Semper 6
Training Course

Semper 6 Training Course Agenda

Preamble to Semper

Overview of the hardware developments that have made low cost image processing possible.

Discussion of framestore operation and some of the terminology and concepts used.

1. The Semper 6 Environment

The command interpreter.
The display management system.
The image filing system.
Simple processing operations.
The on line help system.

2. Lookup tables

Creating and loading lookup tables in Semper.
Monochrome luts.
Manipulating image contrast and brightness with monochrome luts.
False colour luts.
Customised false colour luts.
Hue, saturation and intensity mappings.
Interactive luts using the mouse.
Storing and recalling luts.

3. Image Coordinates & the Crosswires Cursor

Interacting with image data using the xwires cursor.
Frame, picture and partition co-ordinate systems.
Defining regions of interest in one and two dimensions.
Defining regions of arbitrary shape.

4. Storage Forms & the Image Calculator

Image storage forms; byte, integer, floating point and complex floating point.
Image arithmetic with the calculator.
Generating synthetic images with the image calculator.
Examples of various useful operators – ifelse, and, or.

5. Histograms and Histogram Equalisation

Calculating histograms in Semper.
The histogram as a measure of image quality.
Histogram equalisation.

6. Image Filters

Image filters in Semper.
Motivation for image filtering.
Basic high pass and low pass operations.
Rank filtering.

7. Image Analysis 1

Image analysis in Semper .
Using interactive luts to determine thresholds.
The analysis command.
Particle parameter lists and segmented images.
Annotating the image with image analysis results.
Extracting and tabulating image analysis results.

8. Image Analysis 2

Methods of filtering image analysis data to eliminate unwanted particles.
Highlighting particles of interest.
Using analysis parameters to edit particle parameter lists and segmented images.

9. Morphological Operations

Limitations of image analysis in certain situations.
Erosion and dilation operations to separate touching objects.
Skeletonisation and smoothing operations.

10. Image Resampling

Resampling with the **extract** command to change size, aspect ratio or position of image.
Use of *u* and *v* vectors for resampling with respect to arbitrary axes.
Use of this technique to rotate or deskew images.

11. Image Warping

Motivation for image geometry correction.
Map points, image points and the warp polynomial.
Examples of use.

12. Visualisation of 2D Data

Motivation for visualisation – for example, interpreting STM images.
Contour generation.
Wire frame – ymodulus generation.
Rendered surface display.

13. Image Fourier Transforms and Power Spectra

Motivation for calculating fourier transform.
Fourier transform calculation.
Power spectrum calculation and interpretation.
Compressing dynamic range for display.
Constructing and applying a simple filter in the fourier domain.

Semper Training

Worksheet No 1. The Semper 6 Environment

A Brief Overview

Semper provides a complete environment which is the same across a broad range of machines and has all the facilities that are needed to perform image processing operations. This environment comprises a:

- command interpreter
- display management system
- picture filing system
- broad range of processing and analysis operations
- on line help library
- high level menu generation system

This first worksheet will familiarise you with the Semper environment so that you can perform basic operations of picture management and display. Later worksheets will discuss further image processing ideas and techniques and introduce the Semper features that are required to support them.

Starting the Semper Session

To start the Semper session, follow the instructions below, that are relevant for your machine:

PCs

Insert the disk labeled "*Training Worksheet 1*" into the A drive.
Start Semper by typing:

```
semper /run=a:train1
```

Workstations

Start Semper by typing:

```
semper /run=train1
```

These commands start up Semper with the working environment for the training exercise.

The Command Interpreter

When Semper starts up, you see a startup message which displays the serial number of your copy and then a prompt sign:

```
S$
```

This signifies that the command interpreter is awaiting input. The command interpreter provides the lowest level interface between the operator and the system. Semper behaves in very similar way to other interpreted languages like BASIC. Its most important feature is that it is interactive, that is, you can try an operation and see the result immediately.

Try typing:

```
display from 3:1 to display:1
```

This is an example of the basic form of Semper processing operation that specifies an operation from a source to a destination.

Semper commands are often written in a shorter form. The interpreter only looks at the first three letters of any command. and the words **to** and **from** are often implied. So in its briefest form the previous command could be written as:

```
dis 3:1 dis:1
```

Each time a command is executed by the interpreter it is stored in a memory buffer. Any one of the previous ten typed commands can be recalled using up and down arrow keys to scroll through the buffer. The left and right arrows move the cursor and allow editing of commands that have been recalled. Characters can overwrite or be inserted into the recalled line. On the PC the <*Insert*> key toggles the command editor between the two modes. On VAX/VMS systems this is done with the *Control* <*A*> key.

***

Exercise

Recall the first command that you typed and append it with the word **negate** .

***

The result of the exercise should read as follows:

```
display from 3:1 to display:1 negate
```

This is an example of a command *option*. It is a feature that modifies the operation of a command and has a yes/no or on/off on character. Now try typing the following:

```
display 3:1 to display:1 size 64,64
```

This is an example of a command *key* . This modifies the operation of a command in a way that involves the specification of a numerical value.

The Picture Filing System

One of the most important concepts in Semper is that of the *picture disk*. The picture disk is the filing system where the digital images are stored and manipulated. Semper is a disk based processing language, this means that the source and destination images for processing operations are often picture disks. The organisation of images in the display is made to emulate that in the picture disks. Many picture disks may be present on a given machine, but Semper can only access those that have been 'assigned' and allocated a device number by which they are referenced.

Type the command:

```
show devices
```

A list of the devices that are currently assigned is reported.
Device 1 is reported as a framestore. We will return to this later.
Devices 2 is reported as a help library, again this will be explained later.
Devices 3 and 4 are reported as picture disks.
Device 5 is the program library.

Type the command:

```
examine 3:1
```

This gives you all the information about picture number 1 on device number 3. A picture is referred to by its picture number. Up to 999 pictures may be stored on any given disk.

The picture title, size, storage requirement, class, and write protect status is reported. Semper gives each image on this disk a different class. There are ten different classes of image possible. They will be introduced as they arise in later worksheets. The most common class is the *Image* class. This is a picture that can be displayed. The picture class label prevents incompatible operations between differing types of data.

You can obtain a directory of all the pictures on picture disk 3 with the command:

```
examine device 3
```

To copy a picture between the two assigned disks type:

```
copy to 4:1
examine 4:1 full
```

Note how Semper has remembered the picture that you are currently using so that it is not necessary to specify the number explicitly in the commands. The Semper variable *select* stores the current picture. Try typing:

```
type select
```

This variable is related to the picture number and device number by the following relationship:

*select= picture number+device number*1000*

As well as using the notation 4:1 to refer to picture 1 on device 4 it is also valid to use 4001, for example **examine 4001** is equivalent to **examine 4:1.**

You can label pictures with a title string. For example, you may like to try the following:

```
title 4:1 text 'Copy of Mona Lisa'
examine 4:1
```

Another useful operation includes write protecting a picture. Try the following:

```
wp 4:1
delete 4:1
wp off
```

So far the picture disks that were assigned when the Semper session started have been used in this worksheet.

New picture disks can be created with a specified name and size. As a general rule 1000 kilobytes is a useful size. This is enough for three full screen images.

The following can be used to create a new picture disk.

```
assign new name 'test' size 1000
```

***

Exercise

Create a new picture disk. Copy image 3:1 to the new disk, give the new picture a title, write protect it and then display it.

***

Picture Display

A typical framestore is able to display 256 different brightness levels. Typically level 0 is black and level 255 is white. The range of intensity levels in an arbitrary image can take on a very wide range of values. These are normally scaled by Semper to the range 0,255 when the image is loaded into the framestore memory during a display operation. This means Semper will display any image correctly irrespective of its range. Try typing the command:

```
survey 3:2 full
```

You will see that the range of brightness levels in the picture is reported. When the image is displayed the range information is used to scale the data in the framestore memory so as to make the lowest pixel value black and the largest value white.

Display the image with the command:

```
display 3:2 to display:1
```

then type:

```
examine display:1 full
```

The result of this operation is a reference to the image in the display. The grey level values corresponding to white and black are reported.

When the display operation takes place the values of the variables *min* and *max* are set to the two extremes of the range. The information in these variables is used to perform display scaling. Confirm this by typing:

```
type min,max
```

There are occasions, however, when the automatic scaling needs to be overridden. This can occur if you have an image with a large dynamic range. If very bright objects are present together with dim objects, the automatic scaling will prevent the dim objects being seen.

Display image number 3:3 by typing:

```
display 3:3 to display:1
```

You will see a single white dot in the middle of the screen. Now try to survey the image. This command sets *min* and *max* to the two range extremes.

```
survey 3:3
```

The maximum and minimum pixel values are reported. In order to see the dim objects we can override the display scaling and saturate pixel values above a certain threshold. This is analogous to overexposing photographic film.

Type:

```
max=max/100
display 3:3 to display:1 preset
```

Try decreasing the threshold again:

```
max=max/100
display 3:3 to display:1 preset
```

***

Exercise

Display a negated version of image 3:1 by setting the variables *min* and *max*, using the **display** command with the **preset** option.

***

Display Partitions

When images have been displayed, the notation **display:1** has been used for the display destination. This notation refers to a display partition, in this particular case display partition 1.

A partition is a user defined subregion of the screen. Any number of these partitions may be defined which can be of an arbitrary size and position. The currently defined partitions may be seen with the command:

```
show partitions
```

Initially one partition that occupies the full screen has been defined. A new one can be created with the command:

```
partition 10 size 256 top left
```

The border of the screen can be marked out. Try the command:

```
erase frame
mark partition 10 border
```

A picture is displayed in the new partition with the command:

```
display 3:2 to display:10
```

Semper refers to pictures in the display in a similar manner to a picture disk. The device number given to the display is always 1. Active partitions have a picture number associated with them. The contents of the display may be examined with the command:

```
examine device 1
```

Display partition 10 may be examined with the command:

```
examine full 1:10
```

***

Exercise

Create a new partition of size 256 square at the bottom left of the screen. Copy the contents of partition 10 into this new partition.

***

Another important feature of Semper display management is automatic display size scaling. If an image is too large to fit into a partition its size will be adjusted to fit. Try defining a small partition of size 64 and displaying a larger image there.

```
partition 12 size 64 bottom right
display 1 to display:12
```

Type the command:

```
examine full 1:12
```

This reports that the image has been subsampled by a factor of eight to fit the image into the partition.

If the partition is larger than the image by a factor 2 or more the image may be magnified on the display. Try typing the command:

```
display 3:2 to display:1 times 4
```

***

Exercise

Using the Semper command **lmean** (which blurs the image) display image number 1 and a blurred version of it, side by side on the display.

***

Overlays

The annotation on the display does not overwrite the image, but is stored in a separate plane of memory known as the overlay plane. It can be turned off or changed colour. Try the commands:

```
overlay black
overlay red
```

Setting Up the Startup Environment

As well as accepting input directly from the keyboard, the command interpreter can accept instruction from an external file. When a Semper session starts it reads commands from a startup file. This can be the standard default file or a user specified file. The default is named *semper.run* .

This file normally sets up the user working environment. To demonstrate the requirements for a basic environment, stop the current Semper session with the command:

```
stop
File....Exit (Windows)
```

Restart the Semper session without a startup file by typing:

```
semper /norun
```

The first statements in the startup file are 'assignment' statements which establish channels of communication used by Semper. Then type the following commands:

```
assign display
```

This command sets up communication with the framestore hardware.

```
assign help name 'semper.hlb'
```

This brings the on line help into operation with the command:

```
assign name 'a:train1' wp    (PCs)
assign name 'train1' wp      (Workstations)
```

Access to the training worksheet picture disk is obtained with the command:

```
cd=3
```

The Semper "current device" variable is set. This enables references to pictures on this disk to be made by the picture number only.

```
assign scratch size 1000
```

This command sets up a scratch working disk of size 1000 kilobytes. A scratch disk is deleted at the end of the Semper session.

```
partition 1 size 768,512
partition 2 size 384,256 top left
partition 3 size 384,256 top right
partition 4 size 384,256 bottom left
partition 5 size 384,256 bottom right
```

Display partitions are defined with these commands.

```
display 3:1 to display:1
lut 1 monochrome create
view lut 1 frame
```

Finally a monochrome lookup table is created and loaded into the framestore. The use of these commands will be explained in worksheet 2.

These commands are sufficient to set up a basic working environment.

Further Properties of the Command Interpreter

In addition to initiating the Semper processing operations the command interpreter has most of the features of a full programming language.

It supports variables and the full range of arithmetic operations between them. Try:

```
x=34;y=56;z=2;type x*y/z
```

Parameter input and output are available using the **ask** and **type** commands. Try typing:

```
ask 'Input Data 'd;type 'd=',d
```

Note that a semi-colon is used as a command delimiter.

