Subject: Re: Significant Update of cgImage
Posted by Fabzou on Mon, 05 Dec 2011 08:55:36 GMT
View Forum Message <> Reply to Message

Sorry for the multiple post.

It is cgColor that has a bug, not cgContour. I just replaced line 589:

```
IF StrUpCase(theColor) EQ 'USERDEF'
```

by

```
IF (StrUpCase(theColor))[0] EQ 'USERDEF'
```

and all my tests seem to work now.

Fab

On 12/05/2011 09:46 AM, Fabzou wrote:
> Ah, and it seems to me that cgContour is broken in the last update:
>
> IDL> data = cgDemoData(18)
> IDL> cgContour, data, COLOR='grey'
>
>
> Traceback Report from CGCOLOR:
>
> % Expression must be a scalar or 1 element array in this context: <BYTE
> Array[6]>.
> % Execution halted at: CGCOLOR 589
> /home/fab/disk/IDLWorkspace/COYOTE/cgcolor.pro
> % CGCONTOUR 724 /home/fab/disk/IDLWorkspace/COYOTE/cgcontour.pro
> % $MAIN$
>
> On 12/05/2011 09:36 AM, Fabzou wrote:
>> Hi David,
>>
>> Nice updates. I think that most of the features (scaling, missing) have
>> been allready implemented upstream by people (like me!) who needed this
>> for long, but I may change some of my code
>>
>> The same applies for the automatic window XY ratio. In my case the most
>> interesting feature would be do be able to produce a display
>> automatically that takes in account the XY ratio of the "image + color
>> bar + colorbar tags" which are almost allways present when you display
>> any kind of data.

>>
>> Thanks a lot,
>>
>> Fab
>>
>>
>>
>> On 12/04/2011 06:58 PM, David Fanning wrote:
>>> Folks,
>>>
>>> I wanted to alert you to a couple of significant feature
>>> updates to cgImage.
>>>
>>> As I work on my map projection book, I have been using
>>> a lot of GeoTiff images. These images are easy to
>>> navigate and georegister, and I have developed software
>>> that can read a GeoTiff file and display the image with
>>> map annotations, etc. All of that works nicely.
>>>
>>> The problem is that many of these Tiff images are low
>>> contrast, have missing data values in them, etc. This
>>> is the problem that is addressed in this update of
>>> cgImage.
>>>
>>> I have modified cgImage so that if you are displaying
>>> a 2D image array (this does NOT apply to 24-bit color
>>> images!) you now have the ability to scale or stretch
>>> these images eight different ways. Basically, you now
>>> have the stretching capability of XStretch directly
>>> in cgImage. What this means is that not only can you
>>> do a straight linear scaling of the data before display
>>> (the purpose of the old SCALE keyword), but you can
>>> also do Log, Gamma, and Gaussian scaling, do
>>> histogram clipping in various ways, etc.
>>>
>>> For example, many of these images look best when a
>>> two percent histogram clipping is used (e.g. ClipScl).
>>> This is the same default clipping that ENVI uses.
>>> You can affect such a clip like this:
>>>
>>> IDL> cgImage, image2d, Stretch="CLIP"
>>>
>>> Possible values for the STRETCH keyword are: LINEAR,
>>> CLIP, GAMMA, LOG, ASINH, SQUAREROOT, EQUALIZATION,
>>> GAUSSIAN, and MODIS. Alternatively, you can use index
>>> numbers in place of these names. In other words, the
>>> command above can also be written like this:
>>>

>>> IDL> cgImage, image2d, Stretch=2
>>>
>>> The old SCALE keyword simple chooses a LINEAR stretch.
>>> Additional keywords are added that will set the parameters
>>> for the different stretches available.
>>>
>>> Another problem with many GeoTiff images is that they
>>> have missing data in them. So cgImage has also been modified
>>> with three additional keywords to allow you to handle this
>>> missing data appropriately. These keywords are:
>>>
>>> Missing_Value - Used to specify the missing data value in the image.
>>> Missing_Index - Specify the missing index in the output image.
>>> Missing_Color - Specify color of the missing data in output image.
>>>
>>> Suppose, for example, missing data is indicated by the value
>>> -32767 and you would like to display this missing data in a
>>> white color using color index 255 (the default missing color
>>> index). Then you could set up and display your image like this:
>>>
>>> IDL> cgLoadct, 33, NColors=254
>>> IDL> cgImage, image2d, Missing_Value=-32767, Missing_Color='white', $
>>> Stretch="LINEAR"
>>>
>>> The missing data values are now set to !Values.F_NAN before the
>>> scaling is done. The scaling is done into the values 0 to 254.
>>> Then the missing data indices are set equal to 255, the
>>> missing value index. The resulting image shows the missing
>>> values in the color you specify.
>>>
>>> I want to mention one other change. I often want to display
>>> these images in graphics windows having the same aspect ratio
>>> as the image itself. A new DISPLAY keyword to cgImage will
>>> create such a graphics window for me and display the image in
>>> that graphics window. If the DISPLAY keyword is used with
>>> the WINDOW keyword, and new cgWindow will be created. Otherwise
>>> the graphics window will be created with cgDisplay.
>>>
>>> IDL> cgImage, image2d, Missing_Value=-32767, Missing_Color='white', $
>>> Stretch="LINEAR", /Display, /Window
>>>
>>> If a cgWindow is opened in this way, the aspect ratio of the
>>> window is confined to the aspect ratio of the image as the
>>> window is resized.
>>>
>>> A number of other routines were slightly modified to work
>>> smoothly with new functionality in cgImage. This would
>>> be a good time to update your Coyote Library:

>>>
>>> http://www.idlcoyote.com/programs/zip_files/coyoteprograms.z ip
>>>
>>> You can read about the new keywords in the updated cgImage
>>> documentation:
>>>
>>> http://www.idlcoyote.com/idldoc/cg/cgimage.html
>>>
>>> Cheers,
>>>
>>> David
>>>
>>
>