Subject: Re: differences among map_proj_image, map_image and map_patch
Posted by Olivia on Tue, 19 Feb 2013 19:54:06 GMT
View Forum Message <> Reply to Message

Thanks David for your answer, and I am really appreciate it. Now, I am more clear about the differences among these routines.

I learned map projection mainly from book "AcrGIS 9 Understanding Map Projections." I began to doubt whether my understanding about map projection is right. If you can give us an example how to understand map projection will benefit many people.

My research is about the climate change of national parks. For example, what't the temperature change at the end of 21st century under different climate scenarios? Since the geographic coordinate is not area conserve, and if I do analysis on the geographic coordinate directly, there will be some biases towards high latitude. That's the reason I wanna project the original data to the Lambert Azimuthal Equal-Area projection, and I need to know the value of warped image.

I have found an easier way to solve my problem using ENVI. Using this software I can customize the output x/y pixel size, define the way to warp the image, and also know the warped value and its position on the image.

Cheers,

Fuyao

On Monday, February 18, 2013 7:55:21 PM UTC-6, David Fanning wrote:
> Olivia writes:
>
>
>
>> I wanna project air temperature (tmean: [7025 (lon), 3105 (lat)]) over the U.S. from geographic coordinate to Lambert Azimuthal Equal-Area coordinate, and need to know the x,y value of the new variable.
>
>>
>
>> Information of Lambert Azimuthal Equal-Area coordinate:
>
>> The central longitude and latitude is -100 and 50, respectively. Both false northing and false easting are 0. The ellipsoid is WGS 84.
>
>>
>
>> I didn't notice much difference among map_proj_image, map_image and map_patch according to the manual, so I used these three routines, and wanna compare them.
>

>>
>
>> I use tmean1, tmean2 and tmean3 to represent results from map_proj_image, map_image and map_patch, respectively.
>
>>
>
>> The spatial pattern of tmean1 and tmean3 are very similar, both of them showed the warped image correctly. Except, the dimension of tmean1 is [7025, 3105], which is the same with original tmean (I can understand this), and the dimension of tmean2 is [1171, 666].
>
>> tmean2 is totally wrong, and its dimension is [1, 680].
>
>>
>
>> Could someone help me to clarify the differences among these three routines, please?
>
>>
>
>> It seems all these three routines just warp the image, and won't give any information about the value of x, y. How can I know the x,y of projected image?
>
>
>
> OK, I am going to try to answer these questions now. :-)
>
>
>
> First, let me say your questions shows some evidence of confusion. This
>
> is normal and is the result of two different phenomenon. First, you are
>
> probably reading the IDL documentation. This is generally a Big Mistake.
>
> Second, map projections are inherently hard to understand. I once took
>
> at job at the National Snow and Ice Data Center specifically so I could
>
> learn to work with map projections. (This is where I learned reading the
>
> IDL documentation was a Big Mistake.) It took me about three years to
>
> finally come to grips with map projections, and I worked with them
>
> daily. Matt Savoie and I would get together about once a week and I
>
> would convince him that everything he knew about map projections was
>

> wrong, and he would do the same for me. Then we would swap places, until
>
> the next week, when we would convince each other to swap places again.
>
> By iteration, over three years time, we both got to the point that we
>
> sure the other person was out of his mind!
>
>
>
> Anyway, there are two different ways to set up map projections in IDL.
>
> You have used both incorrectly. :-)
>
>
>
> Map_Set is very old and is considered (generously, I think) a semi-
>
> professional way of handling map projections. The serious map
>
> professionals I worked with wouldn't consider using it, although I have
>
> come to think of it as useful when working with some map projections
>
> (although, God knows, NOT Mercator!) when you want a global map and are
>
> not concerned with matters of great precision. It works in lat/lon space
>
> (which is NOT where you want to be when working with map projected
>
> images) and is good for warping images to a map projection (something
>
> that sets many map projection expert's teeth on edge).
>
>
>
> Map_Proj_Init is based on the GCTP software that was state of the art
>
> 15-20 years ago, very professional in its time, and still generally
>
> useful today, although nowhere near the state of the art now. The
>
> Map_Proj** routines work in the much more sensible projected meter
>
> space, which matches map projected images extremely well, and will allow
>
> you to fit a map projection to your image, as God meant for it to be
>

> done. There is almost no support structure built into IDL for working
>
> with Map_Proj_Init. In fact, I'm pretty sure folks at ExelisVis don't
>
> really understand Map_Proj_Init, so strong is their bias for warping
>
> images to map projections, instead of the reverse.
>
>
>
> In the absence of of support for the Map_Proj** routines, I have built
>
> my own supporting routines for working with the Map_Proj** programs.
>
> These are available in the Coyote Library.
>
>
>
> The point is, both of these routines produce "map structures," but they
>
> are different and cannot be used interchangeably, as you are doing in
>
> your example code. You must pick one system or the other, and live with
>
> your choice.
>
>
>
> Map_Proj_Image works with the structure created with Map_Proj_Init, as
>
> you would expect. It does a much better job of "warping" an image from
>
> one map projection to another, because the warping is done on regular
>
> grids of map projected XY (sometimes called UV) coordinates. You can
>
> expect excellent results when warping in this manner.
>
>
>
> Map_Image is the warping routine for Map_Set and I think of it as a sort
>
> of "display" routine. It is not really interested in preserving image
>
> values. It's main purpose is to warp an image to a map projection in a
>
> close-enough-for-government-work sort of way.
>

>
>
> Map_Patch is the back-up shortstop for when Map_Image fails to do its
>
> job (usually when one of the poles is involved). It uses a slightly
>
> different (and slower) algorithm that is often more reliable. It is also
>
> the routine of choice if you have a lat/lon value for every pixel in the
>
> image. But, again, it is used mostly for "display", not science. (I've
>
> written an equivalent routine for Map_Proj_Init, named cgWarpToMap. It
>
> does low-resolution display things very well.)
>
>
>
> The data you have is extremely high resolution (too high to be shown on
>
> a graphics display, unless you have a wall-sized monitor). So, I am not
>
> absolutely sure it matters what routine you use to warp it, although I
>
> am always partial to Map_Proj_Image. I certainly wouldn't warp it and
>
> then inquire of the warped image what it's value is. I'd make a small,
>
> low-resolution image for display and I'd keep the original image around
>
> and use that to get actual values.
>
>
>
> So, I guess the answer to your question is really another question. What
>
> exactly are you planning to do with this image that you need to know its
>
> warped values?
>
>
>
> Cheers,
>
>
>
> David
>

>
>
> P.S. I'm not entirely sure this answer will satisfy you. But, as I said,
>
> you asked a very complicated question, and there is no simple answer. In
>
> the end, it matters what you are hoping to do with the information. Tell
>
> us that, and we may be able to advise you more sensibly.
>
>
>
> --
>
> David Fanning, Ph.D.
>
> Fanning Software Consulting, Inc.
>
> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>
> Sepore ma de ni thue. ("Perhaps thou speakest truth.")