Subject: Re: ENVI_INIT_TILE tiling problem
Posted by a.l.j.ford on Tue, 17 Mar 2009 23:04:24 GMT
View Forum Message <> Reply to Message

On Mar 17, 2:09 pm, "Jean H." <jghas...@DELTHIS.ucalgary.ANDTHIS.ca>
wrote:
> Hi,
>
>>> Yes, yet again you were absolutely correct. the problem was with my
>>> "output_dsm". In order to get a FID from output_dsm I used
>>> ENVI_ENTER_DATA (maybe this isn't the best way??), which then seemed
>>> to prevent output_dsm being used as an array. Therefore, before I used
>>> ENVI_ENTER_DATA I made a copy of output_dsm, called output_dsm_copy!
>>> This meant it was preserved as an array. This might not be the best
>>> way to do things (?), but it worked.
>
> If you have a file, use ENVI_OPEN_FILE, it will give you a valid FID.
> Otherwise, you can use ENVI_SELECT or save your array and open it this
> way. You don't want to have the same content twice in memory!
>
>
>
>>> Things are now working mostly OK and the tiling and interpolation
>>> appear to complete, except that the zero pixels in my array we're
>>> interpolated over. Therefore I changed my interpolation to the
>>> following (I decided to go with MIN_CURVE_SURF in this example, but
>>> the same should be true for TRI_SURF):
>
>>> tile_id=ENVI_INIT_TILE(fid_output_DSM, my_pos,
>>> num_tiles=number_of_tiles)
>>> FOR i=0, number_of_tiles-1 DO BEGIN
>>> tile_data_interp=ENVI_GET_TILE(tile_id, i)
>
> ok, you get the data
>
>
>
>>> ;Processing within Tiling
>>> index= WHERE (output_dsm_copy GT 0.0)
>
>>> x = index MOD DIMS[2]
>>> y = index/DIMS[4]
>
>>> z = output_dsm_copy [index]
>
> ? Don't you want to use the tile data?? tile_data_interp
> What is the point of tiling your data if you end up using the whole array?
>

>
>
>>> tile_data_interp = MIN_CURVE_SURF (z, x, y, gs=[1,1],bounds=[1,1,DIMS
>>> [2],DIMS[4]])
>
> You just erase the data extracted... ok, result of the interpolation
>
>>> ; Close Tiling
>
>>> ENDFOR
>>> ENVI_TILE_DONE, tile_id
>
> but you have not saved the content of the tile.
> You had the array, have modified it and that's it!
>
> Read the help file.. you have the choice of 1) create an array of the
> size of the complete image, store the result of each tile in it and load
> it in memory (using envi_enter_data this time), or 2) open the output
> file (new file), write the content of each tile in it using writeu, and
> when you are done processing every tiles, set up the header.
>
> Jean
>
>
>
>>> Can you spot the problem?? When I run it the interpolation runs out of
>>> memory for creating the array (% Unable to allocate memory: to make
>>> array.
>>>   Not enough space). This is because I'm using DIMS for the original
>>> file outside of the tiling... whereas I need to use different, smaller
>>> DIMS within the tiles (the x,y, dimensions of the tiles themselves).
>>> How can I get the tile dimensions and use them here??
>
>>> Many thanks again!
>
>> There's a couple ways you can figure this out. You can actually
>> control the interleave of the tile, which would mean you would know
>> the tile dimensions most of the time. This isn't true 100% of the
>> time, but i've never run into a case where it wasn't. But anyway, a
>> pretty reliable way to do this would be to put this code in your tile
>> loop:
>
>> if i eq 0 then s = size(tile_data, /dimensions)
>
>> which would make s[0] the size in the x direction and s[1] size in the
>> y direction. You can feed that into min_curve_surf or tri_surf later.
>
>

Yes, very good points!!

This is where I am now. So far I don't get any errors, but the
processing takes forever and never finishes, as if it has hung. Am I
still going wrong somewhere. I have to say I'm finding the mix of
tiling and interpolation somewhat challenging, so please forgive me!
I'm aware that you can use ENVI_REPORT_INIT, ENVI_REPORT_INC and
ENVI_REPORT_STAT to show progress, but right now I dare not use them,
for fear of introducing even more errors! One step at a time I say!
Many thanks for your continued help.

```
openw, unit, 'output_test', /get_lun

;Open Tiling

 tile_id=ENVI_INIT_TILE(fid_output_DSM, my_pos,
num_tiles=number_of_tiles)

 FOR i=0, number_of_tiles-1 DO BEGIN

 tile_data=ENVI_GET_TILE(tile_id, i)

;Processing within Tiling

 if i eq 0 then tile_size = size(tile_data, /dimensions)

 index= WHERE (tile_data GT 0.0)

 x = index MOD tile_size[0]
 y = index/tile_size[1]

 z = tile_data [index]

 tile_data_interpolated = MIN_CURVE_SURF (z, x, y, gs=[1,1],bounds=
[1,1,tile_size[0],tile_size[1]])

 writeu, unit, tile_data_interpolated

ENDFOR

free_lun, unit
  envi_setup_head, fname=output_test, ns=ns, nl=nl, nb=nb, $
    data_type=data_type, offset=0, interleave=0, $
    xstart=xstart+dims[1], ystart=ystart+dims[3], $
    descrip='Interpolation output', /write, /open
```