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

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: 1/4

Here is an image display program for many type of image, which runs under IDL or PV-WAVE, contained in a bunch of files:

```
dis.pro
dismenu.pro
getsize.pro
loadct2.pro
outimage.pro
printgeomtypes.pro
printimtypes.pro
rankar.pro
rdfil.pro
rdsdf.pro
read_pgm.pro
read_sgi_image.pro
readimage.pro
sq.pro
temporary.pro
usercol.pro
vartyp.pro
wrfil.pro
write_pgm.pro
write_sgi_image.pro
wrsdf.pro
```

The files break near the "CUT HERE" lines.

Note that the first procedure, dis.pro, is broken across two mail messages.

-----CUT HERE----dis.pro-----

```
; This is Mitchell R Grunes's image display program (dis).
; A bit slow and kludgy, uses a lot of memory. But it does lots of things
; "right".
```

```
; -----WARNING-----
; Reasonable attempts were made to
; work in multiple X window
; environments including X-windows
; and microsoft windows. However,
; this was hampered by the desire to
; use 256-color pseudo-color maps:
; (1) It is probably not worth using
; under X-window systems such as
```

```

; HP Vue where you have to click
; 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.
; (2) If used from a station (such as
; a PC running an X-window
; emulator under Microsoft
; windows) which does not support
; 256 color maps, the initial
; window call in routine may
; cause an error. Some color
; schemes will work more or less
; correctly if you remove that
; line.
;-----
; Revision date 5/19/97.
;-----
; (main procedure is set up to do image display)
;
; Use something like
;   dis,image_array           ; display image array
;                               ; (can use other parameters)
; or
;   dismenu,'filename'       ; display image file
;-----
; Table of Contents:
;
; pro AutoScale,a,iWin, black,white  ; Autoscale image a, to produce
;                                     ; black,white.
; pro Dis1,a,iter,niter,iWin=iWin... ; Internal routine to display 1 image
;                                     ; in a window.
; pro KBMenu,iWin,iColor,black,white... ; Keyboard Menu.
; pro Dis,a,b,iWin=iWin,iColor=iColor...; Display Image(s) in x-windows or
;                                     ; PC windows.
;-----
; pro AutoScale,a,iWin, black,white  ; Autoscale image a, to produce
;                                     ; black,white.
;                                     ; (iWin needed for wshow test.)
; Written by Mitchell R Grunes.
; white=double(max(a,min=black)) & black=double(black)
; bot=black
; top=white
; if iWin ge 0 then wshow,iWin,0
; print,'Min,max pixel value=',black,white
; print,'Enter desired black,white values (0,0=autoscale):
; read,black,white

```

```

if black ge white and bot ne top then begin
  print,'Autoscaling to .5% cutoff points'
  if VarTyp(a) lt 4 then begin
    bot=long(bot)
    top=long(top)
    bin=1 > (top-bot)/4000
  endif else begin
    bin= (top-bot)/4000
  endelse

  hist=histogram(a,min=bot,max=top,bin=bin)
  for i=1,n_elements(hist)-1 do hist(i)=hist(i)+hist(i-1)

  nPixCut=n_elements(a)*.005
  black=(where(hist ge nPixCut))(0)
  black=black*bin+bot

  nPixCut=n_elements(a)*.995
  white=(where(hist ge nPixCut))(0)
  white=white*bin+bot

  print,'black,white=',black,white
endif
white=white > (black+1d-4)
if iWin ge 0 then wset,iWin
end
;-----
pro Dis1,a,iter,niter, Fac,nCol3,nRow3,iWin=iWin,hWin=hWin, $
  iColor=iColor,black=black,white=white,select=select,iFill=iFill, $
  iGeom=iGeom,Offset=Offset,xmark=xmark,ymark=ymark,title=title, $
  UseContour=Usecontour,Edge=Edge,xLeft,yBot
; Internal routine to display 1 image
; in a window.
; -----INPUTS-----
; a=Image to display.
; iter=Image # (0-origin).
; niter=Number of images-1.
; iWin=Window # to display image in.
; This will generally be a pixmap
; window.
; hWin=Window which needs to be hidden
; if text questions need to be asked.
; iColor=loadct2 color map #.
; black,white =Values to display as
; black and white. Will autoscale if
; black ge white.
; select=[iCol1,iCol2,iRow1,iRow2]

