

---

Subject: Re: ENVI\_INIT\_TILE tiling problem  
Posted by [Jean H.](#) on Tue, 17 Mar 2009 14:09:23 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

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.
```

---