Subject: color value interpolation from colorbar
Posted by j.coenia@gmail.com on Thu, 04 Dec 2008 17:27:14 GMT
View Forum Message <> Reply to Message

Hi,

I have a color-coded medical image.

There is a colorbar on the image, for example a red-to-yellow gradient valued from 1 to 100.

There is *not* a one-to-one correspondence between the colors in the colorbar and the colors in the image. There are many more intermediate reds, oranges, and yellows in the image than in the colorbar.

What is the correct way to interpolate gradient values to all the unknown image colors from the discrete colors/gradients defined in the colorbar?

(The documentation for INTERPOL and INTERPOLATE has not helped me so far.)

Thanks

Subject: Re: color value interpolation from colorbar
Posted by Jeremy Bailin on Fri, 05 Dec 2008 01:00:50 GMT
View Forum Message <> Reply to Message

On Dec 4, 4:35 pm, Paolo <pgri...@gmail.com> wrote:
> So your first step is to "digitize" the color bar:
>
> Create 3 arrays of r,g,b with the colors as a function of pixel size
> and an array x of the pixel index number.
>
> Then, show us a plot of r(x), g(x), b(x)
>
> Ciao,
> Paolo
>
> j.coe...@gmail.com wrote:
>> Thanks. The red-to-yellow colorbar is already on the image -- I don't
>> create it. The map of gradient values to the 3D colorspace would have
>> to be determined from this colorbar, which also has errors in it
>> (garbage in, as Dr. Fanning says.) I suppose this would be an
>> irregular grid of gradient values within the 3D colorspace.
>

>> Basically, there's a color-coded image with a colorbar *on* the
>> image. The colors, even those in the bar, often have errors because
>> they were digitized from analog tape recordings. Still, it is
>> possible to eyeball the images to determine color levels. I want the
>> computer to do something more quantitative than the eyeball analysis.
>
>

How many points do you have on the colour bar? If it's sampled finely enough, it might be okay to assume a straight line through RGB space between them. Then you could calculate the minimum distance of any desired colour from each of those interpolated segments in RGB space to find the closest segment, and then use its component in the direction of the segment.

That's only going to work if the input isn't too garbagy - you'll need points on the colour bar that sample the gradiant quite finely, and the colours on the image can't be too distant in RGB space from the locus that the colour bar defines. But if those conditions are met, this might work.

I'm imagining something like this. You have cbar_r, cbar_g and cbar_b, each of which contain nbar points and represent data values cbar_data. To find the best data value for a pixel with colour image_r, image_g and image_b (untested and unwieldy - can easily be vectorized to work on the image at once, but I'm too hungry):

```
; find the closest colourbar point
cbardist2 = (cbar_r-image_r)^2 + (cbar_g-image_g)^2 + (cbar_b-image_b)
^2
cbarmin = min(cbardist2, segment1)
case segment1 of
  0: segment2=1
  nbar-1: segment2=nbar-2
  else: segment2 = (cbardist2[segment1-1] gt cbardist2[segment1+1) ?
(segment1+1) $
        : (segment1-1)
endcase

; find component along the segment
from1toi = [image_r-cbar_r[segment1],image_g-cbar_b[segment1],image_b-
cbar_b[segment1]]
from1to2 = [cbar_r[segment2]-cbar_r[segment1],cbar_g[segment2]-cbar_g
[segment1], $
   cbar_b[segment2]-cbar_b[segment1]]
projcomp = total(from1toi * from1to2) / (norm(from1to2) * norm
(from1toi))
```

; interpolate data value
imagedata = cbar_data[segment1]*(1.-projcomp) + cbar_data[segment2]
*projcomp

-Jeremy.

Subject: Re: color value interpolation from colorbar
Posted by j.coenia@gmail.com on Fri, 05 Dec 2008 02:35:56 GMT
View Forum Message <> Reply to Message

Thanks everyone. Sorry for the delay, I had to generate the requested pics and create a Picasa web album. If there is a better way to post images to usenet forums, please let me know.

Below is the Picasa link to the two images requested by Vince and Paolo, (1) an example frame grab of a scan, and (2) its colorbar RGB plotted against colorbar location:

http://picasaweb.google.com/j.coenia/ColorInterpolation?auth key=H9iPrIqxX1c#

To answer Jeremy's question, the colorbar length is 140 pixels or so (scaled here from 1 to 100 on the x axis, which is vertical in the scan). You can see from the plot that the colorbar sampling is "garbagy." There are two very bright artifacts, at approximately x=20 and x=80. Such outliers can be tossed or smoothed out somehow I think. For simplicity, I just sampled the values down the vertical center of the colorbar, as the colorbar tends to bleed a little into the dark background near the edges (more errors).

