INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION
INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
I.C.T.P., P.O. BOX 586, 34100 TRIESTE, ITALY, CABLE: CENTRATOM TRIESTE

H4-SMR 471/26

COLLEGE ON MEDICAL PHYSICS

10 - 28 SEPTEMBER 1990

Image Processing Hardware and Practices using a
PIP (Programmable Image Processing) system
&
Use of the Matrox PIP Card for Quality Control Measurements
of Digital Imaging Modalities

I. Pinto
CEVAB
Area di Ricerca
Trieste

&
P. E. Cruvinel
EMBRAPA/NPDIA
Sao Carlos
Brazil

USE OF THE MATROX PIP CARD FOR QUALITY CONTROL MEASUREMENTS OF DIGITAL IMAGING MODALITIES

UNIFORMITY MEASUREMENTS

SPATIAL UNIFORMITY DENOTES THE CAPABILITY OF AN IMAGING SYSTEM TO GIVE THE SAME PIXEL INTENSITY VALUE OF AN OBJECT IRRESPECTIVE OF THE POSITION OF THE OBJECT WITHIN THE SCAN PLANE

HOW TO TRANSFER DIGITAL IMAGES ONTO AN INDEPENDENT IMAGE PROCESSING SYSTEM (WORKSTATION)

1) MAGNETIC TAPE

PROBLEMS: DATA FORMAT

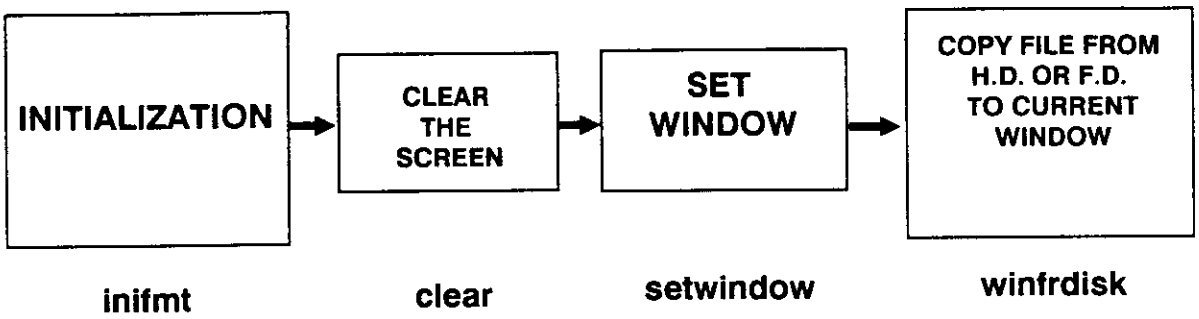
2) VIDEO OUTPUT OF DISPLAY CONSOLE

PROBLEMS: 8 BIT A/D CONVERTER

3) SERIAL INTERFACE (i.e. RS232 etc.)

PROBLEMS: SYSTEM ARCHITECTURE KNOWLEDGE, FORMAT DATA

READ UNIFORMITY TEST OBJECT IMAGE



EXC.1

UNIFORMITY TEST OBJECTS:

CT SCANNERS: WATER PHANTOM

GAMMA CAMERAS: UNIFORM RADIOACTIVITY

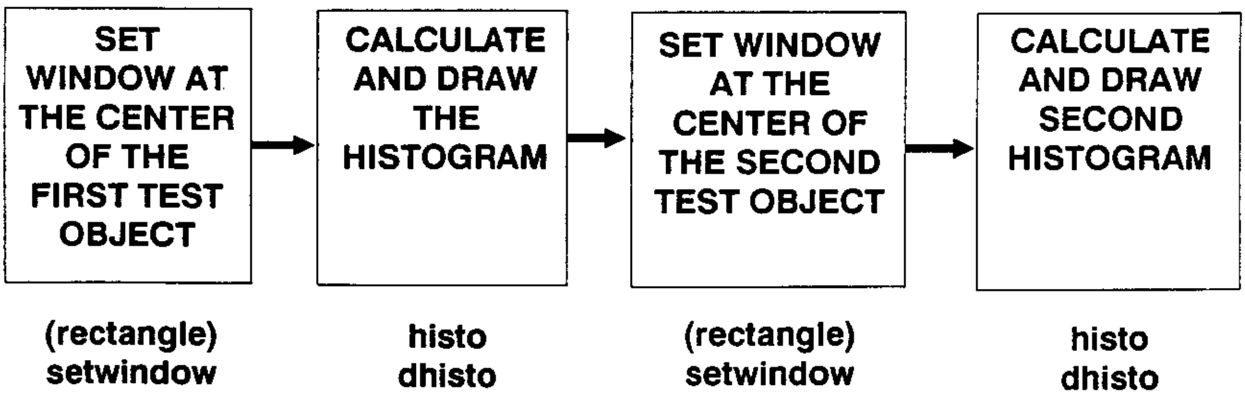
SLAB (FLOOD FIELD PHANTOMS)

MR SCANNERS: PARAMAGNETIC IONS

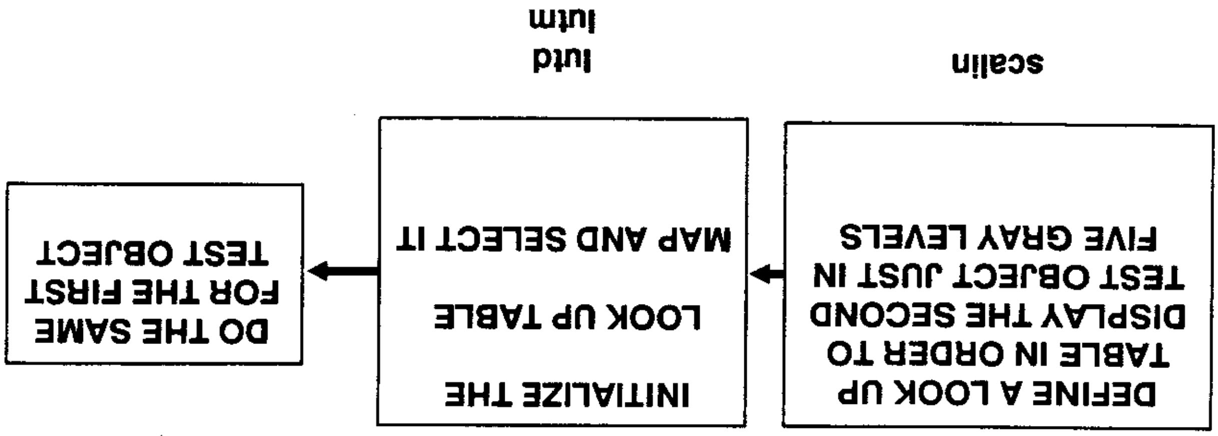
SOLUTIONS (i.e. CuSO4) GIVING UNIFORM MR

SIGNAL ON THE OVERALL FOV

HISTOGRAMS (UNIFORMITY TEST OBJECTS)



