Subject: Issue loading TIFF images with plugin/TIF
Posted by Tom1 on Fri, 13 Jan 2023 16:44:03 GMT
View Forum Message <> Reply to Message

Hi Mirek,

I get a crash when loading certain TIFF -images with plugin/TIF. (I'm testing with U++ examples/ImageView.) Debugger takes me to gtTileContig() in tif_getimage.c:

```
    if (flip & FLIP_HORIZONTALLY) {
     uint32 line;

     for (line = 0; line < h; line++) {
      uint32 *left = raster + (line * w);
      uint32 *right = left + w - 1;

      while ( left < right ) {
       uint32 temp = *left;
       *left = *right;
       *right = temp;
       left++;
    right--;
      }
     }
    }
```

The problem is that 'raster' is a null pointer already when TIFFRGBAImageGet() is called from tifupp.cpp.

```
bool res = TIFFRGBAImageGet(this, 0, width, height);
```

Crashing depends on the filling order of the image and associated flipping thereof.

I have attached a sample set of GeoTIFF images (downloaded from the internet many years ago) that are readable if I comment out the flipping code. However, they obviously load in whatever orientation they like.

Could you please look at this issue, clear the crash and hopefully also solve the problem with orientation? The way 'TIFFRGBAImageGet' is used here is a bit too fascinating for me to handle.

BTW: I must say that I was both surprised and very pleased when I discovered that plugin/TIF internally supported GeoTIFF tags in such a clean and elegant manner. I'm hoping to replace my old GeoTIFF reader with plugin/TIF now.

Thanks and best regards,

Tom

EDIT: Please note that the attached data set also includes files with name tp_image*.tif using some old proprietary Intergraph tag 33918 and associated packets. I'm not expecting to support those old non-standard parts of intergraph files. However, it would be nice to be able to deal with

the TIFFTAG_GEOTIEPOINTS tag even without TIFFTAG_GEOPIXELSCALE using the associated mappings.

Additionally it would be useful to store the data in GeoKeys as metadata. I mean something like attr.GetAdd("ProjectedCSTypeGeoKey") = Value((int)epsgcode); and equivalent for all found GeoKeys. This would enable extending handling of specific files without further changes to plugin/TIF.

File Attachments

1) GeoTIFF_Samples_from_zi_imaging.zip, downloaded 357 times

Subject: Re: Issue loading TIFF images with plugin/TIF
Posted by Tom1 on Mon, 16 Jan 2023 16:42:34 GMT
View Forum Message <> Reply to Message

Hi Mirek,

Please find attached a small extension to plugin/tif/tifupp.cpp. This one supports geotiepoints without scale by mapping a transformation using Xform2D::Map().

Additionally, handling the not so common fill ordering of those sample geotiffs gains help by adding field 'orientation' to page info. (This way I can externally flip the image after reading it in with plugin/tif.) I'm not sure if it should be internally flipped in plugin/tif and just returned in a sensible orientation (i.e. orientation = 1) as any other image. What's your opinion on this?

The horizontal flip crash in tif_getimage remains unsolved.

Best regards,

Tom

EDIT: Removed tifupp.cpp in favor of new version below...

Subject: Re: Issue loading TIFF images with plugin/TIF
Posted by Tom1 on Tue, 17 Jan 2023 09:56:18 GMT
View Forum Message <> Reply to Message

Hi,

There is another issue reading the following TIFF image with plugin/TIF using e.g. examples/ImageView:

https://www.marinha.mil.br/chm/sites/www.marinha.mil.br.chm/ files/geotiff/1511geotiff.zip

A large area in the center of the image gets black. (E.g. Windows built-in photo viewer shows the

image correctly.)

This can be fixed in tifupp.cpp FetchPage():

```
...
bool TIFRaster::Data::FetchPage()
{
 if(page_error)
  return false;
 if(page_fetched)
  return true;

 cache_size = 0;
 rows.Clear();
 int64 bytes = row_bytes * (int64)height;
/* if(bytes >= 1 << 28) {      // << Commenting this out fixes the issue with 1511geotiff.tif
  tmpfile = GetTempFileName();
  if(!filebuffer.Open(tmpfile, FileStream::CREATE)) {
   page_error = true;
   return false;
  }
  filebuffer.SetSize(bytes);
  if(filebuffer.IsError()) {
   filebuffer.Close();
   FileDelete(tmpfile);
   page_error = true;
   return false;
  }
  rows.Alloc(size.cy);
 }
 else
*/  imagebuf.SetCount(size.cy * row_bytes, 0);
...
```

However, I think there is (or has been) some reason to use tmpfile for big images...

