Subject: Re: Display and Navigate Image in IDL 8.2
Posted by David Fanning on Tue, 04 Sep 2012 21:12:20 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> 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. :-(

Oh, sorry. I *can't* get it to display with that command, if
I actually define the variables I am using. I confused myself
when I moved this out of the widget program and over to the
command line.

```
centerlat = 40.6
centerlon = -105.1
limit =  [-84.7500, -180.000, 84.7500, 180.000]
xrange =  [-11711131.0, -11688226.0]
yrange =  [4914254.0, 4937159.5]
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])
```

So, window opens, but completely blank. Any ideas?

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

Folks,

I have learned a little more about this problem this
morning. It's all complicated when nothing shows up
in your graphics window! ;-)

Let me go through it again.

Here is the code that obtains a map projected image:

```
  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 details about the image (note Google uses
a center latitude and longitude of 0 to calculate the
map ranges):

```
  projection = "mercator"
  ellipsoid = "WGS84"
  centerLat = 0.0
  centerLon = 0.0
  limit =  [-84.7500, -180.000, 84.7500, 180.000]
  xrange = [-11711131.0D, -11688226.0D]
  yrange = [4914254.0D, 4937159.5D]
```

Here is the code that gives me a blank window and which I am
trying to fix:

```
  obj = Image(googleImage, MAP_PROJECTION=projection, $
       ELLIPSOID=ellipsoid, GRID_UNITS=1, $
       CENTER_LONGITUDE=centerLon, CENTER_LATITUDE=centerLat, $
       LIMIT=limit, XRANGE=xrange, YRANGE=xrange, $
       DIMENSIONS=[700,700], LOCATION=[50,50], /BOX_AXES)
```

Here is Coyote Graphics code that results in what I expect to see:

```
  cgDisplay, 700, 700
  map = obj_new('cgMap', projection, ELLIPSOID=ellipsoid, $
     CENTER_LONGITUDE=centerLon, CENTER_LATITUDE=centerLat, $
     LIMIT=limit, XRANGE=xrange, YRANGE=yrange)
  cgImage, googleImage, Margin=1.0, OPosition=pos
```

```
  map -> SetProperty, POSITION=pos
  cgMap_Grid, /Box_Axes, MAP=map
```

And here is a PNG file of what I am trying to see in Function Graphics:

  http://www.idlcoyote.com/misc/googlemap.png

I've been working on this for hours, so any help at all gratefully accepted. :-)

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 DavidF[1] on Wed, 05 Sep 2012 15:53:50 GMT
View Forum Message <> Reply to Message

```
>  Here is the code that gives me a blank window and which I am
>  trying to fix:
>
>     obj = Image(googleImage, MAP_PROJECTION=projection, $
>           ELLIPSOID=ellipsoid, GRID_UNITS=1, $
>           CENTER_LONGITUDE=centerLon, CENTER_LATITUDE=centerLat, $
>           LIMIT=limit, XRANGE=xrange, YRANGE=xrange, $
>           DIMENSIONS=[700,700], LOCATION=[50,50], /BOX_AXES)
```

I have now tried this, which works no better. :-(

```
  s = Size(googleImage, /DIMENSIONS)
  xscale = Abs(xrange[1] - xrange[0]) / s[1]
  yscale = Abs(yrange[1] - yrange[0]) / s[2]
  x = (Findgen(s[1])*xscale) + xrange[0] + (xscale/2)
  y = (Findgen(s[2])*yscale) + yrange[0] + (yscale/2)
  obj = Image(googleImage, x, y, MAP_PROJECTION=projection, $
       ELLIPSOID=ellipsoid, GRID_UNITS=1, $
       CENTER_LONGITUDE=centerLon, CENTER_LATITUDE=centerLat, $
       LIMIT=limit, XRANGE=xrange, YRANGE=xrange, $
       DIMENSIONS=[700,700], LOCATION=[50,50], /BOX_AXES)
```

Cheers,

David

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by David Fanning on Thu, 06 Sep 2012 14:09:06 GMT
View Forum Message <> Reply to Message

Coyote writes:

