Semper 6

COMMAND

REFERENCE

Synoptics

Trademarks:

Semper is a registered trademark of *Synoptics Limited*
IBM PC is a trademark of International Business Machines Inc
Hewlett-Packard LaserJet+ is a registered trademark of Hewlett-Packard
Postscript is a registered trademark of Adobe Systems Incorporated
Microsoft is a registered trademark of Microsoft Corporation
Mouse Systems is a trademark of MSC Technologies, Inc
Siiicon Graphics Personal IRIS 4D is a trademark of Siiicon Graphics Inc
Sun is a registered trademark of Sun Microsystems Inc
VAX is a registered trademark of Digital Equipment Corporation

Manual number: SEM002
Update number: 0
Printed: August 1989

Table OF CONTENTS

PART 1

Chapter 1: Introduction

Chapter 2: Command Summary

Appendix E: Error Messages

Appendix F: Protected and Fixed Variables

Appendix G: Pixel Connectivity

Appendix H: Illumination

Appendix I: Customer Report Form

Appendix J. Ascii Key Codes

Chapter 1

INTRODUCTION

Overview

Semper 6 is a high-level language that has been developed to suit universal image processing applications. It consists of a large number of commands that can be used individually, or built into a program. It is *flexible* and provides the building blocks to construct your own individual solutions to your image processing problem.

About this manual

This manual provides a complete reference to Semper commands and is designed to complement the Semper on-line help.

How it's organised

The remainder of this manual is divided into two chapters and nine appendices:

- *Chapter 2: Command Summary* gives a list of Semper commands by function. This allows you to identify the group of commands that are required for a particular task.
- *Chapter 3: Semper 6 Commands* provides a detailed description of commands and command syntax. The commands are listed in alphabetic order.
- The appendices at the end of this manual include *Appendix E: Error Messages* which lists and explains Semper error messages.

This manual itself is split into two volumes:

- *Command Reference Part 1* contains introductory chapters and Semper commands *A* to *O*
- *Command Reference* Part 2 contains Semper commands *P* to *Z* and appendices.

What you need to know

This manual assumes a working knowledge of image processing terms and basic experience in the use of Semper 6 and Semper concepts, for example, *picture* and *device management*. If you are new to Semper 6, consult one of the manuals described in the last section of this chapter: *What else to read*.

Conventions used in this manual

Examples of Semper commands are shown in the following font:

```
semper
```

except when they are embedded in the text, in which case they are highlighted using a **bold** font.

This manual shows commands in their *unabbreviated* form. However, you can abbreviate all Semper commands (and parts of commands) to three characters. For example, the command:

```
pan cre pos 2,3
```

means the same to Semper as:

```
panel create position 2,3
```

and both these commands mean the same to Semper as:

```
pantaloons crease positively 2,3
```

In this guide each Semper command is described under its command heading, giving the following details:

- command syntax
- summary of command functions
- examples of use
- full description of command
- notes
- defaults and ranges

The sections below explain the format of the command syntax, notes, defaults and ranges.

Command syntax

Semper command syntax is divided into two parts:

- keys
- options

A *key* is added to a command to make its meaning more specific. Often a key limits the way in which a command works, for example, by applying the command action to a specified subregion only. A key *value* is required for each key.

An *option* does not require a value and is usually an instruction to alter the way in which a command works. For example, the **mask** command includes the options **inside/outside**. These options determine whether the command works inside or outside a mask boundary.

Each command syntax in this manual is given as in the **mask** command below:

mask

keys:	**[from]**	*<number>*	source picture
	[to]	*<number>*	output picture
	radius	*<number>*	radius of circular mask
	position	*<x>, <y>*	centre position of circular mask
	width	*<number>*	width parameter for 'soft edge' of circular mask
	with	*<number>*	*Plist* picture containing mask boundary curve
	value	*<n1>, <n2>*	reset pixels to specified value
	mark	*<number>*	mark circle, or polygon if ***with***
		<yes *or* no>	mark circle, or polygon if ***with***
options:	**outside/inside**		reset pixels outside or inside a mask

