

---

Subject: DIS and DISMENU image display programs: 3/4

Posted by [grunes](#) on Mon, 19 May 1997 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

DIS and DISMENU IDL/PV-WAVE image display programs: 3/4

-----CUT HERE----dismenu.pro-----

```
;-----  
pro DisMenu,FilNam1,FilNam2,FilNam3,FilNam4,FilNam5,FilNam6, $  
  iType=iType,nCol=nCol,nRow=nRow,iCol1=iCol1, $  
  iCol2=iCol2, iRow1=iRow1,iRow2=iRow2,nHeader=nHeader,iGeom=iGeom, $  
  Black=Black,White=White,iColor=iColor,nFrame=nFrame  
    ; Main program to display image files  
    ; using dis.pro.  
    ; There are many good free image  
    ; display programs, like xv and  
    ; saimage.  
  
    ; All input arguments are optional:  
  
    ; FilNam1=Name of file to display.  
    ; FilNam2,3,4,5 & 6=Names of other  
    ; same size image files to display.  
    ; (Will be flickered with FilNam1.)  
    ; iType=File type--see PrintImTypes.  
    ; If FilNam1 is provided but iType is  
    ; not, type is guessed from the name.  
    ; nCol,nRow=# of columns--needed for  
    ; raw image types. Rows are stored  
    ; contiguously.  
    ; iCol1,iCol2,iRow1,iRow2=0-origin  
    ; range of cols and rows to load into  
    ; memory. The rest will be ignored.  
    ; nHeader=# of header bytes or lines.  
    ; iGeom=ROTATE display geometry type  
    ; (0,2,5, or 7). Affects display,  
    ; not col and row #'s. If you use  
    ; this other rotate geometries  
    ; (1,3,4,or 6), display will work  
    ; right, but not the ZOOM, UP, DOWN,  
    ; LEFT, or RIGHT buttons.  
    ; Black,White=Range of pixel values to  
    ; be displayed.  
    ; Black and White can be dimensioned,  
    ; for separate schemes for each image  
    ; iColor=loadct2 color scheme #.  
    ; iColor can be dimensioned,  
    ; for separate schemes for each image
```

```

; nFrame=# of images stored in file.
; Only applies to raw data. And only
; works with FilNam1.
;Written by Mitchell R Grunes.
if n_elements(FilNam1) eq 0 then FilNam1=""
if n_elements(FilNam2) eq 0 then FilNam2=""
if n_elements(FilNam3) eq 0 then FilNam3=""
if n_elements(FilNam4) eq 0 then FilNam4=""
if n_elements(FilNam5) eq 0 then FilNam5=""
if n_elements(FilNam6) eq 0 then FilNam6=""
if n_elements(iType) eq 0 then begin
  iType=1
  if strpos(FilNam1, '.') ge 0 then begin ; Guess type from file name.
    suffix=strmid(FilNam1, strpos(FilNam1, '.')+1, 999)
    while strpos(suffix, '.') ge 0 do $
      suffix=strmid(suffix, strpos(suffix, '.')+1, 999)
    if suffix eq 'txt' then iType=18
    if suffix eq 'gif' then iType=20
    if suffix eq 'pic' or suffix eq 'pict' then iType=21
    if suffix eq 'srf' then iType=22
    if suffix eq 'wav' then iType=23
    if suffix eq 'x11' then iType=24
    if suffix eq 'xwd' then iType=25
    if suffix eq 'tif' or suffix eq 'tiff' then iType=26
    if iType eq 26 and n_elements(iGeom) eq 0 then iGeom=0
    if suffix eq 'soho' then iType=27
    if suffix eq 'pgm' then iType=29
    if suffix eq 'tla' then iType=30
    if suffix eq 'sealab' then iType=31
    if suffix eq 'hdf' then begin
      iType=32
      if !prompt eq 'IDL> ' then iType=37
    endif
    if suffix eq 'rgb' then iType=33
    if suffix eq 'bw' then iType=34
  endif
endif
if n_elements(nCol) eq 0 then nCol=512L else nCol=long(nCol)
if n_elements(nRow) eq 0 then nRow=512L else nRow=long(nRow)
if n_elements(iCol1) eq 0 then iCol1=0L
if n_elements(iCol2) eq 0 then iCol2=0L
if n_elements(iRow1) eq 0 then iRow1=0L
if n_elements(iRow2) eq 0 then iRow2=0L
if n_elements(nHeader) eq 0 then nHeader=0L
if n_elements(iGeom) eq 0 then iGeom=7
if n_elements(Black) eq 0 then Black=0
if n_elements(White) eq 0 then White=0
if n_elements(iColor) eq 0 then iColor=0

