Subject: Here is an image/movie display program (2/2)
Posted by grunes on Tue, 07 Jan 1997 08:00:00 GMT
View Forum Message <> Reply to Message

dis.pro: 1/2

----------CUT HERE----------
```
 ;------------------------------------------------------------ -----------
  pro ReadImFile,flag,flag2, a,iGeom          ; Internal routine--Prompt & read image.
;Written by Mitchell R Grunes.
  nCol=0L & nRow=0L
  iCol1=0L & iCol2=0L & iRow1=0L & iRow2=0L
  nHeader=0L
  if flag eq 0 then begin
    if flag2 eq 0 then print,'File name to display (blank for grey scale test):
  endif else begin
    if flag2 eq 0 then print,'Second (flickered) File name to display (blank for none):
  endelse
  filnam=''
  if flag2 eq 0 then read,filnam else readf,12,filnam

  if filnam le ' ' then begin
    iType=-1
    nCol=256
    nRow=256
    if flag eq 0 then begin
      goto,StartRead
    endif else begin
      a=0
      return
    endelse
  endif

  if flag2 eq 0 then PrintImTypes
  if flag2 eq 0 then read,iType else readf,12,iType

  if flag ne 0 and flag2 eq 0 then print,'(Must select same size area for second image)

  if iType ge 16 and iType le 19 then begin
    if flag2 eq 0 then print,'# of header lines of text to ignore:
    if flag2 eq 0 then read,nHeader else readf,12,nHeader
  endif else if iType le 15 then begin
    if flag2 eq 0 then print,'# of header bytes to ignore:
    if flag2 eq 0 then read,nHeader else readf,12,nHeader
  endif

  if iType lt 10 then begin
    if flag2 eq 0 then print,'# of Colomns,Rows of pixels:
```

```
    if flag2 eq 0 then read,nCol,nRow else readf,12,nCol,nRow
  endif

  if flag2 eq 0 then print,'Starting,Ending Column for possible display (0-origin) (0,0=load all)
  if flag2 eq 0 then read,iCol1,iCol2 else readf,12,iCol1,iCol2

  if flag2 eq 0 then print,'Starting,Ending Row for possible display (0-origin) (0,0=load all)
  if flag2 eq 0 then read,iRow1,iRow2 else readf,12,iRow1,iRow2

StartRead:
  iGeom=0
  if iType lt 10 or iType eq -1 or iType eq 31 or iType eq 32 then iGeom=7
  ReadImage,filnam,a,nCol,nRow,iCol1,iCol2,iRow1,iRow2, $
   iType,nheader
  end
 ;------------------------------------------------------------ -----------
  pro read_pgm,FilNam,a                    ; Read raw PGM file
;Written by Mitchell R Grunes.
  openr,unit,FilNam,/get_lun
  a='  '
  readu,unit,a
  if a ne 'P5' then print,'ERROR--Invalid PGM File'
  ReadFileInt,unit,ncol
  ReadFileInt,unit,nrow
  ReadFileInt,unit,maxa
  if maxa le 255 then a=bytarr(ncol,nrow)
  if maxa gt 255 then a=intarr(ncol,nrow)
  readu,1,a
  if maxa gt 255 then begin
    print,'Enter 0 if this is a LSB-1st machine (PC).'
    print,'Enter 1 if this is a MSB-1st machine (Sun, SGI):'
    ReadVar,i
    if i eq 0 then byteorder,a,/sswap
  endif
  close,unit
  free_lun,unit
  end
 ;------------------------------------------------------------ -----------
  pro write_pgm,FilNam,a              ; Write raw PGM file--not sure if works.
;Written by Mitchell R Grunes.
  GetSize,a, nCol,nRow
  openw,unit,FilNam,/get_lun
  writeu,unit,'P5'
  writeu,unit,sq(ncol)+' '
  writeu,unit,sq(nrow)+' '
  vt=VarTyp(a)
  if vt eq 1 then maxa=255
  if vt eq 2 then maxa=65535
```

```
  if vt ne 1 and vt ne 2 then stop,'Wrong variable type for write_pgm'
  writeu,unit,sq(maxa)+' '
  if maxa gt 255 then begin
    print,'Enter 0 if this is a LSB-1st machine (PC).'
    print,'Enter 1 if this is a MSB-1st machine (Sun, SGI):'
    ReadVar,i
    if i eq 0 then byteorder,a,/sswap
  endif
  writeu,1,a
  close,unit
  free_lun,unit
  end
 ;------------------------------------------------------------ -----------
  pro ReadFileInt,unit,n                ;Read ASCII # from file
;Written by Mitchell R Grunes.
start:
  n=0
  a=' '
  readu,unit,a
  if a eq '#' then begin                ;Skip comment
    readf,1,a
    goto,start
  endif
  if a le ' ' then goto,start           ;Skip leading white space
  while a ge '0' and a le '9' do begin
    n=n*10+long(a)
    readu,unit,a
  endwhile
  end
 ;------------------------------------------------------------ -----------
  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 Ainsworth''s read_array/write_array formats
print,'   31 SEALAB format files
print,'   32 HDF files (PV-WAVE only)
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 CEOS image
end
```

```
 ;------------------------------------------------------------ -----------
  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
 ;------------------------------------------------------------ -----------
  function SmoothEx,a,n             ; Like Smooth, but works near edges.
     ; Must be 1D or 2D.
