Subject: Isurface, inverting the axis without changing default lighting direction
Posted by cameron bowles on Tue, 04 May 2010 01:16:46 GMT
View Forum Message <> Reply to Message

Hi everyone, long time reader first time poster.

I have run into this problem where I want to plot a surface using the iSurface functionality of IDL 7.0. I want the Y-axis to be inverted (ie. showing the maximum value at the common axes point. I can simply do this with Yrange = [max[Y], min[Y]], but when I do this the lighting vector for the surface flips around the Y=0 plane and the surface is highlighted from some strange angle that doesnt highlight the surface at all.

I tried two other solutions;

The first was trying the Ytickvalues = Ynew, where Ynew was a new vector created to go from Ymax to Ymin with equal length to the Ydata. This keyword had no effect the plotted Yaxis values are still the values of the Yaxis values defined by the iSurface command.

The second was to try to overplot it ontop of another iSurface axis set, but the same issue occurs as the lighting direction is still defined by the axis direction.

Here is some sample code;

```
ISURFACE, Data, Xvalues, Yvalues, $
  Xrange = [(*ptrinfo).Xmin, (*ptrinfo).Xmax], $
  Yrange = [(*ptrinfo).Xmin, (*ptrinfo).Xmax], $
  Zrange = [(*ptrinfo).Zmin, (*ptrinfo).Zmax], $
  clip_planes = [[-1,0,0,(*ptrinfo).Xmin],[0,1,0,Ymin-0.01],
[0,0,-1,Zmin], [1,0,0,-1*(*ptrinfo).Xmax],[0,1,0,Ymax-0.01],
[0,0,1,-1*Zmax]], $
  style = 2, $
  shading = 1, $
  skirt = 0.0, $
  /show_skirt, $
  zero_opacity_skip = 0, $
  color = [225, 184, 0], $
  transparency = 20
```

where Data is a 2D data array, and ptrinfo is just a pointer to my structure array I use to pass variables between different procedures.

I hope someone can help with a solution.

Regards,
Cameron

Subject: Re: Isurface, inverting the axis without changing default lighting direction
Posted by penteado on Wed, 05 May 2010 01:49:58 GMT
View Forum Message <> Reply to Message

On May 4, 10:06 pm, cameron bowles <cameronbowle...@gmail.com> wrote:
> Thankyou so much for your tips Ken, they really helped. In the end I
> got it to look kind of OK with this code;
>
> void = ITGETCURRENT(TOOL=itool_obj)
> lights_id = itool_obj -> FindIdentifiers('*LIGHTS', /
> VISUALIZATIONS) ;Get lights ID
> lights_obj = itool_obj ->
> GetByIdentifier(lights_id) ;Get lights object
> reference
> lights_obj -> SetProperty, INTENSITY = 0.7, DISTANCE =
> -20 ;Set intensity and distance (z axis)
> itool_obj -> RefreshCurrentWindow
>
> However I would really like to be able to set the location and
> direction of the directional light, this should be possible with;
>
> lights_obj -> SetProperty, DIRECTION = [x,y,z], DISTANCE = [x,y,z]
>
> But I found that no matter what setting I had for x/y/z it wouldnt
> affect the lighting. Has anyone done this manually? I roughly want to
> have the light coming with a vector of [1,1,-1] from a position
> direction of [-1,-1,1]. If anyone knows the tricks to get that working
> I would appreciate it.

I think your problem is the object you are picking to edit. By
default, isurface makes two lights, one ambient (isotropic, I guess),
and one directional. For instance,

IDL> ids=itool_obj->findidentifiers('*LIGHT*',/visualization)
IDL> for i=0,n_elements(ids)-1 do print,ids[i]
/TOOLS/SURFACE TOOL/WINDOW/VIEW_1/VISUALIZATION LAYER/LIGHTS
/TOOLS/SURFACE TOOL/WINDOW/VIEW_1/VISUALIZATION LAYER/LIGHTS/LIGHT
/TOOLS/SURFACE TOOL/WINDOW/VIEW_1/VISUALIZATION LAYER/LIGHTS/LIGHT_1

You are selecting the lights object, but the properties you want to
edit are for the directional object, which in this case is LIGHT_1:

IDL> light=itool_obj->getbyidentifier('/TOOLS/SURFACE TOOL/WINDOW/
VIEW_1/VISUALIZATION LAYER/LIGHTS/LIGHT')
IDL> light->getproperty,name=name & print,name
Ambient Light
IDL> light=itool_obj->getbyidentifier('/TOOLS/SURFACE TOOL/WINDOW/
VIEW_1/VISUALIZATION LAYER/LIGHTS/LIGHT_1')
IDL> light->getproperty,name=name & print,name

Directional Light

With the object for the directional light, I can move and point it
around with

light->setproperty,location=[1,1,-1],direction=[-1,-1,1]

Note that it may take a

itool_obj->commitactions

for the image to be updated after you change the objects.