```

```

if n_elements(nFrame) eq 0 then nFrame=1

select=[0,0,0,0]

if !prompt eq 'IDL> ' then device,pseudo=8
if !prompt eq 'IDL> ' then wait,.1
if !d.name eq 'WIN' then window,0
window,0,colors=256
if !prompt eq 'IDL> ' then begin
    device,get_screen_size=ysz
    xsz=ysz(0)
    ysz=ysz(1)
endif else begin
    xsz=1000
    if !d.name eq 'WIN' then xsz=800
    ysz=900
    if !d.name eq 'WIN' then ysz=600
endelse

window,0,colors=256,retain=2,xsize=xsx,ysize=ysz

On_IOerror,Menu

```

Menu:

```

if (!d.flags and 256) ne 0 then wshow,0,0
print,"
print,'---Main Dismenu Menu---
print,'1 Image File to Display   =',FilNam1
print,'2 2nd Image File to Display=',FilNam2
print,'3 3rd Image File to Display=',FilNam3
print,'4 4th Image File to Display=',FilNam4
print,'5 5th Image File to Display=',FilNam5
print,'6 6th Image File to Display=',FilNam6
print,'10 Type of image          =',sq(iType)
if iType le 19 then begin
    print,'11 # of Columns        =',sq(nCol)
    print,'12 # of Rows           =',sq(nRow)
endif
if select(0) eq 0 and select(1) eq iCol2-iCol1 then select(1)=0
if select(2) eq 0 and select(3) eq iRow2-iRow1 then select(3)=0
if iCol1 eq 0 and iCol2 eq nCol-1 then iCol2=0
if iRow1 eq 0 and iRow2 eq nRow-1 then iRow2=0
print,'13 Cols to load (0,0=all)=' ,sq(iCol1),',',sq(iCol2)
print,'  Cols within loaded area to display=',sq(select(0:1))
print,'14 Rows to load (0,0=all)=' ,sq(iRow1),',',sq(iRow2)
print,'  Rows within loaded area to display=',sq(select(2:3))
if iType ge 16 and iType le 19 then begin
    print,'15 # of header lines    =',sq(nHeader)

```

```

endif else if iType le 15 then begin
  print,'15 # of header bytes      =',sq(nHeader)
endif
print,'16 Display Geometry        =',sq(iGeom)
print,'17 Black,White pixel values =',sq(black),',',sq(white)
print,'18 Color Scheme            =',sq(iColor)
if iType le 19 then $
  print,'19 # of image frames      =',sq(nFrame)

print,"
print,'50 Display Image(s)'
print,'51 Execute Operating System Commands
print,'99 Exit Program
print,"
read,'Option #:',choice
if choice eq 1 then begin
  read,'Image File to Display:',FilNam1
endif else if choice eq 2 then begin
  read,'2nd Image File to Display:',FilNam2
endif else if choice eq 3 then begin
  read,'3rd Image File to Display:',FilNam3
endif else if choice eq 4 then begin
  read,'4th Image File to Display:',FilNam4
endif else if choice eq 5 then begin
  read,'5th Image File to Display:',FilNam5
endif else if choice eq 6 then begin
  read,'6th Image File to Display:',FilNam6
endif else if choice eq 10 then begin
  PrintImTypes
  read,'Type of image:',iType
endif else if choice eq 11 then begin
  read,'# of Columns:',nCol
endif else if choice eq 12 then begin
  read,'# of Rows:',nRow
endif else if choice eq 13 then begin
  read,'Cols to load (0,0=all):',iCol1,iCol2
  i=select(0) & j=select(1)
  read,'Cols within loaded area to display=',i,j
  select(0)=i & select(1)=j
endif else if choice eq 14 then begin
  read,'Rows to load (0,0=all):',iRow1,iRow2
  i=select(2) & j=select(3)
  read,'Rows within loaded area to display=',i,j
  select(2)=i & select(3)=j
endif else if choice eq 15 then begin
  read,'Header Size:',nHeader
endif else if choice eq 16 then begin
  PrintGeomTypes

