Subject: Re: Nice Postscript output from shade-surf? Xsize in polyshade?
Posted by l126487 on Tue, 24 Jan 1995 15:44:32 GMT
View Forum Message <> Reply to Message

C H Solterbeck (solter@theo-physik.uni-kiel.de) wrote:

: Hi,

: I would like to get a good output in postscript out of the shade_surf-routine
: of PV-Wave. But if the device has scalable pixels, the output image has
: dimensions less or equal to 512. This is not good enough.
: How can I get a better resolution?

Here's a procedure which may do what you're looking for. This came from
a Tech support engineer at Visual Numerics.

Brian Seifert
l126487@cliffy.lfwc.lockheed.com

```
;============================== ss.pro ==============================
;
; This is an example routine that demonstrates how to get greater resolution
; from the shade_surf command.  It accomplishes this by rendering a larger
; image and then resizing it to the original viewwindow size.  It then draws
; the axis on the graphic.  The AX and AZ keywords work like the keywords to
; the shade_surf command in specifying X and Z rotations.  The RESOLUTION
; keyword provides the ability to specify the quality of the resulting graphic.
; If RESOLUTION is set to the value of 1, no enhancement is made.  If the
; RESOLUTION keyword is set to something like 4 or 5, the resulting graphic is
; much sharper.  An example calling sequence would be:
;
;   ss, z,x,y, ax=60, az=15, resolution=5
;
; The resulting graphic would be drawn to the display or the PostScript device
; (whichever is currently active).  If the current device is PostScript, be
; sure to issure "DEVICE,/CLOSE" to close the output file before you try to
; print it.  It is recommended that the remainder of the keywords for the
; SHADE_SURF command be added to this routine so that it's behavior will be
; identical

pro ss, z,x,y, ax=ax, az=az, resolution=resolution

; Check the parameters to make sure that any that were not included in the
; calling sequence get assigned a default value.

  if n_elements(ax) eq 0 then ax=30
  if n_elements(az) eq 0 then az=30
```

```
   if n_elements(x) eq 0 then x=findgen((size(z))(1))
   if n_elements(y) eq 0 then y=findgen((size(z))(2))
   if n_elements(resolution) eq 0 then resolution=1

; Store the sizes of the currently active window.

   if !d.name eq 'X' then begin
      winid=!d.window & winx=!d.x_vsize & winy=!d.y_vsize
   endif else begin
      winid=-1 & winx=!d.x_vsize/25 & winy=!d.y_vsize/25
   endelse

; Determine the size of the "High Resolution" window.  If the RESOLUTION keyword
; is set to high, you may get a complaint that there is not enough memory to
; allocate the array.

   dev_name=!d.name
   xres=winx*resolution
   yres=winy*resolution

; If the current device is PostScript, change the current device to X so that
; the high res image can be drawn

   if !d.name eq 'PS' then set_plot, 'X'

   window,/free,/pixmap,xsize=xres,ysize=yres  ; Open the High Res window.

; Draw the image into the High Res Window.  The color and position of the
; graphic is needs specific setting for the PostScript device.

   if dev_name eq 'X' then begin
      shade_surf,z,x,y,ax=ax,az=az,xmargin=!x.margin*resolution,xs t=5,yst=5,$
            zst=5,ymargin=!y.margin*resolution,zmargin=!z.margin*resolut ion
   endif else begin
      shade_surf,z,x,y,ax=ax,az=az,xst=5,yst=5,xmargin=!x.margin*r esolution,$
            zst=5,ymargin=!y.margin*resolution,zmargin=!z.margin*resolut ion, $
           background=!d.n_colors-1, position=[.0,.0,1,1]
   endelse


; Read the image from the High Res window then delete the window

   image=rebin(tvrd(0,0,xres,yres),winx,winy)

   wdelete

; If the original output device was PostScript, return to that device
```

```
   if dev_name eq 'PS' then set_plot,dev_name

; Display the High Resolution image and overlay the meshed surface

   tvscl,image

   if dev_name eq 'PS' then surface,z,x,y,ax=ax,az=az,/noerase, $
                            position=[.0,.0,1,1], xst=1,yst=1,zst=1 $
   else surface,z,x,y,ax=ax,az=az,/noerase,xst=1,yst=1,zst=1

end ;ss
```