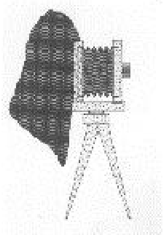
IMAGIC-5

WELCOME TO THE

MAGIC WORLD

OF IMAGIC



BEGINNER

IMAGIC-5 BEGINNER

The BEGINNER manual contains the following introductions:

- PLAYING WITH IMAGIC

  First introduction to working with IMAGIC in which the new user will get to know the generalities of the system.

- MINI COURSE IN IMAGE PROCESSING

  The purpose of this document is to provide the user with guidance in a practical HANDS ON introduction to various aspects of the user-friendly IMAGIC image processing software package.

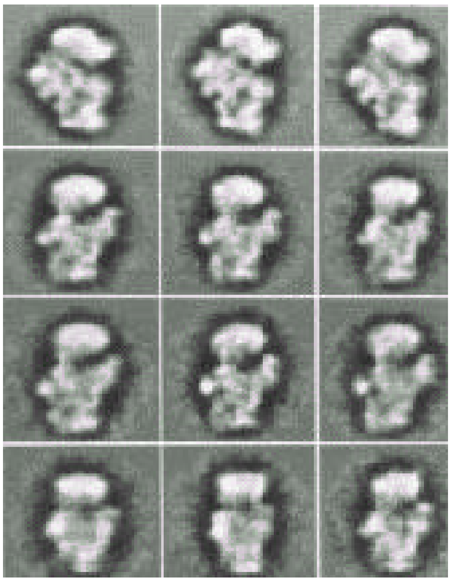
- HANDS-ON IN THE UNIX OPERATING SYSTEM

  This brief introduction will show you some of the important commands of the operating system which you might need for image processing.

- HOW TO USE IMAGIC-5 AS AN MS WINDOWS CONSOLE APPLICATION

  This brief introduction will give you some information on how to use IMAGIC-5 in an MS Windows environment.

PLAYING WITH IMAGIC



"Playing with IMAGIC" is a "hands-on" first introduction to working with IMAGIC in which the new user will get to know the generalities of the system. You will have to "log on" to the computer system and learn how to use the general facilities of the IMAGIC image processing system like: creating a test image, displaying the result, using the interactive help facilities etc. At the end of the session you will learn how to clean up the mess you have been creating in your own corner of the computer.

1. THE LOGIN PROCEDURE

Before we can do anything with the computer (like image processing) the computer has to recognise us as a legitimate user in a procedure know as "logging in". This procedure may be different for each computer and for each installation.

As an example, you could be sitting in front of a workstation which prompts you for a username and a corresponding password, as in:

```
USERNAME:
```

Your user name (contact your system manager to get an account) could, for example, be: myname. Type this name and press RETURN or ENTER again after which the computer might answer:

```
PASSWORD:
```

Type your password followed by (as always) RETURN or ENTER . Since every answer we give to the computer is closed by the pressing the RETURN or ENTER key, we will no longer mention this fact explicitly from now on.

As mentioned before, the procedure may be different for each computer and each installation, but the computer will, after you have logged in answer with the so called system prompt which could look like:

```
$
```

VAX/VMS or:

```
usr/myname>
```

UNIX

In an X Windows system you will not find a prompt. You have to create a special window ("command shell", "command window", the name depends on the workstation you are using; ask your system manager for help) to "find" a prompt.

You are now logged on the computer and all the resources of the computer system are available to you. Your are now directly talking with the Operating System (OS) of the host computer ("shell"). There are very many commands to the host operating system which have nothing to do with image processing. To get an idea (and to get confused) type: HELP (VAX/VMS) and see what happens. In UNIX the equivalent of the HELP command is man. If still nothing happens ask your local computer "guru" for help.

NOTE: there is little that the new user can do wrong at this stage so don't hesitate to just try some operations. This is true for both the OS as for the IMAGIC system as long as there is no important information in the directory the user is working in.

The inverse of the LOGIN operation is called the "logout" operation. It is good practice to logoff when you are not actually using your workstation or terminal! To try that operation you have to type:

```
$ logout
```

VAX/VMS

```
usr/myname> exit
```

UNIX

You will then have to login again of course!

2. LOOKING AROUND IN IMAGIC

The IMAGIC system can be started by typing imagic or just i (depending on your login definitions):

```
∗∗ IMAGIC-5 welcomes you ∗∗

IMAGIC-COMMAND:
```

If nothing happens when you type this magic code, your login procedure may need modification (contact your system manager if necessary).

It is also possible that prior to the IMAGIC command prompt an additional remark appears on the screen stating that there is a user-command file in the project you are working under. As a beginning user you can simply ignore that statement. What a user-command is, is explained elsewhere in the written documentation.

IMPORTANT: One of the design philosophies of IMAGIC is that it should guide its users by providing a sufficient amount of feed-back in the form of interactive help. IMAGIC is meant to be used without continuously scanning through thick manuals. The idea is that even the inexperienced user should find his or her way around in the system by just trying the various options. What is written in this document is meant only as help to get you over a first threshold.

As answers to the IMAGIC-COMMAND prompt, first try such things as ? , HELP and MENU and carefully read all the information that flows over the screen. If necessary, stop the flow of information by pressing the HOLD SCREEN or PAUSE key. Try all the variants of these commands that are suggested in the "information flow".

Examples:

```
? ali
? ∗ali
? *test
help test
help ∗te
```

Try starting any imagic command! Ask help for all questions that appear on the screen. The worst thing that can happen in any normal command (while you are walking around in IMAGIC) is that you accidentally break out of the system and will have to restart IMAGIC.

NOTE: in the IMAGIC system the general quit command is the * . It is a legitimate answer to any question that appears on your screen and its effect is to leave the current operation. When used as an answer to the IMAGIC command prompt, you leave IMAGIC and you will have to restart it as mentioned before. When used as an answer to any specific question asked by a program or by a command sequence it will bring you back to the imagic command prompt. You can thus go into whatever command you like and escape later by typing the * .