All the common mathematical functions are supported by the command interpreter. For example, type:

```
x=pi/3;type sin(x)
```

A complete list of functions is documented in the *Semper 6 Command Reference* manual.

The **for** command is a simple way of repeating an operation a number of times. Try:

```
for x=1,10;type x,root(x);loop
```

This causes the numbers 1 through 10 and their roots to be printed.

The interpreter provides a means to make the execution conditional by prefixing it with an **if** or **unless** clause. For example:

```
for x=1,5;if x>2 type x;loop
```

Refer to the manual for more detailed information.

***

Exercise

Use a **for** loop to display the first four images on picture disk 3 into four adjacent partitions.

***

Help Library

When the on-line help library is assigned, help information on any of the Semper commands is available. Try typing:

```
help display
```

or for a more detailed version:

```
help/full display
```

A list of the topics available in the library is available by typing:

```
help/full/topics
```

Program Library

Sequences of Semper commands may be put together and inserted into a program library from which they can be called and executed.

To make a library accessible to Semper it has to be assigned in the usual way. Type:

```
assign program name 'semper.plb'
```

The programs in the library can be listed with the command:

```
show programs
```

A program is run by typing the **library** command and the name of the program.

Try typing:

```
library copyright
```

Programs are normally written using a text editor running externally to Semper.

A very simple program might be created in a file named *prog.spl* as follows:

```
hello()
type 'hello world'
end
```

It is inserted into the library with the command:

```
add program 'prog.spl'
```

Semper Training

Worksheet 2: Look-up Tables

Introduction

In the first worksheet the need to create and load a look-up table in the Semper working environment was mentioned. This requirement is best understood by making reference to the hardware of a typical monochrome framestore system which is shown in Figure 1 below.

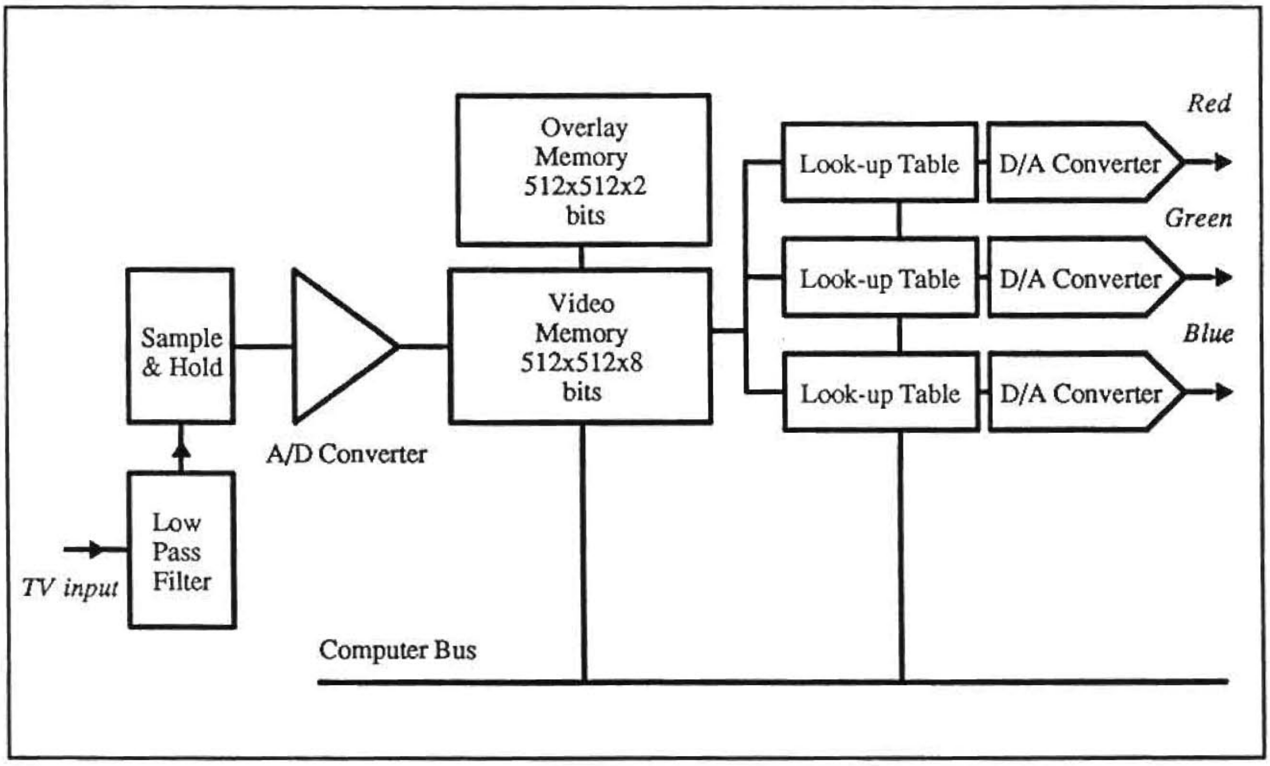


Figure 1. Block Diagram of a Simple Framestore

When a display picture is scaled and loaded into the framestore memory, each pixel has a value between 0 and 255 in the framestore memory. The intensity value is used to control the brightness of a red, green and blue gun in a T.V. display. A pixel value of 0 gives black, 255 gives full brightness. Between the image memory and each of the analogue to digital converters that generate the display signal, there is a look-up table. This is a table of 255 values that can be used to define any arbitrary relationship between the value of a pixel stored in memory and the displayed brightness in each of the three colour channels.

The simplest relationship that can exist is simply a one to one relationship which can be represented by a straight line graph. If the relationship between memory value and displayed intensity is the same in each colour channel then the look-up table is known as a monochrome look-up table. This is represented diagrammatically in Figure 2.

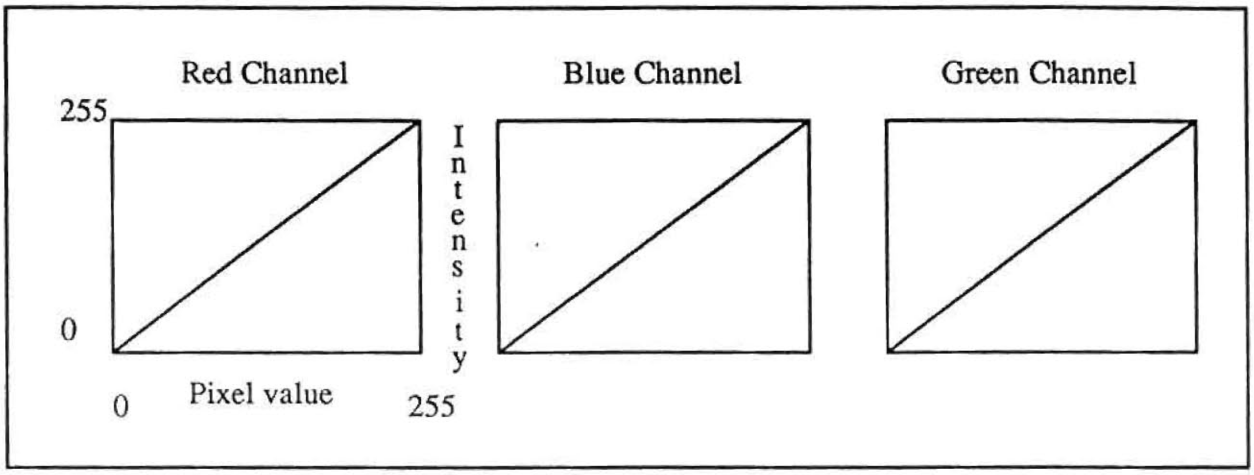


Figure 2. Monochrome Look-up Table

The look-up tables are connected to the computer bus so that they can be loaded and changed by the host computer.

There is another class of look-up table known as a false colour table. Here each different grey level in the source image can be assigned a different colour in the display. This is achieved by putting differing tables in each of the three colour channels. This situation is represented diagrammatically in Figure 3.

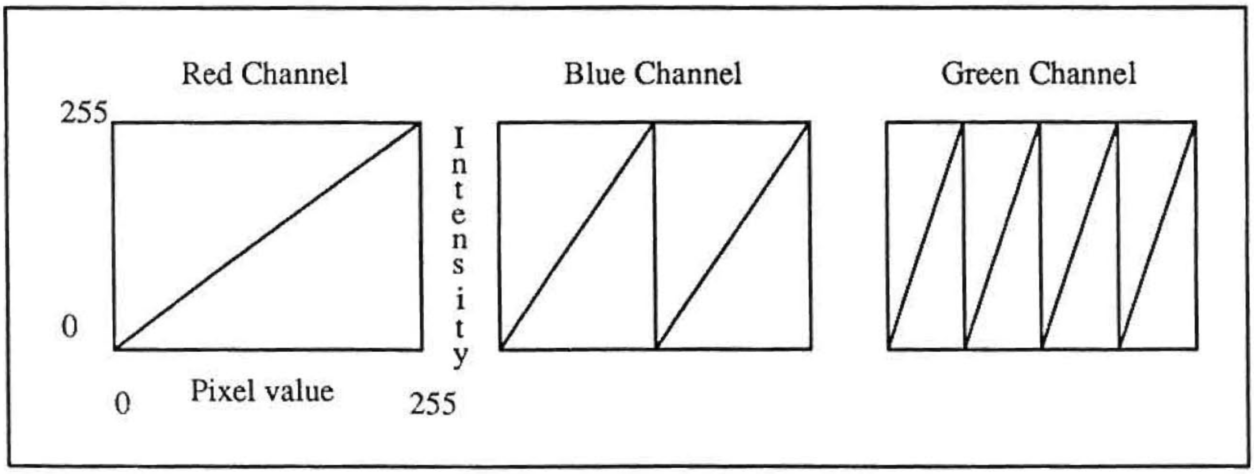


Figure 3. False Colour Look-up Table

Semper supports a third class of look-up table known as a full colour look-up table. This is used on full colour display hardware where there is a separate memory plane for each colour.

When a Semper picture is displayed, the intensity of any particular pixel on the screen is a product of the following:

*Display intensity=original pixel value * Semper display scaling * look-up table value*

Unless the **preset** option is used the Semper **display** command scales the picture pixel values loaded into the framestore memory between the values 0 and 255.

Starting the Semper Session

To start the Semper session, follow the instructions below that are relevant to your machine:

PCs

Insert the disk labeled *"Training Worksheet 2"* into the A drive.
Start Semper by typing the command:

```
semper /run=a:lookup
```

Workstations

Start Semper by typing the command:

```
semper /run=lookup
```

Look-up Tables in Semper

In the Semper environment there are a number of commands to create and manipulate look-up tables. Creating and loading look-up tables are separate operations. Semper can have up to ten look-up tables defined internally at any one time but only one can be loaded and active. An unlimited number of look-up tables can be stored on picture disks as class *Lut* pictures.

We first create a monochrome look-up table as table number 1.

```
lut 1 monochrome create
```

Then we load the framestore with this particular look-up table using the following command:

```
view lut 1 frame
```

This makes lut 1 the current look-up table. Subsequent commands will refer to it unless a new **view** command is issued. We are now in a position to display an image.

```
display 3:1 to display:1
```

Simple Look-up Table Applications

We can now demonstrate some of the uses of look-up tables. One of the simplest operations that you can do is to invert the contrast of an image. We display image 2, which is a photographic negative. The negative can be converted to a positive by a look-up table that displays values in the framestore digitised as black as peak white.

Try typing the following sequence of commands:

```
display 3:2 to display:2
lut 2 monochrome invert create
view lut 2
```

Another operation that has a direct analogy with photography is the manipulation of the brightness and contrast of an image. Try the commands:

```
lut contrast .5 reset
```

or

```
lut brightness .2 reset
```

or

```
lut range 23,120 contrast 0.9 brightness 0.5 reset
```

These commands create a look-up table that is represented below in Figure 4.