```

```

; =selected portion to display.,
; if defined and iCol1 lt iCol2
; and iRow1 lt iRow2.
; iFill=0 for integer zoom,
; 1 or undefined to fill window.
; iGeom=Screen Geometry.
; Offset=# to add to row #'s in prints
; xmark,ymark=coordinates of points to
; be marked. Not yet tested.
; title=title to draw at top of window
; UseContour=0 for image display.
; 1 to display contours
; 2 to display shaded surface.
; Edge=0 is normal, 1=Apply Sobel Edge
; Detection,2=Roberts Edge Detection.
; -----OUTPUTS-----
; iWin,iColor,black,white,select,iFill
; may be modified.
; Fac=Zoom factor
; nCol3,nRow3=size of displayed area.
; xLeft,yBot=Lower left display corner
; position.

```

;Written by Mitchell R Grunes.

```
!order=0
```

```
GetSize,a,nCol,nRow
```

```
iCol1=select(0)
```

```
iCol2=select(1)
```

```
iRow1=select(2)
```

```
iRow2=select(3)
```

```
n=!d.table_size
```

```
Greybar=congrid(reform(indgen(n),n,1),100,20)
```

```
if iWin ge 0 then wset,iWin
```

```
VTa=vartyp(a)
```

```
nCol2=iCol2-iCol1+1
```

```
nRow2=iRow2-iRow1+1
```

```
xsz=!d.x_vsize & ysz=!d.y_vsize ; Screen size
```

```
if UseContour eq 0 then begin ; Will draw image inside axis plot.
```

```
  xrange=[iCol1,iCol2+1]
```

```
  if iGeom eq 2 or iGeom eq 5 then xrange=[iCol2+1,iCol1]
```

```
  if iGeom eq 3 or iGeom eq 4 then xrange=[iRow1,iRow2+1]
```

```
  if iGeom eq 1 or iGeom eq 6 then xrange=[iRow2+1,iRow1]
```

```
  yrange=[iRow1,iRow2+1]
```

```

if iGeom eq 2 or iGeom eq 7 then yrange=[iRow2+1,iRow1]
if iGeom eq 1 or iGeom eq 4 then yrange=[iCol1,iCol2+1]
if iGeom eq 3 or iGeom eq 6 then yrange=[iCol2+1,iCol1]
xrange=xrange+offset-.5 & yrange=yrange-.5 ; So numbers will be centered.
dx=abs(xrange(0)-xrange(1)) & dy=abs(yrange(0)-yrange(1))
Factor=(.96*xsz/dx) < (.94*ysz/dy)
plot,xrange,yrange,xrange=xrange,yrange=yrange,/nodata,title ='!A'+title, $
  xstyle=1,ystyle=1,ticklen=-.006,position=[.035,.035,.035+Fac tor*dx/xsz, $
  .035+Factor*dy/ysz],charsize=.5
xLeft =long( (!x.s(1)*xrange(0)+!x.s(0))*!d.x_vsize +.5)
xRight=long( (!x.s(1)*xrange(1)+!x.s(0))*!d.x_vsize +.5)
yBot =long( (!y.s(1)*yrange(0)+!y.s(0))*!d.y_vsize +.5)
yTop =long( (!y.s(1)*yrange(1)+!y.s(0))*!d.y_vsize +.5)
      ; Enlargement factor
Fac=(double(abs(xRight-xLeft))/dx) < (double(abs(yTop-yBot))/dy)
endif else begin      ; No scaling for contour or shade_surf
  Fac=1
  xLeft=20
  yBot=20
endelse
FacOld=Fac
if Fac ge 1 then begin      ; enlarge
  if iFill eq 0 then Fac=long(Fac)
  xs2=long(nCol2*Fac+.5)      ; size of enlarged 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
if iFill eq 0 then begin
  Factor=Factor*Fac/FacOld
  plot,xrange,yrange,xrange=xrange,yrange=yrange,/nodata,title ='!A'+title, $
    xstyle=1,ystyle=1,ticklen=-.006,position=[.035,.035,.035+Fac tor*dx/xsz, $
    .035+Factor*dy/ysz],charsize=.5
endif