```
>
>>  Here is the code that gives me a blank window and which I am
>>  trying to fix:
>>
>>     obj = Image(googleImage, MAP_PROJECTION=projection, $
>>           ELLIPSOID=ellipsoid, GRID_UNITS=1, $
>>           CENTER_LONGITUDE=centerLon, CENTER_LATITUDE=centerLat, $
>>           LIMIT=limit, XRANGE=xrange, YRANGE=xrange, $
>>           DIMENSIONS=[700,700], LOCATION=[50,50], /BOX_AXES)
>
>
>  I have now tried this, which works no better. :-(
>
>     s = Size(googleImage, /DIMENSIONS)
>     xscale = Abs(xrange[1] - xrange[0]) / s[1]
>     yscale = Abs(yrange[1] - yrange[0]) / s[2]
>     x = (Findgen(s[1])*xscale) + xrange[0] + (xscale/2)
>     y = (Findgen(s[2])*yscale) + yrange[0] + (yscale/2)
>     obj = Image(googleImage, x, y, MAP_PROJECTION=projection, $
>           ELLIPSOID=ellipsoid, GRID_UNITS=1, $
>           CENTER_LONGITUDE=centerLon, CENTER_LATITUDE=centerLat, $
>           LIMIT=limit, XRANGE=xrange, YRANGE=xrange, $
>           DIMENSIONS=[700,700], LOCATION=[50,50], /BOX_AXES)
>
```

Folks! Seriously!? No one has an answer?

The good news is that you are not alone. I've posed the same problem to the good folks in IDL Technical Support and I haven't received an answer from them, either.

This is not really a hard problem. I have a georegistered image. I want to set up a map coordinate system that reflects that so I can draw on top of the image, discover the cursor location in map units, etc. It is done EVERY day in the real world.

Is is really possible that this can't be done in IDL using the

function graphics system?

I'm getting tired of writing "IDL is unusable" articles, but
if experienced people (one might say IDL experts) can't figure
out how to do the very simplest things with your software, there
is something seriously wrong. I've been blaming the software and
documentation, but maybe the users are just not bright enough
to use such sophisticated tools. (Although how long-time users
because idiots all at once is a mystery I don't yet have an
explanation for.)

Anyway, I solder on, looking for solutions anywhere I can find them.
Let me know if you come up with something!

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, 06 Sep 2012 16:48:50 GMT
View Forum Message <> Reply to Message

Le jeudi 6 septembre 2012 16:09:05 UTC+2, David Fanning a écrit :
> Coyote writes:
>
>
>
>>
>
>>> Here is the code that gives me a blank window and which I am
>
>>> trying to fix:
>
>>>
>
>>> obj = Image(googleImage, MAP_PROJECTION=projection, $
>
>>> ELLIPSOID=ellipsoid, GRID_UNITS=1, $
>
>>> CENTER_LONGITUDE=centerLon, CENTER_LATITUDE=centerLat, $
>

>>>   LIMIT=limit, XRANGE=xrange, YRANGE=xrange, $
>
>>>   DIMENSIONS=[700,700], LOCATION=[50,50], /BOX_AXES)
>
>>
>
>>
>
>> I have now tried this, which works no better. :-(
>
>>
>
>>   s = Size(googleImage, /DIMENSIONS)
>
>>   xscale = Abs(xrange[1] - xrange[0]) / s[1]
>
>>   yscale = Abs(yrange[1] - yrange[0]) / s[2]
>
>>   x = (Findgen(s[1])*xscale) + xrange[0] + (xscale/2)
>
>>   y = (Findgen(s[2])*yscale) + yrange[0] + (yscale/2)
>
>>   obj = Image(googleImage, x, y, MAP_PROJECTION=projection, $
>
>>   ELLIPSOID=ellipsoid, GRID_UNITS=1, $
>
>>   CENTER_LONGITUDE=centerLon, CENTER_LATITUDE=centerLat, $
>
>>   LIMIT=limit, XRANGE=xrange, YRANGE=xrange, $
>
>>   DIMENSIONS=[700,700], LOCATION=[50,50], /BOX_AXES)
>
>>
>
>
>
> Folks! Seriously!? No one has an answer?
>
>
>
> The good news is that you are not alone. I've posed the same
>
> problem to the good folks in IDL Technical Support and I haven't
>
> received an answer from them, either.
>
>