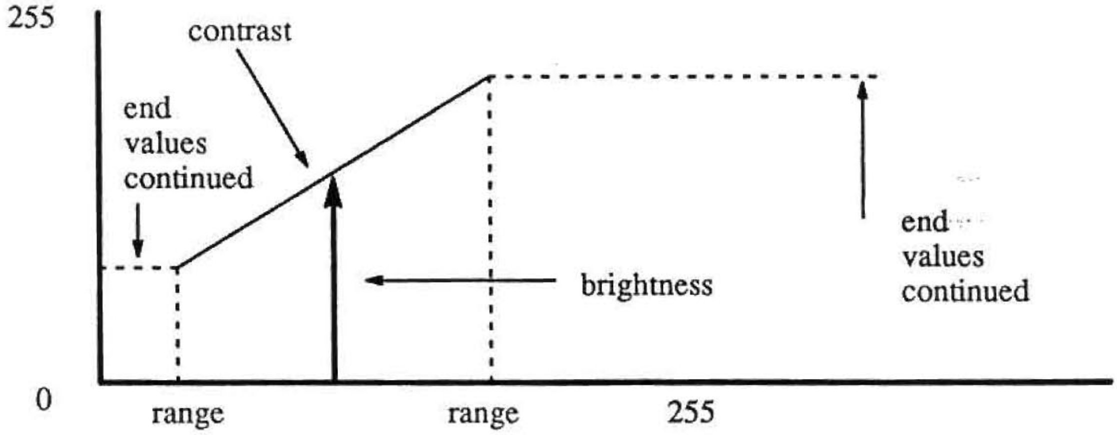


Figure 4. Contrast Adjusting Look-up Table

The **range** key specifies the minimum and maximum values over which the ramp extends. Reducing the range increases the contrast. The **contrast** key specifies the slope of the ramp as a fraction of the maximum possible for the range and brightness values.The **brightness** key gives the output value at the centre of the ramp.

You can examine the look-up table that you have created by storing it on a scratch picture disk. Type the following:

```
lut 2 to 4:1
examine 4:1
```

You will see that it is reported as a 256x1x1 picture of class *Lut*. The *Lut* class is the second of the ten picture classes that Semper supports. It is monochrome and thus has one layer. Its size is equal to the number of grey levels that the hardware supports. On workstations the number of look-up table entries may be less than 256.

You can also display graphically the look-up table that you have created. Try the commands:

```
max=255;min=0
display 4:1 to display:1 preset
```

You can see which look-up tables are defined and which one is current with the commands:

```
show lut
type clut
```

False Colour Look-up Tables

Semper supports another type of look-up table that enables you to assign different colours to each grey level in the image. The simplest option is to use a standard false colour look-up table defined within Semper. You can best examine its effect by displaying a grey scale ramp in one of the partitions. Try typing the following commands:

```
ramp partition 10 times 1

lut 3 false create

view lut 3 frame
```

The false colour look-up table is useful if you want to bring out detail that may not otherwise be visible. A false colour look-up table has three layers, one for each of the red, green and blue guns in the display. You can examine it using the following commands.

```
lut to 4:2
examine 4:2
```

You can specify how the grey level in the image maps to the colours displayed on the screen. Before giving details of this it is worth mentioning the way in which Semper describes the colour produced by a look-up table.

Any particular colour can be specified by three numbers known as the *hue, saturation* and *intensity*.

Intensity is measured on a scale between 0 and 1. This is the brightness of the colour. 0 is black and 1 is full intensity.

The *hue* values specify the colours that form a spectrum through the range 0,360. This is illustrated in Figure 5 below.

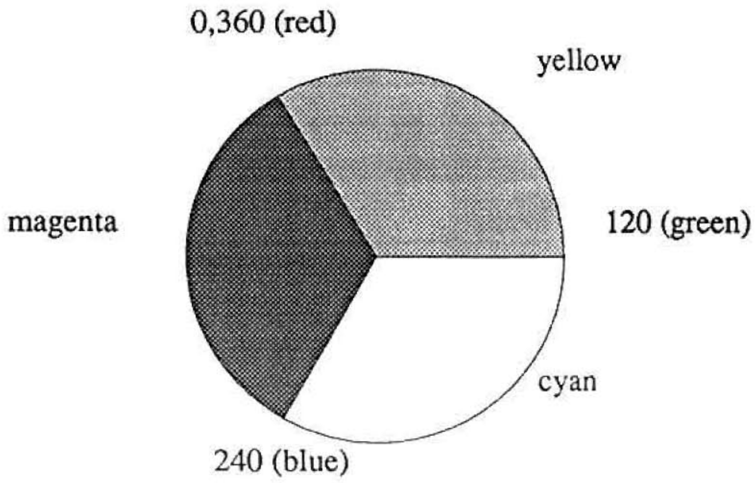


Figure 5. Colour Spectrum

The *saturation* is a measure of the degree to which the colour is diluted with white. This is a number in the range 0,1. 0 is white, 1 represents the pure colour.

To illustrate these ideas first create a false colour look-up table, reset it to monochrome mode and then load the table. Next display a picture:

```
lut 4 false create reset
view lut 4 frame
display 3:3 to display:2
```

Let us change the colour of the image using a look-up table. We use the **lset** command to do this. In this example the brightness will vary over its full range. The saturation will be constant and the colour will be fixed.

```
lset brightness 0,1 saturation 1 hue 100
```

Try experimenting with differing values of hue. A more pastel shade may be obtained by altering the hue:

```
lut reset
lset brightness 0,1 saturation 0.5 hue 100
```

Another effect can be obtained by causing the grey level values in the image to vary the hue. Brightness and saturation are constant.

```
lut reset
lset brightness 1 saturation 1 hue 0,360
```

***

Problem

Display image 3:5. This image has been created by superimposing two stereo views of an object. Alternate pixels are from one image then the other. One image has pixel intensity values in the range 0,127, the other has values in the range 128,255. Construct a false colour look-up table that will display this image as a red/green stereo pair that can be viewed with coloured glasses.

Hint: Consider the use of the **lset** command with the **range** key.

Note: On some workstation displays (VAX, SUN) there are less than 256 values in the display look-up table. If you are working on such a machine, the **scaled** key must be used with the look-up table commands for this problem.

***

Interactive Operation

There is another feature of Semper look-up table handling which is very useful in certain situations. You can vary the current look-up table interactively using the mouse. Movement in an up and down direction can control one parameter, movement left to right can control another.

One important application for this technique is the thresholding of an image prior to image analysis. A highlighted look-up table displays a set range of grey levels to full brightness on the display. This is represented in Figure 6.

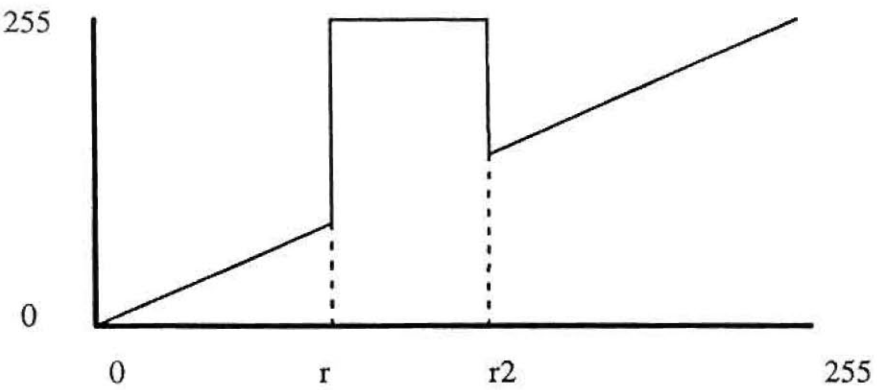


Figure 6. Highlighted Look-up Table

Image analysis requires you to define the outline of the object of interest by specifying a threshold. This is a process that is best carried out interactively.

In this situation we can make motion in one direction to control the upper range of highlighting and motion in the other to control the lower range. We first display an image in partition 1 and a greyscale ramp in partition 2

```
display 3:4 to display:1
ramp partition 10 times 1
```

Then we create a false colour look-up table and switch on the adjustment. The hue, saturation and intensity values of the highlighted region are specified using the **hsv** key.

```
lut 4 false create reset
view lut 4 frame
ladjust upper lower initially hsv 120,1,1
View...Slice (Windows)
```

When you have finished highlighting, press the left mouse key. The Semper variables *r* and *r2* are set to the highlighted range. You can inspect the final highlight values by typing :

```
type r,r2
```

***

Problem

Display image 3:6. Unlike in the previous example, its range is 0,100 and thus it has to be scaled by the **display** command. Try to find the range of pixel values that defines the blood cells using interactive thresholding with the mouse. Refer these values to the original image rather than to pixel values in the display. You will have to refer to the manual entry for **ladjust** to find out how to do this.

Hint: Consider the **scale** option. Confirm your result by listing it on the screen.

***

Saving and Restoring Look-up Tables From Disk

You can save and restore the current look-up table to and from disk. Try typing:

```
lut to 4:5
examine 4:5
```

You will see that picture 5 is a false colour look-up table which has the dimensions 256x3x1. Full colour look-up tables have the dimensions 256x1x3 and monochrome look-up tables have the dimensions 256x1x1. Not all framestores support 256 display levels, so in some cases depending on the hardware that you are using, you may see look-up tables of a smaller size.

To restore a look-up table from disk type:

```
lut from 4:5
```

***

Problem

Consider a situation that often arises when a camera is connected to a framestore. The framestore expects the signal from the camera to be within a specified range. If the signal is too large the system saturates and information is lost. If the signal is too small the full 256 grey level count is not used and again information is lost.

The amplitude of the signal will depend on the iris setting on the camera and on the illumination. It is often not clear from the image on the screen whether the iris and illumination are set up correctly. Consider how you might construct and load a look-up table that would aid the adjustment of the camera.

To test your example use pictures 3:7 and 3:8. Image 3:7 is taken with the camera set up correctly and 3:8 is taken with an incorrect camera setup. Display these images using the **noscale** option to simulate incoming signals from the camera.

Hint: Consider the use of the **lset** command. In particular consider the **range** key.

***

Semper Training

Worksheet No 3. Image Coordinates & The Crosswires Cursor

Introduction

Three sets of interrelated coordinate systems exist in the Semper environment. They are *picture* coordinates, *partition* coordinates and *frame* coordinates. Operations such as defining points or regions of interest within an image are defined with respect to one of these coordinate systems.

The picture coordinate system is represented in Figure 1 below.

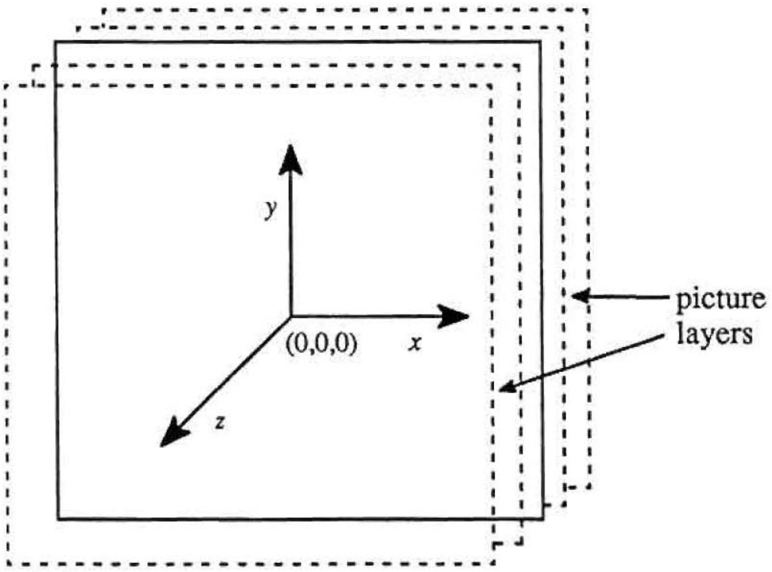


Figure 1. Picture Coordinates

The usual origin is at the centre of the picture with *x* increasing to the right and *y* increasing upwards. Semper supports multi-layer images. In these images a *z* direction is defined with the middle layer as the origin.

The frame is a region defined by the display hardware where pictures can be displayed. A partition is a user defined subregion on the display of arbitrary size and position. The relationship between the display frame, partitions and pictures is represented in Figure 2 overleaf.

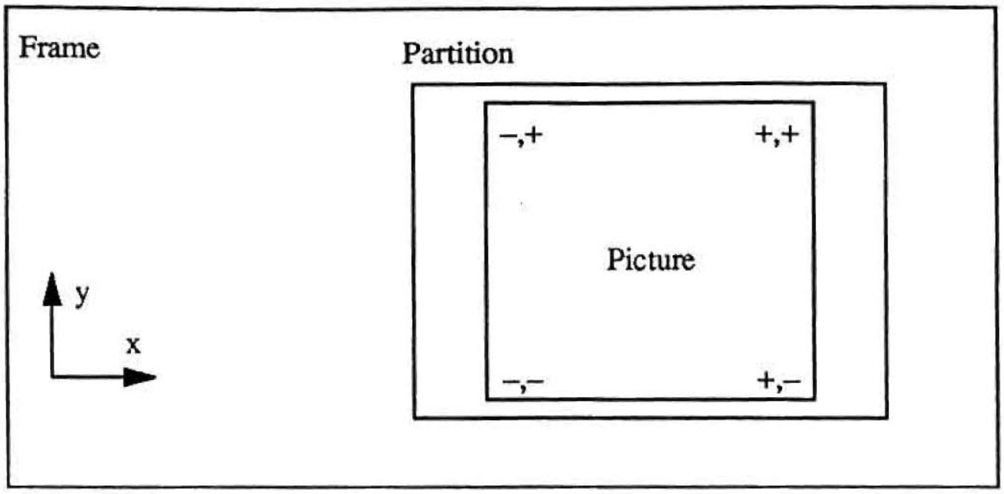


Figure 2. The Relationship between Frame, Partition and Picture

Starting The Semper Session

To start the Semper session, follow the instructions below, that are relevant for your machine:

PCs

Insert the disk labeled "*Training Worksheet 3*" into the A drive.
Start Semper by typing:

```
semper /run=a:xwires
```

Workstations

Start Semper by typing:

```
semper /run=xwires
```

This starts up Semper with the environment required for this worksheet.