Best regards,

Tom

Subject: Re: Issue loading TIFF images with plugin/TIF
Posted by Tom1 on Wed, 18 Jan 2023 10:12:49 GMT

View Forum Message <> Reply to Message

Hi,

Here's my latest tifupp.cpp. I have further fine tuned the GeoTIFF support.

- All meta data available through GeoKeys are now saved and available through GetMetaData() using key "GK*", e.g. "GK3072".
- TIFFTAG_ORIENTATION is stored as metadata (Key "tiff_orientation") for external normalization of orientation.
- The pixel_is_area == true is now default, as some rasters omit this GeoKey and normally behave as such. Only files including GeoKey for 'pixel_is_point' are treated as such and normalized to pixel_is_area style.
- TIFFTAG_GEOTIEPOINTS can now be accepted without TIFFTAG_GEOPIXELSCALE and geo_matrix can be solved from three such points
- TIFFTAG_GDAL_NODATA is now detected and added to metadata with key "nodata".
- FetchPage() tmpfile feature is disabled (to fix display of e.g. 1511geotiff.tif below)

Mirek, could you consider merging these changes?

I ended up normalizing the orientation in my program while reading the data scanline-by-scanline. This saves memory when processing extremely large GeoTIFFs.

Best regards,

Tom

EDIT: There is still something wrong here with the geo_matrix when generated from TIFFTAG_GEOTIEPOINTS without TIFFTAG_GEOPIXELSCALE, i.e. the tp_* files. (Some TIFF -orientations do not work right yet.) I will keep working on this...

EDIT2: OK, the geo_matrix mapping for all combinations seem to work correctly now. (Note that tp_* files with multiple geotiepoints are indeed rotated and should display exactly on top of each other.)

File Attachments

1) tifupp.cpp, downloaded 349 times

Subject: Re: Issue loading TIFF images with plugin/TIF
Posted by Tom1 on Thu, 19 Jan 2023 10:55:19 GMT
View Forum Message <> Reply to Message

Hi,

I noticed that the geographic and projected CRS epsg code gets bundled into a single meta data variable "epsg". Some files contain both codes and in such case it is better to make them separately available in meta data. (If we lose projected CRS code when geographic CRS code overwrites it, there is no way to correctly georeference the raster as projected CRS is not known.)

Best regards,

Tom

Subject: Re: Issue loading TIFF images with plugin/TIF
Posted by Tom1 on Thu, 19 Jan 2023 19:29:52 GMT
View Forum Message <> Reply to Message

Hi,

Another crash issue. Loading the following file to examples/ImageView crashes (in plugin/TIF) on Windows. It fails in some setjmp code creating the decompressor, which I do not understand.

Here's the JPEG compressed TIFF file:

https://www.marinha.mil.br/chm/sites/www.marinha.mil.br.chm/ files/geotiff/22700geotiff.zip

On Linux loading this file does not cause a crash.

Best regards,

Tom

Subject: Re: Issue loading TIFF images with plugin/TIF
Posted by Tom1 on Tue, 24 Jan 2023 13:04:19 GMT
View Forum Message <> Reply to Message

Hi,

I have now implemented those GeoTIFF features I presented above for tifupp.cpp externally, calling libtiff of plugin/tif directly. I also avoided the FetchPage issue (black area with large images) and horizontal mirroring crash with strangely oriented images by using this approach. Therefore, I do not need merging of those new features to plugin/tif anymore. However, someone else might find them useful, so please proceed as you prefer.

The crash when loading JPEG compressed TIFF images is still there with Windows (not Linux though). So, I guess it is a bug in libtiff, not plugin/tif. Anyway, this requires further digging. The sample file is:

https://www.marinha.mil.br/chm/sites/www.marinha.mil.br.chm/ files/geotiff/22700geotiff.zip

I think that updating the libtiff to its latest version (4.5.0 at this time) might be useful as there is a continuous bug fixing and development going on.

Also, it might be useful to add support for COMPRESS_ZSTD and COMPRESS_LZMA, if possible through using existing plugin/zstd and plugin/lzma. (COMPRESS_LZW and COMPRESS_DEFLATE work well as is.)

