Subject: Re: more log-scaled images
Posted by Wout De Nolf on Tue, 16 Dec 2008 09:18:00 GMT
View Forum Message <> Reply to Message

On Mon, 15 Dec 2008 19:00:55 -0800 (PST), "ben.bighair"
<ben.bighair@gmail.com> wrote:

> I have been refreshing my memory on how to show an image with one (or
> both) dimensions log-scaled.

Have a look at this:
http://hea-www.harvard.edu/~pgrigis/idl_stuff/pg_plotimage.p ro

Subject: Re: more log-scaled images
Posted by ben.bighair on Tue, 16 Dec 2008 16:42:28 GMT
View Forum Message <> Reply to Message

On Dec 16, 4:18 am, Wox <s...@nomail.com> wrote:
> On Mon, 15 Dec 2008 19:00:55 -0800 (PST), "ben.bighair"
>
> <ben.bigh...@gmail.com> wrote:
>> I have been refreshing my memory on how to show an image with one (or
>> both) dimensions log-scaled.
>
> Have a look at this:http://hea-www.harvard.edu/~pgrigis/idl_stuff/pg_plotim age.pro

Hi,

Thanks. Yes, I was trying to model my routine on that one. My goal
is to exploit IDL's object graphics capability to work with images of
any interleaving (I'm not sure that is a real word, but you know what
I mean.) In the end, I want a generic routine for smushing the image
instead always plotting which I can do later if needed.

I discovered the issue. Surfaces are defined by the node locations
and texture mapping occurs from the node to the next node to the right
and up. Since my surface ran 1 - n, the last node had nowhere to draw
the texture map to, so the texture drawing rightly truncated at the
last node to at the right and at the top. I think this is the similar
to the issues we all encounter when using the POLYFILLV routine that
defines a pixel location as the lower left.

The trick is to make the surface onto which the image is mapped one
node bigger than the image dimensions. For example, if the image
dimensions are [100,200] then make the surface [101, 201] - and viola!

I'll clean up what I have and post it when I can.

Thanks again!
Ben

P.S. Whew!

Subject: Re: more log-scaled images
Posted by ben.bighair on Tue, 16 Dec 2008 16:42:36 GMT
View Forum Message <> Reply to Message

On Dec 16, 4:18 am, Wox <s...@nomail.com> wrote:
> On Mon, 15 Dec 2008 19:00:55 -0800 (PST), "ben.bighair"
>
> <ben.bigh...@gmail.com> wrote:
>> I have been refreshing my memory on how to show an image with one (or
>> both) dimensions log-scaled.
>
> Have a look at this:http://hea-www.harvard.edu/~pgrigis/idl_stuff/pg_plotim age.pro

Hi,

Thanks. Yes, I was trying to model my routine on that one. My goal is to exploit IDL's object graphics capability to work with images of any interleaving (I'm not sure that is a real word, but you know what I mean.) In the end, I want a generic routine for smushing the image instead always plotting which I can do later if needed.

I discovered the issue. Surfaces are defined by the node locations and texture mapping occurs from the node to the next node to the right and up. Since my surface ran 1 - n, the last node had nowhere to draw the texture map to, so the texture drawing rightly truncated at the last node to at the right and at the top. I think this is the similar to the issues we all encounter when using the POLYFILLV routine that defines a pixel location as the lower left.

The trick is to make the surface onto which the image is mapped one node bigger than the image dimensions. For example, if the image dimensions are [100,200] then make the surface [101, 201] - and viola!

I'll clean up what I have and post it when I can.

Thanks again!
Ben

P.S. Whew!