The Picture Origin

Display the first picture on worksheet picture disk 3.

```
display 3:1 to display:2
```

The position of the picture origin can be inspected with the command:

```
examine 3:1 full
```

The origin may be set at any arbitrary position or reset to the centre. Try typing the following commands:

```
copy 3:1 to 4:1
origin 4:1 top left
examine 4:1 full
```

This operation cannot be performed on picture 3:1 as the picture disk has been assigned as a write protected disk. Reset the origin before continuing with the command:

```
origin reset
```

The Xwires Cursor

The basic operation of the crosswires cursor is to report the coordinates of a point. To illustrate this, we display an image in partition 2 and then instruct the cursor to appear in this partition:

```
display 3:1 to display:2
xwires display:2
```

The cursor now appears in partition 2. Try moving it with the mouse. When you have finished press the mouse button. The point at which you pressed the mouse is marked on the image. The Semper variables *x* and *y* are set to the position of the point. By default the positions are reported with respect to the picture coordinate system. Having obtained this information you could use it to perform some operation such as print out the *x*, *y* position and all the pixel values in the neighbourhood of that point. For example:

```
type x,y
print 3:1 position x,y
```

The default coordinate system used for position reporting is the picture coordinate system. However *x* and *y* can be reported in the frame or the partition coordinate system. Try identifying a point on the image and reporting it in the different coordinate systems.

```
mark frame border
xwires frame
type x,y
```

and

```
mark partition 2 border
xwires partition 2
type x,y
```

Another frequent operation is defining a region of interest in the image. The can be a one dimensional line, a rectangle or a region of an arbitrary shape.

One common requirement is to extract a line section from an image. This can be done as follows, using crosswires to define the line. When the **xwires section** command is executed the Semper variables *r, r2, x, y, theta* are set. These define the coordinates and angle of the line. This information is then used by the **extract** command. The **@section** notation is a macro that expands to:

```
position x,y size r,r2 angle theta
```

Try typing:

```
display 3:1 to display:1
xwires section
extract 3:1 to 4:1 @section
display 4:1 to display:1
```

When the one dimensional image is displayed, it appears as a graph.

Another similar operation is the definition of a region of interest. Here you can use the mouse to define the two corners of a rectangle. Try the following sequence of commands:

```
display 3:1 to display:1
xwires region
extract 3:1 to 4:2 @region
display 4:2 to display:1
```

Here we have extracted the outlined area of interest.

In addition to outlining a rectangular region you can define an arbitrary shaped closed region with the **xwires** command. Two clicks of the mouse button causes the region to snap closed.

Picture 10 generated by this process is a points list class picture (*Plist*) that contains a list of the *x, y* coordinates that define the region. You can then use this information in conjunction with other commands. Try:

```
display 3:1 to display:1
xwires closed curve to 4:10
mask display:1 with 4:10 outside value 0
```

This sequence of commands should mask the region outside the area of interest.

You can do freehand sketches using the mouse and store the resulting points list. Try the command: (Note: sketch command not currently available in Windows version)

```
sketch frame to 4:20
```

Pressing the left hand mouse key starts the process. The amount of data collected can be reduced using the **tolerance** key. Try the command:

```
sketch frame to 4:30 tolerance 5
mark frame with 4:30
```

***

Problem 1

Images 3:2 and 3:3 on the picture disk show a camera resolution test chart taken using two different cameras. How would you get an objective comparison of the resolution performance of each?

Hint: Use the **xwires** command and **extract**.

***

Problem 2

Display image 3:1. This image has a number of features that run diagonally across the image. Try to define a region of interest that spans the features and produce a new smaller extracted image where they run horizontally across the image. You may need to refer to the *Semper 6 Command Reference* manual entry on **xwires** to solve this problem.

***

Semper Training

Worksheet 4: Image Storage Forms & The Image Calculator

Introduction

A very useful feature of Semper is its '*image calculator*' facility. This enables the creation of synthetic images where pixel values are an arbitrary function of position. Also more or less arbitrary mathematical combinations/functions of pictures may be produced.

Use of the image calculator facility demonstrates the need for differing image storage forms. When an image is acquired in a framestore each pixel is digitised to one of 256 levels. Each pixel requires one byte of digital storage. A byte is a basic unit of computer storage and can be represented by 8 digital. This is represented in Figure 1.

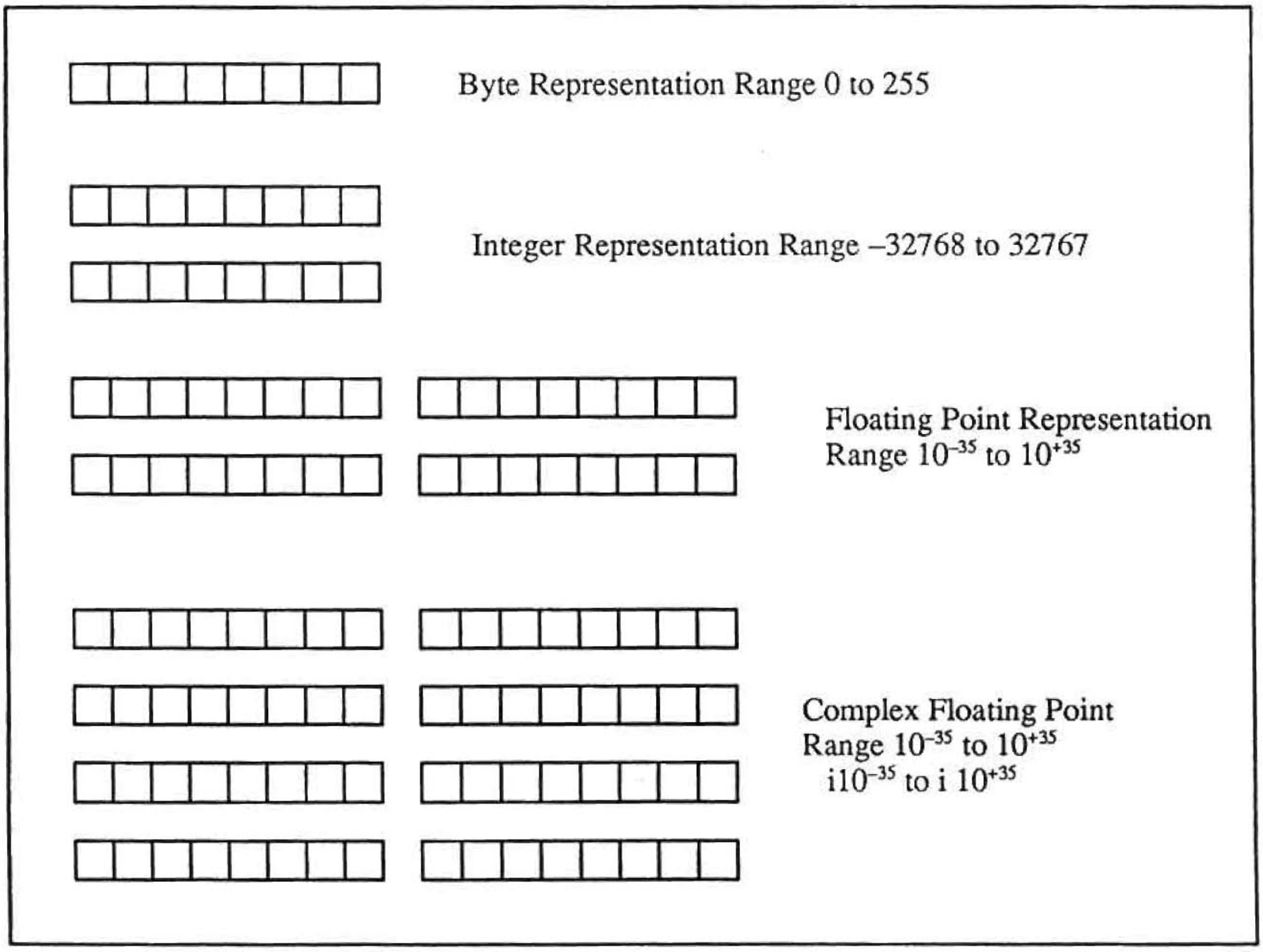


Figure 1. Computer Storage Representation

All the images we have encountered so far have been *Byte* images. There are circumstances where more than 256 levels need to be accommodated. This is the case, for example, when images are digitised to 1024 or more levels by densitometer equipment. In this case *Integer* storage is used. This can represent intensity levels between –32768 and +32767. Two bytes of storage are needed for each pixel.

There are certain situations where even greater intensity resolution is needed. In certain types of image such as *Power Spectra* very bright features are present together with a much lower level information. To represent such images that have a large dynamic range *Floating Point* representation is used. Four bytes of storage per pixel are used for such images.

There is an additional storage class in Semper known as *Complex Floating Point* storage. This is required to represent the real and imaginary parts of the complex results produced by operations such as fourier transforms. The properties of the different storage classes are summarized in the previous diagram, Figure 1.

Starting the Semper Session

To start the Semper session, follow the instructions below that are relevant to your machine:

PCs

Insert the disk labelled "*Training Worksheet 4* " into the A drive.
Start Semper by typing the command:

```
semper /run=a:calc
```

Workstations

Start Semper by typing the command:

```
semper /run=calc
```

These commands start up the Semper session with the environment required for this worksheet.

The Image Calculator

Using the image calculator you can add, subtract, divide or combine images in quite arbitrary ways.

The **calculate** command may be used to add images together. Image 3:6 and 3:7 are different S.E.M. pictures of the same object obtained in low beam current conditions and are thus very noisy. Adding them together should improve the signal to noise ratio of the image. This can be done simply as follows:

```
cd=3
display 3:6 to display:2
display 3:7 to display:3
```

```
calculate :6+:7 to 4:1
display 4:1 to display:4
```

The result of the addition is put into a *Byte* class picture and the results are clearly wrong. This is because the operation has generated numbers that are larger than the range (0 to 255) that a *Byte* class image can represent. Try the command again with the result sent to an *Integer* class picture:

```
calculate :6+:7 to 4:1 integer
display 4:1 to display:1
```

The *Integer* picture is able to accommodate the larger range of pixel values.

Another application of **calculate** , that uses image division, is shading correction prior to image analysis. Here the source image from a microscope is divided by a reference image which is the blank field image derived from the microscope when the sample has been removed. Here image 3:1 is the source and 3:2 is the shading image. The image division is carried out as follows. First make device 3 the current device:

```
erase frame
cd=3
display 3:1 to display:2
display 3:2 to display:3
calculate :1/:2 to 4:1
display 4:1 to display:4
```

The result of this operation is clearly not correct. This is a situation where the floating point representation should be used as the results of the division are in the range 0,1.04. Try repeating the command using a *Floating Point* class picture to store the result:

```
calculate :1/:2 to 4:1 fp
display 4:1 to display:5
```

Viewing the images with a highlighting lookup table reveals the effect of the shading correction. Try the commands:

```
lut 10 from 3:8
view lut 10
```

Reset the lookup table when you have finished with the command:

```
lut reset
```

One of the less obvious operators that can be used with the **calculate** command is the **ifelse** operator. This can be used, for example, to merge two separate pictures with each pixel being selected from one picture or another according to the result of some logical operation. Display the following three images:

```
erase frame
display 3 to display:2
```

```
display 4 to display:3
display 5 to display:4
```

The first two images are those to be merged. Image 5 is a mask that specifies the regions of the two images that are to be merged.

Try the following command:

```
calculate ifelse(:5=1,:3,:4) to 4:2
display 4:2 to display:5
```

Synthetic Images

Another feature of the image calculator is its ability to generate synthetic images, where the pixel values are an arbitrary function of position. For example try:

```
cd=4
create 1 size 256 fp
calculate sin(pi*x/8)
display 4:1 to display:5
```

Note that the initial blank image is *Floating Point*. This is because the sine function generates pixel values between +1 and –1. *Byte* and *Integer* class pictures can only take discreet values and cannot represent the fractional data generated by this function.