>

> This is not really a hard problem. I have a georegistered image.
>
> I want to set up a map coordinate system that reflects that so
>
> I can draw on top of the image, discover the cursor location in
>
> map units, etc. It is done EVERY day in the real world.
>
>
>
> Is is really possible that this can't be done in IDL using the
>
> function graphics system?
>
>
>
> I'm getting tired of writing "IDL is unusable" articles, but
>
> if experienced people (one might say IDL experts) can't figure
>
> out how to do the very simplest things with your software, there
>
> is something seriously wrong. I've been blaming the software and
>
> documentation, but maybe the users are just not bright enough
>
> to use such sophisticated tools. (Although how long-time users
>
> because idiots all at once is a mystery I don't yet have an
>
> explanation for.)
>
>
>
> Anyway, I solder on, looking for solutions anywhere I can find them.
>
> Let me know if you come up with something!
>
>
>
> 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.")

Looking at your map display problem, I got the following issue:

centerlat = 40.6000d
centerlon = -105.1000d
xrange = [-11711131.0d, -11688226.0d]
yrange = [4914254.0d, 4937159.5d]
map = map_proj_init('Mercator', ELLIPSOID='WGS 84', CENTER_LONGITUDE=0,
CENTER_LATITUDE=0)
print, map_proj_inverse(mean(xrange), mean(yrange), MAP_STRUCTURE=map)
; -105.10000 40.410029

I should have found 40.6° for the latitude, is'nt it ?

print, map_proj_inverse(xrange, yrange, MAP_STRUCTURE=map)
; -105.20288 40.331647
; -104.99712 40.488320

Same thing regarding the latitude range boundaries which seem sligthly different
from those in your PNG example (40.525 and 40.675, by eye).

Cheers,
alain.

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by DavidF[1] on Thu, 06 Sep 2012 17:24:10 GMT
View Forum Message <> Reply to Message

Alain writes:

> Looking at your map display problem, I got the following issue:
>
> centerlat = 40.6000d
> centerlon = -105.1000d
> xrange = [-11711131.0d, -11688226.0d]
> yrange = [4914254.0d, 4937159.5d]
> map = map_proj_init('Mercator', ELLIPSOID='WGS 84', CENTER_LONGITUDE=0,
CENTER_LATITUDE=0)

> print, map_proj_inverse(mean(xrange), mean(yrange), MAP_STRUCTURE=map)
> ; -105.10000 40.410029
>
> I should have found 40.6° for the latitude, is'nt it ?
>
> print, map_proj_inverse(xrange, yrange, MAP_STRUCTURE=map)
>
> ; -105.20288 40.331647
> ; -104.99712 40.488320
>
> Same thing regarding the latitude range boundaries which seem sligthly different
>
> from those in your PNG example (40.525 and 40.675, by eye).

The numbers could well be off a little bit. Google doesn't (I suppose deliberately) supply these images in GeoTiff format, so I had to determine the pixel per meter value from other published data. It involved some detective work and assumptions (see cgGoogle_MetersPerPixel). Using the numbers I used, however, my markers lined up exactly with Google's own markers, so I'm guessing the numbers are close enough for my use.

But, all this is beside the point. I really want to know how to navigate and display this image using function graphics routines. I can use whatever projected meter numbers I like, and I *still* can't see the image in a display window. If I could, maybe I could work on the details of whether my map projection numbers are totally accurate. ;-)

Cheers,

David

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by DavidF[1] on Thu, 06 Sep 2012 18:36:52 GMT
View Forum Message <> Reply to Message

Here is where the projected meter numbers come from:

```
limit =  [-84.7500, -180.000, 84.7500, 180.000]
mapStruct = Map_Proj_Init('mercator', /gctp, ellipsoid='wgs 84', limit=limit)
cm = map_proj_forward(-105.100, 40.600, map=mapStruct)
xcenter = cm[0,0]
ycenter = cm[1,0]
Print, 'XCenter: ', xcenter
Print, 'YCenter: ', ycenter
metersPixel = cgGoogle_MetersPerPixel(12)
Print, 'meters per pixels: ', metersPixel
```

```
  xrange = [xcenter - (300*metersPixel), xcenter+(300*meterspixel)]
  yrange = [ycenter - (300*metersPixel), ycenter+(300*meterspixel)]
  Print, 'X Range: ', xrange
  print, 'Y Range: ', yrange
```

So, I have tried passing X and Y vectors like this:

```
  s = Size(googleImage, /DIMENSIONS)
  xscale = Abs(xrange[1] - xrange[0]) / s[1]
  yscale = Abs(yrange[1] - yrange[0]) / s[2]
  Print, 'X Scale: ', xscale
  Print, 'Y Scale: ', yscale
  x = (Findgen(s[1])*xscale) + xrange[0] + (xscale/2)
  y = (Findgen(s[2])*yscale) + yrange[0] + (yscale/2)
  obj = Image(googleImage, x, y, MAP_PROJECTION=projection, $
        ELLIPSOID=ellipsoid, GRID_UNITS=1, $
        CENTER_LONGITUDE=-105.1, CENTER_LATITUDE=40.60, $
        LIMIT=limit, DIMENSIONS=[700,700], LOCATION=[50,50], /BOX_AXES)
```

This produces a map display, but no image.

Or, I have tried passing the X and Y ranges like this:

```
  obj = Image(googleImage, MAP_PROJECTION=projection, $
        ELLIPSOID=ellipsoid, GRID_UNITS=1, $
        CENTER_LONGITUDE=-105.1, CENTER_LATITUDE=40.60, $
        XRANGE=xrange, YRANGE=yrange, $
        LIMIT=limit, DIMENSIONS=[700,700], LOCATION=[50,50], /BOX_AXES)
```

This produces a blank window with nothing in it.

I have even tried using the IMAGE_DIMENSIONS keyword like this:

```
  xdim = Abs(xrange[1] - xrange[0])
  ydim = Abs(yrange[1] - yrange[0])
  Print, xdim, ydim
  obj = Image(googleImage, MAP_PROJECTION=projection, $
        ELLIPSOID=ellipsoid, GRID_UNITS=1, $
        CENTER_LONGITUDE=-105.1, CENTER_LATITUDE=40.60, $
        XRANGE=xrange, YRANGE=yrange, $
        LIMIT=limit, DIMENSIONS=[700,700], LOCATION=[50,50], $
        IMAGE_DIMENSIONS=[xdim, ydim], /BOX_AXES)
```

That also results in a blank window.

And, I've tried numerous variations besides these, too. I'm
really at a complete loss.

Cheers,

David

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by Klemen on Fri, 07 Sep 2012 09:40:10 GMT
View Forum Message <> Reply to Message

David, I have jsut upgraded to IDL 8.2 so I can test it...

The code is, I think, ok (I haven't so much experience with Image). The only problem is your definition of you georeference. I think that your ranges are defined for a usual Mercator projection with origin in 0N, 0E. But you have moved your origin to 105.1W, 40.6N. This point should have in map coordinates values 0, 0. This means that below defined range is false:

XRange: [-11711131.0, -11688226.0] (meters)
YRange: [4914254.0, 4937159.5] (meters)

You could redefine your range as:

```
n = 600
res = 38.1757
x = findgen(n)* res - n*res*0.5
y = reverse(x)
xrange = [min(x), max(x)]
yrange = [min(y), max(y)]
```

Well, then I get at least something, I am not sure if this is exactlly what you are looking for, but right mouse click gives me at least the proper longitude when I scroll over the image.

Cheers, Klemen

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by lecacheux.alain on Fri, 07 Sep 2012 12:47:30 GMT
View Forum Message <> Reply to Message

Le vendredi 7 septembre 2012 11:40:10 UTC+2, Klemen a écrit :
> David, I have jsut upgraded to IDL 8.2 so I can test it...
>
>
>
> The code is, I think, ok (I haven't so much experience with Image). The only problem is your definition of you georeference. I think that your ranges are defined for a usual Mercator projection with origin in 0N, 0E. But you have moved your origin to 105.1W, 40.6N. This point should have in map coordinates values 0, 0. This means that below defined range is false:
>

>
>
> XRange: [-11711131.0, -11688226.0] (meters)
>
> YRange: [4914254.0, 4937159.5] (meters)
>
>
>
> You could redefine your range as:
>
> n = 600
>
> res = 38.1757
>
> x = findgen(n)* res - n*res*0.5
>
> y = reverse(x)
>
> xrange = [min(x), max(x)]
>
> yrange = [min(y), max(y)]
>
>
>
> Well, then I get at least something, I am not sure if this is exactlly what you are looking for, but right mouse click gives me at least the proper longitude when I scroll over the image.
>
>
>
> Cheers, Klemen

Hi David,

Here is my solution:

1st step: define the size of the image in decimal degrees, by using your "magic" resolution number. I guess that it might be retrieved by another way.