```

```

    read,'Geometry Type:',iGeom
endif else if choice eq 17 then begin
    print,'Current black value(s)=' ,sq(black)
    print,'Current white value(s)=' ,sq(white)
    read,'Black pixel value(s):',Black
    read,'White pixel value(s):',White
endif else if choice eq 18 then begin
    read,'Color Scheme(s):',iColor
endif else if choice eq 19 then begin
    read,'# of image frames:',nFrame
endif else if choice eq 50 then begin
    a=0 & b=0
    ReadImage,filnam1,a,nCol,nRow,iCol1,iCol2,iRow1,iRow2, $
    nheader,nFrame,iType=iType
    if FilNam2 gt ' ' then $
        ReadImage,filnam2,b,nCol,nRow,iCol1,iCol2,iRow1,iRow2, $
        nheader,nFrame,iType=iType
    if FilNam3 gt ' ' then $
        ReadImage,filnam3,b3,nCol,nRow,iCol1,iCol2,iRow1,iRow2, $
        nheader,nFrame,iType=iType
    if FilNam4 gt ' ' then $
        ReadImage,filnam4,b4,nCol,nRow,iCol1,iCol2,iRow1,iRow2, $
        nheader,nFrame,iType=iType
    if FilNam5 gt ' ' then $
        ReadImage,filnam5,b5,nCol,nRow,iCol1,iCol2,iRow1,iRow2, $
        nheader,nFrame,iType=iType
    if FilNam6 gt ' ' then $
        ReadImage,filnam6,b6,nCol,nRow,iCol1,iCol2,iRow1,iRow2, $
        nheader,nFrame,iType=iType
    print,'Loaded image characteristics:
    if !prompt eq 'IDL> ' then help,a
    if n_elements(b) gt 1 and !prompt eq 'IDL> ' then help,b
    if !prompt ne 'IDL> ' then info,a    ; Lately, Wave replaced
    if n_elements(b) gt 1 and !prompt ne 'IDL> ' then info,b

    GetSize,a, nColA,nRowA
    GetSize,b, nColB,nRowB
    if n_elements(b) gt 1 and (nColA ne nColB or nRowA ne nRowB) then $
        stop,'*****ERROR: SIZES DO NOT MATCH*****

    if !prompt eq 'IDL> ' then device,pseudo=8
    if !prompt eq 'IDL> ' then wait,.1
    if FilNam3 gt ' ' or FilNam4 gt ' ' or FilNam5 gt ' ' or FilNam6 gt ' ' $
    then begin
        niter=(FilNam1 gt ' ')+(FilNam2 gt ' ')+(FilNam3 gt ' ')+ $
        (FilNam4 gt ' ')+(FilNam5 gt ' ')+(FilNam6 gt ' ')
        bb=0b
        if FilNam1 gt ' ' then bb=bb+a(0)    ; set type

```

```

if FilNam2 gt ' ' then bb=bb+b(0)
if FilNam3 gt ' ' then bb=bb+b3(0)
if FilNam4 gt ' ' then bb=bb+b4(0)
if FilNam5 gt ' ' then bb=bb+b5(0)
if FilNam6 gt ' ' then bb=bb+b6(0)
bb=replicate(bb,nColA,nRowA,niter)
iter=0
title=strarr(niter)
if FilNam1 gt ' ' then begin
  bb(*,*,iter)=a
  a=0
  title(iter)=FilNam1
  iter=iter+1
endif
if FilNam2 gt ' ' then begin
  bb(*,*,iter)=b
  b=0
  title(iter)=FilNam2
  iter=iter+1
endif
if FilNam3 gt ' ' then begin
  bb(*,*,iter)=b3
  b3=0
  title(iter)=FilNam3
  iter=iter+1
endif
if FilNam4 gt ' ' then begin
  bb(*,*,iter)=b4
  b4=0
  title(iter)=FilNam4
  iter=iter+1
endif
if FilNam5 gt ' ' then begin
  bb(*,*,iter)=b5
  b5=0
  title(iter)=FilNam5
  iter=iter+1
endif
if FilNam6 gt ' ' then begin
  bb(*,*,iter)=b6
  b6=0
  title(iter)=FilNam6
  iter=iter+1
endif
dis,bb,black=black,white=white,iGeom=iGeom,iColor=iColor, $
  Select=Select,iFill=iFill,title=title,/keepwindow
help,bb
endif else begin

