons will display the marker until the button is released.

The magnifier marker can be disabled by defining the *displayMagnifier* GUI resource, its size and location can be controlled using the *magnifierArea* and *magnifierGeom* GUI resources respectively.

COORDS BOX MARKER

XImtool provides a limited notion of world coordinates, allowing frame buffer pixel coordinates and pixel values to be converted to some arbitrary linear client-defined coordinate system. The coords box feature is used to display these world coordinates as the pointer is moved about in the image window.

The quantities displayed in the coords box are X, Y, and Z: the X,Y world coordinates of the pointer, and Z, the world equivalent of the pixel value under the pointer. All coordinate systems are linear. The precision of a displayed quantity is limited by the range of values of the associated raw frame buffer value. For example, if the display window is 512x512 only 512 coordinate values are possible in either axis (the positional precision can be increased however by zooming the image). More seriously, at most about 200 pixel values can be displayed since this is the limit on the range of pixel values loaded into the frame buffer. If a display pixel is saturated a "+" will be displayed after the intensity value.

The coords box is a text marker, it can be moved and resized with the pointer like any other marker. The coords box marker can be disabled by defining the *displayCoords* GUI resource, its location can be controlled by the *wcsboxGeom* GUI resource.

MARKER MENU OPTIONS

Except for the panner and WCS markers, MB3 (mouse button 3) calls up the marker menu providing a limited set of functions common to all markers:

- Zoom does an equal aspect zoom of the region outlined by the marker. In this way you can mark a region of the image and zoom it up.
- Fill exactly zooms the area outlined by the marker, making it fill the display window. Since the marker is not likely to be exactly square, the aspect ratio of the resultant image will not be unitary.
- Print prints the region outlined by the marker to the printer or file currently configured by the Print Panel.
- Save saves the region outlined by the marker to the file currently configured by the Save Panel.
- Info prints a description of the marked region. The text is printed in the Info Panel.
- Unrotate unrotates a rotated marker.
- Color is a menu of possible marker colors.
- Type is a menu of possible marker types. This is still a little buggy and it isn't very useful, but you can use it to play with different types of markers.
- Destroy destroys the marker. You can also hit the delete or backspace key in a marker to destroy the marker.

CONTROL PANEL

XImtool has a control panel which can be used to exercise most of the capabilities the program has for image display. The control panel can be accessed either via the **Options** menu from the main window menubar, or by pressing the leftmost button in the row of buttons at the upper right side of the display.

VIEW CONTROLS

The **Frame box** will list only the frame buffers you currently have defined. Currently, the only way to destroy a frame buffer is to change the frame buffer configuration, new frame buffers (up to 4) will be created automatically if requested by the client. The number of frame buffers created at startup can be controlled using the *-nframes* command-line switch or the *defNFrames* resource.

The **text display** window gives the field X,Y center, X,Y scale factors, and the X,Y zoom factors. The scale factor and the zoom factor will be the same unless *autoscale* is enabled. The scale is in units of display pixels per frame buffer pixel, and is an absolute measure (it doesn't matter whether or not autoscale is enabled). Zoom is relative to the autoscale factor, which is 1.0 if autoscaling is disabled. This information is also presented in the Info panel.

The numbers in the **Zoom box** are zoom factors. Blue numbers zoom, red numbers dezoom. *Zoom In* and *Zoom Out* may be used to go to larger or smaller zoom factors, e.g. "Ctrl-5" followed by "Zoom In" will get you to zoom factor 10. Specific zoom factors may also be accessed directly as Control keystrokes, e.g. Ctrl-5 will set zoom factor 5. *Center* centers the field. *Toggle Zoom* toggles between the current zoom/center values, and the unzoomed image.

Aspect recomputes the view so that the aspect ratio is 1.0. Aspect also integerizes the zoom factor (use the version in the View menu if you don't want integerization).

Fit Frame makes the display window the same size as the frame buffer. Note that autoscale has much the same effect, and allows you to resize the display window to any size you want, or view images too large to fit on the screen.

ENHANCEMENT CONTROLS

At the top is a scrolled list of all the available colormaps. Click on the one you want to load. You can add your own colormaps to this list by defining the *cmap[12]* or *cmapDir[12]* command line flags or application resources.

The two sliders adjust the contrast (upper slider) and brightness (lower slider) of the display. The *Invert* button inverts the colormap (multiples the contrast by -1.0). Note that due to the use of the private colormap the sliders are a bit sluggish when dragged to window the display. If this is annoying, using MB3 in the display window is faster.

The *Normalize* button (on the bottom of the control panel) will normalize the enhancement, i.e. set the contrast and brightness to the default one-to-one values (1.0, 0.5). This is the preferred setting for many of the pseudocolor colortables and for private colormaps loaded from disk images. The *Initialize* button does a reset of the server.

