Subject: Re: Fanning's LogScl routine + Colorbar??
Posted by pgrigis on Fri, 23 Jul 2010 17:46:26 GMT
View Forum Message <> Reply to Message

On Jul 23, 1:23 pm, Joe Daal <joe.d...@gmail.com> wrote:
> Hi,
>
> I am using the logscl to enhance the contrast of an image, something
> like:
>
> loadct,39
> image = alog10(lin_image)
> imdisp, logscl(image, min=min(image), max=max(image), Exponent=8,
> Mean=0.65)
>
> where image values vary from -1.34 to +2.05, with zeroes included.
> The image looks nice for what I want, but how do I reflect a correct
> colorbar, either for real values or the scaled ones? What is
> logarithmic, the ticks or the colors?

Hi,

I would argue for both.

Example using
 http://hea-www.cfa.harvard.edu/~pgrigis/idl_stuff/pg_plotima ge.pro
below will show different color bars.

```
;create color bar array
x=findgen(1000)
y=findgen(256)
z=rebin(x,1000,256)

;case 1: log axis and log colors - that's what you want
 pg_plotimage,alog(z),x,y,xrange=[1,1000],/xlog,/ylog,yrange= [1,255],/
xsty,/yst

;case 2: linear axis and log colors
;this is a "correct" but undesirable representation: the beholder of
the plot will
;not be able to make out the colors for low values of the x axis
 pg_plotimage,alog(z),x,y,xrange=[1,1000],/ylog,yrange=[1,255 ],/xsty,/
yst

;case 3: linear axis and linear colors - this is what happens if you
do not log scale the image
 pg_plotimage,z,x,y,xrange=[1,1000],/ylog,yrange=[1,255],/xst y,/yst
```

;case 4: log axis and linear colors - this is what happens if you do
not log scale the image
;but is shown in a not particularly useful representation
 pg_plotimage,z,x,y,xrange=[1,1000],/ylog,yrange=[1,255],/xst y,/yst,/
xlog

Ciao,
Paolo
>
> Thanks,
> Joe

Subject: Re: Fanning's LogScl routine + Colorbar??
Posted by David Fanning on Fri, 23 Jul 2010 19:38:48 GMT
View Forum Message <> Reply to Message

Joe Daal writes:

> I am using the logscl to enhance the contrast of an image, something
> like:
>
> loadct,39
> image = alog10(lin_image)
> imdisp, logscl(image, min=min(image), max=max(image), Exponent=8,
> Mean=0.65)
>
> where image values vary from -1.34 to +2.05, with zeroes included.
> The image looks nice for what I want, but how do I reflect a correct
> colorbar, either for real values or the scaled ones? What is
> logarithmic, the ticks or the colors?

If you are going to display a color bar, wouldn't you
have to transform the color bar values in the same
way to transform the image?

I am thinking code like this:

```
;********************************************************** **
image = LoadData(11)
freqDomainImage = FFT(image, -1)
power = SHIFT(ALOG(ABS(freqDomainImage)), 124, 124)
power = power - Min(power)

minmax, power
ctload, 4, /brewer, /reverse
tvlct, r, g, b, /get
```

```
rr = scale_vector(float(r), min(power), max(power), $
   MIN=0, MAX=255)
gg = scale_vector(float(g), min(power), max(power), $
   MIN=0, MAX=255)
bb = scale_vector(float(b), min(power), max(power),  $
   MIN=0, MAX=255)
rrr = logscl(rr, MEAN=0.45)
ggg = logscl(gg, MEAN=0.45)
bbb = logscl(bb, MEAN=0.45)
tvlct, rrr, ggg, bbb
cindex

window, xsize=500, ysize=500, Title='Log Display'
TVImage, LogScl(power, MEAN=0.45), $
   position=[0.1, 0.1, 0.9, 0.75], /erase
colorbar, range=[min(power), Max(power)], $
   division=4, format='(f0.2)'

ctload, 4, /brewer, /reverse
window, 1, xsize=500, ysize=500, Title='Normal Display'
TVImage, BytScl(power), position=[0.1, 0.1, 0.9, 0.75], /erase
colorbar, range=[min(power), Max(power)], $
   division=4, format='(f0.2)'
END
 ;************************************************************ **
```

