
Subject: Re: linear-space image converted to log-space
Posted by [Vince Hradil](#) on Sat, 23 Feb 2008 23:32:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Feb 23, 4:45 pm, jkj <ke...@vexona.com> wrote:

> Hi,
>
> I have a need to reproduce an image like this:
>
> http://diamondhead.org/linear_logspace.png
>
> beginning with data like this:
>
> http://diamondhead.org/linear_linearspace.png
>
> Using logarithmic scaling, the first image (produced by a custom
> graphics package) expands lower frequency components while contracting
> the higher frequency components. To do this in object graphics, I
> believe I will have to reconstruct the image data so that the lower
> frequency [the bottom] information is replicated according to a log
> scaling scheme.
>
> I can think of some [tortured] ways to do this but keep thinking there
> must be some sophisticated way to handle this in IDL. Any thoughts
> would be appreciated.
>
> Thanks,
> -Kevin

Is it not just `image = alog10(image)`, with the proper use of `finite()`
and `where()` to trap errors?

Subject: Re: linear-space image converted to log-space
Posted by [jkj](#) on Sun, 24 Feb 2008 02:04:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

>
> Is it not just `image = alog10(image)`, with the proper use of `finite()`
> and `where()` to trap errors?

No [but thanks!], that would take the data into log space and leave
the same number of pixels representing a given "row/frequency" of the
data. I need to expand the low frequency portion of the image so that
portions which only received 1 pixel now are smeared out over multiple
pixels in the low freq. range. The data itself does not change, just
the percentage of the image it receives - low frequency elements are
smeared out broadly while high frequency elements (less interesting in

this case) are highly compressed.

I suspect I should be able to do this with interpolate, so that is what I'm trying to pencil out.

Subject: Re: linear-space image converted to log-space
Posted by [ben.bighair](#) on Sun, 24 Feb 2008 02:26:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Feb 23, 5:45 pm, jkj <ke...@vexona.com> wrote:

> Hi,
>
> I have a need to reproduce an image like this:
>
> http://diamondhead.org/linear_logspace.png
>
> beginning with data like this:
>
> http://diamondhead.org/linear_linearspace.png
>
> Using logarithmic scaling, the first image (produced by a custom
> graphics package) expands lower frequency components while contracting
> the higher frequency components. To do this in object graphics, I
> believe I will have to reconstruct the image data so that the lower
> frequency [the bottom] information is replicated according to a log
> scaling scheme.
>
> I can think of some [tortured] ways to do this but keep thinking there
> must be some sophisticated way to handle this in IDL. Any thoughts
> would be appreciated.
>
> Thanks,
> -Kevin

Hi,

I think I would use a flat filled IDLgrSurface that is projected as one might view an image - with a log scaled y axis. Then use the TEXTURE_MAP property to show the image data.

The example below uses XOBJVIEW to show the same image twice. The bottom is "regular" and the top is stretched like in your example. Note that I had to stretch the second model so you can see the effect. I can't recall at this late hour how to get better control of that, but it can be done much better.

Ben

```

**BEGIN
rose = READ_IMAGE(FILEPATH('rose.jpg',
SUBDIRECTORY=['examples','data']))
dim = SIZE(rose,/DIM)
nx = dim[1] & ny = dim[2]
x = findgen(nx)+1
y = findgen(ny)+1
s = replicate(1,nx,ny)

;the first surface is "regular"
o1 = OBJ_NEW("IDLgrSurface", s, x, y, $
  color = [255,255,255], style = 2, $
  texture_map = obj_new("IDLgrImage", rose))
model_1 = OBJ_NEW("IDLgrModel")
model_1->Add,o1
model_1->Translate, 0, -ny/2. - 10, 0

;the second surface is log scale in y
o2 = OBJ_NEW("IDLgrSurface", s, x, alog10(y), $
  color = [255,255,255], style = 2, $
  texture_map = obj_new("IDLgrImage", rose))
model_2 = OBJ_NEW("IDLgrModel")
model_2->Add,o2
model_2->Scale, 1, 100, 1 ;<<<< cheat here to make effect big enough
to see
model_2->Translate, 0,ny/2. + 10, 0

xobjview, [model_1, model_2] , /BLOCK

OBJ_DESTROY, [model_1, model_2]
**END

```

Subject: Re: linear-space image converted to log-space
 Posted by [jkj](#) on Sun, 24 Feb 2008 03:33:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Feb 23, 8:26 pm, "ben.bighair" <ben.bigh...@gmail.com> wrote:
 > On Feb 23, 5:45 pm, jkj <ke...@vexona.com> wrote:
 >
 >
 >
 >> Hi,
 >
 >> I have a need to reproduce an image like this:
 >

```

>> http://diamondhead.org/linear_logspace.png
>
>> beginning with data like this:
>
>> http://diamondhead.org/linear_linearspace.png
>
>> Using logarithmic scaling, the first image (produced by a custom
>> graphics package) expands lower frequency components while contracting
>> the higher frequency components. To do this in object graphics, I
>> believe I will have to reconstruct the image data so that the lower
>> frequency [the bottom] information is replicated according to a log
>> scaling scheme.
>
>> I can think of some [tortured] ways to do this but keep thinking there
>> must be some sophisticated way to handle this in IDL. Any thoughts
>> would be appreciated.
>
>> Thanks,
>> -Kevin
>
> Hi,
>
> I think I would use a flat filled IDLgrSurface that is projected as
> one might view an image - with a log scaled y axis. Then use the
> TEXTURE_MAP property to show the image data.
>
> The example below uses XOBJVIEW to show the same image twice. The
> bottom is "regular" and the top is stretched like in your example.
> Note that I had to stretch the second model so you can see the
> effect. I can't recall at this late hour how to get better control of
> that, but it can be done much better.
>
> Ben
>
> **BEGIN
> rose = READ_IMAGE(FILEPATH('rose.jpg',
> SUBDIRECTORY=['examples','data']))
> dim = SIZE(rose,/DIM)
> nx = dim[1] & ny = dim[2]
> x = findgen(nx)+1
> y = findgen(ny)+1
> s = replicate(1,nx,ny)
>
> ;the first surface is "regular"
> o1 = OBJ_NEW("IDLgrSurface", s, x, y, $
> color = [255,255,255], style = 2, $
> texture_map = obj_new("IDLgrImage", rose))
> model_1 = OBJ_NEW("IDLgrModel")

```

```
> model_1->Add,o1
> model_1->Translate, 0, -ny/2. - 10, 0
>
> ;the second surface is log scale in y
> o2 = OBJ_NEW("IDLgrSurface", s, x, alog10(y), $
>   color = [255,255,255], style = 2, $
>   texture_map = obj_new("IDLgrImage", rose))
> model_2 = OBJ_NEW("IDLgrModel")
> model_2->Add,o2
> model_2->Scale, 1, 100, 1 ;<<<< cheat here to make effect big enough
> to see
> model_2->Translate, 0,ny/2. + 10, 0
>
> xobjview, [model_1, model_2] , /BLOCK
>
> OBJ_DESTROY, [model_1, model_2]
> **END
```

My hero! :-)

Very gracious of you - thanks!
-Kevin

Subject: Re: linear-space image converted to log-space
Posted by [jkj](#) on Sun, 24 Feb 2008 04:46:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

```
> model_2->Add,o2
> model_2->Scale, 1, 100, 1 ;<<<< cheat here to make effect big enough
> to see
```

to make them both the same height:
scale = ny / alog10(ny)
model_2->Scale, 1, scale, 1
