
Subject: DIS and DISMENU image display programs: 2/4

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: 2/4

This is a continuation of dis.pro from prior mail message.

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

```
-----
;-----
pro Dis,a,b,iWin=iWin,iColor=iColor,black=black,white=white,select=select, $
  iFill=iFill,xpick=xpick,ypick=ypick,iGeom=iGeom,dt=dt,Offset =Offset, $
  xsz=xsz,ysz=ysz,title=title,xpos=xpos,ypos=ypos, $
  xmark=xmark,ymark=ymark,KeepWindow=KeepWindow,Edge=Edge
      ; 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).
      ; If this is a string, we will
      ; assume you really meant to call
      ; dismenu.
      ; b=Flickered image, if defined and 2
      ; dimensional. Must be same shape
      ; as a. Do not use if a is a movie.
      ; Optional.
      ; 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.
      ; black,white=Values to display as
      ; black & white. If black ge white
      ; or undefined, will be set to the
      ; min and max.
      ; You may also set black and white to
      ; be vectors to get seperate values
      ; for multiple images.
      ; select=[iCol1,iCol2,iRow1,iRow2]
      ; =selected portion to display.,
      ; if defined and iCol1 lt iCol2
      ; and iRow1 lt iRow2.
      ; iFill=0 integer zoom factor,
      ; 1 or undefined to fill window.
```

```

; xpick,ypick: see below
; 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.
; (default=7).
; (Default used to be 0.)
; dt=flicker delay time, default=1 sec
; Set it to 1E10 if you want to
; freeze frame.
; Offset=# to add to row #'s in prints
; xsz,ysz=size of window to open.
; title=window title.
; Can be a vector of titles, for
; multiple images.
; xpos,ypos=window position on screen.
; xmark,ymark=coordinates of points to
; be marked with bright crosses.
; If this is two dimensional (2nd
; dimension=2), different parts will
; be used for a and b.
; xmark,ymark not yet tested.
; KeepWindow is set if the window does
; not need to be created.
; Edge=0 is normal, 1=Apply Sobel Edge
; Detection, 2=Roberts Edge
; Detection.
; -----OUTPUTS-----
; Many variables other than a and b
; may be modified.
; xpick,ypick=Point selected, if any.
; The option of selecting a point
; only exists if xpick is initially
; non-zero.

```

;Written by Mitchell R Grunes.

```

if VarTyp(a) eq 7 then begin
  print, '***YOU REALLY MEANT TO CALL DISMENU, NOT DIS***'
  DisMenu,a,b,iGeom=iGeom,Black=Black,White=White,iColor=iColor r
  return
endif

!order=0
GetSize,a,nCol,nRow

```

```

s=size(B)
; 0 if B undefined or invalid, else 1.
UsingB=rankAr(B) eq 2 and s(0) eq 2 and s(1) eq nCol and s(2) eq nRow
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=7

if n_elements(Edge) eq 0 then Edge=0
if n_elements(black) eq 0 or n_elements(white) eq 0 then begin
  if Edge eq 0 then begin
    black=min(a)
    white=max(a)
  endif else begin
    if Edge eq 1 then aa=Sobel(a) else aa=Roberts(a)
    black=min(aa)
    white=max(aa)
    aa=0
  endelse
endif

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(KeepWindow) eq 0 then KeepWindow=0
if KeepWindow then xsz=!d.x_vsize
if KeepWindow then ysz=!d.y_vsize
UseContour=0