UNIFORMITY LOOK UP TABLES



UNIFORMITY INDEXES

INTEGERAL UNIFORMITY:

$$\frac{L_{max} - L_{min}}{L_{max} + L_{min}} \times 100$$

UNIFORMITY MAP

M = Modal value = Gray level corresponding to histogram peak

L1: $0 \leq I \leq (M - 0.2\,M)$

L2: $(M - 0.2\,M) \leq I \leq (M - 0.1\,M)$

L3: $(M - 0.1\,M) \leq I \leq M$

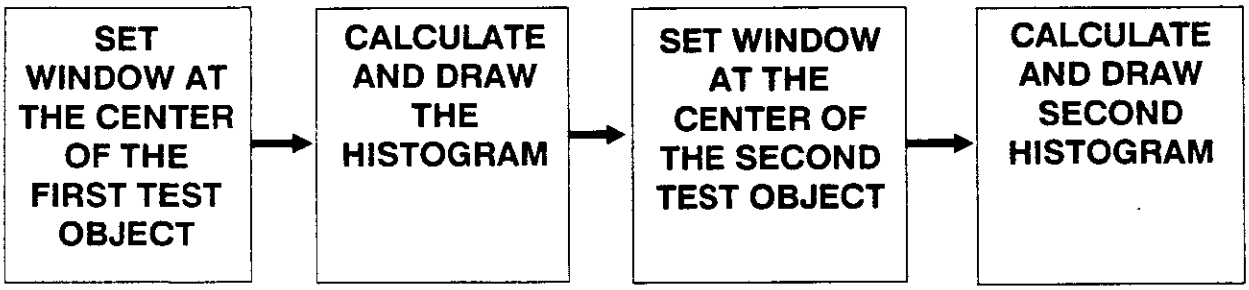
L4: $M \leq I \leq (M + 0.1\,M)$

L5: $(M + 0.1\,M) \leq I \leq (M + 0.2\,M)$

READ UNIFORMITY TEST OBJECT IMAGE

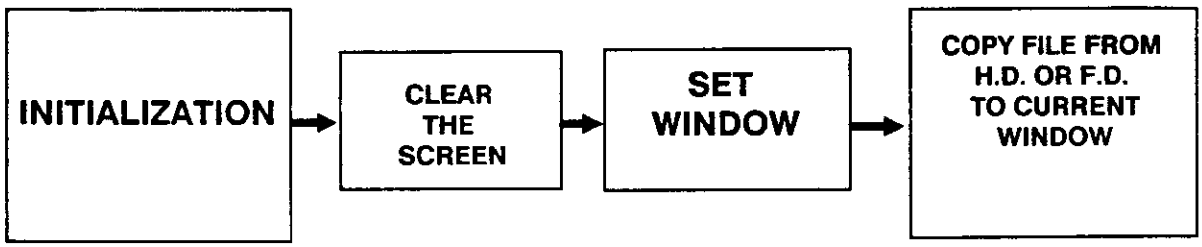
```
inifmt 26c 1 1 0 1 0

clear 0 7

setwindow 0 0 511 511

pause

winfrdisk 4096 imma.1 506a -1
```

HISTOGRAMS (UNIFORMITY TEST OBJECTS)



EXC.2

READ UNIFORMITY TEST OBJECT IMAGE



EXC.1

UNIFORMITY INDEXES

INTEGERAL UNIFORMITY:

$$\frac{L_{max} - L_{min}}{L_{max} + L_{min}} \times 100$$

UNIFORMITY MAP

M = Modal value = Gray level corresponding to histogram peak

L1: $0 < I < (M - 0.2\,M)$

L2: $(M - 0.2\,M) < I < (M - 0.1\,M)$

L3: $(M - 0.1\,M) < I < M$

L4: $M < I < (M + 0.1\,M)$

L5: $(M + 0.1\,M) < I < (M + 0.2\,M)$

HISTOGRAMS (UNIFORMITY TEST OBJECT)

rectangle 75 70 185 180

setwindow 76 71 184 179

pause

histo 506a

pause

dhisto 861 0 500 200 255 255 506a

pause

setind 0

rectangle 337 97 447 212

setwindow 338 98 446 211

pause

histo 506a

pause

dhisto 861 257 500 200 255 255 506a

```
exc.1  READ UNIFORMITY TEST OBJECT IMAGE

inifmt 26c 1 1 0 1 0
clear 0 7
setwindow 0 0 511 511
pause
winfrdisk 4096 imma.1 506a -1

exc.2 HISTOGRAMS (UNIFORMITY TEST OBJECTS)


rectangle 75 70 185 180
setwindow 76 71 184 179
pause
histo 506a
pause
dhisto 861 0 500 200 255 255 506a
pause
setind 0
rectangle 337 97 447 212
setwindow 338 98 446 211
pause
histo 506a
pause
dhisto 861 257 500 200 255 255 506a


exc.4

UNIFORMITY LOOK UP TABLES

scalin 0 0 129 0 506a
scalin 130 50 144 50 506a
scalin 145 120 158 120 506a
scalin 159 190 172 190 506a
scalin 172 255 255 255 506a
lutd 4 1 0 255 506a
lutd 4 2 0 255 506a
lutd 4 3 0 255 506a
lutm 4
pause
lutm 0
pause
scalin 0 0 187 0 506a
scalin 188 50 208 50 506a
scalin 209 120 228 120 506a
scalin 229 190 248 190 506a
scalin 249 255 255 255 506a
lutd 5 1 0 255 506a
lutd 5 2 0 255 506a
lutd 5 3 0 255 506a
lutm 5
scalin 0 0 130 0 506a
scalin 131 200 144 200 506a
scalin 145 0 172 0 506a
```



```
scalin 173 255 255 255 506a
pause
lutd 6 1 0 255 506a
pause
scalin 0 0 144 0 506a
scalin 145 200 158 200 506a
scalin 159 0 172 0 506a
scalin 173 255 255 255 506a
pause
lutd 6 2 0 255 506a
pause
scalin 0 0 158 0 506a
scalin 159 200 172 200 506a
scalin 173 255 255 255 506a
lutd 6 3 0 255 506a
pause
lutm 6
```

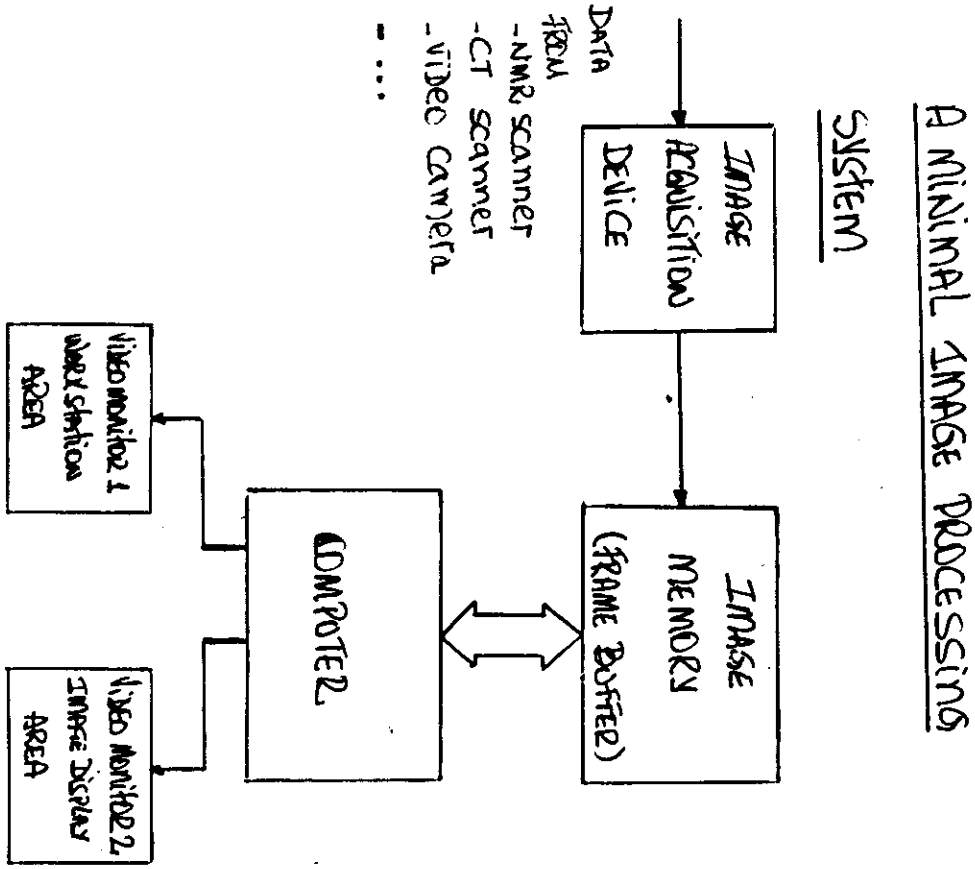
UNIFORMITY MAP

```
scalin 0 0 129 0 506a
scalin 130 50 144 50 506a
scalin 145 120 158 120 506a
scalin 159 190 172 190 506a
scalin 172 255 255 255 506a
lutd 4 1 0 255 506a
lutd 4 2 0 255 506a
lutd 4 3 0 255 506a
lutm 4
pause
lutm 0
pause
scalin 0 0 187 0 506a
scalin 188 50 208 50 506a
scalin 209 120 228 120 506a
scalin 229 190 248 190 506a
scalin 249 255 255 255 506a
lutd 5 1 0 255 506a
lutd 5 2 0 255 506a
lutd 5 3 0 255 506a
lutm 5
scalin 0 0 130 0 506a
scalin 131 200 144 200 506a
scalin 145 0 172 0 506a
scalin 173 255 255 255 506a
pause
lutd 6 1 0 255 506a
pause
```