;Written by Mitchell R Grunes.
  nd2=long(n/2)
  if RankAr(a) eq 2 then begin
    GetSize,a, nCol,nRow
    b=smooth(a,n)
    if nd2 lt nCol then b(0:nd2-1,*)=rebin(reform(b(nd2,*),1,nRow),nd2,nRow)
    if nd2 lt nRow then b(*,0:nd2-1)=rebin(reform(b(*,nd2),nCol,1),nCol,nd2)
    if nCol-nd2 ge 0 then  b(nCol-nd2:nCol-1,*)=rebin(reform(b(nCol-nd2-1,*),1,nRow),nd 2,nRow)
    if nRow-nd2 ge 0 then  b(*,nRow-nd2:nRow-1)=rebin(reform(b(*,nRow-nd2-1),nCol,1),nC
ol,nd2)
  endif else begin
    s=size(a)
    nCol=s(1)
    b=smooth(a,n)
    if nd2 lt nCol then b(0:nd2-1)=b(nd2)
    if nCol-nd2 ge 0 then b(nCol-nd2:nCol-1)=b(nCol-nd2-1)
  endelse
  return,b
  end
 ;------------------------------------------------------------ -----------
  pro  Dis1,a,iter,niter,iWin=iWin,iColor=iColor,range=range,select =select,iFill=iFill, $
    iGeom=iGeom,Offset=Offset,Fac,nCol3,nRow3,dt,noframe=noframe ; Internal routine
     ; to display 1 image in a window.
     ; --------------INPUTS--------------
     ; a=Image
     ; iWin=Window # to display in.
     ; iColor=loadct2 color map #.
     ; range=[black,white]
     ;  =Values to display as black and
     ;  white.  If not a vector, or not
```

```
    ;  defined, will be set to autoscale.
    ; select=[iCol1,iCol2,iRow1,iRow2]
    ;  =selected portion to display.,
    ;   if defined and iCol1 lt iCol2
    ;   and iRow1 lt iRow2.
    ; iFill=0 to display quickly,
    ;  1 or undefined to fill window.
    ; iGeom=Screen Geometry
    ; Offset=# to add to row #'s in prints
    ; dt=delay time
    ; noframe is set if you do not want a frame drawn
    ; --------------OUTPUTS-------------
    ; iWin,iColor,range,select,iFill may be
    ;  modified.
    ; Fac=Zoom factor
;Written by Mitchell R Grunes.
  !order=0
  GetSize,a,nCol,nRow

  wshow,iWin & wset,iWin
  n=!d.table_size
  Greybar=congrid(reform(indgen(n),n,1),100,20)
  if iWin ge 0 then wset,iWin

  VTa=vartyp(a)

  iCol1=select(0)
  iCol2=select(1)
  iRow1=select(2)
  iRow2=select(3)

  nCol2=iCol2-iCol1+1
  nRow2=iRow2-iRow1+1

  if iWin ge 0 then wset,iWin

  black=range(0) & white=range(1)
  if black ge white then begin
    black=min(a)
    white=max(a)
  endif

  xsz=!d.x_vsize & ysz=!d.y_vsize     ; Screen size
  Fac=(double(xsz-4)/nCol2) < (double(ysz-24)/nRow2) ; Enlargement factor
  if Fac ge 1 then begin              ; enlarge
    if iFill eq 0 then Fac=long(Fac)
    xs2=long(nCol2*Fac+.5)            ; size of englarged image
    ys2=long(nRow2*Fac+.5)
```

```
  i2=iCol2                      ; last col/row of image to use
  j2=iRow2
endif else begin                ; shrink
  iFac=long(1.d0/Fac+.999d0)
  if iFill eq 0 then Fac=1.d0/iFac
  xs2=long(nCol2/iFac)
  ys2=long(nRow2/iFac)
  i2=iCol1+xs2*iFac-1
  j2=iRow1+ys2*iFac-1
endelse
if fac lt 1 and iFill ne 0 then begin       ;(eventual size)
  xs2b=long(nCol2*Fac+.5)
  ys2b=long(nRow2*Fac+.5)
endif else begin
  xs2b=xs2
  ys2b=ys2
endelse
   ; Extract image section
c=a(iCol1:i2,iRow1:j2)
VTc=VarTyp(c)
if VTc lt 4 then begin              ; Must within legal range of type
  if VTc eq 1 then begin
    black=0.      > black < 255.
    white=0.d0    > white < 255.d0
  endif else if VTc eq 2 then begin
    black=-32768. > black < 32767.
    white=-32768. > white < 32767.
