
Subject: Here is an image/movie display program (1/2).

Posted by [grunes](#) on Tue, 07 Jan 1997 08:00:00 GMT

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

dis.pro: 1/2

My contribution to the good of the IDL/WAVE community.

Place the contents between the two "CUT HERE" lines into file dis.pro, add in the stuff from the second post, and type

IDL (or WAVE)

.run dis

Then type

dis,a

to display the image in array a, or type

dismenu,'filename'

to display the image in file <filename> (in which case you may have to set other parameters, like image format, then type 50 to perform the display.).

If you have a third dimension, it is the number of frames.

Not as classy looking a display as the widget stuff, rather long and a bit slow and memory-intensive, but works fairly well.

(You may also want to look into some free non-idl/wave image display programs--I like xv and saimage--you'll have to search the net.)

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

; This is Mitchell R Grunes's library of commonly used routines.

; You may include it in any procedure or the current workspace by the command:

; @dis.pro

;-----

; Revision date 1/6/97.

;-----

;WARNING--If you are using an old version of IDL that does not

; include TEMPORARY, you must uncomment the following function:

;;temporary.pro to use if using a version of IDL or PV-WAVE that does

;; not include temporary function

;; By mitchell r grunes

; function temporary,a

; return,a

; end

;-----

; Table of Contents:

; pro GetSize,a, nCol,nRow ; Output # of columns and rowsr.

```

; function RankAr,a          ; # of dimensions in array a.
; function VarTyp,a         ; Variable Type of a.
; function sq,a             ; Format number, squeeze (remove blanks).
; function round,x          ; Round x to nearest integer.
; function CheckAbort,xpos,ypos ; Look for button presses to abort the
;   ; current program. If pressed, set a
;   ; flag.
; pro Pause,n              ; Pause for n seconds
; pro WrFil,FilNam,a       ; Write a to file FilNam.
; pro RdFil,FilNam,a       ; Read a from file FilNam.
; pro read_sgi_image,filename,image ; Read an SGI format image file.
; pro write_sgi_image,filename,image,pseudo=pseudo,scale=scale
;   ; Write an SGI format image file.
; pro OutImage,format,name,image,pseudo=pseudo ; Output an image--many formats.
; pro KBMenu,iWin,iColor,black,white,iCol1,iCol2,iRow1,iRow2,iFill , $
;   iter,niter,nCol,nRow,iGeom,im1,im2,offset,nCol3,nRow3; Internal routine: Keyboard Menu
; pro loadct2,iColor        ; Modified loadct to include my 99.
; pro AutoScale,a,iWin, black,white ; Autoscale image a, to produce black,white
; pro ReadImage,FilNam,a,nCol,nRow,iCol1,iCol2,iRow1,iRow2, $
;   iType,nheader,nFrame ; Read image--many formats.
; pro ReadImFile,flag,flag2, a,iGeom ; Internal routine: Prompt & read image.
; pro read_pgm,FilNam,a ; Read raw PGM file
; pro write_pgm,FilNam,a ; Write raw PGM file--not sure if works.
; pro ReadFileInt,unit,n ; Read ASCII # from file
; pro PrintImTypes ; Print Image Types.
; pro PrintGeomTypes ; Print Geometry Types
; function SmoothEx,a,n ; Like Smooth, but works near edges.
; 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 window.
; pro Dis,a,b,iWin=iWin,iColor=iColor,range=range,select=select,iFill=iFill, $
;   xpick=xpick,ypick=ypick,iGeom=iGeom,dt=dt,Offset=Offset,xsize=xsize,ysize=ysize, $
;   title=title,xpos=xpos,ypos=ypos,noframe=noframe
;   ; Display Image(s).
; pro dismain,flag ; Prompts for input, displays image(s).
; pro DisMenu,FilNam1,FilNam2,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 ; Menu for input, displays image(s).
; -----
; (main procedure is set up to do image display)

; Use something like
;   dis,image_array ; display image array
;   ; (can use other parameters)
; or
;   .run dis
;   dismenu,'filename' ; display image file
; -----

