Subject: Re: Coastal boundaries over sat data
Posted by Sylvain Carette on Wed, 30 Aug 2000 07:00:00 GMT
View Forum Message <> Reply to Message

<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
<tt>Interesting thread</tt><tt></tt>
<p><tt>There is a small application with source code (in Delphy - kind
of OO Pascal)</tt>

 HRPT
programs - from David Taylor, Edinburgh

<tt>It doesnt display continent outline but it can "rectify" the image
which might be the projection you are looking for. The code is well commented
and readable even for a non-pascal programmer (like me - but I'm used to
C so its similar).</tt><tt></tt>
<p><tt>It may or may not be pertinent for your need since it remap the
data - not a real projection, a rectification or an unwrapping of the data.</tt>

<tt>On my side, I need to use that kind of rectification applied along
the path of the satellite to build continuous "ribbon" of avhrr data. Maybe
this could be somewhat applied to your case; instead of applying a projection
on the image array, you proccess each line separately as is it were a whole
image and reset the projection for each line and appending all the lines
together.</tt><tt></tt>
<p><tt>If the only thing you want is to overlay the continent outline over
the "natural" data, I think I would consider the problem as applying a
warping to the outline derivated from the lat/lon values contained in
the data instead of using map projection.</tt><tt></tt>
<p><tt>Hope this could be of some help</tt><tt></tt>
<p><tt>Sylvain Carette</tt>

<tt>VRML designer-composer</tt><tt></tt>
<p><tt>Ben Marriage wrote:</tt>
<blockquote TYPE=CITE><tt>Daniel Peduzzi wrote:</tt>

<tt>></tt><tt></tt>
<p><tt>> Thanks...that is a handy program, and I've used it before in the
past.</tt>

<tt>> I'm not sure that it can be used for what I want to do, though,
since</tt>

<tt>> I don't want to remap the data...only display it in its *native*</tt>

<tt>> projection with coastal boundaries.</tt>

<tt>></tt>

<tt>> In other words, if I have 1000 scanlines of DMSP data (1465 elements</tt>

<tt>> wide), and accompanying 1465x1000 lat/lon arrays, I'd like to
display</tt>

<tt>> a 1465x1000 image overlaid with coastlines.</tt><tt></tt>
<p><tt>I did something like this to check if an AVHRR pixel was over land
or</tt>

<tt>not.</tt>

<tt>I'll post the code here in case you are interested. I had to create

an</tt>

<tt>image which consisted of 0s and 1s corresponding to sea and land.
I did</tt>

<tt>this from IDL using map_set and map_continents (filling it as color=1),</tt>

<tt>then tvrd() and saving into a format handy format (in this case,
idl</tt>

<tt>save format) You could try doing it without filling, just keeping
the</tt>

<tt>continent outline in a file. I then have to restore this file each
time</tt>

<tt>I need to check for land. This is fairly resolution dependent,
but works</tt>

<tt>OK for me (AVHRR data around Antarctica).</tt><tt></tt>
<p><tt>It's a rather quick and dirty method - but *it works for me*(TM)</tt><tt></tt>
<p><tt>Ben</tt>

<tt> ;==========
========== </tt>

<tt>function landmaskcheck, lats, longs</tt><tt></tt>
<p><tt>; land_mask.idl is an image which has a 0 over the sea and a 1 over</tt>

<tt>; land. This was produced from a 2048x2048 window and using the</tt>

<tt>; map_set and map_continents procedures to define areas of land
and sea.</tt>

<tt>; Then, using the convert_coord function we convert latitudes and</tt>

<tt>; longitudes to land mask subscripts to determine if that pixel
is over</tt>

<tt>; land.</tt><tt></tt>
<p><tt>sizeimg = size(lats)</tt><tt></tt>
<p><tt>; this file contains 0s and 1s corresponding to sea/land.</tt>

<tt>; restoring this file creates an IDL variable called MASK</tt><tt></tt>
<p><tt>restore,file='~/cloudcl/data/land_mask.idl'</tt><tt ></tt>
<p><tt>; open up a new window</tt><tt></tt>
<p><tt>oldwin   = !D.window</tt>

<tt>window,xs = 2048,ys=2048,/pix,/free</tt>

<tt>newwin   = !d.window</tt><tt></tt>
<p><tt>; setup the map reprojection used to create the land mask file initially</tt><tt></tt>
<p><tt> map_set,-90,0,0,/ster,/noborder,xmarg=0,ymarg=0,limit=[-30,-
90,-45,0,-80,90,-55,-135],/iso </tt><tt></tt>
<p><tt>; convert the input image longitudes and latitudes to device coordinates</tt>

<tt>in</tt>

<tt>; the new reprojection</tt><tt></tt>
<p><tt>sub = convert_coord(longs,lats,/data,/to_device)</tt><tt></tt>
<p><tt>; squish them to the right size</tt><tt></tt>
<p><tt>xsub = reform(sub[0,*],sizeimg[1],sizeimg[2])</tt>

<tt>ysub = reform(sub[1,*],sizeimg[1],sizeimg[2])</tt><tt></tt>
<p><tt>; this bit produces an image (same size as the input) which contains
a 1</tt>

<tt>over</tt>

<tt>; land and a 0 over the ocean</tt><tt></tt>

<p><tt>flag = mask[xsub,ysub]</tt><tt></tt>
<p><tt>wdelete,newwin</tt>

<tt>wset,oldwin</tt><tt></tt>
<p><tt>return,flag</tt><tt></tt>
<p><tt>end</tt>

<tt> ;==========
========== </tt></blockquote>
<tt></tt></html>