```

```

if (!d.flags and 256) eq 0 then iWin=-1; Non-windowing system--not supported
    ; any more.
title2=title(0)
if n_elements(title) ne 1 then title2=""
if iWin ge 0 then begin
    ; Initialize windowing systems.
    if !prompt eq 'IDL> ' and !d.name ne 'WIN' then device,pseudo=8
    if !prompt eq 'IDL> ' then wait,.1
    if n_elements(xsz) eq 0 or n_elements(ysz) eq 0 then begin
        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
    endif
    ; Following statement causes an error
    ; when used with X-windows emulators
    ; under Microsoft windows. Some color
    ; schemes would work if you removed
    ; it.
    if KeepWindow eq 0 then $
        window,iWin,colors=256,retain=2,xpos=xpos,ypos=ypos,xsize=xsize, $
        ysize=ysz,title=title2
        xsz=!d.x_vsize & ysz=!d.y_vsize
        wset,iWin
    endif
    n=!d.table_size
    Greybar=congrid(reform(indgen(n),n,1),100,20)
    wset,iWin
    erase
    loadct2,iColor(n_elements(iColor)-1<niter)
    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.
    PixMapWin=replicate(-1,niter+1)
    ; No pixmap windows yet?

    wset,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           ; Swap to place in order.
  i=iCol2
  iCol2=iCol1
  iCol1=i
endif

if iRow2 lt iRow1 then begin
  i=iRow2
  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.
title2=title(0)
if n_elements(title) gt 1 then title2=title(n_elements(title)-1<iter)
if title2 gt ' ' then begin
  print,string(reform(byte(replicate('*',strlen(title2)+4))))
  print,'* ',title2,'*'
  print,string(reform(byte(replicate('*',strlen(title2)+4))))
endif
wset,iWin
if xsz ne !d.x_vsize or ysz ne !d.y_vsize then begin
  xsz=!d.x_vsize & ysz=!d.y_vsize
  if xsz ne -1 then print,'Sizing pixmaps:',xsz,ysz
  if iWin ge 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
  Made=bytarr(niter+1)                ; No display images made yet.

```

```

endif

wset,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)

black=double(black) & white=double(white)
if Made(iter) eq 0 then begin          ; Prepare image pixmaps.
  jColor=iColor(n_elements(iColor)-1<iter)
  if RankAr(xmark) eq 2 then begin
    xm=xmark(*,iter)
    ym=ymark(*,iter)
  endif else if n_elements(xmark) gt 0 then begin
    xm=xmark
    ym=ymark
  endif
  black2=black(n_elements(black)-1<iter)
  white2=white(n_elements(white)-1<iter)
  if niter eq 0 or (UsingB and iter eq 0) then begin
    Dis1,a,iter,niter, Fac,nCol3,nRow3,iWin=PixMapWin(iter),hWin=iWin, $
    iColor=jColor,black=black2,white=white2,select=select,iFill= iFill, $
    iGeom=iGeom,Offset=Offset,xmark=xm,ymark=ym,title=title2, $
    UseContour=Usecontour,Edge=Edge,xLeft,yBot
  endif else if UsingB then begin
    Dis1,b,iter,niter, Fac,nCol3,nRow3,iWin=PixMapWin(iter),hWin=iWin, $
    iColor=jColor,black=black2,white=white2,select=select,iFill= iFill, $
    iGeom=iGeom,Offset=Offset,xmark=xm,ymark=ym,title=title2, $
    UseContour=Usecontour,Edge=Edge,xLeft,yBot
  endif else begin
    Dis1,reform(a(*,*,iter)),iter,niter, Fac,nCol3,nRow3, $
    iWin=PixMapWin(iter),hWin=iWin, $
    iColor=jColor,black=black2,white=white2,select=select,iFill= iFill, $
    iGeom=iGeom,Offset=Offset,xmark=xm,ymark=ym,title=title2, $
    UseContour=Usecontour,Edge=Edge,xLeft,yBot
  endelse
  Made(iter)=1
  black(n_elements(black)-1<iter)=black2
  white(n_elements(white)-1<iter)=white2
  wset,iWin
  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)

  device,copy=[0,0,!d.x_vsize<xsz,!d.y_vsize<ysz,0,0, $
  PixMapWin(iter)]

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
if (!err ne 0) then goto,ReadCursor2      ; (button has been pushed)

tvcrs,1
ReadCursor:                               ; Set up display to read
                                           ; cursor,delay.
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(n_elements(iColor)-1<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                                    ; Flicker time loop.
oldtime=systime(1)                        ; Delay a while.
while (!err eq 0) and abs(systime(1)-oldtime) lt dt $
do cursor,x,y,/device,/nowait            ; Read cursor
ReadCursor2:                              ; Actually read cursor.
if (!err ne 0) then begin                 ; (button has been pushed)
plots,[x],[y],psym=2,/device,color=byte(!d.table_size*.82)
                                           ; Mark cursor position.
                                           ; Beep (doesn't always work)
if niter eq 0 then print,string(byte(7)),format='(a1,6hClick!)'
if niter gt 0 then $
print,string(byte(7)),iter,format='(a1,20hClicked from image # ,i2)'
!err=1 & while !err ne 0 do cursor,xx,yy,/device,/nowait ; Wait till up
endif else begin                          ; Switch images
iter=iter+1 & if iter gt niter then iter=0
goto,ReadCursor
endelse

jColor=iColor(n_elements(iColor)-1<iter)
black2=black(n_elements(black)-1<iter)
white2=white(n_elements(white)-1<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,'Exiting image display.
if xpick ne 0 then begin

