Subject: Display and Navigate Image in IDL 8.2
Posted by David Fanning on Tue, 04 Sep 2012 20:53:29 GMT
View Forum Message <> Reply to Message

Folks,

Here is code that will obtain an image from Google as a PNG file
and put it into the variable "googleImage":

```
googleStr = "http://maps.googleapis.com/maps/api/staticmap?" + $
   "center=40.6000,-105.1000&zoom=12&size=600x600" + $
   "&maptype=terrain&sensor=false&format=png32"
netObject = Obj_New('IDLnetURL')
void = netObject -> Get(URL=googleStr, FILENAME="googleimg.png")
Obj_Destroy, netObject
googleImage = Read_Image('googleimg.png')
```

Here are the map details about this image.

Map Projection: "Mercator"
Map Ellipsoid: WGS-84
Center_Latitude: 40.6000
Center_Longitude: -105.1000
Map Limit: [-84.7500, -180.000, 84.7500, 180.000]
XRange: [-11711131.0, -11688226.0] (meters)
YRange: [4914254.0, 4937159.5] (meters)
Meters/pixel: 38.1757
Image Dimensions: [600,600]

I wish to display this image and have the cursor update properly
in map space (latitude and longitude) as I move the cursor over it.
I am required to do this with the IDL 8.2 function graphics commands.
I can get the image to display with this command.

```
obj = Image(googleImage, /BOX_AXES, $
      map_projection='mercator', ellipsoid='WGS84', $
      Center_Latitude=centerLat, Center_Longitude=centerLon, $
      LIMIT=limit, XRANGE=xrange, YRANGE=yrange, GRID_UNITS=1, $
      DIMENSIONS=[700,700], LOCATION=[50,50])
```

But, as you can see, no box axes and what map labels there are
on the image are completely wrong. :-(

Even more interesting, if I execute this exact same command
from within a widget program that obtains the Google image,
I get the properly sized image window to appear, but it is
completely and utterly blank! Nothing in it whatsoever!
(I've proved the image actually exists at this point in the

widget program by displaying it normally with cgImage.)

Any ideas on where to go from here? It doesn't appear to me that
any of my map projection information is being recognized. I
have NO idea why identical commands work from the IDL command line
but not from within a widget program.

Cheers,

David

P.S. Let's just say Function Graphics are *still* completely and utterly
baffling to me!

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by David Fanning on Fri, 07 Sep 2012 22:03:46 GMT
View Forum Message <> Reply to Message

Klemen writes:

```
>  How about like this:
>
>  obj = Image(googleImage, MAP_PROJECTION=projection, $
>            ELLIPSOID=ellipsoid, GRID_UNITS=1, $
>            XRANGE=xrange, YRANGE=yrange, $
>            IMAGE_DIMENSIONS=[max(xrange)-min(xrange), max(yrange)-min(yrange)], $
>            IMAGE_LOCATION=[min(xrange),min(yrange)], $
>            DIMENSIONS=[n, n], /BOX_AXES)
```

I am actually making progress on this now, using something very much
like this. I'm writing it up. Should be ready in an hour or so.
It only took me two full days of effort. A new speed record for
a function graphics plot!

Cheers,

David

P.S. I was hoping for a break on my maintenance bill for
teaching people how to use this...well, interesting piece
of software. Now, I'm just hoping for a T-shirt. Still

haven't heard from the boys at ExcelisVIS. Maybe they
will be surprised someone made it work, too. ;-)

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by David Fanning on Fri, 07 Sep 2012 23:05:06 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> I am actually making progress on this now, using something very much
> like this. I'm writing it up. Should be ready in an hour or so.
> It only took me two full days of effort. A new speed record for
> a function graphics plot!

Success at last! Praise God Almighty! Success at last!

Don't know why I'm channeling Martin Luther King, Jr,
except that it has been a LONG day and I'm ready for
a Friday afternoon beer! Have I mentioned how much I
love Function Graphics.

If you want to skip to the bottom line, you can run the
code you find here:

   http://www.idlcoyote.com/tip_examples/mapnogrid.pro

If you want to read all the gory details of how I figured
this out, you can read this article:

   http://www.idlcoyote.com/ng_tips/mapnogrid.php

If you just want to see the final result, here is the
code:

```
   ; Get the image.
   googleStr = "http://maps.googleapis.com/maps/api/staticmap?" + $
      "center=40.6,-105.1&zoom=12&size=600x600" + $
      "&maptype=terrain&sensor=false&format=png32"
   netObject = Obj_New('IDLnetURL')
   void = netObject -> Get(URL=googleStr, FILENAME="googleimg.png")
```

```
Obj_Destroy, netObject
googleImage = Read_Image('googleimg.png')


; What I am trying to reproduce in Function Graphics, displayed
; with Coyote Graphics.
centerLat = 40.6D
centerLon = -105.1D
scale = cgGoogle_MetersPerPixel(12)
map = Obj_New('cgMap', 'Mercator', ELLIPSOID='WGS 84', /OnImage)
uv = map -> Forward(centerLon, centerLat)
uv_xcenter = uv[0,0]
uv_ycenter = uv[1,0]
xrange = [uv_xcenter - (300*scale), uv_xcenter + (300*scale)]
yrange = [uv_ycenter - (300*scale), uv_ycenter + (300*scale)]
map -> SetProperty, XRANGE=xrange, YRANGE=yrange
cgDisplay, 700, 700, Title='Google Image with Coyote Graphics'
cgImage, googleImage, Margin=0.1
map -> Draw
cgMap_Grid, MAP=map, /BOX_AXES
cgPlotS, -105.1, 40.6, PSYM='filledstar', SYMSIZE=3.0, $
   MAP=map, COLOR='red'

; The code I used to do what I wanted to do. (Needs some
; of the code from above.)
ll = map -> Inverse(xrange, yrange)
limits = [ll[0,0], ll[1,0], ll[0,1], ll[1,1]]
xdim = Abs(xrange[1] - xrange[0])
ydim = Abs(yrange[1] - yrange[0])
xloc = xrange[0]
yloc = yrange[0]
obj = Image(googleImage, MAP_PROJECTION='mercator', $
   ELLIPSOID='WGS 84', GRID_UNITS=1, $
   XRANGE=xrange, YRANGE=yrange, LIMIT=limit, $
   DIMENSIONS=[700,700], POSITION=[0.1, 0.1, 0.9, 0.9], /BOX_AXES, $
   IMAGE_DIMENSIONS=[xdim,ydim], IMAGE_LOCATION=[xloc,yloc])
sym = Symbol(centerLon, centerLat, 'star', /DATA, $
    SYM_COLOR='red', SYM_SIZE=3, SYM_FILLED=1)
obj.mapprojection.mapgrid.BOX_AXES = 1
obj.mapprojection.mapgrid.BOX_THICK = 10
obj.mapprojection.mapgrid.LINESTYLE = 1
obj.mapprojection.mapgrid.GRID_LONGITUDE = 0.04
obj.mapprojection.mapgrid.GRID_LATITUDE = 0.03
obj.mapprojection.mapgrid.LABEL_POSITION = 0
obj.mapprojection.mapgrid.LONGITUDE_MIN=-105.18
obj.mapprojection.mapgrid.LATITUDE_MIN=40.54
```

There is, apparently, no way to get box axes on the plot,

but to get even this far in two days time is a major
achievment! I'm going for a beer.

I learned after I wrote the article that the LIMIT
keyword is not needed. (I wouldn't have thought so,
but I was throwing the kitchen sink at the problem!)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by lecacheux.alain on Sat, 08 Sep 2012 21:07:47 GMT
View Forum Message <> Reply to Message

Le samedi 8 septembre 2012 01:05:06 UTC+2, David Fanning a écrit :
> David Fanning writes:
>
>
>
>> I am actually making progress on this now, using something very much
>
>> like this. I'm writing it up. Should be ready in an hour or so.
>
>> It only took me two full days of effort. A new speed record for
>
>> a function graphics plot!
>
>
>
> Success at last! Praise God Almighty! Success at last!
>
>
>
> Don't know why I'm channeling Martin Luther King, Jr,
>
> except that it has been a LONG day and I'm ready for
>
> a Friday afternoon beer! Have I mentioned how much I
>