The text shown in angular brackets, such as *<number>*, describes the type of value to be supplied with a key. For example:

```
mask radius 25
```

is a valid use of the key **radius** with the **mask** command. You can also supply variables and expressions that give a numerical value. For example:

```
mask radius x+y
```

```
mask radius 3*z
```

Keys given in square brackets **[key]** can be omitted and are still understood by Semper. For example, the keys **from** and **to**, meaning *source picture* and *output picture* respectively, are used so often with Semper commands that they do not need to be stated explicitly:

```
mask 1 2
```

is the same as:

```
mask from 1 to 2
```

Sometimes you will only see square brackets [ ] that do not enclose a key. This means that you cannot supply a key, only a value or set of values. For example, the syntax :

calculate

keys:	[ ]	*<expression>*	*expression to be calculated*
	to	*<number>*	*output picture*

means that you must supply the text of an arithmetic expression for the **calculate** command. For example:

```
calculate (:10+:11)/2 to display
```

Note that Semper includes a number of keys and options that are called *general*, in that they are accepted by every command, for example, the options **byte**, **integer**, **fp** and **complex**. For details of these keys and options, refer to *Appendix C: Semper Keys and Options*.

Defaults and Ranges

The defaults and valid range of values for each command are given in a table at the end of each command description as is shown below:

keys/options	defaults	range
[from]	current picture, held in variable *select*	valid picture number
[to]	source picture	valid picture number
radius	two thirds of the minimum source dimension	real number
position	centre of source picture	within bounds of picture (real numbers)
width	width 0	real number
with	circular mask	valid picture number
value	source picture average around mask boundary	real number
mark	mark off	see *Appendix C*
outside/inside	outside mask	

Defaults

The defaults for each **key** are given in the *defaults and ranges* table. The default of each **option** is not usually shown in the table as, in general, the default is off. A default is shown, however, if the default is *on*, for example, the default for **letter** and **border** is *on* and these are shown in the relevant

table. Note that to turn an option *off* you simply prefix it by **no**, for example, **noletter**, **noborder**, **noverify** etc. A default is also shown for an option if there is a choice, as in the **outside/inside** options in the above table for the **mask** command.

Ranges

The valid range of values that can be supplied with each key are also defined in the defaults and ranges table. Some ranges are not straightforward, for example, the **from** and **to** keys usually expect a *valid picture number*, which can be a simple integer:

```
mask from 42 to 43 inside
```

or can be a combination of device number and picture number separated by a colon:

```
mask 1:42 to 1:43 inside
```

which refers to pictures 42 and 43 on device 1. Alternatively, Semper allows you to refer to pictures 42 and 43 on device 1 as 1042 and 1043, so an equivalent command line is:

```
mask 1042 to 1043 inside
```

Note that many ranges are machine dependent. For example, some machines can only display a limited number of colours, although Semper can define many more. You may see the following range for a key:

integer in range 1 to system limits (type **show system**)

which means that the range is determined by your particular type of installation and the command **show system** provides details of your system limits.

Notes

After the description of each command, you will see a section called **Notes** which draws your attention to some features or restrictions of a command. An explanation of some of the notes is given below:

restrictions:

image sizes must be powers of two: the command requires that the size of the source picture is a power of two, for example, size 256 by 128.

factorisable size: the command requires that the size of a source picture can be factorised into factors of 2, 3, 4 and 5, with at least one factor 4.

unsuitable for direct output to display: the data generated by a command needs to be stored on file before it is sent to the display. This is because the display cannot store some forms of data accurately, for example, data with a range 0.1 to 0.5.

multi-layer pictures:

faulted: the command cannot accept multi-layer pictures.

layers processed independently: this means that each layer is processed as a separate 2-D image

forms used internally: *data form*

This note refers to the form used to process data, regardless of the form of the source data. The possible forms are *byte*, *integer*, *floating point* (*fp*) and *complex*. See *Appendix A: Picture Types* for further detail of picture forms.

display marking: *type of mark*

This note describes how a display is marked by the command if you use the **mark** key. For example, the **mask** command marks the mask boundary on the screen.

variables set: *variable name*

Like most languages, Semper makes use of variables to store values, for example, a variable called *select* holds the number of the current picture and a variable called *cd* holds the number of the current device. This note details any variables whose values are set by a particular command.

variables used: *variable name*

This note details any variables that are used by the command in its processing.

