Subject: Re: TIFF Read/Write Bug?
Posted by Heinz Stege on Wed, 08 May 2013 22:19:43 GMT
View Forum Message <> Reply to Message

Hi David,

below is the result from my IDL 8.0.1, 32 bit, WinXP. No error.

```
IDL> fn='F:\scans\Yms0001raw.tif'
IDL> b=read_tiff(fn)
% Loaded DLM: TIFF.
IDL> help,b
B               UINT      = Array[4, 4704, 6992]
IDL> b=fix(b)
IDL> b=reform(b[1,*,*])
IDL> help,b
B               INT       = Array[4704, 6992]
IDL> write_tiff,'test4.tif',b,/short,/signed
IDL> c=read_tiff('test4.tif')
IDL> help,c
C               INT       = Array[4704, 6992]
IDL> print,array_equal(b,c)
   1
IDL> print,!version
{ x86 Win32 Windows Microsoft Windows 8.0.1 Oct  5 2010      32
64}
```

Cheers, Heinz

Subject: Re: TIFF Read/Write Bug?
Posted by Phillip Bitzer on Thu, 09 May 2013 14:37:15 GMT
View Forum Message <> Reply to Message

I'll add the writing an integer array to a tiff and then reading it back gives the expected integer array result on IDL 8.2.2, OSX 10.8.3, as well.

Subject: Re: TIFF Read/Write Bug?
Posted by David Fanning on Fri, 10 May 2013 20:48:03 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> Is there a problem writing signed integer arrays to TIFF files?
>
> Here is my code:

>
>   outFilename = Filepath('forest.tif')
>   ageImage = Fix(ageImage)
>   Help, ageImage
>   AGEIMAGE   INT   = Array[8600, 8407]
>   ageImage[missingIndices]=-999
>   Write_Tiff, outFilename, Reverse(ageImage,2), /SHORT, /SIGNED
>   testImage = Read_Tiff(outfilename)
>   testimage = Reverse(testimage,2)
>   Help, testImage
>   TESTIMAGE   BYTE   = Array[8600, 8407]
>
> This is IDL 8.2.2, 64-bit, on Windows 7. '

Whoa! I left this out of my example the other day because it seemed (at
the time) completely irrelevant.

Here is test code that works perfectly:

```
;*****************************************
outFilename = 'forest.tif'
ageimage = read_tiff(file_which('boulder.tif'))
ageImage = Fix(ageImage)
Help, ageImage
dims = SIZE(ageimage, /DIMENSIONS)
missingIndices = long(randomu(seed, 1000) * product(dims))
ageImage[missingIndices]=-999
Write_Tiff, outFilename, Reverse(ageImage,2), GEOTIFF=geo, $
   /SHORT, /SIGNED
testImage = Read_Tiff(outfilename)
testimage = Reverse(testimage,2)
Help, testImage
END
;*****************************************
```

Running it gives me:

```
AGEIMAGE        INT       = Array[1071, 1390]
TESTIMAGE       INT       = Array[1071, 1390]
```

BUT!!!

If I add color vectors to the file, watch this:

```
;*****************************************
outFilename = 'forest.tif'
ageimage = read_tiff(file_which('boulder.tif'))
ageImage = Fix(ageImage)
Help, ageImage
```

```
dims = SIZE(ageimage, /DIMENSIONS)
missingIndices = long(randomu(seed, 1000) * product(dims))
ageImage[missingIndices]=-999
TVLCT, r, g, b, /GET
Write_Tiff, outFilename, Reverse(ageImage,2), GEOTIFF=geo, $
   /SHORT, /SIGNED, RED=r, GREEN=g, BLUE=b
testImage = Read_Tiff(outfilename)
testimage = Reverse(testimage,2)
Help, testImage
END
;*****************************************

AGEIMAGE        INT       = Array[1071, 1390]
TESTIMAGE       BYTE      = Array[1071, 1390]
```

Is that weird, or what!?

Can anyone reproduce this on their machines?

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: TIFF Read/Write Bug?
Posted by David Fanning on Fri, 10 May 2013 21:08:55 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> If I add color vectors to the file, watch this:
>
```
> ;*****************************************
> outFilename = 'forest.tif'
> ageimage = read_tiff(file_which('boulder.tif'))
> ageImage = Fix(ageImage)
> Help, ageImage
> dims = SIZE(ageimage, /DIMENSIONS)
> missingIndices = long(randomu(seed, 1000) * product(dims))
> ageImage[missingIndices]=-999
> TVLCT, r, g, b, /GET
```

```
>  Write_Tiff, outFilename, Reverse(ageImage,2), GEOTIFF=geo, $
>     /SHORT, /SIGNED, RED=r, GREEN=g, BLUE=b
>  testImage = Read_Tiff(outfilename)
>  testimage = Reverse(testimage,2)
>  Help, testImage
>  END
>  ;*****************************************
>
>  AGEIMAGE        INT       = Array[1071, 1390]
>  TESTIMAGE       BYTE      = Array[1071, 1390]
>
>  Is that weird, or what!?
```

For me, at least, it doesn't matter what kind of data I put into the TIFF file (I've tried SHORT, LONG, FLOAT, and DOUBLE), if I add color vectors, then I can ONLY get BYTE data back from the file.

