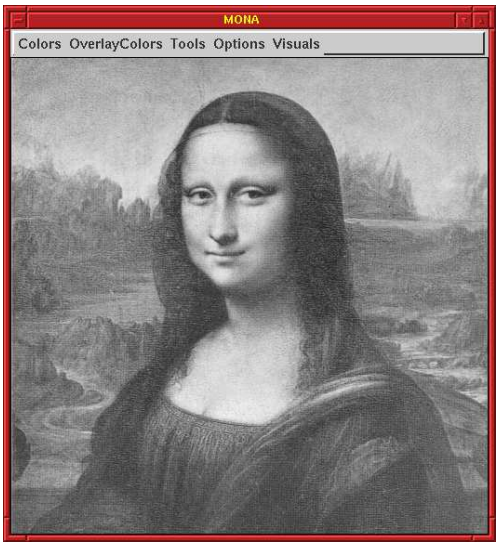
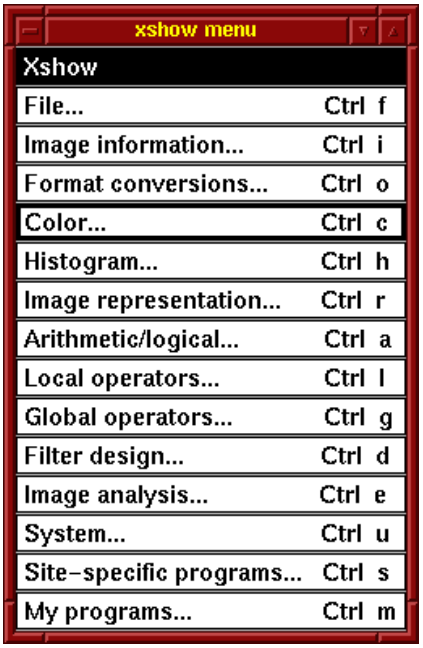
Image Processing Laboratory
Department of Informatics
University of Oslo

Report No. 56





XITE

X-based Image Processing Tools and Environment

User's Manual

For version 3.48

Svein Bøe, Tor Lønnestad and Otto Milvang

September 2013

Bildebehandlingslaboratoriet
Institutt for informatikk
Universitetet i Oslo

Image Processing Laboratory
Department of Informatics
University of Oslo

Tittel/Title:

XITE User's Manual

Forfatter(e)/Author(s):

Svein Bøe, Tor Lønnestad and Otto Milvang

Rapport nr./Report no.: **56, 4th ed.** ISBN: **82-7476-061-1** Dato/Date: **September 2013**

Resymé/Abstract:

XITE consists of display programs with image widget and graphical user interface as well as more than 200 command line programs and 600 subroutines for image processing, all documented on-line.

The command line programs and subroutine library are written in C and run under UNIX and Windows.

The display programs run under UNIX. They work with images of arbitrary size and pixel type on 8-bit PseudoColor and 24-bit DirectColor and TrueColor X11 displays. Images can be zoomed and panned, and colortables can be selected from a menu. The main display program, xshow, gives access to most of the other command line programs via a menu interface which the user can customize and extend to include local programs. Input images for the menu entries can be selected with the mouse, and output images appear on the display.

This report describes how to use the display and command line based programs, the image format, the documentation system, and how to prepare a user account for XITE.

Norske emneord/Indexing terms - Norwegian:

Bildebehandling
Bildebegrep
Bildeprogram
Vindussystemet X
Farger
UNIX
C
Windows

Engelske emneord/Indexing terms - English:

Image Processing
Image Concept
Display Program
X Window System
Colors
UNIX
C
Windows

Adresse:

Bildebehandlingslaboratoriet
Institutt for informatikk
Universitetet i Oslo
Boks 1080 Blindern
0316 Oslo

epost: blab@ifi.uio.no
tlf: 22 85 24 10

Address:

Image Processing Laboratory
Department of Informatics
University of Oslo
P. O. Box 1080 Blindern
N – 0316 Oslo
NORWAY

email: blab@ifi.uio.no
phone: +47 22 85 24 10

XITE

X-based Image processing Tools and Environment

User's Manual

For version 3.48

Image Processing Laboratory
Department of Informatics
University of Oslo

Svein Bøe, Tor Lønnestad and Otto Milvang

September 2013

Contents

List of Figures

List of Tables

List of Examples

Chapter 1

Introduction

XITE (pronounced *excite*) is an acronym for "X-based Image processing Tools and Environment". The non-display programs run under UNIX (System V as well as BSD versions) and MS Windows. The display programs are based on the X Window System, Version 11, currently using release 5, 6 or 7, operating in a UNIX environment.

- Linux (all programs)
- Windows (not display programs)

The purpose of the system is to help you perform image processing in the broad sense, image to image operations as well as image to description operations (image analysis). Currently, though, the image to image operations are outnumbering the image analysis operations.

"Tools and Environment" means that the system contains a set of tools which you can use to create new modules — small routines or large applications — as well as a complete environment ready to perform image processing for you.

1.1 About this report

This report describes

- how to use the on-line documentation system
- the image concept used in XITE
- the use of the xshow program in detail, both as a display program and as a graphical user interface
- how to activate image programs directly from the shell/command window, one program at a time, as well as several in a pipe sequence
- how to write image processing shell scripts
- how to generate shell scripts (or macros) from xshow

This report does *not* describe how to use the XITE function library or how to implement your own algorithms according to XITE standards, neither does it describe how to install the system. Two

Item	Examples of appearance
Program, function, widget names	xshow, man
File and directory names	`xshow_colortabs`
Shell/command window command lines	`§ xshow mona.img`
Environment variables	`XITE_HOME`
Command line options	`-width`
Submenu name	*File...*
Menu entry	*Read image*
Pushbutton widget	*Quit*
Mouse button	`<Btn1>`
Keyboard key	`Ctrl`

Table 1.1: Typographic conventions.

other manuals handle this in detail, the [1, Programmer's Manual] and the [2, System Administrator's Manual]. You should read this report carefully before opening Programmer's Manual.

XITE has also been described elsewhere [6, 4, 7].

If you're anxious to see examples of what XITE can do, skip to section 2.3. For detailed information on how to navigate in the xshow menu hierarchy, refer to section 2.5.1.

1.1.1 Typographic conventions

Table 1.1 shows how various kinds of items will appear throughout this text. For shell/terminal/command window command lines, "§" represents the prompt.

1.2 User interaction levels

There are several ways to use the system. The highest level is through xshow, which is a display program as well as a menu based interface to most of the other programs. The user can activate other programs in the library, send along input images, parts of images, and other arguments. When a program is finished, output images are sent back to xshow and immediately displayed. Output text is sent to a text window which may be saved in a file. xshow does not run under Windows, unless the X Window System is also installed.

Apart from xshow, a number of tailored display programs are included with XITE. These provide user interfaces which are especially convenient for certain tasks, such as region analysis, animation, filter design and FFT. These display programs do not run under Windows.

The non-display programs are available as commands to the command window under Windows and to a UNIX shell. A single program may be started, while specifying all input and output arguments. If any argument is missing, the program will complain and print a usage message. A sequence of programs may also be started, piping images through the programs.

The tools of the system can be used to build new modules. The tools contain three main conceptual parts: A set of image processing routines implementing basic image processing algorithms, a set of image handling routines for administrative manipulation of disk and memory images (and particularly transfer between disk and memory), and finally some widgets and widget tools which simplify the creation of window based image applications. The widgets and widget tools are not available under Windows.

1.3 XITE user group and contact information

A mailing list has been established to enable contact between the users of the XITE system. If you want to join or leave the list, or if you have questions or other feedback to the XITE developers, please send this in email to *xite-request@ifi.uio.no*. The name of the mailing list is *xite@ifi.uio.no*. Thus, to communicate with other users of the system, send mail to *xite@ifi.uio.no*. Bug reports should preferably be sent to *xite-bugs@ifi.uio.no*.

XITE has its own home page on the World Wide Web, at

`http://www.mn.uio.no/ifi/english/research/groups/dsb/resources/software/xite/`

Information about new versions, known bugs and documentation can be found here.

Chapter 2

Getting started

This chapter briefly covers the basics needed to start using XITE, in particular how to set up the environment and a short introduction to the main display program xshow, which is also a graphical user interface to the program library. More information on xshow can be found in chapter 4.

Throughout this documentation, the value of an environment variable will be shown in a UNIX manner, i.e. with a $ prefix for the variable name, like `$XITE_HOME`. Under Windows, the correct way to get the value of an environment variable is to surround the variable name by a pair of %, like %`XITE_HOME`%.

Also, the notation for folders/directories in this documentation uses the slash delimiter ("/"). For Windows, replace the slash by a backslash ("\").

2.1 Setup and environment variables

Before starting to use the XITE system, some environment variables should be initialized properly. This may already have been done for you by your local XITE administrator. In section 2.1.1 we show you the easy way to do it when XITE has been installed properly, in appendix A we explain the meaning of the environment variables in detail.

2.1.1 Recommended setup

2.1.1.1 On UNIX platforms

Installed with a Red Hat rpm If the XITE rpm has been installed, files are located according to FHS (FileSystem Hierarchy Standard for Unix and Linux). If the computer is configured to run XITE scripts when booting, everything should be set up properly.

If the XITE stup scripts are not run automatically during boot-time, run the script /etc/init.d/xite with the `source` command.

Installed with a tarball The most straightforward and recommended way of initializing the required environment variables, is to invoke XITE setup files from the initialization files of your login shell. If your login shell is `csh`, insert the command

```
source $XITE_HOME/etc/xite_cshrc
```

in your `$HOME/.cshrc` file, where `$HOME` represents your home directory, and `$XITE_HOME` should be replaced by the path to the XITE home directory.

Table 2.1 can be used to determine the shell initialization file and what command to insert in this file, depending on what login shell you have.

Login shell	Initialization file	Command
csh, tcsh	`$HOME/.cshrc`	`source $XITE_HOME/etc/xite_cshrc`
sh, bash	`$HOME/.profile`	`. $XITE_HOME/etc/xite_profile`

Table 2.1: Shell initialization file and XITE setup command depending on login shell. Replace `$XITE_HOME` by the actual path to the XITE home directory.

Your system manager will hopefully maintain the XITE setup files and modify them if necessary.

Log out and in again for the changes to take effect.

2.1.1.2 On Windows

Refer to the next section, on manual setup.

2.1.2 Manual setup

2.1.2.1 On UNIX platforms

To just quickly try out some XITE capabilities, you may choose not to edit your shell initialization files, but rather set the required environment variables manually. This should be done in every command window from which you want to invoke XITE commands.

The following commands use csh syntax and may be different for other shells. Assuming a SunOS computer platform, the commands may be

```
§ setenv XITE_HOME /usr/local/xite # Depends on the actual location
§ setenv XSHOWPATH $XITE_HOME/data/xshow
§ setenv PATH $XITE_HOME/bin:$PATH
§ setenv MANPATH $XITE_HOME/man:/usr/local/man:/usr/man
§ setenv XITE_DOC $XITE_HOME/doc
```

2.1.2.2 On Windows

From the Windows Control Panel, find the *System* entry, choose the *Advanced* tab and its *Environment variables* button. Set the environment variable `XITE_HOME` to reflect the folder in which XITE is installed on your system. Change the environment variable `PATH` so that the XITE programs will be found, i.e. add the folder name in which the XITE executable programs reside to this variable. This folder may be something like `$XITE_HOME`/bin/Win32[1]. Finally, set the environment variable `XITE_MAN` to the folder where the preformatted reference manual pages in ordinary text format are located. This is probably `$XITE_HOME`/man. The `XITE_MAN` variable is required for the general XITE program options `-man` and `-whatis` to work.

[1] This is UNIX notation. On Windows, the value of an environment variable is found by surrounding the variable name by a pair of %, instead of prepending the name with a $, and the folder separator is "\" instead of "/".

2.2 File naming conventions

If any part of the XITE system asks you for a file name — the name of a program file, an image file, a color table file, or any kind of file name — you should give the full name of the file, there is no standard file name extension added by the system. To avoid confusion, you should however standardize on file name extensions. We recommend `.img` for images[2], maybe `.biff` if you are using several file formats. BIFF means Blab Image File Format and is the format used by XITE, but XITE can also import and export a few other popular formats. A reasonable extension for color table file names seems to be `.col`.

The system accepts tilde, `~/file` for a file in your home directory, and `~friend/file` for a file in a friend's home directory. The latter is not available under Windows. The home directory is determined from the environment variable HOME, if set. Otherwise, on a UNIX system, the password database is enquired, and on a Windows system, the environment variable HOME or else HOMEPATH is used.

Image files in BIFF format may contain multi band images. If you want to process only a subset of the bands in a multi band image, you may specify this in the file name. Assuming that you want to display image bands with xshow, a few examples are given in example 2.1. The band specification separator is ":" on a UNIX system, ";" on a Windows system. The notation in the

Example 2.1 Notation for processing a subset of bands, in this case by using xshow.

`§ xshow imagefilename:2`	Process only second band (under UNIX)
`§ xshow imagefilename;2`	Process only second band (under Windows)
`§ xshow imagefilename:1:3`	Process first and third band
`§ xshow imagefilename:1-3`	Process first, second and third band
`§ xshow imagefilename:3-1`	Process third, second and first band
`§ xshow imagefilename:2:3-1:3-4:7`	Process bands 2, 3, 2, 1, 3, 4 and 7 (under UNIX)
`§ xshow imagefilename;2;3-1;3-4;7`	Process bands 2, 3, 2, 1, 3, 4 and 7 (under Windows)

example is only available when reading from an image file, not for writing. The general file name syntax is given in appendix B.

The BIFF format is described in some more detail in chapter 7.

2.3 Examples of using xshow

xshow is only available on UNIX systems.

For detailed information on how to navigate in the xshow menu hierarchy, refer to section 2.5.1.

2.3.1 Starting xshow

Assuming a UNIX system and that the required environment variables have been set according to the description in section 2.1.1, go to a directory where there are BIFF image files and start xshow. XITE comes with a few BIFF images which are located in the directory `$XITE_HOME/data/img`, so you may try

```
§ cd $XITE_HOME/data/img
§ xshow
```

[2] or perhaps a pixeltype specific extension such as `.ub` (unsigned byte), `.us` (unsigned short) or `.r` (real)

Figure 2.1: xshow Control window.

The xshow Control window shown in figure 2.1 and the main menu shown in figure 2.2 on the next page will be displayed[3]. If the main menu does not appear, it means that the menufile was not found. Without the menufile, xshow is severely limited. Make sure that you followed the initialization instructions in section 2.1 closely.

When you are ready to proceed, select the *Image information...* submenu from the main menu (by pushing and releasing `<Btn1>`[4] when the mouse pointer is inside the submenu label), and the *List image files* command in the submenu. The message field in the Control window should respond with

```
Job 12345 started - biffinfo -f .
```

or something similar. This means that xshow starts the program biffinfo with command line arguments "`-f .`". If any BIFF files are found in this directory, a separate text window appears (place it with `<Btn1>` if necessary), listing the BIFF files. Finally a text like

```
Job 12345 finished - biffinfo -f .
```

appears in the message field. The window labeled "xshow - terminal" can be removed by typing `q` inside it. Exit xshow by pushing *Quit* in the Control window.

2.3.2 Displaying an image

An image can be displayed by xshow by giving a BIFF filename as command line argument. Try the command

[3] A message appearing in the terminal window, starting with the string "SetShellArgs" and concerning the option `-share`, will be explained in section 4.7.2.1.

[4] usually the left mouse button

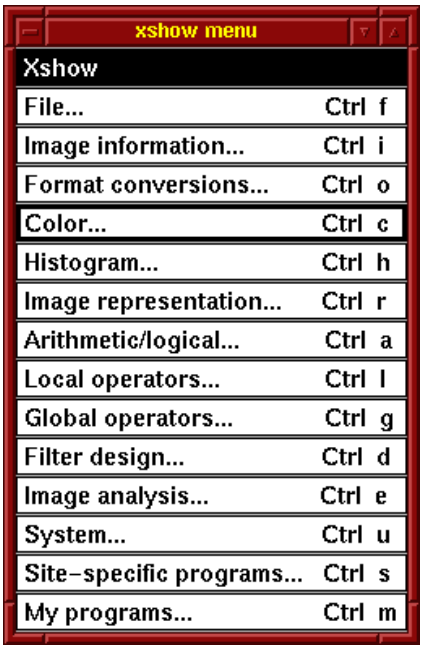


Figure 2.2: xshow main menu.

```
§ xshow lena.img
```

when the current directory is `$XITE_HOME/data/img`.

If the colors on the screen change radically when you move the mouse between windows, and you don't understand why, please refer to section 4.3.1 for an explanation.

You can also read images into xshow while the program is running. From the main menu window, choose the submenu *File...* and then the menu entry *Read image*. A file selector widget will pop up. If your current directory is `$XITE_HOME/data/img`, you should see a list of files on the right hand side. Choose `mona.img`. This can be done by pressing `<Btn1>` on the name `mona.img`. The name will be copied to the filename field in the widget. Then either push the *Select* button or hit the `Return` key inside the filename field. Place the resulting image widget with the mouse, if necessary.

A quicker way to choose an image from the fileselector widget, is to press `<Btn2>`[5] on the name `mona.img`. This will immediately read the BIFF image file and pop up an image widget.

2.3.3 Zooming and panning an image

An image can be zoomed in by holding down the `Ctrl` key and clicking `<Btn1>`, or by resizing the window using the mouse[6]. The zoom factor is displayed in the Control window. A zoom factor equal to one means that one dot on the screen corresponds to one image pixel. The image can be zoomed out by holding down the `Ctrl` key and clicking `<Btn3>`, or again by resizing the window. Clicking `<Btn2>` while holding the `Ctrl` key down will set the zoom factor equal to one (or as small as possible if the window is larger than the number of image pixels).

Larger zoom changes are available by holding down the `Shift` key in addition to the `Ctrl` key.

[5] usually the middle mouse button

[6] Refer to section 4.5.2 for explanation of how the image is resized, depending on whether a fixed aspect ratio should be maintained.

An image which is only partially visible inside its window can be panned with the arrow keys. Larger steps are taken if the `Shift` or `Ctrl` key is held down[7].