```

```

    dis,a,b,black=black,white=white,iGeom=iGeom,iColor=iColor, $
    Select=Select,iFill=iFill,title=[FilNam1,FilNam2],/keepwindo w
endelse
endif else if choice eq 51 then begin
    print,'End this mode with  exit
    spawn
endif else if choice eq 99 then begin
    On_IOerror,null
    if (!d.flags and 256) ne 0 then wdelete,0
    if (!d.flags and 256) ne 0 then wdelete,1
    return
endif
goto,Menu
end
;-----
dismenu
end
-----CUT HERE-----

-----CUT HERE----getsize.pro-----
;-----
pro GetSize,a, nCol,nRow          ; Output # of columns and rows in
                                ; array a.
;Written by Mitchell R Grunes.
s=size(a)
nCol=s(1)
nRow=s(2)
end
-----CUT HERE-----

-----CUT HERE----loadct2.pro-----
;-----
pro loadct2,iColor                ; Modified loadct which adds other
                                ; color schemes.
;Written by Mitchell R Grunes.

;50 is reserved for a user color scheme defined by usercol.pro.
;Note that 99 is reserved for a special color scheme where
; the 253,254,255 are red,green,blue; rest is monochrome.
;(51 will select 50, and call usercol to edit it)
;(52 is white/red/yellow/green/cyan/blue/purple)
; 99 is only usable for byte images.
; i.e., should use
; device,pseudo=8
; wait,.1
; window,n, colors=256,xpos=x,ypos=y
; loadct2,99
;Note that 98 is another such color scheme:

```

```

; 0-127=grey shades, 128-254=yellow shades, 255=red.
;So is 97:
; 0-249 is a grey scale.
; 250 is red, 251 is yellow, 252 is green, 253 is light blue,
; 254 is bright blue, 255 is purple
;96 is like 97, except that it uses fewer colors (good for screens with other
; icons):
; 0-121 is a grey scale.
; 122 is red, 123 is yellow, 124 is green, 125 is light blue,
; 126 is bright blue, 127 is purple
if iColor eq 51 then begin
    usercol,/no_window
    iColor=50
endif
if iColor eq 99 then begin
    r=byte(indgen(256)*255.9d0/252)
    g=r & b=r
    r(253)=255 & g(253)=0 & b(253)=0
    r(254)=0 & g(254)=255 & b(254)=0
    r(255)=0 & g(255)=0 & b(255)=255
    if !d.table_size lt 256 then begin
        i=fix(indgen(!d.table_size)*255./(!d.table_size-1)+.5)
        r=r(i)
        g=g(i)
        b=b(i)
    endif
    tvlct,r,g,b
endif else if iColor eq 98 then begin
    r=byte(2*[indgen(128),indgen(128)]) ; color map: 0-126=ocean=grey
    g=r ; 128-254=land=yellow
    b=r ; 255=red lines
    b(128:255)=0
    r(128:159)=32+indgen(32) ; Make sure land always shows
    g(128:159)=32+indgen(32)
    g(255)=0
    if !d.table_size lt 256 then begin
        i=fix(indgen(!d.table_size)*255./(!d.table_size-1)+.5)
        r=r(i)
        g=g(i)
        b=b(i)
    endif
    tvlct,r,g,b
endif else if iColor eq 97 then begin
    r=byte(indgen(256)*255.9d0/249)
    g=r & b=r
    r(250)=255 & g(250)=0 & b(250)=0
    r(251)=255 & g(251)=255 & b(251)=0
    r(252)=0 & g(252)=255 & b(252)=0