3. CREATING YOUR FIRST IMAGE

This is an example how to use an IMAGIC command.

To create a first image to play around with search for commands containing the string TEST . If you have done what was suggested under point #2, you now know that you should type something like

```
IMAGIC-COMMAND : help *test
```

or

```
IMAGIC-COMMAND : ? *test
```

to find the appropriate command (TEST-IMAGE-CREATION). Go ahead and create a test image. Do not forget to ask for HELP (or ?) for each question that the program asks you.

Example: Create a test image showing a Siemens star and store it in an IMAGIC image file.":

```
Output filename, image loc#s      [test] :
                                          you can give a system
                                          command (for example
                                          to list the directory) by
                                          using a ! or $:
Output filename, image loc#s      [test] : ! ls *.img
                                          (UNIX style)
                                          (VMS/NT style: $ dir *.img)

     my_file1.img
     my_file2.img

Output filename, image loc#s      [test] :
                                          if you only give RETURN
                                          then the default value in
                                          brackets (here "test") is
                                          used as your answer, else
                                          you select the file name
Image dimensions X,Y [128,128]           : 256,256
                                          your own answer

IMAGIC data formats you can choose:
  PACK  INTG  REAL  COMP  RECO
Please specify option [REAL]             : REAL
                                          if you need help give a ?
Currently, you can choose:
  BLOBS      CHECKERS     CIRCLE    CONSTANT
  DISC       GAUSSNOISE   GRID      MANDELBROT
  M2NDEL     MCCACT       NOISE     NUMCOL
  NUMLINE    NUMLOCATION  NUMPIXEL  POINTS
  RECTANGLE  SIEMENS      SPIRAL    WEDGE
Please specify option [BLOBS]            : SIEMENS
```

Once you have your first image on disk, you may want to see what it looks like. In the next chapter you will learn how to use the DISPLAY command to visualise IMAGIC images.

4. THE DISPLAY SYSTEM

One of the most frequently used IMAGIC commands is the DISPLAY command. In an windows environment, you can (and often will) start multiple display windows. To start up the display system type the IMAGIC command DISPLAY-IMAGE:

```
IMAGIC-COMMAND: DISP

** DISPLAY welcomes you **

Input image file, image loc#s       :
```

The first question that will appear on the screen concerns the choice of the file you wish to display; go ahead and try to get one of the test images you just created on the screen. If you have forgotten the names of the images type:

```
Input image file, image loc#s       : $ DIR      VAX/VMS
```

or

```
Input image file, image loc#s       : ! ls       UNIX
```

which is just an operating system call to get a list of the files you own, and look for files with the extension .img .

Now specify the name of the image file you want to display:

```
Input image file, image loc#s       : test
```

NOTE: to understand the organisation of the IMAGIC image files read the corresponding manual text.

Before continuing IMAGIC first shows you some information about the specified image(s), like:

```
Image name and parameters [and history]:
TEST IMAGE
SIZE: 64,64  LOC: 1  TYPE: REAL  CREA.DATE:21-1-1996
TESTIM:SIEMENS STAR,
```

The next question asks for the display system you want to work with. This question only appears if there is a choice, that is, if the installation has more than one display system. The DISPLAY command has, like every IMAGIC program, a considerable amount of interactive help available: don't hesitate to answer every question you see with ? . Usually you are working on a windows system and this question will not appear.

In a windows environment IMAGIC will ask you to give the display window size and style of the cursor. For example:

```
Give size of the display window [512,512]  : 1024,1024
CROSS, SQUARE or CIRCULAR cursor [CROSS]   :
```

Then IMAGIC will then show a list of all parameters (display setting) that are currently set for displaying your images, for example:

```
Current DISPLAY settings:

Input image FILE name              : test
LOCATION numbers                   :    1  to:   1
Output DEVICE                      : XWINDOWS
SCALE factor                       :    1.0000
MINX, MAXX                         :    1  to:   256
MINY, MAXY                         :    1  to:   256
GREYVALUES                         : AUTOMATIC
ERASE screen                       : NO
STARTing point (top left)          :    1,   1
Display of NAME and location       : LOCATION
Video lookup table (VLT)           : STANDARD BLACK/WHITE

Change parameter settings
(ERASE,SCALE,LOC,START,SCAN,SAVE,BYE ...)? [NO]  :
```

If you do not understand the meaning of some parameter, don't hesitate to answer ?.

Then just give RETURN or ENTER, which means "NO MORE CHANGES":

```
Change parameter settings
(ERASE,SCALE,LOC,START,SCAN,SAVE,BYE ...)? [NO]  : NO
```

IMAGIC will display your image(s) and will return to the parameter setting as shown above.

If you want to change some of the parameters, please look at the words written in CAPITALS. If you answer the question "Change parameter settings ?" by typing such a word you can change the corresponding parameter(s).

Example:

In the parameter setting above the scale factor is 1.0. If you want a scaling of 1.3, answer:

```
Change parameter settings
(ERASE,SCALE,LOC,START,SCAN,SAVE,BYE ...)? [NO] : SCALE
```

IMAGIC will show you the current image size and ask you the next question:

```
  Image size is :      256 x 256

Give scale factor for display [Def=1]           :
```

Specify the desired value, like:

```
  Image size is :       64 x 64

Give scale factor for display [Def=1]           : 1.3
```

IMAGIC will change the parameter and show you the new settings:

```
Current DISPLAY settings:

Input image FILE name               : test
LOCATION numbers                    :    1  to:  1
Output DEVICE                       : XWINDOWS
SCALE factor                        :    1.3000
MINX, MAXX                          :    1  to:  256
MINY, MAXY                          :    1  to:  256
GREYVALUES                          : AUTOMATIC
ERASE screen                        : NO
STARTing point (top left)           :  1,  1
Display of NAME and location        : LOCATION ONLY
Video lookup table (VLT)            : STANDARD BLACK/WHITE

Change parameter settings
(ERASE,SCALE,LOC,START,SCAN,SAVE,BYE ...)? [NO]  :
```

You don't have to worry if want to change more than just one parameter, the command will loop back to this question after you changed something. Once the setting will be correct and you want to see the images. Answer "NO CHANGES" by giving RETURN or ENTER :

```
Change parameter settings
(ERASE,SCALE,LOC,START,SCAN,SAVE,BYE ...)? [NO]  : NO
```

IMAGIC will display the images #1, #2 and #3 of file test on the windows terminal using the specified parameters.

