Gipstool† and Xgips† at a Glance

gipstool [**–W** *SunView options*] [**–KNc**] [**–C** *map*]
and [**–n** *colors*] [**–Iamo** *file*] [**–e** *n n n n n*]
xgips [**–xy** *size,org,off*]
[**–XY** *size,left,right*] [**–zZ** *zoom*] [*name*]

Options

All options beginning **–W** are passed directly to the window manager. See *sunview*(1).

–*depth*
forces the image to be stored in a 1-bit (**–1**), or 8-bit (**–8**) frame buffer.

–C *name*
override the "*colormap name*" of the image window (see **–a**).

–I *file*
name of a sub-command script file to be executed at startup time. (default is "~/*.gipstoolrc*").

–K detach the program from its parent process once the image has been read.

–N use a *dynamic* (private) colormap.

–a *file*
name of a file containing 256 red, green, and blue *colormap* intensities to use when displaying image pixels.

–c when the input is from a disk file, read only the *xsize*×*ysize* sub-image specified by the **–x** and **–y** options.

–e *n n n n n*
error-diffusion coefficients used with the **–Z0** option.

–m *file*
name of a General Image file whose header will be merged with that of the input image while reading the latter. Alternatively, *file* can be a list of header variables in the format "*name=value*", and those values will be merged with the input image header.

–n *colors*
number of colors (a power of 2) to use to display the image (default is 128 for *gipstool*, 64 for *xgips).*

–o *file*
all keystroke and mouse commands will be copied to *file*.

–r *buffers*
number of buffers to use when reading an image that was filtered through *ihgfmt*(1).

–x *xsize,xorg,xoff*
–y *xsize,xorg,xoff*
determine how much of the input image is to be saved in virtual memory, i.e. *xsize* pixels starting at *xorg*, and *ysize* pixels starting at *yorg* will be stored in virtual memory. Within this array, the display will begin at offset *xorg* and *yorg*, but the remainder of the array can be displayed via the pan commands.

–X *Xsize,Xleft,Xright*
–Y *Xsize,Xtop,Xbottom*
image area will occupy *Xsize* × *Ysize* screen pixels, with borders of *Xleft*, *Xright*, *Ytop*, and *Ybottom* screen pixels.

–z *xzoom*
–Z *yzoom*
x and *y* zoom factors. **–Z0** (monochrome images only) uses an *error diffusion* algorithm.

Pixel Values

Each pixel is first transformed according to the value of the *dscale* header variable:

s $pix \rightarrow \text{float}(pix)$

n $pix \rightarrow dfact \times pix + dzero$

l $pix \rightarrow \log_{10}(pix)$

p $pix \rightarrow dparm_0 + dparm_1 \times pix + dparm_2 \times pix^2 +$
$+ \ldots$ (*npoly*+1 terms)

f $pix \rightarrow n$, $f[n] \le pix < f[n+1]$, for *f* in the file *dfile*.

Then, unless *dscale=f*, the pixel is further transformed by

$$pix \rightarrow 256 \times (pix - dmin)/(dmax - dmin),$$

where *dmax* and *dmin* default to 0 and 256, respectively, if not defined in the input image header. If the resulting pixel value is less than 0, it is set to zero; if more than 255, it is set to 255.

Sub-Command Keystrokes

^A pan to the left edge of the image.

^B pan to the left (see **P**).

^C terminate the program. You will be asked to confirm this action.

^D draw line from the image mark to the cursor location, and reset the image mark to this pixel.

^E pan to the right edge of the image.

^F pan to the right (see **P**).

^G run *gipsmongo*(1) in a separate window.

^H prompt for a keystroke and display information about the corresponding sub-command.

^I display image information.

^K remove all graphics created under the current graphic index.

^L re-display the image and all graphics that are currently turned "*on*".

^M remove all pop-up windows and error messages.

^N pan downward (see **P**).

^O reset the image mark to the cursor location; enter "*rubber band*" sub-image selection mode.

^P pan upwards (see **P**).

^Q toggle cursor cross-hairs (*gipstool* only).

^R scroll the list of *tagged* image pixels (see **{**, **T**, and **R**).

^S reset the image marker to the cursor location; enter "*rubber band*" line drawing mode.

^T pan to the top of the image.

^U pan to the bottom of the image.

^V display the coordinates and real-world location of the cursor (same as the MIDDLE mouse button).

^W display a color wedge insert.

^X make the cursor location the top left corner of the image display area.