Best regards,

Tom

EDIT: So, the above JPEG compressed TIFF image loads correctly in Linux 64 bit. However it also loads correctly in Windows when ImageViewer is built with CLANG (32-bit) or MSBT22 (32-bit), but crashes in __intrinsic_setjmp() with both CLANG64 and MSBT22x64. Don't know how to fix this.

Subject: Re: Issue loading TIFF images with plugin/TIF
Posted by mirek on Mon, 13 Feb 2023 23:46:24 GMT
View Forum Message <> Reply to Message

Hi,

jpeg crash should be now fixed. I am not sure about geo updates, is the file in the first post the most recent? Should I merge that?

Mirek

Subject: Re: Issue loading TIFF images with plugin/TIF
Posted by Tom1 on Tue, 14 Feb 2023 08:02:56 GMT
View Forum Message <> Reply to Message

Hi Mirek,

Thanks, jpeg-tiff works now.

As for your question of merging my latest posted tifupp.cpp, I mostly think you could, but please consider the following:

1. The issue with:

https://www.marinha.mil.br/chm/sites/www.marinha.mil.br.chm/ files/geotiff/1511geotiff.zip

is simply solved by entirely disabling the tmpfile in FetchPage(). I bet you would come up with a more advanced solution yourself.

2. The crash with some of the GeoTIFF_Samples_from_zi_imaging, e.g. image1.tif, due to image flipping remains unsolved:

https://www.ultimatepp.org/forums/index.php?t=getfile&id =6753&

These orientation related issues need to be solved in one way or another. When going with libtiff directly, I ended up requesting the image in its stored orientation stated by TIFFTAG_ORIENTATION. This way I could avoid the internal flipping code which caused crashes. (Then I flipped it afterwards to a natural U++ Image orientation and corrected the georeferencing for the new orientation.) It is important to note that even if the image would be

flipped inside the plugin to U++ Image orientation, the original orientation is still important to know in order to fix the georeferencing information. In my opinion, optimally the plugin/TIF could return the images readily flipped to U++ Image orientation and maybe also with a georeferencing solution corrected accordingly.

Best regards,

Tom

EDIT: I looked at our earlier work on plugin/JPG EXIF orientation and flipping: Yes, the image is pre-flipped and returned in its U++ orientation. So it would seem logical to do the same in plugin/TIF.

EDIT2: I came up with an idea to avoid the flipping crash. This involves further changes to tifupp.cpp, so please stand-by until I finish with this one.

EDIT3: OK, now I have prevented libtiff flipping crash and enabled correct Raster flipping, i.e. the same code that flips JPEG images. Just need to adjust the geomatrix to reflect this change... I will return.

Subject: Re: Issue loading TIFF images with plugin/TIF
Posted by mirek on Tue, 14 Feb 2023 14:28:51 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Tue, 14 February 2023 09:02Hi Mirek,

Thanks, jpeg-tiff works now.

As for your question of merging my latest posted tifupp.cpp, I mostly think you could, but please consider the following:

1. The issue with:

https://www.marinha.mil.br/chm/sites/www.marinha.mil.br.chm/ files/geotiff/1511geotiff.zip

is simply solved by entirely disabling the tmpfile in FetchPage(). I bet you would come up with a more advanced solution yourself.

Well, after some deliberation, I have completely removed tmpfile. After all, we have swap, right? (apart from 64GB RAM).

Quote:
EDIT: I looked at our earlier work on plugin/JPG EXIF orientation and flipping: Yes, the image is pre-flipped and returned in its U++ orientation. So it would seem logical to do the same in plugin/TIF.

EDIT2: I came up with an idea to avoid the flipping crash. This involves further changes to tifupp.cpp, so please stand-by until I finish with this one.

EDIT3: OK, now I have prevented libtiff flipping crash and enabled correct Raster flipping, i.e. the same code that flips JPEG images. Just need to adjust the geomatrix to reflect this change... I will return.
[/quote]

Waiting then

Mirek

Subject: Re: Issue loading TIFF images with plugin/TIF
Posted by Tom1 on Tue, 14 Feb 2023 19:54:28 GMT
View Forum Message <> Reply to Message

Hi,

OK, finally geomatrix is done.

Please test and merge. (This one is derived from the early morning version, so it may be on a bit different base, if you already ditched the tmpfile and pushed to trunk later today...)