You should play around a little bit and/or read the description of the DISPLAY command. If you want to leave the DISPLAY program give BYE or *. Also try to use command EXIT and see what happens:

```
Change parameter settings
(ERASE,SCALE,LOC,START,SCAN,SAVE,BYE ...)? [NO]  : BYE
```

5. IMAGE SEQUENCES IN IMAGIC

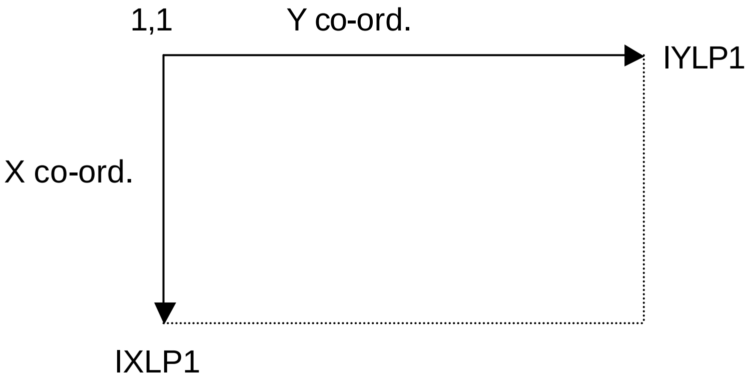
In image processing, and in particular in electron microscopy, one often wants to work with large sets of similar images. IMAGIC is very efficient in dealing with such large sequences since an IMAGIC image file can contain such a sequence of images. If a user specifies an input file name to an IMAGIC command and the input file contains a sequence of images, all images in the sequence will automatically be treated by the procedure unless the user specifies a set of image location numbers explicitly, like in:

```
IMAGIC-COMMAND: COPY-IM

** COPYIM welcomes you **

Input file, image loc#s                     : test,10,64
                              etc.
```

in which case the images number 10 to 64 are used-

6. THE IMAGIC COORDINATE SYSTEM

The IMAGIC co-ordinate system is a right-handed system with its (1,1) origin in the top-left corner of the image. The length of the lines is IYLP1 and the number of lines is IXLP1:



The centre of the image, for almost all operations on the image, is given by:

(IXMID,IYMID) = (IXLP1/2 + 1 , IYLP1/2 + 1).

This point is also the default centre of rotational symmetric masks and the like.

7. STARTING IMAGIC FROM MANY WINDOWS

You can start IMAGIC-5 in more than only one terminal window. For example: it can be very helpful to use one window for command DISPLAY while playing around / testing filter parameters with command BAND-PASS-FILTER in an other window.

8. SOME SUGGESTIONS

To continue your expedition through the IMAGIC system we give some further suggestions (see also: "MINI COURSE IN IMAGE PROCESSING"):

a) POWER SPECTRA

Try creating different types of test images. Display the various images. Calculate the forward Fourier transform (command FFT2D with option FORWARD) of the different types of images and display these too.

Note that the Fourier transforms are of complex type which - in principle - can not be displayed directly. Therefore the DISPLAY program automatically displays only the power spectra of the images (actually the log of the intensities of the complex images).

b) <u>DIGITISE A REAL-LIFE IMAGE</u>

Usually images are read in using an on-line automatic densitometer which scans your input negative and stores the density values of the digitised image as a disk file directly. If the images do not have IMAGIC format, the (format conversion) action will be done by command IMPORT-EXPORT" (try HELP ∗IM-EX, for example)).

Display the new image on a display system (command DISPLAY). The first logical step after having read in the image is to check its quality by calculating a power spectrum of the input image or of a part thereof; this is the purpose of the following operations.

Use DISPLAY to look for a nice-looking sub-area of the image. Use option COOS get the central co-ordinates of this sub-area:

```
Change parameter settings

(ERASE,SCALE,LOC,START,SCAN,SAVE,BYE...)? [NO]  : COOS
Should the values be stored in a PLT file?      : YES

                        ...
```

To stop reading co-ordinates select the same point twice or give STOP.

Leave the DISPLAY command with BYE or start IMAGIC in another window to extract the selected sub-area from your input image. Use the CUT-IMAGE command (try: HELP ∗CUT). In CUT-IMAGE use option APERIODIC. Later you can also play around with other options like CENTER, GENERAL etc. DISPLAY the extracted sub-file; calculate its Fourier transform (command FFT2D, option FORWARD). Don't forget to give HELP ∗FFT) and display the Fourier transform too.

You will probably see "spikes" in the power spectrum (overwhelming cross-shaped details in the centre of the image). These arise from the "wrap-around artefact": the FFT algorithm assumes the input images to be cyclical in both the X and the Y direction, meaning that sharp jump can result from the difference between the right edge and the left edge of the image and from the difference between the lower edge and the top edge.

To avoid spikes place a circular mask on the (sub)image using the PRETREAT-WHOLE-IMAGE command (options NORMVAR or ZEROFLOAT and CIRCLE) prior to the calculation of FFT2D. Display the Fourier transforms.