```

```

pro GetSize,a, nCol,nRow      ; Output # of columns and rows in
    ; matrix a.
;Written by Mitchell R Grunes.
s=size(a)
nCol=s(1)
nRow=s(2)
end
;-----
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
;-----
function VarTyp,a            ;Variable Type of a
    ; 0=undefined
    ; 1=byte
    ; 2=integer*2
    ; 3=integer*4
    ; 4=real*4
    ; 5=real*8
    ; 6=complex
    ; 7=string
;Written by Mitchell R Grunes.
s=size(a)
return,s(n_elements(s)-2)
end
;-----
function sq,a                ; format number, squeeze
    ; (remove blanks).
    ; -----INPUT-----
    ; a=a numeric value
    ; -----OUTPUT-----
    ; returns formatted string, with
    ; no blanks.
;Written by Mitchell R Grunes.
b=a(0)
if VarTyp(b) ne 7 then begin
    if abs(b) lt 2.d0^31 then begin
        if b eq long(b) then b=long(b)
    endif
endif
return,strcompress(string(b),/remove_all)
end
;-----

```

```

function round,x          ; Round x to nearest integer
;Written by Mitchell R Grunes.
return,long(x+ (x ge 0)-.5)
end
;-----
function CheckAbort,xpos,ypos      ; Look for button presses to abort the
; current program. If pressed, set a
; flag.
; -----INPUTS-----
; xpos,ypos=position of window.
; if xpos=-1, flag should be reset,
; and window deleted.
; -----OUTPUTS-----
; Returns 0 if abort has occurred.
;Written by Mitchell R Grunes.
common CmCheckAbort,CheckFlag,CheckWindow ; Only used by this routine.

if n_elements(xpos) eq 0 then xpos=0
if n_elements(ypos) eq 0 then ypos=200
if n_elements(CheckWindow) eq 0 then CheckWindow=-1
if n_elements(CheckFlag) eq 0 then CheckFlag=0      ; Initialize flag
savewindow=!d.window

if xpos eq -1 then begin          ; Special reset call.
  OldCheckFlag=CheckFlag
  CheckFlag=0
  if CheckWindow ge 0 then wdelete,CheckWindow
  if savewindow ge 0 then wset, SaveWindow
  if savewindow ge 0 then wshow,SaveWindow
  return,OldCheckFlag
endif

if CheckFlag then return,1

if CheckWindow eq -1 then begin   ; Create window if does not exist.
  Window,/free,xsize=300,y size=30,colors=256,retain=2,xpos=xpo s,ypos=ypos
  CheckWindow=!d.window
  xyouts,0,5,'Click in this window to abort.',/device
endif

tvcrs,1 & tvcrs,5,5
cursor,x,y,/device,/nowait

wset,CheckWindow & wshow,CheckWindow

if !err ne 0 and x ge 0 and y ge 0 then begin
  print,string(byte(7)),format='(a1,6hClick!)' ; Beep
  erase

```

```

xyouts,0,0,'Abort has occurred.',/device
cursor,xdummy,ydummy,/device,/nowait      ; To flush X-Windows buffer
cursor,xdummy,ydummy,/device,/nowait
wait,2
CheckFlag=1
CheckWindow=-1
endif

return,CheckFlag
end

;-----
pro Pause,n                ; Pause for n seconds
;Written by Mitchell R Grunes.

OldTime=systime(1)
while abs(OldTime-systime(1)) lt n do dummy=0
end
; -----WrFil-----
pro WrFil,FilNam,a          ; Write a to file FilNam
; By Mitchell R Grunes.
print,'-----'
print,'Writing file ',FilNam
openw,unit,FilNam,/get_lun
writeu,unit,a
close,unit
free_lun,unit
print,'Shape given by:'
if !prompt eq 'IDL> ' then help,a
if !prompt ne 'IDL> ' then info,a      ; Lately, Wave replaced
    ; 'help' with 'info'.
print,'-----'
end
; -----RdFil-----
pro RdFil,FilNam,a          ; Read a from file FilNam
; By Mitchell R Grunes.
print,'-----'
print,'Reading file ',FilNam
openr,unit,FilNam,/get_lun
readu,unit,a
close,unit
free_lun,unit
print,'Shape given by:'
if !prompt eq 'IDL> ' then help,a
if !prompt ne 'IDL> ' then info,a      ; Lately, Wave replaced
    ; 'help' with 'info'.
print,'-----'
end
;-----read_sgi_image-----