Best regards,

Tom

File Attachments
1) tifupp.cpp, downloaded 345 times

Subject: Re: Issue loading TIFF images with plugin/TIF
Posted by Tom1 on Wed, 15 Feb 2023 06:56:42 GMT
View Forum Message <> Reply to Message

Hi Mirek,

Thanks for merging. Unfortunately two necessary orientation related details were dropped while merging:

```
bool TIFRaster::Data::Create()
{
 TIFFSetErrorHandler(NULL);
 TIFFSetErrorHandlerExt(NULL);
 TIFFSetWarningHandler(NULL);
```

```
 TIFFSetWarningHandlerExt(NULL);

 tiff = TIFFStreamOpen("tiff@" + Format64((intptr_t)this), "r", &stream);
 if(!tiff)
  return false;

 int count = TIFFNumberOfDirectories(tiff);
 if(count <= 0)
  return false;
 for(int i = 0; i < count; i++) {
  Page& page = pages.Add();
  TIFFSetDirectory(tiff, i);
  TIFFGetField(tiff, TIFFTAG_IMAGEWIDTH, &page.width);
  TIFFGetField(tiff, TIFFTAG_IMAGELENGTH, &page.height);
  float xres, yres;
  TIFFGetFieldDefaulted(tiff, TIFFTAG_XRESOLUTION, &xres);
  TIFFGetFieldDefaulted(tiff, TIFFTAG_YRESOLUTION, &yres);
  uint16 resunit;
  TIFFGetFieldDefaulted(tiff, TIFFTAG_RESOLUTIONUNIT, &resunit);
  TIFFGetFieldDefaulted(tiff, TIFFTAG_BITSPERSAMPLE, &page.bits_per_sample);
  TIFFGetFieldDefaulted(tiff, TIFFTAG_SAMPLESPERPIXEL, &page.samples_per_pixel);
  TIFFGetFieldDefaulted(tiff, TIFFTAG_PHOTOMETRIC, &page.photometric);

  TIFFGetFieldDefaulted(tiff, TIFFTAG_ORIENTATION, &page.orientation); // << Actual
orientation must be picked up if present

  double dots_per_unit = (resunit == RESUNIT_INCH ? 600.0 : resunit ==
RESUNIT_CENTIMETER
   ? 600.0 / 2.54 : 0);
  page.dot_size.cx = (xres ? fround(page.width * dots_per_unit / xres) : 0);
  page.dot_size.cy = (yres ? fround(page.height * dots_per_unit / yres) : 0);


bool TIFRaster::Data::FetchPage()
{
 if(page_error)
  return false;
 if(page_fetched)
  return true;

 cache_size = 0;
 rows.Clear();
 int64 bytes = row_bytes * (int64)height;
 imagebuf.SetCount(size.cy * row_bytes, 0);

 req_orientation = pages[page_index].orientation; // << This is needed in order to get the image in
its natural orientation and avoid crash
```

```
 bool res = TIFFRGBAImageGet(this, 0, width, height);
 TIFFRGBAImageEnd(this);
 page_open = false;

 page_fetched = true;
 return true;
}
```

Best regards,

Tom

Subject: Re: Issue loading TIFF images with plugin/TIF
Posted by mirek on Wed, 15 Feb 2023 07:15:20 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Wed, 15 February 2023 07:56Hi Mirek,