Jeremy's answer makes some sense to me. So is it possible to reasonably guess the color levels in that artery using the colorbar on the side of the scan? I know there is no scale on the colorbar -- I've been instructed to assume linear gradient from 1 to 100. Radiologists and researchers use these colors; can the computer quantify them to extract more meaningful information?

Thanks again.

Subject: Re: color value interpolation from colorbar
Posted by j.coenia@gmail.com on Fri, 05 Dec 2008 03:23:37 GMT
View Forum Message <> Reply to Message

Oops, sorry, there's something wrong with that plot and with my artifact interpretation. It's not just a bright artifact, it must be

an error I made. The plot is good enough to give you an idea, though
-- just ignore the glitch. I'll fix the problem tomorrow at work when
I have access to the data.

Subject: Re: color value interpolation from colorbar
Posted by Peter Mason on Fri, 05 Dec 2008 04:31:41 GMT
View Forum Message <> Reply to Message

j.coenia@gmail.com wrote:
> Thanks everyone. Sorry for the delay, I had to generate the requested
> pics and create a Picasa web album. If there is a better way to post
> images to usenet forums, please let me know.
>
> Below is the Picasa link to the two images requested by Vince and
> Paolo, (1) an example frame grab of a scan, and (2) its colorbar RGB
> plotted against colorbar location:
>
> http://picasaweb.google.com/j.coenia/ColorInterpolation?auth key=H9iPrIqxX1c#
>
> To answer Jeremy's question, the colorbar length is 140 pixels or so
> (scaled here from 1 to 100 on the x axis, which is vertical in the
> scan). You can see from the plot that the colorbar sampling is
> "garbagy." There are two very bright artifacts, at approximately x=20
> and x=80. Such outliers can be tossed or smoothed out somehow I
> think. For simplicity, I just sampled the values down the vertical
> center of the colorbar, as the colorbar tends to bleed a little into
> the dark background near the edges (more errors).
>
> Jeremy's answer makes some sense to me. So is it possible to
> reasonably guess the color levels in that artery using the colorbar on
> the side of the scan? I know there is no scale on the colorbar --
> I've been instructed to assume linear gradient from 1 to 100.
> Radiologists and researchers use these colors; can the computer
> quantify them to extract more meaningful information?

Whoever made these images in the first place must have had a system, surely?
But who knows what might have happened in between printing and digitising.
For the plots, did you take a single column of colourbar pixels or did you
average across some of its width? (Averaging, median-filtering or
something like that would clean up some of the noise.)
So anyway the objective is to find a function that maps image_colour to
data_value. Judging from the plot, blue is sort-of linear while red and
green look like the're through some gamma. Along the lines of
R_now=R_orig^0.9 and G_now=G_orig^1.1 (for example). Yuk.
A brute-force solution is to use SVD, for example, to fit a polynomial to
this lot and map it to your data values. Divide your 140-element
colourbar-derived R, G, B arrays by 255.0 and try something like this.

(This has x^2 and x^3 terms for R and G to try to map some curvature.
Maybe you don't need the x^3 terms. Maybe you need more. Maybe you need
a higher term for blue too. I haven't tried this out.)

```
A=fltarr(8,140)
A[0,*]=1
A[1,*]=R & A[2,*]=G & A[3,*]=B
A[4,*]=R*R & A[5,*]=G*G
A[6,*]=R*R*R & A[7,*]=G*G*G
SVDC, A, W, U, V
CF=SVSOL( U, W, V, (findgen(140)+1.0)/1.4 )
```

(I might have columns and rows mixed up here. I'm only human.)
CF will have coefficients that you can use to get a data value given some
image pixel's RGB. Get the image pixel's R,G,B, divide by 255, and
calculate image_data_value=CF[0] + CF[1]*R + CF[2]*G + CF[3]*B + CF[4]*R*R +
CF[5]*G*G + CF[6]*R*R*R + CF[7]*G*G*G.

RGB isn't a great colour space for colour matching work and you might have
better luck converting to some form of HSV (hue-saturation-intensity) for
this modelling. You can use IDL's COLOR_CONVERT for this. (Watch out for
wrapping hues. Shouldn't be a problem for red-to-yellow hues as you can
clip any hue above 330, say, to 0.) You might get away with just using
intensity and saturation in this dataset (with a stronger weight on
intensity), or perhaps intensity alone, as long as you know that you will
only query the data values of the coloured bits of imagery (which is an
obvious constraint whatever you do). If you can work with intensity alone
then it'll make the modelling much simpler.