```

```

pro read_sgi_image,filename,image    ; Read an SGI format image file.
; ----INPUT-----
; filename=Name of file--e.g., 'junk.rgb'
; ----OUTPUT-----
; image=  2 or 3 dimensional image.
;       Monochrome images are 2D.
;       RGB images are 3D--3rd dimension
;       has 3 planes, or 4 to include
;       an "alpha" channel.  If you
;       want to understand alpha channels
;       log onto an sgi and type
;       man multisample
; By Mitchell R Grunes.
; We do NOT use run length encoding (RLE), or anything fancy.
; We also assume this is being run a MSB-first machine, such as a Sun Sparc
; or SGI workstation.
; That means it would fail on a PC-compatible or most HP or DEC machines.
openr,unit,filename,/get_lun
idummy=0
readu,unit,idummy          ; IRIS image file magic number
if idummy ne 474 then stop,'***read_sgi_image encountered bad magic number.'
readu,unit,idummy
if idummy ne 0 then stop,'***read_sgi_image can not handle RLE.'
bpp=byte(0) & nDim=0 & nCol=0 & nRow=0 & nChan=0
readu,unit,bpp             ; Bytes/pixel
if bpp ne 1 and bpp ne 2 and bpp ne 4 then stop,'ERROR: Wrong bytes/pixel in read_sgi_image!'
readu,unit,nDim             ; # of dimensions
if nDim lt 2 or nDim gt 3 then stop,'ERROR: Wrong # of dimensions in read_sgi_image!'
readu,unit,nCol             ; # of columns
readu,unit,nRow             ; # of rows
readu,unit,nChan            ; # of channels
if nDim eq 2 then begin
  if bpp eq 1 then begin
    image=bytarr(nCol,nRow)
  endif else if bpp eq 2 then begin
    image=intarr(nCol,nRow)
  endif else if bpp eq 4 then begin
    image=lonarr(nCol,nRow)
  endif
endif else if nDim eq 3 then begin
  if bpp eq 1 then begin
    image=bytarr(nCol,nRow,nChan)
  endif else if bpp eq 2 then begin
    image=intarr(nCol,nRow,nChan)
  endif else if bpp eq 4 then begin
    image=lonarr(nCol,nRow,nChan)
  endif
endif
endif

```

```

iDummy=0L
readu,unit,iDummy          ; Lo scaling value--ignored
readu,unit,iDummy          ; Hi scaling value--ignored
readu,unit,iDummy          ; Dummy value
dummy=bytarr(80)
readu,unit,dummy           ; Image name--not used
readu,unit,idummy          ; Colormap ID--not used
dummy=bytarr(404)
readu,unit,dummy           ; Dummy values.
readu,unit,bytarr(404)     ; Dummy values
readu,unit,image           ; The image
close,unit
free_lun,unit
end
;-----write_sgi_image-----
pro write_sgi_image,filename,image,pseudo=pseudo,scale=scale
; Write an SGI format image file.
; -----INPUT-----
; filename=Name of file--e.g., 'junk.rgb'
;     It is suggested that
;     monochrome file names end in .bw
;     rgb file names end in .rgb
;     rgb+alpha file names end in .rgba
; image= 2 or 3 dimensional image.
;     Monochrome images are 2D.
;     RGB images are 3D--3rd dimension
;     has 3 planes, or 4 to include
;     an "alpha" channel. If you
;     want to understand alpha channels
;     log onto an sgi and type
;     man multisample
;     Must be of type byte or int.
;     However, image is handled as
;     unsigned numbers.
; /pseudo: Indicates that monochrome images (i.e.,
;     2 dimensional) should be mapped to an
;     RGB mage using the pseudo-colors of the
;     current window.
; /scale: Indicates that the minimum and maximum
;     should be used by be used by SGI utilities
;     to scale the image.
;     If not present, indicates that
;     0 and 255 should be used.
; -----OUTPUT-----
; image will be modified if /pseudo is selected
; By Mitchell R Grunes.
; We do NOT use run length encoding (RLE).
; We also assume this is being run a MSB-first machine, such as a Sun Sparc