  endif
endif
        ; If bad, derive from image section
if black ge white then AutoScale,c,iWin, black,white
white=white > (black+1d-4)

range(0)=black & range(1)=white     ;Store back in range

print,'Image',sq(iter),' Cols ',sq(iCol1),'-',sq(iCol2), $
 ' Rows ',sq(iRow1+offset),'-',sq(iRow2+offset), $
 ' Black,white=',sq(black),',',sq(white),' Zoom=',sq(Fac)

if iGeom ne 0 then begin
if !prompt eq 'IDL> ' then begin
    c=rotate(temporary(c),iGeom)
  endif else begin
    c=rotate(c,iGeom)
  endelse
endif

        ; Save graphics pixels--
```

```
 graphsave=-1                    ; (No saved graphics)
 if (icolor eq 96 or icolor eq 97 or icolor eq 99) and Fac lt 1 then begin
   if icolor eq 96 then ix=where(c ge 122)    ; one dimensional locations of graphics
   if icolor eq 97 then ix=where(c ge 250)
   if icolor eq 99 then ix=where(c ge 253)
   if ix(0) ge 0 then begin
     print,'Saving ',sq(n_elements(ix)),' graphics pixels
     graphsave=c(ix)
     iy=ix/(i2-iCol1+1)                    ; Convert to x,y coordinates
     ix=ix-iy*(i2-iCol1+1)
     if Fac eq long(Fac) then begin          ; Convert to resized image coordinates
ix=ix*long(Fac)
iy=iy*long(Fac)
     endif else begin
ix=long(ix*Fac+.5)
iy=long(iy*Fac+.5)
     endelse
     ix=ix+iy*xs2b                       ; Convert back to one dimensional coord
     iy=0                              ; (save memory)
   endif
 endif

    ; Resize image
 if fac gt 1 and iFill ne 0 then begin
   c=congrid(c,xs2,ys2)
 endif else if fac gt 1 then begin
   c=rebin(c,xs2,ys2,/sample)
 endif else if fac lt 1 then begin
   if icolor eq 98 then c=rebin(c,xs2,ys2,/sample) else c=rebin(c,xs2,ys2)
   if iFill ne 0 then begin          ; (Note that it was averaged down
     c=congrid(c,xs2b,ys2b)          ;  first so bright and dark spots
   endif                           ;  were not lost.)
 endif

 if VTc lt 4 then begin              ; Type must match bytscl
   black2=long(black) & white2=long(white)
   if black2 eq white2 then begin
     if black2 ge 255 then black2=white2-1 else white2=black2+1
   endif
 endif else begin
   black2=black       & white2=white
 endelse

 top=!d.table_size-1
 if icolor eq 99 then top=252
 if icolor eq 98 then top=127
 if icolor eq 97 then top=249
 if icolor eq 96 then top=121
```

```
if icolor eq 99 and Fac ge 1 then begin
  c=(c lt byte(253)) * bytscl(c,black2,white2,top=top) + $
    (c ge byte(253)) * c
  c=byte(c)
endif else if icolor eq 98 and Fac ge 1 then begin
  c=(c lt byte(128)) * bytscl(c,black2,white2,top=top) + $
    (c ge byte(128)) * (bytscl(c and 127,black2,white2,top=top)+byte(128))
  c=byte(c)
endif else if icolor eq 97 and Fac ge 1 then begin
  c=(c lt byte(250)) * bytscl(c,black2,white2,top=top) + $
    (c ge byte(250)) * c
  c=byte(c)
endif else if icolor eq 96 and Fac ge 1 then begin
  c=(c lt byte(122)) * bytscl(c,black2,white2,top=top) + $
    (c ge byte(122)) * c
  c=byte(c)
endif else begin
  if !prompt eq 'IDL> ' then begin
    c=bytscl(temporary(c),black2,white2,top=top)
  endif else begin
    c=bytscl(c,black2,white2,top=top)
  endelse
endelse

if n lt 256 then begin
  if !prompt eq 'IDL> ' then begin
    c=byte((temporary(c)*(n-1))/255)
  endif else begin
    c=byte((c*(n-1))/255)
  endelse
endif

   ; Restore  graphics pixels
if graphsave(0) ge 0 then c(ix)=graphsave
ix=0 & & graphsave=0                ; Save memory

erase
Getsize,c, nCol3,nRow3
tv,c,2,20

top=byte(top*.82)                   ; Color that will show in all color schemes.
if noframe eq 0 then  begin          ; Draw frame
   plots,[0,nCol3+1,nCol3+1,0,0],[19,19,nRow3+20,nRow3+20,19],/ device, $
   color=top                         ;draw frame
   plots,[1,nCol3+2,nCol3+2,1,1],[18,18,nRow3+21,nRow3+21,18],/ device, $
   color=top
endif
xyouts,!d.x_vsize-25,0,sq(iter),/device,size=.8,color=top
```

```
 xyouts,4,  0,'Quit',/device,size=.8,color=top
 xyouts,40, 0,'Zoom',/device,size=.8,color=top
 xyouts,90,0,'Out',/device,size=.8,color=top
 xyouts,125,0,'Full',/device,size=.8,color=top
 tv,Greybar,210,0
 plots,[160,160],[0,19],/device,color=top
 plots,[210,210],[0,19],/device,color=top
 plots,[260,260],[0,19],/device,color=top
 plots,[360,360],[0,19],/device,color=top
 xyouts,370,0,'Keybd',/device,size=.8,color=top
 xyouts,430,0,'Left',/device,size=.8,color=top
 xyouts,470,0,'Right',/device,size=.8,color=top
 xyouts,520,0,'Up',/device,size=.8,color=top
 xyouts,550,0,'Down',/device,size=.8,color=top
 if niter gt 0 then begin
   xyouts,600,0,'Back',/device,size=.8,color=top
   xyouts,650,0,'Forw',/device,size=.8,color=top
   xyouts,700,5,'Speed',/device,size=.7,color=top
   plots,[700,750],[16,16],/device,color=top
   plots,[700,750],[ 2, 2],/device,color=top
 endif
 end
;------------------------------------------------------------ -----------
 pro  Dis,a,b,iWin=iWin,iColor=iColor,range=range,select=select,iF ill=iFill, $
   xpick=xpick,ypick=ypick,iGeom=iGeom,dt=dt,Offset=Offset,xsiz e=xsize,ysize=ysize, $
  title=title,xpos=xpos,ypos=ypos,noframe=noframe
    ; Display Image(s) in x-windows or
    ;  PC windows.