^Y make the cursor location the bottom right corner of the image display area.

! invoke "*/bin/sh*" to execute a shell command.

. prompt for a pair of pixel addresses, then position the cursor on that pixel. Address formats are

† © 1986–1992, Massachusetts Institute of Technology.

n absolute (unzoomed) image pixel
±*n* image pixel relative to cursor
:*n* absolute screen pixel address
<*n* screen pixel relative to cursor
>*n* screen pixel relative to cursor
f realworld coordinate

/ prompt for the name, and optionally an **"="** sign and replacement value, for a GIPS header variable.

ignore all keyboard input until the next carriage return.

< execute a file containing sub-commands, one per line.

> begin or end echoing sub-commands to a disk file or pipe.

1 assign a sub-command to the LEFT mouse button.

A reset the text drawing angle, in degrees counter-clockwise from vertical.

B reset the background color: a triplet of RGB integers in the range 0–255.

C reset the graphic color value: an integer in the range 0–255 specifying the colormap value for subsequent graphics.

D hide the current graphic plane.

F reset the drawing font for all subsequent text.

G reset the current graphic index: an integer in the range 0–255.

H write the colormap and pixel histogram to the specified file or socket, in *General Image* format. The histogram is taken over the sub-image defined by the image mark and the current cursor location. It contains 256 rows and 7 columns:

- pixel index (0 through 255)
- realworld pixel value
- red colormap value
- green colormap value
- blue colormap value
- # pixels with this value
- # pixels <= this value

I select the rectangular sub-image defined by the image mark and the cursor location; write it to a disk file or socket.

J write the line of pixel values stretching from the image mark to the cursor location to the specified disk file or socket as an *n*×7 General Image whose rows represent the *n* individual pixels in the line and whose columns represent

- the pixel count,
- integer x-value
- real x-value
- integer y-value
- real y-value,
- integer pixel value
- real data value

K write the current colormap to a file or pipe.

L direct output from the **l** command to a disk file or pipe. Omitting the file name suspends this function.

M read a new colormap file.

N reset the numeric precision of all displays of floating-point numbers.

O reset the text origin code: an integer 0–8, specifying the relationship between the cursor location and the origin of graphic text strings:

1 2 3
4 0 5
6 7 8

P set the pan increment: the number of screen pixels that the display moves for **^B**, **^F**, **^N**, and **^P**.

R point cursor at tagged pixel location (see **T**).

S apply contrast-stretch (0–100%) to the colormap.

T prompt for a *tag* name by which to remember the cursor location (see **R**).

W select width of paint brush for **p** command.

Z set the zoom insert size: supply the following parameters for the MIDDLE mouse button action:

- number of screen pixels
- additional zoom factor
- colormap index of cross-hairs

a prompt for changes to *xaxis* and *yaxis* parameters, and draw both axes.

f set or clear the window frame title.

h convert an in-core GIPS header variable value to ASCII and write it into the current graphic.

l write the current pixel's coordinate and data values to the logging file (see **L**).

m draw grid lines on an image that has non-separable coordinates.

p paint pixels in overlay plane.

t write a text string into the current graphic.

u undo the most recent graphic operation.

v convert the current pixel coordinate (**x**, **y**, and/or **d**) to ASCII and write it into the current graphic.

w write overlay plane to disk file as GIPS image.

x update *xaxis* parameters and draw the *x*-axis.

y update *yaxis* parameters and draw the *y*-axis.

{ read a file of tag names and display it in a pop-up window.

} save the current tag name list in to a file.

Axis Drawing Parameters

The following variables are used in *xaxis* and *yaxis* header strings and when prompted by the **a**, **x**, and **y** commands.

mark= maximum number of scale labels.

tic= number of tick marks per scale label.

font= font for scale annotation.

place= where to write the axis; one or more characters:

l left of center, or
c centered, or
r right of center.

i inside image area, or
o outside image area.

t top of image, and/or
m within the image, and/or
b bottom of image

loc= scale value at which to write **m**iddle titles.

from=
to=
by= where the scale values are to be drawn; integer values imply pixel coordinates, floating-point notation implies realworld coordinates.

line= non-zero to connect scale marks.

form= the *printf*(3S) format with which to draw the scale annotation.

ticlen= length in screen pixels of the scale and tick marks.

scale= special scale-label formatting:

s simple fixed or float numbers
l ten to the power of ...
d degrees, minutes, and seconds
h hours, minutes, and seconds