Subject: JPEG Images do not get rotated correctly
Posted by Tom1 on Mon, 01 Oct 2018 09:01:10 GMT
View Forum Message <> Reply to Message

Hi,

It appears that most of my images taken on a smartphone are loaded upside down with plugin/jpg. Windows 10 Explorer and Windows 10 image viewer shows them correctly, but Windows 7 Explorer does not, nor does its default image viewer. I tried to go around this in Windows 7 using ImageView example in U++, but the result was still upside down. Then I read on the net that on smartphones the photos are usually stored with EXIF orientation tag to inform the viewer to show them in correct orientation, however, this does not appear work in plugin/jpg. Would it be possible to fix this in U++ plugin/jpg? There's a sample picture attached portraying the issue... (So this is viewed correctly in Windows 10 and upside down in Windows 7 and also U++ ImageView.)

Best regards,

Tom

File Attachments
1) 20170619T232744.jpg, downloaded 384 times

Subject: Re: JPEG Images do not get rotated correctly
Posted by Zbych on Wed, 03 Oct 2018 17:32:41 GMT
View Forum Message <> Reply to Message

EXIF says that your image was made upside down and it should be rotated afterwards. I guess Upp simply ignores EXIF data.

EXIF
ImageWidth 2592
ImageHeight 1944
Make SAMSUNG
Model SM-G388F
Orientation Rotate 180
XResolution 72
YResolution 72
ResolutionUnit inches

Subject: Re: JPEG Images do not get rotated correctly
Posted by Tom1 on Thu, 04 Oct 2018 06:58:44 GMT
View Forum Message <> Reply to Message

OK, thanks for confirmation. That was kind of expected.

Now, is there anyone out there capable of adding the EXIF rotation support in U++ plugin/jpg?

Best regards,

Tom

---

Subject: Re: JPEG Images do not get rotated correctly
Posted by Oblivion on Thu, 04 Oct 2018 09:28:13 GMT
View Forum Message <> Reply to Message

Hello Tom,

There seems to be support for EXIF rotation in U++, jpeg plugin, but I think its usage is -reasonably, IMO- left to client code.

See plugin/jpg/jpgupp.cpp, ln: 341, 343 (JpegRaster:Data:ExifDir):

```
 if(type == BASE_IFD) {
  if(tag == 0x112)
   metadata.Add("orientation", Exif16(data));
```

Now, I did not test this yet, as I rarely work with jpg images directly, but it should be possible to extract the information using JpegRaster::GetMetaData("orientation").

Edit: Ok, I've tested it and it works with ImageView too:

```
void ImageView::Load(const char *filename)
{
 img.SetImage(Null);
 FileIn in(filename);
 One<StreamRaster> r = StreamRaster::OpenAny(in);
 if(!r)
  return;

 JPGRaster *jpg = dynamic_cast<JPGRaster*>(~r);
 if(jpg) {
  DUMP(jpg->GetMetaData("orientation")); // will return "3" for the image you've provided.
 }

 Size rsz = img.GetSize();
 Size isz = r->GetSize();
 if(isz.cx >= rsz.cx || isz.cy >= rsz.cy) {
  if(isz.cx * rsz.cx < rsz.cy * isz.cy)
```

```
   rsz.cx = isz.cx * rsz.cy / isz.cy;
  else
   rsz.cy = isz.cy * rsz.cx / isz.cx;
  ImageEncoder m;
  Rescale(m, rsz, *r, isz);
  img.SetImage(m);
 }
 else
  img.SetImage(r->GetImage());
}
```

Best regards,
Oblivion

---

Subject: Re: JPEG Images do not get rotated correctly
Posted by Tom1 on Thu, 04 Oct 2018 12:03:14 GMT
View Forum Message <> Reply to Message

Hi Oblivion,

Well that was fast, thanks!

It would surely be handy to have StreamRaster::GetImage() or maybe a non-existing JPGRaster::GetImage() to rotate/normalize the Image to its intended orientation on-the-fly before returning it.

Thanks and best regards,

Tom

---

Subject: Re: JPEG Images do not get rotated correctly
Posted by Tom1 on Thu, 04 Oct 2018 12:33:03 GMT
View Forum Message <> Reply to Message

Hi Oblivion,

Based on your findings, here's what needs to be done:

```
...
 Image rimg(r->GetImage());

 JPGRaster *jpg = dynamic_cast<JPGRaster*>(~r);
 if(jpg) {
  switch((int)jpg->GetMetaData("orientation")){
```

```
   // 1 = natural orientation
   case 2:
    rimg=MirrorHorz(rimg);
    break;
   case 3:
    rimg=Rotate180(rimg);
    break;
   case 4:
    rimg=MirrorVert(rimg);
    break;
   case 5:
    rimg=RotateAntiClockwise(MirrorHorz(rimg));
    break;
   case 6:
    rimg=RotateClockwise(rimg);
    break;
   case 7:
    rimg=RotateClockwise(MirrorHorz(rimg));
    break;
   case 8:
    rimg=RotateAntiClockwise(rimg);
    break;
  }
 }
 img.SetImage(rimg);
...
```

But how to embed it in plugin/jpg in the correct place so that it just works quietly in the background?