What else to read

For information on the Semper system, read the:

Semper 6 Guide

This guide is a collection of documents that detail each aspect of Semper. It includes the following sections:

- *Beginners' User Guide*
- *Advanced Users' Guide*
- *Quick Reference List*
- *User Interface Guide*
- *Tutor User Guide*

If you are new to Semper 6, first read the *Beginners' User Guide*. This document provides structured examples and illustrations of the Semper language. The *Advanced Users' Guide*

provides a more detailed and comprehensive description of Semper and its facilities. For an introduction to Semper 6 *Plus* user interface creation read the *Tutor User Guide* and the *Semper 6 Plus User Interface Guide*.

If you require a general introduction to the field of image processing, we recommend:

Digital Image Processing
by Rafael C. Gonzalez and Paul Wintz
published by Addison-Wesley, 1987
I.S.B.N. 0-201-11026-1

For an overview of *Remote Sensing* (Semper 6 contains a Remote Sensing package), we recommend:

Remote Sensing Digital Image Analysis: An Introduction
by John A. Richards
published by Springer-Verlag, 1986
I:S.B.N. 3-540-16007-8 Springer-Verlag Berlin Heidelberg New York
I.S.B.N. 0-387-16007-8 Springer-Verlag New York Heidelberg Berlin

On-line help

You can also refer to Semper on-line help for definitive information of a command. For example, for help on the **mask** command, start Semper and type the following command at the terminal:

```
help mask
```

For a description of **mask** syntax, type:

```
help mask.syntax
```

For the complete on-line help available for **mask**, type

```
help/full mask
```

For a list of all the commands and any other topics for which help is available, type:

```
help/topics
```

For further information, refer to the **help** command entry in this manual.

Chapter 2

COMMAND SUMMARY

Overview

This chapter gives a summary of the Semper commands that are detailed in this manual. The commands are grouped according to function, rather than alphabetically. This allows you to isolate the commands that are suitable for a particular task, for example, particle analysis. Figure 2-1 overleaf illustrates the functions of Semper commands.

This chapter also lists commands that are installation-specific, that is, commands that apply only to a particular hardware combination. These commands are mainly concerned with framestore control and image capture.

For a list of commands in *alphabetic* order and full details of each command and syntax, refer to *Chapter 3: Semper 6 Commands*.

A list of commands by function is given below.

Variables and terminal information

name=value	sets the named key/option/variable to a specified value
ask	asks for a value from the terminal with a user-defined prompt
beep/bell/buzz	sounds bell on the terminal
cls	clears the terminal screen and places the cursor at home position
echo	directs logical output streams to the terminal and/or log files
help	provides on-line documentation during a session
inkey	waits for a terminal key to be pressed and returns key value
local	declares a variable to be restored after program execution
log	prints text/values in the log file
page	controls formatting and paging of terminal output
p(ixel)	resets an individual pixel to a user-defined value
pointer	sets gearing and sensitivity of pointing device (ie. mouse)
report	prints message for last reported/trapped error
show	prints information on variables set, devices assigned, time etc.
syntax	prints keys and options for a given command
time	prints elapsed time in seconds
type	prints text/values on terminal
unset	deletes a variable