Again, Windows 64-bit and IDL 8.2.2.

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: TIFF Read/Write Bug?
Posted by Phillip Bitzer on Fri, 10 May 2013 21:29:16 GMT
View Forum Message <> Reply to Message

I similar behavior David. Integer array in, integer array out .. unless the color vectors are included. Then it's byte out.

```
IDL> help, !version
** Structure !VERSION, 8 tags, length=104, data length=100:
   ARCH            STRING    'x86_64'
   OS              STRING    'darwin'
   OS_FAMILY       STRING    'unix'
   OS_NAME         STRING    'Mac OS X'
   RELEASE         STRING    '8.2.2'
   BUILD_DATE      STRING    'Jan 23 2013'
   MEMORY_BITS     INT             64
   FILE_OFFSET_BITS
                   INT             64
```

Subject: Re: TIFF Read/Write Bug?
Posted by Heinz Stege on Fri, 10 May 2013 23:04:38 GMT
View Forum Message <> Reply to Message

On Fri, 10 May 2013 14:29:16 -0700 (PDT), Phillip Bitzer wrote:

> I similar behavior David. Integer array in, integer array out .. unless the color vectors are included. Then it's byte out.
>
Same here. !version={ x86 Win32 Windows Microsoft Windows 8.0.1 Oct 5 2010 32 64}.

However, I wonder if it makes any sense to write integer arrays in TIFF files with color palettes. I don't know the TIFF standard. But the IDL manual for the Red, Green and Blue keywords says: "If you are writing a Palette color image, set these keywords equal to the color table vectors, scaled from 0 to 255."

My understanding is, that a TIFF file written by
```
  Write_Tiff, outFilename, Reverse(ageImage,2), GEOTIFF=geo, $
    /SHORT, /SIGNED, RED=r, GREEN=g, BLUE=b
```
has to be read by
```
  testImage = Read_Tiff(outfilename,r,g,b)
```
And the pixel color values can be calculated by
```
  red=r[testImage]
  green=g[testImage]
  blue=b[testImage]
```

r, g and b have 256 elements maximum. So it does not make any sense for testImage to be integer instead of byte.

Cheers, Heinz

Subject: Re: TIFF Read/Write Bug?
Posted by tom.grydeland on Sun, 12 May 2013 11:03:51 GMT
View Forum Message <> Reply to Message

On Saturday, May 11, 2013 1:04:38 AM UTC+2, Heinz Stege wrote:

> I don't know the TIFF standard.

The TIFF standard is here:

http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf

The description of palette images is in section 5. For palette images, the number of bits per sample can be 4 or 8, corresponding to 16 or 256 colors (the color values are short integers,

however, giving 65536 levels for each of red, blue and green at each index). For more colors than that, you must go to one of the RGB full color modes.

> [...] So it does not make any sense
> for testImage to be integer instead of byte.

Exactly.

> Cheers, Heinz

--T

Subject: Re: TIFF Read/Write Bug?
Posted by David Fanning on Sun, 12 May 2013 14:08:43 GMT
View Forum Message <> Reply to Message

Tom Grydeland writes:

> The description of palette images is in section 5. For palette images, the number of bits per sample can be 4 or 8, corresponding to 16 or 256 colors (the color values are short integers, however, giving 65536 levels for each of red, blue and green at each index). For more colors than that, you must go to one of the RGB full color modes.

Yes, I think my mistake was treating TIFF images as if they were netCDF or HDF files. I had classified data in the byte range, but I wanted to store "missing" data with the value -999, which was why I was using an integer array. At the same time, I didn't want to lose the color vectors that "explained" the data.

Of course, when you store "color" images in TIFF files, it is difficult, if not impossible, to retrieve the science information. What I should have used, of course, is a netCDF file where I can store the palette information separately from the data itself. Unfortunately, netCDF files are just much more difficult for my client to handle as she likes.

There are ways to work around the "problem", but some documenation that the problem exists, in a location where I was likely to read it, would have helped. :-)

Cheers,

David

--
David Fanning, Ph.D.