```

```

c=a((iCol1<i2):i2,(iRow1<j2):j2) ; Extract image section

if Edge eq 1 then c=Sobel(c) else if Edge eq 2 then c=Roberts(c)

if black ge white then begin ; If bad, derive from image section
;black=double(min(c))
;white=double(max(c))
AutoScale,c,hWin, black,white
wset,iWin
endif

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
white=white > (black+1d-4)

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)

; 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) ; Graphics locations
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 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
iy=0 ; (save memory)

```

```

endif
endif

if Fac gt 50 then begin          ; Save original values for print.
  cSave=c
  xSave=long(c*0)
  ySave=xSave
  for i=iCol1,iCol2 do xSave(i-iCol1,*)=i
  for j=iRow1,iRow2 do ySave(*,j-iRow1)=j+Offset
endif

if Fac gt 1 and iFill ne 0 then begin ; Resize image
  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 UseContour eq 0 then begin
  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
endif
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
endif
                                ; Restore graphics pixels
if graphsave(0) ge 0 then c(ix)=graphsave
ix=0 & & graphsave=0                                ; Save memory

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

erase

if UseContour eq 0 then begin
    tv,c,xLeft,yBot
    plot,xrange,yrange,xrange=xrange,yrange=yrange,/nodata,title ='!A'+title, $
    xstyle=1,ystyle=1,ticklen=-.006,position=[.035,.035,.035+Fac tor*dx/xsz, $
    .035+Factor*dy/ysz],charsize=.5,/noerase

    if Fac gt 50 then begin                                ; Draw values at hi zoom.
        GetSize,cSave, nCol4,nRow4
        for j=0,nRow4-1 do begin
            for i=0,nCol4-1 do begin
                if cSave(i,j) gt (black2+white2)*.5 then temp=0 else temp=top
                                ; Draw Pixel value.
                xyouts,xSave(i,j),ySave(i,j),sq(float(cSave(i,j))),size=Fac* .01, $
                color=temp,align=.5
            endfor
        endfor
    endif
endif

```

```

    endfor
  endfor
  cSave=0 & xSave=0 & ySave=0      ; (save memory)
endif
endif else if UseContour eq 1 then begin
  contour,c,title=title,xstyle=1,ystyle=1,zstyle=1
endif else begin
  shade_surf,c,title=title,zrange=[black2,white2],xstyle=1,ystyle=1,zstyle=1
endelse
Getsize,c, nCol3,nRow3

for i=0,n_elements(xmark)-1 do begin ; Mark xmark,ymark points on screen.
  ix=xmark(i)+.5
  iy=ymark(i)+.5
  if ix ge iCol1 and ix le iCol2 and iy ge iRow1 and iy le iRow2 then begin
    ix=long((ix-iCol1)*Fac+.5)+2
    iy=long((iy-iRow1)*Fac+.5)+20
    if iGeom eq 2 or iGeom eq 5 then ix=xs2b-1-ix
    if iGeom eq 2 or iGeom eq 7 then iy=ys2b-1-iy
    plots,[ix],[iy],psym=2,device=UseContour eq 0,color=n-1,symsize=1.5, $
      thick=2
    plots,[ix],[iy],psym=1,device=UseContour eq 0,color=0,symsize=1.5, $
      thick=2
  endif
endfor

top=byte(top*.82)      ; A visible color in almost all color schemes.
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 KBMenu,iWin,iColor,black,white,iCol1,iCol2,iRow1,iRow2,iFill , $
  iter,niter,nCol,nRow,iGeom,im1,im2,offset,nCol3,nRow3,UseContour, $
  Edge,xLeft,yBot
                                ; Keyboard Menu.
;Written by Mitchell R Grunes.
menu1:
  tvlct,/get,red,green,blue
  tvcrs,0,0

  if iWin ge 0 then wshow,iWin,0

  if niter eq 0 then print,'---Keyboard Menu---'
  if niter gt 0 then print,'---Keyboard Menu (from image # ',sq(iter),'---'
  print,'1 Load Color Scheme #=',sq(iColor)
  print,'2 Black,white pixel val=',sq(black),',',sq(white)
  print,'3 Columns to display  =',sq(iCol1),',',sq(iCol2)
  print,'4 Rows to display    =',sq(iRow1+offset),',',sq(iRow2+offset)
  print,'5 0=integer zoom,1=iFill window=',sq(iFill)
  print,'6 Screen Geometry    =',sq(iGeom)
  print,'7 Save image to file
  print,'8 Print help for things to click on'
  print,'9 Operating system commands'
  print,'10 Find average pixel value in region'
  print,'11 Edge Detection     =',sq(Edge)
  print,'12 0=Image 1=Contour 2=Shaded Surface =',sq(UseContour)
  print,'13 Print pixel values'
  print,'0 Exit Menu'
  on_ioerror,menu1
  read,ichoice

  if ichoice ne 0 and ichoice lt 5 then print,'Enter new value(s):'
  if ichoice eq 1 then begin
    if niter gt 0 then begin
      i=0
      if niter gt 0 then $
        read,'Enter 0 to use the same values for all images, else 1:',i
      if i eq 0 then iColor=iColor(0)
      if i ne 0 and n_elements(iColor) lt niter+1 then $
        iColor=[iColor,replicate(iColor(0),niter+1-n_elements(iColor))]
      if n_elements(iColor) gt niter+1 then iColor=iColor(0:niter)
    endif
    print,'Color Schemes:
    print,'0=B-W LINEAR; 1=(black/)BLUE/WHITE; 2=(black-)GRN-RED-BLU-WHT;
    print,'3=RED TEMPERATURE black/red/white; 4=black/BLEUE/GREEN/RED/YELLOW;
    print,'5=STD GAMMA-II black/blue/red/yellow/white;