```

```

; or SGI workstation.
; That means it would fail on a PC-compatible or most HP or DEC machines.
if n_elements(scale) eq 0 then scale=0
if n_elements(pseudo) eq 0 then pseudo=0
openw,unit,filename,/get_lun
writeu,unit,474 ; IRIS image file magic number
writeu,unit,byte(0) ; No RLE
s=size(image)
nDim=s(0) ; # of dimensions.
; Monochrome images are two
; dimensional.
; RGB images have a third dimension.
if pseudo ne 0 and nDim eq 2 then begin
  tvlct,/get,r,g,b
  if total(r ne g)+total(r ne b) ne 0 then $
    image=[[[r(image)]],[[g(image)]],[[b(image)]]]
  s=size(image)
  nDim=s(0) ; # of dimensions.
endif
if nDim lt 2 or nDim gt 3 then stop,'ERROR: Wrong # of dimensions in write_sgi_image!'
nCol=s(1) ; # of columns of pixels
nRow=s(2) ; # of rows of pixels
if nDim eq 2 then nChan=1 else nChan=s(3) ; # of channels (i.e., colors)
if nDim eq 2 then VarTyp=s(3) else VarTyp=s(4) ;1=byte,2=integer*2,3=integer*4
if VarTyp eq 1 then writeu,unit,byte(1) ; Bytes/pixel
if VarTyp eq 2 then writeu,unit,byte(2)
if VarTyp lt 1 or VarTyp gt 2 then stop,'ERROR: Wrong variable type in write_sgi_image'
writeu,unit,fix(nDim)
writeu,unit,fix(nCol)
writeu,unit,fix(nRow)
writeu,unit,fix(nChan)
if Scale eq 0 then begin
  writeu,unit,long(0) ; Minimum pixel value--0 is allowed
  writeu,unit,long(255) ; Maximum pixel value--255 is allowed
endif else begin
  writeu,unit,long(min(image))
  writeu,unit,long(max(image))
endelse
writeu,unit,long(0) ; Dummy value
writeu,unit,bytarr(80) ; Image name--not used
writeu,unit,long(0) ; Colormap ID
writeu,unit,bytarr(404) ; Dummy values
writeu,unit,image ; The image
close,unit
free_lun,unit
end
;-----
pro OutImage,format,name,image,pseudo=pseudo ; Output an image--many formats

```


;Written by Mitchell R Grunes.

```
; -----INPUT-----
; format= See PrintImTypes
;   If undefined, will prompt for format.
; name =Name of output file.
;   If is blank or undefined, will
;   be replaced by sequential name.
;   However, 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 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=1
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 pseudo-color to
RGB
    hi=n_elements(r)-1
    lo=min(image2) < hi
    hi=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
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
  close,unit
  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,image2
    endif else begin
tiff_write,nn,image2,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 $
    stop,'ERROR: Can not output a color Tiff with PV-WAVE Tiff_Write
  tiff_write,nn,image2
endif else
endelse