Subject: Re: Isurface, inverting the axis without changing default lighting direction
Posted by cameron bowles on Wed, 05 May 2010 07:27:30 GMT
View Forum Message <> Reply to Message

On May 5, 10:49 am, pp <pp.pente...@gmail.com> wrote:
> On May 4, 10:06 pm, cameron bowles <cameronbowle...@gmail.com> wrote:
>
>
>
>> Thankyou so much for your tips Ken, they really helped. In the end I
>> got it to look kind of OK with this code;
>
>> void = ITGETCURRENT(TOOL=itool_obj)
>> lights_id = itool_obj -> FindIdentifiers('*LIGHTS', /
>> VISUALIZATIONS) ;Get lights ID
>> lights_obj = itool_obj ->
>> GetByIdentifier(lights_id) ;Get lights object
>> reference
>> lights_obj -> SetProperty, INTENSITY = 0.7, DISTANCE =
>> -20 ;Set intensity and distance (z axis)
>> itool_obj -> RefreshCurrentWindow
>
>> However I would really like to be able to set the location and
>> direction of the directional light, this should be possible with;
>
>> lights_obj -> SetProperty, DIRECTION = [x,y,z], DISTANCE = [x,y,z]
>
>> But I found that no matter what setting I had for x/y/z it wouldnt
>> affect the lighting. Has anyone done this manually? I roughly want to
>> have the light coming with a vector of [1,1,-1] from a position
>> direction of [-1,-1,1]. If anyone knows the tricks to get that working
>> I would appreciate it.
>
> I think your problem is the object you are picking to edit. By

> default, isurface makes two lights, one ambient (isotropic, I guess),
> and one directional. For instance,
>
> IDL> ids=itool_obj->findidentifiers('*LIGHT*',/visualization)
> IDL> for i=0,n_elements(ids)-1 do print,ids[i]
> /TOOLS/SURFACE TOOL/WINDOW/VIEW_1/VISUALIZATION LAYER/LIGHTS
> /TOOLS/SURFACE TOOL/WINDOW/VIEW_1/VISUALIZATION LAYER/LIGHTS/LIGHT
> /TOOLS/SURFACE TOOL/WINDOW/VIEW_1/VISUALIZATION LAYER/LIGHTS/LIGHT_1
>
> You are selecting the lights object, but the properties you want to
> edit are for the directional object, which in this case is LIGHT_1:
>
> IDL> light=itool_obj->getbyidentifier('/TOOLS/SURFACE TOOL/WINDOW/
> VIEW_1/VISUALIZATION LAYER/LIGHTS/LIGHT')
> IDL> light->getproperty,name=name & print,name
> Ambient Light
> IDL> light=itool_obj->getbyidentifier('/TOOLS/SURFACE TOOL/WINDOW/
> VIEW_1/VISUALIZATION LAYER/LIGHTS/LIGHT_1')
> IDL> light->getproperty,name=name & print,name
> Directional Light
>
> With the object for the directional light, I can move and point it
> around with
>
> light->setproperty,location=[1,1,-1],direction=[-1,-1,1]
>
> Note that it may take a
>
> itool_obj->commitactions
>
> for the image to be updated after you change the objects.

Thanks pp!

for pointing out my error, as always it is teh simple error that
stumps me.

For anyone reading this, I ended up just defininn my "lights_id"
idebntifier to be that of Light_1 using this line of code in place of
the code in my second post;

lights_id = itool_obj -> FindIdentifiers('*LIGHT_1', /VISUALIZATIONS)

it now works a treat and looks good to boot.

Thanks again Ken and PP.

Subject: Re: Isurface, inverting the axis without changing default lighting direction
Posted by penteado on Wed, 05 May 2010 19:38:58 GMT
View Forum Message <> Reply to Message

On May 5, 3:24 pm, "Kenneth P. Bowman" <k-bow...@null.edu> wrote:
> In article
> < f460e955-39e9-40d4-be31-0beb89b53...@u20g2000pru.googlegroup s.com >,
> cameron bowles <cameronbowle...@gmail.com> wrote:
>
>> For anyone reading this, I ended up just defininn my "lights_id"
>> idebntifier to be that of Light_1 using this line of code in place of
>> the code in my second post;
>
>> lights_id = itool_obj -> FindIdentifiers('*LIGHT_1', /VISUALIZATIONS)
>
>> it now works a treat and looks good to boot.
>
>> Thanks again Ken and PP.
>
> I just want to make clear that this does not mean that I
> actually understand anything about how the iTools work. ;-)

Yes, they are a bit complicated. As far as I know (somebody please correct this if I am wrong), the only book on it is the iTool Developer's Guide, in IDL's help system.

In this particular case, the difference between the object with id (...)/LIGHTS and the other two (with (...)/LIGHTS/LIGHT and (...)/LIGHTS/LIGHT_1) is that (...)/LIGHTS is an _IDLITVISUALIZATION, which is there as a container for the lights, to keep all the objects in a hierarchy. The ambient and directional lights are IDLITVISLIGHT objects, which are the actual lights. If you ask one of these two for their parent, you get the object with the (...)/LIGHTS id:

```
IDL> light_1=ot->getbyidentifier('/TOOLS/SURFACE TOOL/WINDOW/VIEW_1/
VISUALIZATION LAYER/LIGHTS/LIGHT_1')
IDL> lights=ot->getbyidentifier('/TOOLS/SURFACE TOOL/WINDOW/VIEW_1/
VISUALIZATION LAYER/LIGHTS')
IDL> light_1->getproperty,parent=par & print, par eq lights
   1
IDL> print,lights->findidentifiers()
/TOOLS/SURFACE TOOL/WINDOW/VIEW_1/VISUALIZATION LAYER/LIGHTS/LIGHT
/TOOLS/SURFACE TOOL/WINDOW/VIEW_1/VISUALIZATION LAYER/LIGHTS/LIGHT_1
IDL> help,lights
LIGHTS          OBJREF    = <ObjHeapVar18919(_IDLITVISUALIZATION)>
IDL> help,light_1
LIGHT_1         OBJREF    = <ObjHeapVar19107(IDLITVISLIGHT)>
```