Subject: Re: Read _ tiff with Sub_Rect Keyword
Posted by btt on Fri, 22 Jul 2005 13:52:59 GMT
View Forum Message <> Reply to Message

raval.chintan@gmail.com wrote:
> Hi...
>
> I have a geo tiff file which containing the 14000 X 14000 Pixels
> (Samples and Lines), I am reading it through the read_tiff function.
> Over here i want to show that image on to the 500 X 500 window Means
> by buffer for image will contain dimension [3,500,500]. For that I am
> reading with the help of read_tiff function with the sub_rect keyword.
> Where i m reading the pixels based on the ratio of 14000/500, but it is
> taking to much time. While the same thing in ENVI it is taking less
> time to read the image and display it. So is there any other method for
> read that image fast. for that i have to write my own code in IDL to
> read the tiff file?
>
> Regards,
> Chintan Raval
>
Hello,

I don't think I understand what it is that you are trying to do, but I
would assume that IDL and ENVI are accessing the image using the same
procedure. Here is an example of how to use the SUB_RECT keyword (which
I use all the time on much smaller images with no problem.)

```
file = FILE_SEARCH(!DIR, 'image.tif')
whole = READ_TIFF(file[0])
sub = READ_TIFF(file[0], sub_rect = [200, 200, 50, 100] )
TV, whole
TV, sub, 100, 0
```

Is this example similar to how you are using the SUB_RECT keyword?

Ben

Subject: Re: Read _ tiff with Sub_Rect Keyword
Posted by raval.chintan on Mon, 25 Jul 2005 10:36:58 GMT
View Forum Message <> Reply to Message

Hi Ben,

Thank You for your answer.

Over here you are right , for reading the image file which has small

dimension, But my image dimension is 14000x14000 or more than that, and i want to display this image in the size of 500X500 window. Here is my code, try out with the image which has dimension more than 6000 X 6000.

```
pro read_geotiff
  file = dialog_pickfile(filter=['*.tif', '*.tif'])
  if file eq '' then return
  print,systime(0)
  res = query_tiff(file,info,geotiff=geostruct)
  samples = info.dimensions[0]
  lines = info.dimensions[1]
  xper = 25
  yper = 25
  ximg = samples/xper
  yimg = lines/yper
  print,samples,lines,ximg,yimg
  image = bytarr(3,ximg,yimg)

  for i = 0, yimg*(yper-1), yimg do begin
   for j = 0, ximg*(xper-1), ximg do begin
     a = read_tiff(file,sub_rect=[j,i,ximg,yimg])
     sx = j/xper
     sy = i/yper
     image[0:2,sx:sx+ximg/xper,sy:sy+yimg/yper] = a[0:2,$
0:ximg-1:xper,0:yimg-1:yper]
   endfor
  endfor

  print,systime(0)
  window,0,xsize=ximg,ysize=yimg
  tv, image, /order, /true

end
```

If you know the other methode apart from this then please let me know. Over here one solution is to make our program to read tiff file, but i do not want to do this. Because if ENVI is using this read_tiff function then there should be other method as I think (Because envi is taking less time to display the big tiff image).

Regards
Chintan
Ben Tupper wrote:
> raval.chintan@gmail.com wrote:
>> Hi...
>>
>> I have a geo tiff file which containing the 14000 X 14000 Pixels
>> (Samples and Lines), I am reading it through the read_tiff function.

