Subject: Using map projections to display images
Posted by seanr on Mon, 24 Aug 1998 07:00:00 GMT
View Forum Message <> Reply to Message

I'm trying to take digital air photo data with a known ground sample distance resolution and a single georeference point (taken from a GPS receiver) and display it in an arbitrary map projection. I have looked at the documentation for MAP_SET, MAP_IMAGE, and MAP_PATCH. Using the example provided in the documentation of MAP_PATCH, I have managed to get the image to come up in the right projection, but I am unsure how to keep the resolution of the image the same.

Does anyone have a good example to display an image warped to an arbitrary map projection? (perfered projection would be transverse mercator)

Thanks in advance for any replys,

Sean

---------- ----------
Sean P. Rumelhart Positive Systems, Inc.
seanr@possys.com 250 Second St. East
PH: 406.862.7745 Whitefish MT 59937
FAX: 406.862.7759 www.possys.com
---------- ----------

Subject: Re: Using map projections to display images
Posted by Paul van Delst on Fri, 28 Aug 1998 07:00:00 GMT
View Forum Message <> Reply to Message

I notice that Liam Gumley has already replied to this question and it's probably a bit incestuous (Liam works just down the hall) but I have to recommend his IMAGEMAP.PRO (http://cimss.ssec.wisc.edu/~gumley/imagemap.html) procedure. For me, and others here at CIMSS, it has made displaying satellite imagery in any projection a no-brainer. He has also made changes recently to overlay multiple images also so you can display more than one orbit of satellite data without stringing all the input data into one huge array and then displaying that (although I don't think that version is available on his webpage yet).

That's all.

paulv

seanr@possys.com wrote:

> UPDATE:
>
> Well, for those who have been following this thread, I have been playing with
> MAP_SET and MAP_IMAGE and feel that I understand them much better now. I have
> discovered a way to keep the resolution of my imagery *almost* the same.
> Basically, MAP_SET will create a window of a default size if one does not
> exist, and MAP_IMAGE will place the image warped to the selected projection
> within that window...in a best fit. So, I set things up so that the window
> size = image size of the raw image. (I will have to use tiling on my full
> implementation anyway, so having a small window to put this all to is no big
> deal, I will probably use a pixmap window, or possibly the z-buffer). Here is
> a small snippet of my test code that will place the sub image in the map
> projection and keep it at the correct resolution:
>

<code snippet snipped>

> For my limited example, this works like a charm.
>
> The one remaining problem I have is sometimes I can get a resulting image
> back that is 187 by 125 or some such, usually only a pizel or two. What I
> would like and have looked into a little is for MAP_SET and MAP_IMAGE to use
> xsize and ysize values as passed in, and not the window size. Has anyone
> attempted to do this, or should I go ahead and make my own?

--
Paul van Delst
Space Science and Engineering Center
University of Wisconsin-Madison
1225 W. Dayton St., Madison WI 53706
Ph/Fax: (608) 265-5357, 262-5974
Email: paul.vandelst@ssec.wisc.edu
Web: http://airs2.ssec.wisc.edu/~paulv

"Religious and cultural purity is a fundamentalist fantasy"
V.S. Naipaul

Subject: Re: Using map projections to display images
Posted by seanr on Fri, 28 Aug 1998 07:00:00 GMT

View Forum Message <> Reply to Message

UPDATE:

Well, for those who have been following this thread, I have been playing with
MAP_SET and MAP_IMAGE and feel that I understand them much better now. I have
discovered a way to keep the resolution of my imagery *almost* the same.
Basically, MAP_SET will create a window of a default size if one does not

exist, and MAP_IMAGE will place the image warped to the selected projection within that window...in a best fit. So, I set things up so that the window size = image size of the raw image. (I will have to use tiling on my full implementation anyway, so having a small window to put this all to is no big deal, I will probably use a pixmap window, or possibly the z-buffer). Here is a small snippet of my test code that will place the sub image in the map projection and keep it at the correct resolution:

```
image = bytarr(188,124)
openu, lun,'image.dat', /get_lun
readu,lun, IMAGE
close, lun
free_lun, lun
window, 0, xsize =188, ysize =124
TV, IMAGE  ;Display the image so we can see what it looks like before warping.

  pi = 3.1415925  LL_rad = 1.268 * 2.D * !pi / 360.D  degfix = 1.0 /
double(cos(LL_rad))  ; Earth radius = 6378.17km ==> 111.32km/degree  ; of
longitude at the equator, or 0.0089 deg/km ;  xdegpkm = .00899D * degfix  ;
fix size of longitudinal mile based on ;  ydegpkm = .00899D  ; cosine of
latitude  x_mpdeg = double(111320.0 * degfix)  y_mpdeg = double(111320.0)

;(image is approx .25 meters per pixel)
Minlon = double(-71.829 - ((94.0 *.25)/x_mpdeg))    ;71.829 lon W for center
pixel
Maxlon = double(-71.829 + ((94.0 *.25)/x_mpdeg))
Minlat = double(1.268 - ((62.0 * .25) / y_mpdeg))   ;1.268 lat N for center
pixel
Maxlat = double(1.268 + ((62.0 * .25) / y_mpdeg))
limit1 = [minlat, minlon, maxlat, maxlon]
loncenter = double(-71.829)
window, 1, xsize =188, ysize =124
MAP_SET, 0, loncenter, 0, /TRANSVERSE, limit=limit1, /noborder, xmargin=[0,0],
ymargin=[0,0]

result=MAP_IMAGE(image,startx,starty,xsize,ysize,$
   latmin=limit1(0),latmax=limit1(2),$
   lonmin=limit1(1),lonmax=limit1(3),$
   compress=1)

tv,result,0,0  ;Display the warped image on the map at the proper position.
```

For my limited example, this works like a charm.