```

```

endif else if format eq 29 then begin
  write_pgm,nn,image2
endif else if format eq 34 then begin
  write_sgi_image,nn,image2
endif else if format eq 33 then begin
  write_sgi_image,nn,image2,/pseudo
endif else if format eq 34 or 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 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.'
  openw,unit,nn,/append,/get_lun
  writeu,unit,byte(4)
  close,unit
  free_lun,unit
endif

end
;-----
pro KBMenu,iWin,iColor,black,white,iCol1,iCol2,iRow1,iRow2,iFill , $
  iter,niter,nCol,nRow,iGeom,im1,im2,offset,nCol3,nRow3
  ; Keyboard Menu
;Written by Mitchell R Grunes.
menu1:
  tvlct,/get,red,green,blue
  tvcrs,0,0

  if iWin ge 0 then wshow,iWin,0

  print,'---Keyboard Menu---'
  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=fast,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,'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
  print,'Normal LOADCT color schemes 0-15 are available.
  print,'Color Scheme 50 is the user scheme defined by USERCOL.PRO.
  print,'Note that 99 is a special color scheme where
  print,' 253,254,255 are red,green,blue; rest is monochrome.
  print,'98 is a special color scheme for map images where
  print,' 0-127=grey shades, 128-254=yellow shades, 255=red.
  print,'97 is a color scheme for detect.pro output where 0-249 are a grey scale,
  print,' 250-255 are red,yellow,green,light blue,blue,purple
  print,'96 is a color scheme for detect.pro output where 0-121 are a grey scale,
  print,' 122-127 are red,yellow,green,light blue,blue,purple
  print,'99,98,97, and 96 are only usable for byte images, in 256 color windows.'
  read,iColor
  if (iColor lt 0 or iColor ge 16) and iColor ne 50 and iColor ne 96 and iColor ne 97 $
    and iColor ne 98 and iColor ne 99 then iColor=0
  if iWin ge 0 then wshow,iWin
  loadct2,iColor
  if iWin ge 0 then wshow,iWin,0
  tvcrs,10,5
endif else if ichoice eq 2 then begin
  print,'(use 0,0 to autoscale)
  black=double(black) & white=double(white)
  read,black,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 ichoice eq 5 then begin
  read,iFill
endif else if ichoice eq 6 then begin
  PrintGeomTypes
  read,iGeom
endif else if ichoice 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 begin
    aSave=tvrd(2,20,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
OutImage,format,nam,aSave,pseudo=pseudo
endif else if ichoice 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 ichoice eq 9 then begin
print,'End this mode with exit
spawn
endif else if ichoice 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)
endif else
if lo lt hi then test=lo > test < hi

```

```

    print,'Average value for image ',sq(iter),'=',total(test)/n_elements(test)
endfor
endif
if ichoice ne 0 then goto,menu1
if iWin ge 0 then wshow,iWin
if iWin ge 0 then wset,iWin
on_ioerror,null
end
;-----
pro loadct2,iColor          ; Modified loadct to include my 99.
;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.
; 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 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
tv!ct,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 then begin
    openr,unit,'usercol.dat',/get_lun
    readf,unit,ncolorr,ncolorg,ncolorb    ; # of mark points

    xrf=intarr(ncolorr)                  ;independent coord of transfer function: R,G,B

```



```

xgf=intarr(ncolorg)
xbf=intarr(ncolorb)

rf=xrf & gf=xgf & bf=xbf      ;dependent coord for R,G,B
readf,unit,xrf,xgf,xbf,rf,gf,bf
close,unit
free_lun,unit

r=fix(interpol(rf,xrf,indgen(256))+.5)
g=fix(interpol(gf,xgf,indgen(256))+.5)
b=fix(interpol(bf,xbf,indgen(256))+.5)

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 begin
  loadct,iColor
endelse
end
;-----
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
if iWin ge 0 then wshow,iWin
end
;-----
pro ReadImage,FilNam,a,nCol,nRow,iCol1,iCol2,iRow1,iRow2, $
iType,nheader,nFrame          ; Read image--many formats.
; image=name of image file.
; Array to read image to.
; nCol=# of columns in image.
; Used only for raw formats.
; nRow=# of rows in image.
; Used only for raw formats.
; iCol1=0-origin Column # to start load at.
; iCol2=0-origin Column # to end load at.
; iRow1=0-origin Row # to start load at.
; iCol2=0-origin Row # to end load at.
; iType=Type of image file--see PrintlmType.
; nHeader=# of header bytes or lines to skip.
; Used only in raw formats.
; nFrame=# of movie frames stored in file.
; Used only in raw formats.
;Written by Mitchell R Grunes.
Start:
if iType lt 20 and iType ge 0 then begin
iCol1=0 > iCol1
if iCol2 le iCol1 or iCol2 gt nCol-1 then iCol2=nCol-1
Row1=0 > iRow1
if iRow2 le iRow1 or iRow2 gt nRow-1 then iRow2=nRow-1
print,'Reading image ',FilNam,' cols',iCol1,iCol2,' rows',iRow1,iRow2
endif else begin
print,'Reading image ',FilNam
endelse
if n_elements(nFrame) eq 0 then nFrame=1
if iType eq -1 then begin
a=byte(255)-byte(indgen(16,16))          ;multi-row Gray scale