```

```

print,'6=PRISM black/red/green/blue/black; 7=black-RED-PURPLE-white;
print,'8=GREEN/WHITE LINEAR black/green/white; 9=GRN/WHT EXPONENTIAL;
print,'10=black-GREEN-blue-PINK-white; 11=black-BLUE-RED-white;
print,'12=16 LEVEL; 13=RAINBOW black-purple-blue-green-red;
print,'14=STEPS (a mess); 15=STERN SPECIAL (ugly);
print,'50 is the user scheme defined by usercol.pro. By default it is
print,' grunes"s black-spectral-white.
print,'51 will select 50, and call usercol to edit it.
print,'52=grunes"s scheme used to represent certain feature classes.
print,'96=detect.pro scheme where 0-121 are a grey scale,
print,' 122-127 are red,yellow,green,light blue,blue,purple
print,'97 is a color scheme where 0-249 are a grey scale,
print,' 250-255 are red,yellow,green,light blue,blue,purple
print,'98 is a land/water map scheme where 0-127=monochrome grey,
print,' 128-254=monochrome yellow, 255=red.
print,'99=monochrome (0-252), red,green,blue (253-255).
print,' 0-127=grey shades, 128-254=yellow shades, 255=red.
print,'96-99 are only usable for byte images, in 256 color windows.'
print,'Current value(s)=',sq(iColor)
read,'ColorScheme(s):',iColor
if iWin ge 0 then wshow,iWin
loadct2,iColor(n_elements(iColor)-1<iter)
if iWin ge 0 then wshow,iWin,0
tvcrs,10,5
endif else if ichoice eq 2 then begin
print,'(use black=white=0 to autoscale)
if niter gt 0 then begin
i=0
if niter gt 0 then $
read,'Enter 0 to use the same values for all images, else 1:',i
if i eq 0 then black=black(0)
if i eq 0 then white=white(0)
if i ne 0 and n_elements(black) lt niter+1 then $
black=[black,replicate(black(0),niter+1-n_elements(black))]
if i ne 0 and n_elements(white) lt niter+1 then $
white=[white,replicate(white(0),niter+1-n_elements(white))]
if n_elements(black) gt niter+1 then black=black(0:niter)
if n_elements(white) gt niter+1 then white=white(0:niter)
endif
print,'Black pixel value(s):',sq(Black)
print,'White pixel value(s):',sq(White)
black=double(black) & white=double(white)
read,'New black value(s):',black
read,'New white value(s):',white
endif else if ichoice eq 3 then begin
print,'(possible range=0,',sq(nCol-1),')'
read,iCol1,iCol2
endif else if ichoice eq 4 then begin