BLINK CONTROLS

Blink frames is the list of frames to be blinked. When blink mode is in effect ximtool just cycles through these frames endlessly, pausing "blink rate" seconds between each frame. The same frame can be entered in the list more than once. To program an arbitrary list of blink frames, hit the Reset button and click on each blink frame button until it is set to the desired frame number.

The *Blink Rate* can be adjusted as slow or as fast as you want using the arrow buttons. If you set the blink rate small enough it will go to zero, enabling single step mode (see below).

The *Register* button registers all the blink frames with the current display frame. Frames not in the blink list are not affected.

The *Match LUTs* button sets the enhancement of all blink frames to the same values as the display frame. Frames not in the blink list are not affected.

The *Blink* button turns blink on and off. When the blink rate is set to zero the Blink button will single step through the blink frames, one frame per button press.

NOTE: You can blink no matter what ximtool options are in effect, but many of these will slow blink down. To get the fastest blink you may want to turn off the panner and coords box, and match the LUTs of all the blink frames. All the ximtool controls are fully active during blink mode, plus you can load frames etc.

OPTIONS:

Panner

Toggles whether to display the panner marker.

Coords Box

Toggles whether to display the coordinate box marker.

Autoscale

If autoscale is enabled then at zoom=1, the frame buffer will be automatically scaled to fit within the display window. With autoscale disabled (the default), the image scale is more predictable, but the image may be clipped by the display window, or may not fill the display window.

Antialiasing

When dezooming an image, i.e., displaying a large image in a smaller display window, antialiasing causes all the data to be used to compute the displayed image. If antialiasing is disabled then image is subsampled to compute the displayed image. Antialiasing can prevent subsampling from omitting image features that don't fall in the sample grid, but it is significantly slower than dezooming via subsampling. The default is no antialising.

Tile Frames

The default display mode is to view one frame at a time. In tile frames mode, 2 or 4 frames may be viewed simultaneously in the display window. All the usual operations (zoom and pan, colortable enhancement, cursor readback, etc.) still work for each frame even when in tile frames mode.

Warnings

The warnings options toggles whether you see warning dialog boxes in situations like overwriting an existing file, clearing the frame buffer, etc.

COLORMAP SELECTION

By default XImtool will display images using either a grayscale colormap (e.g. if loaded by a client), or a private colormap when loading an image from disk that contains a colormap. Each frame defines its own colormap so you can define different colormaps or enhancements for each frame, they will change automatically as you cycle through the frames.

BUILTIN COLORMAPS

Once loaded, the colormap may either be changed using the builtin colormap menu under the View menu button on the main window, or from the Enhancement box on the control panel. XImtool has about a dozen colormap options builtin, other user-defined colormaps may optionally be loaded. It is not presently possible to save colormaps for later use.

USER-DEFINED COLORMAPS

The *cmap[12]* and *cmapDir[12]* resources (or command line arguments) are used to tell which specific colormaps to make available or where to look for colortables respectively. The colortables are loaded when ximtool starts up, or when it is reinitialized (e.g. by pressing the Initialize button in the control panel). XImtool will ignore any files in the colormap directory which do not look like colortables. New colortables will also be added automatically for each image loaded from disk.

The format of a user lookup table is very simple: each row defines one colortable entry, and consists of three columns defining the red, green, and blue values scaled to the range 0.0 (off) to 1.0 (full intensity).

```
R G B
R G B
(etc.)
```

Blank and comment lines (lines beginning with a '#') are ignored.

Usually 256 rows are provided, but the number may actually be anything in the range 1 to 256. XImtool will interpolate the table as necessary to compute the colortable values used in XImtool. XImtool uses at most 201 colors to render pixel data, so it is usually necessary to interpolate the table when it is loaded.

The name of the colortable as it will appear in the XImtool control panel is the root name of the file, e.g., if the file is "rainbow.lut" the colortable name will be "rainbow". Lower case names are suggested to avoid name collisions with the builtin colortables. Private colormaps for disk images will be have the same name as the image loaded. If the same colortable file appears in multiple user colortable directories, the first one found will be used.

MINIMIZING COLORMAP CONFLICTS

The Gterm widget used by XImtool (i.e. the main display window) uses a private global colormap for display, this allows it to have greater control over color cell allocation but can occasionally also cause "colormap flashing" as the mouse is moved in and out of the application. The problem here is that in a system with only an 8-bit colormap (256 colors) all applications must compete for colors, programs such as XV or Netscape allocate colors from the default colormap leaving only a few free cells for XImtool. Since XImtool defines a private global colormap it is still able to allocate the needed cells rather than failing, but it's allocating cells already used by other applications. As the mouse moves out of the ximtool window those cells are once again defined in terms of the default colormap, so the ximtool window is then using a different colormap. It is this switching of the colormap context that causes the flashing to occur, but there are a few things that can be done to help minimize this.