Here are the results from running this program:

  http://www.dfanning.com/misc/logscl.png

Cheers,

David

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

Subject: Re: Fanning's LogScl routine + Colorbar??
Posted by pgrigis on Fri, 23 Jul 2010 19:51:35 GMT

View Forum Message <> Reply to Message

On Jul 23, 3:38 pm, David Fanning <n...@dfanning.com> wrote:
> Joe Daal writes:
>> I am using the logscl to enhance the contrast of an image, something

>> like:
>
>> loadct,39
>> image = alog10(lin_image)
>> imdisp, logscl(image, min=min(image), max=max(image), Exponent=8,
>> Mean=0.65)
>
>> where image values vary from -1.34 to +2.05, with zeroes included.
>> The image looks nice for what I want, but how do I reflect a correct
>> colorbar, either for real values or the scaled ones? What is
>> logarithmic, the ticks or the colors?
>
> If you are going to display a color bar, wouldn't you
> have to transform the color bar values in the same
> way to transform the image?
>
> I am thinking code like this:
>
> ;** **
> image = LoadData(11)
> freqDomainImage = FFT(image, -1)
> power = SHIFT(ALOG(ABS(freqDomainImage)), 124, 124)
> power = power - Min(power)
>
> minmax, power
> ctload, 4, /brewer, /reverse
> tvlct, r, g, b, /get
> rr = scale_vector(float(r), min(power), max(power), $
> MIN=0, MAX=255)
> gg = scale_vector(float(g), min(power), max(power), $
> MIN=0, MAX=255)
> bb = scale_vector(float(b), min(power), max(power), $
> MIN=0, MAX=255)
> rrr = logscl(rr, MEAN=0.45)
> ggg = logscl(gg, MEAN=0.45)
> bbb = logscl(bb, MEAN=0.45)
> tvlct, rrr, ggg, bbb
> cindex
>
> window, xsize=500, ysize=500, Title='Log Display'
> TVImage, LogScl(power, MEAN=0.45), $
> position=[0.1, 0.1, 0.9, 0.75], /erase
> colorbar, range=[min(power), Max(power)], $
> division=4, format='(f0.2)'
>
> ctload, 4, /brewer, /reverse
> window, 1, xsize=500, ysize=500, Title='Normal Display'
> TVImage, BytScl(power), position=[0.1, 0.1, 0.9, 0.75], /erase

> colorbar, range=[min(power), Max(power)], $
>   division=4, format='(f0.2)'
> END
> ;** **
>
> Here are the results from running this program:
>
> http://www.dfanning.com/misc/logscl.png

Hi David,
it looks like two different color tables have been used for the two
images:
the one on the left looks more green and the one on the right looks
more blue.
This makes it harder to see what is really going on.

I tried your commands with black and white colors, but the bar looks
then identical in both plots...

Ciao,
Paolo

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

Subject: Re: Fanning's LogScl routine + Colorbar??
Posted by pgrigis on Fri, 23 Jul 2010 19:53:29 GMT
View Forum Message <> Reply to Message

On Jul 23, 3:51 pm, Paolo <pgri...@gmail.com> wrote:
> On Jul 23, 3:38 pm, David Fanning <n...@dfanning.com> wrote:
>
>
>