PRACTICES

PRACTICE	MACRO NAME
1.	pgb
2	pga
3	pr (pra, prb, prs)
4	prc
5	prd (prrd)
6	pgca, pgc
7	pca lowpass
	pcd highpass
	pcg laplacian
	pch sobel
	pcj vertical/horizontal
	pce average
	pcf median
8	pcl
9	pda
10	pdb

lectures notes
(44-SNR-471/08)



(1.)

PIP 640-B PLUG IN VIDEO FRAME GRABBER-DIGITIZER BOARD

RESOLUTION - 640X512 PIXELS

POWER CONSUPTION - 17W

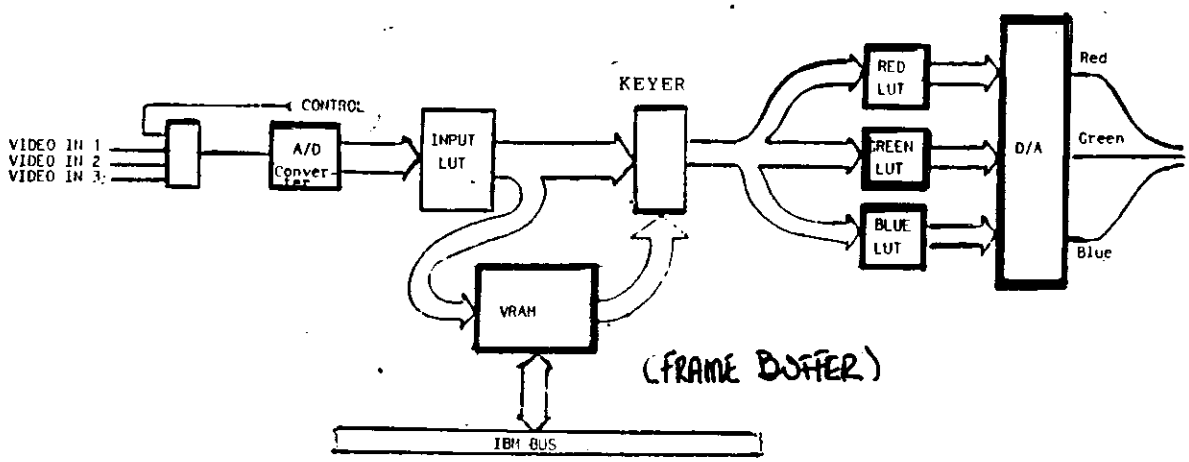


Figure 1 - The operation of PIP.640-B

(2)

- BOTH THE SYSTEM UNIT AND CRT-CONTROLLER HAVE SIMULTANEOUS TRANSPARENT ACCESS TO THE FRAME BUFFER
- 1024x1024 STORAGE AREA AND 640x512 DISPLAY AREA

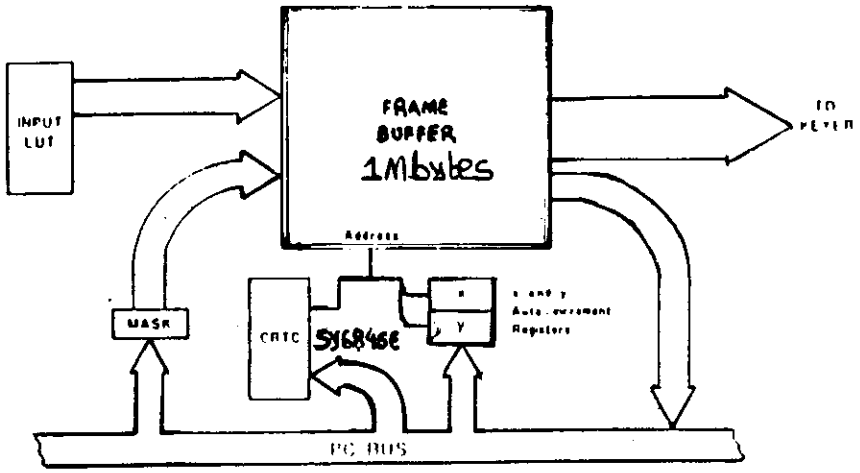
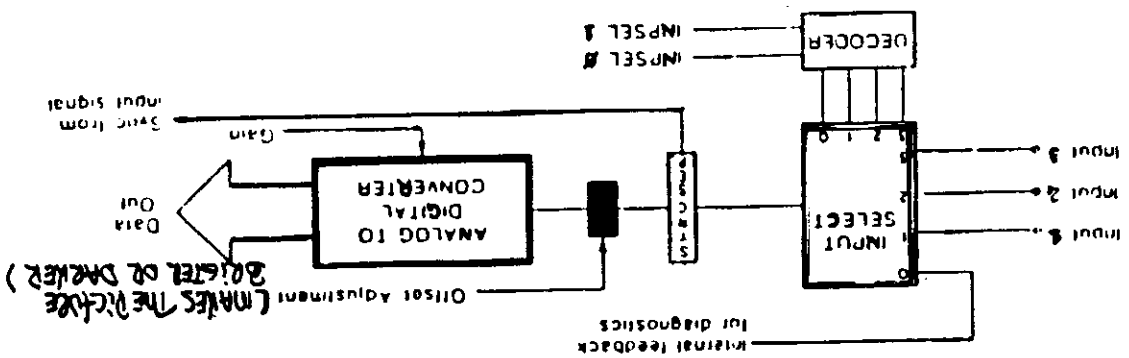


Figure 3 - Frame buffer data access in block diagram

(4)

(5)

Figure 2 - Input section



— THE VIDEO INPUT IS DIGITIZED IN REAL TIME
THREE INPUT PORTS USING SIGNALLYS SYNC.
— INPUT SIGNALS ARE SELECTED IN SOFTWARE FROM ONE OF