C) IMAGE FILTERING

Filtering operations such as low-pass filtering or band-pass filtering are often of crucial importance in image analysis. They are also conceptually very important since they allow us to get a direct feeling for the concepts of Fourier optics. Filtering is normally implemented as a multiplicative mask in Fourier space. This operation is equivalent to a convolution in real space.

Further information you will find in chapter "MINI COURSE IN IMAGE PROCESSING and manuals "SINGLE MOLECULES IN IMAGIC" or "2D-CRYSTALS IN IMAGIC".

9. BATCH / COMMAND FILES

Imagine, that you have a huge sequence of images and you want to calculate Fourier the transform for each of them. This corresponds to a lot of work for your computer and costs a lot of its (and your) time. While calculating all this stuff your terminal remains allocated.

There might be another problem: you want to start a sequence of commands and you do not want to wait until the end of each program to start the next IMAGIC command.

To deal with such problems IMAGIC can "accumulate" commands and can create command or job files.

HOW TO ACCUMULATE COMMANDS

First call IMAGIC and give the command:

```
IMAGIC-COMMAND : MODE-ACCU
```

IMAGIC answers with:

```
IMAGIC-COMMAND (ACC.) :
```

Now you can give any IMAGIC command you want. As usual give the corresponding answers to all questions of IMAGIC and use ? to get help. You can also give more than one command here. If you want to stop after all the commands are given, then type the IMAGIC command MODE-STOP:

```
IMAGIC-COMMAND (ACC.) : MODE-STOP
```

IMAGIC will now create a batch / command file that contains all the commands you have typed in one after the other. IMAGIC prompts you for a name to be given to this command file.

```
IMAGIC-COMMAND (ACC.) : MODE-STOP

Give file name for command file [bigjob] :
```

You can give any name you want. Example:

```
IMAGIC-COMMAND (ACC.) : MODE-STOP

Give file name for command file [bigjob] : myjob
```

IMAGIC will create a command / batch file using the specified name with the extension "..b" (UNIX) or ".com" (VAX/VMS). IMAGIC will also give you some help how to start the command / batch file:

```
To run your job on the monitor please use
the UNIX command:  my_job.b                         UNIX

To run it as a hidden job please use the UNIX
command: myjob.b > myjob.log &
```

or

```
To submit your job to the batch processor
please use VAX/VMS command:
SUBMIT/LOG=MYJOB.LOG MY_MYJOB.COM                   VAX/VMS

For help use VAX/VMS and type: HELP SUBMIT
```

Before starting the job you can, of course, edit the corresponding command / batch file.

NOTE: In IMAGIC-COMMAND (ACC.): you can also use ? or HELP as an answer to every question and * to leave a given command without aborting the MODE-ACCUMULATE status.

NOTE: If you want to interrupt the whole process because all of input is wrong or you want to stop your work and go have a beer (or many because of FRUSTRATION-DAY(ACC.)), you can type instead:

```
IMAGIC-COMMAND (ACC.) : MODE-ABORT
```

IMAGIC then ignores all commands and returns to the normal mode with:

```
IMAGIC-COMMAND:
```

See also commands MODE-SEND-TO-BATCH and MODE-RUN which can be used instead of MODE-STOP.

10. ADMINISTRATION OR CLEANING UP THE MESS

Last but not least, it is easy to create and to maintain a complete chaos in your corner of the computer. This, we must say, has some disadvantages:

a) You will soon loose overview.

b) You will use up much more disk space than necessary and this, in turn, will bring you in direct conflict with other users.

c) Nobody else will understand what you are doing which may have serious consequences. For example, some important types of images may be backed up automatically every day on the basis of the file name. Choosing some fancy non-standard name for a standard type of image can thus even lead to the loss of important data.

Important IMAGIC commands in this context:

DELETE-IMAGE

Important operating system commands are:

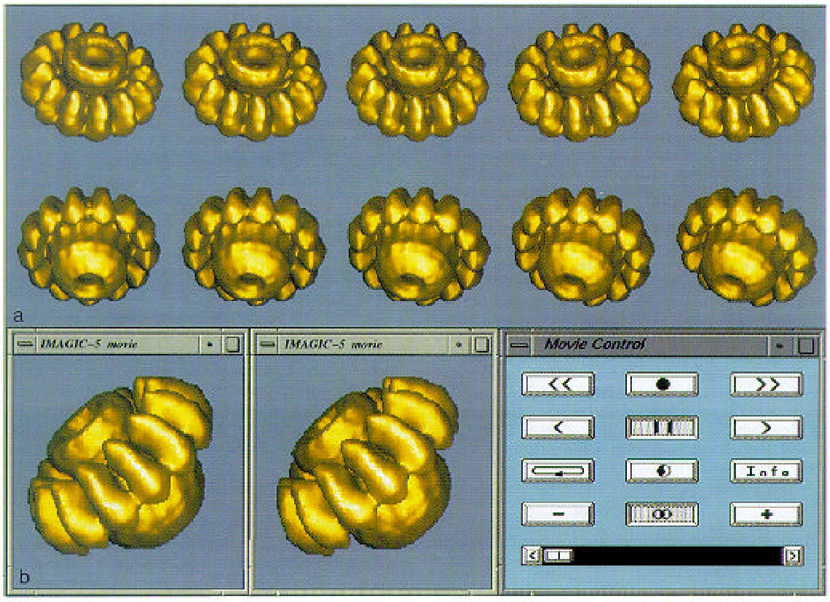
```
$   DIR   *.*
$   DIR/DAT=MODI   *.IMG
$   DIR/SIZE=ALL

$   DEL/CONFIRM   *.*
```

etc. (VAX/VMS)

```
>   ls   *
>   ls -lisa   *.img

>   rm   -i   *.*
```

etc. (UNIX)

MINI COURSE IN IMAGE PROCESSING



The purpose of this document is to provide friendly users with guidance in a practical hands on INTRODUCTION to various aspects of the user-friendly IMAGIC image analysis software package.