    ; --------------INPUTS--------------
    ; a=Image (two dimensional array).
    ;   Can also be a movie (3D array, last
    ;   dimension=# of images).
    ; b=Flickered image, if defined and
    ;   two dimensional.  Must be same shape as
    ;   a.  Do not use if a is a movie.
    ; iWin=Window # to display in, if
    ;  defined and ge 0.
    ; If b is being displayed, will also
    ;  use iWin+1
    ; iColor=loadct2 color map #, if
    ;  defined (default=0=monochrome).
    ;  iColor can have 2 elements for
    ;  different images.  See loadct2
    ;  for more details.
    ; range=[black,white]
    ;  =Values to display as black and
    ;  white.  If not a vector, or not
    ;  defined, will be set to the min and max.
```

```
    ; You may also set
    ;  range=[black,white,black2,white2]
    ;  to get seperate values for a and b
    ; select=[iCol1,iCol2,iRow1,iRow2]
    ;  =selected portion to display.,
    ;   if defined and iCol1 lt iCol2
    ;   and iRow1 lt iRow2.
    ; iFill=0 to display quickly,
    ;  1 or undefined to fill window.
    ; xpick,ypick: see below
    ; iGeom=Screen Geometry
    ; dt=delay time between flickers (default=1 sec).
    ;    Set to 1E10 to stop initially.
    ; Offset=# to add to row #'s in prints
    ; xsize,ysize=size of window to open.
    ; title=window title.
    ; xpos,ypos=window position.
    ; noframe is set if you do not want a frame drawn
    ; --------------OUTPUTS-------------
    ; iWin,iColor,range,select,iFill may be
    ;  modified.
    ; xpick,ypick=Point selected, if any.
    ;  The option of selecting a point only
    ;  exists if xpick is first set non-zero.

    ; ----------WARNING-----------
    ; On HP workstations, you have to clip into
    ; the top of the display window to get the
    ; right color scheme, and you have to click
    ; into the text window to see the text.  Very
    ; confusing, but that is the price of being
    ; able to see all 256 pseudo-colors.
;Written by Mitchell R Grunes.
  !order=0
  GetSize,a,nCol,nRow
  s=size(B)
  UsingB=rankAr(B) eq 2 and s(0) eq 2 and s(1) eq nCol and s(2) eq nRow ; 0 if B not defined or
invalid, else 1.
  niter=UsingB                    ; # of images-1
  s=size(A)
  if RankAr(A) eq 3 then niter=s(3)-1

  print,'---DIS---for help, click on Keybd, select help'

  if RankAr(iColor) lt 0 then iColor=0
  if RankAr(iGeom) ne 0 then iGeom=0

  if RankAr(range) eq 1 and $
```

```
 n_elements(range)/2*2 eq n_elements(range) then begin
  range=double(range)
endif else begin
  range=[min(a),max(a)]
endelse

if n_elements(select) eq 4 then begin
  iCol1=select(0)
  iCol2=select(1)
  iRow1=select(2)
  iRow2=select(3)
endif else begin                    ; (will be changed)
  iCol1=0 & iCol2=0 & iRow1=0 & iRow2=0
endelse
if RankAr(xPick) ne 0 then begin
  xPick=0 & yPick=0
endif
if RankAr(iFill) ne 0 then iFill=1
if n_elements(dt) eq 0 then dt=1
if RankAr(offset) ne 0 then offset=0
if RankAr(iWin) ne 0 then iWin=0
if n_elements(title) eq 0 then title=''
if n_elements(xpos) eq 0 then xpos=300
if n_elements(ypos) eq 0 then ypos=200
if n_elements(noframe) eq 0 then noframe=0

if (!d.flags and 256) eq 0 then iWin=-1 ; flag value for non-windowing system
if iWin ge 0 then begin             ; Windowing systems must be initialized
  if !prompt eq 'IDL> ' then device,pseudo=8
  if !prompt eq 'IDL> ' then wait,.1
  if n_elements(xsize) eq 0 then xsize=1000
  if n_elements(ysize) eq 0 then ysize=900
  if niter eq 0 then xsize=xsize > 625
  if niter gt 0 then xsize=xsize > 800
   window,iWin,colors=256,retain=2,xpos=xpos,ypos=ypos,xsize=xs ize,ysize=ysize,title=title
  wshow,iWin
  wset,iWin
endif
n=!d.table_size
Greybar=congrid(reform(indgen(n),n,1),100,20)
wset,iWin
erase
if n_elements(iColor) eq 1 then loadct2,iColor
if n_elements(iColor) ne 1 then loadct2,iColor(0)
tvcrs,10,5
   ;Wait till mouse button is up, flush X buffer
!err=1 & while !err ne 0 do cursor,x,y,/device,/nowait
```

```
  xsz=-1 & ysz=-1
  iter=0                                ; First image.
  if niter gt 0 then PixMapWin=replicate(-1,niter+1)    ; No pixmap windows opened yet.