>> Joe Daal writes:
>>> I am using the logscl to enhance the contrast of an image, something
>>> like:
>
>>> loadct,39
>>> image = alog10(lin_image)
>>> imdisp, logscl(image, min=min(image), max=max(image), Exponent=8,
>>> Mean=0.65)
>
>>> where image values vary from -1.34 to +2.05, with zeroes included.
>>> The image looks nice for what I want, but how do I reflect a correct
>>> colorbar, either for real values or the scaled ones? What is
>>> logarithmic, the ticks or the colors?
>
>> If you are going to display a color bar, wouldn't you
>> have to transform the color bar values in the same
>> way to transform the image?
>
>> I am thinking code like this:
>
>> ;** **
>> image = LoadData(11)
>> freqDomainImage = FFT(image, -1)
>> power = SHIFT(ALOG(ABS(freqDomainImage)), 124, 124)
>> power = power - Min(power)
>
>> minmax, power
>> ctload, 4, /brewer, /reverse
>> tvlct, r, g, b, /get
>> rr = scale_vector(float(r), min(power), max(power), $
>>   MIN=0, MAX=255)
>> gg = scale_vector(float(g), min(power), max(power), $
>>   MIN=0, MAX=255)
>> bb = scale_vector(float(b), min(power), max(power), $
>>   MIN=0, MAX=255)
>> rrr = logscl(rr, MEAN=0.45)
>> ggg = logscl(gg, MEAN=0.45)
>> bbb = logscl(bb, MEAN=0.45)
>> tvlct, rrr, ggg, bbb
>> cindex
>
>> window, xsize=500, ysize=500, Title='Log Display'
>> TVImage, LogScl(power, MEAN=0.45), $
>>   position=[0.1, 0.1, 0.9, 0.75], /erase
>> colorbar, range=[min(power), Max(power)], $
>>   division=4, format='(f0.2)'
>
>> ctload, 4, /brewer, /reverse

>> window, 1, xsize=500, ysize=500, Title='Normal Display'
>> TVImage, BytScl(power), position=[0.1, 0.1, 0.9, 0.75], /erase
>> colorbar, range=[min(power), Max(power)], $
>> division=4, format='(f0.2)'
>> END
>> ;** **
>
>> Here are the results from running this program:
>
>> http://www.dfanning.com/misc/logscl.png
>
> Hi David,
> it looks like two different color tables have been used for the two
> images:
> the one on the left looks more green and the one on the right looks
> more blue.
> This makes it harder to see what is really going on.
>
> I tried your commands with black and white colors, but the bar looks
> then identical in both plots...

OK - i see why - you scaled the colors and not the data.
Still I would argue that using log scaling should not alter the
original colors?

Ciao,
Paolo

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

Subject: Re: Fanning's LogScl routine + Colorbar??
Posted by pgrigis on Fri, 23 Jul 2010 20:10:10 GMT
View Forum Message <> Reply to Message

And one last message for the day :)

The two color bars plotted here

http://hea-www.cfa.harvard.edu/~pgrigis/idl_stuff/colbars.pn g

represent the same color scaling of table 5 (stretched
logarithmically)
but the linear representation does compress the "interesting" bits
of the color bar quite strongly on the left side - while the log
representation seems to do a better job of display the quick variation
at the left end.