2.3.4 Histogram of an image

When you have `mona.img` displayed on the screen as described in section 2.3.2, choose the menu entry *Histogram* from the *Tools* menu in the menubar above the image. This will pop up a new window containing the histogram of the image. You can manipulate this[8] by pressing `<Btn1>` in the histogram part of this window and moving the mouse while holding the button down. Choose the type of transformation function from the *Actions* menu in the histogram widget, try e.g. *Linear*. The manipulations of the histogram will be reflected simultaneously in the original image.

You can remove the histogram window by typing `q` inside the histogram field or the color-map field (at the bottom).

Refer to section 4.4.2 for more information on manipulating the histogram. Also, section 5.4 describes a simple, specialized program, xhistogram, for display of image histograms only.

2.3.5 Fourier transform of an image

First, notice the appearance of the bottom button below the "-info-" field in the xshow Control window. It is labeled "Mouse : N" in a diffuse color. This means that the button is insensitive.

Now, with an image displayed on the screen, choose the submenu *Global operators...* from the main menu, then *Fft...* and finally the menu entry *Complex result* below the label "2D Forward". A dialog window will appear, expecting you to give the scaling of the FFT. Simply push the *Accept* button for now. The xshow Control window will display a text similar to

```
Job 13467 started - fft2d -r <infile> <outfile>
```

and the button in the Control window previously labeled "Mouse : N" will now be labeled "Mouse :JC", where "JC" is short for "Job Control". The button is now sensitive. This is the sign that xshow is waiting for you to tell what image should be sent to the FFT program. Choose an image by clicking `<Btn2>` in an image window. A new image window will pop up, but instead of displaying an image, it will contain a text telling you that the resulting image has a pixeltype which can't be displayed. xshow is unable to display the pixel values because the result of the Fourier transform has complex valued pixels[9].

To see the absolute value of the 2D FFT result, traverse the menu hierarchy again as described above, until you reach the submenu *Fft...*. Choose menu entry *Log(magnitude)*. Press *Accept* in the following dialog box, then click `<Btn2>` in the window showing the original image. This will supply the image as input to the FFT program, followed by a transformation which returns the logarithm of the magnitude. This final result is real valued, and a new window will pop up to display the result, with the zero frequency in the center of the window. The same result could be achieved by taking the logarithm of the complex-valued FFT image which xshow could not display properly, thus avoiding to recompute the FFT. To do this, choose the submenu *Arithmetic/logical...*, its *Logarithm* entry and select the complex-valued FFT image as input. In

[7] Some window managers catch these key/mouse events and use them for special purposes. In this case the action is not available for xshow, unless you change the window manager configuration.

[8] If the image is displayed with a visual class which does not allow changing the hardware color-map, the histogram window will be severely limited and not allow any manipulation. No *Actions* menu will be available. Refer to sections 4.3 and 4.6 for more information on color-maps and visual classes.

[9] The image can still be selected as input to other programs.

the resulting dialog window, select the field "Shift band by (xsize/2,ysize/2)" and push the *Accept* button.

Instead of passing a whole image along to a program, you may select a rectangular subpart by clicking and dragging `<Btn1>` in the image. A rectangular outline will show the selected area, and the position and size of the rectangle are updated continuously in the Control window.

The transformed images may be removed by typing `q` inside them.

Section 5.2 describes a specialized display program, xfft, for 2 dimensional FFT. This handles the Fourier transformation and the display function in a more user friendly manner.

2.3.6 Adding two images

Display two images in xshow, e.g. `mona.img` and `lena.img`. Choose the submenu called *Arithmetic/logical...* and the menu entry *Weighted sum*. Push the *Accept* button, then `<Btn2>` on the two images. A new window will pop up with the resulting sum.

Section 5.1 describes a specialized display program, xadd, for addition of two images. This program can interactively change the weight of the two images when the user manipulates a slider.

2.4 The Reference Manual

The Reference Manual, i.e. the documentation for each program and function, is available in hypertext format as well as man files (the man files are included preformatted with the prebuilt Windows distribution).

2.4.1 Reading the on-line hypertext Reference Manual

On UNIX, to check that the on-line hypertext Reference Manual is installed correctly, click the *Help* button in the xshow Control window. By default, this will try to start firefox. If successful, it will display a page with links to alphabetically sorted lists of all XITE programs and functions, as shown in figure 2.3 on the facing page. Also, a copy of the xshow menu hierarchy is shown. You may find it useful to move about in this hypertext hierarchy at the same time as you are navigating in xshow's menu hierarchy.

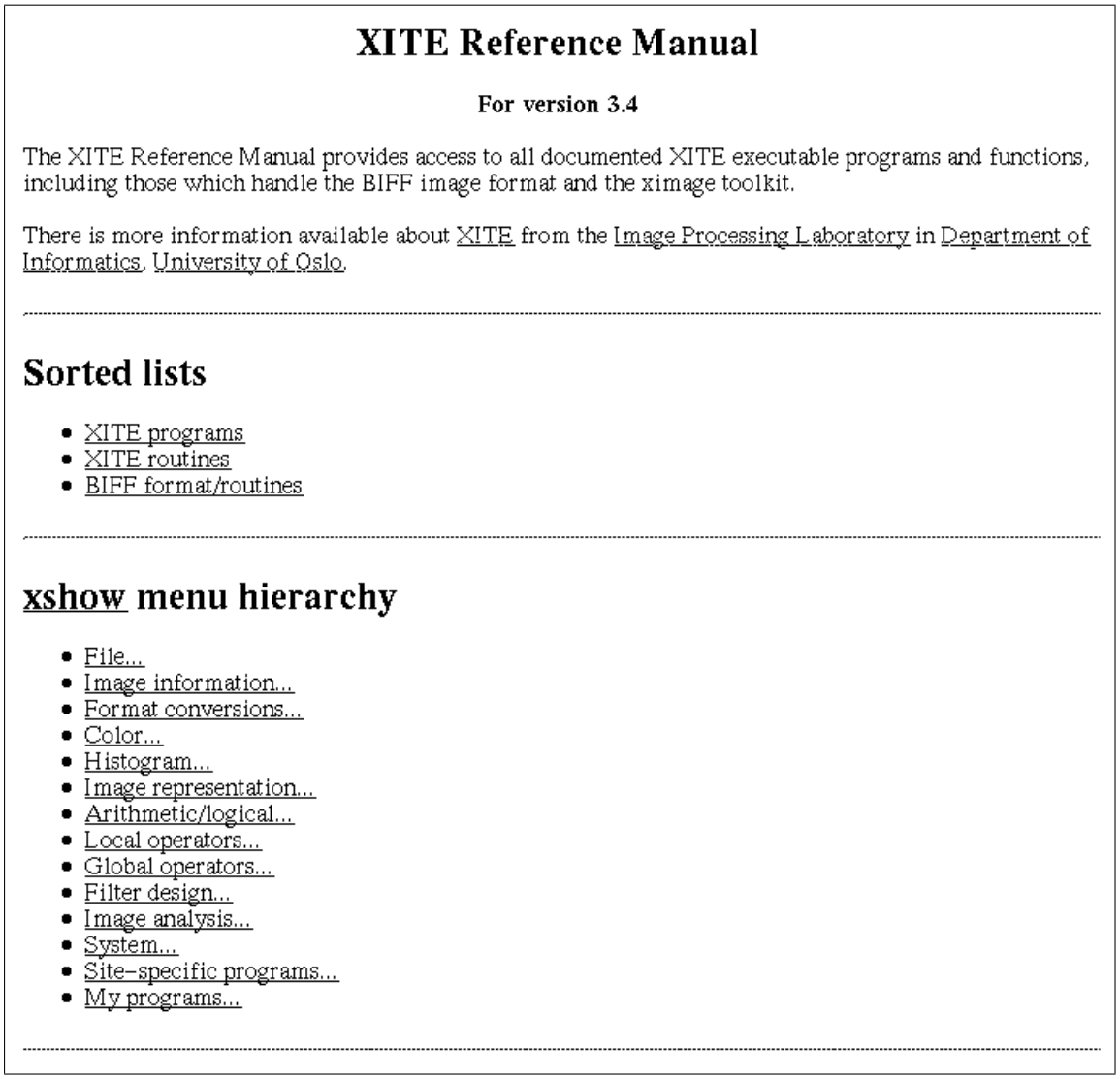
If xshow fails to launch firefox, you may choose to let xshow start a different program. Refer to the description of the environment variable `XITE_HELPER` in appendix A.

On all platforms, the hypertext Reference Manual can also be read by pointing your WWW browser to the file `$XITE_HOME/doc/ReferenceManual/Contents.html`[10]

2.4.2 man

To see if the UNIX man system works with the XITE Reference Manual pages, give the commands `man biffinfo`, `biffinfo -man` and `biffinfo -whatis`. The latter two should also work under Windows.

[10] `%XITE_HOME%\doc\ReferenceManual\Contents.html` in Windows notation.



XITE Reference Manual

For version 3.4

The XITE Reference Manual provides access to all documented XITE executable programs and functions, including those which handle the BIFF image format and the ximage toolkit.

There is more information available about XITE from the Image Processing Laboratory in Department of Informatics, University of Oslo.

Sorted lists

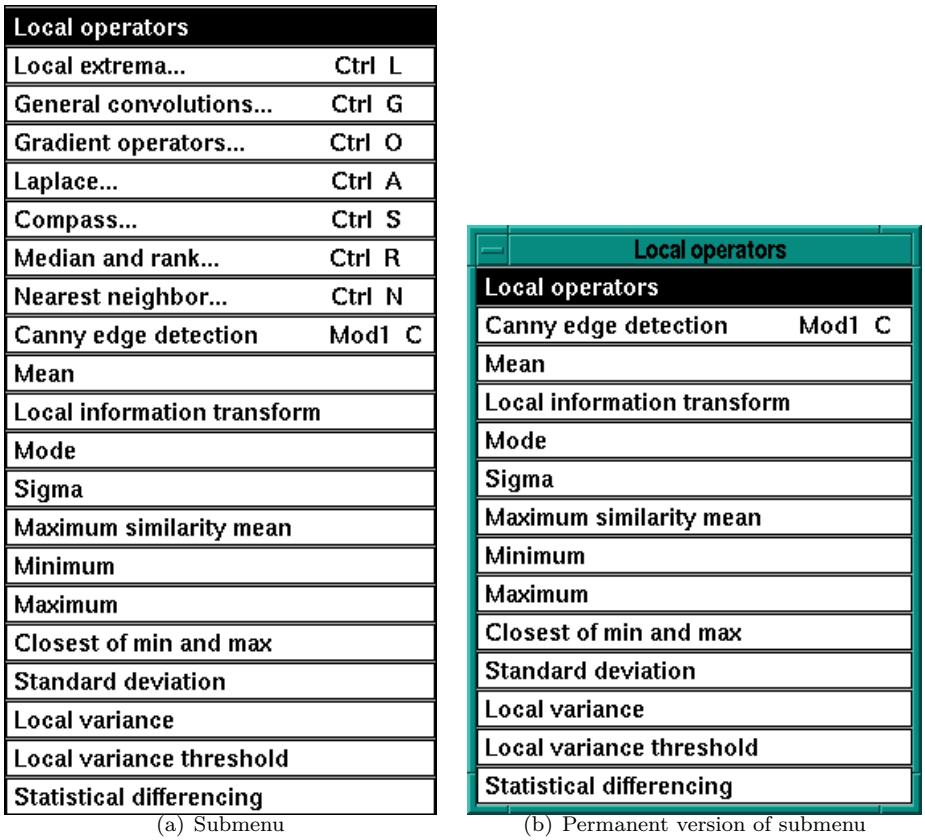
- XITE programs
- XITE routines
- BIFF format/routines

---

xshow menu hierarchy

- File...
- Image information...
- Format conversions...
- Color...
- Histogram...
- Image representation...
- Arithmetic/logical...
- Local operators...
- Global operators...
- Filter design...
- Image analysis...
- System...
- Site-specific programs...
- My programs...

Figure 2.3: XITE on-line Reference Manual front page.



(a) Submenu (b) Permanent version of submenu

Figure 2.4: *Local operators...* submenu and its permanent version.

2.5 Speeding up user interaction with xshow

A number of shortcuts are available for user interaction with xshow.

2.5.1 Navigation in the xshow menu hierarchy

The menus available from the main menu window in xshow may be manipulated in three different ways which are described below.

2.5.1.1 Navigation with the mouse only

Navigation with the mouse only is typically preferred by the novice user.

Start by moving the mouse pointer into the main menu window and position the pointer on the entry labeled *Local operators...*. The three trailing dots in the label signal that a submenu is available at this point. Push and release `<Btn1>`. The submenu shown in figure 2.4(a) will pop up.

If you move the mouse pointer towards the right hand side of the submenu, keeping it in the entry labeled *Local extrema...*, a new submenu will pop up. Moving the mouse down into the entry labeled *General convolutions...* will remove the submenu called *Local extrema...* and pop up a

submenu called *General convolutions...*.

Choose a command by clicking the mouse in a menu entry which is not a submenu.

A submenu may also be popped up by clicking the mouse instead of moving the mouse pointer over to the right hand side.

If you move the mouse pointer outside a submenu, it will usually disappear.

2.5.1.2 Using keyboard accelerators/shortcuts

The labels in the main menu all contain a right justified string starting with the letters "Ctrl" followed by an additional letter. If you move the mouse so that the pointer is inside the Control window (but not on a button), you may type e.g. `Ctrl f` (i.e. hold the key labeled `Ctrl` down and hit the `f` key). This will pop up the *File...* submenu where the mouse pointer is located. You can choose one of the menu entries as usual, by clicking the mouse over the chosen entry.

Most of the submenus in the menu hierarchy may be popped up with a keyboard accelerator (sometimes called a keyboard shortcut). They typically require some key combination including the `Ctrl` key.

In the submenu *File...*, some of the entry labels contain a right justified string starting with the letters "Mod1" followed by an additional letter. If you again move the mouse so that the pointer is inside the Control window (but not on a button), you may type e.g. `Mod1 r`[11] to read a BIFF image from file. This will pop up the file selector widget directly. Notice that there is no need to first pop up the submenu.

The majority of the command entries in the menu hierarchy can be executed via keyboard accelerators. They typically require some key combination including the `Mod1` key.

Each individual user can change the default keyboard accelerators. Refer to the Reference Manual for xshow and its description of X resources for details.

2.5.1.3 Permanent display of submenus

Extracts of the submenus can be placed on the screen for the duration of an xshow session. This can be done in two ways.

1. Pop up a submenu (either by traversing the menu hierarchy with the mouse pointer or by using a keyboard accelerator). Click `<Btn1>` when the pointer is in the menu titlefield (the top label).

2. Use the submenu keyboard accelerator without the `Ctrl` key.

Remember that the accelerators only work when the mouse pointer is inside the Control window (but not on a button). Either of the above alternatives will make an extract of the submenu and place it on the screen. It can be moved and iconified as any other window.

An example of a permanent submenu is shown in figure 2.4(b) on the facing page. It is an extract of the one shown in figure 2.4(a) on the preceding page. A permanent submenu will often not be complete. It contains only the command entries, not entries for other submenus.

[11] The `Mod1` key is labeled "Alt" on a Silicon Graphics Indy keyboard. Your computer may be different. The program `xmodmap` will tell you which key corresponds to the `Mod1` key on your keyboard. If no key is bound to `Mod1`, you may be able to bind a key with a command such as `xmodmap -e 'add mod1 = Alt_L'`.

Button	Accelerator
Abort	`q`
Accept	`Return`
Help	`h`

Table 2.2: Accelerators for dialog window buttons.

You may navigate inside the submenus with arrow keys or alternatively with `Ctrl n` and `Ctrl p` for moving to the next or previous entry respectively. Hitting the `Return` key will activate a menu entry.

A permanent submenu may be removed by hitting the `q` key while the mouse pointer is somewhere inside the submenu.

2.5.2 Control window accelerators

The three buttons in the leftmost column in the xshow Control window have their own accelerators. The *Macro* button is activated with `Mod2 m`[12], the *Help* button is activated with `Mod2 h` and the *Quit* button is activated with `Mod2 q`.

The Control window is described in more detail in section 4.2.

2.5.3 Dialog window accelerators

In the dialog windows which pop up for some of the programs in the xshow menu hierarchy, there are three buttons, *Abort*, *Accept* and *Help*. The accelerators for these buttons are shown in table 2.2. For *Abort* and *Help*, the mouse pointer must be positioned on the dialog window background, not on any input fields or buttons.

[12]The `Mod2` key is labeled "Num Lock" on a Silicon Graphics Indy keyboard. Your computer may be different. The program `xmodmap` will tell you which key corresponds to the `Mod2` key on your keyboard. If no key is bound to `Mod2`, you may be able to bind a key with a command such as `xmodmap -e 'add mod2 = Num_Lock'`.

Chapter 3

Help and documentation

The help and documentation system is divided in two parts, general information and reference documentation on each program and function. The User's Manual, which you are currently reading, is an example of the former, while the hypertext Reference Manual (consisting of `html` files) is an example of the latter. All the documentation is available on-line in the directory given by the environment variable `XITE_DOC` (by default the same as `$XITE_HOME/doc`[1]). It can of course be printed.

3.1 General documentation

The following three manuals are classified as general documentation

- XITE User's Manual [3]
- XITE Programmer's Manual [1]
- XITE System Administrator's Manual [2]

The purpose of this XITE User's Manual is explained in the introduction: To give end users the basic information needed to start using XITE.

The purposes of the XITE Programmer's Manual are to describe

- the BIFF file and memory format in detail, and the routines available to utilize these formats
- how to write new modules which use existing modules in the system
- how to call ordinary image processing operations
- the data structures and routines provided for region analysis
- how to create X based image processing applications, built with the ximage toolkit supplied with XITE (on UNIX systems only)
- how to write new modules that are intended to be a part of the system itself

The last manual, XITE System Administrator's Manual, is intended for those, at every site, that are responsible for the local installation of XITE. The manual describes the installation and how to recompile parts of or the whole system.

[1] `%XITE_HOME%\doc` on Windows

3.2 Reference documentation

For every program and most functions in the system, there is a hypertext and man page description of its purpose and usage. There are several ways to access this information. Some of the tools you can use are

- Your favourite WWW browser (Firefox, Internet Explorer etc)
- man (UNIX only)
- standard XITE program options (described in section 3.3)

Point your browser to

```
$XITE_DOC/ReferenceManual/Contents.html
```

for the first page of the reference manual. This provides two possibilities for looking up reference information. The first is via lists of either XITE programs, XITE routines or BIFF format and routines The information in the man pages is the same as in the hypertext Reference Manual. The second possibility is through a hierarchy which is similar to the menu hierarchy in the xshow display program.