>> Over here i want to show that image on to the 500 X 500 window Means
>> by buffer for image will contain dimension [3,500,500]. For that I am
>> reading with the help of read_tiff function with the sub_rect keyword.
>> Where i m reading the pixels based on the ratio of 14000/500, but it is
>> taking to much time. While the same thing in ENVI it is taking less
>> time to read the image and display it. So is there any other method for
>> read that image fast. for that i have to write my own code in IDL to
>> read the tiff file?
>>
>> Regards,
>> Chintan Raval
>>
> Hello,
>
> I don't think I understand what it is that you are trying to do, but I
> would assume that IDL and ENVI are accessing the image using the same
> procedure. Here is an example of how to use the SUB_RECT keyword (which
> I use all the time on much smaller images with no problem.)
>
> file = FILE_SEARCH(!DIR, 'image.tif')
> whole = READ_TIFF(file[0])
> sub = READ_TIFF(file[0], sub_rect = [200, 200, 50, 100])
> TV, whole
> TV, sub, 100, 0
>
> Is this example similar to how you are using the SUB_RECT keyword?
>
> Ben

Subject: Re: Read _ tiff with Sub_Rect Keyword
Posted by Chris[2] on Mon, 25 Jul 2005 15:32:59 GMT
View Forum Message <> Reply to Message

Hi Chintan,

Well, you might try reading in complete scanlines (rows) instead of just doing tiles. For many TIFF files, the tiles are actually stored using scanlines. So for example your tilesize in the file might be 14000x1. So if you are trying to read in 500x500 chunks, it is extremely inefficient because it needs to read in 500 of the 14000x1 tiles, and then throw away most of the information. Then you go on to the next tile and it *again* reads in the same 500 14000x1 tiles.

Also, even if your TIFF tilesize isn't 14000x1, it is sometimes still more efficient to read in entire scanlines because your operating system will tend to cache contiguous blocks of the file in memory.

I think ENVI reads using scanlines.

Hope this helps.

-Chris
Research Systems, Inc.

<raval.chintan@gmail.com> wrote in message
news:1122287818.648388.298350@g14g2000cwa.googlegroups.com.. .
> Hi Ben,
>
> Thank You for your answer.
>
> Over here you are right , for reading the image file which has small
> dimension, But my image dimension is 14000x14000 or more than that, and
> i want to display this image in the size of 500X500 window. Here is my
> code, try out with the image which has dimension more than 6000 X 6000.
>
> pro read_geotiff
> file = dialog_pickfile(filter=['*.tif', '*.tif'])
> if file eq '' then return
> print,systime(0)
> res = query_tiff(file,info,geotiff=geostruct)
> samples = info.dimensions[0]
> lines = info.dimensions[1]
> xper = 25
> yper = 25
> ximg = samples/xper
> yimg = lines/yper
> print,samples,lines,ximg,yimg
> image = bytarr(3,ximg,yimg)
>
> for i = 0, yimg*(yper-1), yimg do begin
> for j = 0, ximg*(xper-1), ximg do begin
> a = read_tiff(file,sub_rect=[j,i,ximg,yimg])
> sx = j/xper
> sy = i/yper
> image[0:2,sx:sx+ximg/xper,sy:sy+yimg/yper] = a[0:2,$
> 0:ximg-1:xper,0:yimg-1:yper]
> endfor
> endfor
>
> print,systime(0)
> window,0,xsize=ximg,ysize=yimg
> tv, image, /order, /true
>
> end

>
> If you know the other methode apart from this then please let me know.
> Over here one solution is to make our program to read tiff file, but i
> do not want to do this. Because if ENVI is using this read_tiff
> function then there should be other method as I think (Because envi is
> taking less time to display the big tiff image).
>
> Regards
> Chintan
> Ben Tupper wrote:
>> raval.chintan@gmail.com wrote:
>>> Hi...
>>>
>>> I have a geo tiff file which containing the 14000 X 14000 Pixels
>>> (Samples and Lines), I am reading it through the read_tiff function.
>>> Over here i want to show that image on to the 500 X 500 window Means
>>> by buffer for image will contain dimension [3,500,500]. For that I am
>>> reading with the help of read_tiff function with the sub_rect keyword.
>>> Where i m reading the pixels based on the ratio of 14000/500, but it is
>>> taking to much time. While the same thing in ENVI it is taking less
>>> time to read the image and display it. So is there any other method for
>>> read that image fast. for that i have to write my own code in IDL to
>>> read the tiff file?
>>>
>>> Regards,
>>> Chintan Raval
>>>
>> Hello,
>>
>> I don't think I understand what it is that you are trying to do, but I
>> would assume that IDL and ENVI are accessing the image using the same
>> procedure. Here is an example of how to use the SUB_RECT keyword (which
>> I use all the time on much smaller images with no problem.)
>>
>> file = FILE_SEARCH(!DIR, 'image.tif')
>> whole = READ_TIFF(file[0])
>> sub = READ_TIFF(file[0], sub_rect = [200, 200, 50, 100])
>> TV, whole
>> TV, sub, 100, 0
>>
>> Is this example similar to how you are using the SUB_RECT keyword?
>>
>> Ben
>