```
centerlat = 40.6000d
centerlon = -105.1000d
res = 38.1757
map = Map_Proj_Init('mercator', /gctp, ELLIPSOID='wgs 84')
cm = map_proj_forward(centerlon, centerlat, MAP=map)
xrange = cm[0] + [-300,300]*res
yrange = cm[1] + [-300,300]*res
xr = map_proj_inverse(xrange, yrange, MAP=map)
print,xr
;    -105.20288       40.521535
;    -104.99712       40.678372
```

As you can see, numbers are slightly different from yours (in latitude).

2nd step: draw the image and add the grid map
```
obj = IMAGE(googleimage, $
  GRID_UNITS=2,
MARGIN=[0.05,0.15,0.05,0.05], $
  IMAGE_LOCATION=xr[0:1,0],$
  IMAGE_DIMENSIONS=[xr[0,1]-xr[0,0],xr[1,1]-xr[1,0]],$
  MAP_PROJECTION='Mercator')
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
```

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by lecacheux.alain on Fri, 07 Sep 2012 12:58:18 GMT
View Forum Message <> Reply to Message

Le vendredi 7 septembre 2012 11:40:10 UTC+2, Klemen a écrit :
> David, I have jsut upgraded to IDL 8.2 so I can test it...
>
>
>
> The code is, I think, ok (I haven't so much experience with Image). The only problem is your definition of you georeference. I think that your ranges are defined for a usual Mercator projection with origin in 0N, 0E. But you have moved your origin to 105.1W, 40.6N. This point should have in map coordinates values 0, 0. This means that below defined range is false:
>
>
>
> XRange: [-11711131.0, -11688226.0] (meters)
>
> YRange: [4914254.0, 4937159.5] (meters)
>
>
>
> You could redefine your range as:
>
> n = 600
>
> res = 38.1757
>
> x = findgen(n)* res - n*res*0.5
>
> y = reverse(x)

>
> xrange = [min(x), max(x)]
>
> yrange = [min(y), max(y)]
>
>
>
> Well, then I get at least something, I am not sure if this is exactlly what you are looking for, but right mouse click gives me at least the proper longitude when I scroll over the image.
>
>
>
> Cheers, Klemen

I unfortunately pressed the button too fast, so that my previous message was sent before that I could finish it.
The trick (that I have found in spite of the cryptic Exelis documentation, but better solutions are likely to exist) was to consider that any NG building consists in a tree of graphics objects. Then, when plotting your image, the use of the MAP_PROJECTION keyword implies that such a corresponding MAP object is added to the IMAGE one, then that a MAPGRID object should be also available. The latter can be retrieved through the MAPPROJECTION property. Please consider what a simple underline can change to your life !
Now I did not succeed with the BOX_AXES keyword. I leave that for your exercizing ...
Alain.

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by David Fanning on Fri, 07 Sep 2012 13:15:37 GMT
View Forum Message <> Reply to Message

Klemen writes:

> David, I have jsut upgraded to IDL 8.2 so I can test it...

Oh, sorry. This is the third time I've paid my money and
upgraded to something unusable to me, too. I'm a really
slow learner. :-(

> The code is, I think, ok (I haven't so much experience with Image). The only problem is your definition of you georeference. I think that your ranges are defined for a usual Mercator projection with origin in 0N, 0E. But you have moved your origin to 105.1W, 40.6N. This point should have in map coordinates values 0, 0. This means that below defined range is false:
>
> XRange: [-11711131.0, -11688226.0] (meters)
> YRange: [4914254.0, 4937159.5] (meters)
>
> You could redefine your range as:
> n = 600

> res = 38.1757
> x = findgen(n)* res - n*res*0.5
> y = reverse(x)
> xrange = [min(x), max(x)]
> yrange = [min(y), max(y)]
>
> Well, then I get at least something, I am not sure if this is exactlly what you are looking for, but right mouse click gives me at least the proper longitude when I scroll over the image.

Maybe I'm seeing something different from you, but
when I put this code into my program, I see a
map projection filling the window, and a small spec
in the middle of my window which presumably represents
my image.

Probably not what I am looking for. :-)

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 Fri, 07 Sep 2012 13:58:37 GMT
View Forum Message <> Reply to Message

Le vendredi 7 septembre 2012 15:15:35 UTC+2, David Fanning a écrit :
> Klemen writes:
>
>
>
>> David, I have jsut upgraded to IDL 8.2 so I can test it...
>
>
>
> Oh, sorry. This is the third time I've paid my money and
>
> upgraded to something unusable to me, too. I'm a really
>
> slow learner. :-(
>

>
>
>> The code is, I think, ok (I haven't so much experience with Image). The only problem is your definition of you georeference. I think that your ranges are defined for a usual Mercator projection with origin in 0N, 0E. But you have moved your origin to 105.1W, 40.6N. This point should have in map coordinates values 0, 0. This means that below defined range is false:
>
>>
>
>> XRange: [-11711131.0, -11688226.0] (meters)
>
>> YRange: [4914254.0, 4937159.5] (meters)
>
>>
>
>> You could redefine your range as:
>
>> n = 600
>
>> res = 38.1757
>
>> x = findgen(n)* res - n*res*0.5
>
>> y = reverse(x)
>
>> xrange = [min(x), max(x)]
>
>> yrange = [min(y), max(y)]
>
>>
>
>> Well, then I get at least something, I am not sure if this is exactlly what you are looking for, but right mouse click gives me at least the proper longitude when I scroll over the image.
>
>
>
> Maybe I'm seeing something different from you, but
>
> when I put this code into my program, I see a
>
> map projection filling the window, and a small spec
>
> in the middle of my window which presumably represents
>
> my image.
>
>
>

> Probably not what I am looking for. :-)
>
>
>
> 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.")

Hi David,

Here is my solution:

1st step: define the size of the image in decimal degrees, by using your "magic" resolution number. I guess that it might be retrieved by another way.