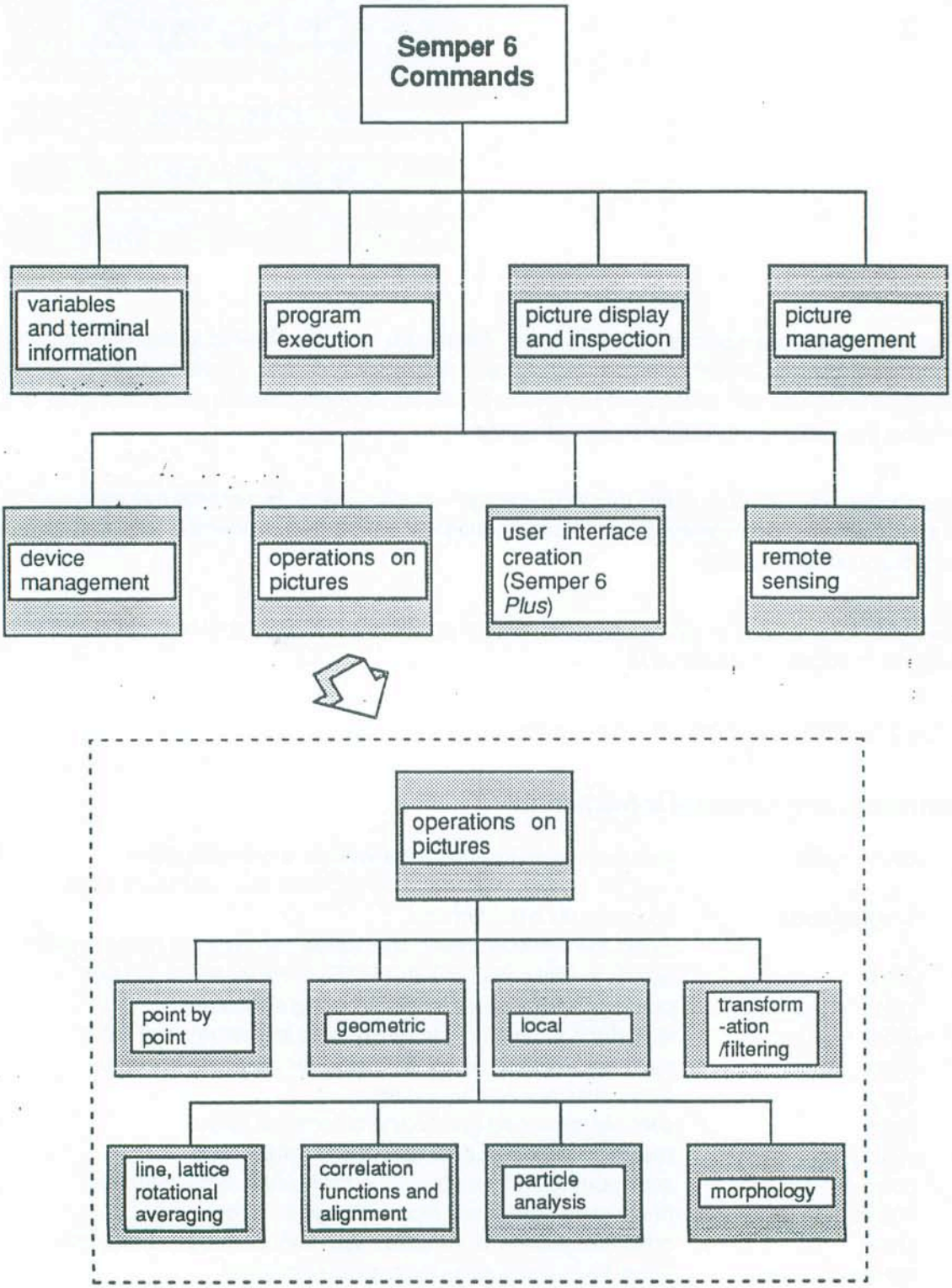

Figure 2-1. Semper 6 Commands By Function

Variables and terminal dialogue

name=val	Sets named key/option/variable to given value
ask	Seeks value(s) from terminal with user-defined prompt
beep	Sounds bell on terminal
bell	Sounds bell on terminal
buzzer	Sounds bell on terminal
cls	Clears terminal screen and places cursor at home position
diagnostic	Prints text (values to diagnostic output stream
echo	Directs logical output streams to terminal and/or log files
event	Provides access to Semper event queues
help	Provides on-line documentation during session
Inkey	Waits for terminal keys to be pressed and returns key values
local	Declares variables to be restored after program execution
log	Prints text/values to log output stream
page	Controls formatting and paging of terminal output
p(ixel)	Resets individual pixel(s) in picture to user-defined value(s)
pointer	Defines pointer (i.e. mouse) gearing and sensitivity
report	Prints message for last reported/trapped error
show	Prints information on variables set, devices assigned, time, etc.
syntax	Prints keys and options for given command
time	Prints elapsed time in seconds
type	Prints text/values to console output stream
unset	Unsets variable(s) permanently