  wset,iWin & wshow,iWin
  while 1 do begin

    iCol1= long( 0 > iCol1 < (nCol-1) )         ; Make sure col and row #'s
    iCol2= long( 0 > iCol2 < (nCol-1) )         ;  are legal.
    iRow1= long( 0 > iRow1 < (nRow-1) )
    iRow2= long( 0 > iRow2 < (nRow-1) )

    if iCol2 lt iCol1 then begin                ; If you clicked on right corner
      i=iCol2                                   ;  before left, reverse order.
      iCol2=iCol1
      iCol1=i
    endif

    if iRow2 lt iRow1 then begin                ; If you clicked on lower corner
      i=iRow2                                   ;  before upper, reverse order.
      iRow2=iRow1
      iRow1=i
    endif

    if iCol1 eq iCol2 then begin
      iCol1=0
      iCol2=nCol-1
    endif
    if iRow1 eq iRow2 then begin
      iRow1=0
      iRow2=nRow-1
    endif
    select=[iCol1,iCol2,iRow1,iRow2]

    nCol2=iCol2-iCol1+1
    nRow2=iRow2-iRow1+1

ReSize:                                 ; Create windows and pixmaps.
    wset,iWin & wshow,iWin
    if xsz ne !d.x_vsize or ysz ne !d.y_vsize then begin
      if xsz ne -1 then print,'Sizing pixmaps.
      xsz=!d.x_vsize & ysz=!d.y_vsize
      if iWin ge 0 then begin
 if niter eq 0 then xsz2=xsz > 625
 if niter gt 0 then xsz2=xsz > 800
 if xsz2 ne xsz then begin
   xsz=xsz2
    window,iWin,colors=256,retain=2,xpos=xpos,ypos=ypos,xsize=xs z,ysize=ysz,title=title
```

```
endif
if niter eq 0 then PixMapWin=[iWin]
if niter gt 0 then begin
  for iiter=0,niter do begin
    if PixMapWin(iiter) ge 0 then wdelete,PixMapWin(iiter)
     window,/pixmap,colors=256,retain=0,xsize=xsz,ysize=ysz,/free
    PixMapWin(iiter)=!d.window
  endfor
endif
     endif
     Made=bytarr(niter+1)                    ; No display images made yet.
   endif

   wset,iWin & wshow,iWin
   if xsz ne !d.x_vsize or ysz ne !d.y_vsize then goto,ReSize
   cursor,x,y,/device,/nowait                ; To flush X-Windows buffer
   if (!err ne 0) then goto,ReadCursor2      ; (button has been pushed)

   if Made(iter) eq 0 then begin             ; Prepare image pixmaps.
     black=range(0) & white=range(1)
     if n_elements(range) gt 2 then begin
black=range(iter*2) & white=range(iter*2+1)
     endif
     range2=[black,white]
     jColor=iColor
     if n_elements(iColor) gt 1 then jColor=iColor(iter)
     if niter eq 0 or (UsingB and iter eq 0) then begin
 Dis1,a,iter,niter,iWin=PixMapWin(iter),iColor=jColor,range=r ange2, $
 select=select,iFill=iFill, $
  iGeom=iGeom,Offset=Offset,Fac,nCol3,nRow3,dt,noframe=noframe
     endif else if UsingB then begin
 Dis1,b,iter,niter,iWin=PixMapWin(iter),iColor=jColor,range=r ange2, $
 select=select,iFill=iFill, $
  iGeom=iGeom,Offset=Offset,Fac,nCol3,nRow3,dt,noframe=noframe
     endif else begin
 Dis1,reform(a(*,*,iter)),iter,niter,iWin=PixMapWin(iter),iCo lor=jColor,range=range2, $
 select=select,iFill=iFill, $
  iGeom=iGeom,Offset=Offset,Fac,nCol3,nRow3,dt,noframe=noframe
     endelse
     Made(iter)=1
     wset,iWin & wshow,iWin
     ; If window has been resized, go back
     if xsz ne !d.x_vsize or ysz ne !d.y_vsize then goto,ReSize
     cursor,x,y,/device,/nowait
     if (!err ne 0) then goto,ReadCursor2      ; (button has been pushed)

     if niter gt 0 then device,copy=[0,0,!d.x_vsize<xsz,!d.y_vsize<ysz,0,0,PixMapWin(iter)]
     black=range2(0) & white=range2(1)
```

```
      if n_elements(range) gt 2 then begin      ; Set black,white
 range(2*iter)=black & range(2*iter+1)=white
      endif else begin
 range(0)=black & range(1)=white
      endelse
      oldtime=systime(1)

    endif

    wset,iWin & wshow,iWin                    ; If window has been resized, go back
    if xsz ne !d.x_vsize or ysz ne !d.y_vsize then goto,ReSize
    cursor,x,y,/device,/nowait
    if (!err ne 0) then goto,ReadCursor2      ; (button has been pushed)

    tvcrs,1
ReadCursor:
    wset,iWin & wshow,iWin
    if xsz ne !d.x_vsize or ysz ne !d.y_vsize or Made(iter) eq 0 then goto,ReSize
    if niter gt 0 and n_elements(iColor) gt 1 then loadct2,iColor(iter)
    if niter gt 0 then device,copy=[0,0,!d.x_vsize,!d.y_vsize,0,0,PixMapWin(iter)]
    wset,iWin & wshow,iwin
    cursor,xdummy,ydummy,/device,/nowait        ; To flush X-Windows buffer
    if xsz ne !d.x_vsize or ysz ne !d.y_vsize then goto,ReSize

    tvcrs,1                         ; Turn on cursor
    !err=0                          ; Will flicker between images every few sec
    while (!err eq 0) and abs(systime(1)-oldtime) lt dt $
     do cursor,x,y,/device,/nowait      ;Read cursor
ReadCursor2:
    oldtime=systime(1)
    if (!err ne 0) then begin           ; (button has been pushed)
      plots,[x],[y],psym=2,/device,color=byte(!d.table_size*.82) ; Draw cursor position
      Made(iter)=0                      ; So cursor position will be erased.