```

```

for i=1,15,2 do a(*,i)=a(15-indgen(16),i)
a=rebin(a,32,32)
for i=1,31,4 do a(0:29,i)=0
for i=3,31,4 do a(2:31,i)=0
a=a(*,0:30)
nCol=256
nRow=256
endif else if iType lt 20 then begin
  if iCol1 lt 0 then iCol1=0
  if iCol2 le iCol1 or iCol2 gt nCol-1 then iCol2=nCol-1
  if iRow1 lt 0 then iRow1=0
  if iRow2 le iRow1 or iRow2 gt nRow-1 then iRow2=nRow-1
  openr,unit,FilNam,/get_lun
  ;print,'0 Normal--rows are contiguous left > right, rows are top down'
  ;print,'1 Rotate CW 90 deg: may not work'
  ;print,'2 Rotate 180 deg'
  ;print,'3 Rotate CCW 90 deg: may not work'
  ;print,'4 Reflect across main diag (i.e., transpose): may not work'
  ;print,'5 Reflect across vertical (reverse cols)'
  ;print,'6 Reflect across other diag: may not work'
  ;print,'7 Reflect across horizontal (reverse rows)'
  nBytePix=1
  if iType eq 2 or iType eq 3 or iType eq 9 or iType eq 10 then nBytePix=2
  if iType eq 4 or iType eq 5 or iType eq 11 or iType eq 12 then nBytePix=4
  if iType eq 6 or iType eq 7 or iType eq 13 or iType eq 14 then nBytePix=8
  if iType eq 8 or iType eq 15 then nBytePix=16
  if iType eq 27 or iType eq 28 then nBytePix=2
  ;iskip=[iRow1,      nCol-1-iCol2, nRow-1-iRow2, iCol1, $
  ;      iCol1,      iRow1,      nCol-1-iCol2, nRow-1-iRow2]
  ;width=[nCol,      nRow,      nCol,      nRow, $
  ;      nRow,      nCol,      nRow,      nCol]
  ;keep =[iRow2-iRow1, iCol2-iCol1, iRow2-iRow1, iCol2-iCol1, $
  ;      iCol2-iCol1, iRow2-iRow1, iCol2-iCol1, iRow2-iRow1]+1
  ;iskip=iskip(iGeom)
  ;width=width(iGeom)
  ;keep =keep (iGeom)
  iskip=iRow1
  width=nCol
  keep=iRow2-iRow1
  if iType ge 16 and iType le 19 then begin
    a=' '
    if nheader gt 0 then for i=1,nheader do readf,unit,a
    if iType eq 16 then begin
if nFrame eq 1 then a=intarr(width,keep)
if nFrame gt 1 then a=intarr(width,keep,nFrame)
endif else if iType eq 17 then begin
if nFrame eq 1 then a=lonarr(width,keep)
if nFrame gt 1 then a=lonarr(width,keep,nFrame)