The logical operators may be used by the calculator. Try the commands:

```
create 2 size 256 byte
calculate sin(root(x*x/8+y*y/8)) > 0
display 4:2 to display:5
```

This operation generates a binary thresholded image.

The following functions are recognised by Semper:

sin(x)	sine (x in radians)
asin(x)	arcsin (result in radians), for mod(x) <= 1
cos(x)	cosine (x in radians)
acos(x)	arccosine (result in radians), for mod(x) <= 1
tan(x)	tangent (x in radians)
phase(x)	arctan (result in radians)
phase(x,y)	arg(x+iy) (result in radians), for mod(x) <= 1
exp(x)	exponent (e to the power of x)
ln(x)	ln (natural or Naperian log), for x>0
min(x,y)	lower of values x,y
max(x,y)	higher of values x,y
mod(x)	modulus (absolute value)
mod(x,y)	modulus(x+iy), mod(x,y) = root(x*x + y*y)

msq(..)	= mod(..) squared
root(x)	square root, with x>= 0
rem(x,y)	remainder when x is divided by y, rem(x,y) = x – fix(x/y)*y
round(x)	nearest integer to x
fix(x)	next integer towards zero from x
p(x,y,z)	value of pixel x,y,z of current picture
re(x,y,z)	real part of pixel x,y,z, same as *p(x,y,z)*
im(x,y,z)	imaginary part of pixel
and(n,m)	bitwise **and** of round(x) and round(y)
or(n,m)	bitwise **or** of round(x) and round(y)
not(n)	bitwise **not** of round(x)
rad(x)	angle in radians equivalent to x degrees
deg(x)	angle in degrees equivalent to x radians
ifelse(x,y,z)	returns y if x is non-zero and z if x is zero
set(*variable name*)	returns 1 if named variable is set; otherwise 0

***

Problem 1

You wish to assess the performance of number of cameras from different manufacturers. Try to produce images of black and white line pairs which, when printed on a laser printer, could be used to assess resolution.

Problem 2

Another important consideration when assessing a camera is the possibility of geometrical distortion. Try to produce a series of evenly spaced dots that could be used as a geometry distortion test image.

Problem 3

Image 4:1 is in *Floating Point* representation which takes up a lot of disk space and takes a long time to display. Consider how you might convert this to a *Byte* form image to save disk space.

Hint: Look at the **scale** command in the *Semper 6 Command Reference* manual.

Problem 4

Certain types of cathode ray tube displays have a characteristic where the display intensity is proportional to the square of the drive voltage applied to the tube. Create a *Lookup Table* class picture and use the **calculate** command to create a correcting lookup table that makes displayed intensity linearly proportional to pixel value. Load and apply this lookup table. You may have to refer back to worksheet 2 for details of lookup tables. This process is known as gamma correction.

***

Semper Training

Worksheet No 5. Histograms and Histogram Equalisation

Introduction.

An image histogram is a plot of the distribution of grey levels in an image. It consists of a number of '*bins*', each corresponding to a grey level value or range of levels. Each bin contains the number of pixels in the particular range of grey levels. A histogram gives you considerable information on the subjective qualities of an image. A "good" image tends to have a uniform spread of grey levels. A "poor" image tends to have its grey levels concentrated within a narrow range. Various methods are available to improve images in this situation.

Starting the Semper Session

To start the Semper session, follow the instructions given below, that are relevant for your machine:

PCs

Insert the disk labelled "*Training Worksheet 5* " into the A drive.
Start Semper by typing:

```
semper /run=a:histogra
```

Workstations

Start Semper by typing:

```
semper /run=histogra
```

This starts up the Semper session with the environment required for this worksheet.

Image Histograms

To illustrate the properties of image histograms there are several images on the picture disk. Picture 1 is called 'Suzy'. This image has a nearly ideal grey level histogram. As a first step, display image 1:

```
display 3:1 to display:1
```

To calculate the histogram type:

```
histogram 3:1 to 4:1
examine 4:1
```

You will see that a *Histogram* class image has been produced. Display this with the command:

```
display 4:1 to display:2
```

In contrast, image 2 has the majority of its grey levels at the bottom end of the range, giving a "dark" appearance. Type the following commands:

```
display 3:2 to display:3
histogram 3:2 to 4:2
display 4:2 to display:4
```

The **histogram** command takes certain default values for the number of channels and the grey values associated with the end values.

The end values default to *min* and *max* , the range of pixel values seen when the image is displayed. You override this and set the end values and the number of channels. Try the following:

```
max=255;min=0
histogram 3:2 to 4:3 channels 100 preset
display 4:3 to display:4 times 3 replicate
```

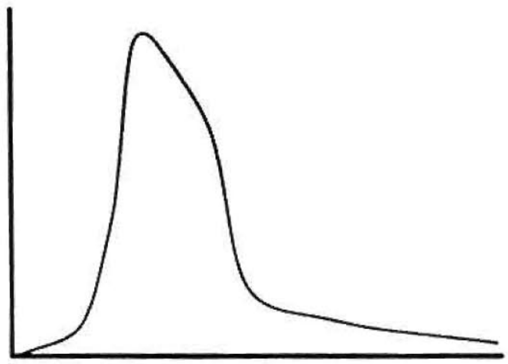
By default, a one hundred channel histogram will be displayed 100 pixels wide. It needs to be magnified when displayed. The **replicate** option prevents the display interpolating between the bins of the histogram.

Another feature of the **histogram** command is that you can calculate the histogram for a subregion of the picture, defined using a mouse and the **xwires** command. Try the following commands:

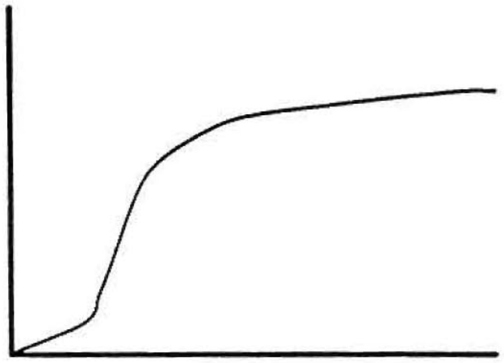
```
display 3:2 to display:3
xwires region
histogram 3:2 to 4:4 @region
display 4:4 to display:4
```

Histogram Equalisation

When the grey levels in the image are not uniformly distributed the image can appear subjectively light or dark. It is possible to stretch the contrast in the image by constructing a special lookup table which is the cumulative sum of the image histogram. This process is known as histogram equalisation. This is illustrated graphically in Figure 5-1 below.



Histogram



Equalising Lookup Table

Figure 5-1. Histogram Equalisation

Try the following commands.

```
display 3:2 to display:3
histogram 3:2 to 4:10
map 3:2 with 4:10 to 4:11
display 4:11 to display:4
```

Try taking the histogram of the enhanced image and comparing the two histograms of each image before and after enhancement.

The **map** command is used to store the result of the modification of an image by a lookup table. Normally the action of a lookup table only affects the way an image is displayed, not the way it is stored.

***

Problem

Image 3:3 shows a picture of a plant on a window ledge. The background illumination from the sky is very bright and hence not much detail of the plant can be seen. How would you improve the contrast of the region that contains the plant ?

Hint: Consider the use of the **xwires...region** , **histogram** and **map** commands.

***

Semper Training

Worksheet 6: Image Filters

Introduction

Semper contains a number of image filters. These modify the properties of an image in various ways. They are often applied prior to operations such as image analysis. They can be broadly classified into *high pass* , *low pass* and *median* filters.

- High pass filters can remove slowly varying components of an image, such as shading caused by uneven illumination.
- Low pass filters can be used to remove unwanted fine detail and noise.
- Median filters can remove isolated noise spikes and interference.

Starting the Semper Session

To start the Semper session, follow the instructions given below, that are relevant for your machine:

PCs

Insert the disk labelled "*Training Worksheet 6* " into the A drive.
Start Semper by typing:

```
semper /run=a:filter
```

Workstations

Start Semper by typing:

```
semper /run=filter
```

These commands start up the Semper session with the environment required for this worksheet.

Fir Filters

Many filters are implemented in Semper as *finite impulse response* filters (F.I.R). In this type of filter each pixel is replaced by a weighted sum of its neighbours. At the edge of the picture the edge pixel values are extended outwards. The values of these weights determine the properties of the filter. This set of weighting coefficients is known as the *kernel*. The size of the kernel can be specified. Generally larger kernels take longer to calculate but give a more pronounced effect. The user can use one of a standard set of filtering operations or can define his own.

It is possible to view directly the operation of a filter. Image 3:4 is a special test image that has all its pixels set to zero, except the centre which is set to 1. Applying any of the F.I.R filters to this image will reproduce the kernel in the resulting image.

Examine the pixel values of the test image with the command:

```
print 3:4
```

Now try a blurring filter and then view the result by typing the commands:

```
fir 3:4 gaussian to 4:1
print 4:1
```

High pass filtering is also implemented using a convolution kernel. Try the following commands:

```
hp 3:4 over 7 to 4:2
print 4:2
```

Look at the effect of a low pass filter:

```
lmean 3:4 to 4:3
print 4:3
```

High pass filters

Display image 1 on the disk by typing:

```
display 3:1 to display:2
```

Image 1 shows a typical situation where you might wish to apply a high pass filter. In image analysis features are identified on the basis of their grey level. In this image the features of interest are partially masked by a variation in shading across the image. High pass filtering can remove the shading but retain the features of interest. Try the following commands:

```
hp 3:1 to 4:1 over 20 integer
display 4:1 to display:3
```

Note that we specified that the result of the filtering should be sent to an *Integer* class picture even though the source picture is *Byte* . This is an important point to note when using high pass filters. Some filtering operations generate results that are outside the 0,255 range that can be supported by a *Byte* class picture.

Applying a highlighting lookup table illustrates the effect of filtering on the two images. This can be shown by the following commands:

```
lut false create reset
lset range 140,255 brightness 1 saturation 1 hue 0
```

when you have finished the comparison, reset the lookup table with the command :

```
lut reset
```

The high pass filter works by converting a given pixel to the weighted sum of its neighbours. You can specify the size of the neighbourhood used. As a rule of thumb, the size of the neigbourhood in pixels should be in the order of size of the features of interest in the image. This type of filter is known as a *finite impulse response* filter (F.I.R.)

Low pass filter

Display image 2 from the disk:

```
erase frame
display 3:2 to display:2
```

This is an example of an image where you would wish to apply low pass filtering prior to analysis. In a fringe pattern we are interested in the position of the fringe but not the fine detail in the image. There are a number of low pass filter operations in Semper. The local mean is the complement of the high pass filter used earlier.

Try the following commands:

```
lmean 3:2 to 4:2
display 4:2 to display:3
```

Another implementation of the finite impulse response filter is the **fir** command.

Try:

```
fir 3:2 to 4:3 gaussian
display 4:3 to display:4
```

Another type of filter, which gives you considerable control over the degree of filtering that is applied, is the recursive filter. The parameter **a**, which can vary between –1 and +1 controls the filter. Try values near to one to give a very sharply smoothed picture:

```
rf 3:2 to 4:4 a .8
display 4:4 to display:5
```

Again it is instructive to apply a highlighting lookup table to these images to see how they would appear prior to image analysis:

```
lut false create reset
lset ran 150,255 brightness 1 hue 1 saturation 0.5
lut reset
```

Median Filters

Display image 3 on the disk:

```
erase frame
display 3:3 to display:2
```

Here we have an example of an image that has been corrupted by impulsive interference. This can be improved using the median filter. The median filter replaces a given pixel by the median value of the pixels in the immediate neigbourhood. Try applying the median filter to image 3:

```
rank 3:3 to  4:5
display 4:5 to display:3
```

We can see that the interference has been reduced without blurring the features in the image.

***

Problem

A new fir filter kernel has been devised as shown below. Try to apply this kernel to filter image 3:1.

Hint: Refer to the *Semper 6 Command Reference* manual entries for **fir**, **p**, the pixel set operator and **create** , the picture create operation.

–.25	–.5	–.25
–0.5	4	–0.5
–0.25	–0.5	–.25

***

Semper Training

Worksheet 7: Image Analysis 1

Introduction

The Semper image analysis facilities enable you to extract quantitative information about the objects in an image. This information can then be tabulated and printed out or it can be used to annotate the image.

Starting the Semper Session

To start the Semper session, follow the instructions given below, that are relevant for your machine:

PCs

Insert the disk labelled "*Training Worksheet 7* " into the A drive.
Start Semper by typing:

```
semper /run=a:analyse
```

Workstations

Start Semper by typing:

```
semper /run=analyse
```

These commands start up the Semper session with the environment required for this worksheet.

Image Analysis

Display image 1 from the disk, using the command:

```
display 3:1 to display:1
```

This image presents a typical image analysis situation. We will discover how to extract and tabulate information about the area and positions of particles in the image.