Best regards,

Tom

---

Subject: Re: JPEG Images do not get rotated correctly
Posted by Oblivion on Thu, 04 Oct 2018 16:13:03 GMT
View Forum Message <> Reply to Message

Hello Tom,

Well, adding this to Raster is not so trivial, as -AFAIK- this EXIF orientation tag is currently jpeg specific and Raster::GetImage is not virtual.

I'd rather add this as a JPGRaster method.

```
Image JPGRaster::GetNormalizedImage()
{
 Value n = GetMetaData("orientation");
 Image img(GetImage());

 if(!IsNull(n))
  switch((int)n){
   case 2:
    img = MirrorHorz(img);
    break;
   case 3:
    img = Rotate180(img);
    break;
   case 4:
    img = MirrorVert(img);
    break;
   case 5:
    img = RotateAntiClockwise(MirrorHorz(img));
    break;
   case 6:
    img = RotateClockwise(img);
    break;
   case 7:
    img = RotateClockwise(MirrorHorz(img));
    break;
   case 8:
    img = RotateAntiClockwise(img);
    break;
   default:
    // Natural orientation
    break;
  }
 return pick(img);
}
```

Of course, it is up to Mirek and/or the developer(s) of the plugin to decide. :)

Best regards,
Oblivion

---

Subject: Re: JPEG Images do not get rotated correctly
Posted by Tom1 on Thu, 04 Oct 2018 17:05:47 GMT

View Forum Message <> Reply to Message

Hi Oblivion,

Well, I hope Mirek picks this up and places it where it belongs. 'Everything belongs somewhere' ... :)

Best regards,

Tom

Subject: Re: JPEG Images do not get rotated correctly
Posted by mirek on Wed, 10 Oct 2018 11:20:47 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Thu, 04 October 2018 19:05Hi Oblivion,

Well, I hope Mirek picks this up and places it where it belongs. 'Everything belongs somewhere' ... :)

Best regards,

Tom

I have noticed and filed in RM. I know exactly what to do with that:

- define orientation constants
- add GetOrientation as standard Raster virtual method
- add "autocorrect" parameter to LoadImageAny with "true" default value

but I think I will need to implement "generic orientation routine":

img = RotateAntiClockwise(MirrorHorz(img));

should be done in single step IMO.

Subject: Re: JPEG Images do not get rotated correctly
Posted by Tom1 on Wed, 10 Oct 2018 11:28:47 GMT
View Forum Message <> Reply to Message

Hi Mirek,

Sounds great! Thanks!

Best regards,

Tom

Subject: Re: JPEG Images do not get rotated correctly
Posted by mirek on Fri, 12 Oct 2018 17:09:57 GMT
View Forum Message <> Reply to Message

Done, please check.

Subject: Re: JPEG Images do not get rotated correctly
Posted by Tom1 on Fri, 12 Oct 2018 18:40:42 GMT
View Forum Message <> Reply to Message

Hi Mirek,

Thanks for taking the challenge!

Unfortunately, there's an issue of mirroring and false rotations on some specific cases. Please check the following 8 (or 16) reference images with ImageView example to easily see the situation for each case. They are available here:

https://github.com/recurser/exif-orientation-examples

(Just googled to find such images, I have no association with them.)

Thanks and best regards,

Tom

Update: Please note that this may be urgent as even default FLIP_NONE is mirrored with left and right swapped.

Subject: Re: JPEG Images do not get rotated correctly
Posted by Tom1 on Fri, 12 Oct 2018 19:21:23 GMT
View Forum Message <> Reply to Message

Hi,

Found it.
In Raster.h it should read:

```
enum FlipMode {
 FLIP_NONE              = 1,
 FLIP_MIRROR_HORZ       = 2,
```

```
 FLIP_ROTATE_180          = 3,
 FLIP_MIRROR_VERT         = 4,
 FLIP_TRANSPOSE           = 5,
 FLIP_ROTATE_CLOCKWISE    = 6,
 FLIP_TRANSVERSE          = 7,
 FLIP_ROTATE_ANTICLOCKWISE = 8,
}
```

And in jpgupp.cpp:

```
Raster::Info JPGRaster::GetInfo()
{
 ASSERT(data);
 Raster::Info info;
 info.kind = IMAGE_OPAQUE;
 if(data->cinfo.output_components == 1) {
  info.bpp = 8;
  info.colors = 256;
 }
 else {
  info.bpp = 24;
  info.colors = 0;
 }
 info.dots = data->dot_size;
 info.hotspot = Null;
 Value v = GetMetaData("orientation");
 if(IsNumber(v))
  info.orientation = clamp((int)v, 1, 8);
 return info;
}
```

I.e. The Exif orientation value runs from 1 to 8 instead of 0 to 7.