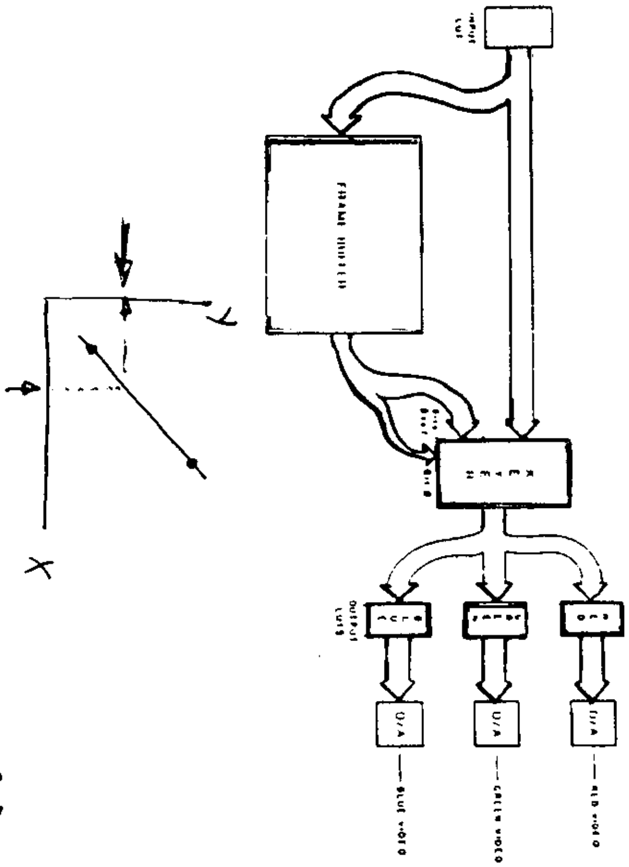


Figure 4 - Keying and output section

- Two sources of data for output { Frame buffer, Input LUT }
- It has one input and three output LUT (one channel x 3)

Image processing includes:

- Enhancement
- Restoration
- Measurement of Image Elements
- Classification of Image Elements
- Recognition

In these practices a set of tools will be used for image processing such as:

- Point Processing
- Area Processing
- Geometric Processing
- Frame Processing

RACTICE 1

HOW IS IT POSSIBLE TO INITIALIZE THE PIP BOARD IN THE SYSTEM UNIT TO A KNOWN STATE AS WELL AS TO INTERFACE A CAMERA AND SAVE THE IMAGE, FROM THE FRAME BUFFER ON DISK ?

OMMAND DESCRIPTIONS:

inif(mt) base_addr, mode, speed, class, vid_type, zoom
This COMMAND INITIALIZES the PIP BOARD IN the SYSTEM UNIT to A KNOWN STATE

base_addr: It SPECIFIES THE OFFSET OF THE I/O ADDRESS USED.

mode: It tells the SOFTWARE WHETHER THE PIP CARD IS STRAPPED to ALLOW FOR ZOOM RESOLUTION OR NOT
ZERO: COMPATIBILITY
NON ZERO: ZOOM RESOLUTION ALLOWED

speed: It SPECIFIES THE CLOCK SPEED.
ZERO: PIP512 OR PIP1024
NON ZERO: PIP640

class: It SPECIFIES THE VIDEO FORMAT

0	512X512	(European)
1	640X512	(European)
⋮	⋮	⋮

vid_type: It SPECIFIES WHETHER THE VIDEO FORMAT IS TO BE A 50 Hz OR 60 Hz
ZERO: AMERICAN STANDARD VIDEO
NON ZERO: EUROPEAN

ZOOM: It SPECIFIES WHETHER THE FORMAT IS TO USE FULL RESOLUTION OR ZOOM RESOLUTION
ZERO: FULL RESOLUTION
NON ZERO: ZOOM RESOLUTION

setw(indow) x1 y1 x2 y2

This COMMAND SPECIFIES the LOWER LEFT AND UPPER RIGHT CORNERS OF the WINDOW

x1, x2 : THESE PARAMETERS INDICATE the X COORDINATES OF CORNERS 1 AND 2 OF the DISPLAY WINDOW

y1, y2 : THESE PARAMETERS INDICATE the Y COORDINATES OF CORNER 1 AND 2 OF the DISPLAY WINDOW

clear) smap umap

This COMMAND CLEARS the SCREEN by SETTING ONE OF the UNUSED MAPS

smap: It SELECTS ONE the B INPUT LUT MAPS TO BE THE ACTIVE MAP FOLLOWING the CLEAR OPERATION

umap: It SELECTS ONE OF the B MAPS OF THE INPUT LUT TO BE SET TO THE CURRENT DRAW INDEX.

pause

It IS USED to HALT the INTERPRETER' OPERATION UNTIL A KEY IS STRUCK

channel) channel

This COMMAND SELECTS the ACTIVE VIDEO INPUT PORT

channel : It SELECTS the INPUT CHAN
- 0 : SELECTS CHANNEL 0
- 1 : SELECTS CHANNEL 1
- 2 : SELECTS CHANNEL 2
- 3 : INTERNAL LOOPBACK

sync mode

This COMMAND IS USED to SELECT the SOURCE OF the SYNC SIGNAL

mode:
- 0 : THE INTERNAL SYNC IS SELECTE
- 1 : THE EXTERNAL SYNC IS SELECTE

sbuf) mode

This COMMAND IS USED to SELECT the VIDEO OUTPUT SOURCE

mode:

Ø: The OUTPUT OF The INPUT LUT IS DISPLAYED, KEYING IS DISABLED

1: The OUTPUT OF the FRAME BUFFER IS DISPLAYED. KEYING IS ALLOWED

int(odisk) [size <x>] file <s> off <x> seg <x>

This COMMAND IS USED to COPY the CONTENTS OF The CURRENT DISPLAY WINDOW to DISK

[size <x>]: It SPECIFIES the SIZE OF AN INTERMEDIATE BUFFER (The OPTIMAL BUFFER SIZE IS 4096 bytes)

file <s>: It PROVIDES A FILE NAME

off <x>: It gives the OFFSET OF the INTERMEDIATE BUFFER WITHIN SEGMENT

seg <x>: It SPECIFIES The SEGMENT that CONTAINS the INTERMEDIATE BUFFER (USE -1)

PRACTICE 1:

```
initmt 26c 1 1 Ø 1 Ø
setwindow Ø Ø 511 511
clear Ø 7
pause
channel 2
sync 1
sbuf Ø
pause
sbuf 1
pause
sbuf Ø
pause
sbuf 1
pause
snap 1
pause
wintodisk 4096 image.bin 506a -1
pause
```

The image was obtained from the dual energy CT technique. The object beneath the cross section is the bone mineral phantom with five calibration materials: fat equivalent material, water, 50 mg/cc, 100 mg/cc and 200 mg/cc K_2HPO_4 respectively and a high energy (140 kVp, 100 mA) scan through the abdominal region

(imag3.bin)

(To be used with practices from 2 to 10)

(11)

PRACTICE 2

How is it possible to copy a disk file to the current display window?

COMMAND DESCRIPTIONS:

Winf [rdisk] bsize<e> file<s> workbuffer<e> seg<

This command is used to copy a disk file to the current display window.

bsize<e>: It specifies the size of intermediate buffer that MS-DOS requires to make the transfer

file<s>: It provides a file name for the transfer.

workbuffer<e>: It gives the offset of the intermediate buffer within segment

seg<e>: It specifies the segment that contains the intermediate buffer (use -1)

<Fg>

ACTICE 2 : clear ø 7 pga

```
initmt 26c 1 1 ø 1 ø
setwindow ø ø 511 511
clear ø 7
pause.
winfrdisk 4096 imag3.bin 506a-1
```


PRACTICE 3 : MATRIX

HOW is it possible :

- to DECLARE AND STORE A MACRO?
- to EXECUTE A MACRO?
- to SET A PRE-DEFINED WINDOW?
- to READ AND WRITE THE VALUE OF A PIXEL IN THE FRAME BUFFER?