Thanks for merging. Unfortunately two necessary orientation related details were dropped while merging:

```
bool TIFRaster::Data::Create()
{
 TIFFSetErrorHandler(NULL);
 TIFFSetErrorHandlerExt(NULL);
 TIFFSetWarningHandler(NULL);
 TIFFSetWarningHandlerExt(NULL);

 tiff = TIFFStreamOpen("tiff@" + Format64((intptr_t)this), "r", &stream);
 if(!tiff)
  return false;

 int count = TIFFNumberOfDirectories(tiff);
 if(count <= 0)
  return false;
 for(int i = 0; i < count; i++) {
  Page& page = pages.Add();
  TIFFSetDirectory(tiff, i);
  TIFFGetField(tiff, TIFFTAG_IMAGEWIDTH, &page.width);
  TIFFGetField(tiff, TIFFTAG_IMAGELENGTH, &page.height);
  float xres, yres;
  TIFFGetFieldDefaulted(tiff, TIFFTAG_XRESOLUTION, &xres);
  TIFFGetFieldDefaulted(tiff, TIFFTAG_YRESOLUTION, &yres);
  uint16 resunit;
  TIFFGetFieldDefaulted(tiff, TIFFTAG_RESOLUTIONUNIT, &resunit);
  TIFFGetFieldDefaulted(tiff, TIFFTAG_BITSPERSAMPLE, &page.bits_per_sample);
  TIFFGetFieldDefaulted(tiff, TIFFTAG_SAMPLESPERPIXEL, &page.samples_per_pixel);
```

```
  TIFFGetFieldDefaulted(tiff, TIFFTAG_PHOTOMETRIC, &page.photometric);

  TIFFGetFieldDefaulted(tiff, TIFFTAG_ORIENTATION, &page.orientation); // << Actual
orientation must be picked up if present

  double dots_per_unit = (resunit == RESUNIT_INCH ? 600.0 : resunit ==
RESUNIT_CENTIMETER
   ? 600.0 / 2.54 : 0);
  page.dot_size.cx = (xres ? fround(page.width * dots_per_unit / xres) : 0);
  page.dot_size.cy = (yres ? fround(page.height * dots_per_unit / yres) : 0);


bool TIFRaster::Data::FetchPage()
{
 if(page_error)
  return false;
 if(page_fetched)
  return true;

 cache_size = 0;
 rows.Clear();
 int64 bytes = row_bytes * (int64)height;
 imagebuf.SetCount(size.cy * row_bytes, 0);

 req_orientation = pages[page_index].orientation; // << This is needed in order to get the image in
its natural orientation and avoid crash

 bool res = TIFFRGBAImageGet(this, 0, width, height);
 TIFFRGBAImageEnd(this);
 page_open = false;

 page_fetched = true;
 return true;
}
```

Best regards,

Tom

Uhm, sorry, I was cherrypicking...

What happened to

```
  page.orientation = 1; // Default if no tag
```

attr.GetAdd("tiff_orientation") = Value((int)page.orientation);

?

Mirek

Subject: Re: Issue loading TIFF images with plugin/TIF
Posted by Tom1 on Wed, 15 Feb 2023 07:26:32 GMT
View Forum Message <> Reply to Message

Hi,

Updated: I do not need "tiff_orientation" metadata anymore as image is now returned in its U++ orientation and the geomatrix is readily adjusted accordingly. Therefore, this can be dropped:

attr.GetAdd("tiff_orientation") = Value((int)page.orientation);

TIFFGetFieldDefaulted sets page.orientation correctly to its default value when TIFFTAG_ORIENTATION is missing from TIFF file. Therefore, this can be dropped:
page.orientation = 1; // Default if no tag

BR,

Tom

Subject: Re: Issue loading TIFF images with plugin/TIF
Posted by Tom1 on Thu, 16 Feb 2023 16:29:19 GMT
View Forum Message <> Reply to Message

Hi Mirek,

I noticed you have pushed fixes. There is one small issue.

Please change this:
TIFFGetField(tiff, TIFFTAG_ORIENTATION, &page.orientation);
to this:
TIFFGetFieldDefaulted(tiff, TIFFTAG_ORIENTATION, &page.orientation);
Otherwise the default case will not be correct and results in an upside down image.

Thanks and best regards,

Tom

Subject: Re: Issue loading TIFF images with plugin/TIF
Posted by mirek on Fri, 17 Feb 2023 10:04:43 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Thu, 16 February 2023 17:29Hi Mirek,

I noticed you have pushed fixes. There is one small issue.

Please change this:
TIFFGetField(tiff, TIFFTAG_ORIENTATION, &page.orientation);
to this:
TIFFGetFieldDefaulted(tiff, TIFFTAG_ORIENTATION, &page.orientation);
Otherwise the default case will not be correct and results in an upside down image.

Thanks and best regards,

Tom

All fine now?

(Next time, we should follow a good practice of PR.... Would save us a lot of trouble...)

Subject: Re: Issue loading TIFF images with plugin/TIF
Posted by Tom1 on Fri, 17 Feb 2023 15:37:07 GMT
View Forum Message <> Reply to Message

Hi Mirek,

Thanks, it's all good now!

As for the future, where can I find step by step instructions to do it the right way? I'm really bad with git, sorry about that. (Luckily TheIDE handles the uppsrc updates for me

Best regards,

Tom