> love Function Graphics.
>
>
>
> If you want to skip to the bottom line, you can run the
>
> code you find here:
>
>
>
> http://www.idlcoyote.com/tip_examples/mapnogrid.pro
>
>
>
> If you want to read all the gory details of how I figured
>
> this out, you can read this article:
>
>
>
> http://www.idlcoyote.com/ng_tips/mapnogrid.php
>
>
>
> If you just want to see the final result, here is the
>
> code:
>
>
>
> ; Get the image.
>
> googleStr = "http://maps.googleapis.com/maps/api/staticmap?" + $
>
> "center=40.6,-105.1&zoom=12&size=600x600" + $
>
> "&maptype=terrain&sensor=false&format=png32"
>
> netObject = Obj_New('IDLnetURL')
>
> void = netObject -> Get(URL=googleStr, FILENAME="googleimg.png")
>
> Obj_Destroy, netObject
>
> googleImage = Read_Image('googleimg.png')
>
>
>

```
>
>
>    ; What I am trying to reproduce in Function Graphics, displayed
>
>    ; with Coyote Graphics.
>
>    centerLat = 40.6D
>
>    centerLon = -105.1D
>
>    scale = cgGoogle_MetersPerPixel(12)
>
>    map = Obj_New('cgMap', 'Mercator', ELLIPSOID='WGS 84', /OnImage)
>
>    uv = map -> Forward(centerLon, centerLat)
>
>    uv_xcenter = uv[0,0]
>
>    uv_ycenter = uv[1,0]
>
>    xrange = [uv_xcenter - (300*scale), uv_xcenter + (300*scale)]
>
>    yrange = [uv_ycenter - (300*scale), uv_ycenter + (300*scale)]
>
>    map -> SetProperty, XRANGE=xrange, YRANGE=yrange
>
>    cgDisplay, 700, 700, Title='Google Image with Coyote Graphics'
>
>    cgImage, googleImage, Margin=0.1
>
>    map -> Draw
>
>    cgMap_Grid, MAP=map, /BOX_AXES
>
>    cgPlotS, -105.1, 40.6, PSYM='filledstar', SYMSIZE=3.0, $
>
>       MAP=map, COLOR='red'
>
>
>
>    ; The code I used to do what I wanted to do. (Needs some
>
>    ; of the code from above.)
>
>    ll = map -> Inverse(xrange, yrange)
>
>    limits = [ll[0,0], ll[1,0], ll[0,1], ll[1,1]]
>
```

```
>    xdim = Abs(xrange[1] - xrange[0])
>
>    ydim = Abs(yrange[1] - yrange[0])
>
>    xloc = xrange[0]
>
>    yloc = yrange[0]
>
>    obj = Image(googleImage, MAP_PROJECTION='mercator', $
>
>      ELLIPSOID='WGS 84', GRID_UNITS=1, $
>
>      XRANGE=xrange, YRANGE=yrange, LIMIT=limit, $
>
>      DIMENSIONS=[700,700], POSITION=[0.1, 0.1, 0.9, 0.9], /BOX_AXES, $
>
>      IMAGE_DIMENSIONS=[xdim,ydim], IMAGE_LOCATION=[xloc,yloc])
>
>    sym = Symbol(centerLon, centerLat, 'star', /DATA, $
>
>        SYM_COLOR='red', SYM_SIZE=3, SYM_FILLED=1)
>
>    obj.mapprojection.mapgrid.BOX_AXES = 1
>
>    obj.mapprojection.mapgrid.BOX_THICK = 10
>
>    obj.mapprojection.mapgrid.LINESTYLE = 1
>
>    obj.mapprojection.mapgrid.GRID_LONGITUDE = 0.04
>
>    obj.mapprojection.mapgrid.GRID_LATITUDE = 0.03
>
>    obj.mapprojection.mapgrid.LABEL_POSITION = 0
>
>    obj.mapprojection.mapgrid.LONGITUDE_MIN=-105.18
>
>    obj.mapprojection.mapgrid.LATITUDE_MIN=40.54
>
>
>
> There is, apparently, no way to get box axes on the plot,
>
> but to get even this far in two days time is a major
>
> achievment! I'm going for a beer.
>
>
>
```

> I learned after I wrote the article that the LIMIT
>
> keyword is not needed. (I wouldn't have thought so,
>
> but I was throwing the kitchen sink at the problem!)
>
>
>
> Cheers,
>
>
>
> David
>
>
>
>
>
>
> --
>
> David Fanning, Ph.D.
>
> Fanning Software Consulting, Inc.
>
> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

David,

I continued to study your problem. Here is my present understanding.
I can see two very different steps in your problem, the second one, which deals with (NG) graphics usage, being not the hardest!

The image you get from the GoogleMaps API is already a projection, with definite horizontal and vertical linear scales in projected meter space. But if you know exactly what this projection is, you can indifferently use the meter space or the lat/lon space. Therefore one get a first step which consists in retrieving the right image parameters, then a second (graphical) step in which the image and the relevant map grid (a line plot with labels, etc...) have to be overlaid.