COMMAND DESCRIPTIONS:

a) to DECLARE AND STORE A MACRO :

macro macro_number

macro_number: This PARAMETER is the IDENTIFYING NUMBER of the MACRO AND its PURPOSE IS to PERMITE FUTURE IDENTIFICATION OF A PARTICULAR SEQUENCE OF COMMANDS

savemacro file_name<s> macro_number

This command is used to store a macro in a disk file

file_name<s>: The name of the file in which the macro will be stored.

macro_number: A number specifying the macro to be saved

To declare

```
<pipint> ... (call the interpreter)
macro 1  ... (from 0 through 19)
  {      }... (commands)
blank line ... (It is terminated by
 <CR>          entering a blank
               line followed by a
               carriage return)
```

To store

savemacro exp 1

b) To execute a macro:

getmacro file_name<s> macro_number

This command is used to transfer a macro stored in a file to the interpreter

file_name<s>: The name of the file which contains the macro

macro_number: A number specifying the macro to be transferred.

execute macro_number

It is used to execute a sequence of commands defined in a macro

macro_number: It specifies the macro to be executed

To execute

getmacro exp 1
execute 1

) TO SET A PRE-DEFINED WINDOW :

grid incx incy

It is USED to DRAW A GRID USING THE CURRENT INDEX INSIDE THE CURRENT WINDOW

incx, incy : THE NUMBER OF PIXELS BETWEEN THE LINES OF THE GRID IN THE X AND Y DIRECTION

bmread workbuffer <z> seg <z>

This COMMAND IS USED TO COPY THE CONTENTS OF THE CURRENT DISPLAY WINDOW TO A BUFFER

workbuffer <z> : THE ADDRESS OF THE BUFFER WHERE THE CONTENTS OF THE WINDOW WILL BE STORED

seg <z> : THE SEGMENT PORTION OF THE MEMORY BUFFER ADDRESS

(This COMMAND will RETURN THE NUMBER OF BYTES that HAVE BEEN TRANSFERRED)

putfile file_name <s> address <z> count

It is USED to COPY the contents OF A BUFFER LOCATED in the system's MEMORY into A DISK FILE

file_name <s> : The NAME OF THE FILE INTO which the CONTENTS OF THE BUFFER is TO BE COPIED.

address <z> : The ADDRESS OF the BUFFER TO BE COPIED.

COUNT : The NUMBER OF bytes TO BE TRANSFERRED

(This COMMAND RETURNS THE NUMBER OF BYTES TRANSFERRED TO THE FILE)

clr(memory) address <z> count

It is USED TO CLEAR to (00)H THE CONTENTS OF A PORTION OF THE INTERPRETER'S 16K BUFFER

address <z> : THE ADDRESS OF THE initial MEMORY LOCATION TO BE CLEARED

COUNT : It SPECIFIES HOW MANY MEMORY LOCATIONS WILL BE CLEARED STARTING AT address <z>

get(file) file_name<s> address<x>

It is USED TO COPY the contents OF A file into A SPECIFIED BUFFER

file_name<s>:
THE NAME OF THE File TO BE COPIED

address<x>:
THE ADDRESS OF THE BUFFER WHERE THE content OF THE File IS TO BE COPIED.

(This COMMAND will RETURN the NUMBER OF BYTES TRANSFERRED FROM the FILE.

DISP.WRITE) WorkBuffer<x> Seg<x>

It is USED TO COPY the contents OF A BUFFER TO THE CURRENT Display Window

WorkBuffer<x>: It contains the ADDRESS OF THE BUFFER CONTAINING THE DATA TO BE TRANSFERRED TO THE CURRENT DISPLAY WINDOW

Seg<x>: THE SEGMENT PORTION OF THE MEMORY BUFFER ADDRESS.

d) TO READ AND WRITE THE VALUE OF A PIXEL IN THE FRAME BUFFER.

pixread) x y

It is USED TO READ THE VALUE OF A PIXEL IN the FRAME BUFFER.

x,y: COORDINATES OF THE PIXEL TO BE READ

pixwrite) x y value

It is USED TO WRITE A PIXEL TO THE FRAME BUFFER

x,y: COORDINATES OF the PIXEL TO BE WRITTEN

value: VALUE TO BE WRITTEN TO THE FRAME BUFFER (FROM Ø TO 255, A DECIMAL VALUE)

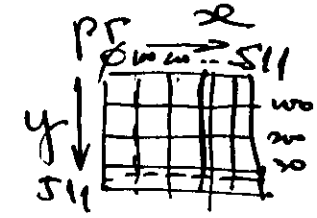
ACTICE 3:

```
initmt 26c 1 1 0 1 0
setwindow 0 0 511 511
winfrdisk 4096 imag3.bin 506a -1
Pause
grid 100 100
setwindow 200 200 300 300
winread 506a FFFF
putfile wind1.bin 506a 10201
Pause
clear 0 7
Pause
cmemory 506a 10201
getfile wind1.bin 506a
Pause
winwrite 506a FFFF
Pause
grid 10 10
Pause
pixread 201 201
Pause
Pixread 250 270
Pause,
Pix write 250 270 0
    ⋮
Pix write 260 280 0
```

pr 1

pr

<F9>



$V = 2^m$

m ←

$2^8 = 256$

PRACTICE 4

HOW is it POSSIBLE TO CALCULATE A HISTOGRAM OF THE GREY LEVEL IN the CURRENT WINDOW?

COMMAND DESCRIPTIONS:

histogram) buffer <x>

It is USED TO CALCULATE A histogRAM OF THE GREY LEVEL IN THE CURRENT WINDOW

buffer <x>: It SPECIFIES THE ARRAY OF 256 ELEMENTS which IS USED to store the histogRAM VALUE

dh(isto) max <x> x y scale gad tad but <x>

It is USED to dRAW A histogRAM IN the FRAME BUFFER

max <x> : It specifys the count of the most frequently occuring pixel value in the histogram.

x, y : They are used to specifie the x and y position of the histogram.

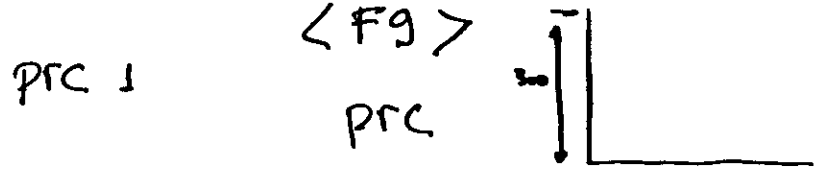
scale : This parameter specifies the height in pixels of the highest bar of the histogram all other values are drawn relative to this height

gval, tval : The index (color) to be used to draw the bar and the labels of the histogram.

buf <x> : Address of the buffer where the histogram created by the histo command is stored.

PRACTICE 4

```
inifmt 26c 1 1 0 1 0
Set window 0 0 511 511
winfrdisk 4096 imag3.bin 506a -1
Pause
histo 506a
Pause
dhisto 6aca 40 400 300 255 255 5
Pause
dhisto 6aca 40 400 300 0 255 50
Pause
dhisto 6aca 40 400 300 255 255 506
Pause
clear 0 7
Pause
winfrdisk 4096 imag3.bin 506a -1
```



PRACTICE 5

HOW IS IT POSSIBLE TO CALCULATE A HISTOGRAM OF THE GREY LEVEL USING DIFFERENTS WINDOWS AT THE SAME IMAGE ?