PANIC AND WORRY

Don't.

Please remember:

You can always ask for help. Type HELP followed by the command name to get an explanation of what the particular command does.

Within a command, type a question mark ? to get helpful explanation of what is being asked. First use the default answers (in brackets), but if you need some trouble at any point try your own one.

You can always use the default answers (in brackets) by just giving CR , but, of course, you can also type your own ones.

At any level of IMAGIC-5 you can give an operating system command by typing ! or $ followed by the system command.

When you are really ready to bail out, type * until you get back to the operating system.

BASIC IMAGE MANIPULATION

To get a feeling what images in a computer are and what is (simple) image processing, you will play around with some easy IMAGIC-5 commands. First you will create some test images like checker-boards, stars, blobs etc. Then you will use some commands to display your test models and to see how those images can be treated further.

Please note: in the following text all IMAGIC-5 commands are written in GREEN.

(1) Use TEST-IMAGE to create a test image, viz., test. First use the default options, i.e., just give carriage returns.

(2) DISPLAY the image on the screen. Then use option GREY to give different grey levels, e.g., -15,15 or 20,25.

Also try different SCALE factors: 2, 5, 200.

(3) Play with TEST-IMAGE again. Create REAL images, but of different sizes, and of different types. DISPLAY the images.

At the end, create a 128x128 image of SIEMENS. DISPLAY it to make sure it's okay.

(4) Use DISPLAY to select points which will be the central co-ordinate of sub-areas of your image. This is done with the COOS and YES STORE option. The co-ordinates are written to an output file with the extension ".plt".

(5) Use CUT-IMAGE with the APERIODIC option to cut-out the specified sub-areas of the image which will be stored in a new file. DISPLAY these smaller images.

(6) Create an other test image containing BLOBS. Use DISPLAY to select a number of points which define a contour around an object / molecule of your image. This is done with the COOS and YES STORE option. The co-ordinates are written to an output file with the extension ".plt".

(7) Use PRETREAT-IMAGE with the CONTOUR option. Inputs are the image and the ".plt" file you have used/created in (6). DISPLAY this contoured image.

(8) Use the COARSE command with factors of 2, 4, and 6 to give coarsened images.

DISPLAY them and see the effects of digitisation size on your resolution.

(9) Use the BLOW-UP-IMAGE command (option BLOWUP) to blow up the coarsened images back to the original size.

DISPLAY the results and compare them with the original images.

(10) Back to the original image, use ROTATE-IMAGE to ROTATE, REALSHIFT, and BLOWUP the image (or any combination of these three).

DISPLAY the results that you have performed.

CREATING A SET OF TEST IMAGES WITH ADDED NOISE

You will create a sequence of test images and add noise to them. In the subsequent sections, you will practice various means of restoring these noisy images.

(1) Use TEST-IMAGE to create 128 test images, viz., test,1,128, of CHECKERS. Make the images REAL and of size 128,128.

(2) Use TEST-IMAGE to create 128 test images, viz., noise,1,128, of GAUSSIAN noise with a default standard deviation. Make the images REAL and 128,128.

(3) Use ADD-IMAGES (option ADD) to add corresponding images of test and noise to give tnoise.

DISPLAY the results.

NOISE AMELIORATION BY IMAGE COARSENING

Coarsening is one of the tools to reduce noise in images.

(1 Use COARSE on "tnoise" with factors of 2 and 4, giving coarse2, coarse4, ...

(2) DISPLAY the results and see the relationship between resolution of visual information and noise in images. You can also use GREY-LEVEL-HISTOGRAM and FFT2D with option FORWARD.

NOISE AMELIORATION BY BAND-PASS FILTERING

Filtering an image is another important tool to minimise noise. Filter can reduce the influence of unimportant or disturbing spatial frequencies.

The very high frequencies represent mostly noise and their contribution can be weighted down using a special filter.

More important in practice is the suppression of the very low spatial frequencies which also represent unwanted information, e.g., gradual fluctuations in the average densities which depend largely on the amount and uniformity of specimen illumination.

In "real science" the frequency limits of band-pass filtering generally vary from specimen to specimen and will be used in a way that they fit to size of the particle and the expected resolution.

(1) Use the command INCORE-BAND-PASS to filter out the high frequencies in tnoise. Use LF cut off = 0.0, remaining LF transmission = 1.0 and HF cut off = 0.5, giving filt1. Type ? to get help.

DISPLAY the low-pass filtered images to see the effect. As before, use GREY-LEVEL-HISTOGRAM. You can also check the Fourier transforms (command FFT2D with option FORWARD).

(2) Use the command INCORE-BAND-PASS to filter out the low frequencies in tnoise. Therefore use LF cut off = 0.2, remaining LF transmission = 0.005 and HF cut off = 1.0, giving filt2.

Again DISPLAY the high-pass filtered images to see the effect. You can also use GREY-LEVEL-HISTOGRAM and FFT2D with option FORWARD.

(3) Now use the command INCORE-BAND-PASS to filter out both, the high and the low frequency noise in the images tnoise (giving filt3). Use LF cut off = 0.1, remaining LF transmission = 0.05 and HF cut off = 0.8.

As before use DISPLAY, GREY-LEVEL-HISTOGRAM and FFT2D with option FORWARD to analyse the results.

(4) Continue with (3) using other filter parameters.

(5) Redo (4) with some microscopical images of worm hemoglobin (if available) (you can use: input file, loc#s: pic:h_sml,1,10).

Again play around with the filter parameters and check the results with DISPLAY.

NOISE AMELIORATION BY IMAGE AVERAGING

You will get some impression what image averaging means and why image processing can enhance the resolution of noisy images:

(1) Use INCORE-SUM (option SOMESUM) to make sums of 1, 2, 4, 8, 16, 32, 64, and 128 of the images in the file tnoise to get output files sum1, sum2, sum4, ..., sum128, respectively.