Program execution

@name	Inserts named macro text in command line
@number	Inserts numbered macro text in command line (executes macro)
add	Adds new program(s) to program library
break	Resumes execution after end of **for** loop
edit	Edits contents of numbered macro
end	Last command in program or run file
exit	Halts session
for	Repeats subsequent commands (up to corresponding **loop**)
if, unless	Makes command execution conditional
jump	Resumes execution at given label
library	Executes program from program library
list	Prints program text or numbered macro text
loop	Terminating command **for** loop
macro	Causes rest of input line to be stored as numbered macro
next	Resumes execution at start of next cycle of **for** loop
order	Defines/prints search order for program libraries
quit	Halts session
rename	Renames program in program library
return	Returns from program or run file
run	Executes commands from named file (run file)
stop	Halts session
wait	Suspends execution until key pressed or specified time elapsed

Picture display/inspection

contour	Draws contour map on display (line graphical form)
display	Displays picture/region in various modes
drag	Drags line/arc/circle/sub-region/curve on the display with the cursor
examine	Prints picture size, class, data form, title, etc.
histogram	Generates/draws grey-level histogram of picture/region
ladjust	Provides mouse-controlled adjustment of look-up tables
lset	Generates/modifies look-up tables (colour scales, highlighting)
lut	Generates and manipulates monochrome/colour look-up tables
mark	Marks position(s), frame, circle or text on a display
ovread	Reads image stored in the overlay into a picture
ovwrite	Writes a binary image into the overlay plane
pfilter	Reduces the data in a position list picture after **sketch** operation
postscript	Writes picture/display region to file in PostScript format
print	Prints small picture region in numerical form
rgb	Converts a full colour image into monochrome/false colour form
sheet	Displays 2-D picture as height of solid sheet
solid	Displays thresholded 3-D picture as solid body
sketch	Records curve drawn free-hand on the display as a position list picture
spc	SuperPoses Contour levels on picture
survey	Scans picture/region and sets variables *min*, *max*, *mean*, *me2* and *sd*
view	Selects viewing field, zoom factor and (mono, false or colour) lut
xwires	Records position/direction/frame indicated via display cursor
ymod	Displays picture/region as surface height (line graphical form)

Miscellaneous

null	Default command, displays picture/region
pack	Prints packed integer values for given names
unpack	Prints three-character names for given packed integer values
user	Provides skeleton routine for user to adapt

Picture management

copy	Copies picture or program
create	Allocates/initialises picture storage (not normally needed)
delete	Deletes picture or program
letter	Generates lettering on a 5 by 6 pixel matrix
origin	Moves or resets picture coordinate origin
pcb	Stores picture coordinate range, class, data form in variables
picture	Returns picture parameters in variables
read	Reads picture from file outside Semper
reclass	Changes recorded picture class without altering data
renumber	Changes picture number without altering data
save	Saves picture in binary file outside Semper
select	Selects new current picture
title	Sets or alters picture titles
wp	Sets or unsets picture write-protect flag
write	Writes picture to file outside Semper

Device management

assign	Assigns memory, disc file, tape, display, log file, program library or help library as device
close	In some installations, outputs contents of display to hardware
compress	Compresses disc device, collecting empty space together
deassign	Deassigns device
directory	Prints directory state for disc device
erase	Erases image or overlay memory for display region
flush	Forces disc memory buffers to be copied to physical disc
partition	Assigns frames and positions for display partitions
ramps	Fills display region with repeated grey-level ramps
reinitialise	Re-initialises device, destroying any existing data
rewind	Rewinds tape device