On a UNIX system, if you are running xshow and a WWW browser at the same time, this is a convenient way to learn what each program is doing.

The browser will most likely allow text search in the reference documentation of each program or function. A search mechanism can also be set up to allow full text searches in the complete Reference Manual. Ask your system administrator about this.

Documentation about the latest version of XITE is of course also available on-line from its authors, at the following address

```
http://www.mn.uio.no/ifi/english/research/groups/dsb/resources/software/xite/
```

3.2.1 man

man is the traditional text based interface to the manual pages on a UNIX system. In its simplest form, the command line

```
§ man <name>
```

prints the man page for `<name>` to the screen.

3.2.2 Printing reference documentation

3.3 Program help options

Some options are generally available in all XITE programs, on UNIX as well as Windows platforms.

`-help`, `-usage`: Gives the usage for the program and terminates.

`-whatis:` Gives the one line description of the program and terminates.

`-man:` Gives the same as `man prog`. This works also under Windows, although `man` is only available under UNIX.

`-verbose:` Makes some programs report the actions taking place.

In addition to these options, activating an XITE program with no arguments generally gives the usage message and terminates the program (with a few exceptions such as xshow).

3.4 Image history

Documentation on a completely different level is provided by the image history system, which aggregates a textual description of the processing steps applied to an image. Each processing step is documented by three items

- Date and time of the processing (usually when it was completed).
- The name of the program.
- Arguments to the program.

By looking at the history of an image — try

```
§ biffinfo -h <filename>,
```

you will see exactly what has happened to it, and you may recreate the process that ended up with this image.

Chapter 4

xshow

xshow is the main display program as well as a graphical user interface to the program library. Other display programs (xadd, xfft, xfilter, xhistogram, xmovie, xpyramid and xregion) are described in chapter 5. None of the display programs are available under Windows, unless the X Window System is already installed.

The typical command line[1] for starting xshow is

```
§ xshow <image>
```

As seen in section 2.3, xshow starts by displaying a control window, a menu window and optionally one or more image windows. The image data are read from BIFF files, described in chapter 7. Information on how to zoom and pan an image was given in section 2.3.3.

Several popular image formats may be imported to and exported from xshow, such as TIFF, PNG, pnm, raw binary, ascii, Sunraster and MATLAB. Images may also be printed on a PostScript printer.

When an image is displayed with xshow, the colors of the image may change as the mouse pointer is moved in and out of the image window. This behaviour, which depends on the capabilities of your display hardware and X server, is explained in section 4.7.

4.1 Running a program from xshow

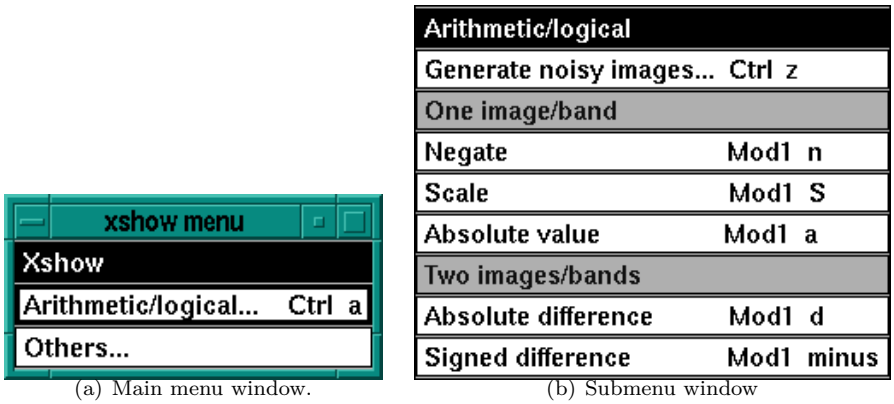
Section 2.3 explained how to start a program from the menu hierarchy, and section 2.5.1 described menu navigation, the use of keyboard accelerators and permanent display of submenus.

Most of the programs expect parameters or input images and display resulting output images. Recall from section 2.3.5 that the *Mouse* state toggle button of the Control window indicates when xshow expects you to choose an input image. The button is described in detail in section 4.2.2.3 for easy reference.

If the program reads from `stdin` or writes to `stdout`, a text window for the job will be created. You may delete the window by hitting the `q` key or save it via the `Shift <Btn2>` combination.

If the program writes to `stderr`, the text will be written in the message area of the Control window.

[1] Refer to section 4.9 for the general command line syntax



(a) Main menu window. (b) Submenu window

Figure 4.1: The main menu window and one submenu window generated by menu-file example 4.1 on the next page.

To abort a program, press `<Btn1>` on the *Mouse* state toggle button.

4.1.1 Program menus

xshow will search for files with name `xshow_menues` in all the directories specified by the environment variable `XSHOWPATH`. This may be overridden by using the `-m` option, in which case only the file given by this option will be read.

The XITE distribution contains a menu-file which should be loaded automatically when the environment variables are set according to section 2.1. You only need to read the rest of this section if you intend to extend or modify the menu hierarchy. Otherwise, skip to section 4.2.

4.1.1.1 Example menu-file

Example 4.1 on the following page shows a simple menu-file, and the resulting main menu window and first submenu are displayed in figure 4.1. The example creates eight rows in the submenu *Arithmetic/logical...*:

1. *Generate noisy images...*. This refers to a new submenu.
2. "One image/band". This is a menu separator.
3. *Negate*.
4. *Scale*.
5. *Absolute value*.
6. "Two images/bands". A menu separator.
7. *Absolute difference*.
8. *Signed difference*.

Text to the left of "`;`" in the menu-file is the entry name, text to the right of "`;`" is the command line (ordinary UNIX commands).

Example 4.1 A possible menu-file for xshow.

```
1  !
2  ! Sample menu file.
3  ! Lines beginning with ;, ! or # are comment lines
4  !
5  ! @ in first column reads an include file
6  ! : in first column creates a new menu
7  ! + in first column means that the entry is a new menu
8  ! - in first column creates a menu separator
9  ! ? in first column defines a dialog
10 !   (blank) in first column means a simple entry
11 !
12 ! ONE MENU IN ONE OF THE MENU FILES MUST HAVE THE NAME
13 ! "Xshow" (root menu) !!
14 :Xshow
15 +Arithmetic/logical
16 +Others
17 !
18 !
19 :Arithmetic/logical
20 +Generate noisy images
21 -One image/band
22  Negate          ; negate <infile> <outfile>
23  Scale           ; scale <infile> <outfile> ?dialog:scale
24  Absolute value  ; absValue <infile> <outfile>
25 -Two images/bands
26  Absolute difference; absDiff <infile> <infile> <outfile>
27  Signed difference ; signDiff <infile> <infile> <outfile>
28 !
29 ?scale
30  scale the input image according to the formula \
31  \
32         output(x,y) = scale*input(x,y) + offset
33  # scale:  # -scale  # f # 1.0 \
34  # offset: # -offset # f # 0.0
```

4.1.1.2 Special menu-file command line arguments

A menu-file allows a few special command line arguments. These are

<infile>: Read input image indicated with the mouse.

<outfile>: Send output to xshow (image or color-table).

<xterm>: Send textual output to a separate text window. The command must be able to interpret "-1" for `stdout`.

?writeBIFFfile: Standard dialog (ask for BIFF output filename).

?readBIFFfile: Standard dialog (ask for BIFF input filename).

?dialog:dialogname: Fetch argument(s) from a dialog.

There are standard filename dialogs also for other file types than BIFF files.

A dialog is defined over three lines, and the dialogname is global across all the menu-files found in directories listed in the environment variable `XSHOWPATH`.

line 1: `?dialogname`

line 2: Short help text for the user.

line 3: Description of expected input arguments.

Refer to the standard menu-file `$XSHOWPATH/xshow_menues` and the Reference Manual for the FormDialog function for more information, especially on the third line of a menu dialog.

Lines 2 and 3 of a dialog definition may be continued if the last character on the line is a backslash.

4.1.1.3 Concatenating menu-files

If more than one menu file is found among the directories given by `XSHOWPATH`, all the entries are concatenated to form a larger menu hierarchy. Two conditions must be met for this to work.

1. One of the menu files must have a top level menu with the name "Xshow". This file is called a master menu-file.
2. The names of the top level menu of each of the other menu files must exist as entries in the top level menu of the master menu-file.

The menu-file supplied with the XITE distribution has a top level menu with the name "Xshow". Two of the other top level submenus are *Site-specific programs...* and *My programs...*. So, if you

1. have a file called `xshow_menues` in your home directory
2. you call the top level menu in your file "My programs"
3. set the environment variable `XSHOWPATH` to include your home directory (in addition to the directory where xshow's own menu file is located)

your menu hierarchy will pop up under the submenu *My programs...*.

In the same way, a menu-file specific to the local site or project can be created and included in the menu hierarchy.

4.2 Control window

xshow may display several images. The Control window gives information about existing images and running jobs. It is divided in four major parts: a column of menu buttons, a column of information buttons, a column of image data and a message area which displays information about jobs/programs and error messages. The Control window at startup is shown in figure 2.1 on page 7.

4.2.1 Menu column

This column contains three command buttons.

4.2.1.1 Macro

The *Macro* toggle button indicates whether or not the macro generating facility is active. It provides a mechanism for running a sequence of algorithms on several images, with a minimum of mouse button activity (and with the same arguments to the algorithms for each run).

Press the *Macro* button[2] and you will be asked to enter a name for a macro. All activity generated via the menu window will be logged to a file with the macro name. The macro can optionally be added to the menu hierarchy as the last entry in a menu-file. You will be asked for the location of such a file (section 4.1.1). The menu hierarchy of xshow is updated automatically to reflect the change. The macro file can alternatively be executed from the UNIX command line.

When running a macro from within the xshow menu hierarchy, you must give the input images in the same order as when the macro was generated (by clicking a mouse button as described in section 4.1).

A macro run may leave fewer images displayed than when the macro was generated. The rule here is that only those images which were not used as input to other commands (within the macro), are displayed. If you want to display also intermediate results, use the menu entry *Copy image* in the *File...* menu to display an extra copy while generating the macro.

A current limitation of the macro facility is that any dialog input during macro generation is fixed in the resulting macro (which of course can be manually edited).

4.2.1.2 Help

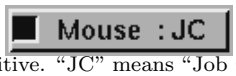
This button[3] will start an external program of choice. How to choose a program is explained in appendix A, in the paragraph concerning the environment variable `XITE_HELPER`. A typical choice is

- firefox. The WWW browser will be started on the front page of the on-line hypertext Reference Manual.

The environment variable `XITE_DOC` determines where the Reference Manual is found, i.e. in `$XITE_DOC/ReferenceManual/Contents.html`. The environment variable `XITE_MAN` determines where the formatted xshow manual page is found, i.e. in `$XITE_MAN/cat1/xshow.1`.

[2] or use the accelerator described in section 2.5.2
[3] or the accelerator described in section 2.5.2

(a) Insensitive. "N" means "None".



(b) Sensitive. "JC" means "Job Control".

Figure 4.2: The two possible states of the Mouse state toggle button in the Control window.

4.2.1.3 Quit

Quit xshow.

4.2.2 Info column

The fields in this column give some information about the state of xshow.

4.2.2.1 Images

The number of image windows displayed.

4.2.2.2 Jobs

The number of running jobs started by xshow. Press `<Btn1>` on the *Jobs* command button to get a list of all running jobs. The list is written in the message area.

4.2.2.3 Mouse

The two possible states of the *Mouse* state button are shown in figure 4.2. Figure 4.2(b) means that the *Mouse* state toggle button is sensitive and that xshow expects you to select an image as input to a program started from the menu hierarchy. The *Mouse* state toggle button enters this state when a program is started with the special argument `<infile>` (in the menu-file).

Select an image by clicking `<Btn2>` in an image window. Select an area (sub-band or region of interest, ROI) by dragging `<Btn1>`. A rectangular outline will show the selected area, and the position and size of the rectangle are continuously updated in the third row of the "Active Images column" of the Control window. Select all the bands of an image by clicking `<Btn3>` in an image window which displays one of the bands of the multi-band image.

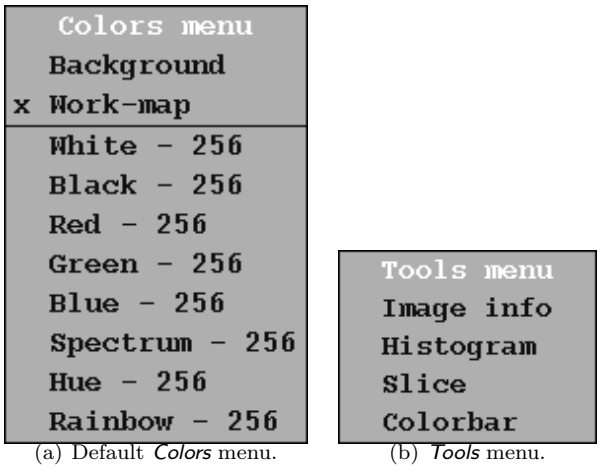
A job started from the menu hierarchy may be aborted by pressing `<Btn1>` on the *Mouse* state toggle button when it is sensitive.

Figure 4.2(a) means that the mouse has its ordinary function and that the *Mouse* state toggle button is insensitive.

4.2.3 Active Images column

When the cursor is on an image, xshow will update the three rows of this column.

First row: Image name, generated from the title of the BIFF image. The n'th band of an image with title equal to "title" will get the name "title:n". If additional images have the same title, they will be called "title #m" for some integer m, or "title:n #m".



(a) Default *Colors* menu. (b) *Tools* menu.

Figure 4.3: Default *Colors* menu and the *Tools* menu.

Second row: Magnification and zoom. This field displays the ratio between the image pixel size and the screen dot size. A zoom factor of 4 indicates that an area of 4x4 dots on the screen corresponds to one BIFF image pixel. Also, the percentage of image pixels visible in the image window is displayed.

Third row: Depending on the *Log position* option (in the *Options* image menu or on the command line), this field will either print the image size or the cursor position and pixel value.

When a sub-image or ROI is selected as input to a program, this row will display the position and size of the indicated ROI.

4.3 Colors menu

Each pixel on a computer screen is shown with a color. The color is determined by a look-up table (LUT), using the pixel-value as a table index. In principle this color-table (or color-map) has one entry for each possible pixel value. The indexing procedure is performed in hardware, using the hardware color-map.

In the menu-bar for each image[4] there is a *Colors* menu which can be used to choose different color-tables for the image. The default menu is shown in figure 4.3(a). A mark is present on the left side of the menu entry which is in use for this particular image window.

By default, the color-table entry labeled *Work-map* should be active in an image. Try to switch to the one labeled *White – 256*. There should be no changes to the image in this case. The two color-tables are identical gray-scale color-tables. Now try e.g. the *Spectrum* or *Hue* color-table.

All the images in an xshow session share the same list of color-tables in the *Colors* menu.

If the *Colors* menu contains only one entry labeled *Background*, you are most likely running xshow on a display with immutable color-maps, probably with the TrueColor visual class. You will not be able to switch or manipulate color-tables. Refer to section 4.6 and in particular to the options mentioned there for hints on using other visual classes. It may also be helpful/convenient to make the X server of your display use visual class PseudoColor or DirectColor as default. Talk to your system manager about this.

[4] or via the key/mouse combination `Shift <Btn1>` while the mouse pointer is inside an image window

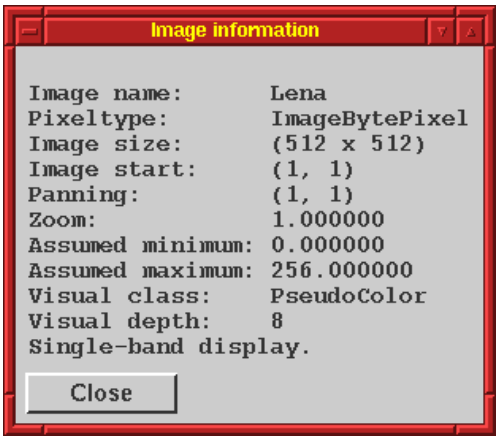


Figure 4.4: Image information window.

4.3.1 Color flashing

When the *Work-map* color-table is set for an image and the mouse pointer is inside the image window, you should see a black-and-white (actually shades of gray) version of the image. When the mouse pointer is moved outside the image, the colors of the image may change, depending on your display hardware. This does not indicate a malfunctioning hardware, nor a defect in xshow. In fact probably most workstations will show this behavior, while some don't.

The color flashing behavior described above will be explained in section 4.7. At this point, it is sufficient to understand that the majority of computer displays have only one hardware color-map, but on most systems a virtual color-map (software table) can be installed into the hardware color-map[5]. The color-flashing occurs because the xshow image and the other windows or display background don't want the same virtual color-map installed into the hardware color-map.

If you didn't experience any color flashing, your display probably has multiple hardware color-maps.

4.4 Tools menu

The *Tools* menu available from the image menu-bar[6] offers a small number of frequently used tools. It is shown in figure 4.3(b) on the preceding page. The menu hierarchy provided by the main xshow menu window offers lots of other image processing tools.

4.4.1 Image info

The *Image info* entry will print some information about the image, such as image title, pixel type, visual class[7], depth and image history. An example is shown in figure 4.4. Close the information window by hitting the `q` key or using the *Close* button.

[5] On displays with only an immutable color-map, the hardware color-map can not be changed or replaced.
[6] or with the key/mouse combination `Shift <Btn2>`
[7] Refer to section 4.6 for information about visual classes.