We start from the GoogleMaps request, which contains four parameters: center coordinates, zoom factor and image size in pixels.

```
;  googleStr = "http://maps.googleapis.com/maps/api/staticmap?" + $
;    "center=40.6000,-105.1000&zoom=12&size=600x600" + $
;    "&maptype=terrain&sensor=false&format=png32"
```

From GoogleMAPS API documentation, the projection is a simple conformal Mercator projection from the WGS84 ellipsoid with true scaling along the equator:

```
  clon = -105.1d0
  clat = 40.6d
  m = Map_Proj_Init('mercator', /GCTP, ELLIPSOID='WGS 84', $
    CENTER_LONGITUDE=clon, TRUE_SCALE_LATITUDE=0d)
```

The scale of the image, in meter/pixel, is given by the "google" zoom factor deduced from the lowest resolution image (zoom=0) which maps the entire Earth equator over 256 pixels. Other zoom factors form a decreasing series in successive powers of 2. Then:

```
  zoom = 12
  res = (2*!dpi*m.A)/256/2^zoom  ;m.A is the equatorial radius of the involved ellipsoid.
  print,res
;       38.218514   ;a slight difference with your value ?
```

Taking into account the requested image size (in pixels), the boundaries (four corners) of the image, both in meter space and lat/lon space, can be further computed:

```
  xy_ = [[-1,-1],[1,-1],[1,1],[-1,1]]*res*299.5
  cm = map_proj_forward(MAP=m, clon, clat)
  xy_ += cm # [1,1,1,1]
  xy = map_proj_inverse(MAP=m, xy_)
  print, xy
;      -105.20283       40.521578
;      -104.99717       40.521578
;      -104.99717       40.678329
;      -105.20283       40.678329
```

As this point, the right grid map can be built. (However, one thing is still missing: the altitude information. Indeed, GoogleMaps image display true coordinates (i.e. measured on the geoid), while above calculations are done on the ellipsoid. Since geoid and ellipsoid altitude differences remain within +/- 100m over the world, only the highest resolution images will be affected).

Now the second step: the graphics display.

First, the image, unlabelled. (The included margin is only needed because of overlaid grid annotations).

```
  img = IMAGE(googleimage, MARGIN=[0.05,0.15,0.05,0.05])
```

Second, the grid: just an overlay in the current window:

```
  mp = Map('Mercator', ELLIPSOID='WGS 84', $
    CENTER_LONGITUDE=clon, TRUE_SCALE_LATITUDE=0d, $
    LIMIT=[xy[1,0],xy[0,0],xy[1,-1],xy[0,1]], $
    GRID_LONGITUDE=1./20, GRID_LATITUDE=1./30, LINESTYLE=1, $
    LABEL_POSITION=0, $
    BOX_AXES=1, BOX_ANTIALIAS=1, $
```

POSITION=[0.05,0.15,0.95,0.95], /CURRENT)

The MAP object rendering could be easily enhanced by using child objects like mp.MAPGRID, mp.MAPGRID.LONGITUDES, etc...
Note that the mp.MAPPROJECTION property defines exactly the same projection than the above Map_Proj_Init does; in fact it uses it; there is certainly a way to avoid such a duplicate calculation.
I guess that the IMAGE function, throughout the MAP_PROJECTION keyword, is doing similar overlay, but I did not succeed in this way.

Quite simple, indeed ?

Cheers,

Alain

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by DavidF[1] on Mon, 10 Sep 2012 18:29:48 GMT
View Forum Message <> Reply to Message

I wrote last Friday:

> There is, apparently, no way to get box axes on the plot,
> but to get even this far in two days time is a major
> achievement! I'm going for a beer.

I learned today, with some help from the folks at ExcelisVIS,
how to add a box axis to the image.

The secret is to NOT use the LIMIT and BOX_AXES
keywords when you create the map projection, since
this results in a blank window. (I don't know why
I didn't think of this!)

Rather, you should set the LIMIT and BOX_AXES
properties on your map projection sometime after
you display the image and sometime before you
throw the goddamn computer out the window.

In other words, don't do this:

```
  obj = Image(googleImage, MAP_PROJECTION='mercator', $
      ELLIPSOID='WGS 84', GRID_UNITS=1, $
      XRANGE=xrange, YRANGE=yrange, $
      DIMENSIONS=[700,700], POSITION=[0.05, 0.05, 0.85, 0.85], $
      IMAGE_DIMENSIONS=[xdim,ydim], IMAGE_LOCATION=[xloc,yloc], $
      LIMIT=limit, /BOX_AXES)
```

Do this instead:

```
  obj = Image(googleImage, MAP_PROJECTION='mercator', $
      ELLIPSOID='WGS 84', GRID_UNITS=1, $
      XRANGE=xrange, YRANGE=yrange, $
      DIMENSIONS=[700,700], POSITION=[0.05, 0.05, 0.85, 0.85], $
      IMAGE_DIMENSIONS=[xdim,ydim], IMAGE_LOCATION=[xloc,yloc])
  obj.mapprojection.mapgrid.BOX_AXES=1
  obj.maprojection.LIMIT = limit
```

It's true, the box axes are not labeled correctly, but at least
they are there. :-)

But, now I see the real reason for the bad documentation. The
technical writer, understanding that reading the documentation
is not going to help at all (indeed, it seems designed to lead
you astray!) has decided to do as poor a job of writing as he
can get away with, causing you to more easily give up on the
documentation altogether. In this way, he leads you inevitably toward
the true path of learning: unfocused and random experimentation.
It is really pure genius, when you think about it!