      if niter eq 0 then $
       print,string(byte(7)),format='(a1,6hClick!)'     ; Beep
      if niter ne 0 then $
       print,string(byte(7)),iter,format='(a1,21hClicked from image # ,i1)'
      !err=1 & while !err ne 0 do cursor,xx,yy,/device,/nowait ; Wait till up again
    endif else begin                    ;Switch images
      iter=iter+1 & if iter gt niter then iter=0
      goto,ReadCursor
    endelse

    black=range(0) & white=range(1)
    if n_elements(range) gt 2 then begin
      black=range(iter*2) & white=range(iter*2+1)
    endif
    jColor=iColor
```

```
   if n_elements(iColor) gt 1 then jColor=iColor(iter)

   if x ge 0 and y ge 0 then begin     ; Do things based on position.
     if y lt 20 then begin             ;Select option
if           x ge 4   and x lt 40  then begin ; Quit
  print,'Quit'
  if xpick ne 0 then begin
    xpick=-1 & ypick=-1  ; flag for not selected
  endif
  goto,endpoint
endif else if x ge 40  and x lt 90 then begin ; Zoom
  print,'Zooming'
  tv,bytarr(200,20)
  xyouts,0,0,'Click on corners',/device,size=.8,color=byte(!d.table_size*.82)
  iCol1Save=iCol1
  iRow1Save=iRow1
  !err=1 & while !err ne 0 do cursor,xx,yy,/device,/nowait
  cursor,iCol1,iRow1,/device
  while !err ne 0 do cursor,xx,yy,/device,/nowait
   plots,[iCol1],[iRow1],psym=2,/device,color=byte(!d.table_siz e*.82)
  if iCol1 gt 0 then iCol1=0 > (iCol1-2)
  if iGeom eq 2 or iGeom eq 5 then iCol1=nCol3-1-iCol1
  iRow1=iRow1-20
  if iGeom eq 2 or iGeom eq 7 then iRow1=nRow3-1-iRow1
  iCol1=long(iCol1/Fac)+iCol1Save
  iRow1=long(iRow1/Fac)+iRow1Save
  print,'iCol1,iRow1=',iCol1,iRow1
  !err=1 & while !err ne 0 do cursor,xx,yy,/device,/nowait
  cursor,iCol2,iRow2,/device
   plots,[iCol2],[iRow2],psym=2,/device,byte(!d.table_size*.82)
  if iCol2 gt 0 then iCol2=0 > (iCol2-2)
  if iGeom eq 2 or iGeom eq 5 then iCol2=nCol3-1-iCol2
  iRow2=iRow2-20
  if iGeom eq 2 or iGeom eq 7 then iRow2=nRow3-1-iRow2
  iCol2=long(iCol2/Fac)+iCol1Save
  iRow2=long(iRow2/Fac)+iRow1Save
  print,'iCol2,iRow2=',iCol2,iRow2
  !err=1 & while !err ne 0 do cursor,xx,yy,/device,/nowait
endif else if x ge 90 and x lt 125 then begin  ; Zoom out
  print,'Zooming out a little
  d =1 > (iCol2-iCol1)
  d2=1 > (iRow2-iRow1)
  if d/double(xsz) ge d2/double(ysz-20) then begin
    d2=1 > long(d *double(ysz-20)/(xsz   ))
  endif else begin
    d =1 > long(d2*double(xsz   )/(ysz-20))
  endelse
  t=long((iCol1+iCol2)/2) & iCol1=t-d  & iCol2=t+d
```

```
  t=long((iRow1+iRow2)/2) & iRow1=t-d2 & iRow2=t+d2
 endif else if x ge 125 and x lt 160 then begin  ; Full De-Zoom
  print,'Full De-Zoom
  iCol1=0 & iCol2=0 & iRow1=0 & iRow2=0
 endif else if x ge 160 and x lt 260 then begin  ; Adjust black
  print,'Adjusting black value
  d=white-black
  black=black-(210.d0-x)/50.d0*d
  if n_elements(range) lt 4 then begin   ;Store back in range
    range(0)=black & range(1)=white
  endif else begin
    range(2*iter)=black & range(2*iter+1)=white
  endelse
 endif else if x ge 260 and x lt 360 then begin  ; Adjust white
  print,'Adjusting white value
  d=white-black
  white=white+(x-310.d0)/50.d0*d
  if n_elements(range) lt 4 then begin   ;Store back in range
    range(0)=black & range(1)=white
  endif else begin
    range(2*iter)=black & range(2*iter+1)=white
  endelse
 endif else if x ge 370 and x lt 430 then begin  ; Keyboard Menu
   KBMenu,iWin,jColor,black,white,iCol1,iCol2,iRow1,iRow2,iFill , $
   iter,niter,nCol,nRow,iGeom,a,b,offset,nCol3,nRow3
  if n_elements(range) lt 4 then begin   ;Store back in range
    range(0)=black & range(1)=white
  endif else begin
    range(2*iter)=black & range(2*iter+1)=white