Ciao,
Paolo

On Jul 23, 3:38 pm, David Fanning <n...@dfanning.com> wrote:
> Joe Daal writes:
>> I am using the logscl to enhance the contrast of an image, something
>> like:
>
>> loadct,39
>> image = alog10(lin_image)
>> imdisp, logscl(image, min=min(image), max=max(image), Exponent=8,
>> Mean=0.65)
>
>> where image values vary from -1.34 to +2.05, with zeroes included.
>> The image looks nice for what I want, but how do I reflect a correct
>> colorbar, either for real values or the scaled ones? What is
>> logarithmic, the ticks or the colors?
>
> If you are going to display a color bar, wouldn't you
> have to transform the color bar values in the same
> way to transform the image?
>
> I am thinking code like this:
>
> ;** **
> image = LoadData(11)
> freqDomainImage = FFT(image, -1)
> power = SHIFT(ALOG(ABS(freqDomainImage)), 124, 124)
> power = power - Min(power)
>
> minmax, power

```
>  ctload, 4, /brewer, /reverse
>  tvlct, r, g, b, /get
>  rr = scale_vector(float(r), min(power), max(power), $
>     MIN=0, MAX=255)
>  gg = scale_vector(float(g), min(power), max(power), $
>     MIN=0, MAX=255)
>  bb = scale_vector(float(b), min(power), max(power),  $
>     MIN=0, MAX=255)
>  rrr = logscl(rr, MEAN=0.45)
>  ggg = logscl(gg, MEAN=0.45)
>  bbb = logscl(bb, MEAN=0.45)
>  tvlct, rrr, ggg, bbb
>  cindex
>
>  window, xsize=500, ysize=500, Title='Log Display'
>  TVImage, LogScl(power, MEAN=0.45), $
>     position=[0.1, 0.1, 0.9, 0.75], /erase
>  colorbar, range=[min(power), Max(power)], $
>     division=4, format='(f0.2)'
>
>  ctload, 4, /brewer, /reverse
>  window, 1, xsize=500, ysize=500, Title='Normal Display'
>  TVImage, BytScl(power), position=[0.1, 0.1, 0.9, 0.75], /erase
>  colorbar, range=[min(power), Max(power)], $
>     division=4, format='(f0.2)'
>  END
>   ;************************************************************ **
>
>  Here are the results from running this program:
>
>    http://www.dfanning.com/misc/logscl.png
>
>  Cheers,
>
>  David
>
>  --
>  David Fanning, Ph.D.
>  Fanning Software Consulting, Inc.
>  Coyote's Guide to IDL Programming:http://www.dfanning.com/
>  Sepore ma de ni thui. ("Perhaps thou speakest truth.")
```

Subject: Re: Fanning's LogScl routine + Colorbar??
Posted by David Fanning on Fri, 23 Jul 2010 20:32:18 GMT
View Forum Message <> Reply to Message

Paolo writes:

> The two color bars plotted here
>
> http://hea-www.cfa.harvard.edu/~pgrigis/idl_stuff/colbars.pn g
>
> represent the same color scaling of table 5 (stretched
> logarithmically)
> but the linear representation does compress the "interesting" bits
> of the color bar quite strongly on the left side - while the log
> representation seems to do a better job of display the quick variation
> at the left end.

Yes, I was hoping someone might have the definitive
answer here, because this whole topic confuses me, too.
I do know it is quite easy to produce "normal" log scaled
data and color tables. It gets quite a bit harder, it seems
to me, when using these fairly arbitrary log-scale functions.

Ah! Maybe it is like this:

```
tvlct, r, g, b, /get
rr = scale_vector(float(r), min(power), max(power), MIN=0, MAX=255)
gg = scale_vector(float(g), min(power), max(power), MIN=0, MAX=255)
bb = scale_vector(float(b), min(power), max(power), MIN=0, MAX=255)
rrr = logscl(rr, MEAN=0.45)
ggg = logscl(gg, MEAN=0.45)
bbb = logscl(bb, MEAN=0.45)
tvlct, r[rrr], g[ggg], b[bbb]
```

That probably makes more sense!

See the new results here:

http://www.dfanning.com/misc/logscl_new.png

Cheers,

David

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

Subject: Re: Fanning's LogScl routine + Colorbar??
Posted by David Fanning on Sat, 24 Jul 2010 15:40:53 GMT
View Forum Message <> Reply to Message

Joe Daal writes:

> I am using the logscl to enhance the contrast of an image, something
> like:
>
> loadct,39
> image = alog10(lin_image)
> imdisp, logscl(image, min=min(image), max=max(image), Exponent=8,
> Mean=0.65)
>
> where image values vary from -1.34 to +2.05, with zeroes included.
> The image looks nice for what I want, but how do I reflect a correct
> colorbar, either for real values or the scaled ones? What is
> logarithmic, the ticks or the colors?