I hope this helps
Cheers
Peter Mason

Subject: Re: color value interpolation from colorbar
Posted by Jeremy Bailin on Fri, 05 Dec 2008 14:36:52 GMT
View Forum Message <> Reply to Message

On Dec 4, 9:35 pm, "j.coe...@gmail.com" <j.coe...@gmail.com> wrote:
> Thanks everyone. Sorry for the delay, I had to generate the requested
> pics and create a Picasa web album. If there is a better way to post
> images to usenet forums, please let me know.
>
> Below is the Picasa link to the two images requested by Vince and
> Paolo, (1) an example frame grab of a scan, and (2) its colorbar RGB
> plotted against colorbar location:
>
> http://picasaweb.google.com/j.coenia/ColorInterpolation?auth key=H9iPr...
>
> To answer Jeremy's question, the colorbar length is 140 pixels or so

> (scaled here from 1 to 100 on the x axis, which is vertical in the
> scan). You can see from the plot that the colorbar sampling is
> "garbagy." There are two very bright artifacts, at approximately x=20
> and x=80. Such outliers can be tossed or smoothed out somehow I
> think. For simplicity, I just sampled the values down the vertical
> center of the colorbar, as the colorbar tends to bleed a little into
> the dark background near the edges (more errors).
>
> Jeremy's answer makes some sense to me. So is it possible to
> reasonably guess the color levels in that artery using the colorbar on
> the side of the scan? I know there is no scale on the colorbar --
> I've been instructed to assume linear gradient from 1 to 100.
> Radiologists and researchers use these colors; can the computer
> quantify them to extract more meaningful information?
>
> Thanks again.

I think that your colour bar is sampled well enough for the approach I suggested to work. You'll need to smooth out your R,G,B curves first, though - I would first use Peter's suggestion of taking the mean over a few columns within the colour bar, and then I'd pass it through a median filter to get rid of the artifacts and further smooth it.

I quite like Peter's approach, actually - assuming that the curves can be fit to a sufficiently low-order polynomial, which you'll have to check. It should be a lot faster, and is definitely more elegant! The approach I suggested should work pretty generically for any bizarre colour table, but yours looks it would be reasonably well-behaved once smoothed.

-Jeremy.

Subject: Re: color value interpolation from colorbar
Posted by pgrigis on Fri, 05 Dec 2008 15:17:32 GMT
View Forum Message <> Reply to Message

I would fit a*x^b for red and blue and c+dx for blue.

Ciao,
Paolo

Jeremy Bailin wrote:

>> images to usenet forums, please let me know.
>>

>> Below is the Picasa link to the two images requested by Vince and
>> Paolo, (1) an example frame grab of a scan, and (2) its colorbar RGB
>> plotted against colorbar location:
>>
>> http://picasaweb.google.com/j.coenia/ColorInterpolation?auth key=H9iPr...
>>
>> To answer Jeremy's question, the colorbar length is 140 pixels or so
>> (scaled here from 1 to 100 on the x axis, which is vertical in the

>> center of the colorbar, as the colorbar tends to bleed a little into
>> the dark background near the edges (more errors).
>>

>> reasonably guess the color levels in that artery using the colorbar on

>> I've been instructed to assume linear gradient from 1 to 100.
>> Radiologists and researchers use these colors; can the computer
>> quantify them to extract more meaningful information?
>>
>> Thanks again.
>
> I think that your colour bar is sampled well enough for the approach I
> suggested to work. You'll need to smooth out your R,G,B curves first,
> though - I would first use Peter's suggestion of taking the mean over
> a few columns within the colour bar, and then I'd pass it through a
> median filter to get rid of the artifacts and further smooth it.
>
> I quite like Peter's approach, actually - assuming that the curves can
> be fit to a sufficiently low-order polynomial, which you'll have to
> check. It should be a lot faster, and is definitely more elegant! The
> approach I suggested should work pretty generically for any bizarre
> colour table, but yours looks it would be reasonably well-behaved once
> smoothed.
>
> -Jeremy.

Subject: Re: color value interpolation from colorbar
Posted by j.coenia@gmail.com on Fri, 05 Dec 2008 15:50:42 GMT
View Forum Message <> Reply to Message

I fixed the RGB plot and uploaded it to Picasa:

http://picasaweb.google.com/j.coenia/ColorInterpolation?auth key=H9iPrIqxX1c#

I will look into how the images were made. The colors were automatically overlaid on the images by the scanning equipment. Maybe I will have to contact the manufacturers.