```

```

print,'(remember bottom=row 0)
print,'(possible range=',sq(offset),',',sq(nRow-1+offset),')'
read,iRow1,iRow2
iRow1=iRow1-offset & iRow2=iRow2-offset
endif else if icheice eq 5 then begin
  read,'(0=integer zoom 1=fill window):',iFill
endif else if icheice eq 6 then begin
  PrintGeomTypes
  read,iGeom
endif else if icheice eq 7 then begin
  print,'Warning: Image will reflect status before entering this menu!
  read,'0=Image only; 1=With Buttons:',WithButton
  if WithButton then begin
    aSave=tvrd(0,0,!d.x_vsize,!d.y_vsize)
  endif else if UseContour ne 0 then begin
    aSave=tvrd(0,20,!d.x_vsize,!d.y_vsize)
  endif else begin
    aSave=tvrd(xLeft,yBot,nCol3,nRow3)
    GetSize,aSave, nCol3,nRow3
    while total(aSave(nCol3-1,*) ne 0) eq 0 do nCol3=nCol3-1
    aSave=aSave(0:nCol3-1,*)
    while total(aSave(*,nRow3-1) ne 0) eq 0 do nRow3=nRow3-1
    aSave=aSave(*,0:nRow3-1)
  endelse
  GetSize,aSave, nCol4,nRow4
  print,'This is a byte image of size ',nCol4,nRow4
  nam=""
  read,'Name of file to write to:',nam
  if icolor eq 0 then pseudo=0 else pseudo=1
  format=-99 & OutImage,format,nam,aSave,pseudo=pseudo
endif else if icheice eq 8 then begin
  print,'---DIS: Things to Click On---'
  print,'Quit =Exit display routine
  print,'Zoom =Zoom in on image--you pick corners
  print,'Out =Zoom out factor of 2
  print,'Full =De-Zoom to full image
  print,'Keybd =Keyboard Entry of parameters and options
  print,"
  print,'Greyscale:
  print,'1st quarter=left vertical line to black end of greyscale.
  print,' Clicking here moves the black threshold value down,
  print,' so fewer things will be black.
  print,'2nd quarter=black end of greyscale to middle vertical line.
  print,' Clicking here sets the black threshold to the chosen shade,
  print,' so more things will be black.
  print,'3rd quarter=middle vertical line to white end of greyscale.
  print,' Clicking here sets the white threshold to the chosen shade,
  print,' so more things will be white.

```

```

print,'4th quarter=white end of greyscale to right vertical line.
print,' Clicking here moves the white threshold value up,
print,' so fewer things will be white.
endif else if icoice eq 9 then begin
  print,'End this mode with  exit
  spawn
endif else if icoice eq 10 then begin
  print,'If you want to clip the image(s) to a range first,
  print,' input (lo,hi) clip range, else input (0,0):
  read,lo,hi
  for iiter=0,niter do begin
    if RankAr(im1) eq 3 then begin
      test=reform(im1(iCol1:iCol2,iRow1:iRow2,iiter))
    endif else if iiter eq 0 then begin
      test=im1(iCol1:iCol2,iRow1:iRow2)
    endif else begin
      test=im2(iCol1:iCol2,iRow1:iRow2)
    endelse
    if lo lt hi then test=lo > test < hi
    print,'Average value for image ',sq(iiter),'=',total(test) $
      /n_elements(test)
  endfor
endif else if icoice eq 11 then begin
  read,'Edge detection (0=no,1=Sobel,2=Roberts):',Edge
endif else if icoice eq 12 then begin
  print,'NOTE: the positioning on countour plots does not work well.
  read,'0=image display, 1=contour plots, 2=shaded surface:',UseContour
endif else if icoice eq 13 then begin
  if (iCol2-iCol1+1)*(iRow2-iRow1+1) gt 5000 then begin
    print,'*****Sorry--can not print over 5000 pixels for your safety*****
  endif else begin
    for iiter=0,niter do begin
      print,"
      if RankAr(im1) eq 3 then begin
        test=reform(im1(*,*,iiter))
      endif else if iiter eq 0 then begin
        test=im1
      endif else begin
        test=im2
      endelse
      for j=iRow1,iRow2 do begin
        print,'image',sq(iiter),'(',sq(iCol1),':',sq(iCol2),',', $
          sq(j),')=',test(iCol1:iCol2,j)
      endfor
    endfor
  endelse
endif
if icoice ne 0 then goto,menu1

```

```
if iWin ge 0 then wset,iWin  
on_ioerror,null  
end
```

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

File dis.pro is longer than that.
Continued in next mail message.

----- -

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