Subject: Re: more log-scaled images
Posted by ben.bighair on Tue, 16 Dec 2008 18:23:01 GMT
View Forum Message <> Reply to Message

On Dec 16, 11:42 am, "ben.bighair" <ben.bigh...@gmail.com> wrote:
> On Dec 16, 4:18 am, Wox <s...@nomail.com> wrote:
>
>> On Mon, 15 Dec 2008 19:00:55 -0800 (PST), "ben.bighair"
>
>> <ben.bigh...@gmail.com> wrote:
>>> I have been refreshing my memory on how to show an image with one (or
>>> both) dimensions log-scaled.
>
>> Have a look at this:http://hea-www.harvard.edu/~pgrigis/idl_stuff/pg_plotim age.pro
>
> Hi,
>
> Thanks. Yes, I was trying to model my routine on that one. My goal
> is to exploit IDL's object graphics capability to work with images of
> any interleaving (I'm not sure that is a real word, but you know what
> I mean.) In the end, I want a generic routine for smushing the image
> instead always plotting which I can do later if needed.
>
> I discovered the issue. Surfaces are defined by the node locations
> and texture mapping occurs from the node to the next node to the right
> and up. Since my surface ran 1 - n, the last node had nowhere to draw
> the texture map to, so the texture drawing rightly truncated at the
> last node to at the right and at the top. I think this is the similar
> to the issues we all encounter when using the POLYFILLV routine that
> defines a pixel location as the lower left.
>
> The trick is to make the surface onto which the image is mapped one
> node bigger than the image dimensions. For example, if the image
> dimensions are [100,200] then make the surface [101, 201] - and viola!
>
> I'll clean up what I have and post it when I can.
>
> Thanks again!
> Ben
>
> P.S. Whew!

Hi,

Here is what I came up with: http://www.tidewater.net/~pemaquid/hbb_logimage.pro

To run an example...

```
IDL> .compile hbb_logimage
IDL> test_logimage
```

Cheers,
Ben

Subject: Re: more log-scaled images
Posted by ben.bighair on Tue, 16 Dec 2008 18:23:08 GMT
View Forum Message <> Reply to Message

On Dec 16, 11:42 am, "ben.bighair" <ben.bigh...@gmail.com> wrote:
> On Dec 16, 4:18 am, Wox <s...@nomail.com> wrote:
>
>> On Mon, 15 Dec 2008 19:00:55 -0800 (PST), "ben.bighair"
>
>> <ben.bigh...@gmail.com> wrote:
>>> I have been refreshing my memory on how to show an image with one (or
>>> both) dimensions log-scaled.
>
>> Have a look at this:http://hea-www.harvard.edu/~pgrigis/idl_stuff/pg_plotim age.pro
>
> Hi,
>
> Thanks. Yes, I was trying to model my routine on that one. My goal
> is to exploit IDL's object graphics capability to work with images of
> any interleaving (I'm not sure that is a real word, but you know what
> I mean.) In the end, I want a generic routine for smushing the image
> instead always plotting which I can do later if needed.
>
> I discovered the issue. Surfaces are defined by the node locations
> and texture mapping occurs from the node to the next node to the right
> and up. Since my surface ran 1 - n, the last node had nowhere to draw
> the texture map to, so the texture drawing rightly truncated at the
> last node to at the right and at the top. I think this is the similar
> to the issues we all encounter when using the POLYFILLV routine that
> defines a pixel location as the lower left.
>
> The trick is to make the surface onto which the image is mapped one
> node bigger than the image dimensions. For example, if the image
> dimensions are [100,200] then make the surface [101, 201] - and viola!
>
> I'll clean up what I have and post it when I can.
>
> Thanks again!
> Ben
>
> P.S. Whew!

Hi,

Here is what I came up with: http://www.tidewater.net/~pemaquid/hbb_logimage.pro

To run an example...

```
IDL> .compile hbb_logimage
IDL> test_logimage
```

Cheers,
Ben

Subject: Re: more log-scaled images
Posted by Wout De Nolf on Wed, 17 Dec 2008 10:31:22 GMT

View Forum Message <> Reply to Message

On Tue, 16 Dec 2008 10:23:08 -0800 (PST), "ben.bighair"
<ben.bighair@gmail.com> wrote:

> Here is what I came up with: http://www.tidewater.net/~pemaquid/hbb_logimage.pro

I always get very depressed when doing this type of plotting :-). It's all in the details, half pixel shifts etc. I tried to compare your's with Paolo Grigis' procedure and it's different (see below).