```

```

r(253)=0 & g(253)=255 & b(253)=255
r(254)=0 & g(254)=0 & b(254)=255
r(255)=255 & g(255)=0 & b(255)=255
if !d.table_size lt 256 then begin
    i=fix(indgen(!d.table_size)*255./(!d.table_size-1)+.5)
    r=r(i)
    g=g(i)
    b=b(i)
endif
tv!ct,r,g,b
endif else if iColor eq 96 then begin
    r=byte(indgen(128)*255.9d0/121)
    g=r & b=r
    r(122)=255 & g(122)=0 & b(122)=0
    r(123)=255 & g(123)=255 & b(123)=0
    r(124)=0 & g(124)=255 & b(124)=0
    r(125)=0 & g(125)=255 & b(125)=255
    r(126)=0 & g(126)=0 & b(126)=255
    r(127)=255 & g(127)=0 & b(127)=255
    if !d.table_size lt 128 then begin
        i=fix(indgen(!d.table_size)*127./(!d.table_size-1)+.5)
        r=r(i)
        g=g(i)
        b=b(i)
    endif
    tv!ct,r,g,b
endif else if iColor eq 50 or iColor eq 52 then begin
    if iColor eq 52 then begin
        ncolorr=7 & ncolorg=7 & ncolorb=7
        xrf=[0,42,84,127,170,212,255] & xgf=xrf & xbf=xrf; independent variable.
        rf=[255,255,255,0,0,0,255] ; dependent variable:
        gf=[255,0,255,255,255,0,0] ; black,blue,cyan,green,yellow,
        bf=[255,0,0,0,255,255,255] ; red,white
        goto,GoodFile
    endif
    On_IOerror,NoFile
    openr,unit,'usercol.dat',/get_lun
    readf,unit,ncolorr,ncolorg,ncolorb ; # of mark points

    xrf=intarr(ncolorr) ;Independent coord of transfer
    xgf=intarr(ncolorg) ; function: R,G,B
    xbf=intarr(ncolorb)

    rf=xrf & gf=xgf & bf=xbf ;dependent coord for R,G,B
    readf,unit,xrf,xgf,xbf,rf,gf,bf
    free_lun,unit
    goto,GoodFile
NoFile: ; Default color scheme if no file

```

```

ncolorr=7 & ncolorg=7 & ncolorb=7
xrf=[0,42.5,110,127.5,145,212.5,255] & xgf=xrf & xbf=xrf
rf=[0, 0, 0, 0,255,255,255] ; black,blue,cyan,green,yellow,red,
gf=[0, 0,255,255,255, 0,255] ; white.
bf=[0,255,255, 0, 0, 0,255]

GoodFile:
On_IOerror,null

r=interpol(float(rf),float(xrf),float(indgen(256))) ;interpolate.
g=interpol(float(gf),float(xgf),float(indgen(256)))
b=interpol(float(bf),float(xbf),float(indgen(256)))

scale=tan(1.) ; Try to compensate for saturation,
r=fix( (scale+tan((r-127.5)/127.5)) * (255/scale/2) +.5) ; and round.
g=fix( (scale+tan((g-127.5)/127.5)) * (255/scale/2) +.5)
b=fix( (scale+tan((b-127.5)/127.5)) * (255/scale/2) +.5)

if !d.table_size It 256 then begin
  i=fix(indgen(!d.table_size)*255./(!d.table_size-1)+.5)
  r=r(i)
  g=g(i)
  b=b(i)
endif
tv!ct,r,g,b
return
endif else begin
  loadct,iColor
endelse
end
-----CUT HERE-----

-----CUT HERE----outimage.pro-----
;-----
pro OutImage,format,name,image,pseudo=pseudo; Output an image--many formats.
;Written by Mitchell R Grunes.

; -----INPUT-----
; format= See PrintImTypes.
; If undefined or -99, will prompt
; user for format.
; name=Name of output file.
; If is blank or undefined, will be
; replaced by sequential name.
; If name does not contain a ".", a
; suffix determined by the image
; type will be added.
; image =Image array to write.
; If undefined, will be taken
; from current window.