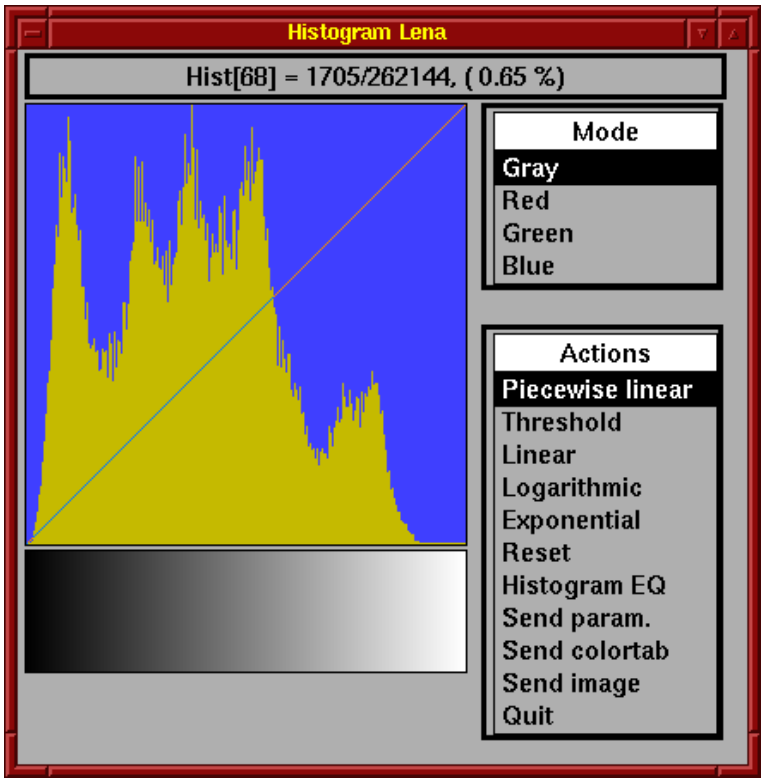


Figure 4.5: Histogram window.

4.4.2 Histogram

The *Histogram* entry was introduced in section 2.3.4. It will create a histogram window consisting of four parts, shown in figure 4.5[8]. The main part is a graphical display of the image histogram. The header at the top of the window displays the horizontal mouse pointer coordinate in the histogram graph (equal to the pixel value, 68 in the figure), and the histogram value (1705 in the figure), as well as the percentage (0.65 in the figure) of the total number of image pixels (262144 in the figure). Press `Ctrl <Btn1>` in the histogram part to toggle between histogram and cumulative histogram.

When the histogram tool is started, a new gray-scale color-table will be installed for the image, regardless of which color-table is active when the menu entry is chosen. Check the *Colors* menu to see that a new color-table is added as the last color-table entry, and that this new table is marked active.

Below the histogram graph is a rectangular patch which displays the gray-scale color-table. If this patch is missing, the *Colors* menu in the image menu-bar probably only contains one entry. In this case, refer to section 4.3 for advice.

The color-table (or look-up table, LUT) may be manipulated with the mouse pointer in the histogram part of the window. The result depends on the action chosen from the *Actions* list and the mode chosen from the *Mode* list.

Before proceeding, make sure that the *Reduced colors* option is turned on in the *Options* menu.

[8] If only the graph part of the histogram window appears, it means that the image is displayed with a visual class which only has an immutable color-map. The histogram tool is then severely limited.

If the image colors turn false when the mouse pointer is inside the *Mode* or *Actions* fields of the histogram window and you find this behavior disturbing, you may want to restart xshow with option `-share`. Refer to section 4.7.2.1 for more information.

4.4.2.1 Actions

The actions can be used to change the mapping between image pixel values and color[9]. This can be done with transformations on the new color-table or directly on the pixel-values. The advantage of working with the color-table is that it is much faster. The image is updated continuously, but the effect is the same in terms of color for both kinds of transformations.

Piecewise linear Use the mouse pointer to specify a piecewise linear polynomial as a transformation for the color-table.

Clicking `<Btn1>` will insert a new breakpoint and change the initial ramp function into a piecewise linear function. At the same time, the color-table patch below the graph will be updated accordingly, along with the image itself.

Clicking (and dragging) `<Btn2>` near a breakpoint will move the breakpoint vertically. Click `<Btn3>` near a breakpoint to delete it.

Click *Reset* in the *Actions* list to restore the linear ramp function (and the color-table patch and image).

With this action, highly specialized color-tables can be produced. When choosing the breakpoints of the transformation it may be helpful to zoom in the image to have the pixel-values displayed.

Threshold Use the mouse pointer to set a global threshold value for the image. Clicking or dragging `<Btn1>` horizontally will change the threshold value.

Linear Click or drag `<Btn1>` to specify a linear transformation, $p = ax + b$. The vertical position of the mouse pointer determines the inclination a of the linear function, while the horizontal position determines the line crossing b. With the mouse pointer in the bottom half of the histogram graph window, $a > 0$, and with the mouse pointer in the upper half, $a < 0$.

Logarithmic or Exponential The horizontal position of the mouse pointer when `<Btn1>` is pressed, determines the logarithmic or exponential function.

Reset Reset the initial linear mapping (for the selected mode).

Histogram EQ Make a new image with equalized histogram. This action actually transforms the image pixel-values. The histogram of the new image can also be inspected with a histogram tool.

Send param. If the *Linear* action is chosen, the linear transformation will be applied to all color-tables used by this image (until the *Reset* action is selected).

[9] None of the actions are available when the image is displayed with a visual class which only provides an immutable color-map.

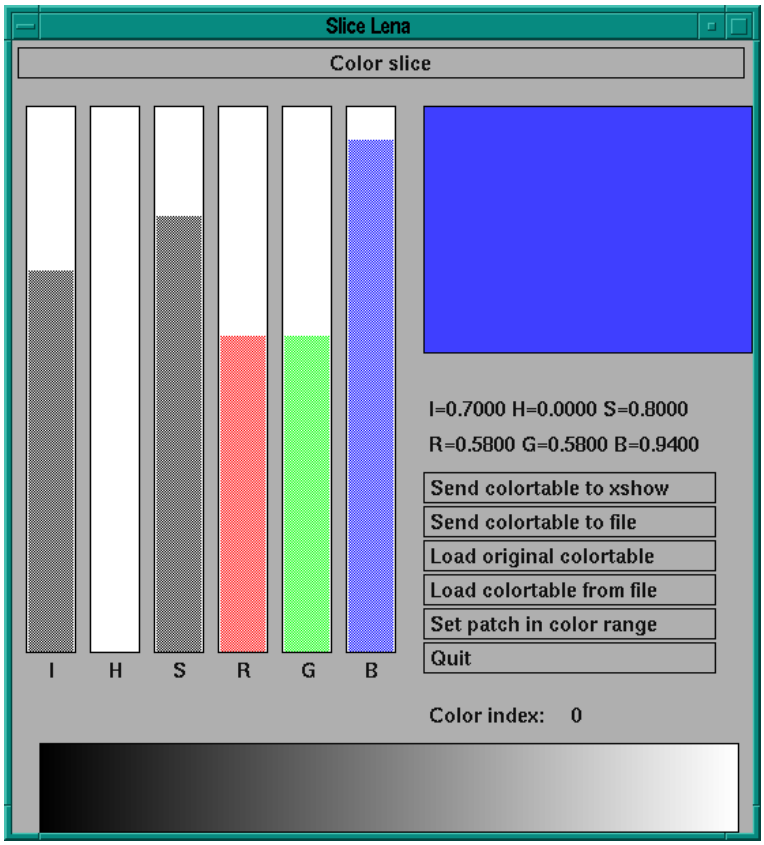


Figure 4.6: Slice window.

Send colortab Send color-table back to xshow (add it to the list of available color-tables) or to a file.

Send image Transform the image pixel-values according to the current color-table transformation (for the selected mode only), and send the transformed image back to xshow.

4.4.2.2 Mode

With *Gray* mode active, all the three primary colors undergo the same transformation, and they are all used if the image pixel-values are transformed.

With *Red*, *Green* or *Blue* mode only the corresponding primary color is affected (and used in pixel-value transformations). A separate curve is displayed in the histogram graph window for each primary color.

4.4.3 Slice

The *Slice* entry creates a new window which provides a way to mix colors in a palette by specifying RGB or IHS values[10]. An example is shown in figure 4.6.

[10]This tool is not available for images displayed with a visual class which only provides an immmutable color-map.

A piecewise constant color-table (look-up table, LUT) of pseudo-colors is added to the list of color-tables in the *Colors* menu. It is installed for the image and displayed in the bottom rectangular area of the slice window.

The other main parts of the slice window are a set of six vertical scrollbars, a single color palette or patch and a column of command buttons.

The scrollbars can be used to specify RGB or IHS components to get the desired color in the color palette. If your display has only one hardware color-map[11], the image, palette and color-table field colors will look right only when the mouse pointer is inside one of these three areas or inside a scrollbar.

The color mixed in the palette can be inserted into the color-table. Also, a color can be copied from the color-table to the palette. To achieve these changes, the following actions may be invoked in the color-table area at the bottom of the slice window.

<Btn1Down>: Fill the color-table at the cursor position with the palette color. This may influence the image.

<Btn2Down>: Set the palette color equal to the LUT value at the cursor position. This will influence the color palette as well as the RGB/IHS scrollbars and numerical labels.

<Btn3Down>: Replace a constant part of LUT values around the cursor position with the palette color. This may influence the image.

Drag <Btn1>: Same as **<Btn1Down>**.

The command buttons are probably self explanatory. The *Set patch in color range* button has the same effect as **<Btn1Down>**, except that the color-table indices can be specified numerically.

4.4.4 Color-bar

The *Colorbar* entry will display the currently active color-table for the image. The color-table window is a regular image window, but without a menu-bar. Only the *Options* menu is available with the ordinary key/mouse combination. Refer to section 4.5 for information about the *Options* menu.

The behavior of the color-bar image deserves some explanation. On a display with only one hardware color-map, the color-table will appear with correct colors only when the mouse pointer is inside the color-bar or an image which has the same active color-table. When the mouse pointer moves into the Control window or some other window which uses the default display color-map, the default display color-map will be shown in the color-bar.

This is not the full story of the color-bar. Refer to section 4.7.3 for the rest.

4.5 Options menu

Some of the entries in the *Options* menu, available from the menu-bar[12] and shown in figure 4.7 on the next page, have the same effect as some of the command line options of the ximage toolkit (refer to chapter 6). The menu is divided in three sections. The first section (entries *Reduced colors*, *Fixed aspect* and *Menubar*) concerns the appearance of the image window. The second

[11] see section 4.3
[12] or with the key/mouse combination **Shift <Btn3>**

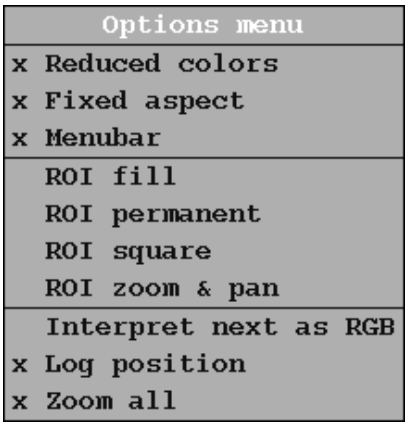


Figure 4.7: *Options* menu.

section concerns the Region of Interest (ROI). The third section (entries *Interpret next as RGB*, *Log position* and *Zoom all*) contains auxiliary options.

The Region of Interest is used when only a rectangular subpart of an image is selected as input to a program started via the menu hierarchy. This is done by clicking and dragging `<Btn1>`. The rectangular outline represents the ROI.

4.5.1 Reduced colors

This option is explained in section 4.7.2. It is insensitive for images displayed with a visual class which only provides an immutable color-map. The corresponding command line option of the ximage toolkit is `-full`.

4.5.2 Fixed aspect

Toggle whether the image should maintain a fixed aspect ratio. The corresponding command line option from the ximage toolkit is `-aspect`.

When resizing a window with fixed aspect ratio turned off, the window will get the size indicated by the window manager during the resize operation. The new shape is achieved by clipping.

If fixed aspect ratio is turned on, the window manager will not show the correct size during the resize operation. The final window size is in this case determined by the following rule: If only one of the window width and window height is changed, this will determine the new window size. If both window width and window height are changed, the new window size is determined by the new window width.

4.5.3 Menubar

Toggle the appearance of a menu-bar for this image. The corresponding command line option from the ximage toolkit is `-mb`. There are separate command line options available to choose which menus will be provided by the menu-bar.

4.5.4 ROI fill

If set, the ROI (Region Of Interest) will be inverted when you drag `<Btn1>`.

4.5.5 ROI permanent

If set, keep the last ROI visible.

4.5.6 ROI square

If set, force ROI to be a square.

4.5.7 ROI zoom & pan

If set, the ROI size and position will remain constant relative to the image when the image is zoomed and panned. Otherwise, the ROI will have a fixed screen size in a fixed screen position.

4.5.8 Interpret next as RGB

If set, the next image created will be interpreted as a three band RGB image. The corresponding command line option from the ximage toolkit is `-rgb`.

4.5.9 Log position

If set, the mouse pointer position in the image and the corresponding image pixel value will be displayed in the Control window. Otherwise, display image size. The corresponding command line option from the ximage toolkit is `-logpos`.

4.5.10 Zoom all

If set, force all images to get the same zoom factor during future zooming operations. This does not necessarily mean that all images change zoom factor every time one image is zoomed in or out. An enlarged image with all pixels visible is e.g. not able to get a zoom factor smaller than 1.0.

The corresponding command line option from the ximage toolkit is `-zoomall`.

4.6 Visuals menu

xshow was originally developed for color workstations with 8-bit PseudoColor displays, capable of showing 256 different colors at the same time. It should now work fine also on 24-bit DirectColor and TrueColor displays. xshow will give a warning message if it does not like the kind of visual you are using. A description of visuals can be found in the Xlib Programmer's Manual [8].

On displays which provide more than one of the above visual classes, you can choose the default visual class for the image windows (and certain parts of the histogram and slice windows) with option `-iv`. Otherwise, the PseudoColor visual class will be used if it is available.

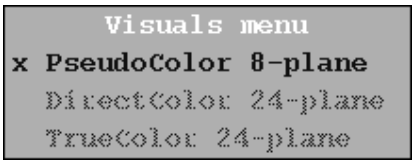


Figure 4.8: *Visuals* menu.

The *Visuals* menu lets you create a copy of an existing image to be displayed with a different visual class. For the Control window, dialog windows, menus, buttons etc. the default visual class of the display is always used. Whether the entries in this menu are sensitive or not, depends on the visual capabilities of the display, which visual/depth is the default for the display, and the ximage toolkit options `-multivisual` and `-iv`. An example of a *Visuals* menu is shown in figure 4.8.

4.6.1 PseudoColor 8-plane

Create a copy of this image for display with a PseudoColor visual of depth 8.

For PseudoColor visuals of depth 8, the color-maps have 256 entries, and each image pixel value is used to look up a color (the intensity for each of the three primary colors red, green and blue) in a color-map.

4.6.2 DirectColor 24-plane

Create a copy of this image for display with a DirectColor visual of depth 24.

For DirectColor visuals of depth 24, each color-map consists of three sub-maps, one for each of the primary colors red, green and blue. The color-map is also called a composite color-map. Each sub-map has 256 entries. Each image pixel value also consists of three parts, each part will index one of the sub-maps directly.

4.6.3 TrueColor 24-plane

Create a copy of this image for display with a TrueColor visual of depth 24.

TrueColor visuals of depth 24 also have composite color-maps, just like the DirectColor visuals. However, a TrueColor visual only has one color-map, and its entries can not be changed. This kind of color-map is called immutable. The colors of an image can not be changed by manipulating a color-map. The only way to change the colors is to change the image pixel values.

4.7 Colors

Being an image processing program, xshow is of course capable of displaying images with different colors. Because of the wide variety of display hardware available, the color models of the X Window System are fairly complex, and an application may not always behave the way a novice user would expect. You may have experienced the color flashing behaviour of section 4.3, with a preliminary explanation given in section 4.3.1. This section will provide more background.

Initially, the hardware color-map will usually contain a copy of the default display color-map. Most window based applications are designed to try to satisfy their color needs by using the

closest colors available in the default color-map, and not request special virtual color-maps to be installed in the hardware color-map. As long as all the applications on the screen use this friendly strategy, there will not be any color flashing when the mouse pointer is moved. All the windows are content with the colors provided by the default color-map which is installed in the hardware color-map. However, some applications have special color needs. This is the case for xshow when displaying images.

The color flashing behavior is sometimes called "going technicolor".

4.7.1 Technicolor in general

The Control window, menu windows and dialog windows displayed by xshow will use colors from the standard color-map, while images and a few other windows may require special color-tables and therefore refer to their own virtual color-maps. The window manager will install the virtual color-map of an image window into the hardware color-map when the mouse pointer (or keyboard focus) enters the image window and reinstall the previous color-map when the mouse pointer (or keyboard focus) leaves the image window[13]. This causes the color flashing, because all the windows use the single hardware color-map as a look-up table, while each image window actually requires the colors provided by its own virtual color-map.

If a system has multiple hardware color-maps, the color flashing induced by moving the mouse pointer between windows, will be reduced. With e.g. four hardware color-maps, four windows requiring different color-maps may appear just fine at the same time.

4.7.2 Reduced versus full color display

In the *Options* menu available from the menu-bar[14] above each image, there is an entry called *Reduced colors*. By default this option is on (indicated by a mark on the left). The menu entry will be insensitive (and in effect turned off) for visual classes with only an immutable color-map.

The decision of using *Reduced colors* or not affects the color flashing, or technicolor behavior, as well as the possibility of using image overlays. Technicolor was discussed in general in section 4.7.1 and will be treated in more detail in section 4.7.2.1. Overlays are briefly introduced in section 4.7.2.2 and discussed again in section 4.8.

With *Reduced colors* turned on, images are displayed with fewer colors than the display hardware offers. For 8-plane PseudoColor displays, the option toggles between using 128 and the maximum of 256 colors. For 24-plane DirectColor displays, it toggles between approximately 2 million and the maximum of approximately 16 million[15] colors.

4.7.2.1 Technicolor in detail

With *Reduced colors* turned on, all image pixel-values are transformed to the range 64–191[16], and the resulting value is used to look up the color from the color-table. This means that from a 256 element color-table, only the 128 entries with indices 64–191 are used.

Among the remaining 128 color-table entries, the 96 color entries for indices 0–63 and 224–255 are copied from the default color-table of the display. Window manager decoration, ordinary

[13] This will of course not happen on displays with only an immutable color-map.

[14] or with the key/mouse combination `Shift <Btn3>`

[15] Each of the primary colors red, green and blue can take a value in the range 0–255, for a total of $256^3 = 16777216$ colors.

[16] On 24-plane DirectColor displays, each primary color is transformed to this range.

application colors, display background, cursor colors etc. are often found among the first 64 color entries. In this way the rest of the display is more likely to keep its original colors, even when the mouse cursor is inside the image and only one hardware color-table is available.