I think you need to set xrange=[min(x),max(x)] and yrange=[min(y),max(y)] (don't ask me why). Furthermore you have the half-pixel difference and additionally some small aberration in your case (check xaxis at 3).

```
PRO TEST_LOGIMAGE
 nx = 300
 ny = 1000
 image = rebin(lindgen(1,ny),nx,ny,/sample)
 image[*,0:4] = rebin(lindgen(nx)*4,nx,5,/sample)
 image = bytscl(image)

 print,'First color x:',min(where(image[*,0] ne 0))
 print,'First color y:',min(where(image[0,*] ne 0))

 loadct,38

 window,0
 x = FINDGEN(nx)+1
 y = FINDGEN(ny)+1
 PLOT, x,y, /XLOG, /YLOG, /NODATA, XSTYLE = 1, YSTYLE = 1
 logimage = HBB_LOGIMAGE(image, x,y, /XLOG, /YLOG, $
   ODIM = odim, TRUE = true, START = start)
 TV, logimage, start[0],start[1], XSIZE = odim[0], YSIZE = odim[1]
 PLOT, x,y, /XLOG, /YLOG, /NODATA, /NOERASE, XSTYLE = 1, YSTYLE = 1

 window,1
 x = FINDGEN(nx)+1
 y = FINDGEN(ny)+1
 pg_plotimage,image,x,y,/xlog,/ylog,/xs,/ys
END
```

Subject: Re: more log-scaled images
Posted by ben.bighair on Wed, 17 Dec 2008 18:33:06 GMT
View Forum Message <> Reply to Message

On Dec 17, 5:31 am, Wox <s...@nomail.com> wrote:
> On Tue, 16 Dec 2008 10:23:08 -0800 (PST), "ben.bighair"
>
> <ben.bigh...@gmail.com> wrote:
>> Here is what I came up with:http://www.tidewater.net/~pemaquid/hbb_logimage.pro
>
> I always get very depressed when doing this type of plotting :-). It's
> all in the details, half pixel shifts etc. I tried to compare your's
> with Paolo Grigis' procedure and it's different (see below).

Nuts! It bums me out, too. But thanks for looking at it with such a careful eye.

I modified your test routine so that the image has easily identified steps - now the issue at axis = 3 jumps right out.
Sasafrasarasa....

I'm too maxed-out right now to fuss with it, but will pick it up again another time (next year!)

Thanks again,
Ben

Subject: Re: more log-scaled images
Posted by Brian Larsen on Thu, 18 Dec 2008 15:39:03 GMT
View Forum Message <> Reply to Message

To throw another version into the mix I extended Paolo Grigis' routine to include a colorbar and more keywords. It is still incomplete but may be worth a look. (It requires several solarsoft routines).

```
IDL>  nx = 300
IDL>  ny = 1000
IDL>  image = BYTSCL(BYTSCL(DIST(nx,ny), TOP = 32))
% Compiled module: DIST.
IDL>  x = FINDGEN(nx)+1
IDL>  y = FINDGEN(ny)+1
IDL> imagesc, image, x, y, /xlog, /ylog, /xstyle
% Compiled module: IMAGESC.
% Compiled module: TO_EPS.
% Compiled module: SETDEFAULTVALUE.
% Compiled module: PG_PLOTIMAGE.
% Compiled module: INTERPOL.
% Compiled module: COLORBAR.
% Compiled module: CONGRID.
```

http://people.bu.edu/balarsen/IDLdoc/imagesc.html

If by some odd chance you use this and make any improvements (or find horrific mistakes) let me know and I will incorporate them.

Cheers,

Brian

 --- --------------
Brian Larsen
Boston University
Center for Space Physics
http://people.bu.edu/balarsen/Home/IDL