```

```

;      RGB images should have a 3rd
;      dimension of 3.
; /pseudo indicates that the
; pseudo-color scheme of the current
; window should be applied. If the
; image is monochrome and format =
; Iris RGB, /pseudo is implied.
; Note that values over the number of
; colors in the current pseudo-color
; map will be clipped.
if n_elements(format) eq 0 then format=-99
if format eq -99 then begin
    PrintImTypes,/write
    read,'Output file format (33?):',format
endif

cursor,xdummy,ydummy,/device,/nowait      ; To flush X-Windows buffer

if n_elements(name) eq 0 then nn="" else nn=name
common CmOutImage,number
if n_elements(number) eq 0 then number=0
if nn eq "" or nn eq ' ' then begin
    if number lt 1000 then nn=string(number,format='(i3.3)') $
    else nn=sq(number)
endif
if strpos(nn, '.') lt 0 then begin
    if format ge 0 and format le 15 then nn=nn+'.im'
    if format ge 16 and format le 19 then nn=nn+'.txt'
    if format eq 20 then nn=nn+'.gif'
    if format eq 21 then nn=nn+'.pic'
    if format eq 22 then nn=nn+'.srf'  ;??
    if format eq 23 then nn=nn+'.wav'  ;??
    if format eq 26 then nn=nn+'.tif'
    if format eq 27 or format eq 28 then nn=nn+'.soho'
    if format eq 29 then nn=nn+'.pgm'
    if format eq 30 then nn=nn+'.tla'  ;??
    if format eq 31 then nn=nn+'.sealab';??
    if format eq 32 then nn=nn+'.hdf'
    if format eq 33 then nn=nn+'.rgb'
    if format eq 34 then nn=nn+'.bw'
    if format eq 35 then nn=nn+'.ps'
endif
number=number+1
if format eq -1 then return
print,'Writing image file: ',nn

if n_elements(image) eq 0 then begin ; read image from window if undefined
    if !d.table_size gt 256 then begin

```

```

image2=tvrd(0,0,!d.x_vsize,!d.y_vsize,true=3)
if total(image2(*,*,0) ne image2(*,*,1)) + $
    total(image2(*,*,0) ne image2(*,*,2)) eq 0 then $
    image2=reform(image2(*,*,0))
endif else begin
    image2=tvrd(0,0,!d.x_vsize,!d.y_vsize)
endelse
endif else begin
    image2=image
endelse
s=size(image2)
nDim=s(0)                ; # of dimensions.
nCol=s(1)
nRow=s(2)
if !prompt ne 'IDL> ' and nDim eq 2 and format eq 26 then $
    image2=rotate(image2,5)
                        ; convert to RGB if needed
if n_elements(pseudo) eq 0 then pseudo=0
if nDim eq 2 and format eq 33 then pseudo=1
if (nDim eq 3 or pseudo) and format eq 34 then $
    stop,'ERROR: Can not output a monochrome image with pseudo-color

tv!ct,/get,r,g,b
if pseudo and nDim eq 2 and format eq 33 then begin ; May need to change
    hi=n_elements(r)-1                ; pseudo-color to RGB
    lo=long(min(image2) < hi)
    hi=long(max(image2) < hi)
    if total(r(lo:hi) ne g(lo:hi))+total(r(lo:hi) ne b(lo:hi)) ne 0 then begin
        image3=hi < image2
        image2=bytarr(nCol,nRow,3)
        image2(*,*,0)=r(image3)
        image2(*,*,1)=g(image3)
        image2(*,*,2)=b(image3)
        image3=0 ; (save memory)
        nDim=3
    endif
endif

typ=s(s(0)+1)
if typ eq 1 then print,'Type=Byte Size=',s(1:s(0))
if typ eq 2 then print,'Type=Integer*2 Size=',s(1:s(0))
if typ eq 3 then print,'Type=Integer*4 Size=',s(1:s(0))
if typ eq 4 then print,'Type=Real*4 Size=',s(1:s(0))
if typ eq 5 then print,'Type=Real*8 Size=',s(1:s(0))
if typ eq 6 then print,'Type=Complex Size=',s(1:s(0))

if format ge 0 and format le 19 then begin
    openw,unit,nn,/get_lun