If the colors of the xshow Control window change when the mouse pointer enters the image window, there may be a way to avoid this. Depending on your display, a warning message may have been issued in the terminal window when xshow was started. This warning instructs you to use the option `-share` when starting the program. With the option, the non-image windows will use the initial image window color-map. This is not possible on all display configurations, and using the option may also have its drawbacks, in particular that the Control window may not be able to appear with the default Control window colors. The `-share` option is described also in section 6.1.

Displays with multiple hardware color-maps tend to avoid the technicolor problem and thus don't benefit from the *Reduced colors* scheme to the same degree.

When turning *Reduced colors* off, the whole screen, except perhaps the current image, may change in terms of colors. The text in the image menu-bar may even become invisible. Moving the mouse pointer into the menu-bar will bring out the menu names again, and a combination of the "Shift" key with a mouse button will pop up the menus in the image window. If you see no change when the *Reduced colors* option is turned off, it most likely means that the display hardware has multiple color-maps.

4.7.2.2 Overlays

With *Reduced colors* turned on, a second image may be put on top of the first one, thus called an overlay. The overlay uses the 32 entries with indices 192–223 in the image color-table, and the pixel values of the overlay are transformed to this range before looking up the colors. The transformation from overlay pixel value p to color-table index i is by default $i = (p \bmod 32) + 192$.

Refer to section 4.8 for a discussion of the use of overlays and a description of how to choose colors for the overlay.

4.7.3 The color-bar revisited

If you turn off the *Reduced colors* option in the *Options* menu of the color-bar and again move the mouse pointer into a window which uses the default display color-map, the color-bar will look different from its appearance when the *Reduced colors* option was turned on. Now, with the option off, you can see the complete default display color-map in the color-bar. With the option on, only the color-table entries with indices in the range 64–191 were visible in the color-bar.

Moving the mouse pointer into the original image (which has *Work-map* as its active color-table, *Std-overlay* as its active overlay color-table and the *Reduced colors* option turned on), while the *Reduced colors* option of the color-bar image is still turned off, the color-bar image will display the reduced-colors version of the *Work-map* color-table in the middle (in the index range 64–191), the active overlay color-map in the index range 192–223 and the default display color-map in the index ranges 0–64 and 224–255[17].

The above behavior provides a visualization of the color-table partitioning described in sections 4.7.2.1, 4.7.2.2 and 4.8.1. On displays with multiple hardware color-maps, this is not available.

[17] If the visual class of the image is not the same as the default visual class of the display (the command-line option `-iv` was used, or the *Visuals* image menu has been used), the latter two ranges will display a gray-scale color-map instead of the default display color-map.

4.7.4 Adding extra color-tables

xshow will by default provide a few predefined color-tables in the image *Colors* menu, but new ones may be added[18]. This can be done from inside xshow or when the program starts.

When xshow starts, it will search for a file called `xshow_colortabs` in the directories listed in the environment variable `XSHOWPATH`[19]. The file should contain a list of color-table filenames. The color-table files are expected to be found in the same directory as the file `xshow_colortabs`.

The `-cl` option (from the ximage toolkit, see chapter 6) can be used to supply a file containing a list of color-tables. This will be used instead of `xshow_colortabs`. Also, option `-ct` can be used to supply a single additional color-table which will be used initially instead of *Work-map* for the images given on the command line.

Example 4.2 shows an `xshow_colortabs` file. The file also lists overlay color-tables, discussed in section 4.8.

Example 4.2 An `xshow_colortabs` file.

```
1   !
2   ! Sample color-tables file.
3   ! Lines beginning with ;, ! or # are comment lines
4   ! Lines beginning with :S start standard colortabs
5   ! Lines beginning with :O start overlay colortabs
6   ! Lines beginning with @ read an include file
7   !
8   ! First read all standard color-tables:
9   !
10  @$XSHOWPATH/xshow_colortabs
11  !
12  ! Add some private color-tables
13  !
14  :Standard color-tables
15  mywhite.col
16  myblack.col
17  !
18  ! Add an overlay table
19  !
20  :Overlay color-tables
21  myoverlay.col
```

From inside xshow, a color-table may be added by reading from file with the *Read colortable* entry of the *File...* and *Color...* menus. New color-tables also result from using the histogram or slice tools of the *Tools* image menu and from some of the entries in the main menu hierarchy, such as *RGB* → *pseudocolor* in the *Color...* menu and *From TIFF palette* in the *Format conversions...* menu. The new color-tables produced by entries in the menu hierarchy are automatically set active for the images produced by the same entries.

[18] For visual classes with only an immutable color-map, there will only be one sensitive entry in the menu, i.e. the one for the fixed display colormap.

[19] You can see the contents of the `XSHOWPATH` environment variable by giving the UNIX command `echo $XSHOWPATH`.

4.8 Image overlays

As mentioned in section 4.7.2.2, when the *Reduced colors* option is turned on, a part of the image color-table may be used to represent colors for an image overlay.

XITE comes with two kinds of image widgets, the Image class and the ImageOverlay class. The Image class is not equipped to handle overlays or ROIs, but the ImageOverlay class is an extension which can handle both. In xshow, all images are of class ImageOverlay, while some of the other display programs use images of class Image.

ROIs are described in section 4.5 and will not be discussed further at this point.

Overlays provide a way to label parts of an image without changing the pixel-values of the image itself. An overlay is itself an image, and any image with pixel-type unsigned byte can be used as an overlay. Try e.g. the image in the file `mask.img` as an overlay for `mona.img`, using the commands

```
§ cd $XITE_HOME/data/img
§ xshow mona.img +mask.img mask.img
```

In the above command line, the character "`+`" at the front of the filename `mask.img` instructs xshow to use this image as an overlay for the preceding image. The command will in this example display two images, the first containing `mona.img` with an overlay `mask.img` outlining the letters in the string "overlay", with separate colors for each letter. The second image contains `mask.img`, but appears all black[20]. Switch color-table for the second image to *Hue*. Now the letters "overlay" appear, but not with the same colors as in the overlay of the first image.

Zooming in on the second image until the pixelvalues are displayed as numbers, it can be seen that they are all in the range 0–7. This is why the image appeared all black with the default grayscale color-table in which pixelvalue zero indexes black.

Now concentrating on the first image, the one with "overlay" displayed as an overlay, turn the option *Reduced colors* off. This changes the colors of the "overlay" letters to shades of gray. Recall from sections 4.7.2.1 and 4.7.2.2 that with *Reduced colors* turned on, the image color-table is split in four parts. The central part contains a compressed version of the selected 256-element image color-table, and the image only uses this part to look up colors. A different part of the table is used to determine the colors of the overlay. With the option turned off, the color-table contains all the 256 elements of the selected image color-table. In this example, the image color-table is grayscale, and the "overlay" string appears with shades of gray.

Zooming in on the letter "y" in "overlay" until the numerical pixelvalues appear and toggling the option *Reduced colors*, shows that the pixel values remain constant, while the colors change.

Hitting the key `g` while the mouse pointer is inside the image will toggle the visibility of the overlay.

4.8.1 OverlayColors menu

From the *Colors* image menu, an image color-table can be chosen. From the *OverlayColors* menu available from the menubar[21] and shown in figure 4.9 on the facing page, a color-table for the overlay is selected. A change in the selection of an overlay color-table will only be visible when the *Reduced colors* option is turned on. By default the overlay color-table selected is the entry *Std-overlay*. Changing to *White – mono* will make the string "overlay" all white.

[20] at least when the mouse pointer is inside this image
[21] or with the key/mouse combination `Shift Alt <Btn1>`

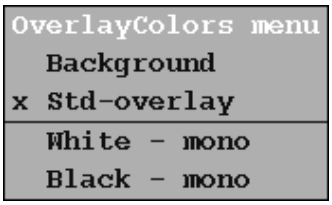


Figure 4.9: Default *OverlayColors* menu.

Please note that the overlay colors are always installed in the color entries with indices 192–223 in the current color-table for the image itself. Since all images use the same list of color-tables (in the *Colors* menu), no two images with overlays can use the same color-table and at the same time different overlay color-tables[22]. (Of course, one may work around this limitation by adding an extra copy of the image color-table and choose a different overlay color-table to be installed into this copy.)

4.8.2 Adding extra overlay color-tables

As explained in section 4.7.4, xshow will search for the file `xshow_colortabs` in the directories listed in the environment variable `XSHOWPATH`. In addition to a list of color-table filenames, this file may contain a list of filenames for overlay color-tables. Refer to example 4.2 on page 35.

The ximage toolkit option `-ovt` can be used to supply a single additional overlay color-table which will be used initially instead of *Std-overlay* for the overlays given on the command line.

4.8.3 More about overlays

xshow does not really take full advantage of overlays in its default configuration, although the ImageOverlay widget is equipped to handle free-hand drawing with the mouse. A different display program, xregion, described in section 5.7, is tailor-made to handle drawing and analysis of regions. In fact, the overlay image `mask.img` used above was created with xregion.

4.9 Command line

The general form of the xshow command line is

```
§ xshow [<option>...]  [<BIFF-file>...]  [+] [+<BIFF-file>...]
```

The interpretation of the above command line is that the characters "`<`", "`>`" enclose a required argument. The string inside the delimiters describes the kind of argument expected and should be replaced by an actual argument. A pair of square brackets denotes that its contents is an optional argument. Other characters, in this case "`+`", should be interpreted literally. Trailing dots, "`...`", denote additional arguments of the same kind. According to this, xshow may be started without any command line arguments or with any combination of zero or more options and zero or more filename arguements.

A "`BIFF-file`" can contain an image or a colortable. A "`+`" in front of an image filename indicates that the image is an overlay. A space delimited "`+`" indicates an empty overlay. The overlay will be

[22] This is not the whole story. Images displayed with different visuals can use the same color-table and at the same time different overlay color-tables.

written to the image listed in front of it on the command line. A "+" sign in front of a color-table filename indicates an overlay color-table.

All color-tables given as arguments will be available from the *Colors* menu of each image. This resembles the effect of the `-ct` option of the ximage toolkit. However, an image will be displayed with the rightmost non-option color-table argument preceding the image argument on the command line. With no such color-table argument, the initial color-table for all image arguments is determined by the `-ct` option. Without this option, the images are displayed initially with the standard *Work-map* gray-scale color-table.

Arguments which are overlay color-tables, are treated the same way as ordinary color-tables, except that they become available from the *OverlayColors* menu and that the corresponding ximage toolkit option is `-ovt`.

4.9.1 Examples

Display an image with the standard grayscale color-table:

```
§ xshow mona.img
```

Display an image with the standard grayscale color-table and an extra given color-table available from the Colors menu:

```
§ xshow mona.img mona.col
```

Display an image with a given color-table which is also available from the *Colors* menu:

```
§ xshow mona.col mona.img
```

Display two images, both with the same given color-table:

```
§ xshow mona.col mona.img lena.img
```

Display two images with different given color-tables:

```
§ xshow mona.col mona.img blue.col lena.img
```

Display two images, `mona.img` with the color-table `mona.col`, `lena.img` with the color-table `blue.col`, and make `green.col` available from the *Colors* menu:

```
§ xshow green.col mona.col mona.img blue.col lena.img
§ xshow -ct green.col mona.col mona.img blue.col lena.img
```

Display an image with a given overlay image:

```
§ xshow mona.img +lena.img
```

Recall from section 4.8 that hitting the key `g` while the mouse pointer is inside the image will toggle the visibility of the overlay.

Display the image `mona.img` with the color-table `mona.col` and an overlay `lena.img` with the overlay color-table `black.ovl.col`:

```
§ xshow -ovt black.ovl.col mona.col mona.img +lena.img
```

Use xshow at the end of a pipe:

```
§ median mona.img - 5 | xshow -
```

Refer to section 8.2 for information about pipes.

Display an rgb image in a single window (assuming that the display supports the DirectColor visual class):

```
§ xshow -rgb -iv DirectColor reine_rgb.img
```

4.10 Options

xshow accepts all of the standard X Toolkit command line options as well as the options defined by the ximage toolkit. Refer to chapter 6 for more information on the ximage toolkit.

Additionally, xshow supports the command line options listed below.

-i chan: Use input channel `chan`. This option may be used when xshow is forked out of other programs. Refer to the Reference Manual for the program fork_xshow and the function start_xshow.

-m filename: The file given by `filename` contains the menu layout. No other menu-files will be read. Default: `$Dir/xshow_menues` and `$Dir/.xshowrc`, where `$Dir` represents each directory listed in the environment variable `XSHOWPATH`. See also section 4.1.1.

4.11 More information

Please refer to the reference documentation for xshow and ximage. The Programmer's Manual [1] may also be of interest if you wish to write your own programs based on functions supplied with XITE.

Chapter 5

Other X display programs

The tailor-made display programs xadd, xfft, xfilter, xhistogram, xmovie, xpyramid and xregion are described in this chapter. They are not available under Windows, unless the X Window System is already installed. The more general main display program xshow was treated in chapter 4.

5.1 xadd

xadd calculates the pixel-wise weighted sum of two images. The weight is controlled by a scrollbar. One or both images may be negated, and the sum may be saved to a file. The images must be of equal size. The xadd control panel is shown in figure 5.1.

5.1.1 Arguments and options

xadd is started with a command of this form

```
§ xadd [<option>...]  <inimage-1> <inimage-2> [<outimage>]
```

The `<outimage>` argument can supply a filename which will be used when saving the image sum, or the filename can be given interactively.

xadd supports all standard X Toolkit command line arguments (see the manual page for X as well as the XITE ximage toolkit command line arguments, described in chapter 6.

Unless your display has multiple hardware color-maps, you will most likely want to use the ximage toolkit option `-share` so that the image sum appears with correct colors when you pull the slider in the scrollbar[1]. If you change the active color-table, you of course loose the benefit of this option.

[1]This is not possible if the image uses a different visual class than the default visual class of the display, because the slider always uses the default visual class of the display.

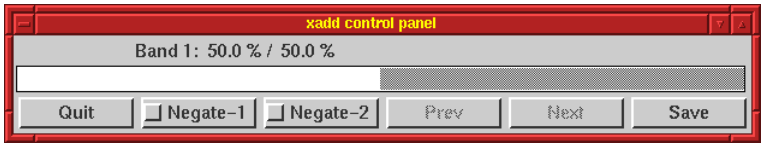


Figure 5.1: xadd control panel.

Figure 5.2: xfft control panel.

Refer to sections 4.7.2.1 and 6.1.

5.1.2 Examples

```
§ xadd img1 img2 img3
```

5.1.3 More information

Please refer to the Reference Manual for xadd, addw, ximage and Image.

5.2 xfft

xfft computes the 2D Fourier transform of a part of an image while displaying the result. Initially, the input image is displayed in one window, and the 2D Fourier transform of the upper, left corner in another window. The xfft control panel is shown in figure 5.2.

You may move the processing window (or ROI, region of interest) by pushing `<Btn1>` at the desired new position in the image, or change the size of the ROI by pushing the corresponding control panel buttons *Expand* and *Shrink*.

The *Save* button can be used to save the Fourier image in a file, while the *Info* button gives information about current sizes and positions.

The display in the Fourier window is the logarithm of the absolute value of the 2D Fourier transform, shifted to get the zero frequency in the center of the image window.

5.2.1 Arguments and options

xfft is started with a command of this form

```
§ xfft [<option>...]  <inimage>
```

xfft accepts all X toolkit command line options (see the manual page for X) as well as the XITE ximage toolkit options.

If your display only has a single hardware color-map, you may wish to use the ximage toolkit option `-share` for the image colors to appear correctly when you push the *Expand* and *Shrink* buttons[2]. Refer to section 6.1.

To change the geometry of the ROI with command line options, use the general X11 command line option for setting X resources, as shown in the following examples.

[2]Just as with xadd this may not work.

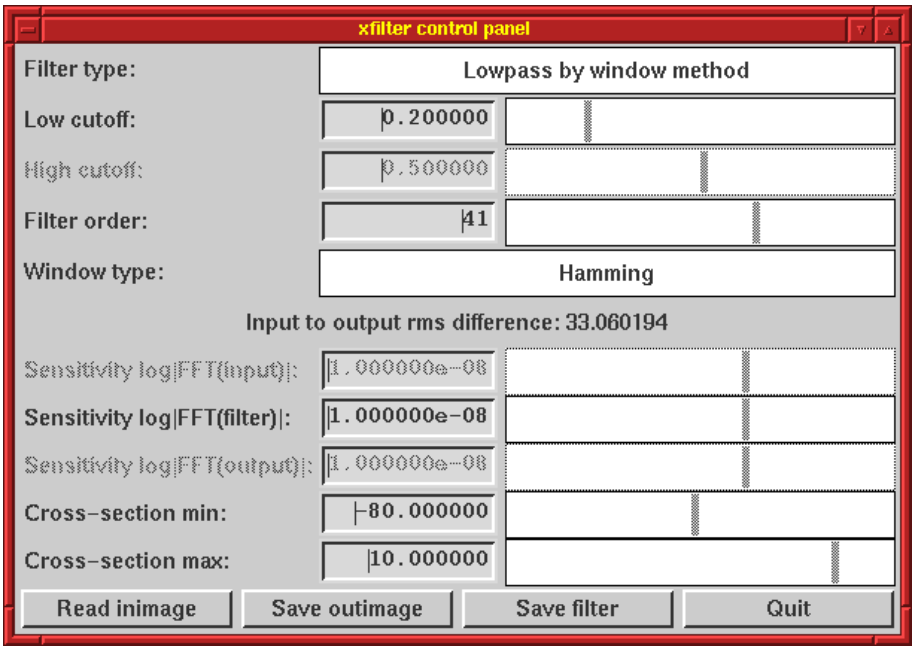


Figure 5.3: xfilter control panel.

`-xrm "XFft*ImageOverlay.roiX: 10"` Set horizontal coordinate for upper-left corner of region in spatial domain image to be Fourier-transformed.

`-xrm "XFft*ImageOverlay.roiY: 5"` Set vertical coordinate for upper-left corner of region in spatial domain image to be Fourier-transformed.

`-xrm "XFft*ImageOverlay.roiWidth: 64"` Set width of region in spatial domain image to be Fourier-transformed.

`-xrm "XFft*ImageOverlay.roiHeight: 64"` Set height of region in spatial domain image to be Fourier-transformed. This must equal the width of the region.

5.2.2 More information

Please refer to the Reference Manual for xfft, fft2d, logarithm and ximage.

