Subject: Re: filling multi spectral image
Posted by Jeremy Bailin on Thu, 26 Mar 2009 13:15:44 GMT
View Forum Message <> Reply to Message

On Mar 25, 8:22 am, a.mozafari1...@googlemail.com wrote:
> Hi
> Folks I have one Hyperion scene (multi spectral image) with 250 bands
> and each band has some no datas number. I want to fill this no data
> for all 250 bands with the average of neighbour data. Is there any
> easy way to do this?
> Any help highly will be appreciated.
> Cheers

This question keeps coming up lately, doesn't it? Might be worth searching the newsgroup...

Anyway, this would be my quick-and-dirty solution if you can safely assume that no "bad" pixels are either at the edge of the image or adjacent to another bad pixel. Assume that scene is a [nx,ny,nband] floating point array, and bad pixels are marked by "badpixelvalue".

```
badpix = where(scene eq badpixelvalue, nbadpix)
if nbadpix gt 0 then begin
  badpix_xyb = array_indices(scene, badpix)
  xneighbours = rebin(badpix_xyb[0,*],4,nbadpix)+rebin([-1,1,-1,1],
4,nbadpix)
  yneighbours = rebin(badpix_xyb[1,*],4,nbadpix)+rebin([-1,-1,1,1],
4,nbadpix)
  scene[badpix] = total(scene[xneighbours,yneighbours,rebin(badpix_xyb
[2,*],4,nbadpix)])/4.
endif
```

-Jeremy.

Subject: Re: filling multi spectral image
Posted by a.mozafari1211 on Fri, 27 Mar 2009 10:03:54 GMT
View Forum Message <> Reply to Message

On Mar 26, 2:15 pm, Jeremy Bailin <astroco...@gmail.com> wrote:
> On Mar 25, 8:22 am, a.mozafari1...@googlemail.com wrote:
>
>> Hi
>> Folks I have one Hyperion scene (multi spectral image) with 250 bands
>> and each band has some no datas number. I want to fill this no data
>> for all 250 bands with the average of neighbour data. Is there any
>> easy way to do this?
>> Any help highly will be appreciated.

```
>>  Cheers
>
>  This question keeps coming up lately, doesn't it? Might be worth
>  searching the newsgroup...
>
>  Anyway, this would be my quick-and-dirty solution if you can safely
>  assume that no "bad" pixels are either at the edge of the image or
>  adjacent to another bad pixel. Assume that scene is a [nx,ny,nband]
>  floating point array, and bad pixels are marked by "badpixelvalue".
>
>  badpix = where(scene eq badpixelvalue, nbadpix)
>  if nbadpix gt 0 then begin
>    badpix_xyb = array_indices(scene, badpix)
>    xneighbours = rebin(badpix_xyb[0,*],4,nbadpix)+rebin([-1,1,-1,1],
>  4,nbadpix)
>    yneighbours = rebin(badpix_xyb[1,*],4,nbadpix)+rebin([-1,-1,1,1],
>  4,nbadpix)
>    scene[badpix] = total(scene[xneighbours,yneighbours,rebin(badpix_xyb
>  [2,*],4,nbadpix)])/4.
>  endif
>
>  -Jeremy.
```

thank you Jeremy. seems work perfect.
Cheers