Point by point operations

calculate	Calculates arbitrary arithmetical/relational/logical expression involving pixels, constants, variables and functions
correct	Forces picture modulus to supplied reference value
fit	Fits linear ramp to picture and subtracts/divides to remove ramp
gausslan	Generates single Gaussian-profile peak
lorentzlan	Generates single Lorentzian-profile peak
map	Processes picture with user-defined intensity mapping
mask	Resets pixels inside/outside circular or polygonal region
negate	Rescales picture interchanging *min* and *max*
noise	Generates (Gaussian or Poisson) noise-limited pictures
scale	Rescales picture, linearly or with histogram equalisation
threshold	Produce binary image given one or more intensity threshold

Geometrical operations

cut	Cuts region out of picture
expand	Horizontally resamples a picture
extract	Extracts sub-/super-region (magnified/rotated/skewed/warped)
find	Finds lowest/highest pixel or centre-of-mass of picture/region
fullplane	Converts half-plane Fourier picture to full-plane form
halfplane	Converts full-plane Fourier picture to half-plane form
lpd	Finds local maxima in a 1-D picture
magnify	Magnifies picture/region by integer factor
paste	Inserts one picture into another
peaks	Locates local peaks in picture and records positions in *Plist*
rotate	Rotates picture by large angles relatively efficiently
separate	Separates layers of multi-layer (3-D) picture as 2-D pictures
stack	Combines 2-D pictures as multi-layer (3-D) picture
transpose	Transposes picture (i.e. interchanges rows and columns)
turn	Performs simple picture reflections/rotations

Local operations

differentiate	Calculates picture derivatives via 3-point operator
edge	Applies magnitude or Roberts edge operator
fir	Applies arbitrary small block (FIR) filter
hp	Applies a square-block high-pass operator
lmean	Calculates local block mean (smoothes picture)
lsd	Calculates local standard deviation
lvariance	Calculates local block variance
rank	Applies local ranking (median, eroding or dilating) filter
rf	Applies two point recursive (IIR) filter for smoothing/sharpening
sharpen	Applies square-block edge-enhancing operator

Transformation/filtering

backproject	Back-projects 1-D picture into 2-D picture
ctf	Generates/applies electron-optical transfer function
fourier	Calculates Fourier transform (FTs)
hilbert	Calculates Hilbert transform
image	Re-calculates image from Fourier or Walsh transform
phr	Resets Fourier transform phases to random values
ps	Calculates a power spectrum (i.e. FT intensity)
walsh	Calculates Walsh transform
weight	Applies radial filter defined in 1-D picture
window	Applies window array filter (for lattice averaging)

Line, lattice and rotational averaging

base	Least-squares fits lattice parameters to list of lattice sites
flc	Fits Lattice Components to FT peaks (for lattice averaging)
lattice	Generates/marks positions of perfect lattice sites
motif	Calculates real-space average over given list of positions
project	Calculates 1-D projections or averages
section	Extracts radial sections averaged over arbitrary sectors
strain	Deduces local strain levels for distorted lattice

Correlation functions and alignment

acf	Calculates auto-correlation function
ocf	Calculates angular correlation function (for rotational alignment)
rcf	Calculates radial correlation function or phase residuals between two Fourier transforms
xcf	Calculates (spatial) cross-correlation function

Particle Analysis

analyse	Finds particles in picture according to given intensity thresholds
pcalculate	Calculates further parameters for given particle
pcurve	Calculates equivalent particle parameters for area inside curve
pdraw	Edits picture by drawing cursor-defined lines
pedit	Edits contents of particle parameter list and/or segmentation map
pextract	Extracts image of single particle from source picture
pferet	Calculates up to 9 feret diameters for given particle
phistogram	Generates histogram of given parameter for selected particles
pid	Returns particle id corresponding to given picture position
pmark	Marks selection of displayed particles
pset	Stores specified particle parameters in variables
pshow	Highlights selection of displayed particles
ptype	Prints parameters for selected particle(s)