Cheers,

David

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by DavidF[1] on Mon, 10 Sep 2012 18:38:07 GMT
View Forum Message <> Reply to Message

I wrote:

> The secret is to NOT use the LIMIT and BOX_AXES
> keywords when you create the map projection, since
> this results in a blank window.

By the way, setting the LIMIT of the map projection in an
Image function is a known bug that *always* causes the image
to disappear. I have the useless reference number that you
can't use to get additional information, if anyone needs it. ;-)

Cheers,

David

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by David Fanning on Mon, 10 Sep 2012 21:44:44 GMT
View Forum Message <> Reply to Message

I wrote earlier today:

> I learned today, with some help from the folks at ExcelisVIS,
> how to add a box axis to the image.
>
> The secret is to NOT use the LIMIT and BOX_AXES
> keywords when you create the map projection, since
> this results in a blank window. (I don't know why
> I didn't think of this!)
>
> Rather, you should set the LIMIT and BOX_AXES
> properties on your map projection sometime after
> you display the image and sometime before you
> throw the goddamn computer out the window.

As I was writing this up for my article on the topic I noticed that setting the LIMIT like this after the fact will blur the image substantially. I think this is because the image will be warped into the map projection space, which is the thing I am trying desperately to avoid. :-(

You can see this by comparing the Coyote Graphics version of this plot with the Function Graphics version in this article:

http://www.idlcoyote.com/ng_tips/mapnogrid.php

Or, you can run the code in this example and compare the two results:

http://www.idlcoyote.com/tip_examples/mapnogrid.pro

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by lecacheux.alain on Tue, 11 Sep 2012 08:03:18 GMT
View Forum Message <> Reply to Message

Le lundi 10 septembre 2012 23:44:42 UTC+2, David Fanning a écrit :
> I wrote earlier today:
>
>
>
>> I learned today, with some help from the folks at ExcelisVIS,
>
>> how to add a box axis to the image.
>
>>
>
>> The secret is to NOT use the LIMIT and BOX_AXES
>
>> keywords when you create the map projection, since
>
>> this results in a blank window. (I don't know why
>
>> I didn't think of this!)
>
>>
>
>> Rather, you should set the LIMIT and BOX_AXES
>
>> properties on your map projection sometime after
>
>> you display the image and sometime before you
>
>> throw the goddamn computer out the window.
>
>
>
> As I was writing this up for my article on the topic
>
> I noticed that setting the LIMIT like this after the
>
> fact will blur the image substantially. I think this
>
> is because the image will be warped into the map
>
> projection space, which is the thing I am trying
>
> desperately to avoid. :-(
>
>
>

> You can see this by comparing the Coyote Graphics
>
> version of this plot with the Function Graphics
>
> version in this article:
>
>
>
> http://www.idlcoyote.com/ng_tips/mapnogrid.php
>
>
>
> Or, you can run the code in this example and compare
>
> the two results:
>
>
>
> http://www.idlcoyote.com/tip_examples/mapnogrid.pro
>
>
>
> Cheers,
>
>
>
> David
>
>
>
>
>
>
>
>
>
>
> --
>
> David Fanning, Ph.D.
>
> Fanning Software Consulting, Inc.
>
> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

> As I was writing this up for my article on the topic
> I noticed that setting the LIMIT like this after the

> fact will blur the image substantially. I think this
> is because the image will be warped into the map
> projection space, which is the thing I am trying
> desperately to avoid. :-(

You should avoid any warping if you simply OVERLAY the image and the map grid calculated by using exactly the SAME projection as the one which was used when the (projected) image was computed.

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by David Fanning on Tue, 11 Sep 2012 14:09:09 GMT
View Forum Message <> Reply to Message

alx writes:

> You should avoid any warping if you simply OVERLAY the
> image and the map grid calculated by using exactly the
> SAME projection as the one which was used when the
> (projected) image was computed.