DISPLAY each result and see the effects of image averaging on the signal to noise ratio.

(2) Use GREY-LEVEL-HISTOGRAM to display the histograms of the grey levels of sums 1, 2, 4, ..., 128. Use the grey level range of -60 to +60, with 30 bins and a width of 80.

Also try to use PLOT to display the results (input is the GREY-LEVEL-HISTOGRAM output file with extension "plt").

(3) Use FFT2D, option FORWARD to calculate the Fourier transforms of the sums, giving fft1, fft2, fft3, ..., fft128 . DISPLAY the transforms (i.e., the power spectra) and see the relative powers of noise versus the periodic information.

THREE DIMENSIONAL IMAGES
(The 3D module is required)

In the meanwhile you have got some feeling about 2D images. OK, let's go ahead and continue with the processing of three dimensional images. Remember the ? to get help for every question.

(1) Create 3D test images using THREED-TEST-IMAGE or THREED-MODEL, giving test3d1, test3d2, ...

DISPLAY your results to see how those images are stored and handled in an IMAGIC-5 file.

(2) An other representation of 3D images as in (1) is the use of stereo images. Create those images from your 3D models test3d1, test3d2, ... with the command THREED-SURFACE-VIEWS. First use the default options, giving surface1, surface2, ...

DISPLAY and compare all results.

Use THREED-SURFACE-VIEWS with option STEREO to create stereo views. Use MOVIE to look at the moving surface representations.

Move the cursor into the image and click with the middle mouse button. In the control window click the "**8**" switch to get moving stereo images. Now roll your eyes and try to see neighboured images in stereo (3D). (Use the "●" switch to leave MOVIE.)

TRUE 3D RECONSTRUCTION FROM PROJECTIONS
(The 3D module is required)

High resolution electron microscopy is transmission microscopy and hence produces images that are 2D projections of the explored particles. But if there are different views of the same particle and the angle between the various projections is known, it is possible to calculate the 3D volume.

(1) Create a 3D test image using THREED-TEST-IMAGE or THREED-MODEL-MOLECULE. Preferably make something that is contained within a roughly spherical region.

DISPLAY the results.

(2) Also use THREED-SURFACE-VIEWS to see how the model looks like.

(3) Create 2D projections of your 3D test image using THREED-FORWARD option FORWARD.

Use DISPLAY to look at the results.

(4) Reconstruct the 3D image using TRUE-THREED-RECONSTRUCTION, option EXACT using the projections in (3) as input. Before playing around, use the default values.

(5) Calculate stereo images of the reconstructed volume using THREED-SURFACE.

DISPLAY or MOVIE all results on the monitor and compare them with the results in (1) and (2). Use a larger number of projections in (3) and re-do the 3D reconstruction (4-5). Compare the results.

HANDS ON IN THE UNIX OPERATING SYSTEM

This brief introduction will show you some of the important commands of the UNIX operating system which you might need for image processing. However, it is not the purpose of this hands on to train you to become a system manager of your computer.

1. THE LOGIN PROCEDURE

Before anything in the computer can be done (like testing the commands in this little "Hands On") you first must "log in".

If you are sure that your terminal is turned on you press the key

RETURN

or

ENTER

or

↵

(more than once if necessary).

The computer answers with some words of welcome, like

```
THIS IS NODE VENUS
UNIX System V Release 3.1
```

(for example) and then probably will ask you something like

```
login:
```

or

```
USERNAME:
```

Your username is required here (most probably your own name). If you do not know this name ask your system manager to get help.

Give this username here. Press again RETURN (ENTER) and the computer asks:

```
Password:
```

The answer to this question is a (more or less) secret word for every username to open the "door" to the computer. If the combination of username and password is OK, the computer recognises you as a legitimate user.

May be there are now some friendly words of welcome. After all you reach the so called "system prompt"

```
venus$                          (for example)
```

or

```
>                               (for example)
```

Now you are logged on the UNIX computer and the operating system of the computer is available to you. The login command has also done some useful assignments and definitions for your further work (if necessary ask your system manager for help).

Work can begin now. Perhaps you must use the mouse, move the curser to an available window and press the right mouse bottom. Then, typing

```
imagic
```

or

```
i
```

will run the IMAGIC software.

2. LOGOUT

The reverse to "login" operation is the so called "logoff" operation to determine the communication with your computer.

Type:

```
logout
```

or

```
exit
```

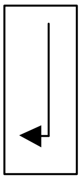
and give RETURN (ENTER) or just give

```
ctrl d
```

Now your computer session is at its end.

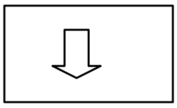
3. SOME USEFUL TERMINAL KEYS

a) RETURN or ENTER



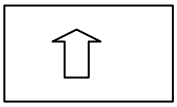
After every answer or command you must press this key which let the computer know that your input is finished. From now on this fact will no longer be mentioned explicitly.

b) CAPS LOCK



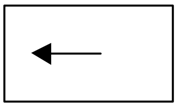
This key changes the mode whether a given character is written in capitals or not.

c) SHIFT

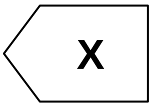


The shift key is always used together with another key of your terminal. If you press the SHIFT and a character key, then this character is written in capitals. Some of the terminal keys can write more then one sign. The sign that is written on top of those keys can be chosen by pressing the key and SHIFT simultaneously.

d) DELETE



or:



deletes the last input sign (without giving RETURN (ENTER)).

h) CTRL

The computer knows more sign codes than are written on the terminal keys. Together with the ctrl key it is possible to generate some further commands. Without giving RETURN you press "CTRL" and a corresponding other key simultaneously, like:

CTRL and z

stops the current computer activity running on your terminal.