After thinking about this some more this morning, I decided
I would write an article about it. Normally, when I write an
article I am about 90% sure I know what I'm talking about.
(My wife says I have been overly optimistic my whole life!)

In this case, it's more like 50%. But I figure the worst
thing that could happen to me would be that I might learn
something. ;-)

You can find the article here:

http://www.dfanning.com/ip_tips/logscaledbar.html

Cheers,

David

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

Subject: Re: Fanning's LogScl routine + Colorbar??
Posted by pgrigis on Mon, 26 Jul 2010 14:59:43 GMT
View Forum Message <> Reply to Message

On Jul 24, 11:40 am, David Fanning <n...@dfanning.com> wrote:

> Joe Daal writes:
>> I am using the logscl to enhance the contrast of an image, something
>> like:
>
>> loadct,39
>> image = alog10(lin_image)
>> imdisp, logscl(image, min=min(image), max=max(image), Exponent=8,
>> Mean=0.65)
>
>> where image values vary from -1.34 to +2.05, with zeroes included.
>> The image looks nice for what I want, but how do I reflect a correct
>> colorbar, either for real values or the scaled ones? What is
>> logarithmic, the ticks or the colors?
>
> After thinking about this some more this morning, I decided
> I would write an article about it. Normally, when I write an
> article I am about 90% sure I know what I'm talking about.
> (My wife says I have been overly optimistic my whole life!)
>
> In this case, it's more like 50%. But I figure the worst
> thing that could happen to me would be that I might learn
> something. ;-)
>
> You can find the article here:
>
> http://www.dfanning.com/ip_tips/logscaledbar.html
>

Well, let me try to explain my argument.

Basically, there are 2 (mutually exclusive) ways to proceed
with this:
a) rescaling the color tables
b) rescaling the data

These 2 ways are *not* mathematically equivalent, as explained below.
In general a) will lead to the rescaled image having (in some pixels)
colors that were not present in the image before rescaling, while b)
will not. Think of a) as being a more powerful transformation - but
with
great power come great responsibility as you all know :)

Personally I dislike a) creating new colors not present in the
original
image and therefore I stick to b).

Here the mathematical argument:

what we think of a "color table" is the combination of 2 operations.

The first is the process of assigning every pixel of the image an
index between
0 and N-1 (N can be any number, 256 is often used but it's important
to
realize that this number is not tied to that). You can think of this
as
a function f going from the real numbers to [0,1,...,N-1].

The second is the process that assigns every index a color (in the
case
of the current hardware, a color is a triple of bytes R, G, B). This
is a set of three functions R,G,B going from [0,1,...,N-1] to
[0,1,...,255].
The fact that we have 256 shades for 3 main colors is fixed and
limited
by the hardware.

To display an image "im" we have to compute R(f(im)),G(f(im)),B(f(im))
for
all pixels - this is what we mean by using a color table.

Now in case we are not happy with the result we can try rescaling
using the a) or b) process.

The a) rescaling means we have functions R2,G2,B2 that go from
[0,1,...,255]
to [0,1,...,255]. We then display the image R2(R(f(im))),
G2(G(f(im))), B2(B(f(im))).
Depending on the details of R2,G2,B2 it's quite easy to create new
colors by this
transformation (why? because there are only N different triples before
the
transformation, and 16777216 (!) different triples that they can be
transformed
into).

The b) rescaling means we have a function h that goes from
[0,1,...,N-1] to
[0,1,...,N-1]. We then display the image
R(h(f(im)),G(h(f(im)),B(h(f(im))).
Because the R,G,B functions themselves are not changed, the new image
can only
consist of colors in the color table (i.e. no new colors will appear).

That sums it up... Hopefully this helps shed some light (or muddle

up the water even more instead).