The image analysis process identifies objects on the basis of their grey level within the image. So the first stage in the analysis process is to identify the grey levels associated with the object. The method commonly used involves highlighting the grey levels between an upper and lower threshold to full intensity on the display.

A false colour highlighting lookup table can be used (as described in worksheet 2). The position of the highlight bands can be controlled interactively using the mouse. Vertical movement moves the upper band, horizontal movement moves the lower band. Try the following commands:

```
ramp partition 10 times 1
display 3:1 to display:1
lut false create reset
ladjust upper lower scale display:1 initial range 100,150
View...Slice
```
(Windows)

Now try moving the mouse. The ramp gives you a clearer idea of the operation of the highlighting process in operation. When you think that you have defined the objects in the image, press the left hand mouse button. Semper sets the variables *r* and *r2* to the highlighted values. The key **scaled display:1** means that the highlight values will be scaled correctly to the range of grey levels in the picture. The key **initial range 100,150** sets the initial values that are highlighted, before you start moving the mouse.

When suitable threshold values have been determined, the image analysis command can be invoked. In this example, the **area** key rejects particles of an area less the 400 pixels. Type:

```
analyse 3:1 segment 4:100 ge r le r2 area 400
examine all
```

The analysis procedure generates two sets of information:

- a list of all the particle parameters
- a segmented image, where the outline of each particle is stored.

In this case, the segmented image is stored in picture 100 and, by default, the particle parameter list is stored in picture 998. Note that the segmented image is only produced if you specify the **segment** key. The following table lists all the generated parameters and the variable names for each particle.

Name	Var	Description	Name	Var	Description
xref, yref	*xr, yr*	reference point	aferet, bferet	*af, bf*	feret diameters – 45° 135°
id	*pid*	particle identifier	hproj, vproj	*hp, vp*	horizontal and vertical projections
parent	*pa*	parent identifier			
holes	*h*	number of holes	perimeter	*p*	perimeter
background	*bg*	background flag	area	a	area
contact	*ec*	edge contact flag	xcen, ycen	*xc, yc*	centre of area
xmin, xmax	*x1, x2*	limits – min, max x	mmin, mmax	*m1, m2*	principal second moments of area – min, max
ymin, ymax	*y1, y2*	limits – min, max y			
hferet, vferet	*hf, vf*	horizontal and vertical projections	angle	*theta*	orientation
			circularity	*c*	circularity

The information in the particle parameter list can be used to annotate the image so that you can identify the information associated with each particle. Any of the particle parameters can be used to annotate the image with the centre of mass and the identity of each particle. Type the following commands:

```
pmark display:1 cm
pmark display:1 id
```

This marks the image with the identifier for each particle using the information in the particle parameter list. You can inspect the parameters associated with any particular particle. To examine the properties of particle 3, for example, type:

```
ptype all index 3
```

To examine the area and *x, y* positions of all the particles, type:

```
ptype area xcen ycen
```

The segmented image contains the outline of the particles and each particle is assigned an identity number. You can display this image with the following commands:

```
display 4:100 to display:1 noscale
pshow
```

Here the segmented image has been loaded into the display with the automatic grey level scaling switched off. The second command creates a special lookup table which allows the particles to be seen.

You can keep a record of the results produced by the analysis process by directing the results to a log file. Try typing the command:

```
assign new file name 'results'
```

The system will reply with a message, giving the number of the next free device which it assigns to the log file:

device 6 assigned

Now type the command:

```
echo console device 6
```

This command ensures that any results that you see on the screen will be sent to the log file *results.log*, where they can be edited later.

***

Problem 1

Display image 3:2 and analyse this image. Display the segmented image and annotate it with the particle identity and area.

Problem 2

Obtain a list of all the particle areas and positions and store them in an external file, sorted in order of ascending area.

***

Semper Training

Worksheet 8: Image Analysis 2

Introduction

Often the basic image analysis procedure, as described in worksheet 7, generates information on particles other than those of interest. Semper provides the means to filter this data according to a quite arbitrary criteria based on the image analysis parameters of the particles.

Starting the Semper Session

To start the Semper session, follow the instructions below, that are relevant for your machine.

PCs

Insert the disk labelled *"Training Worksheet 8 "* into the A drive.
Start Semper by typing:

```
semper /run=a:analyse2
```

Workstations

Start Semper by typing:

```
semper /run=analyse2
```

These commands start up the Semper session with the environment required for this worksheet.

Image Analysis

To demonstrate the parameter editing features of Semper, we will use the image analysis data that was produced in worksheet 6. To recall, an upper and lower threshold was defined using an interactive lookup table. These levels were then used with the **analyse** command to generate two sets of information, the segmented image and the particle parameter list in images 100 and 998 respectively. The parameters associated with each particle and the associated Semper variable name are tabulated overleaf.

Name	Var	Description	Name	Var	Description
xref, yref	*xr, yr*	reference point	aferet, bferet	*af, bf*	feret diameters – 45° 135°
id	*pid*	particle identifier	hproj, vproj	*hp, vp*	horizontal and vertical projections
parent	*pa*	parent identifier			
holes	*h*	number of holes	perimeter	*p*	perimeter
background	*bg*	background flag	area	*a*	area
contact	*ec*	edge contact flag	xcen, ycen	*xc, yc*	centre of area
xmin, xmax	*x1, x2*	limits – min, max x	mmin, mmax	*m1, m2*	principal second moments of area – min, max
ymin, ymax	*y1, y2*	limits – min, max y			
hferet, vferet	*hf, vf*	horizontal and vertical projections	angle	*theta*	orientation
			circularity	*c*	circularity

First display the segmented image with the commands:

```
display 3:100 to display:1 noscale
pshow
```

The automatic display scaling must be disabled with the **noscale** option when displaying segmented images.

The particle selection mechanism in Semper may be used with the **pshow** command to enable you to highlight particles according to any arbitrary criteria constructed from the image analysis parameters. This is useful as a check prior to editing. To take an example of this, one of the parameters that is calculated for each particle is the edge contact flag, called *ec*. This is set to 1 if the particle is touching the edge of the frame. To highlight all particles that do not touch the edge, that is, where the contact flag is zero, type the following command:

```
pshow if contact=0
```

Several conditions may be applied in conjunction with combinations of *and* and *or* operators. To highlight all particles that do not touch the edge of the frame and that have an area greater than 2000 try:

```
pshow if contact=0 & area > 2000
```

In addition to the basic highlighting operation each particle that is selected may be assigned a given colour or range of colours. The colours are specified in terms of hue and saturation values. For a single colour try:

```
pshow if contact=0 & area > 2000 saturation 1 hue 100
```

For a range of colours try:

```
pshow if contact=0 & area > 2000 saturation 1 hue 100,200
```

When you are satisfied with the particle selection criteria you can then use it to edit the particle parameter lists and segmented image.

Try the commands:

```
pedit 3:998 to 4:998 segment 3:100,4:100 if contact=0
```

To check the results, display image 101 and annotate it with the edited parameter list file. Note that you have to specify the particle parameter list explicitly as you no longer wish to use the default. Type:

```
lut reset
display 4:100 to display:1 noscale
pshow plist 4:998
```

Another important feature of the image analysis package is the ability to access the particle parameters in a Semper program. Thus, to take a simple example, you could sum all the areas of the particles of interest.

The Semper command **pset** is used to gain access to the particle parameters. Referring to the parameter table, using **pset** with a parameter name will set the appropriate variable name. The **pset** command also returns the total number of particles in the variable *n*. Try the following commands:

```
pset plist 4:998 count
total=0
for i=1,n;pset plist 4:998 area index i;total=total+a;loop
type 'Total Particle Area=',total
```

Here the area of each particle in the edited list is extracted and summed to produce the total area. Note that as we are using the edited particle parameter list, we have to specify it explicitly rather than use the default values that are initially produced by the image analysis procedure.

***

Problem 1

Display image 3:2 on the disk. Analyse the image and produce a table of the areas of all the particles that do not touch the edge of the picture and only have one hole. Confirm your result visually.

Note that when this image is thresholded it is not possible to separate all the particles. Consult the *Semper 6 Command Reference* manual and consider the use of the **pdraw** command, which enables you to modify the image by hand, to separate particles that touch.

Problem 2

Find the orientation of the angle of each of the particles of interest in image 3:1. Annotate each particle with a line at the same orientation going through its centre of mass.

Hint: Consider the use of the **pset** command in a **for** loop. You may have to refer to the manual entries on **pset** and **mark**.

***

Semper Training

Worksheet 9: Morphological Operations

Introduction

In many image analysis situations the process of thresholding does not adequately delineate adjacent particles because their grey levels are too similar. In these situations the morphological operations of erosion and dilation can sometimes be used to separate adjacent objects. In other situations, skeletonisation and other related operations can help to clarify relationships between objects as illustrated by the examples in this worksheet.

Starting the Semper Session

To start the Semper session, follow the instructions given below, that are relevant for your machine:

PCs

Insert the disk labelled "*Training Worksheet 9* " into the A drive.
Start Semper by typing:

```
semper /run=a:morpholo
```

Workstations

Start Semper by typing:

```
semper /run=morpholo
```

These commands start up the Semper session with the environment required for this worksheet.

Morphological Operations

Display image number 1 with the command:

```
display 3:1 to display:1
```

This shows an image of fungal spores in which certain components are separate but cannot be differentiated clearly on the basis of grey level information.

Image 3:2 is the segmented image produced as a result of grey level thresholding and image analysis. Use the following commands to view this image:

```
display 3:2 to display:1 noscale
pshow plist 5
```

The differing components can be separated by eroding a single pixel layer from around the image. This can be applied as many times as necessary until the component parts are separated. If the operation is carried out on the display it can be seen in operation. Try the following:

```
erode 3:2 to 4:1 times 3
display 4:1 to display:1 noscale
pshow plist 5
```

The separated parts can then be restored to something like their original size by dilating the image again. The separation of the objects, however, can be maintained by specifying the **separate** option. Try typing the commands:

```
dilate 3:2 to 4:1 times 3 separate
display 4:1 to display:1 noscale
pshow plist 5
```

The resulting image can then be subjected to further analysis to determine the particle sizes.

Another aspect of the morphology commands is demonstrated by image number 3:4. Before this can be displayed the special lookup table created by the **pshow** command has to be reset. Type the following commands:

```
lut reset
display 3:3 to display:1
```

This is an interference pattern produced by an interferometer. It is often required to produce a set of contour lines that lie on the centre of fringes. This can be achieved using the morphology commands. Display the segmented image that was produced when this image was analysed. Remember that the **noscale** option should be used when displaying segmented images.

```
display 3:4 to display:1 noscale
pshow plist 3:5
```

The skeletonisation operations reduce the objects in the image to single pixel wide threads. Type the commands:

```
erode 3:4 to 4:1 skeleton
display 4:1 to display:1
```

The skeletonise operation leaves some unwanted threads that are at right angles to the contour lines. Another morphology operation can be invoked to separate the unwanted strands, which can then be removed using the particle editing facilities. Try the commands:

```
erode 4:1 nodes
display 4:1 to display:1
```

This removes the pixels at the intersections of the skeletonised picture, thus breaking it into discrete particles.

At this point the eroded image is analysed again to exploit the particle editing facilities of Semper. The branches can be removed using a size criterion:

```
analyse display:1 to 4:2 segment 4:100 ge 0.5
display 4:100 noscale
pshow plist 4:2
```

Remove the small branches and display the segmented image with the command

```
pedit 4:2 to 4:3 segment 4:100,4:101 if area > 50
display 4:101 noscale
pshow plist 4:3
```

The resulting segmented image is then dilated to rejoin the areas that were separated by the **erode nodes** command, giving a clear set of contours along the fringe lines. Type the commands:

```
dilate 4:101 times 2
display 4:101 to display:1
erode 4:101 skeleton
display 4:101 to display:1
```

The result should show contour lines along the centre of the fringes.

***

Problem

Picture 3:6 represents the segmented image of some idealised sperm cells. Each consists of a head and a thin tail. How would you use the morphology and image analysis features of Semper to separate the head and tail and measure the length of the tail?

Hint: Consider the use of the **erode** command with the **ends** option and the use of the **calculate** command. You may have to refer to the *Semper 6 Command Reference* for details of these commands.

Semper Training

Worksheet 10: Image Resampling

Introduction

When an image is first acquired by a camera and framestore, it is sampled on a regular grid and the individual samples are stored in memory to become individual pixels. When the image is redisplayed on a monitor the regions between the individual samples are reconstructed from adjacent sample values. If you need to change the size, position, aspect ratio or orientation of an image, then you can use the resampling facilities of Semper.