5.3 xfilter

xfilter will filter a spatial domain image in the frequency domain. The filter type, filter order and window function can be interactively selected while the display of some or all of the involved images and power spectra are continuously updated.

The xfilter control panel is shown in figure 5.3. To change filter type, click `<Btn1>` on the current filter type entry and choose from the menu which appears. The window type can be changed in the same manner. The other parameters in the control panel can be changed either by operating the scrollbars or by editing the input fields.

By default the input image, the logarithm of the chosen filter's discrete Fourier transform magnitude, the cross-section of the latter and the output (filtered) image are displayed. Use the options

described below to change the selection of images to display.

The actual filtering is performed in the Fourier domain by multiplication of the discrete Fourier transform of the input image by the discrete Fourier transform of the filter. (Actually, the FFT is not directly computed, rather the Fast Hartley transform is used because it is more effective than the FFT. The results are the same as if the FFT had been used, except for roundoff errors.)

There are two main classes of filters available. One class contains realizable filters, the other contains non-realizable filters (realizable in this context means that the filters can be implemented in finite space).

The realizable filters are "lowpass", "highpass", "bandpass" and "bandstop". These are all based on the ideal Fourier transform magnitudes of the corresponding type. The spatial domain filter functions are known analytically, and their spatial extent is infinite. In this application the spatial domain filter functions are multiplied by 2-dimensional window functions to make them realizable. The resulting Fourier transform magnitudes are displayed.

The non-realizable filters are "lowpass_ideal", "butterworth_lp" and "exponential_lp". The former, "lowpass_ideal", is made from the ideal Fourier transform magnitude. Instead of using the analytically correct spatial domain filter function, "lowpass_ideal" computes the spatial domain filter function via inverse FFT. The result is not equal to the inverse Fourier transform on the continuous frequency domain because the frequency domain sampling results in aliasing in the spatial domain. "butterworth_lp" and "exponential_lp" have also been constructed from closed-form Fourier domain filter functions, and the spatial domain functions have been found with inverse FFT.

5.3.1 Arguments and options

xfilter is started with a command of this form

```
§ xfilter [<option>...]  <inimage>
```

xfilter accepts all X toolkit command line options (see the manual page for X) as well as the XITE ximage toolkit options.

Unless your display has multiple hardware color-maps, we recommend using the ximage toolkit option `-share` so that the images appear with correct colors when you operate the sliders in the control panel[3]. If you change the active color-table, you of course loose the benefit of this option. Refer to sections 4.7.2.1 and 6.1.

The following is a list of some of the additional options recognized by xfilter. Refer to the Reference Manual for a complete list.

`-ftype filter-type` Type of filter. Use one of the strings in table 5.1 on the next page. The default filter-type is "lowpass".

`-low cutoff` Lower cutoff frequency, normalized by the Nyquist frequency. This means that `cutoff` equal to 1.0 corresponds to half the sampling frequency. `cutoff` can be adjusted with a slider after startup. Default: 0.2.

`-high cutoff` Higher cutoff frequency (normalized by the Nyquist frequency). This can be adjusted with a slider after startup. Default: 0.5.

[3] Just as with xadd this may not work.

Full name	Short form
lowpass	lp
highpass	hp
bandpass	bp
bandstop	bs
lowpass_ideal	ilp
butterworth_lp	blp
exponential_lp	elp

Table 5.1: Filter type specification for xfilter.

Full name	Short form
rectangle	r
bartlett	b
triangle	t
hamming	hm
hanning	hn

Table 5.2: Window type specification for xfilter.

`-order filter-order` For the window-based filters, this is the filter size in pixels (diameter of window function). For the Butterworth and Exponential filters, `filter-order` is the parameter in the defining formula. It can be adjusted with a slider after startup. The default value is 41. For the "lowpass_ideal" filter, this option has no meaning.

`-wtype window-type` Type of window function to multiply with filter in the spatial domain. Use one of the strings in table 5.2. The default window function is "hamming". This option applies only to the "lp", "hp", "bp" and "bs" filter types.

`-rms` Do not calculate and display the RMS difference between input and output.

`-wall` Display input FFT, filter and output FFT in addition to input, output, filter FFT and its cross-section.

`-win` Do not display input.

`-wfin` Do not display input FFT.

`-wfilt` Do not display filter.

`-wffilt` Do not display filter FFT.

`-wout` Do not display output. Also implies option `-rms`.

`-wfout` Do not display output FFT.

`-wcross` Do not display cross-section of filter FFT.

5.3.2 More information

Please refer to the Reference Manual for xfilter, lowpass, highpass, bandpass, bandstop, lowpassIdeal, butterworth, exponential, fft2d, fht2d, ht2ft, fhtPower, logarithm, ximage and Image.

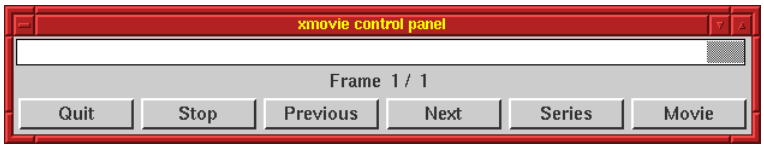


Figure 5.4: xmovie control panel.

5.4 xhistogram

xhistogram is a simple program which can be used to display the histogram of an image.

5.4.1 Arguments and options

xhistogram is started with a command of this form

```
§ xhistogram [-split] <inimage> [<outimage>]
```

The command line argument `<outimage>` can supply the name of an output file. This enables the *Histogram EQ* and *Send colortab* actions which will send the resulting image or color-table to the file.

xhistogram supports all standard X Toolkit command line arguments (see the manual page for X) as well as the XITE ximage toolkit command line options.

One additional option is recognized:

`-split` If the input image is multi-band, the default behavior is to show a total histogram for all the bands. This option forces one histogram per band.

5.4.2 More information

Please refer to the Reference Manual for xhistogram, the ximage toolkit and the Histogram widget.

5.5 xmovie

xmovie can display a sequence of images (animation). Three-band rgb images may be mixed with single- or multiband pseudocolor images. The scrollbar in the control panel determines the movie speed. The xmovie control panel is shown in figure 5.4.

The images can be zoomed and panned (with the usual mouse button and key combinations). The image window can also be resized with the window manager.

5.5.1 Arguments and options

xmovie is started with a command of this form

```
§ xmovie [<option>...]  <inimage>...
```

xmovie accepts all X toolkit command line options as well as the XITE ximage toolkit options.

If your display only has a single hardware color-map, you may wish to use the ximage toolkit option `-share` for the image colors to appear correctly when you operate the slider[4]. Refer to section 6.1.

One additional option is recognized:

`-f` If you don't intend to resize or zoom/pan the image window or use the *Histogram* or *Slice* entries in the image *Tools* menu, this allows for simpler calculations and will enable a faster movie, probably at least double speed.

Also, one of the options from the ximage toolkit deserves some comment:

`-rgb` Any three-band image command-line argument will be interpreted as an rgb image. In this way, xmovie can include pseudocolor as well as rgb images in the same movie. For an rgb image to appear as a color image, you must use a DirectColor or TrueColor visual. Refer to the ximage option `-iv`.

5.5.2 Examples

```
§ xmovie img1 img2 img3
§ xmovie -f nbandimg.img:1-5
§ xmovie -f -rgb -iv DirectColor reine_rgb.img mona.img lena.img
```

5.5.3 More information

Please refer to the Reference Manual for xmovie and ximage.

5.6 xpyramid

xpyramid will create and display a pyramid representation of a BIFF image. The resolution and number of gray levels can be selected interactively. The xpyramid control panel is shown in figure 5.5 on the facing page.

5.6.1 Arguments and options

xpyramid is started with a command of this form

```
§ xpyramid [<option>...]  <filename>
```

xpyramid supports all standard X Toolkit command line options (see the manual page for X) as well as the XITE ximage toolkit command line arguments.

5.6.2 More information

Please refer to the Reference Manual for xpyramid, pyramid and ximage.

[4] Just as with xadd this may not work.

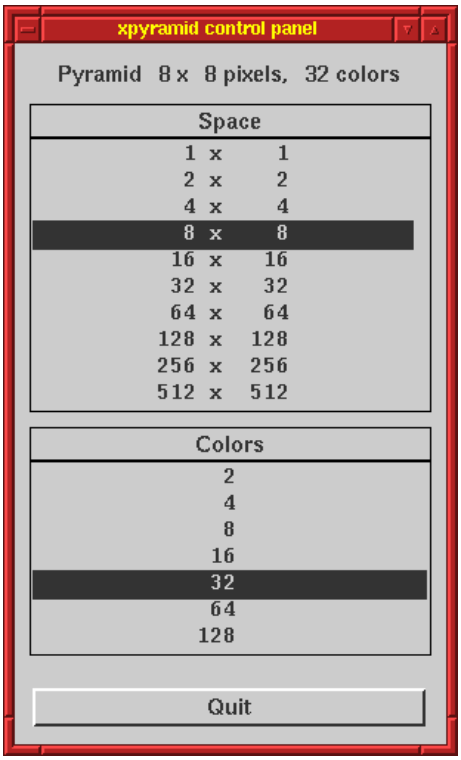


Figure 5.5: xpyramid control panel.

5.7 xregion

xregion is used to interactively draw regions in an XITE overlay image and have the regions analyzed statistically. The overlays are drawn with an overlay color palette. Each color may represent a class in a classified image. xregion has an interface to regionAnalyse and statistics which makes it very powerful.

The regions can be stored as masks in an output image file, with background equal to zero and region-pixels equal to a user specified value ($1 - 254$, default equal to one). Regions must be filled with the same class-number as they are drawn with.

An example of running xregion is given in figures 5.6 on page 49, 5.7 on page 49 (control panel) and 5.8 on page 50.

5.7.1 Arguments and options

xregion is started with a command of this form

```
§ xregion [<option>...]  <image-filename> [<maskimage>]
```

xregion accepts all of the standard X Toolkit command line options (see the manual page for X) as well as the XITE ximage toolkit command line options.

5.7.2 Editing

`<Btn1Down>` Draw line with the present class color.

`<Btn2Down>` Fill region with the present class color.

`<Btn3Down>` Clear region.

5.7.3 Pushbuttons for region drawing

Refer to figure 5.7 on the facing page.

Prev Work on the previous band of the image.

Next Work on the next band of the image.

Clear Remove all regions/masks (clear all overlays).

Read mask Read a file of regions/masks (an overlay image).

Save mask Save the regions/masks (an overlay image) in a file.

Save gray Save the image in a file, with pixel values set equal to zero where no regions/masks have been drawn.

Set class Set region number (class) for drawing (new draw color). The class number can also be incremented by hitting the `c` key inside the image window.

Close path Close curve.

Fill regions Fill all regions (all holes). This command will remove small areas of background pixels. This can also be done for single regions by pressing `<Btn2>` inside a region.

Checkpoint Save the current regions/masks (overlays). Go back to this state by pressing the *Undo* button. Checkpoint is performed automatically before each fill and delete operation.

Undo Go back to the last checkpoint state.

Graphics Toggle graphic overlay on/off. This can also be done by hitting the `g` key inside the image window.

Region analyse Start the program regionAnalyse on the current regions. Set options with the Parameters pushbuttons.

regionAnalyse will create a new window. To kill the window, hit the `q` key, or save the content of the window via the menu you get by pressing `Shift <Btn2>` in the text window.

Statistics Invoke the program statistics. Set options with the Parameters pushbuttons.

Quit Exit the program.

5.7.4 Pushbuttons for selecting options/parameters

Refer to figure 5.7 on the next page. These pushbuttons set options for regionAnalyse and/or statistics.

From the image in figure 5.6 on the facing page, xregion (with the help of regionAnalyse) will give the result shown in figure 5.8 on page 50. The 7 characters 'O', 'V', 'E', 'R', 'L', 'A' and 'Y' were drawn with class (or object) numbers $1, \ldots, 7$. Class number 0 represents the background. The result shows some features from the background, the 7 objects, and the three holes (in 'O', 'R', and 'A', region numbers 9, 10 and 11 in figure 5.8 on page 50).

Figure 5.6: Regions drawn in an XITE overlay image with xregion.

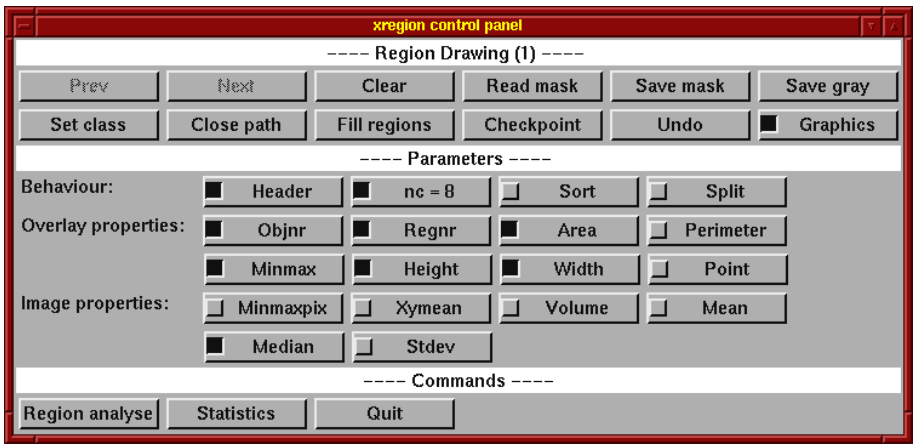


Figure 5.7: Layout of xregion control panel. Result of region analysis is shown in figure 5.8 on the next page.

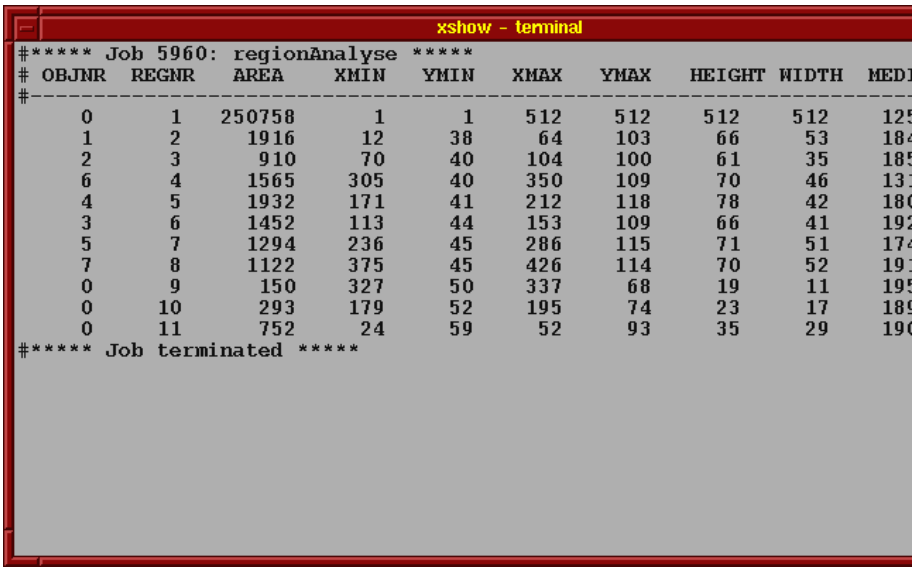


Figure 5.8: Result of running regionAnalyse from xregion.

5.7.5 More information

Please refer to the Reference Manual for xregion, regionAnalyse, statistics, ximage, Image and ImageOverlay.

Chapter 6

The ximage toolkit

All the XITE display programs are based on the ximage toolkit, itself an XITE component. The toolkit uses the Image and ImageOverlay widgets as well as some Athena and Free Widget Foundation (FWF) widgets. The latter are supplied with XITE. The ximage toolkit is not available under Windows, unless the X Window System is already installed.

In this document we focus on the usage of XITE programs, so only the command line options of ximage will be described.

6.1 Options

ximage based programs accept all standard X Toolkit command line options. The most useful additional options defined by the ximage toolkit are listed in this section. Consult the toolkit reference documentation for a complete list.

All the options have their X resource equivalents. This is the reason why some of the options have two legal forms (leading "+" and "-"), one to give a certain effect, the other to cancel the effect. This is necessary in order to be able to override X resource settings. Refer to the reference documentation for information about the X resources.

Unique abbreviations of the options are accepted.

`-aspect` Don't preserve image aspect ratio when resizing windows. Normally an image window will have the same width/height ratio as the image data. When a window is resized, it will be forced to keep this ratio unchanged. If `-aspect` is specified, it is possible to resize the window independently in the width and height directions. This can be toggled individually for each image in the *Options* menu.

`-cl filename`, `-colorlist filename` The file given by `filename` contains a list of color-table filenames which will be used instead of the default list of color-tables. Leading "~" and environment variables in `filename` will be expanded. If not found, try to find `filename` in one of the directories listed in the environment variable `XSHOWPATH`. The color-tables will be available from the *Colors* menu of the image windows. See also option `-colortable`.

The default value for `filename` is `xshow_colortabs`.

`-ct filename`, `-colortable filename` The file given by `filename` contains a color-table. Leading "~" and environment variables in `filename` will be expanded. If not found, try to

find `filename` in one of the directories listed in the environment variable `XSHOWPATH`. The color-table will be available from the *Colors* menu of the image windows.

Image command line arguments will initially be displayed with this color-table. Default is a monotonously rising gray-scale color-map, labeled *Work-map* in the *Colors* menu.

`-full` Do not use *Reduced colors* display of images to reduce "technicolor" problems. The default is that only 128 colors will be used to display images with PseudoColor visuals and only about 2 million colors will be used to display images with DirectColor visuals. This can be toggled individually for each image in the *Options* menu.

`-ih height`, `-imageheight height` Height of image widgets. The option `-aspect` can be useful with `-ih`, along with toggling the fixed aspect ratio entry of the image *Options* menu.

`-iv VisualClassName` Use `VisualClassName` for all image windows (and certain parts of the histogram and slice windows). Default: PseudoColor. If this is not available, try DirectColor. If this also fails, try TrueColor.

Legal choices: PseudoColor, DirectColor and TrueColor. The depth of the visual can not be chosen. For PseudoColor, a depth of 8 is used, for DirectColor and TrueColor, a depth of 24 is used. See also the *Visuals* menu of the image widget, and the options `-visualsmenu`, `-multivisual` and `-rgb`.

