Subject: Re: Locate pixels that fall within other pixel-geo search
Posted by rogass on Thu, 09 Sep 2010 20:41:11 GMT

View Forum Message <> Reply to Message

On 9 Sep., 22:07, Snow53 <jennifer_wa...@hotmail.com> wrote:
> Hi,
>
> I have two images with slightly different dimensions, but the same
> projection. Image 1 has coarse resolution. Image 2 has fine
> resolution.
>
> For each coarse pixel in Image 1, I need to find all fine resolution
> pixels that fall within that pixel (based on the condition that they
> fall within the same geographic extent of that coarse pixel).
>
> I'm reading in the file with envi_open_file, so the files should have
> all the geographic info needed.
>
> Has anyone done something similar? Could anyone suggest a good way to
> do this selection?
>
> Thanks!

Maybe you can use this routine or some ideas frome it:

```
function cr_get_tiles,im, wx,wy
 s = size(im,/dimensions)
 ;l = make_array(s,/index,/ulong)
 run1 = 0l
 run2 = 0l
 tilex  = round((s[0]  -  (s[0]  mod wx))/wx)
 tilexpart = round(float(s[0]   mod wx)/wx)
 tilex  += tilexpart
 tiley  = round((s[1]  -  (s[1]  mod wy))/wy)
 tileypart = round(float(s[1]   mod wy)/wy)
 tiley  += tileypart
; tiley = round((s[1]  -  (s[1]  mod wy))/wy    +  float(s[1]
mod wy)/wy)
   newim  =  make_array([wx,wy,s[2],tilex*tiley],type=size(im,/
type))
 for i=0l, tilex-1l do begin
  for j=0l, tiley-1l do begin
    if i eq (tilex-1l) && tilexpart gt 0 then $
    newim[0: s[0]-i*wx-1l  ,*,*,run1] = im[i*wx:*,j*wy:((j
+1l)*wy)-1l,*] else $
    if j eq (tiley-1l) && tileypart gt 0 then $
    newim[*,0:s[1]-j*wy-1l,*,run1]  = im[i*wx:((i
+1l)*wx)-1l,j*wy:*,*] else $
```

```
    newim[*,*,*,run1] = im[i*wx:((i+1l)*wx)-1l,j*wy:((j+1l)*wy)-1l,*]
   run1++
  endfor
 endfor
 undefine, im
return, newim
end
```

wx and wy is the ratio between coarse and fine image.

Hope it helps

CR

Subject: Re: Locate pixels that fall within other pixel-geo search
Posted by Maxwell Peck on Fri, 10 Sep 2010 08:28:50 GMT

On Sep 10, 6:07 am, Snow53 <jennifer_wa...@hotmail.com> wrote:
> Hi,
>
> I have two images with slightly different dimensions, but the same
> projection. Image 1 has coarse resolution. Image 2 has fine
> resolution.
>
> For each coarse pixel in Image 1, I need to find all fine resolution
> pixels that fall within that pixel (based on the condition that they
> fall within the same geographic extent of that coarse pixel).
>
> I'm reading in the file with envi_open_file, so the files should have
> all the geographic info needed.
>
> Has anyone done something similar? Could anyone suggest a good way to
> do this selection?
>
> Thanks!

There's probably a few ways to do this depending on the exact dataset
(especially the relative resolution differences). One way would be to
obtain the corner coordinates of the pixel in your low resolution
image then use envi_convert_file_coordinates to convert these to
corner pixels in the high resolution image. Then use envi_get_data on
the subsetted dimensions. Be careful with adding/taking 0.5 of a pixel
when appropriate.

Max

Subject: Re: Locate pixels that fall within other pixel-geo search
Posted by tegus on Sat, 11 Sep 2010 13:07:10 GMT
View Forum Message <> Reply to Message

On Sep 9, 4:07 pm, Snow53 <jennifer_wa...@hotmail.com> wrote:
> Hi,
>
> I have two images with slightly different dimensions, but the same
> projection. Image 1 has coarse resolution. Image 2 has fine
> resolution.
>
> For each coarse pixel in Image 1, I need to find all fine resolution
> pixels that fall within that pixel (based on the condition that they
> fall within the same geographic extent of that coarse pixel).
>
> I'm reading in the file with envi_open_file, so the files should have
> all the geographic info needed.
>
> Has anyone done something similar? Could anyone suggest a good way to
> do this selection?
>
> Thanks!

Hi!

In IDL you could do this using J.D. Smith's Hist_nd histogram routine
which is available on David Fanning's website (dfanning.com). While on
David's website you'll also want to check out JD Smith's HISTOGRAM:
The Breathless Horror and Disgust.
The way I work this problem is to rebin the fine resolution image
(i.e., 2D histogram) to match the resolution (binning) of the coarse
image (coverage and projection of the two images need to coincide).
The key is to create a reverse index array using the REVERSE_INDICES
keyword of Hist_nd. The reverse index allows you to look up the
contents of each bin in the new histogram, based on the finer
resolution, at the coarser resolution.

Bill

Subject: Re: Locate pixels that fall within other pixel-geo search
Posted by rogass on Sat, 11 Sep 2010 16:16:49 GMT
View Forum Message <> Reply to Message

On 11 Sep., 15:07, tegus <tegusbillhar...@gmail.com> wrote:
> On Sep 9, 4:07 pm, Snow53 <jennifer_wa...@hotmail.com> wrote:
>
>

>
>
>
>> Hi,
>
>> I have two images with slightly different dimensions, but the same
>> projection. Image 1 has coarse resolution. Image 2 has fine
>> resolution.
>
>> For each coarse pixel in Image 1, I need to find all fine resolution
>> pixels that fall within that pixel (based on the condition that they
>> fall within the same geographic extent of that coarse pixel).
>
>> I'm reading in the file with envi_open_file, so the files should have
>> all the geographic info needed.
>
>> Has anyone done something similar? Could anyone suggest a good way to
>> do this selection?
>
>> Thanks!
>
> Hi!
>
> In IDL you could do this using J.D. Smith's Hist_nd histogram routine
> which is available on David Fanning's website (dfanning.com). While on
> David's website you'll also want to check out JD Smith's HISTOGRAM:
> The Breathless Horror and Disgust.
> The way I work this problem is to rebin the fine resolution image
> (i.e., 2D histogram) to match the resolution (binning) of the coarse
> image (coverage and projection of the two images need to coincide).
> The key is to create a reverse index array using the REVERSE_INDICES
> keyword of Hist_nd. The reverse index allows you to look up the
> contents of each bin in the new histogram, based on the finer
> resolution, at the coarser resolution.
>
> Bill

Cool Way! Really.

Regards

Chris