  endelse
  if n_elements(iColor) eq 1 then iColor=jColor else iColor(iter)=jColor
     ; Move left,right,up,down
 endif else if (x ge 430 and x lt 470 and (igeom eq 0 or  igeom eq 7)) or $
       (x ge 470 and x lt 520 and (igeom ne 0 and igeom ne 7)) then begin
  print,'Move
  temp=iCol1
  iCol1=0>(iCol1-fix(nCol2/2.1))
  if temp eq iCol1 then print,string(byte(7)),format='(a1,$)' ;beep
  iCol2=iCol2+(iCol1-temp)
 endif else if (x ge 430 and x lt 470 and (igeom ne 0 and igeom ne 7)) or $
       (x ge 470 and x lt 520 and (igeom eq 0 or  igeom eq 7)) then begin
  print,'Move
  temp=iCol2
  iCol2=(nCol-1)<(iCol2+fix(nCol2/2.1))
  if temp eq iCol2 then print,string(byte(7)),format='(a1,$)' ;beep
  iCol1=iCol1+(iCol2-temp)
 endif else if (x ge 520 and x lt 550 and (igeom ne 0 and igeom ne 5)) or $
       (x ge 550 and x lt 600 and (igeom eq 0 or  igeom eq 5)) then begin
```

```
  print,'Move
  temp=iRow1
  iRow1=0>(iRow1-fix(nRow2/2.1))
  if temp eq iRow1 then print,string(byte(7)),format='(a1,$)' ;beep
  iRow2=iRow2+(iRow1-temp)
endif else if (x ge 520 and x lt 550 and (igeom eq 0 or  igeom eq 5)) or $
      (x ge 550 and x lt 600 and (igeom ne 0 and igeom ne 5)) then begin
  print,'Move
  temp=iRow2
  iRow2=(nRow-1)<(iRow2+fix(nRow2/2.1))
  if temp eq iRow2 then print,string(byte(7)),format='(a1,$)' ;beep
  iRow1=iRow1+(iRow2-temp)
endif else if x ge 600 and x lt 650 and niter gt 0then begin    ; Back
  print,'Back one frame
  iter=iter-1 & if iter lt 0 then iter=niter
  dt=1e10
  goto,ReadCursor
endif else if x ge 650 and x lt 700 and niter gt 0then begin    ; Forward
  print,'Forward one frame
  iter=iter+1 & if iter gt niter then iter=0
  dt=1e10
  goto,ReadCursor
endif else if x ge 700 and x lt 750 and niter gt 0then begin    ; Speed
  dt=(750-x)*3./50
  print,'Nominal delay time=',dt
  goto,ReadCursor
endif else begin
  goto,ReadCursor
endelse
Made=bytarr(niter+1)           ; Images may have changed.
    endif else if xPick ne 0 and x ge 0 then begin ; Selected point
if iGeom eq 2 or iGeom eq 5 then x=nCol3-1-x
if iGeom eq 2 or iGeom eq 7 then y=nRow3-1-(y-20)+20
xPick=long(x/Fac)+iCol1
yPick=long((y-20)/Fac)+iRow1
goto,endpoint
    endif else begin
goto,ReadCursor
    endelse
    tv,bytarr(600,20)               ; Clear activity bar
    cursor,xx,yy,/device,/nowait      ; To flush X-Windows buffer
   ; (X-Windows buffers output requests.
   ;  It may not flush the buffer until
   ;  input requests--such as button
   ;  status--are made.
   ;  Therefore it may not draw things
   ;  on the screen, until you request
   ;  such information.)
```

```
    endif else begin
      goto,ReadCursor
    endelse
  endwhile
endpoint:
  if niter gt 0 then begin
    for iter=0,niter do begin
      if PixMapWin(iter) ge 0 then wdelete,PixMapWin(iter)
    endfor
  endif
  end
 ;------------------------------------------------------------ -----------
;=========================main procedure=============================
  pro dismain,flag                    ; Prompts for input, displays image(s).
     ; If flag is defined and non-zero, input
     ;  will come from dismain.input.
;Written by Mitchell R Grunes.

  if n_elements(flag) eq 0 then flag=0

  if flag eq 0 then begin
    print,'Ordinarally dis.pro is called with something like
    print,'  dis,image_array
    print,'But you must have called this via something like
    print,'  run dis.pro
    print,'  dismain
    print,'Therefore I will prompt for information
  endif else begin
    close,12
    openr,12,'dismain.input
  endelse

  ReadImFile,0,flag, a,iGeom            ; Prompt & read image.