```

```

endif else if iType eq 18 then begin
if nFrame eq 1 then a=fltarr(width,keep)
if nFrame gt 1 then a=fltarr(width,keep,nFrame)
endif else if iType eq 19 then begin
if nFrame eq 1 then a=dblarr(width,keep)
if nFrame gt 1 then a=dblarr(width,keep,nFrame)
endif
endif else begin
print,'skipping ', nheader+iskip*long(width)*nBytePix
point_lun, unit, nheader+iskip*long(width)*nBytePix
endelse
if iType eq 0 or iType eq 1 then begin
if nFrame eq 1 then a=bytarr(width,keep)
if nFrame gt 1 then a=bytarr(width,keep,nFrame)
jtype=1 ; Will be stored in byte &&&
endif else if iType eq 2 or iType eq 3 or iType eq 9 or iType eq 10 or iType eq 16 then begin
if nFrame eq 1 then a=intarr(width,keep)
if nFrame gt 1 then a=intarr(width,keep,nFrame)
jtype=2 ; Will be stored in int
if iType eq 3 or iType eq 10 then jtype=3 ; unsigned will be converted to long
endif else if iType eq 4 or iType eq 11 or iType eq 17 then begin
if nFrame eq 1 then a=lonarr(width,keep)
if nFrame gt 1 then a=lonarr(width,keep,nFrame)
jtype=3 ; Will be stored in long
endif else if iType eq 5 or iType eq 12 or iType eq 18 then begin
if nFrame eq 1 then a=fltarr(width,keep)
if nFrame gt 1 then a=fltarr(width,keep,nFrame)
jtype=4 ; Will be stored in float
endif else if iType eq 6 or iType eq 13 or iType eq 19 then begin
if nFrame eq 1 then a=dblarr(width,keep)
if nFrame gt 1 then a=dblarr(width,keep,nFrame)
jtype=5 ; Will be stored in double
endif else if iType eq 7 or iType eq 14 then begin
if nFrame eq 1 then a=complexarr(width,keep)
if nFrame gt 1 then a=complexarr(width,keep,nFrame)
jtype=6 ; Will be stored in complex
endif
; &&& Type 8 and 15--not sure how to do
if iType ge 16 and iType le 19 then readf,unit,a
if iType ge 0 and iType lt 16 then readu,unit,a
print,"File read complete"
;if iGeom eq 0 or iGeom eq 2 or iGeom eq 5 or iGeom eq 7 then $
if nFrame eq 1 then a=a(iCol1:iCol2,0:keep-1)
if nFrame gt 1 then a=a(iCol1:iCol2,0:keep-1,*)
;if iGeom eq 1 or iGeom eq 3 or iGeom eq 4 or iGeom eq 6 then $
; a=a(iRow1:iRow2,0:keep-1)
if iType eq 9 or iType eq 10 then byteorder,a,/sswap
if iType eq 11 or iType eq 12 then byteorder,a,/lswap

```

```

; &&&Type 13 and 15--not sure how to do
if iType eq 2 or iType eq 3 or iType eq 10 then begin ;unsigned--clip hi bits
    if min(a) lt 0 then begin
if iType eq 1 then a=a and 255
if iType eq 3 or iType eq 10 then begin
    if !prompt eq 'IDL> ' then begin
        a=temporary(a) and 65535
    endif else begin
        a=a and 65535
    endelse
endif
    endif
endif

; if jtype eq 4 then begin
; ; if min(a eq long(a)) eq 1 then begin
; ; a=long(a)
; ; jtype=3
; ; endif
; endif
if jtype eq 2 or jtype eq 3 then maxa=max(a,min=mina)
if jtype eq 3 then begin
    if mina ge -32768 and maxa le 32767 then begin
a=fix(a)
jtype=2
    endif
endif
if jtype eq 2 then begin
    if mina ge 0 and maxa le 255 then a=byte(a)
endif
endif else begin
    if iType eq 20 then begin
        read_gif,FilNam,a,r,g,b
    endif else if iType eq 21 then begin
        READ_PICT,FilNam,a,r,g,b
    endif else if iType eq 22 then begin
        READ_SRF,FilNam,a,R,G,B
    endif else if iType eq 23 then begin
        ;Not yet understood function description--probably won't work
        read_wave,FilNam,a,names,dimensions
    endif else if iType eq 24 then begin
        READ_X11_BITMAP,FilNam,a
    endif else if iType eq 25 then begin
        a=READ_XWD(FilNam,r,g,b)
    endif else if iType eq 26 then begin
        if !prompt eq 'IDL> ' then a=TIFF_READ(FilNam,R,G,B)
        if !prompt ne 'IDL> ' then a=TIFF_READ(FilNam)
    endif else if iType eq 27 or iType eq 28 then begin ;SOHO Compressed