```
centerlat = 40.6000d
centerlon = -105.1000d
res = 38.1757
map = Map_Proj_Init('mercator', /gctp, ELLIPSOID='wgs 84')
cm = map_proj_forward(centerlon, centerlat, MAP=map)
xrange = cm[0] + [-300,300]*res
yrange = cm[1] + [-300,300]*res
xr = map_proj_inverse(xrange, yrange, MAP=map)
print,xr
;    -105.20288       40.521535
;    -104.99712       40.678372
```

As you can see, numbers are slightly different from yours (in latitude).

2nd step: draw the image and add the grid map

```
obj = IMAGE(googleimage, $
  GRID_UNITS=2,    ;units of degrees
  MARGIN=[0.05,0.15,0.05,0.05], $  ;to manage space for labels
```

```
   IMAGE_LOCATION=xr[0:1,0],$
   IMAGE_DIMENSIONS=[xr[0,1]-xr[0,0],xr[1,1]-xr[1,0]],$
   MAP_PROJECTION='Mercator')
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
```

The trick (that I have found in spite of the cryptic Exelis documentation, but better solutions are likely to exist) was to consider that any NG building consists in a tree of graphics objects. Then, when plotting your image, the use of the MAP_PROJECTION keyword implies that such a corresponding MAP object is added to the IMAGE one, then that a MAPGRID object should be also available. The latter can be retrieved through the MAPPROJECTION property. Please consider how much a simple underline can change to your life !
Now I did not succeed with the BOX_AXES keyword. I leave that for your exercizing ...

Alain.

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by David Fanning on Fri, 07 Sep 2012 14:03:44 GMT
View Forum Message <> Reply to Message

Alain writes:

> Here is my solution:

I'm feeling generous this morning, Alain, so I'll let you get
away with the word "solution". At least for now. ;-)

> 1st step: define the size of the image in decimal degrees, by using your "magic" resolution number. I guess that it might be retrieved by another way.
> centerlat = 40.6000d
> centerlon = -105.1000d
> res = 38.1757
> map = Map_Proj_Init('mercator', /gctp, ELLIPSOID='wgs 84')
> cm = map_proj_forward(centerlon, centerlat, MAP=map)
> xrange = cm[0] + [-300,300]*res
> yrange = cm[1] + [-300,300]*res
> xr = map_proj_inverse(xrange, yrange, MAP=map)
> print,xr
> ; -105.20288 40.521535
> ; -104.99712 40.678372
> As you can see, numbers are slightly different from yours (in latitude).

There are a couple of problems with this approach (other than it doesn't do exactly what I want). First, I have to get Map_Proj_Init involved, which the Map function is suppose to handle for me. I already know how to use Map_Proj_Init and believe me, you don't want to go through these contortions to use it! I use Map_Proj_Init (via cgMap, but it is a direct wrapper for it) in my Coyote Graphics solution. It works perfectly to navigate this image with the input data I have provided. This image is gridded in projected meter space, and Map_Proj_Init can work in projected meter space perfectly.

The second (and, I suspect, more fundamental) problem is that your solution requires me to work in lat/lon space rather than projected meter space. While it is true that I can work in this space with the Map function, and I can actually see my image with map annotations around it, the image is distorted because it has been warped into this map space. The distortion is only slight in this Mercator projection and over this map range, but if you look carefully at this image and the one created with the Coyote Graphics solution you can easily see the difference. The Google text, in particular, shows the obvious distortions.

Now, IDL has always had a bias toward warping the image to fit the map projection (I am approaching a dozen years of harping about this to them!) and working in lat/lon space. But, this is NOT the space the remote sensing community that I work with wants to work in. It makes no sense! It's like working in one of those fun-house rooms without a square corner. You get queasy just being in there! Every GeoTiff file in the world (I guess there are a few exceptions) works in projected meter space. Every satellite image on my computer works in projected meter space. I DON'T WANT TO WORK IN LAT/LON SPACE!!!

So, here is the problem with the friggin' function graphics system.

I can display an image. And I can set up a proper map projection with the Map_Proj_* routines. I just can't use the two together in a graphics window!

Does this strike you, as it does me, as ludicrous? Surely you and I and everyone else are missing the obvious solution. Is it possible that the function graphics system is not meant to be used by professional programmers?

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 Fri, 07 Sep 2012 14:05:03 GMT
View Forum Message <> Reply to Message

alx writes:

> Now I did not succeed with the BOX_AXES keyword. I leave that for your exercizing ...

Oh, yeah. I got carried away. Yes, I can't get box axes on my
map projection. ;-)

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 Fri, 07 Sep 2012 14:46:23 GMT
View Forum Message <> Reply to Message

Le vendredi 7 septembre 2012 16:05:02 UTC+2, David Fanning a écrit :
> alx writes:
>
>
>
>> Now I did not succeed with the BOX_AXES keyword. I leave that for your exercizing ...
>
>
>
> Oh, yeah. I got carried away. Yes, I can't get box axes on my
>

> map projection. ;-)
>
>
>
> 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.")

> I can display an image. And I can set up a proper map
> projection with the Map_Proj_* routines. I just can't
> use the two together in a graphics window!

Yes, I guess that you can !
If you got an image from Google which is properly rectified, you do not need for any further map projection, as far as the axis labeling is in meter/kilometer.
If now you want axes graduated in longitude/latitude, you only need for a projection tool in order to manage the scaling which is (slightly in your case) not linear.
I suppose that 'map_proj_init'/'map_proj_inverse' is what you can use. Then you can add axes with the proper labeling. This is likely what MAP_PROJECTION is doing for you in the IMAGE function! Is'nt it ?
Maybe what I say is pure non-sense, because I am not a specialist of mapping.
I am just a scientist who *needs* for clever and efficient programming tools, what IDL still is, I guess, in spite of some irritating and uncorrected lack for a serious (i.e. usable) documentation.
Alain.

Subject: Re: Display and Navigate Image in IDL 8.2
Posted by David Fanning on Fri, 07 Sep 2012 15:23:33 GMT
View Forum Message <> Reply to Message

alx writes:

>> I can display an image. And I can set up a proper map
>> projection with the Map_Proj_* routines. I just can't
>> use the two together in a graphics window!
>
> Yes, I guess that you can !
> If you got an image from Google which is properly rectified, you do not need for any further map projection, as far as the axis labeling is in meter/kilometer.

OK, I can see how to do this:

```
limit =  [-84.7500, -180.000, 84.7500, 180.000]
mapStruct = Map_Proj_Init('mercator', /gctp, $
   ellipsoid='wgs 84', limit=limit)
cm = map_proj_forward(-105.100, 40.600, map=mapStruct)
xcenter = cm[0,0]
ycenter = cm[1,0]
Print, 'XCenter: ', xcenter
Print, 'YCenter: ', ycenter
metersPixel = cgGoogle_MetersPerPixel(12)
Print, 'meters per pixels: ', metersPixel
xrange = [xcenter - (300*metersPixel), xcenter+(300*meterspixel)]
yrange = [ycenter - (300*metersPixel), ycenter+(300*meterspixel)]
Print, 'X Range: ', xrange
print, 'Y Range: ', yrange