I'm not sure exactly what you are talking about now.
Here is the code I am using currently. When I substitute
OVERPLOT (I assume this is what you mean by OVERLAY)
for the CURRENT keyword, my image disappears and I see
only a map projection in my display window.

```
  img = Image(googleImage, POSITION=[0.1, 0.1, 0.9, 0.9], $
     DIMENSIONS=[700,700])
  mp = Map('mercator', ELLIPSOID='WGS 84', $
     CENTER_LONGITUDE=centerLon, $
     LIMIT=limit, /BOX_AXES, $
     POSITION=[0.1, 0.1, 0.9, 0.9], /CURRENT, $
     GRID_LONGITUDE=0.04, GRID_LATITUDE=0.03, LABEL_POSITION=1, $
     LONGITUDE_MIN=-105.18, LATITUDE_MIN=40.54, LINESTYLE=1)
   sym = Symbol(centerLon, centerLat, 'star', /DATA, $
     SYM_COLOR='red', SYM_SIZE=3, SYM_FILLED=1)
```

This *does* result in an unblurred image. But, I notice
another very strange thing. Even though I use IDENTICAL
values to set up the map projection this way, I get
DIFFERENT results!

What the hell!?

I've been working on this four days now! Every time I try to
do something simple in Function Graphics it is like this.
It is a friggin' nightmare!

You can run this program to see what I mean. You will
see the Coyote Graphics result, and two different
Function Graphics results, even though the map projection
has been set up in *exactly* the same way!

  http://www.idlcoyote.com/tip_examples/mapnogrid.pro

Even if I could get this to work, I don't believe I could
ever trust the results!!

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by David Fanning on Tue, 11 Sep 2012 14:24:01 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> This *does* result in an unblurred image. But, I notice
> another very strange thing. Even though I use IDENTICAL
> values to set up the map projection this way, I get
> DIFFERENT results!
>
> What the hell!?

OK, this appears to be due to the LIMIT not being
applied properly to the map projection when I set
the map projection up. If I reapply the LIMIT after
I have set the map projection up, it appears to
work properly:

```
mp = Map(...)
mp.LIMIT = limit
```

I think the bottom line is that the LIMIT keyword to
a map projection just plain doesn't work at all.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by lecacheux.alain on Tue, 11 Sep 2012 15:03:04 GMT
View Forum Message <> Reply to Message

Le mardi 11 septembre 2012 16:24:14 UTC+2, David Fanning a écrit :

Several things:

1) by OVERLAY, I mean that both the image and the grid should be drawn at the same position in the window (through CURRENT and POSITION keywords).

2) I found that keywords LONGITUDE/LATITUDE_MAX/MIN have same effect as LIMIT. You should use only one of them !

3) where your values "LONGITUDE_MIN=-105.18, LATITUDE_MIN=40.54" are coming from ? Please refer to my previous (saturday) post, for the recovery of the right (?) values.

4) I guess that your code should be:

```
img = IMAGE(googleimage, POSITION=[0.05,0.15,0.95,0.95], DIMENSIONS=[700,700])

mp = Map('Mercator', $
;set the overall projection
  ELLIPSOID='WGS 84', CENTER_LONGITUDE=clon, $
  TRUE_SCALE_LATITUDE=0d, $
;set the grid limits (you could use LIMIT too)
  LATITUDE_MIN=40.521578, LATITUDE_MAX=40.678329, $
  LONGITUDE_MIN=-105.20283, LONGITUDE_MAX=-104.99717, $
;set the grid appearance
  GRID_LONGITUDE=1./20, GRID_LATITUDE=1./30, LINESTYLE=1, $
  LABEL_POSITION=0, BOX_AXES=1, BOX_ANTIALIAS=1, $
;put the grid at the right place in the current window
  POSITION=[0.05,0.15,0.95,0.95], /CURRENT)
```

Cheers,
Alain.

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by David Fanning on Tue, 11 Sep 2012 15:28:25 GMT
View Forum Message <> Reply to Message

Alain writes:

> Several things:
>
> 1) by OVERLAY, I mean that both the image and the grid should be drawn at the same position in the window (through CURRENT and POSITION keywords).

Just curious, but what do you think they mean by OVERPLOT?
Clearly, it's not what I mean, but it must mean something. :-)

> 2) I found that keywords LONGITUDE/LATITUDE_MAX/MIN have same effect as LIMIT. You should use only one of them !

Well, this could be true, but if so it probably explains
why things are not working. As I read the documentation
(a futile activity, I know), the LONGITUDE/LATITUDE_MAX/MIN
keywords apply to the MapGrid object and locate the latitudes
and longitudes that should be drawn on the map. (And, this
is the way I am using them.) LIMIT should clearly apply to
the map projection space, NOT to the grid! I suppose you
could use the LIMIT to set the LONGITUDE/LATITUDE_MAX/MIN
values, but it would be wrong to do it the other way round.
(And the more I think about it, the more I think this could
be the source of all the errors I've been seeing.)

> 3) where your values "LONGITUDE_MIN=-105.18, LATITUDE_MIN=40.54"
> are coming from ? Please refer to my previous (saturday) post,
> for the recovery of the right (?) values.

