9. Displaying Image Data

- Read an image, i.e. 'galaxy.dat' or 'ctscan.dat'

```
OPENR,lun,DIALOG_PICKFILE( ), /GET_LUN

im=BYTARR(256,256)

...

TV, im              ; default position: lower left corner
TV, im, /order      ; reverse the row order
```

- Position the image in the window

```
TV,im,300,200          ; image offset in pixels
```

Display a color bar on the left of the image:

```
bar=BYTARR(20,256)
FOR i=0,19 DO bar[i,*]=BINDGEN(256)
TV, ....
```

- Process image with basic filters and display them in one window :

```
TV,im,0                    ; position 0, upper left corner
TV,smooth(im,5),1          ; boxcar average 5 by 5, pos. 1
TVSCL,MEDIAN(im,3),2       ; median smoothing
      (suppress noise tips, does not blur large edges...)
TVSCL, SOBEL(im),3         ;edge enhancement operator
```

- Define you own filter
  - arbitrary filter kernels with the use of IDL's CONVOL-function
  - Frequency Domain Filtering, example:

```
imf=FFT(im,-1)                ; image to frequency domain
filter= DIST(256) LT 50       ; define a low pass
TVSCL,SHIFT(filter,128,128)   ; display filter
imf=filter*imf                ; apply the filter
TVSCL, FFT(imf,1),4           ; back to the spatial domain
```

Working with images

- Using '<' '>' operators for contrast enhancement:

```
TV,im>40,0                ; '>' operator returns a value
                          ;equal to larger of its operants
TVSCL,im>40,1             ;TVSCL uses the full color range
TVSCL,im > 40 < 200,3     ; set a range of value
i=WHERE ( (im GE 200) or (im le 100) )
im[i]=0
TVSCL,im
```

- Exercise 9.1, processing the file 'cereb.dat', steps:

  -Read two X-Ray images into one array:

  data=BYTARR(512,512,2)

  -Display the images side by side in one window

  -Substract the images to display the differences: (Tip: cast 1. image to integer with 'FIX': DIFF=FIX(images2)-images1

  - Plot the histogram of the difference:

  PLOT,HISTOGRAM (DIFF)

  - Contrast enhancement by :

  DIFF= HIST_EQUAL(DIFF)

  - Plot the histogram after histogram equalization and compare the images

  (diff.pro)

13. Gridding Random or Irregular Data

- Dataset with 200 random points in the xy plane:

  x=RANDOMU(SEED,200)

  y=RANDOMU(SEED,200)

  data = (x+y)^2

- Plot the location of the points in the xy plane:

  PLOT,....., PSYM=...

- Triangulating the x-y points:

```
TRIANGULATE,x,y,tr          ;Delauny triangulation
Help,tr                     ;Number of triangles
```

- Create a regular grid:

```
grid=TRIGRID(x,y,data,tr)
HELP,grid                    ; by default 51 intervals:
                             ; (MAX-MIN)/50
SURFACE, grid
```

- Regular grid with more intervals:

```
spacing=[0.01,0.01]  ;
grid=TRIGRID(x,y,data,tr,spacing)
                             ;optional argument to trigrid !
SHADE_SURF,grid
```

14. Volume Visualization

- Create a simple dataset:

  vol_data=BYTARR(50,50,50)

  vol_data[10:30,5:25,10:30]=10

  vol_data[25:45,30:45,25:40]=200

- SHADE_VOLUME generates a list of vertices ('v') and a list of polygons ('p'). They define a 3D surface at a constant density level :

  SHADE_VOLUME,vol_data,5,v,p

  ; density level (isosurface)=5

- set up a user defined 3D-transformation:

  SCALE3,XRANGE=[0,50],YRANGE=[0,50], $

  ZRANGE=[0,50]

- create a shaded representation of the surface defined by the polygons 'p' and the vertices 'v' :

  TV,POLYSHADE(v,p,/t3d)

- set the density level to 100 :

  : :

- Voxel rendering in IDL's Object Graphics

```
oVol = obj_new('idlgrvolume', vol_data)

xobjview, oVol
```

- Volume Visualisation with IDL's SLICER3 -Tool:

```
vol_ptr = PTR_NEW(vol_data)  ;put data to a pointer

SLICER3, vol_ptr             ;SLICER3 with a pointer arg.
```

- reading the volumetric data file '.../ data/head.dat'
  with 57 images with 80 x 100 pixels:

```
data=bytarr(80,100,57)

OPENR,unit,DIALOG_PICKFILE( ),/GET_LUN

READU, unit, data                    ;Read the data

FREE_LUN, unit

SLICER3, PTR_NEW(data, /no_copy)
```