Someone has given me a brute force HSV color matching function. There is no curve fitting involved, but it seems promising. The text below is from the header notes:

; The input colors (sampled from the colorbar) are broken into blocks every n colors, and
; the extremes of H, S, and V are used to define a cube. All possible HSV-tuples
; within the cube are selected that correspond to possible colors in RGB space, which can
; optionally be reduced by a compression factor. The function returns a 3xn array of
; rgb triples corresponding to the INTERPOLATED colors in the reduced rgb colorspace
; (num of colors = (256/compression)^3).

I'll post the results from this function soon, and I'll try to implement Peter's more elegant approach (next week?).

Thanks again.

Subject: Re: color value interpolation from colorbar
Posted by Jeremy Bailin on Sat, 06 Dec 2008 12:44:27 GMT
View Forum Message <> Reply to Message

On Dec 5, 10:50 am, "j.coe...@gmail.com" <j.coe...@gmail.com> wrote:
> I fixed the RGB plot and uploaded it to Picasa:
>
> http://picasaweb.google.com/j.coenia/ColorInterpolation?auth key=H9iPr...
>
> I will look into how the images were made. The colors were
> automatically overlaid on the images by the scanning equipment. Maybe
> I will have to contact the manufacturers.
>
> Someone has given me a brute force HSV color matching function. There
> is no curve fitting involved, but it seems promising. The text below
> is from the header notes:
>
> ; The input colors (sampled from the colorbar) are broken into blocks
> every n colors, and
> ; the extremes of H, S, and V are used to define a cube. All
> possible HSV-tuples
> ; within the cube are selected that correspond to possible colors in

> RGB space, which can
> ; optionally be reduced by a compression factor. The function
> returns a 3xn array of
> ; rgb triples corresponding to the INTERPOLATED colors in the reduced
> rgb colorspace
> ; (num of colors = (256/compression)^3).
>
> I'll post the results from this function soon, and I'll try to
> implement Peter's more elegant approach (next week?).
>
> Thanks again.

Looking at the new version, I think the R and G curves are broken lines, not single curves, in which case I'm not sure how well the methods that try to fit them to single functions (either polynomials, like Peter said, or power laws like Paolo suggested) will work. Maybe you could split the image points by intensity and do the linear method using either the lower part or the upper part separately, but you might well get artifacts for colours around the break point.

The good news is, if they really are broken lines then you can make a huge simplification to my algorithm - instead of using the entire colour bar, you can reduce it to 3 points: the bottom point, the break point, and the top point.

-Jeremy.

Subject: Re: color value interpolation from colorbar
Posted by R.G. Stockwell on Sun, 07 Dec 2008 05:02:02 GMT
View Forum Message <> Reply to Message

> "Paolo" <pgrigis@gmail.com> wrote in message
> news:740da949-63fc-4bdc-89c1-3834bf7a5246@k19g2000yqg.google groups.com...
> I would fit a*x^b for red and blue and c+dx for blue.

> Ciao,
> Paolo

My two cents, red and green are piecewise linear, with the slope changing at index 40 (or thereabouts). That kind of thing is something you might see in a color table.

I'd fit red[0:40] to a line, red[40:*] to another line
I'd fit green[0:40] to a line, green[40:*] to another line
and blue to one line.

cheers,
bob

Subject: Re: color value interpolation from colorbar
Posted by j.coenia@gmail.com on Tue, 16 Dec 2008 03:49:04 GMT
View Forum Message <> Reply to Message

I finally posted the results from the HSV color interpolation function that was supplied to me. I used the colors it found to recolor the images from red to yellow, instead of the original black-red-yellow, which was harder to see. These colors were extruded according to their interpolated gradient values to produce an animation that I uploaded to Picasa:

http://picasaweb.google.com/j.coenia/ColorInterpolation?auth key=H9iPrIqxX1c#

I think now that this inefficient HSV interpolation function is not necessary. As Dr. Fanning says, if the images were generated correctly by the scanner, then originally the colorbar on them should contain all the image colors. There are more colors (errors) in this problem because these images are digitized frame grabs from a session recorded to videotape from an analog source, but the colors should still be close enough to extract something meaningful.

It might be preferable to just sample the colorbar; find all the colors in the image; calculate how close the image colors are to the colorbar colors (in HSV?); then assign corresponding values to the image colors if they are closer to the nearest colorbar color than some threshold. This is similar to what was originally proposed by Jeremy Bailin, although not as elegant as the later fitting methods described in this thread. There is even the IDLgrPalette::NearestColor method that will find the nearest color for you if supplied with both the sampled colorbar and the image color to match, although I don't know exactly how it finds the nearest color.

Thanks all.