s = Size(googleImage, /DIMENSIONS)
xscale = Abs(xrange[1] - xrange[0]) / s[1]
yscale = Abs(yrange[1] - yrange[0]) / s[2]
Print, 'X Scale: ', xscale
Print, 'Y Scale: ', yscale
x = (Findgen(s[1])*xscale) + xrange[0] + (xscale/2)
y = (Findgen(s[2])*yscale) + yrange[0] + (yscale/2)
obj = Image(googleImage, x, y, DIMENSIONS=[700,700], Margin=0.1)
```

> If now you want axes graduated in longitude/latitude, you only need
for a projection tool in order to manage the scaling which is (slightly
in your case) not linear.

My point is that my image is LINEAR!! It is gridded, as ALL of my
satellite images are, onto a projected meter scale. I wish to fit a
map projection to the image, not fit the image to a map projection.
There is a very big difference here!

> I suppose that 'map_proj_init'/'map_proj_inverse' is what you can use.

Exactly. I can set up the map projection with Map_Proj_Init and I
can use Map_Proj_Inverse to convert lat/lon space to projected
meter space. Now, I can draw symbols, lines, etc. on top of my image.

What I can NOT draw on top of my image are map annotations!!

> Then you can add axes with the proper labeling. This is likely what
MAP_PROJECTION is doing for you in the IMAGE function! Is'nt it ?

Are you suggesting I give up on the Map function and use the Axis
function to create my map annotations?

> Maybe what I say is pure non-sense, because I am not a specialist of mapping.
> I am just a scientist who *needs* for clever and efficient programming tools,
> what IDL still is, I guess, in spite of some irritating and
> uncorrected lack for a serious (i.e. usable) documentation.

It has been my experience that there are scientists in
the world who care more about something "looking" right
than actually "being" right. This is what I meant by suggesting
that maybe the function graphics system is made for this type
of scientist and not for professional programmers. But, having
something that is "almost" right is going to bite someone someday.
And, anyway, IDL has the tools to do this correctly. Why not just
use them? The documentation, such as it is, gives no indication that
this wouldn't be possible.

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 Fri, 07 Sep 2012 16:02:43 GMT
View Forum Message <> Reply to Message

alx writes:

> Yes, I guess that you can !
> If you got an image from Google which is properly rectified, you do not need for any further map projection, as far as the axis labeling is in meter/kilometer.
> If now you want axes graduated in longitude/latitude, you only need for a projection tool in order to manage the scaling which is (slightly in your case) not linear.
> I suppose that 'map_proj_init'/'map_proj_inverse' is what you can use. Then you can add axes

with the proper labeling. This is likely what MAP_PROJECTION is doing for you in the IMAGE function! Is'nt it ?
> Maybe what I say is pure non-sense, because I am not a specialist of mapping.
> I am just a scientist who *needs* for clever and efficient programming tools, what IDL still is, I guess, in spite of some irritating and uncorrected lack for a serious (i.e. usable) documentation.

You can see the problem with your particular approach
by using a larger area:

```
googleStr = "http://maps.googleapis.com/maps/api/staticmap?" + $
   "center=40.6000,-105.1000&zoom=4&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')


centerlat = 40.6000d
centerlon = -105.1000d
res = cgGoogle_MetersPerPixel(4)
map = Map_Proj_Init('mercator', /gctp, ELLIPSOID='wgs 84')
cm = map_proj_forward(centerlon, centerlat, MAP=map)
xrange = cm[0] + [-300,300]*res
yrange = cm[1] + [-300,300]*res
xr = map_proj_inverse(xrange, yrange, MAP=map)
obj = IMAGE(googleimage, $
   GRID_UNITS=2, $     ;units of degrees
   MARGIN=0.1, $  ;to manage space for labels
   IMAGE_LOCATION=xr[0:1,0],$
   IMAGE_DIMENSIONS=[xr[0,1]-xr[0,0],xr[1,1]-xr[1,0]],$
   MAP_PROJECTION='Mercator', DIMENSIONS=[700,700])
```

Here we see the map, and the image. But, the map annotations
are clearly wrong. The latitude 40.1 which should be the center
of Fort Collins, Colorado is now located somewhere far north,
up near Yellowstone National Park in Wyoming!

The Coyote Graphics routines to display the same image:

```
map = Obj_New('cgMap', 'mercator', ELLIPSOID='wgs 84', /OnImage, $
  XRange=xrange, YRange=yrange)
cgDisplay, 700, 700
cgImage, googleimage, Margin=0.1
map -> Draw
cgMap_Grid, Map=map, /Box_Axes, Color='goldenrod'
cgMap_Continents, /USA, Map=map, Color='goldenrod'
```

Here are the two images, for those of you who don't want to
run the code:

http://www.idlcoyote.com/misc/fg_google_map.png
http://www.idlcoyote.com/misc/cg_google_map.png

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 Klemen on Fri, 07 Sep 2012 20:03:20 GMT
View Forum Message <> Reply to Message

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)
```