```

```

    xpick=-1 & ypick=-1 ; flag for not selected
endif
wshow,iWin,0
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_size*.82)
if iCol1 gt 0 then iCol1=0 > (iCol1-xLeft)
if iGeom eq 2 or iGeom eq 5 then iCol1=nCol3-1-iCol1
iRow1=0 > (iRow1-yBot)
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-xLeft)
if iGeom eq 2 or iGeom eq 5 then iCol2=nCol3-1-iCol2
iRow2=0 > (iRow2-yBot)
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
Made=bytarr(niter+1) ; Images may have changed.
erase ; So can tell was pushed.
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
Made=bytarr(niter+1) ; Images may have changed.
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
Made=bytarr(niter+1)          ; Images may have changed.
endif else if x ge 160 and x lt 260 then begin ; Adjust black
print,'Adjusting black value
d=white2-black2
black2=black2-(210.d0-x)/50.d0*d
black(n_elements(black)-1<iter)=black2
white(n_elements(white)-1<iter)=white2
if n_elements(black) eq 1 then Made=bytarr(niter+1) $
else Made(iter)=0
endif else if x ge 260 and x lt 360 then begin ; Adjust white
print,'Adjusting white value
d=white2-black2
white2=white2+(x-310.d0)/50.d0*d
black(n_elements(black)-1<iter)=black2
white(n_elements(white)-1<iter)=white2
if n_elements(white) eq 1 then Made=bytarr(niter+1) $
else Made(iter)=0
endif else if x ge 370 and x lt 430 then begin ; Keyboard Menu
KMenu,iWin,iColor,black,white,iCol1,iCol2,iRow1,iRow2,iFill , $
iter,niter,nCol,nRow,iGeom,a,b,offset,nCol3,nRow3,UseContour , $
Edge,xLeft,yBot
; Move left,right,up,down
Made=bytarr(niter+1)          ; Images may have changed.
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
temp=iCol1
iCol1=0>(iCol1-fix(nCol2/2.1))
if temp eq iCol1 then print,string(byte(7)),format='(a1,$)' ;beep
iCol2=iCol2+(iCol1-temp)
Made=bytarr(niter+1)          ; Images may have changed.
endif else if (x ge 430 and x lt 470 and (igeom ne 0 and igeom ne 7))$
or (x ge 470 and x lt 520 and (igeom eq 0 or igeom eq 7)) then begin
temp=iCol2
iCol2=(nCol-1)<(iCol2+fix(nCol2/2.1))
if temp eq iCol2 then print,string(byte(7)),format='(a1,$)' ;beep
iCol1=iCol1+(iCol2-temp)
Made=bytarr(niter+1)          ; Images may have changed.
endif else if (x ge 520 and x lt 550 and (igeom ne 0 and igeom ne 5))$
or (x ge 550 and x lt 600 and (igeom eq 0 or igeom eq 5)) then begin
temp=iRow1
iRow1=0>(iRow1-fix(nRow2/2.1))
if temp eq iRow1 then print,string(byte(7)),format='(a1,$)' ;beep
iRow2=iRow2+(iRow1-temp)
Made=bytarr(niter+1)          ; Images may have changed.
endif else if (x ge 520 and x lt 550 and (igeom eq 0 or igeom eq 5))$
or (x ge 550 and x lt 600 and (igeom ne 0 and igeom ne 5)) then begin

```

```

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)
Made=bytarr(niter+1) ; Images may have changed.
endif else if x ge 600 and x lt 650 and niter gt 0 then 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 0 then 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 0 then begin ; Speed
dt=(750-x)*3./50
print,'Nominal delay time=',dt
goto,ReadCursor
endif else begin
goto,ReadCursor
endelse
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
print,'Picked:',sq(xPick),',',sq(yPick)
goto,endpoint
endif else begin
goto,ReadCursor
endelse
tv,bytarr(800,20) ; 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
```

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

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