CTRL and d

is the logout command.

4. HELP

If you need help you must not read the whole UNIX hand-books, they are frighteningly thick anyway. For many questions and problems there are little explanations in the computer itself. If you need information about a special command then type "man" and the name of the command, as written below:

```
man  command
```

where `command` is the name of the command you need help for.

Use the space bar or RETURN (or ENTER) to continue within the helping text and `q` to quit.

For further help type:

```
man man
```

5. WILD CARDS: ✶ AND %

These are the so-called "wild cards". With many commands you can reach not only one file or one directory. With the wild cards you can work with several files that have similar names. With ✶ you can call a string with any given length, while `%` is the wild card for only one sign.

Examples

`A✶.✶`	all files with any extension beginning with A
`✶junk✶.f`	all files with the extension `.f` and a name containing `junk`
`✶.%`	all files with an extension of one sign

6. COMMANDS TO HAVE A "LOOK" INTO DIRECTORIES

One of the corresponding commands is:

```
ls
```

which shows a list of all files of the default directory in alphabetic order. Using wild cards (✶ or %) you can search for particular files, in which you are interested at the moment.

Examples:

`ls ✶.img`	display all files with the extension `.img`
`ls /usr/bin/✶.hed`	display all files with the extension `.hed` in the directory named `/usr/bin`

There are many qualifiers with you can use to get more information:

`ls -a`	displays also "hidden" files
`ls -l`	a "long" list is shown
`ls -s`	shows the size of the files

You can also use any possible combination, like:

`ls -lisa ✶.✶`	which displays all files with every available information

There is another important command:

```
find
```

which looks for specified files.

Examples:

`find *.for` searches for all files with the extension .for in the default directory

`find /usr/*.txt` searches for all files with the extension .txt in the directory /usr

Some useful qualifiers:

`find -size N` displays all files with size N

`find -atimeN` displays all files that have been used within the last N days

`find -mtimeN` displays all files that have been modified within the last N days

`find -user NAME` displays all files that belong to the user named NAME

7. COMMANDS TO DELETE FILES

Some useful commands not only for the "dustman".

a) rm deletes the specified files.

Examples:

`rm ✶.img` deletes all files with the extension .img on the default directory

`rm ✶.HED` deletes all files with extension .HED

`rm /home/A✶.✶` deletes all files beginning with A in the directory named /home

`rm ✶.✶` ATTENTION: Delete all files !!

b) Interactive

`rm -i` same command as rm , but for every file you are asked if you are sure that this file should be deleted. This is a very useful command if you are using wild cards.

8. SOME COMMANDS TO HANDLE WITH FILES

a) Copy files

`cp` copies the input file to the output one (without deleting the input)

Examples:

```
cp  MYIMAGE. ✶ /home/ourimage. ✶
```

copies all files named MYIMAGE (with any extension) in the default directory to the directory named /home with the new names ourimage . The extensions remain the same.

```
cp  ✶.✶ /usr/✶.✶
```

copies all files in the default directory to the directory named /usr without changing names and their extensions.

b) Move / rename files

```
mv
```

changes (moves) the input file name to the new specified one.

Examples:

```
mv  text.lis  yourtext.txt
```

the file called text.lis gets the new name yourtext.txt .

```
mv  text.txt /home/mytext.txt
```

the files named text.txt are moved from the current directory to the directory /home with the new name mytext.txt .

c) Differences between files

```
diff
```

compares two files and displays all records that do not match.

`comp` this similar command only shows the first difference between the two files. This is useful if you only want to see if two files are the same or not.

Examples:

```
comp text1.lis Text2.lis
```

compares the file `text.list` with the file named `Text2.lis` and displays the first difference.

```
diff text1.lis /usr/text2.lis
```

compares `text.list` in the default directory with `text2.lis` in directory `/usr` and displays all differences.

A useful qualifier is:

`diff -b` ignores differences due to spaces in the files.

c) Search in files

`grep` searches a string of signs in the specified files.

Example:

`grep HELP *.f` the operating system searches the string HELP in all files with the extension .f

There are some useful qualifiers:

`grep -v` all lines are printed that do NOT contain the searched pattern

`grep -l` the name of the files that contain the searched pattern are displayed `grep -I` ignores differences due to uppercase or lowercase characters

`grep -f` prints the number of the lines where differences are found

9. SOME USEFUL COMMANDS TO READ AND WRITE "TEXT" FILES

Some UNIX commands are:

a) vi

The most common UNIX editor is called by typing

`vi filename` where filename is the name of the file you want to read or edit

To get further help please give

```
man vi
```

and/or ask your system manager.

c) cat

`cat file` the specified file named file is printed on the terminal. Similar commands are `pg` " or `more` (use command `man` to see if the commands are available in your system).

d) lp or lpr

`lp` `lpr`	prints the specified file on the (default) printer.

There are some useful qualifiers:

`lp  -nX`
`lpr -#X`
	X copies are printed (X = 1,2,3,...)

`lp  -dname`
`lpr -Pname`
	the printer called *name* is used

10. RUN JOBS OR PROGRAMS

Just type the command or program name.

11. COMMANDS TO GET SYSTEM INFORMATION

Some corresponding commands are:

a) ps

`ps`	shows some information about the current system processes.

b) pwd

`pwd`	shows the default directory you are working in.

c) df

`df` shows free disc(s) space.

d) who

`who` shows all users that are working at present with the computer. In addition the users terminal names and login times are shown.

e) lpq or list

`lpq`
`list` list jobs in a queue.

12. COMMANDS FOR DIVERSE SYSTEM SETTINGS

Some corresponding commands are:

a) cd

`cd` command to set the default directory you want to work in.

Example:

```
cd /usr/bin/imagic
```

the default directory is set to /usr/bin/imagic

If you now type ls , for example, the directory command is executed in the directory named /usr/bin/imagic

b) telnet

`telnet` command to reach another computer (if available).