XImtool logically defines 200 colors which the client image display program can use to render pixels. However, ximtool may or may not actually allocate all of those colors. By default it currently allocates only about 192 colors, to reserve 64 colors for the other windows on the screen. You don't normally notice this as 1) usually the default screen colormap has enough free cells to allow ximtool to match the colors, and 2) the extra unallocated cells correspond to the brightest pixels in the rendered image, and these colors may not be used or usually only correspond to a few small regions near the saturated cores of bright objects.

You can eliminate this problem by setting the *basePixel* resource to e.g. 48 instead of 64, which will let the gterm widget allocate all 200 colors. However, this isn't recommended for normal use as it will increase the likelihood of colormap flashing. If you change *basePixel*, either restart the X server or set the resource *cmapInitialize=True* to force the gterm widget to update its global colormap resource in the X server. The colormap resource may also be deleted by using the command

xprop -root -remove GT_image

These options may also be set on the command line when first starting up.

In general one can set the Gterm widget resources *basePixel* and *maxColors* to specify the region of colormap space to be used for image display. If you set *maxColors* to a small value, the 200 logical colors defined by the widget will be mapped by the imtool color model into whatever number of colors are actually available to the widget. For example, in the default setup, 200 color values are really being mapped into 192 color cells used for display, the remaining colors are used for buttons, menus etc and are allocated from the default colormap by the X toolkit when the application starts up.

Even though the Gterm widget uses a private colormap, it is a private *global* colormap meaning that all Gterm widgets share the same colormap. An example of colormap sharing in ximtool is the main image window and the colorbar window. These are two separate gterm widgets that share the same colormap. They have to share the same colormap, as otherwise when you windowed the main image window the colorbar window would not accurately reflect the modified colormap. By default two separate ximtools would also share the same colormap meaning contrast enhancements in one window would affect the other. By resetting the *cmapName* command line option or resource you can change the name of the private colormap used causing separate ximtools to use different colormaps, but note this also creates colormap flashing between the two windows that cannot easily be avoided. By setting the *cmapName* to "default" the widget will allocate colors from the default colormap, but this is of little use at the moment.

There are a number of other resources that can be used to modify the behavior of the Gterm widget color management scheme, but these are the most useful ones. For question and further information feel free to contact *iraf@noao.edu*.

LOAD PANEL

The Load Panel allows you load images from disk directly to the frame buffer, this is analogous to loading an image on the command line except that browsing is possible. At present recognized formats include IRAF OIF format (i.e. *.imh* extension), simple FITS files, GIF, and Sun rasterfiles. The task will automatically sense the format of the image and load it appropriately. Images with private colormaps (such as GIF) will be loaded using the private colormap (meaning that changing the brightness/contrast enhancements will render a random-colored image), all others will be loaded with a grayscale colormap. If the *Grayscale* button is enabled the image colormap will be converted to grayscale and loaded as the standard grayscale colormap. The Load panel will close automatically once the image has loaded unless the *Browse* button has been set.

When loading new images the frame buffer configuration table will be searched for a frame buffer that is the same size or larger than the new image size, if no frame buffer can be found a custom buffer exactly the size of the image will be created. This means that the image may not fill the display window when loaded, or you may see a subsection of the image in the main display window. Setting the *autoscale* option will scale the entire image to fit the main display window, the full frame buffer will always be visible in the Panner marker window.

Images with more colors than can be displayed will automatically be quantized to the number of available colors before display, 24-bit formats are not currently supported (but may be in the future and will be similarly quantized).

Formats which permit larger than 8-bit pixels will be sampled on a grid to determine an optimal range in the data to be used to compute a linear transformation to the number of display colors. This is the same sampling and transformation used by the IRAF *DISPLAY* task when computing the *z1/z2* values and provides a much better initial display than simple truncation to 8-bits.

Directory Browsing

The load panel contains a list of files in the current directory that may be selected for loading by selecting with left mouse button. If the file is a directory the contents of the new directory will be loaded, if it's a plain file an attempt will be made to load it as an image otherwise an error popup will appear. Directories in the list are identified with a trailing '/' character, you will always see any subdirectories listed even if a filter is specified.

The *Root* button will reset the current directory to the system root directory. The *Home* button will reset the current directory to the user's login directory, the *Up* button moves up one directory level, and *Rescan* reloads the file list by rescanning the directory. The current working directory is given below the file selection window.

File Patterns

By default all files and directories will be listed. You may specify a filter to e.g. select only those files with a given extension like "*.fits" to list only files with a ".fits" extension. Directories will always be seen in the list and are identified with a trailing '/' character. Any valid unix pattern matching string will be recognized.