COMMAND DESCRIPTIONS:

rectangle x1 y1 x2 y2

THIS COMMAND IS USED TO DRAW A RECTANGLE HAVING (X1.Y1) AND (X2.Y2) AS OPPOSITE CORNERS

x1, x2 :

THEY SPECIFIE THE FIRST AND THE SECOND X COORDINATE

y1,y2 :

THEY SPECIFIE THE FIRST AND THE SECOND Y COORDINATE

ALWAYS IT WILL BE NECESSARY, FIRST, TO USE THE setwindow COMMAND.

"

PRACTICE 5 :

```
initmt 26c 1 1 0 1 0
clear 0 7
setwindow 0 0 511 511
Pause
winfrdisk 4096 image3.bin 506a -1
Pause
rectangle 100 100 200 200
setwindow 100 100 200 200
Pause
histo 506a
Pause
dhisto 21d 50 400 300 255 255 506a
Pause                                          <F9>
setwindow 0 0 511 511
Pause                                          prd
clear 0 7
Pause
winfrdisk 4096 image3.bin 506a -1              prd
Pause
rectangle 350 100 430 200
setwindow 330 100 430 200
Pause
dhisto 1b8 250 456 300 255 255 506a
Pause
```


"

```
setwindow 0 0 511 511
clear 0 7
Pause
windrdisk 4096 imag3.bin 506a -1
Pause
rectangle 200 250 300 350
setwindow 200 250 300 350
Pause
histo 506a
Pause
dhisto 190 250 480 300 255 255 506a
```


PRACTICE 6

HOW IS IT POSSIBLE TO INITIALIZE A LOOKUP TABLE MAP (THE USE OF PSEUDOCOLOR) AS WELL AS HOW IS IT POSSIBLE TO COMBINE DIFFERENT FUNCTIONS DRIVES (R, G, B)?

COMMAND DESCRIPTIONS:

6.1 - TO INITIALIZE A LOOKUP TABLE.

scaling x1 y1 x2 y2 buffer(x)

THE FUNCTION scaling MAPS THE VALUES OF x_n onto y_n WITH x AS THE INDEX TO THE LUT AND y AS THE MAPPED VALUE. ON THE BASIS OF THIS COMMAND IT IS POSSIBLE TO EMPHASIZE ONE PART OF AN IMAGE AT THE EXPENSE OF ANOTHER.

x1, x2: THE FIRST AND THE LAST ADDRESS IN BUFFER TO BE SCALED

y1, y2: THE LOW AND THE UPPER END OF THE SCALE TO BE STORED AT BUF[x1] AND BUF[x2] RESPECTIVELY.

buffer(x): IT IS THE ADDRESS OF THE BUFFER WHERE THE RESULT OF THE SCALING FUNCTION IS TO BE STORED.

lutd (define) map color start length buffer <e>

This COMMAND INITIALIZES A LOOKUP TABLE MAP.

map: It SPECIFIES which OF THE 8 MAPS OF ANY ONE PARTICULAR LUT IS TO BE LOAD with the DATA.

color: It SPECIFIES THE LOOKUP TABLE which will BE AFFECTED BY THE CHANGE OF DATA
- Ø → INPUT LUT
- 1 → BLUE LUT
- 2 → GREEN LUT
- 3 → RED LUT
- 4 → ALL OUTPUT LUT

start: It INDICATES which OF the 256 bytes OF THE SELECTED MAP will SERVE AS THE STARTING Point TO INICIALIZATION

length: It INDICATES HOW MANY BYTES will be REWRITTEN USING the scaling FUNCTION (IT MUST BE BETWEEN 1 AND 256 inclusive)

Buffer <e>: LOCATION OF the BUFFER containing the VALUES OF the initialization.

PRACTICE 6.1 :

```
initmt 26c 1 1 Ø 1 Ø
setwindow Ø Ø 511 511
clear Ø 7
Pause
scaling Ø 255 255 Ø 506a
Pause
lutd Ø 1 Ø 256 506a
Pause
initmt 26c 1 1 Ø 1 Ø
Pause
clear Ø 7
Pause
winfrdisk 4096 img3.bin 506a-1
pause
scaling Ø 255 255 Ø 506a
pause
lutd Ø 2 Ø 256 506a
Pause
clear Ø 7
Pause
winfrdisk 4096 img3.bin 506a-1
pause
scaling Ø 255 255 Ø 506a
Pause
lutd Ø 3 Ø 256 506a
Pause
```

Pgca

PRACTICE 6.2 : TO COMBINE DIFFERENT FUNCTIONS DRIVES AND TO RESTORE THE ORIGINAL IMAGE.

```
initmt 26c 1 1 0 1 0
setwindow 0 0 511 511
clear 0 7
Pause
winrdisk 4096 imag3.bin 506a -1
Pause
scaling 0 255 255 0 506a
Pause
lutd 0 1 0 256 506a
pause
lutd 0 2 0 256 506a
Pause
lutd 0 3 0 256 506a
Pause
scaling 0 0 255 255 506a
Pause
lutd 0 1 0 256 506a
Pause
lutd 0 2 0 256 506a
Pause
lutd 0 3 0 256 506a
```

Pgc

Pgc

(32)

This PROGAME will initialize MAP 0 OF THE BLUE LUT STARTING AT BYTE 0 GOING UP TO 255.
IN A SIMILAR WAY MAP 0 BECOMES NOW THE GREEN LUT → CYAN (BLUE AND GREEN TOGETHER)
IN A SIMILAR WAY MAP 0 BECOMES NOW THE RED LUT → THE IMAGE OF THE MONITOR IS NOW THE RESULT OF THE THREE OUTPUT SIGNALS

→ THE VALUE IN THE LUT DETERMINE THE COLOR INTENSITY

→ THE DIFERENCE BETWEEN THE THREE VALUES CONTROLS THE COLOR

A whitepixel (255) is the sum of:
one blue (255) + one green (255) + one red (255)

A blackpixel (0) is the sum of:
one blue (0) + one green (0) + one red (0)

→ TO RESTORE THE ORIGINAL IMAGE IT WAS USED A INVERSE SCALING FUNCTION AND ALL OTHERS THREE LUTs.

(3

PRACTICE 7

HOW IS IT POSSIBLE TO CONVOLVE THE IMAGE IN THE CURRENT WINDOW WITH A PRE-DEFINED KERNEL?