`-iw width`, `-imagewidth width` Width of image widgets. The option `-aspect` can be useful with `-iw`, along with toggling the fixed aspect ratio entry of the image *Options* menu.

`-logpos` Turn log position mode on. When the cursor is moved inside an image and the *Log position* option is on, the cursor position and pixel value are printed. Default: Off. This can be toggled globally for all the images in the *Options* image menu.

`-mb`, `-menubar` Start without menu-bar above image windows. This can be toggled individually for each image in the *Options* image menu.

`-multivisual`, `+multivisual` Enable display of images for all available visual classes, via the image *Visuals* menu. Implies option `-visualsmenu`.

`-ovt filename`, `-overlaytable filename` Similar to option `-colortable`, but `filename` now contains a color-table for an image overlay. It will be available from the image *OverlayColors* menu for widgets of class ImageOverlay.

Image overlay command line arguments will initially be displayed with this color-table. Default is a built-in 32 element table.

`-protect` Set `-protect` mode on. It will then not be possible to overdraw nonzero pixels in the overlay plane.

`-rgb` Assume that all three-band images represent 24-bit color images, where the first, second and third bands represent the red, green and blue intensities respectively. If a three-band image is displayed on a 24 bit-plane DirectColor or TrueColor screen and the `-rgb` option is specified, the three bands will be combined into a color image.

If the image is displayed on a different kind of screen, only the first band will be visible, although all the three bands are hooked to the data structure of the image window. In this way, the *Histogram* entry of the image *Tools* menu will calculate the combined histogram for all the three bands. All the three bands can be made visible e.g. with the *Copy bands of image* entry of the *File...* submenu in the main menu window[1].

This can be toggled globally for all the images in the *Options* image menu.

For this to work, you may need to use option `-iv`. Default: No.

[1] Remember to use `<Btn3>` when selecting the input image so that all the bands are passed along to the copy program.

`-share` Let non-image windows use the same initial color-map as image windows. This can reduce color-map flashing on displays with a single hardware color-map. It is especially useful in this case for applications which involve manipulation of a non-image widget with real-time image changes, such as using the slider in the XITE applications xadd and xfilter.

The color-map for non-image windows will not change when the image color-map is changed via the *Colors* image menu.

Every ximage based application will issue a recommendation to use this option if it determines that the option may reduce color-map flashing.

The drawback of using this option is that the colors of the non-image windows may not be set equal to the colors requested in the X resource file (or with X toolkit options `-bg` and `-fg`). Warnings will be issued to this effect, from the X toolkit.

This option only works when the preferred image visual class equals the default display visual class.

`-version` Print XITE version number and exit. Default: Don't.

`-zoomall` Turn *Zoom all* mode on. With this mode set, all images will be zoomed/panned with the same parameters. This can be toggled globally for all the images in the *Options* image menu.

6.2 More information

Please refer to the reference documentation for the ximage toolkit.

Chapter 7

The BIFF image concept

The purpose of this chapter is to describe what we mean by a "digital image" in the XITE system. This is done by describing the different elements of a digital image, and the possibilities and limitations of this concept.

A more in-depth description of the BIFF format and routine library can be found in the [5, BIFF Manual].

Conceptually the digital image contains three main parts, the info-data, the band-data and the block-data. The info-data contains various information about the image, textual as well as numerical. The band-data contains numerical information representing the intensity distributions in the bands. The image may contain any number of bands, numbered from one. The bands have individual sizes and pixel types. Sizes are chosen freely and a set of pixel types are defined. Position and magnification of each band is described relative to a global coordinate system. Finally the block-data contains whatever the user wants, no restrictions are placed on it. (The block-data is rarely used, and is kept mainly for historical reasons).

7.1 Definitions

- A PIXEL is a basic unit, a variable able to contain information about the intensity of a local area of a picture. A pixel has a certain type and thus a certain domain, and a value within this domain.
- A BAND is a rectangular arrangement of pixels, all of them having the same type.
- THE LOCAL COORDINATE SYSTEM. To be able to name each individual pixel within a band, a coordinate system local to the band is used. The local coordinate system has two axes, X and Y, the X index grows from left to right starting with one, ending with xsize, and the Y index grows from top to bottom starting with one and ending with ysize. Thus, the upper left pixel of any band has local coordinates (X,Y) = (1,1), and the bottom right pixel has coordinates (X,Y) = (xsize,ysize). A band in its local coordinate system is illustrated in figure 7.1 on the next page
- A LINE is a sequence of pixels from one band, containing all pixels with some fixed Y coordinate, ordered with growing X coordinate.
- A COLUMN is a sequence of pixels from one band, containing all pixels with some fixed X coordinate, ordered with growing Y coordinate.

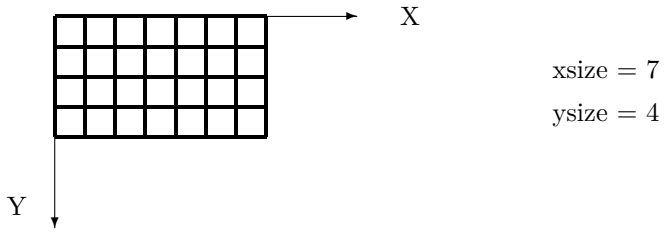


Figure 7.1: A band in its local coordinate system.

- THE GLOBAL COORDINATE SYSTEM. We often need more than one band in an image. Two bands in an image can represent the same spatial area (on the surface of the earth, or inside an engineering drawing), or different spatial areas. They can have identical, or different spatial resolution. To store such information, we use a global coordinate system. For each band, two parameters are used to describe the spatial resolution. Xmag is a magnification factor in the X direction, and ymag is a magnification factor in the Y direction. If each pixel value of a band is repeated xmag times in the X direction and ymag times in the Y direction, then each of these new pixels will cover the same "true" area as a pixel from another band in the same image, treated the same way.

  To describe the relative positions of these "blown-up" bands, we use two more parameters, xstart and ystart. They specify the coordinates in a global coordinate system of the pixel that is numbered (1,1) in the local system. The axes of the global coordinate system must be parallel to the axes of the local systems. Figure 7.2 gives an example, the band from figure 7.1 is shown in a global coordinate system. All six parameters, xsize, ysize, xstart, ystart, xmag and ymag, are integers, and except for xstart and ystart they should be positive. When there could be confusion between the pixels in the local coordinate system, and the elements in the global coordinate system, the names locel and globel could be used.

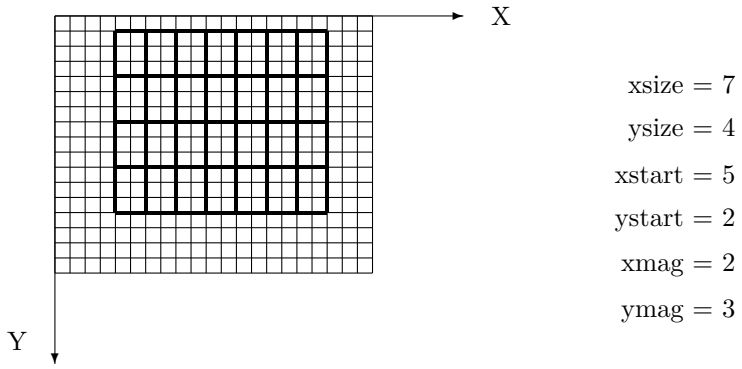


Figure 7.2: A band located in the global coordinate system.

7.2 Info-data

The info-data contains all information necessary to describe the bands, and other information connected to the image.

- TITLE A short text describing the image.
- NBANDS The number of bands in the image.

- PARAM Eight user defined parameters that may contain any sort of information, each with the size of an integer. We recommend that the first parameter is used as an identification of the type of image. The use of the last seven parameters could then be standardized among the users of this particular image type.
- BANDARRAY Numerical information about each band, containing:
  - PIXTYP The pixel type of the band. The pixel type must be one of the following 9 types: unsigned byte, signed byte, unsigned short, signed short, integer, real (float), complex, double precision real (double) and double precision complex.
  - XSIZE Horizontal band size.
  - YSIZE Vertical band size.
  - XSTART Horizontal positioning.
  - YSTART Vertical positioning.
  - XMAG Horizontal magnification factor.
  - YMAG Vertical magnification factor.
- NCHARS Number of characters in text-data.
- TEXT-DATA Textual description of the image. The text-data is a sequence of characters of any length, and can even grow as new comments are added.
- NBLOCKS Number of blocks in block-data.

The bands of an image may have different pixel types, as well as different sizes, magnifications and locations.

7.3 Band-data

The band-data contains the pixel values in every pixel of every band. There are NBANDS bands, NBANDS >= 1. Thus, the band-data only contains a sequence of bands, possibly with different pixel types and sizes, in their individual local coordinate systems.

7.4 Block-data

Block-data is a collection of blocks, each containing 512 bytes. The contents of each block is entirely decided by the user, and the user must keep in mind what is stored where, and how to interpret the data. The blocks can be used to store histograms, color tables etc. It mainly has interest for non textual data, as info-data can contain any amount of text. There are NBLOCKS blocks, NBLOCKS >= 0. The number of blocks can increase, new blocks can be added to the image.

7.5 Examples

Look in the `$XITE_HOME/data/img` directory for an image file, and try

```
§ biffinfo <filename>
```

to see the image parameters and the image history. The image history of these images may be empty. Run some XITE program like the median filter,

```
§ median <input image> <output image>
```

and compare the text fields reported by biffinfo of the two images. The output image should have an additional entry describing the median filtering. Then test the newtitle program used to change the title of an image,

```
§ newtitle <image> <new title>.
```

Finally, use the bifftext program to modify the text field containing the image history,

```
§ bifftext <image> <new text>
```

appends a text line,

```
§ bifftext <image>
```

lets you edit the whole text. In the latter case, your `EDITOR` environment variable is examined to start your favorite editor. If the environment variable is unset, emacs will be used on a UNIX system, and Notepad will be used on a Windows system.

Chapter 8

Command based image processing

This chapter describes how to activate XITE programs from the shell/command window, one program at a time, as well as several in a pipe, and through shell scripts. We will not go into detail about the different programs, for this we refer to the reference documentation.

8.1 Activating a program

If your PATH environment variable is modified to account for the XITE system (see section 2.1), all you have to do to activate a program in the XITE program library, is typing the name of the program. Most programs will respond with their usage and then terminate, if activated with no arguments. (There are a few exceptions, programs that are able to behave sensibly with no arguments.) In section 3.3 a number of help options were described. In section 2.2 we explained how to specify image file names, including a syntax to specify that only a subset of the bands are to be processed.

The file format allows every band to have its own size and pixel type, and allows you to choose between a number of different pixel types. Unfortunately, many programs are not able to handle all pixel types, the most generally accepted type is unsigned byte. This pixel type is well suited for handling gray level and pseudocolor images and allows 256 different colors. High quality color images can be represented as three band images, each band of type unsigned byte. If you specify as input argument an image with a pixel type which the actual program is not able to handle, you will get an error message stating this. This is also the case if the size of the image is not correct. A few programs may, due to the implemented algorithm, require that the size is some power of two, that the image is quadratic, or that the size is equal to that of another image.

8.2 Piping

Some program collections follow the design strategy that all programs always read from standard input and write to standard output, always using the piping mechanism. XITE uses a different strategy. File names must be given as arguments to the program. The reason for this is that we want to make the system as simple to use as possible for the unexperienced user. Starting an XITE program with no arguments will give you the usage message. Giving a similar command to another system will make the system wait forever — wait for you to supply the whole image byte by byte.

However, with the XITE strategy, how do you specify piping if that is really what you want? The solution is the special file name "-". If given as input file name, "-" is interpreted as standard input, if given as output file name, "-" is interpreted as standard output. A few examples follow. File names should be substituted for <input> and <output>.

8.2.1 Examples

The character "§" in the examples represents the prompt from the shell/command window (e.g. csh, sh or the Windows command window) you are using. The string "<input>" should be replaced by the name of the file which contains the input image, and the string "<output>" should be replaced by the name of a file for storage of the result.

Example 8.1 is a command line for displaying the output of a median filter. It demonstrates the use of the argument "-" to denote the standard output (of the median program) as well as the standard input (for xshow). Also, the piping symbol "|" is used to connect the standard output of median to the standard input of xshow. This example does not work under Windows because xshow is only available on UNIX systems, unless the X Window System is also installed on the Windows system.

Example 8.1 Command line to display the output from a median filter of size 5.

```
§ median <input> - 5 | xshow -
```

Example 8.2 adds a thresholding operation to the median output. It uses the argument "-" for both the standard input and standard output of the threshold program.

Example 8.2 Command line to display the result of thresholded median output. Threshold equals 100.

```
§ median <input> - 5 | threshold - - 100 | xshow -
```

Example 8.3 connects an edge detector between the median and threshold. The result is not sent to the display but saved in a file.

Example 8.3 Save the result of a three-stage image processing pipe.

```
§ median <input> - 5 | sobel - - | threshold - <output> 100
```

Example 8.4 on the next page implements a different edge detector. Unlike the previous examples, this example needs be written differently The median filter preserves edges quite well, the mean filter doesn't. The difference will give an edge detector. The parentheses group the two commands median and mean. They output one image each, and they are both passed along to absDiff.

You may also use the syntax "-<n>" to indicate a file name, where <n> is any non-negative number. This special file name is interpreted as channel number <n>. Thus, the "-" may be replaced by "-0" at all input positions and by "-1" at all output positions. For daily use, this has little purpose, but it may be useful when forking out XITE programs from other processes (refer to the Reference Manual for fork_xshow and start_xshow, available on UNIX systems).

Example 8.4 Command line for edge detector which displays the result.

```
# This is a comment for the UNIX shell
# The following pipeline uses UNIX shell syntax
§ ( median <input> - 5 ; mean <input> - 5 ) | \
absDiff - - - | \
threshold - - 10 | \
xshow -
```

```
REM This is a comment for the Windows XP command processor
REM The following pipeline uses Windows XP command processor syntax
§ ( median <input> - 5 & mean <input> - 5 ) | absDiff - - - | threshold - <output> 10
```

8.3 Shell programming

If you are regularly using long command sequences, typing them gets boring. You should then create a script containing the command sequence, or you may find the automatic macro generation described in section 8.4 useful. In these examples we assume that you have some general knowledge about UNIX shell scripts or the Windows command processor syntax, otherwise, see the manual page for the UNIX shell you are using (sh, csh, tcsh, etc.), or consult Windows Help. The UNIX versions of the examples are written for sh, as evident from their first line.

Example 8.4 in the previous section may be a typical example. Let's create a UNIX shell script in a file called `edge`, and a Windows batch program in a file called `edge.cmd`, both shown in example 8.5. We have removed the display program, compared to the last example in section 8.2.1. This makes the script applicable for display as well as for file generation.

Example 8.5 Script for edge detector which does not display the result.

```
#!/bin/sh

( median $1 - 5 ; mean $1 - 5 ) | \
absDiff - - - | \
threshold - $2 10
```

```
@echo off
REM Windows XP command processor syntax

( median %1 - 5 & mean %1 - 5 ) | absDiff - - - | threshold - %2 10
```

For display:

```
§ edge <input> - | xshow -
```

For file generation:

```
§ edge <input> <output>
```

Note that in example 8.5, the input file name "`$1`" ("`%`" in the Windows version) appears twice in the script. This means that a real file name has to be given, you may not pipe an image into the edge script. If you want to make the script more general, the window sizes of median and mean, as well as the threshold value, may be arguments to the script. This is shown in example 8.6 on the next page. The example also shows how to test the number of arguments. The "`1>&2`" appended to the echo command in the UNIX shell version, makes sure that the usage message appears on the screen even if the output from the script is piped into some other program.

Example 8.6 Script which checks the number of input arguments.

```
#!/bin/sh

if test $# != 5; then
  echo "Usage: edge <input> <output> <medianfiltersize> \n\
       <meanfiltersize> <threshold>" 1>&2
  exit 2
fi

( median $1 - $3 ; mean $1 - $4 ) | \
absDiff - - - | \
threshold - $2 $5
```

```
@echo off
REM Windows XP version

REM At least 5 parameters?
if     "%5" == "" goto usage

REM No more than 5 parameters?
if not "%6" == "" goto usage

( median %1 - %3 & mean %1 - %4 ) | absDiff - - - | threshold - %2 %5
goto end

:usage
echo "Usage: edge <input> <output> <medianfiltersize>"
echo "            <meanfiltersize> <threshold>"

:end
```

You can even give the edge script the ability to process the standard XITE help options with the version in example 8.7. Refer to the Reference Manual for xiteStdOpt for information on this XITE-supplied auxiliary script. This is not available in the Windows version.

Example 8.7 Script which can process standard XITE help options.

```
1  #!/bin/sh
2
3  progName=`basename $0`
4  usage="Usage: $progName <input> <output> <medianfiltersize> \n\
5         <meanfiltersize> <threshold>"
6  eval `xiteStdOpt 5 5 $0 $@ "$usage"`
7
8  ( median $1 - $3 ; mean $1 - $4 ) | \
9  absDiff - - - | \
10 threshold - $2 $5
```

Example 8.8 shows that file conversion from pbm format to BIFF format can be accomplished with a simple UNIX shell script, using a converter from raw binary format to BIFF. (A separate C program in the XITE distribution will do this task more efficiently. It is also available as an entry in the xshow menu, under *Format conversions...*.)

Example 8.8 Conversion from pbm to BIFF file format.

```
1  #!/bin/sh
2
3  # grab the pbm header
4  head -3 $1 | tr " " "\012" >  /tmp/pbm2biff_i_$$
5
6  # find the horizontal size of the image
7  xsize=`cat /tmp/pbm2biff_i_$$ | head -2 | tail -1`
8
9  # find the vertical size of the image
10 ysize=`cat /tmp/pbm2biff_i_$$ | head -3 | tail -1`
11
12 # find the size of the pbm header
13 head=`cat /tmp/pbm2biff_i_$$ | wc -c`
14
15 # find the size of the whole pbm file
16 tsize=`cat $1 | wc -c`
17
18 # find the number of bands in the pbm file
19 bands=`expr "(" $tsize - $head ")" "/" "(" $xsize "*" $ysize ")"`
20
21 # convert to BIFF
22 raw2biff -ih $head -org lsb $1 $2 $xsize $ysize $bands
23
24 # clean up
25 /bin/rm /tmp/pbm2biff_i_$$
26
27 exit 0
```

In a similar way, a BIFF file can be converted to other file formats, using biffinfo with its different options to obtain information about the image parameters.