Anyway, thanks for this new easier way of dealing with photos!

Best regards,

Tom

---

Subject: Re: JPEG Images do not get rotated correctly
Posted by mirek on Sat, 13 Oct 2018 17:07:17 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Fri, 12 October 2018 21:21Hi,

Found it.
In Raster.h it should read:

```
enum FlipMode {
 FLIP_NONE               = 1,
 FLIP_MIRROR_HORZ        = 2,
 FLIP_ROTATE_180         = 3,
 FLIP_MIRROR_VERT        = 4,
 FLIP_TRANSPOSE          = 5,
 FLIP_ROTATE_CLOCKWISE   = 6,
 FLIP_TRANSVERSE         = 7,
 FLIP_ROTATE_ANTICLOCKWISE = 8,
}
```

And in jpgupp.cpp:

```
Raster::Info JPGRaster::GetInfo()
{
 ASSERT(data);
 Raster::Info info;
 info.kind = IMAGE_OPAQUE;
 if(data->cinfo.output_components == 1) {
  info.bpp = 8;
  info.colors = 256;
 }
 else {
  info.bpp = 24;
  info.colors = 0;
 }
 info.dots = data->dot_size;
 info.hotspot = Null;
 Value v = GetMetaData("orientation");
 if(IsNumber(v))
  info.orientation = clamp((int)v, 1, 8);
 return info;
}
```

I.e. The Exif orientation value runs from 1 to 8 instead of 0 to 7.

Anyway, thanks for this new easier way of dealing with photos!

Best regards,

Tom

Ops. FlipMode constants are as I wanted them (I think it is a good idea that 0 is "do nothing"), but

I have forgot to subtract 1 in jpg. Fix commited.

---

Subject: Re: JPEG Images do not get rotated correctly
Posted by Tom1 on Sat, 13 Oct 2018 20:33:55 GMT
View Forum Message <> Reply to Message

Hi,

It's all good now!

I agree zero based enumeration with zero meaning NOP is more logical and, therefore, better.

Thanks and best regards,

Tom

---

Subject: Re: JPEG Images do not get rotated correctly
Posted by Tom1 on Sun, 21 Oct 2018 14:20:14 GMT
View Forum Message <> Reply to Message

Hi Mirek,

Could you possibly add two lines of code in jpgupp.cpp in order to support Exif "DateTime" -tag:

```
int JPGRaster::Data::ExifDir(const char *begin, int offset, IFD_TYPE type)
{
 const char *e = begin + offset;
 int nitems = Exif16(e);
// puts(NFormat("directory %08x: %d items", dir, nitems));
 e += 2;
 for(int i = 0; i < nitems; i++, e += 12) {
  int tag = Exif16(e);
  int fmt = Exif16(e + 2);
  int count = Exif32(e + 4);
  static const int fmtlen[] = {
   1, 1, 2, 4, 8, 1, 1, 2, 4, 8, 4, 8
  };
  int len = 0;
  if(fmt > 0 && fmt <= __countof(fmtlen))
   len = fmtlen[fmt - 1] * count;
  const char *data = e + 8;
  if(len > 4)
   data = begin + Exif32(data);
//  puts(NFormat("[%d]: tag %04x fmt %d, count %d, data %s",
```

```
//   i, tag, fmt, count, BinHexEncode(data, data + len)));
  if(type == BASE_IFD) {
   if(tag == 0x112)
    metadata.Add("orientation", Exif16(data));
   if(tag == 0x132)
    metadata.Add("DateTime", String(data, 20));
   if(tag == 0x8825) {
    int offset = Exif32(data);
 //   puts(NFormat("GPS IFD at %08x", offset));
    ExifDir(begin, offset, GPS_IFD);
   }
  }
  else if(type == GPS_IFD) {
   if((tag == 2 || tag == 4) && fmt == EXIF_RATIONAL && count == 3) {
    metadata.Add(tag == 2 ? "GPSLatitude" : "GPSLongitude",
     ExifF5(data + 0) + ExifF5(data + 8) / 60 + ExifF5(data + 16) / 3600);
//    puts(NFormat("GPSLatitude: %n %n %n", n1, n2, n3));
   }
   else if(tag == 6 && fmt == EXIF_RATIONAL && count == 1)
    metadata.Add("GPSAltitude", ExifF5(data));
   else if(tag == 16 && fmt == EXIF_ASCII && count == 2 && *data)
    metadata.Add("GPSImgDirectionRef", String(*data, 1));
   else if(tag == 17 && fmt == EXIF_RATIONAL && count == 1)
    metadata.Add("GPSImgDirection", ExifF5(data + 0));
  }
 }
 int nextoff = Exif32(e);
// puts(NFormat("next offset = %08x", nextoff));
 return nextoff;
}
```