Subject: Re: Read _ tiff with Sub_Rect Keyword

Hi Chris,

Thank You for this valuable suggestion, it is working fine, and is more efficient. You can find the new code, for the opening the image, below:

```
pro read_geotiff
  file = dialog_pickfile(filter=['*.tif', '*.tif'])
  if file eq '' then return
  print,systime(0)
  res = query_tiff(file,info,geotiff=geostruct)
  samples = info.dimensions[0]
  lines = info.dimensions[1]
  xper = 25
  yper = 25
  ximg = samples/xper
  yimg = lines/yper
  print,samples,lines,ximg,yimg
  image = bytarr(3,ximg,yimg)
  j=0
  m=0
  for i = 0, lines-1, yper+1 do begin

     a = read_tiff(file,sub_rect=[0,i,samples,1])
     k=0
     for j =0 ,samples-1,xper+1 do begin
       image[0,k,m] = a[0,j]
       image[1,k,m] = a[1,j]
       image[2,k,m] = a[2,j]
       k++
     endfor
     m++

   endfor

  print,systime(0)
  window,0,xsize=ximg,ysize=yimg
  tv, image,/order,  /true

end
```

Any new suggestions are welcome.
With Regards,
Chintan

Chris Torrence wrote:
> Hi Chintan,
>
> Well, you might try reading in complete scanlines (rows) instead of just
> doing tiles. For many TIFF files, the tiles are actually stored using
> scanlines. So for example your tilesize in the file might be 14000x1. So if
> you are trying to read in 500x500 chunks, it is extremely inefficient
> because it needs to read in 500 of the 14000x1 tiles, and then throw away
> most of the information. Then you go on to the next tile and it *again*
> reads in the same 500 14000x1 tiles.
>
> Also, even if your TIFF tilesize isn't 14000x1, it is sometimes still more
> efficient to read in entire scanlines because your operating system will
> tend to cache contiguous blocks of the file in memory.
>
> I think ENVI reads using scanlines.
>
> Hope this helps.
>
> -Chris
> Research Systems, Inc.
>
>
> <raval.chintan@gmail.com> wrote in message
> news:1122287818.648388.298350@g14g2000cwa.googlegroups.com.. .
>> Hi Ben,
>>
>> Thank You for your answer.
>>
>> Over here you are right , for reading the image file which has small
>> dimension, But my image dimension is 14000x14000 or more than that, and
>> i want to display this image in the size of 500X500 window. Here is my
>> code, try out with the image which has dimension more than 6000 X 6000.
>>
>> pro read_geotiff
>> file = dialog_pickfile(filter=['*.tif', '*.tif'])
>> if file eq '' then return
>> print,systime(0)
>> res = query_tiff(file,info,geotiff=geostruct)
>> samples = info.dimensions[0]
>> lines = info.dimensions[1]
>> xper = 25
>> yper = 25
>> ximg = samples/xper
>> yimg = lines/yper
>> print,samples,lines,ximg,yimg
>> image = bytarr(3,ximg,yimg)
>>