Morphology

bclose	Closes regions in a binary image with a given structuring element
bdilate	Dilates regions in a binary image with a given structuring element
berode	Erodes regions in a binary image with a given structuring element
bhmt	Applies binary Hit or Miss transforms defined by structuring element
bmlut	Generates 3 x 3 neighbourhood mapping table for use with **bmmap**
bmmap	Applies 3 x 3 neighbourhood mapping to a binary image
bopen	Opens regions in a binary image with a given structuring element
chull	Generates the convex hull of the regions in a binary image
dclose	Closes regions in a binary image with a circular disc
ddilate	Dilates regions in a binary image with a circular disc
derode	Erodes regions in a binary image with a circular disc
dilate	Dilates simply or selectively, and enlarges objects without joining
dopen	Opens regions in a binary image with a circular disc
dt	Generates the distance transform for regions in a binary image
erode	Erodes simply or selectively, and generates outlines or skeletons
flood	Floods/marks regions in a binary image which overlap the seed image or seed points
hfill	Fills holes in regions in a binary image
label	Labels separate regions in a binary image
median	Smoothes binary picture
skiz	Generates the skeleton by zones of influence (skiz) of the regions in a binary or labelled image
zone	Generates the zones of influence of the regions in a binary or labelled image

Remote Sensing

box	Classifies picture using box or parallelepiped method
covariance	Calculates covariance of multi-spectral picture
destripe	Corrects picture for differences in line sensor characteristics
learn	Calculates statistics of training areas prior to classification
likelihood	Classifies picture using *maximum-likelihood* method
mindistance	Classifies picture using *minimum-distance-to-mean* method
pct	Calculates principal components transform of multi-spectral picture
rhistogram	Calculates 2-D histogram between bands of multi-spectral picture
warp	Corrects picture for geometric distortions

User Interface Commands

cell	Controls creation and behaviour of cell element
device	Returns limits of display, redraws screen, stacks cursor positions
execute	Defines actions that are executed before/after each Semper command sequence
justification	Controls positioning of displayed objects
menu	Controls creation and behaviour of menu element
mouse	Defines actions/positioning for mouse (pointing device)
panel	Controls creation and behaviour of panels
textfield	Controls creation and behaviour of textfields
uif	Controls loading, saving and execution of user interfaces

For further information please contact:

Synoptics Limited
271 Cambridge Science Park
Milton Road
Cambridge
CB4 4WE, UK
Tel: (0223) 423223. Telex:81417 INNCEN G.
Fax: (0223) 420020
Int: +44 223 423223.
Int Fax +44 223 420020

Synoptics Limited
164 Chief Justice Cushing Highway # 5
Cohasset
MA 02025
U.S.A
Tel: 0101 617 383 2289
Fax: 0101 617 383 2727

Chapter 3

SEMPER 6

COMMANDS

h

halfplane
help
hfil
hilbert
histogram
hp

i

if, unless
image
inkey

j

jump
justification

l

label
ladjust
lattice
learn
letter
library
likelihood
list
lmean
local
log
loop
lorentzian
lpd
lsd
lset
lut
lvariance

m

macro
magnify
map
mark
mask
median
menu
mindistance
monitor
motif
mouse

n

negate
next
noise

o

ocf
order
origin
ovread
ovwrite

p

pack
page
panel
partition
paste
pcalculate
pcb
pct
pcurve
pdraw
peaks
pedit
pextract
pferet
pfilter
phistogram
phr
picture
pid
p (pixel)
pmark
postscript
print
project
ps
pset
pshow
ptype

q

quit

r

ramps
rank
rcf
read
reclass
reinitialise
rename
renumber
report
return
rewind
rf
rgb
rhistogram
rotate
run

s

save
scale
section
select
separate
sharpen
sheet
show
sketch
skiz
solid
spc
stack
stop
strain
survey
syntax

t

textfield
threshold
time
title
transpose
turn
type

u

uif
unpack
unset
user

v

view

w

wait
waish
warp
weight
window
wp
write

x

xcf
xwires

y

ymod

z

zone