Well, I'm not sure about "right (?)" values. I use them
so I can compare the results with the correct Coyote
Graphics display. I use them to locate the first drawn
grid line in the lat and lon directions. They certainly
seem to be doing the job for me.

> 4) I guess that your code should be:
>
> img = IMAGE(googleimage, POSITION=[0.05,0.15,0.95,0.95], DIMENSIONS=[700,700])
>
> mp = Map('Mercator', $
> ;set the overall projection
> ELLIPSOID='WGS 84', CENTER_LONGITUDE=clon, $
> TRUE_SCALE_LATITUDE=0d, $
> ;set the grid limits (you could use LIMIT too)
> LATITUDE_MIN=40.521578, LATITUDE_MAX=40.678329, $

> LONGITUDE_MIN=-105.20283, LONGITUDE_MAX=-104.99717, $
> ;set the grid appearance
> GRID_LONGITUDE=1./20, GRID_LATITUDE=1./30, LINESTYLE=1, $
> LABEL_POSITION=0, BOX_AXES=1, BOX_ANTIALIAS=1, $
> ;put the grid at the right place in the current window
> POSITION=[0.05,0.15,0.95,0.95], /CURRENT)

Except then I can't compare it to my other plot, which
I know from independent verification is correct. :-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by David Fanning on Tue, 11 Sep 2012 15:32:35 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> Except then I can't compare it to my other plot, which
> I know from independent verification is correct. :-)

To say I don't trust Function Graphics results would
be an understatement!

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by lecacheux.alain on Tue, 11 Sep 2012 16:12:10 GMT
View Forum Message <> Reply to Message

Le mardi 11 septembre 2012 17:28:39 UTC+2, David Fanning a écrit :
>
> Just curious, but what do you think they mean by OVERPLOT?
> Clearly, it's not what I mean, but it must mean something. :-)
>
OVERPLOT in NG is quite different from OPLOT in DG ! Basically, scales of first and second plots are BOTH adjusted to fit in the window. This can be quite surprising... Documentation does not help much.
To keep the first plot unchanged, you must do something like:
```
p = plot(...)
q = plot(..., /OVERPLOT, XRANGE=p.XRANGE, YRANGE=p.YRANGE)
```
In order to simply superimpose two plots (whatever their respective scaling are), CURRENT is enough (it plays nearly the same role as NOERASE in DG).

>> 3) where your values "LONGITUDE_MIN=-105.18, LATITUDE_MIN=40.54"
>> are coming from ? Please refer to my previous (saturday) post,
>> for the recovery of the right (?) values.
>
> Well, I'm not sure about "right (?)" values. I use them
> so I can compare the results with the correct Coyote
> Graphics display. I use them to locate the first drawn
> grid line in the lat and lon directions. They certainly
> seem to be doing the job for me.

I would prefer to get them from available image metadata. In the case of GoogleMaps, we can just speculate, but in case of other more documented satellite image (for which you know exactly the used projection and boundary values) you should be able to check everything precisely.

Cheers,
Alain.

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by David Fanning on Tue, 11 Sep 2012 16:25:44 GMT
View Forum Message <> Reply to Message

alx writes:

> I would prefer to get them from available image metadata.
> In the case of GoogleMaps, we can just speculate, but in
> case of other more documented satellite image (for which
> you know exactly the used projection and boundary values)
> you should be able to check everything precisely.

If I can get two weeks off of work, I'll try this with
a GeoTiff file. ;-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by chris_torrence@NOSPAM on Wed, 12 Sep 2012 21:13:32 GMT
View Forum Message <> Reply to Message

Hi David,

Does this help?

```
  googleImage = Read_Image('googleimg.png')

  XRange = [-11711131d, -11688226d] ; (meters)
  YRange = [4914254d, 4937159.5d] ; (meters)

  im = IMAGE(googleImage, /BOX_AXES, GRID_UNITS='meters', $
    IMAGE_LOCATION=[xrange[0],yrange[0]], $
    IMAGE_DIMENSIONS=[xrange[1]-xrange[0],yrange[1]-yrange[0]], $
    MAP_PROJECTION='Mercator', ELLIPSOID='WGS84')
  m = im.mapprojection
  lonlat = m.MapInverse(im.xrange, im.yrange)
  m.limit = [lonlat[[1,0,3,2]]]
```

At least with IDL 8.2.1, I'm getting an image in "meters", with a box grid. When I click on the image, and then move the mouse, it reports lat/lon position.