More specifically those reading:

```
   if(tag == 0x132)
    metadata.Add("DateTime", String(data, 20));
```

This would be great help!

Best regards,

Tom

---

Subject: Re: JPEG Images do not get rotated correctly
Posted by mirek on Wed, 31 Oct 2018 10:12:02 GMT
View Forum Message <> Reply to Message

Sorry for late reply, applied, thanks, makes sense.

Subject: Re: JPEG Images do not get rotated correctly
Posted by Tom1 on Wed, 31 Oct 2018 18:30:13 GMT
View Forum Message <> Reply to Message

Hi Mirek,

Thanks for adding that part. And no worries, this is merely a no-rush 'hobby' project of mine, which attempts to rename pictures by their shooting date and time, in order to efficiently remove duplicates from my over ten thousand photos and to make it easier to group them by the event or context.

As it turned out later, adding support for DateTime tag covered most cases but unfortunately not quite all. Then I added a couple more to get better coverage:

```
 else if(tag == 0x132)
  metadata.Add("DateTime", String(data, 20));
 else if(tag == 0x9003)
  metadata.Add("DateTimeOriginal", String(data, 20));
 else if(tag == 0x9004)
  metadata.Add("DateTimeDigitized", String(data, 20));
```

However, after processing all my digital photos, I found that there are still some pictures with erroneous time stamps because they have been processed (e.g. rotated) with Windows picture viewer some time over the past years. The only 'tag' holding the desired time and date of the picture for those pictures is Windows originated tag 'Date taken', which is not an Exif tag. I have not yet figured out where and how to dig up that information... (Windows Explorer can show it as a property for the picture though.) When that gets solved, I would sure like to have support for that added too for the JPG plugin. Anyway, I have still to figure out the way to pick up Windows 'Date taken' tag.

Thanks and best regards,

Tom

Subject: Re: JPEG Images do not get rotated correctly
Posted by mirek on Thu, 01 Nov 2018 07:34:17 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Wed, 31 October 2018 19:30Hi Mirek,

Thanks for adding that part. And no worries, this is merely a no-rush 'hobby' project of mine, which

attempts to rename pictures by their shooting date and time, in order to efficiently remove duplicates from my over ten thousand photos and to make it easier to group them by the event or context.

Funny, I some time ago I was working on the similar project... (but not finished it) :)

Quote:
```
 else if(tag == 0x132)
  metadata.Add("DateTime", String(data, 20));
 else if(tag == 0x9003)
  metadata.Add("DateTimeOriginal", String(data, 20));
 else if(tag == 0x9004)
  metadata.Add("DateTimeDigitized", String(data, 20));
```

I guess there is no harm adding these too.

That said, now looking at these, maybe we should actually store Time instead of String?

Mirek

---

Subject: Re: JPEG Images do not get rotated correctly
Posted by Tom1 on Thu, 01 Nov 2018 08:57:40 GMT
View Forum Message <> Reply to Message

Hi Mirek,

Time would be just fine instead of those Strings. :)

The rest is just about the case that started this all:

The original reason to get into this was my bad habit of taking multiple backups of my photos at different times to different HDDs or USB flash drives and over the years I had ended up with two to four copies of each image with partially different names and some rotated or cropped changing the file contents and possibly DateTime*** tags. I wanted to have a single copy of each image on my NAS. (Well, yes I do keep a backup of that NAS. :)  )

As for my code, I first renamed files based on their shooting time stamp, or more precisely, a prioritized selection of their various DateTime*** Exif tags with file modification time as a backup. I renamed files instead of removing assumed duplicates in order to get groups for assumed-to-be-same photos. Then I removed the duplicates if file sizes and binary comparison of the files in group mathced.

Later I have found that some time stamps are just wrong. Interestingly Windows Explorer can still

show their correct 'Date taken' info as a property... In order to get event grouping working properly, I would still somehow need to extract the 'Microsoft Date taken' for the photos that do not have correct Exif DateTime*** tags.

If you like, I can post some code here later, but it is really simple anyway.

Thanks and best regards,

Tom

---