SPATIAL FILTERING IS TYPICALLY USED FOR EDGE ENHANCEMENT OR NOISE REDUCTION PRIOR TO EDGE OR OBJECT DETECTION

COMMAND DESCRIPTIONS:

ker3 k1 k2 k3 k4 k5 k6 k7 k8 k9 buffer<z>

THIS COMMAND IS USED TO GENERATE THE 3x3 CONVOLUTION KERNEL

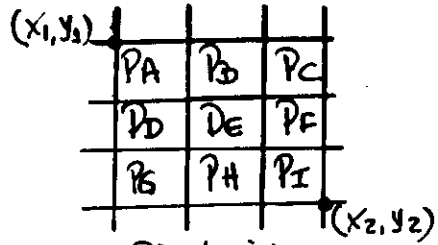
k1...k9 : VALUES WITHIN THE KERNEL AS BELOW

$$k = \begin{bmatrix} k1 & k2 & k3 \\ k4 & k5 & k6 \\ k7 & k8 & k9 \end{bmatrix}$$

buffer<z> : THE STORAGE LOCATION FOR THE KERNEL

con(3) source dest buf3<z>

IT CONVOLVES THE IMAGE IN THE CURRENT WINDOW IN THE SOURCE WORKSPACE AND COPIES THE RESULTS TO AN IDENTICAL WINDOW IN THE DESTINATION WORKSPACE. IT USES A KERNEL PREVIOUSLY STORED IN A PREDEFINED BUFFER



PIXELS IN CURRENT WINDOW

$$k = \begin{bmatrix} k1 & k2 & k3 \\ k4 & k5 & k6 \\ k7 & k8 & k9 \end{bmatrix}$$

$$PE = \frac{k1 PA + k2 PB + \cdots + k9 PI}{(k1 + k2 + \cdots + k9)}$$

IF $(k1 + \ldots + k9) = 0$ con3 TAKES THE ABSOLUTE VALUE OF THE RESULT AND TRUNCATES IT TO 255.

source : WORKSPACE IN WHICH THE CONVOLUTION IS TO BE PERFORMED.

dest : WORKSPACE INTO WHICH THE CONVOLUTED IMAGE WILL BE COPIED.

buf3<z>: THE LOCATION WHERE THE KERNEL IS STORED.

7.1 LOWPASS:

$$R = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

- LOW PASS FILTERING OF AN IMAGE PRODUCES AN OUTPUT IMAGE IN WHICH HIGH SPATIAL FREQUENCY COMPONENTS HAVE BEEN ATTENUATED.

- IT IS OFTEN USED AS A SMOOTHING OPERATION TO REMOVE VISUAL NOISE

```
inifmt 26c 1 1 0 1 0
setwindow 0 0 511 511
clear 0 7
Pause
winfrdisk 1096 imag3.bin 506a -1
Pause
rectangle 100 100 400 400
setwindow 100 100 400 400
Ker3 1/16 2/16 1/16 2/16 4/16 2/16 1/16 2/16 1/16 5200
Pause
con3 0 0 5200
setwindow 0 0 511 511
```


7.2 High Pass:

$$R = \frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- High Pass FILTERING OF A IMAGE PRODUCES AN OUT IMAGE IN WHICH high SPATIAL FREQUENCY COMPO ARE ACCENTUATED.

- It is OFTEN USED IN the ENHANCEMENT OF EDGES

```
inifmt 26c 1 1 0 1 0
setwindow 0 0 511 511
clear 0 7
Pause
winfrdisk 1096 imag3.bin 506a -1
Pause
Ker3 1/10 1/10 1/10 1/10 2/10 1/10 1/10 1/10 1/10 :
Pause
rectangle 100 100 400 300
setwindow 100 100 400 300
Pause
con3 0 0 5200
```

3 LAPLACIAN

cp1 source dest cp2 source dest

$k_{p1} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ $k_{p2} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$ $k = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

- LAPLACIAN EDGE ENHANCEMENT PRODUCES AN OUTPUT IMAGE IN WHICH HIGH SPATIAL FREQUENCY COMPONENTS ARE HIGHLY ACCENTUATED AND LOW SPATIAL FREQUENCY COMPONENTS ARE SHARPLY ATTENUATED.
- USEFUL FOR EXTRACTION OF OBJECT EDGES OR BOUNDARIES.

```
imt 26c 1 1 0 1 0
window 0 0 511 511
ear 0 7
use
nfrdisk 4096 imag3.bin 506a 1
use
ctangle 100 100 400 300
window 100 100 400 300
use
1 0 0
use
window 0 0 511 511
infrdisk 4096 imag3.bin 506a -1
use
ctangle 100 100 400 300
t window 100 100 400 300
use
p2 0 0
ause
window 0 0 511 511
infrdisk 4096 imag3.bin506a-1
use
53 -1 -1 -1 -1 3 -1 -1 -1 -1 5200
```


7.1 SOBEL

sobel source dest

This command performs a 3x3 Sobel filter operation on the current window. The Sobel filter compares the results of two convolutions to estimate the strength and orientation of edges in the image.

$$\text{STRENGTH} = \sqrt{X*X + Y*Y}$$

$$\text{ORIENTATION} = \arctan(Y/X)$$

```
initmt 26c 1 1 0 1 0
 setwindow 0 0 511 511
clear 0 7
 Pause
winfrdisk 4096 imag3.bin 506a-1
 Pause
rectangle 100 100 400 300
 setwindow 100 100 400 300
 Pause
 Sobel 0 0
```

7.5 HORIZONTAL AND VERTICAL EDGE DETECTION.

horfilter) source dest verfilter) source dest

THESE COMMANDS PERFORM A HORIZONTAL AND A VERTICAL EDGE DETECTION TRANSFORMATION ON THE CURRENT WINDOW.

$$K_H = \begin{bmatrix} -2 & -2 & -2 \\ 0 & 0 & 0 \\ 2 & 2 & 2 \end{bmatrix} \qquad K_V = \begin{bmatrix} -2 & 0 & 2 \\ -2 & 0 & 2 \\ -2 & 0 & 2 \end{bmatrix}$$

```
initmt 26c 1 1 0 1 0      — HORIZONTAL EDGE ENHANCEMENT
setwindow 0 0 511 511     — VERTICAL EDGE ENHANCEMENT
clear 0 7
pause
winfrdisk 4096 imag3.bin 506a -1
pause
rectangle 100 100 400 300
setwindow 100 100 400 300
pause
horfilter 0 0
pause
setwindow 0 0 511 511
winfrdisk 4096 imag3.bin 506a -1
pause
rectangle 100 100 400 300
setwindow 100 100 400 300
pause
verfilter 0 0
```


7.6 AVERAGE

average source dest

THIS COMMAND PERFORMS A PIXEL AVERAGING TRANSFORMATION ON THE CURRENT WINDOW.

$$K = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

```
initmt 26c 1 1 0 1 0
setwindow 0 0 511 511
clear 0 7
pause
winfrdisk 4096 imag3.bin 506a -1
pause
rectangle 100 100 400 300
setwindow 100 100 400 300
pause
average 0 0
```

MEDIAN

median) source dest

This COMMAND PERFORMS A 3x3 MEDIAN FILTER OPERATION ON the CURRENT WINDOW.

```
initmt 26c 1 1 0 1 0
setwindow 0 0 511 511
clear 0 7
Pause
wintrdisk 4096 imag3.bin 506a-1
Pause
rectangle 100 100 400 300
Set window 100 100 400 300
Pause
median 0 0
```

PRACTICE 8

How is it POSSIBLE TO USE At the SAME TIME Point AND AREA PROCESS ALGORITHMS ?