Ciao,
Paolo

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

Subject: Re: Fanning's LogScl routine + Colorbar??
Posted by Joe Daal on Thu, 29 Jul 2010 03:47:53 GMT
View Forum Message <> Reply to Message

On Jul 26, 9:59 am, Paolo <pgri...@gmail.com> wrote:
> On Jul 24, 11:40 am, David Fanning <n...@dfanning.com> wrote:
>
>
>
>> Joe Daal writes:
>>> I am using the logscl to enhance the contrast of an image, something
>>> like:
>

>>> loadct,39
>>> image = alog10(lin_image)
>>> imdisp, logscl(image, min=min(image), max=max(image), Exponent=8,
>>> Mean=0.65)
>
>>> where image values vary from -1.34 to +2.05, with zeroes included.
>>> The image looks nice for what I want, but how do I reflect a correct
>>> colorbar, either for real values or the scaled ones? What is
>>> logarithmic, the ticks or the colors?
>
>> After thinking about this some more this morning, I decided
>> I would write an article about it. Normally, when I write an
>> article I am about 90% sure I know what I'm talking about.
>> (My wife says I have been overly optimistic my whole life!)
>
>> In this case, it's more like 50%. But I figure the worst
>> thing that could happen to me would be that I might learn
>> something. ;-)
>
>> You can find the article here:
>
>>   http://www.dfanning.com/ip_tips/logscaledbar.html
>
> Well, let me try to explain my argument.
>
> Basically, there are 2 (mutually exclusive) ways to proceed
> with this:
> a) rescaling the color tables
> b) rescaling the data
>
> These 2 ways are *not* mathematically equivalent, as explained below.
> In general a) will lead to the rescaled image having (in some pixels)
> colors that were not present in the image before rescaling, while b)
> will not. Think of a) as being a more powerful transformation - but
> with
> great power come great responsibility as you all know :)
>
> Personally I dislike a) creating new colors not present in the
> original
> image and therefore I stick to b).
>
> Here the mathematical argument:
> what we think of a "color table" is the combination of 2 operations.
>
> The first is the process of assigning every pixel of the image an
> index between
> 0 and N-1 (N can be any number, 256 is often used but it's important
> to

> realize that this number is not tied to that). You can think of this
> as
> a function f going from the real numbers to [0,1,...,N-1].
>
> The second is the process that assigns every index a color (in the
> case
> of the current hardware, a color is a triple of bytes R, G, B). This
> is a set of three functions R,G,B going from [0,1,...,N-1] to
> [0,1,...,255].
> The fact that we have 256 shades for 3 main colors is fixed and
> limited
> by the hardware.
>
> To display an image "im" we have to compute R(f(im)),G(f(im)),B(f(im))
> for
> all pixels - this is what we mean by using a color table.
>
> Now in case we are not happy with the result we can try rescaling
> using the a) or b) process.
>
> The a) rescaling means we have functions R2,G2,B2 that go from
> [0,1,...,255]
> to [0,1,...,255]. We then display the image R2(R(f(im))),
> G2(G(f(im))), B2(B(f(im))).
> Depending on the details of R2,G2,B2 it's quite easy to create new
> colors by this
> transformation (why? because there are only N different triples before
> the
> transformation, and 16777216 (!) different triples that they can be
> transformed
> into).
>
> The b) rescaling means we have a function h that goes from
> [0,1,...,N-1] to
> [0,1,...,N-1]. We then display the image
> R(h(f(im)),G(h(f(im)),B(h(f(im))).
> Because the R,G,B functions themselves are not changed, the new image
> can only
> consist of colors in the color table (i.e. no new colors will appear).
>
> That sums it up... Hopefully this helps shed some light (or muddle
> up the water even more instead).
>
> Ciao,
> Paolo
>
>> Cheers,
>

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

Paolo and David,

This thread clarified it all. Thank you so much!
I'd go with scaling the data instead of color scaling.

Cheers,
-Joe