  ReadImFile,1,flag, b,iGeom
  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 flag eq 0 then begin
    print,'You may also select initial portion of image to display:'
    print,'Cols to display (0,0 or 0,',sq(nColA-1),'=all):'
    read,icol1,icol2
```

```
  print,'Rows to display (0,0 or 0,',sq(nRowA-1),'=all):'
  read,irow1,irow2
endif else begin
  readf,12,icol1,icol2
  readf,12,irow1,irow2
endelse
select=[iCol1,iCol2,iRow1,iRow2]

if RankAr(b) eq 2 then begin
  if flag eq 0 then begin
    print,'Enter 0 to display 2nd image with same black and white values.
    print,'Enter 1 for seperate'
    read,i
  endif else begin
    readf,12,i
  endelse
endif else begin
  i=0
endelse

black=0.d0 & white=255.d0
if i eq 0 then begin
  if flag eq 0 then begin
    print,'Enter black,white values (0,0 to autoscale):
    read,black,white
  endif else begin
    readf,12,black,white
  endelse
  range=[black,white]
endif else begin
  if flag eq 0 then begin
    print,'Enter black,white values for first image (0,0 to autoscale):
    read,black,white
    print,'Enter black,white values for 2nd   image (0,0 to autoscale):
    read,black2,white2
  endif else begin
    readf,12,black,white
    readf,12,black2,white2
  endelse
  range=[black,white,black2,white2]
endelse

if flag eq 0 then print,'Display Geometry (',sq(igeom),'?):'
if flag eq 0 then read,iGeom else readf,12,iGeom

if flag eq 0 then print,'Color scheme (0?):'
if flag eq 0 then read,iColor else readf,12,iColor
```

```
  if !prompt eq 'IDL> ' then device,pseudo=8
  if !prompt eq 'IDL> ' then wait,.1
   dis,a,b,range=range,iGeom=iGeom,iColor=iColor,Select=Select, iFill=1
  if (!d.flags and 256) ne 0 then wdelete,0
  if (!d.flags and 256) ne 0 then wdelete,1
  end
 ;------------------------------------------------------------ -----------
;=========================another main procedure=============================
  pro  DisMenu,FilNam1,FilNam2,iType=iType,nCol=nCol,nRow=nRow,iCol 1=iCol1,iCol2=iCol2, $
    iRow1=iRow1,iRow2=iRow2,nHeader=nHeader,iGeom=iGeom,Black=Bl ack,White=White, $
   iColor=iColor,nFrame=nFrame          ; Menu for input, displays image(s).

     ; Not quite as many options as DisMain,
     ;  but a little easier to change things.
     ; All input arguements are optional:
     ; FilNam1=Name of file name to display.
     ; FilNam2=Name of second same size
     ;  file to display.
     ; iType=file type to display--see PrintImTypes.
     ; If FilNam1 is provided, but iType is not,
     ;  a type may be guessed from the suffix.
     ; 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=Display Geometry type (0,2,5, or 7).
     ;  Affects display, not col and row #'s.
     ; Black,White=Range of pixel values to
     ;  be displayed.
     ; iColor=loadct2 color scheme #.
     ; nFrame=# of images stored in file.
     ;  Only applies to raw data.
;Written by Mitchell R Grunes.
  if !prompt eq 'IDL> ' then device,pseudo=8
  if !prompt eq 'IDL> ' then wait,.1
  window,0,colors=256
  if n_elements(FilNam1) eq 0 then FilNam1=''
  if n_elements(FilNam2) eq 0 then FilNam2=''
  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)
      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 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 itype=32
      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=0
  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
  On_IOerror,Menu

Menu:
  if (!d.flags and 256) 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  Type of image            =',sq(iType)
  if iType le 19 then begin
    print,'4  # of Columns             =',sq(nCol)
    print,'5  # of Rows                =',sq(nRow)
  endif
  print,'6  Cols to load (0,0=all)=',sq(iCol1),',',sq(iCol2)
  print,'7  Rows to load (0,0=all)=',sq(iRow1),',',sq(iRow2)
  if iType ge 16 and iType le 19 then begin
    print,'8  # of header lines        =',sq(nHeader)
  endif else if iType le 15 then begin
    print,'8  # of header bytes        =',sq(nHeader)
  endif
  print,'9  Display Geometry         =',sq(iGeom)
  print,'10 Black,White pixel values =',sq(Black),',',sq(White)
  print,'11 Color Scheme             =',sq(iColor)
```

```
if iType le 19 then $
  print,'12 # 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
  PrintImTypes
  read,'Type of image:',iType
endif else if choice eq 4 then begin
  read,'# of Columns:',nCol
endif else if choice eq 5 then begin
  read,'# of Rows:',nRow
endif else if choice eq 6 then begin
  read,'Cols to load (0,0=all):',iCol1,iCol2
endif else if choice eq 7 then begin
  read,'Rows to load (0,0=all):',iRow1,iRow2
endif else if choice eq 8 then begin
  read,'Header Size:',nHeader
endif else if choice eq 9 then begin
  PrintGeomTypes
  read,'Geometry Type:',iGeom
endif else if choice eq 10 then begin
  read,'Black,White pixel values:',Black,White
endif else if choice eq 11 then begin
  read,'Color Scheme:',iColor
endif else if choice eq 12 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, $
   iType,nheader,nFrame
  if FilNam2 gt ' ' then $
   ReadImage,filnam2,b,nCol,nRow,iCol1,iCol2,iRow1,iRow2, $
   iType,nheader,nFrame
  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*****

   Range=[Black,White]

   if !prompt eq 'IDL> ' then device,pseudo=8
   if !prompt eq 'IDL> ' then wait,.1
    dis,a,b,range=range,iGeom=iGeom,iColor=iColor,Select=Select, iFill=iFill

   if (!d.flags and 256) ne 0 then wdelete,0
   if (!d.flags and 256) ne 0 then wdelete,1

   Black=Range(0)
   White=Range(1)
  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
   return
  endif
  goto,Menu
  end
--------------CUT HERE-----------------
 ------------------------------------------------------------- --------
```

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