Direct File Load

If you know exactly which file you wish to load, you may enter its name in the *Load File* text box and either hit <cr> or the Load button to load it. An absolute or relative path name may be given, if a simple filename is specified it will be searched for in the current working directory.

Frame Selections

By default images will be loaded into frame number 1, you may select a different frame using the Frame menu button to cycle through the available frames.

SAVE PANEL

The Save Panel lets you save the current contents of the main display window to a disk file (including the Panner/Coords markers, any general graphics markers, or overlay graphics displayed by the client program). Presently, only the contents of the main display window may be saved, there is no facility for saving the undisplayed contents of the entire frame buffer other than to enable the autoscale feature. A limited number of formats are currently available, others will be added in future versions.

File Name

The File Name text box allows you to enter the file name of the saved file. A "%d" anywhere in the name will be replaced by a sequence number allowing multiple frames to be saved with unique names.

Format

The Format box allows you to choose the format of the image to be created. Not all formats are currently implemented.

Color

The Color box lets you choose the color type of the image to be created. The options will change depending on the format, e.g. FITS doesn't allow color so no color options will be allowed. Formats which allow 24-bit images will be written using the current colormap after converting to a 24-bit image, pseudocolor images will be written with the current colormap.

PRINT PANEL

The Print Panel allows you dump the contents of the main display window as Encapsulated Postscript to either a named printer device or to a disk file. The *Print To* selects the type of output, the *Print Command* box will adjust accordingly, either as a Unix printer command or as a file name. A "%d" anywhere in the name for disk output will be replaced by a sequence number allowing multiple frames to be saved with unique names. Selecting printers from the installed list will automatically change the command to be used to generate the output. This command does not necessarily need to be a printer command, the printer configuration file lets you define any command string to process the image.

COLOR OPTIONS

The Color box lets you choose the color type of the image to be created. PseudoColor or 24-bit postscript will be created using the current colormap and enhancements.

POSTSCRIPT OPTIONS

Orientation

Set the page orientation.

Paper Size

Select the paper size to be used.

Image Scale

Set the scale factor used to compute the final image size. No checking is done to make sure the image will fit correctly on the page.

PROCESSING OPTIONS

Auto Scale

Toggles whether or not the image is automatically scaled to fit the page. If not enabled, the image scale will be used to determine the output image size, otherwise the image will be scaled down (if necessary) to fit on the page.

Auto Rotate

Determines whether or not the image will be rotated to fit on the page. When set, an image larger than the current orientation will be rotated and possibly scaled to fit the page, otherwise the image may be scaled so that it fits in the current orientation.

Max Aspect

Automatically increases the scale so the image fills the page in the current orientation.

Annotate

The annotate option toggles whether or not the final file includes annotation such as the image title, a colorbar, and axis labels. There is currently no option for partial annotation.

ANNOTATION OPTIONS

Annotate

Selects whether Postscript image is to be annotated. **Title** Annotate with a title on the top of the image. **Borders** Annotate with borders surrounding the image giving image coordinates. **Colorbar** Annotate with colorbar at the bottom of the image **Title String** Title string to use when *title* is selected. The special value *imtitle* will force the title to be the currently displayed image title, otherwise it will be this user-selected field.

PRINTER SELECTION

The printer selection list lets choose the printer to be used. The printer configuration file is /usr/local/lib/ximprint.cfg by default or may be reset using the *-printConfig* command line switch or *printConfig* resource. The format of the file is simply

name\tcommand

The *name* value is what appears in the selection list and may be more than a single word, the *command* can be any command that accepts EPS input from a pipe, the two fields must be separated by a tab character. Normally the command will be a simple *lpr -Pfoo* or some such, but can also include converters or previewers. At most 128 printer commands may be used.

INFO PANEL

The information panel is underused at present but is meant to provide basic information about the frame being displayed. It is updated to be current while changing enhancements, pan/zoom regions, or frame selection. In cases where the image title string is truncated in the main display window, the user can always pop up the info window to see the full title.

TCLSHELL

The *TclShell* allows the user to type commands directly to the TCL interpreter, letting you send messages to the object manager or execute specific procedures in the TCL code that makes up the GUI. It is used as a development or debugging tool for the GUI, but for an example of what it does, bring it up and type a command such as

send helpButton set background red

ENVIRONMENT

DISPLAY	specifies which display terminal to use
IMTOOLRC	frame buffer configuration file
imtoolrc	frame buffer configuration file (alternative)

FILES

/usr/local/lib/imtoolrc	default frame buffer configuration file
/usr/local/lib/ximprint.cfg	default printer configuration file
/usr/local/lib/imtoolcmap	default colormap directory
/dev/imt1i	default input fifo
/dev/imt1o	default output fifo
/tmp/.IMT%d	default unix socket

BUGS

SEE ALSO

xgterm(1), xtapemon(1)

COPYRIGHT