```

```

if format lt 16 then writeu,unit,image2
if format ge 16 then writef,unit,image2
free_lun,unit
endif else if format eq 20 then begin ; gif (2D only)
  write_gif,nn,image2,r,g,b
endif else if format eq 21 then begin ; PICT (2D only)
  Write_PICT,nn,image2,r,g,b
endif else if format eq 22 then begin ; Sun Rasterfile
  if nDim eq 2 then Write_SRF,nn,image2,R,G,B
  if nDim eq 3 then Write_SRF,nn,image2
endif else if format eq 23 then begin ; .wave or .bwave file (Adv Data Vis)
  write_srf,nn,image2,/bin
endif else if format eq 26 then begin ; Tiff
  if !prompt eq 'IDL> ' then begin
    if pseudo eq 0 or nDim eq 3 then begin
      if nDim eq 3 then begin ; Needs dim in other order
        image3=image2
        image2=bytarr(3,nCol,nRow)
        for i=0,2 do image2(i,*,*)=image3(*,*,i)
      endif
      tiff_write,nn,reverse(image2,2),1
    endif else begin
      tiff_write,nn,reverse(image2,2),1,red=r,green=g,blue=b
    endelse
  endif else begin
    ;The following could be used in place of tiff_write, but
    ; this procedure wouldn't be able to compile at all under IDL.
    ;class='Grayscale'
    ;if pseudo then class='Palette Color'
    ;if nDim eq 3 then class='RGB Full Color'
    ;palette=transpose([[r],[g],[b]])
    ;dummy=dc_write_tiff(nn,image2,class=class,palette=palette)
    if nDim eq 3 or pseudo then $
      print,'ERROR: Can not output a color Tiff with PV-WAVE Tiff_Write.
      print,'Writing monochrome instead
      tiff_write,nn,reverse(image2,2),1
    endelse
  endif else if format eq 29 then begin
    write_pgm,nn,image2
  endif else if format eq 30 then begin
    write_array,nn,image2
  endif else if format eq 33 then begin
    write_sgi_image,nn,image2,/pseudo
  endif else if format eq 34 then begin
    write_sgi_image,nn,image2
  endif else if format eq 35 then begin
    savedevice=!d.name
    set_plot,'ps

```

```

device,filename=nn
if ncol gt nrow then device,/landscape
if pseudo or nDim eq 3 then device,/color
if pseudo then device,bits_per_pixel=8
if pseudo then tvlct,r,g,b
if nDim eq 2 then tv,image2
if nDim eq 3 then tv,image2,true=1
device,/close_file
set_plot,savedevice
print,'Appending a control-D to end of file--',$
  'will cause run-time warning in trial version.'
openw,unit,nn,/append,/get_lun
writeu,unit,byte(4)
free_lun,unit
endif else if format eq 37 then begin
  WrSDF,filename,a,varname,frames=frames,palette=palette
endif else begin
  stop,'Invalid outimage format'
end
end

```

-----CUT HERE-----

-----CUT HERE----printgeomtypes.pro-----

```

;-----
pro PrintGeomTypes          ; Print Geometry Types.
;Written by Mitchell R Grunes.
print,'Geometry only affects display, not the way column and row numbers
print,' are counted!
print,'Available Geometries:
print,'0 Normal--rows are contiguous left > right, displayed bottom up'
;print,'1 Rotate CW 90 deg'
print,'2 Rotate 180 deg (reverse cols and rows)'
;print,'3 Rotate CCW 90 deg'
;print,'4 Reflect across main diag (i.e., transpose)'
print,'5 Reflect across vertical (reverse cols)'
;print,'6 Reflect across other diag'
print,'7 Reflect across horizontal (reverse rows) (use for JPL images)'
end

```

-----CUT HERE-----

-----CUT HERE----printimtypes.pro-----

```

;-----
pro PrintImTypes, Write=Write ; Print Image Types.
                                ; If /Write is selected, image
                                ; types are restricted to those
                                ; available for output.
;Written by Mitchell R Grunes.
print,'Type of file:

```

```

if n_elements(Write) eq 0 then Write=0
if Write eq 0 then begin
    print, ' -1=Grey scale test image
endif else begin
    print, ' -1=No output
endelse
if Write eq 0 then begin
    print, ' Raw Pixel Data
    print, ' 0=signed 8 bit
    print, ' 1=unsigned 8 bit
    print, ' 2=signed 16 bit
    print, ' 3=unsigned 16 bit
    print, ' 4=signed 32 bit
    print, ' 5=real 32 bit
    print, ' 6=real 64 bit
    print, ' 7=complex 64 bit (use absolute value)
    ;print, ' 8=complex 128 bit (use absolute value)
    print, ' Byte Reversed Raw Integral Pixel Data
    print, ' 9=signed 16 bit
    print, ' 10=unsigned 16 bit
    print, ' 11=signed 32 bit
    print, ' 12=real 32 bit
    ;print, ' 13=real 64 bit
    print, ' 14=complex 64 bit (use absolute value)
    ;print, ' 15=complex 128 bit (use absolute value)
endif else begin
    print, ' 0 =Raw Pixel Data
endelse
if Write eq 0 then begin
    print, ' ASCII Text of Pixel Data
    print, ' 16=Integral between -32768 and 32767
    print, ' 17=Integral between -2^31 and 2^31-1
    print, ' 18=real, requiring single precision
    print, ' 19=real, requiring double precision
endif else begin
    print, ' 16=ASCII Text of Pixel Data
endelse
print, ' Fancy image file formats
print, ' 20 GIF
print, ' 21 PICT
print, ' 22 Sun Rasterfile
print, ' 23 .wave or .bwave (Adv Data Vis)
if write eq 0 then begin
    print, ' 24 X11 Bitmap
    print, ' 25 XWD Dump
endif
print, ' 26 TIFF
print, ' 27 SOHO Compressed'

```

```

print,' 28 SOHO Compressed with header from IGSE'
print,' 29 Raw PGM (PBMPlus Grayscale format from jpeg)
print,' 30 Tom L Ainsworth's read_array/write_array formats
if write eq 0 then begin
print,' 31 SEALAB format files
print,' 32 HDF files (PV-WAVE only)
endif
print,' 33 Iris RGB image.
print,' 34 Iris BW image.
if Write ne 0 then $
  print,' 35 Postscript.
if Write eq 0 then $
  print,' 36 DRM ERS-1 CEOS image
print,' 37 HDF file with SD group (IDL only)

```

end

-----CUT HERE-----

-----CUT HERE----rankar.pro-----

```

function RankAr,a                ; # of dimensions in array a.
                                ; Returns -1 if variable has
                                ; not been defined.
;Written by Mitchell R Grunes.
s=size(a)
if s(0) eq 0 and s(1) eq 0 and s(2) eq 1 then return,-1
return,s(0)
end

```

-----CUT HERE-----

-----CUT HERE----rdfil.pro-----

; -----RdFil-----

```

pro RdFil,FilNam,a,init=init,skip=skip; Read a variable from a raw file.
                                ; ---INPUTS---
                                ; FilNam=Name of file.
                                ; a=Variable of proper type and
                                ; dimensions. (Modified on output.)
                                ; init, if defined, is used instead of
                                ; a to determine type and dimensions.
                                ; input. i.e.
                                ;   RdFil,FilNam,a,init=fltarr(10)
                                ; is equivalent to
                                ;   a=fltarr(10)
                                ;   RdFil,FilNam,a
                                ; skip, if defined, is the number of
                                ; bytes to skip. Default=0
                                ; ---OUTPUTS---
                                ; a=The value(s) read from the file.

```

; By Mitchell R Grunes.

```

print,'-----
print,'Reading file ',FilNam
openr,unit,FilNam,/get_lun
if n_elements(init) gt 0 then a=init
if n_elements(skip) eq 0 then skip=0
print,'Shape given by:'
if !prompt eq 'IDL> ' then help,a
if !prompt ne 'IDL> ' then info,a ; Lately, Wave replaced 'help'
point_lun,unit,skip
readu,unit,a
free_lun,unit

; with 'info'.

print,'-----
end

```

-----CUT HERE-----

-----  
 Mitchell R Grunes, grunes@imsy1.nrl.navy.mil. Opinions are mine alone.

---