PCL

PRACTICE 8 :

```
initmt 26c 1101 0
setwindow 0 0 511 511
clear 0 7
Pause
Winfrdisk 4096 imag3.bin 506a -1
Pause
scaling 0 100 100 0 506a
lutd 0 1 0 266 506a
Pause
rectangle 200 200 300 300
setwindow 200 200 300 300
pause
histo 506a
Pause
dhisto 190 100 200 200 255 255 506a
Pause
ker3 1/16 2/16 1/16 2/16 4/16 2/16 1/16 2/16 1/16 5200
Pause
con3 0 0 5200
Pause
histo 506a
```



```
histo 506a
Pause
dhisto 190 100 450 200 255 255 506a
Pause
setwindow 0 0 511 511
clear 0 7
Pause
Winfrdisk 4096 imag3.bin 506a -1
Pause
rectangle 200 200 300 300
setwindow 200 200 300 300
Pause
histo 506a
Pause
dhisto 190 100 200 200 255 255 506a
Pause
dhisto 190 100 200 200 255 255 506a
Pause
ker3 0 -1 0 -1 5 -1 0 -1 0 5200
Pause
histo 506a
Pause
```

```
con3 0 0 5200
Pause
histo 506a
Pause
dhisto 318 100 450 200 255 255 506a
Pause
setwindow 0 0 511 511
clear 0 7
Pause
setwindow 0 0 511 511
clear 0 7
Pause
winfrdisk 4096 imag3.bin 506a -1
Pause
rectangle 200 200 300 300
setwindow 200 200 300 300
Pause
histo 506a
Pause
dhisto 196 100 200 200 255 255 506a
Pause
average 0 0
```



```
Pause
dhisto 19d 100 450 200 255 255 506a
Pause
setwindow 0 0 511 511
clear 0 7
Pause
winfrdisk 4096 imag3.bin 506a -1
Pause
rectangle 200 200 300 300
setwindow 200 200 300 300
Pause
histo 506a
Pause
dhisto 196 100 200 200 255 255 506a
Pause
median 0 0
Pause
histo 506a
Pause
dhisto 318 100 450 200 255 255 506a
Pause
setwindow 0 0 511 511
```

```
clear 0 7
Pause
winfrdisk 4096 ima43.bin 506a-1
Pause
rectangle 200 200 300 300
setwindow 200 200 300 300
Pause
histo 506a
Pause
cp2 0 0
Pause
histo 506a
Pause
dhisto 307 100 450 200 255 255 506a
Pause
setwindow 0 0 511 511
clear 0 7
Pause
winfrdisk 4096 ima43.bin 506a -1
Pause
rectangle 200 200 300 300
setwindow 200 200 300 300
"
```



```
Pause
ber3 -1 -1 -1 -1 3 -1 -1 -1 -1 52000
Pause
con3 0 0 52000
Pause
histo 506a
Pause
dhisto 190 100 450 200 255 255 506a
Pause
setwindow 0 0 511 511
clear 0 7
Pause
Winfrdisk 4096 ima43.bin 506a -1
Pause
rectangle 200 200 300 300
setwindow 200 200 300 300
Pause
histo 506a
Pause
dhisto 190 100 200 200 255 255 506a
Pause
sh2 0 0
Pause
"
```

isto 506a
use
nisto 752 100 150 200 255 255 506a
use
twindow 0 0 511 511
ear 0 7
ause
infrdisk 4096 imag3.bin 506a -1
use
ctangle 200 200 300 300
twindow 200 200 300 300
use
isto 506a
use
histo 196 100 200 200 255 255 506a
use
tart 0 0
use
isto 506a
use
histo 34c 100 450 200 255 255 506a
use
twindow 0 0 511 511

"

clear 0 7
Pause
winfrdisk 4096 imag3.bin 506a -1
Pause
rectangle 200 200 300 300
setwindow 200 200 300 300
Pause
histo 506a
Pause
dhisto 196 100 200 200 255 255 506a
Pause
setwindow 0 0 511 511
clear 0 7
Pause
winfrdisk 4096 imag3.bin 506a -1
Pause
rectangle 200 200 300 300
setwindow 200 200 300 300
Pause
histo 506a
Pause
dhisto 196 100 200 200 255 255 506a
pause

"

```
setwindow 0 0 511 511
clear 0 7
Pause
winfrdisk 4096 ima93.bin 506a -1
Pause
rectangle 200 200 300 300
setwindow 200 200 300 300
Pause
dhisto 190 100 200 200 255 255 506a
Pause
horfilter 0 0
Pause
histo 506a
Pause
dhisto 300 100 450 200 255 255 506a
Pause
setwindow 0 0 511 511
clear 0 7
Pause
winfrdisk 4096 ima93.bin 506a -1
Pause
rectangle 200 200 300 300
```



```
setwindow 200 200 300 300
Pause
histo 506a
Pause
dhisto 190 100 200 200 255 255 506a
Pause
verfilter 0 0
Pause
histo 506a
Pause
dhisto 3e2 100 450 200 255 255 506a
```

PRACTICE 9

How is it POSSIBLE TO DILATE OR ERODE THE contents into A PRE-DEFINED WINDOW?

OMMAND DESCRIPTIONS:

Pda

dilate) source dest

It PERFORMS A DILATION ON THE CONTENTS OF THE WINDOW IN THE SOURCE WORKSPACE. IT USES A 3x3 STRUCTURING ELEMENT TO PERFORM THE DILATION.

erode) source dest

It PERFORMS AN "EROSION" ON THE CONTENTS OF THE WINDOW IN THE SOURCE WORKSPACE. IT USES A 3x3 STRUCTURING ELEMENT TO PERFORM THE "EROSION."

PRACTICE 9 :

pda

```
initmt 26c 1101
set window 00 511 511
clear 07
pause
scaling 0 100 100 0 586a
lutd 0 1 0 256 586a
pause
set window 200 200 300 300
rectangle 200 200 300 300
pause
dilate 0 0
pause
dilate 0 0
pause
dilate 0 0
pause
dilate 0 0
pause
dilate 0 0
pause
dilate 0 0
pause
dilate 0 0
pause
```

```
dilate ϕ ϕ
pause
dilate ϕ ϕ
pause
erode ϕ ϕ
pause
erode ϕ ϕ
pause
erode ϕ ϕ
pause
erode ϕ ϕ
pause
erode ϕ ϕ
pause
erode ϕ ϕ
pause
erode ϕ ϕ
pause
erode ϕ ϕ
pause
erode ϕ ϕ
pause
erode ϕ ϕ
pause
set window ϕ ϕ 511 511
```


PRACTICE 10

HOW is it POSSIBLE TO USE ZOOM?

COMMAND DESCRIPTIONS:

zoom mode

This COMMAND SELECTS FULL RESOLUTION OR ZOOM RESOLUTION. THE POSITION OF IMAGES WITHIN A QUADRANT IS SET BY THE PAN AND SCROLL COMMANDS.

mode:

Ø → STANDARD VIDEO

NONZERO → ZOOMED VIDEO

SCROLL(REL) offset

IT IS USED TO SCROLL the IMAGE RELATIVE TO its CURRENT POSITION ON THE SCREEN.

offset: IT SPECIFIES THE RELATIVE SCROLL of the WINDOW. SCROLLING is DONE in BLOCKS OF SIXTEEN PIXELS

panr(el) offset

It is USED to shift the DISPLAYED IMAGE RELATIVE TO its CURRENT Position on the SCREEN.

offset:

It SPECIFIES the NUMBER OF PIXELS BY Which THE IMAGE will move.

Panning is PERFORMED IN Blocks OF EIGht PIXELS

(Pdb)

PRACTICE 10 :

```
Initmt 26c 1101 0
Jet window 0 0 511 511
clear 07
Pause
Scaling 0 100 100 0 506a
lutd 0 1 0 256 506a
Pause
winfrdisk 4096 imag3.bin 506a
Pause
300m 1
Pause
panrel 80
pause
scrorel 80
Pause
scrorel 80
pause
scrorel 70
pause
scrorel -70
pause
scrorel -80
Pause
scrorel -80
pause
panrel 80
pause
```

```
scrorel 80
Pause
scrorel 80
Pause
scrorel 70
pause
scrorel -70
Pause
scrorel -80
Pause
scrorel -80
Pause
panrel 100
Pause
scrorel 80
Pause
scrorel 80
Pause
scrorel 70
Pause
scrorel -70
Pause
scrorel -80
Pause
scrorel -80
Pause
```

panel - 80
pause
panel - 90
pause
panel - 100
pause
zoom 0