Subject: Re: sorting out polygon connectivity in a simple 2D rectangular dataset...
Posted by David Fanning on Fri, 16 Nov 2012 00:25:08 GMT
View Forum Message <> Reply to Message

George.millward@yahoo.com writes:

> it would seem to me that the default would be for this to 'just work' ?

I think you must be thinking about the "old" IDL. ;-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: sorting out polygon connectivity in a simple 2D rectangular dataset...
Posted by Dick Jackson on Fri, 16 Nov 2012 04:27:55 GMT
View Forum Message <> Reply to Message

Hi George,

Yep, it's as easy as you thought it should be... you just used IDLgrPolygon (which allows/requires that you specify the connectivity for the mesh) instead of IDLgrSurface (which creates the grid for you. This looks like what you want. If you really wanted to use IDLgrPolygon, the Mesh_Obj procedure (type=2) would help you to make the required vertex and connectivity lists.

```
PRO VertColorSurfaceTest

;;    Simple test of Object Graphics surface with Vert_Colors
;;
;;    Dick Jackson Software Consulting  --  www.d-jackson.com

x = LIndGen(180,60) MOD 180 + RandomU(seed, 180,60)*0.5
y = LIndGen(180,60) / 180 + RandomU(seed, 180,60)*0.5
z = Dist(180,60) + RandomU(seed, 180,60)
density = BytScl(x+y)     ; to be the vert_colors

oPalette = Obj_New('IDLgrPalette') ; Make a palette to show density
oPalette -> LoadCT, 2

oSurface = Obj_New('IDLgrSurface', z, x, y, Style=2, $
```

```
            Vert_Colors=density[*], $ ; (must be a vector)
            Palette=oPalette)

XObjView, oSurface   ; Display in a viewer window

END
```

Hope this helps!

Cheers,
-Dick

Dick Jackson Software Consulting
Victoria, BC, Canada --- www.d-jackson.com

On Thursday, November 15, 2012 3:48:02 PM UTC-8, George....@yahoo.com wrote:
> Hi There,
>
> I'm getting stumped by what I think should be really easy.
>
> I have a 180 by 60 rectangular dataset and I want this to be the vert_colors
> on an idlgrpolygon. But how do I calculate the connectivity - shouldn't it be really easy for a simple rectangular dataset ?
>
> ie, I have 3D cartesian coords: x(180,60), y(180,60), z(180,60) and some data
> Density(180,60) that I want to be the vert_colors.
>
> It works if I use 'style=0' but thats because this just plots 'dots' so the connectivity is not important. But for the standard surface 'style=2' it's not looking right. Should one need to define the connectivity for a simple 2D rectangular surface - it would seem to me that the default would be for this to 'just work' ?
>
> Cheers
>
> George.

Subject: Re: sorting out polygon connectivity in a simple 2D rectangular dataset...
Posted by George.millward on Fri, 16 Nov 2012 18:56:58 GMT
View Forum Message <> Reply to Message

On Thursday, November 15, 2012 9:27:55 PM UTC-7, Dick Jackson wrote:
> Hi George,
>
>
>
> Yep, it's as easy as you thought it should be... you just used IDLgrPolygon (which allows/requires that you specify the connectivity for the mesh) instead of IDLgrSurface (which

creates the grid for you. This looks like what you want. If you really wanted to use IDLgrPolygon, the Mesh_Obj procedure (type=2) would help you to make the required vertex and connectivity lists.

```
>
>
>
>  PRO VertColorSurfaceTest
>
>
>
>  ;;    Simple test of Object Graphics surface with Vert_Colors
>
>  ;;
>
>  ;;    Dick Jackson Software Consulting  --  www.d-jackson.com
>
>
>
>  x = LIndGen(180,60) MOD 180 + RandomU(seed, 180,60)*0.5
>
>  y = LIndGen(180,60) / 180 + RandomU(seed, 180,60)*0.5
>
>  z = Dist(180,60) + RandomU(seed, 180,60)
>
>  density = BytScl(x+y)     ; to be the vert_colors
>
>
>
>  oPalette = Obj_New('IDLgrPalette') ; Make a palette to show density
>
>  oPalette -> LoadCT, 2
>
>
>
>  oSurface = Obj_New('IDLgrSurface', z, x, y, Style=2, $
>
>                  Vert_Colors=density[*], $ ; (must be a vector)
>
>                  Palette=oPalette)
>
>
>
>  XObjView, oSurface   ; Display in a viewer window
>
>
>
>  END
>
```

>
>
> Hope this helps!
>
>
>
> Cheers,
>
> -Dick
>
>
>
> Dick Jackson Software Consulting
>
> Victoria, BC, Canada --- www.d-jackson.com
>
>
>
> On Thursday, November 15, 2012 3:48:02 PM UTC-8, George....@yahoo.com wrote:
>
>> Hi There,
>
>>
>
>> I'm getting stumped by what I think should be really easy.
>
>>
>
>> I have a 180 by 60 rectangular dataset and I want this to be the vert_colors
>
>> on an idlgrpolygon. But how do I calculate the connectivity - shouldn't it be really easy for a simple rectangular dataset ?
>
>>
>
>> ie, I have 3D cartesian coords: x(180,60), y(180,60), z(180,60) and some data
>
>> Density(180,60) that I want to be the vert_colors.
>
>>
>
>> It works if I use 'style=0' but thats because this just plots 'dots' so the connectivity is not important. But for the standard surface 'style=2' it's not looking right. Should one need to define the connectivity for a simple 2D rectangular surface - it would seem to me that the default would be for this to 'just work' ?
>
>>
>

>> Cheers
>
>>
>
>> George.

Dick,

Perfect - I forgot about IDLgrSurface.

Thanks for your help.

George.
