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.