```

```

print,'The free trial copy of IDL cannot do this, and will give ', $
'warnings.'
if FilNam ne 'test.cim' then spawn,'cp '+FilNam+' test.cim'
spawn,'rm test.rim test.siz'
if iType eq 27 then spawn,'sohorecon test.cim test.rim'
if iType eq 28 then spawn,'sohorecon test.cim test.rim skip'
openr,unit,'test.siz',/get_lun
readf,unit,nCol,nRow,iType
FilNam='test.rim'
iType=2
goto,Start
endif else if iType eq 29 then begin      ;PGM
  read_pgm,FilNam,a
endif else if iType eq 30 then begin;Tom Ainsworth's read_array format
  read_array,FilNam,a
endif else if iType eq 31 or iType eq 32 then begin
  rd_image,FilNam,a,imginfo,nimgs  ; Bob Jansen's routines to read
  ; Sealab and HDF files.
endif else if iType eq 33 then begin; IRIS RGB Image
  write_sgi_image,filenam,image,/pseudo
endif else if iType eq 34 then begin; IRIS BW Image
  write_sgi_image,filenam,image
endif else if iType eq 36 then begin; DRM CEOS format
  print,'Extracting data from CEOS header
  close,1
  openr,1,SarFile
  temp='      '      ; 8 spaces
  point_lun,1,19498      ; location for # of rows
  readu,1,temp
  nCol=long(temp)+192/2      ; (include initial non-image 192 bytes/record)
  point_lun,1,19486      ; location for # of columns
  readu,1,temp
  nRow=long(temp)
  jcol1=96
  jcol2=nCol-1
  jrow1=1
  jrow2=nRow-1
  nRow3=nRow<500      ; Test portion
  a=intarr(nCol,nRow3)      ; It appears that image has zeroed cols and rows
  ByteSkip=19250
  point_lun,1,ByteSkip      ; at border, not documented in header.
  readu,1,a
  while jcol1 lt jcol2-2 and total(a(jcol1:jcol1,30:nRow3-30) eq 0) gt (nRow3-60)/2 do
jcol1=jcol1+1
  while jcol2 gt jcol1+2 and total(a(jcol2:jcol2,30:nRow3-30) eq 0) gt (nRow3-60)/2 do
jcol2=jcol2-1
  while jrow1 lt nRow3-2 and total(a(jcol1+30:jcol2-30,jrow1) eq 0) gt (nCol-60)/2 do
jrow1=jrow1+1

```

```

point_lun,1,ByteSkip+(nRow-1-nRow3)*nCol*abs(PixelType)
readu,1,a
while jrow2 gt nrow-(nRow3-2) and total(a(jcol1+30:jcol2-30,nRow3-(nrow-jrow2)) eq 0) $
  gt (nCol-60)/2 do jrow2=jrow2-1
  nCol2=jCol2-jCol1+1
  nRow2=jRow2-jRow1+1
  a=intarr(nCol,nRow2)
  point_lun,1,ByteSkip+jRow1*long(nCol)*2
  readu,1,a
  a=a(jCol1:jCol2,*)
  close,1
endif
s=size(a)
nCol=s(1)
nRow=s(2)
if iCol1 lt 0 then iCol1=0
if iCol2 le iCol1 or iCol2 gt nCol-1 then iCol2=nCol-1
if iRow1 lt 0 then iRow1=0
if iRow2 le iRow1 or iRow2 gt nRow-1 then iRow2=nRow-1
print,'Selecting cols',iCol1,iCol2,' rows',iRow1,iRow2
GetSize,a, nCol3,nRow3
if iCol1 ne 0 or iRow1 ne 0 or iCol2 ne nCol3-1 or iRow2 ne nRow3-1 then $
  a=a(iCol1:iCol2,iRow1:iRow2)
endelse
;if iGeom ne 0 then begin          ; Done elsewhere
; if !prompt eq 'IDL> ' then begin
;   a=rotate(temporary(a),iGeom)
; endif else begin
;   a=rotate(a,iGeom)
; endelse
;endif
end

```

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

(continued in next post)

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