Starting the Semper Session

To start the Semper session, follow the instructions given below, that are relevant for your machine:

PCs

Insert the disk labelled "*Training Worksheet 10*" into the A drive.
Start Semper by typing:

```
semper /run=a:sample
```

Workstations

Start Semper by typing:

```
semper /run=sample
```

These commands start up the Semper session with the environment required for this worksheet.

Image Resampling

The most common reason for resampling an image is to change its apparent size. This is represented diagramatically in Figure 1 overleaf.

To resample an image, try the following commands:

```
display 3:1 to display:1
extract 3:1 to display:1 sampling 2 size 128
extract 3:1 to display:1 sampling 0.5 size 512
```

These two uses of the **extract** command resample the original image on grids that are half and twice the size of the original spacing.

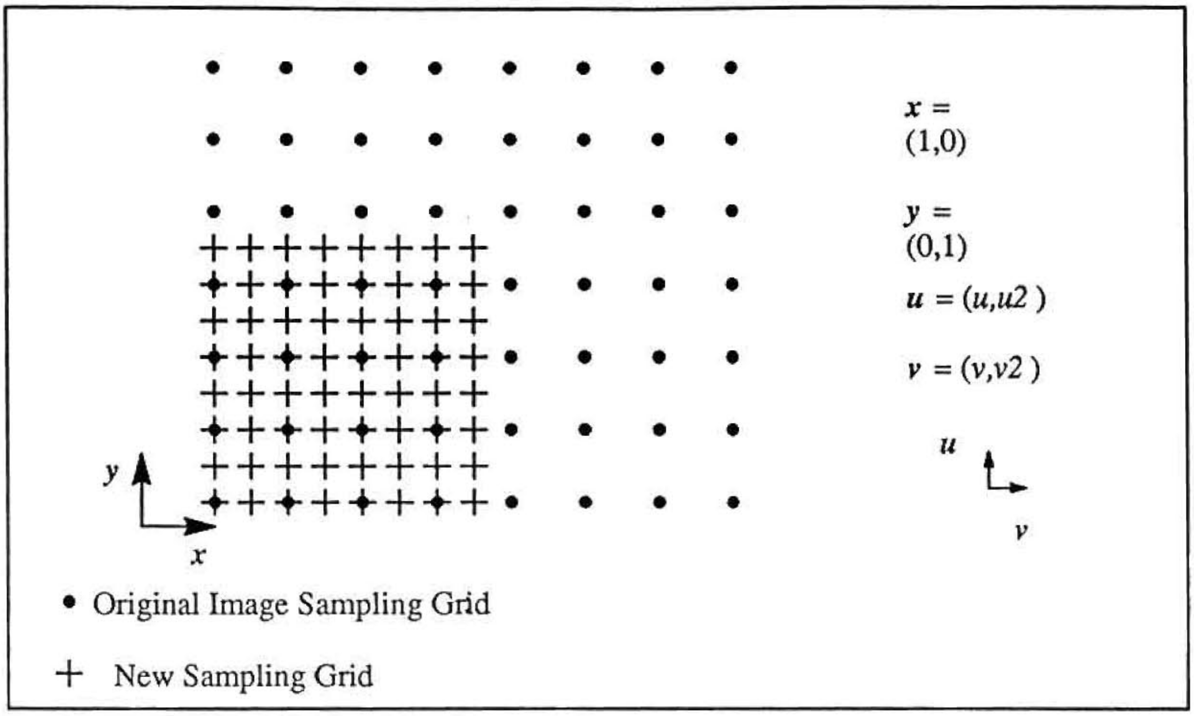


Figure 1. Image Magnification

Note that if you do not specify the output picture size you will see an output picture the same size as the input picture. Where the output picture has been reduced in size the result will be repeated periodically. Try the following command:

```
extract 3:1 to display:1 sampling 4
```

When an image is reduced in size, information is effectively thrown away. When the image is magnified no new information is created. The new sample values are calculated from adjacent values by a process known as *interpolation* .

The **extract** command can also be used to change the aspect ratio of an image. This is sometimes required, for example, if the image has been acquired by a framestore that has non square pixels. In order to specify a more general type of sampling grid two vectors *u* and *v* are introduced. These specify the new sampling grid produced by the **extract** command in terms of the original sampling grid specified by *x* and *y* . This is represented in Figure 2 overleaf.

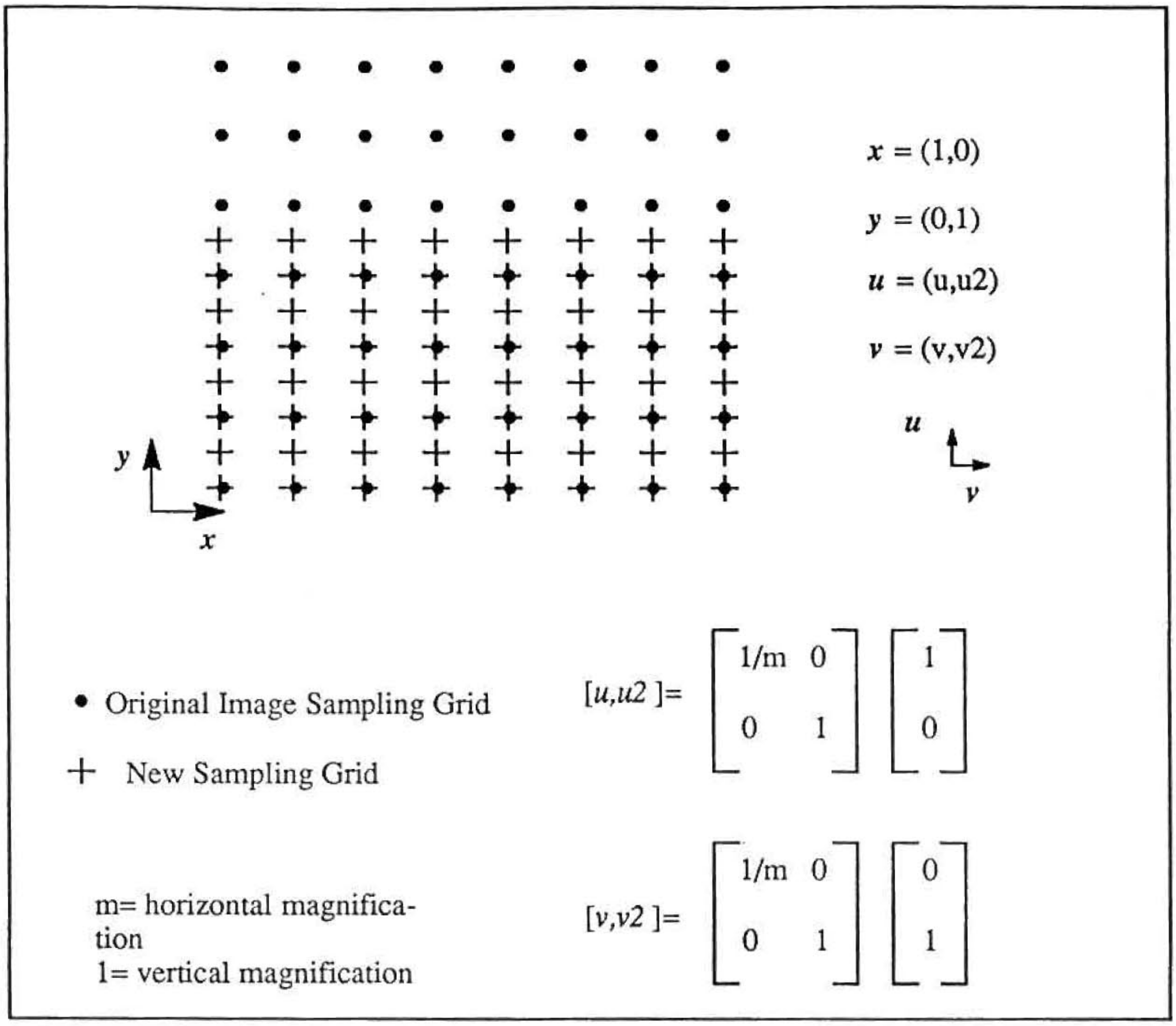


Figure 2. Aspect Ratio Change

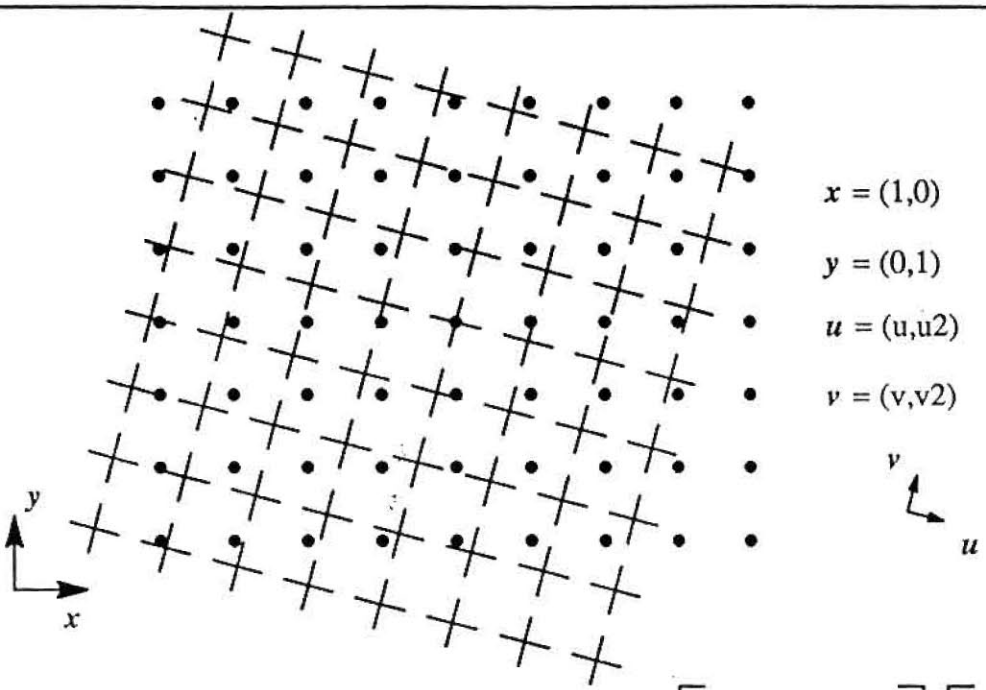
Consider the image 3:2.

```
erase frame
display 3:2 to display:2
```

This is a square image that has been acquired by a framestore with non square pixels and consequently it appears distorted when displayed on a square pixel display or printed out on a laser printer. It can be corrected by resampling on a non square grid which may be specified by the *u* and *v* vectors. Try the following commands:

```
u=0.75;u2=0
v=0;v2=1
extract 3:2 to display:3 uv
```

Another common use of the **extract** command is to perform small image rotations and shifts that are required when aligning images prior to other processing operations. This is represented in Figure 3 below.



• Original Image Sampling Grid

+ New Sampling Grid

θ=angle

$$[u, u2] = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$[v, v2] = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Figure 3. Image Rotation

Try the following commands:

```
theta=pi/8
u=cos(theta);u2=sin(theta)
v=-sin(theta);v2=cos(theta)
display 3:2 to display:2
extract 3:2 to display:3 uv
```

To perform image shifts, try the following command:

```
extract 3:2 to display:3 position 23,-34
```

***

Problem 1

Image 3:1 is of the size 256x256 pixels. Produce a version of the image of size 300 by 200 so that the image is stretched horizontally and compressed vertically.

Problem 2

Image 3:3 has been acquired from a linescan sensor in a satellite. One dimension of the image is produced from a linear CCD array. The second dimension of the image is produced by mechanically scanning the image over the sensor with a rotating mirror. It was found that the CCD array was misaligned by an angle of *pi/20*. The result is skew distortion of the image. Consider how you might use the **extract** command with the **uv** option to correct for the distortion in the image.

Hint: Try to express the relationship between points in physical space and image space as a matrix transformation. Find the correcting inverse transformation and use this to calculate the u and v vectors to use with the **extract** command.

If you do not wish to do this calculation, you can use the library program **invert** . If u and v are set up to describe the original sampling grid then the program will calculate a new u and v, which when used with **extract** , will correct the distortion.

***

Semper Training

Worksheet 11: Image Geometry Correction

Introduction

Many imaging devices based on raster scanning systems suffer from some form of geometrical distortion. Some of the mechanisms that can give rise to this are misalignment of scanning coils or barrel or pincushion distortion in the beam deflection system. This is particularly evident at low magnifications in scanning electron microscopes.

Other forms of distortion arise from the geometry of the imaging situation. For example it occurs in aerial photography, when the camera is viewing the scene obliquely.

The Semper command **warp** can be used to correct distortion in such a situation.