How to use IMAGIC-5 as
an MS Windows console application

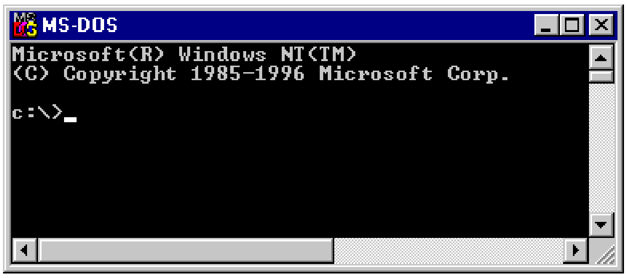
This brief introduction will give you some information on how to use IMAGIC-5 in an MS Windows environment. However, it is not the purpose of this hands on to train you to become a system manager of your computer.

1. Start IMAGIC-5

Currently, IMAGIC-5 is only running as a console application in an MS-DOS window. You must therefore first open such a MSDOS window. Use the corresponding icon

or start a Command Prompt using Start / Programs / Command Prompt

You will get an MS-DOS window like this:



```
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

c:\>_
```

Then change to your working directory:

```
c:\> cd c:\<my directory>
```

you specify <my directory>

To start IMAGIC-5 type:

```
c:\> c:\imagic\imagic
```

IMAGIC-5 will respond

```
** IMAGIC-5 welcomes you **

IMAGIC-COMMAND:
```

and is waiting for your command(s).

2. Some helpful hints when working in an MS-DOS window

First of all, it is a good idea to activate the so called DOSKEY feature which is available for most systems. When you enter and activate the DOSKEY command, you will later be able to use the arrow keys for scrolling through the list of commands you have entered before.

If you are running Windows NT, another helpful feature is the number of buffered lines of your MS-DOS console window. Use the right mouse button and click into the title bar of the MS-DOS window to activate the properties window. Please check out more details yourself.

3. LOGOUT

Quit IMAGIC-5

```
IMAGIC-COMMAND: ✶
```

and type

```
c:\> exit
```

to close the MS-DOS window.

4. WILD CARDS: ✶ AND %

These are the so-called "wild cards". With many commands you can reach not only one file or one directory. With the wild cards you can work with several files that have similar names. With ✶ you can call a string with any given length, while `%` is the wild card for only one sign.

Examples

`A✶.✶`	all files with any extension beginning with A
`✶junk✶.f`	all files with the extension `.f` and a name containing `junk`
`✶.%`	all files with an extension of one sign

5. MS-DOS COMMANDS

a) HELP

If you want a listing of all available commands type

```
help
```

or get help for a specific command

```
help <command>
```

where `command` is the name of the command you need help for.

Example: help type).

`help help`	help on "type"
`help type`	help on command "type"

b) HAVE A "LOOK" INTO DIRECTORIES

The command is:

```
dir
```

which shows a list of all files of the default directory in alphabetic order. Using wild cards (✶ or `%`) you can search for particular files, in which you are interested at the moment.

Examples:

`dir *.img`	display all files with the extension `img`
`dir c:\imagic\test/*.hed`	display all files with the extension `hed` n the directory `\c:\imagic\test`

There are many qualifiers with you can use to get more information:

`dir -p`	pauses after each screenful of information
`dir -W`	Uses wide list format
	more …

You can also use any possible combination, like:

`dir /w /L*`	all files in wide list format and lowercase

c) DELETE

Delete files:

To delete files use command

```
del
```

Examples:

`del *.img`	delete all files with the extension .img in the default directory
`del *.hed`	delete all files with extension .hed
`del /home/a*.*`	delete all files beginning with a in the directory named /home
`del *.*`	ATTENTION: Delete all files !!

Useful qualifiers are, for example:

`del /P`	prompts for confirmation before deleting
`del /F`	force deleting read-only files
`del /S`	delete all specified files from directory and all subdirectories
	more …

Delete directories/subdirectories:

The related command is

```
        rmdir
```

d) HANDLE WITH FILES

Copy files

```
copy
```

copies the input file to the output one (without deleting the input)

Examples:

```
copy  myimage. * \test\ourimage. *
```

copies all files named myimage (with any extension) in the default directory to the directory named test with the new names ourimage . The extensions remain the same.

```
copy  *.* \tmp\*.*
```

copies all files in the default directory to the directory named tmp without changing names and their extensions.

Move / rename files

```
rename
```

changes (moves) the input file name to the new specified one.

Examples:

```
rename  text.lis  yourtext.txt
```

the file called text.lis gets the new name yourtext.txt .

```
rename  text.txt \tmp\mytext.txt
```

the files named text.txt are moved from the current directory to the directory tmp with the new name mytext.txt.

Differences between files

```
comp
```

compares two files. Call help comp to get all qualifiers.

Search in files

`find` searches a string of signs in the specified files.

Example:

`find "HELP" *.f` searches the string HELP in all files with the extension .f .

There are some useful qualifiers:

`find /V` all lines are printed that do NOT contain the searched pattern

`find /N` the line number is printed with the displayed line

more …

Create (sub)directories

`mkdir` create a new directory

e) TYPE TEXT FILES

The related command is

`type`

Example:

`type mytext.txt` the content of the text file mx_text.txt is printed on the monitor

f) RUN JOBS OR PROGRAMS

Just type the command or program name.

g) COMMANDS TO GET SYSTEM INFORMATION

Show directory

`chdir` shows the name default directory you are working in.

h) COMMANDS FOR DIVERSE SYSTEM SETTINGS

Change directory

`cd` command to set the default directory you want to work in.

Example:

```
cd c:\imagic\test
```

the default directory is set to c:\imagic\test

Telnet

`telnet` command to reach another computer (if available).