-Chris
ExelisVIS

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by David Fanning on Wed, 12 Sep 2012 21:23:51 GMT
View Forum Message <> Reply to Message

Chris Torrence writes:

> Does this help?
>
> googleImage = Read_Image('googleimg.png')

>
>   XRange = [-11711131d, -11688226d] ; (meters)
>   YRange = [4914254d, 4937159.5d] ; (meters)
>
>   im = IMAGE(googleImage, /BOX_AXES, GRID_UNITS='meters', $
>   IMAGE_LOCATION=[xrange[0],yrange[0]], $
>   IMAGE_DIMENSIONS=[xrange[1]-xrange[0],yrange[1]-yrange[0]], $
>   MAP_PROJECTION='Mercator', ELLIPSOID='WGS84')
>   m = im.mapprojection
>   lonlat = m.MapInverse(im.xrange, im.yrange)
>   m.limit = [lonlat[[1,0,3,2]]]
>
> At least with IDL 8.2.1, I'm getting an image in "meters", with a box grid. When I click on the image, and then move the mouse, it reports lat/lon position.

Well, the last step has the effect of blurring the image, I think
because it causes the image to be warped into the map projection
space. To avoid blurring, you have to separate the display of
the image from the map projection. Basically, you want to lay
the map projection down "on top" of the image.

See the end of this article for a further explanation:

  http://www.idlcoyote.com/ng_tips/mapnogrid.php

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by lecacheux.alain on Thu, 13 Sep 2012 07:20:39 GMT
View Forum Message <> Reply to Message

Le mercredi 12 septembre 2012 23:24:05 UTC+2, David Fanning a écrit :
> Chris Torrence writes:
>
>
>
>> Does this help?
>

>>
>
>>   googleImage = Read_Image('googleimg.png')
>
>>
>
>>   XRange = [-11711131d, -11688226d] ; (meters)
>
>>   YRange = [4914254d, 4937159.5d] ; (meters)
>
>>
>
>>   im = IMAGE(googleImage, /BOX_AXES, GRID_UNITS='meters', $
>
>>   IMAGE_LOCATION=[xrange[0],yrange[0]], $
>
>>   IMAGE_DIMENSIONS=[xrange[1]-xrange[0],yrange[1]-yrange[0]], $
>
>>   MAP_PROJECTION='Mercator', ELLIPSOID='WGS84')
>
>>   m = im.mapprojection
>
>>   lonlat = m.MapInverse(im.xrange, im.yrange)
>
>>   m.limit = [lonlat[[1,0,3,2]]]
>
>>
>
>> At least with IDL 8.2.1, I'm getting an image in "meters", with a box grid. When I click on the image, and then move the mouse, it reports lat/lon position.
>
>
>
> Well, the last step has the effect of blurring the image, I think
>
> because it causes the image to be warped into the map projection
>
> space. To avoid blurring, you have to separate the display of
>
> the image from the map projection. Basically, you want to lay
>
> the map projection down "on top" of the image.
>
>
>
> See the end of this article for a further explanation:
>
>

>
> http://www.idlcoyote.com/ng_tips/mapnogrid.php
>
>
>
> Cheers,
>
>
>
> David
>
>
>
>
>
>
> --
>
> David Fanning, Ph.D.
>
> Fanning Software Consulting, Inc.
>
> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

For those who might be interested, my equation:
res = ((m.a + 135)*2*!dpi)/256/2^zoom

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by David Fanning on Thu, 13 Sep 2012 13:15:00 GMT
View Forum Message <> Reply to Message

alx writes:

> For those who might be interested, my equation:
> res = ((m.a + 135)*2*!dpi)/256/2^zoom

135!? You are going to have to explain that. :-)

Cheers,

David

--
David Fanning, Ph.D.

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by lecacheux.alain on Thu, 13 Sep 2012 14:23:07 GMT
View Forum Message <> Reply to Message

Le jeudi 13 septembre 2012 15:15:14 UTC+2, David Fanning a écrit :
> alx writes:
>
>
>
>> For those who might be interested, my equation:
>
>> res = ((m.a + 135)*2*!dpi)/256/2^zoom
>
>
>
> 135!? You are going to have to explain that. :-)
>
>
>
> Cheers,
>
>
>
> David
>
>
>
>
>
>
>
>
> --
>
> David Fanning, Ph.D.
>
> Fanning Software Consulting, Inc.
>
> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

My previous message should have been supressed and not published !
Anyhow, what I intended to propose is the formula :

res = ((m.a + altitude)*2*!dpi)/256/2^zoom
in which 'altitude' is the altitude difference between the geoid and the ellipsoid at the center of the map. This would makes sense. Unfortunately, since its value is not equal to the 'true' altitude (that counted above sea level and which is usually well known: 135m in the particular example I was checking!), this cannot help us much.
There is no other way, I guess, than additionnaly using relevant geoid reference data.
Sorry for the parasitic post...
alx.