To list all BIFF files in a directory, give the command `lsbiff <files>`, where `<files>` is as for

ls, and lsbiff for the UNIX shell is shown in example 8.9.

Example 8.9 Script to list all BIFF files in a directory.

```
1  #!/bin/sh
2
3  for i in `biffinfo -f $1`; do
4    echo $i
5  done
6
7  exit 0
```

The program statistics is also well suited for shell scripts, giving unformatted or formatted output of the desired statistical properties of one or more images. The UNIX shell script in example 8.10 will print the names of all BIFF images, together with a header and lower and upper values for each band.

Example 8.10 Script to process a number of images.

```
1  #!/bin/sh
2
3  for i in `biffinfo -f $1`; do
4    statistics -fhlu $i
5  done
6
7  exit 0
```

8.4 Automatic shell/macro generation

The xshow display program has the ability to generate simple shell scripts (or macros) automatically. There can be no looping in the scripts. Use the button labeled *Macro* in the xshow Control window to generate macros. Refer to section 4.2.1.1 for more information.

Appendix A

The environment variables in detail

The operation of XITE depends on a few environment variables which are described in detail below. Refer to section 2.1.1 for instructions on how to set the variables.

On a UNIX system, the value of an environment variable is accessed when the variable is prefixed with "$", e.g. `$XITE_HOME`. Under Windows, the value is accessed when the variable is surrounded by a pair of "%", e.g. `%XITE_HOME%`. Most of this document uses the UNIX notation, but you should be able to figure out the Windows notation in each case.

XITE_HOME The home directory of the XITE system on your local computer.

XSHOWPATH A list of directories where the xshow program should look for specific data files. This will typically be equal to

```
$XITE_HOME/data/xshow.
```

PATH This is a standard UNIX and Windows environment variable, containing a list of directories to be searched for a matching file when the user gives a command. Your standard `PATH` should be extended with a directory for XITE executables. Under UNIX, the list elements are separated by ":", under Windows they are separated by ";".

The executables will most likely be found in

- `%XITE_HOME%\bin` for Windows.
- `$XITE_HOME/bin` for unix/linux

The directory `$XITE_HOME/bin` should always be in your path, shell scripts may also be found there.

MANPATH `MANPATH` is used by the UNIX man command on some systems, while man on other UNIX systems may construct a `MANPATH` automatically from `PATH`.

If the man command on your system needs a `MANPATH` variable, it must be a list of all directories in which the man system should start searching for man files. If you have already defined a `MANPATH`, it can be extended with `$XITE_HOME/man`. If you have no such environment variable, it may be defined as

```
$XITE_HOME/man:/usr/local/man:/usr/man
```

or something similar.

Consult your local system manager if you don't know where to search for man files.

XITE_MAN This determines the location of the manual pages for XITE on both UNIX and Windows systems. It is used when an XITE program is given one of the options `-man` or `-whatis`. It is also used by xshow when the *Help* button in the Control window is pushed and the helper application is more, less or cat.

XITE_HELPER A colon-separated list of programs, one of which will be started to provide help when the *Help* button in the Control window of the xshow display program is pushed. xshow will start the first program in the list. If it fails, it will start the next, and so on. By default this variable is set to

```
firefox:opera:man:more:less:cat
```

This means that xshow will first try to start firefox, a World Wide Web browser which will give you access to the on-line hypertext Reference Manual. If they all fail, then the manual page of xshow will be displayed by the man program.

more, less and cat will by default display the formatted manual page for xshow. The location of this manual page is given by the environment variable `XITE_MAN`. A different file can be chosen with the environment variable `XITE_HELPER_OPTION`.

If `XITE_HELPER` is not set, the X application resource `xiteHelper` is checked in the same manner.

XITE_HELPER_OPTION This can be set to a list containing command line options for the corresponding program given by the variable `XITE_HELPER`.

If `XITE_HELPER_OPTION` is not set, the X application resource `xiteHelperOption` is checked in the same manner.

XITE_DOC This is the name of the directory where the XITE Reference Manual is located. It will typically be set to `$XITE_HOME/doc`, and the front page of the Reference Manual is

`$XITE_DOC/ReferenceManual/Contents.html`.

The variable `XITE_DOC` is needed by the WWW browsers mentioned above, when they are launched by xshow.

Appendix B

General file name syntax

Examples of specifying a subset of image bands were given in section 2.2. Using the meta symbols "`()`", "`<>`", "`OR`" and "`*`", the general file name syntax for specifying a subset is

```
<specified file name>       ::= <real file name><band specification>
<band specification>        ::= (:<simple band specification>)*
<simple band specification> ::= (<band number> OR <band range>)
<band range>                ::= <band number>-<band number>
<band number>               ::= 1 OR 2 OR 3 OR ....
```

The band specification separator is ":" on UNIX systems, ";" on Windows systems.

Appendix C

List of available programs

Below is a list of programs available in XITE. On a UNIX system, most of them can also be started from the menu hierarchy in xshow. The display programs are not available for Windows, unless the X Window System is also installed.

absDiff	absolute difference between two images
absValue	Take the absolute value of an image
addGauss	add gaussian random noise to an image
addPoisson	add Poisson noise to the output image
addw	add two bands pixel by pixel, with weights
affine	affine geometric transform
ascii2biff	convert an ascii file to BIFF format
average	computes the average of several bands in an input image
bandpass	make an "ideal", truncated, lowpass filter in the spatial or Fourier domain
bandstop	make an "ideal", truncated, lowpass filter in the spatial or Fourier domain
bdf2biff	converts a Bitmap Distribution Format font file to a BIFF file
biff2ascii	dump part of an image in ascii format
biff2png	Convert an image from BIFF to PNG format
biff2pnm	Convert a BIFF image to pnm (ppm, pgm or pbm) format
biff2ps	convert BIFF image to PostScript
biff2raw	write part of an image as raw, binary data
biff2sunraster	convert BIFF image to Sun rasterfile
biff2tiff	Convert an image from BIFF to TIFF format
biffConvert	convert image between different pixel types
biffDump	dump a part of an image to stdout or file
biffcopy	copy a biff file
biffinfo	extract information from BIFF file
biffmerge	merge several images into one image
biffswap	swap byte order for bands in biff-file
bifftext	append a text line or edit text on image
biffwrite	write text on a biff image
butterworth	make an "ideal", truncated, lowpass filter in the spatial or Fourier domain
canny	detect edges in image using Canny operator

cdoc	Extract documentation from comments in c-programs.
closestMinMax	closest of minimum and maximum, noise reduction
color2ascii	dump a colortable in ascii format
colorquant	qolor quantization and rgb to pseudocolor conversion
combine	combine two images
complexConjug	Take the complex conjugate value of an image
contour	make a contour map
conv3x3	convolution between image and 3x3 matrix
convolve	perform an n x m convolution
crossSection	Find cross sections, row or column of image
crossing	find zero crossing or any other crossing
dither	create a dithering image
divide	divide two images pixel by pixel
exponential	make an "ideal", truncated, lowpass filter in the spatial or Fourier domain
extrema1	detection of local minima and maxima
extrema2	detection of local minima and maxima
fft2d	Two dimensional Fourier Transform
fftAmp	Take the absolute value of an image
fftDisplay	Take logarithm of BIFF band (compress dynamic range)
fftImag	extract imaginary part of complex band
fftMagPhase	2D Fourier transform and conversion into magnitude and phase
fftMakeBp	make an "ideal", truncated, lowpass filter in the spatial or Fourier domain
fftMakeBs	make an "ideal", truncated, lowpass filter in the spatial or Fourier domain
fftMakeHp	make an "ideal", truncated, lowpass filter in the spatial or Fourier domain
fftMakeLp	make an "ideal", truncated, lowpass filter in the spatial or Fourier domain
fftMult	multiply two bands pixel by pixel
fftPhase	phase of complex image
fftPower	take the power of each pixel value
fftReal	extract real part of complex band
fht2d	2 dimensional fast Hartley transform
fhtPhase	find Fourier transform phase from Hartley transform
fhtPower	calculate power spectrum from Hartley transform
fork_xshow	an example of how to pass data to xshow
fractile	noise reduction filtering
ft2ht	2d fourier to hartley conversion
gammaAdjust	Simple gamma correction
glRunLength	Grey value distribution of run lengths
glcm	Computes Grey Level Cooccurrence Matrix
glcmParameter	Computes different features from the Grey Level Co-occurrence Matrix (GLCM)
grad	gradient like operators
gradInv	gradient invers noise reduction
gradient	gradient like operators
haar	Two dimensional forward or inverse Haar transform
haar2d	Two dimensional forward or inverse Haar transform
haarTexture	texture extraction from 2D Haar transform
highpass	make an "ideal", truncated, lowpass filter in the spatial or Fourier domain

histo2ps	make a Postscript plot of a histogram
histoEq	histogram equalization
histoEqCol	histogram equalization color table
histoMeanStd	linear scaling to desired mean and std.
histoNorm	histogram normalization
houghCircle	Hough transform to detect circles
houghLine	Hough transform for line detection
ht2ft	2d hartley to fourier conversion
ihs2rgb	Convert from IHS to RGB
imag	extract imaginary part of complex band
image_minmax	find minimum and maximum in an image
isoCluster	ISODATA clustering, unsupervised classification
kMeansCluster	Kmeans clustering
kncn	k nearest connected neighbour noise reduction
knn	k nearest neighbour noise reduction
lapZeroCross	find zero crossing of lapacian
laplace	edge and line detection
linearTrans	linear image to image transformation
lit	Local Information Transform texture measure
litSnn	texture measure, combination of lit and snn
logarithm	Take logarithm of BIFF band (compress dynamic range)
logical	logical pixel by pixel operations
lowpass	make an "ideal", truncated, lowpass filter in the spatial or Fourier domain
lowpassIdeal	make an "ideal", truncated, lowpass filter in the spatial or Fourier domain
makepseudo	Make a pseudocolor image with corresponding colortable from rgb color image
maxHisto	texture measure
maxSimilar3x3	maximum similarity 3x3 noise reduction
maxarea	max operator over a local window
maxima	detection of local maxima
mbkncn	multi band k nearest connected neighbour noise reduction
mbknn	multi band k nearest neighbour noise reduction
mct	Create a color table from ascii data.
mean	compute local mean
median	noise reduction filtering
minarea	min operator over a local window
minima	detection of local minima
minmaxTexture	Min-Max filters for texture measurement
mkFractalSra	make fractal brownian motion noise
mkGauss	make an image with gaussian random noise
mkHisto	make and print a histogram
mkImg	make an image with specified size, pixel type and value
mkMorph	make a morphological structuring element
mkPoisson	make an image with Poisson random noise.
mode	noise removal and edge sharpening
morphClose	morphological grayscale operations on an image
morphDilate	morphological grayscale operations on an image
morphErode	morphological grayscale operations on an image
morphOpen	morphological grayscale operations on an image
mosaic	combine several images to a new one band image

multGauss	multiply image with random gaussian noise
multiply	multiply two bands pixel by pixel
negate	negate an image
newtitle	insert new title into an image
overlap	calculate overlapping band areas
peanoScan	scan an image in "peano" order
phase	phase of complex image
pixel_mapper	map pixel values into new value
png2biff	Convert an image from PNG format to BIFF format
pnm2biff	Convert a pnm (ppm, pgm or pbm) image to BIFF format
power	take the power of each pixel value
prewitt	gradient like operators
profile	Calculate pixel values along a line
pseudo2rgb	convert 1band image + colortable to rgb image
pseudomedian3x3	median like noise reduction filtering
pyramid	create a pyramid data structure
quadratic	quadratic geometric transform
rainbow	create a color map based on ihs
rank	noise reduction filter
raw2biff	convert raw data to BIFF file
real	extract real part of complex band
regionAnalyse	split BIFF image into regions and describe regions
regionConvexHull	find the convex hull of regions
reorganize	rotate, mirror or transpose an image
resample	stretch or compress an image
rgb2ihs	Convert from RGB to IHS
rms	root mean square difference between two images
roberts	gradient like operators
rotate	rotate an image
saturmod	modify saturation in color map file
scale	linear scaling of pixel values
scatter	make a scatterplot
segmRandom	Create image with two regions, random border.
segmSpannWilson	segmentation by the Spann and Wilson method
shift_img	Set new origo on a BIFF image
sigma	noise reduction filtering
signDiff	signed difference between two images
snn	symmetric nearest neighbour noise reduction
sobel	gradient like operators
square	take the square of each pixel value
squareRoot	take the square root of each pixel value
statistics	extract statistical information from an image
stdev	calculation of local standard deviation
stdiff	local statistical differencing
stvar	local variance
subcopy	copy a part of an image into a new image
sunraster2biff	convert Sun rasterfile to BIFF-file
thresBernsen	local variable treshold by the method of Bernsen.
thresLloyd	threshold by the method of Lloyd
thresMinErr	minimum error threshold.
thresMl	multi level supervised thresholding
thresMlCentroid	multi level thresholding by local centroid method
thresMlComCur	multi-level thresholding by Reddi using complexity curve

thresMlReddi	multi-level thresholding by Reddi et al.
thresMlWaHa	Finds multiple thresholds by Wang & Haralick method
thresOtsu	threshold by the method of Otsu
thresPtile	threshold at specified percentage
thresRidCal	threshold by the Ridler and Calvard method
thresSigma	threshold at fraction of standard deviation
thresUniErr	Uniform Error Threshold
threshold	threshold an image with specified threshold
tiff2biff	Convert an image from TIFF to BIFF format.
transpose_img	transpose every band of a BIFF image
warp	polynomial control point mapping
window	Make a BIFF image containing a window function
xadd	add two BIFF images and display under X11
xfft	X based interactive 2D fourier transform and display
xfilter	filter image and display power spectra
xhistogram	Show the histogram of an image in XITE.
ximage_example	example application for the XITE ximage toolkit
xmovie	X-based animation (movie) program for BIFF images in XITE
xpyramid	pyramid representation of image
xregion	interactively draw regions in an XITE overlay image
xshow	XITE X11 displayprogram for images and GUI for image processing
xwarp	compare two images
zernike	zernike moment image of a gray scale or binary image

Bibliography

[1] Svein Bøe. XITE – X-based Image Processing Tools and Environment – Programmer's Manual for version 3.4. Report 92, Image Processing Laboratory, Department of Informatics, University of Oslo, P. O. Box 1080 Blindern, 0316 Oslo, Norway, June 1998, http://www.mn.uio.no/ifi/english/research/groups/dsb/resources/software/xite/ProgrammersManual 2, 15, 39

[2] Svein Bøe. XITE – X-based Image Processing Tools and Environment – System Administrator's Manual for version 3.47. Report 91, Image Processing Laboratory, Department of Informatics, University of Oslo, P. O. Box 1080 Blindern, 0316 Oslo, Norway, June 2004, http://www.mn.uio.no/ifi/english/research/groups/dsb/resources/software/xite/SysAdmMan/. 2, 15

[3] Svein Bøe, Tor Lønnestad, and Otto Milvang. XITE – X-based Image Processing Tools and Environment – User's Manual for version 3.47. Report 56, Image Processing Laboratory, Department of Informatics, University of Oslo, P. O. Box 1080 Blindern, 0316 Oslo, Norway, June 2004, http://www.mn.uio.no/ifi/english/research/groups/dsb/resources/software/xite/UsersManual/. 15

[4] H.I. Christensen and J.L.Crowley, editors. *XITE – X-based Image Processing Tools and Environment*, pages 63–88. World Scientific, Singapore, 1994. ISBN 981-02-1510-X. 2

[5] Tor Lønnestad. The BIFF Image Concept, File Format, and Routine Library. Report 29, Image Processing Laboratory, Department of Informatics, University of Oslo, P. O. Box 1080 Blindern, 0316 Oslo, Norway, February 1990. 54

[6] Otto Milvang and Tor Lønnestad. An image widget for image processing. *The X-Journal*, 1(2), November/December 1991. 2

[7] Otto Milvang and Tor Lønnestad. An object oriented image display system. In *Proceedings, 11th International Conference on Pattern Recognition*, pages D218–221, The Hague, The Netherlands, August/September 1992. 2

[8] Adrian Nye. *Xlib Programming Manual*. O'Reilly & Associates, Inc., 1989. 31

Index

REPORTS FROM THE IMAGE PROCESSING LABORATORY

78 Maurycy Szmurlo :
"A comparative study of statistically classifiable features
used within the area of Optical Character Recognition"
Cand.Scient. (Master) thesis May 1995.

79 Ingvil Hovig :
"Verktøy og metoder for komprimering av MR bilder"
Dr.Scient. (Ph.D) thesis August 1995.

80 Øyvind Akerhaugen :
"Automatisk plassering av navn på kart ved hjelp av simulated annealing"
Cand.Scient. (Master) thesis February 1996.

81 Marius Midtvik :
"Reversibel komprimering av MR bilder basert på statisk kildemodellering"
Cand.Scient. (Master) thesis May 1996.

82 Tor Øyvind Didriksen :
"Linjefinning og klassifikasjon med Random Hough-transform
— en eksperimentell studie"
Cand.Scient. (Master) thesis June 1996.

83 Edward Allen Smith :
"Image Processing Techniques on DNA Fingerprint Images
and its Application to Genetic Similarity Analysis"
Cand.Scient. (Master) thesis August 1996.

84 Parviz Heydari :
"Line Following in Digitized Map"
Cand.Scient. (Master) thesis November 1996.

85 Ramin Gordjianfar :
"Vectorization of the Cartographic Data"
Cand.Scient. (Master) thesis November 1996.

86 Luren Yang and Torfinn Taxt :
"Robust Methods for Sonar Bottom Detection"
February 1997.

87 Christian Wladimir Hansson :
"Strukturgrammatikk — en høyeredimensjonal grammatikk
for syntaktisk mønstergjenkjenning av 3D objekter i bilder"
Cand.Scient. (Master) thesis December 1996.

88 Sverre H. Huseby :
"Video on the World Wide Web —
Accessing Video from WWW Browsers"
Cand.Scient. (Master) thesis February 1997.

89 Håvard Lauritzen :
"Raster til vektor konvertering ved simulert størkning"
Cand.Scient. (Master) thesis May 1997.

90 Irene Rødsten :
"Texture Segmnetation using Moment based Features
obtained by Locally Adaptive Thresholding"
Cand.Scient. (Master) thesis May 1997.