>>   for i = 0, yimg*(yper-1), yimg do begin
>>   for j = 0, ximg*(xper-1), ximg do begin
>>   a = read_tiff(file,sub_rect=[j,i,ximg,yimg])
>>   sx = j/xper
>>   sy = i/yper
>>   image[0:2,sx:sx+ximg/xper,sy:sy+yimg/yper] = a[0:2,$
>> 0:ximg-1:xper,0:yimg-1:yper]
>>   endfor
>>   endfor
>>
>>   print,systime(0)
>>   window,0,xsize=ximg,ysize=yimg
>>   tv, image, /order, /true
>>
>> end
>>
>> If you know the other methode apart from this then please let me know.
>> Over here one solution is to make our program to read tiff file, but i
>> do not want to do this. Because if ENVI is using this read_tiff
>> function then there should be other method as I think (Because envi is
>> taking less time to display the big tiff image).
>>
>> Regards
>> Chintan
>> Ben Tupper wrote:
>>> raval.chintan@gmail.com wrote:
>>>> Hi...
>>>>
>>>> I have a geo tiff file which containing the 14000 X 14000 Pixels
>>>> (Samples and Lines), I am reading it through the read_tiff function.
>>>> Over here i want to show that image on to the 500 X 500 window Means
>>>> by buffer for image will contain dimension [3,500,500]. For that I am
>>>> reading with the help of read_tiff function with the sub_rect keyword.
>>>> Where i m reading the pixels based on the ratio of 14000/500, but it is
>>>> taking to much time. While the same thing in ENVI it is taking less
>>>> time to read the image and display it. So is there any other method for
>>>> read that image fast. for that i have to write my own code in IDL to
>>>> read the tiff file?
>>>>
>>>> Regards,
>>>> Chintan Raval
>>>>
>>> Hello,
>>>
>>> I don't think I understand what it is that you are trying to do, but I
>>> would assume that IDL and ENVI are accessing the image using the same
>>> procedure. Here is an example of how to use the SUB_RECT keyword (which
>>> I use all the time on much smaller images with no problem.)

>>>
>>> file = FILE_SEARCH(!DIR, 'image.tif')
>>> whole = READ_TIFF(file[0])
>>> sub = READ_TIFF(file[0], sub_rect = [200, 200, 50, 100])
>>> TV, whole
>>> TV, sub, 100, 0
>>>
>>> Is this example similar to how you are using the SUB_RECT keyword?
>>>
>>> Ben
>>

Subject: Re: Read _ tiff with Sub_Rect Keyword
Posted by edward.s.meinel@aero. on Tue, 26 Jul 2005 13:15:46 GMT
View Forum Message <> Reply to Message

Isn't it more efficient to write

```
for j =0 ,samples-1,xper+1 do begin
  image[0:2,k,m] = a[0:2,j]
  k++
endfor
```

Subject: Re: Read _ tiff with Sub_Rect Keyword
Posted by Chris[2] on Wed, 27 Jul 2005 19:28:55 GMT
View Forum Message <> Reply to Message

Couldn't you eliminate the loop entirely using stride?

```
  image[*, *, m] = a[*,0:samples-1:xper+1]
```

Or, even better, since you are inserting the "a" array into a contiguous
block of memory in "image", you can just use 0,0 as the subscripts:

```
  image[0, 0, m] = a[*,0:samples-1:xper+1]
```

This will insert the entire "a" subarray into "image", starting at position
[0,0]. This is *much* faster than using index ranges, or even the *'s,
because IDL doesn't have to compute the locations internally. It just copies
the data as one block.

One other point. I think you want just "xper" in your loop, not "xper+1".
And finally, if your image isn't a multiple of "xper", your array indexing
will run off the end. So I think you really want to compute ximg and yimg
as:

```
    ximg = (samples + xper - 1)/xper
    yimg = (lines + yper - 1)/yper
```

-Chris

<meinel@aero.org> wrote in message
news:1122383746.592213.326040@g44g2000cwa.googlegroups.com.. .

```
>  Isn't it more efficient to write
>
>       for j =0 ,samples-1,xper+1 do begin
>         image[0:2,k,m] = a[0:2,j]
>         k++
>       endfor
>
```