Starting the Semper Session

To start the Semper session, follow the instructions given below, that are relevant for your machine:

PCs

Insert the disk labelled *"Training Worksheet 11"* into the A drive.
Start Semper by typing:

```
semper /run=a:warp
```

Workstations

Start Semper by typing:

```
semper /run=warp
```

These commands start up the Semper session with the environment required for this worksheet.

Image Warping

The Semper **warp** command effectively resamples the image in a similar way to the **extract** command. The resampling grid is generated by a polynomial in x and y. The order of the polynomial and the type of interpolation used can be specified by the user.

The simplest first order type of distortion correction uses polynomials of the form:

$$u=a_0+a_1x+a_2y$$
$$v=b_0+b_1x+b_2y$$

Where u and v are the new sample points and x and y are the old sample points.

Six coefficients specify the distortion. Note that a first order warp is very similar to the **extract** command used with the **uv** option. First order warping can correct for aspect ratio errors and image skew and rotation.

More complicated distortions requires higher order polynomials. Pincushion and barrel distortion are at least third order. Third order correction is specified by 10 coefficients. The polynomial has the form:

$$u=a_0+a_1x+a_2y+a_3x^2+a_4y^2+a_5xy+a_6x^3+a_7y^3+a_8x^2y+a_9y^2x$$
$$v=b_0+b_1x+b_2y+b_3x^2+b_4y^2+b_5xy+b_6x^3+b_7y^3+b_8x^2y+b_9y^2x$$

When the command is used the polynomial coefficients are not specified directly by the user, but are calculated from a set of image points and a set of map points. A least squares fitting procedure is used to find the polynomial coefficients. The command requires two Semper points list pictures (*Plists*). The *map* points are the coordinates of known features in the image. The *image* points are the position of these points measured in the image coordinate system.

To demonstrate the warp operation display image 3:1:

```
display 3:1 to display:1
```

Two sets of *Plist* pictures have been defined. Annotate the image with the map points:

```
mark display:1 with 3:2
```

Display the image points:

```
mark partition 2 with 3:3
```

Demonstrate warp with the command:

```
warp 3:1 to display:2 image 3:2 map 3:3 order 5
```

***

Problem

Image 3:4 is an aerial photo taken from a camera. The camera was not directly over the scene and hence the image is distorted. On the ground there were 4 surveyors at known positions shining torches into the sky so that they are visible on the photo. The positions of the light sources on the ground are:

(–112,80)
(–120,–118)
(120,38)
(56,–104)

Display the image with the following commands. The image points will appear highlighted:

```
display 3:4 to display:1
lut false reset create
lset range 240,255 sat 1 brightness 1 hue 0
```

Construct a map file and an image file and try to perform a distortion correction.

Hint: Use the **xwires** command to construct the map image file and **p**, the pixel set operator, to construct the image file.

***

Semper Training

Worksheet No 12: Visualisation of 2D Data

Introduction

Semper provides a number of ways of visualising 2D data. This is particularly useful where the pixel value may be interpreted as representing surface topography. This is the case when dealing with *tunnelling microscopy* images.

Starting the Semper Session

To start the Semper session, follow the instructions given below, that are relevant for your machine:

PCs

Insert the disk labelled "*Training Worksheet 12*" into the A drive.
Start Semper by typing:

```
semper /run=a:visual
```

Workstations

Start Semper by typing:

```
semper /run=visual
```

These commands start up the Semper session with the environment required for this worksheet.

Visualising 2D data

Try displaying the following image, which represents a scanning tunneling microscope image displayed conventionally with brightness proportional to surface height:

```
display 3:1 to display:2
```

The surface features may be thrown into relief in a number of ways. The simplest method is to use a contour map. Try the following:

```
contour 3:1 to display:3
```

Another way in which surface features may be represented is by a wire frame model. Try the following command:

```
ymodulus 3:1 to display:4
```

The most powerful of the commands in this area is the **sheet** command. This enables rendered surface views to be produced. It enables the user to specify the viewing angle and the angle at which the light sources are set up. In order to reduce the computation time, image number 2 is used which is a reduced size version of image 1.

Try the following command:

```
erase frame
sheet 3:2 to display:5 times 2
```

To view the image from another angle try the following command:

```
sheet 3:2 theta pi/9 psi pi/3 to display:4 times 2
```

The *theta* angle specifies the rotation about the z axis, the subsequent rotation about the x axis is specified by the angle *psi*, as shown in Figure 1 below:

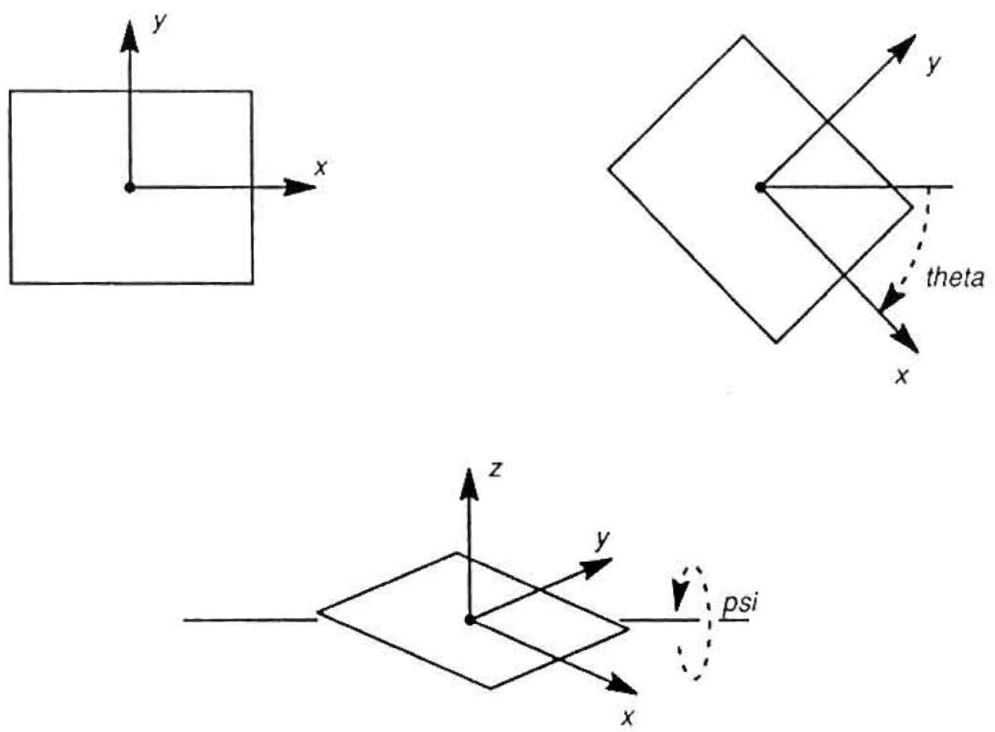


Figure 1. Changing the Viewing Orientation

Sheet also allows you to specify the intensity of three sources of illumination. These are a fixed source, a non-directional source and a movable source.

Try the following command:

```
sheet 3:2 to display:3 theta pi/9 psi pi/3 main 50 ambient 100+
+forward 40 ltheta pi lphi pi times 2
```

Note the the plus sign denotes a continuation line. This command sets up the specified viewing and lighting conditions.

Another factor under control of the user is the scaling and origin of the *z* axis in the picture. The **range** key specifies the range of heights to be represented by actual pixel values. Try the following command:

```
sheet 3:2 to display:4 range 1,100 times 2
```

***

Problem

Image 3:3 is a scanning tunneling microscope image courtesy of I.B.M. Zurich. Use the visualisation commands to obtain a good presentation of the image.

***

Semper Training

Worksheet No 13: Image Fourier Transforms and Power Spectra

Introduction

The display and calculation of fourier transforms and power spectra are common requirements in many areas of work. In electron microscopy, for example, the power spectrum of an image can reveal information of the underlying periodic structure of an image and on the operating conditions of the microscope.

Starting the Semper Session

To start the Semper session, follow the instructions given below, that are relevant for your machine:

PCs

Insert the disk labelled *"Training Worksheet 13"* into the A drive.
Start Semper by typing:

```
semper /run=a:four
```

Workstations

Start Semper by typing:

```
semper /run=four
```

Fourier Analysis

Display the first image from the disk using the command:

```
display 3:1 to display:1
```

If it is required to present the power spectrum visually then the outer edge of the picture is masked down to zero. This prevents artifacts due to edge effects. Try the commands:

```
erase frame
mask 3:1 to 4:1 radius 100 width 20
display 4:1 to display:2
```

Calculate the fourier transform with the command:

```
fourier 4:1 to 4:2
```

At this point try to display the result:

```
display 4:2 to display:4
```

The display requires some explanation.What is seen is the real and complex part of the fourier transform displayed side by side. The fourier transform of a real picture is symmetric so Semper calculates only the half plane image fourier transform to save space. The transform has a large dynamic range so very little detail may be seen when it is scaled to fit the display.

For display purposes it is useful to calculate the power spectrum. This is the magnitude of the complex fourier transform. Then the full plane can be reconstructed. Try the commands:

```
ps 4:2 to 4:3
fullplane 4:3
display 4:3 to display:3
```

At this stage all that is visible should be a small white dot in the centre of the image. This is because the central pixel in the fourier transform is normally much larger than all the other information in the picture and the automatic display scaling effectively hides the lower level information.

One way to see the power spectrum is to saturate the image above a certain value.This can be done with the **preset** option in the **display** command while setting *max*, the maximum pixel intensity, to a suitable value. Try the commands:

```
survey 4:3
max=max/100
display 4:3 to display:5 preset
```

Try reducing the saturation threshold further:

```
max=max/1000
display 4:3 to display:4 preset
```

Another technique that can be used to reduce the dynamic range is logarithmic compression. Try the following commands:

```
ps 4:2 to 4:4 ln
full 4:4
display 4:4 to display:2
```

The display of the log power spectrum can be further improved by saturating the lower levels to black. This can be done as follows:

```
survey 4:4 full
min=mean
display 4:4 preset to display:3
```

Interpreting the Power Spectrum

The discrete fourier transform may be considered to express a sum of basic sine wave components that are submultiples of the original sampling frequency. This process may be better understood by reference to the following images and their power spectra. Image 2 is a sine wave in the x direction with two cycles across the image which is of size 64,64:

```
display 3:2 to display:1
ps 3:2 to 4:1 ln
fullplane 4:1
display 4:1 to display:2
```

You will see that the power spectrum has a single off axis peak at pixel position 0,2. This is a low spatial frequency with two cycles across the image in the x direction and thus it is on the x axis at position 0,2 close to the centre. The value of the central peak is interpreted as the baseline level of the sine wave.

Image 3:3 is a high frequency sine wave in the y direction. Calculate and display the power spectrum:

```
display 3:3 to display:1
ps 3:3 to 4:1 ln
fullplane 4:1
display 4:1 to display:2
```

In this case the off axis peak is on the y axis at a position near the edge of the power spectrum.

Image 3:5 has frequency components in the x and y directions. Calculate the power spectrum:

```
display 3:5 to display:1
ps 3:5 to 4:1 ln
fullplane 4:1
display 4:1 to display:2
```

Here the peak has both x and y coordinates.

Image Filters

These examples suggest that filters in the fourier domain may be implemented. A radial symmetric high pass filter can be constructed by weighting the fourier components of an image down to zero outside a given radius. The **calculate** command can be used to produce a one dimensional function which can be used to weight the fourier transform to produce a filter of the desired characteristics.

In the case of a 256 square picture we will produce a high and low pass weight whose corner frequency is a quarter of the highest frequency present.

A low pass weight function can be produced using the **calculate** command

Try:

```
create 4:100 size 128,1 fp
origin left
calculate 1/(1+x*x*x*x/1e6)
display 4:100 to display:2
```

A high pass characteristic can be derived from the low pass weight

```
create 4:101 size 128,1 fp
origin left
calculate x*x*x*x/(1e6+x*x*x*x)
display 4:101 to display:3
```

These two weight functions can be used to filter an image. The fourier transform is calculated, the weight is applied and then the inverse transform applied to recover the image. To low pass filter try:

```
fourier 3:1 to 4:500
weight 4:500 with 4:100 to 4:501
image 4:501 to 4:502
display 4:502 to display:3
```

To high pass filter try:

```
weight 4:500 with 4:101 to 4:600
image 4:600 to 4:601
display 4:601 to display:4
```

***

Problem 1

Use the **calculate** command to produce a low pass filter whose cutoff corner is half the highest frequency in the image. Use this to filter image 3:1 and display the result.

Problem 2

Display image 3:5. This image has been corrupted by interference which is characterised by a narrow band of spatial frequencies. Consider how you might identify these frequencies and construct a filter to remove them.

Hint: Consider the use of the **mask** command.

***