The one remaining problem I have is sometimes I can get a resulting image back that is 187 by 125 or some such, usually only a pizel or two. What I

would like and have looked into a little is for MAP_SET and MAP_IMAGE to use xsize and ysize values as passed in, and not the window size. Has anyone attempted to do this, or should I go ahead and make my own?

---------- ----------
Sean P. Rumelhart Positive Systems, Inc.
seanr@possys.com 250 Second St. East
PH: 406.862.7745 Whitefish MT 59937
FAX: 406.862.7759 www.possys.com
---------- ----------

Subject: Re: Using map projections to display images
Posted by Chris McCarthy on Mon, 31 Aug 1998 07:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:

> Can you imagine sitting on the deck writing IDL programs
> with Glacier National Park looming up behind you.

Yes.

Subject: Re: Using map projections to display images
Posted by Erard on Wed, 02 Sep 1998 07:00:00 GMT
View Forum Message <> Reply to Message

In article <6s6l44$mf1$1@nnrp1.dejanews.com>, seanr@possys.com wrote:

> Well, for those who have been following this thread, I have been playing with
> MAP_SET and MAP_IMAGE and feel that I understand them much better now. I have
> discovered a way to keep the resolution of my imagery *almost* the same.
> Basically, MAP_SET will create a window of a default size if one does not
> exist, and MAP_IMAGE will place the image warped to the selected projection
> within that window...in a best fit. So, I set things up so that the window
> size = image size of the raw image. (I will have to use tiling on my full
> implementation anyway, so having a small window to put this all to is no big
> deal, I will probably use a pixmap window, or possibly the z-buffer). Here is
> a small snippet of my test code that will place the sub image in the map
> projection and keep it at the correct resolution:
>

I'm not sure the following answers your problem, but here is a small procedure to simplify superpositions of images and maps. It works correctly for Mercator and cylindrical projection with Map_set 1.33 (shipped with IDL 4), and haldles PS output correctly. Just forget the prep_map function, it just corrects a personal problem I had with map_set.

```
pro TVmap, image, _extra=pipo, Ps=ps, grid=grid, label=label,
position=position,$
 limit=limit

; ===================================================

; Superimpose a map to an image.
;


; Les limitse sont souvent dï¿½calï¿½es d'1 pixel, problï¿½me d'arrrondi
; (dï¿½pend du systï¿½me)...

; ===================================================


; Il faut intercepter les mots-clefs ï¿½ tester, et ils ne sont alors plus
dans _extra
if keyword_set(position) then begin
 message, /continue, 'Keyword POSITION no allowed (automatic)'
 return
endif
if n_elements(grid) eq 0 then grid=1   ; defaut is ON
if n_elements(label) eq 0 then label=1

; Limit is adjusted if provided as a 4-elt vector.
; Otherwise, map_set defaults are used
If n_elements(limit) eq 4 then begin prep_map, latcent, longcent, limit,
_extra=pipo
                    endif else latcent=(longcent=(limit=0))


sz=size(image)
Xscreen=sz(1) + 40.     ; dï¿½calages pour mise en page
Yscreen=sz(2) + 80.

!P.font=-1
```

```
th=1
if keyword_set(ps) then begin
 OldDevice = !D.name
 set_plot,'ps'
 !P.font=0
;  coefPS=!D.x_px_cm/40.  ;image ï¿½ la mï¿½me taille que sur l'ï¿½cran
 coefPS=min([!D.X_vsize/Xscreen,!D.Y_vsize/Yscreen]) ;image sur toute la feuille
 addPS=0.
 device,filename='TVmap.ps',/color,bits=8,/landscape,/Bold
 th=2
endif else begin
 window, /free, xsize=Xscreen, ysize=Yscreen
 coefPS=1.
 addPS=1.
endelse



px0 = 20.         ; small shift for page layout
py0 = 20.
;print, sz, coefPS, !D.x_vsize,!D.y_vsize

   ; Back to normal coordinates for Map_set
   ; adjust for odd and even dimensions (frame is always suposed to be
inside image)
qx=[px0,px0+sz(1)-(sz(1) mod 2)]*coefPS/(!D.X_vsize+addPS)
qy=[py0,py0+sz(2)-(sz(2) mod 2)]*coefPS/(!D.Y_vsize+addPS)


tv, image, px0*coefPS,py0*coefPS,xsize=sz(1)*coefPS, ysize=sz(2)*coefPS

;print, qx, qy
del=(qx(1)-qx(0))*0.01     ; Compensate a small dilatation of limits
qx(0)=qx(0)-del              ; performed in MAP_SET with a similar dilatation
qx(1)=qx(1)+del         ; of the plotting area.
del=(qy(1)-qy(0))*0.01
qy(0)=qy(0)-del
qy(1)=qy(1)+del


print, latcent, longcent
print, limit

map_set,latcent, longcent ,grid=grid, limit=limit,/noerase, label=label,$
 londel=30.,latdel=30.,position=[qx(0),qy(0),qx(1),qy(1)],$
 glinethick=th, _extra=pipo

; Ici, il faut tester la condition de sortie de Map_set et continuer en cas
```

d'erreur

```
!P.font=-1
if keyword_set(ps) then begin
 device,/close
 Set_Plot, OldDevice
endif

end
```

--
Stï¿½phane Erard

Institut d'Astrophysique Spatiale
Orsay, France
www.ias.fr/cdp
erard@ias.fr