JPEG Image Orientation and Exif

The other day I came across a strange bug during work. My colleagues gave me some photographs taken with a smartphone. On their machines (Windows 7), all the photos were shown correctly in landscape mode. However, when I checked those photos, I found that some of those photos were shown in portrait mode (rotated 90 or 270 degrees) or upside-down (rotated 180 degrees).

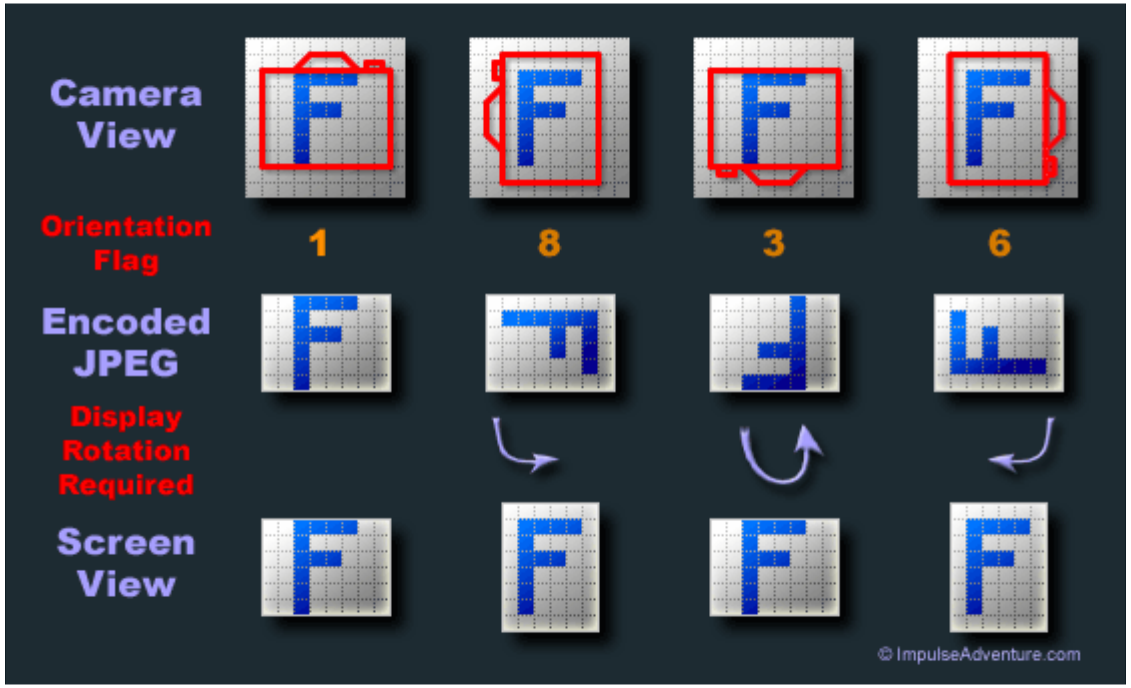
I was curious what happened and learned about Exif and all its related stuff.

How Does Exif Orientation Work

What is Exif

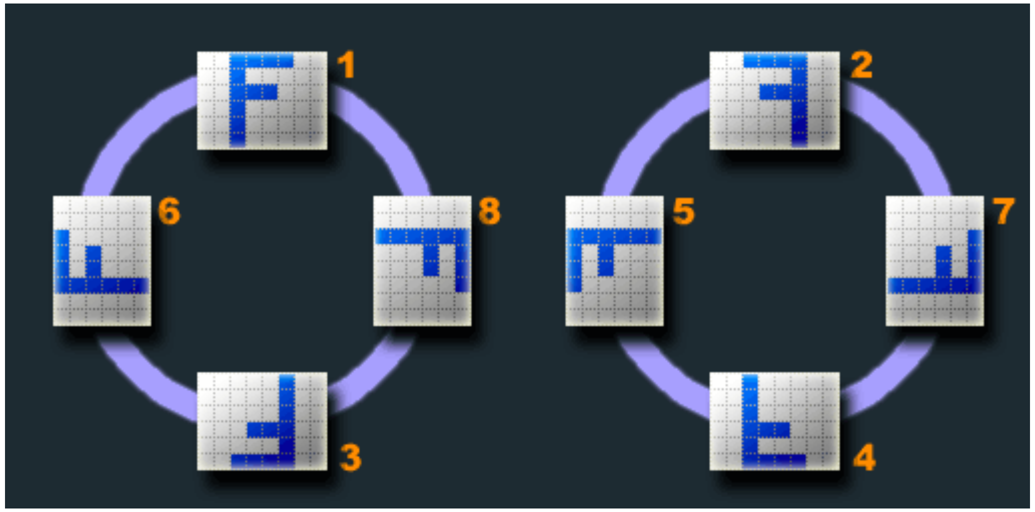
Exif (Exchangeable image file format) is a protocol to store various meta-information about the images taken by digital cameras. Exif are stored along with the actual image data. Some of the meta info in Exif includes camera maker, shutter speed, focal length, orientation, shooting time etc. These meta info is called tags and each tag has a specific tag number decided by the Exif format standard. A comprehensive list of tags and their related informations can be found here.

camera, we may not always hold the camera in such a position that the camera top corresponds to top of the scene. Image below from this blog illustrates this idea clearly:



But no matter how you are holding the camera, if you check the image on your computer, the image is shown in correct orientation. This has something to do with Exif orientation flag. When you are holding the camera with non-upright position, the *raw photo* you take is stored as a rotated image. The digital device (be a smartphone or digital camera) has a sensor to record the orientation of the camera and writes that information into orientation flag in Exif.

The Exif orientation flag can have nine different values from 1 to 9[1]. The image[2] below shows eight of them:



Typically, you will only get flag 1, 8, 3, 6 for digital photos. Flag 2, 7, 4, 5 represent mirrored and rotated version of images.

Why Are My Images Wrongly Displayed?

When you use a photo viewer to check the image on your computer, if the photo viewer can read the Exif info and respects that info, it will automatically rotate the raw image based on the orientation info. The end result is that you can see the correctly-oriented photo no matter how it is actually stored.

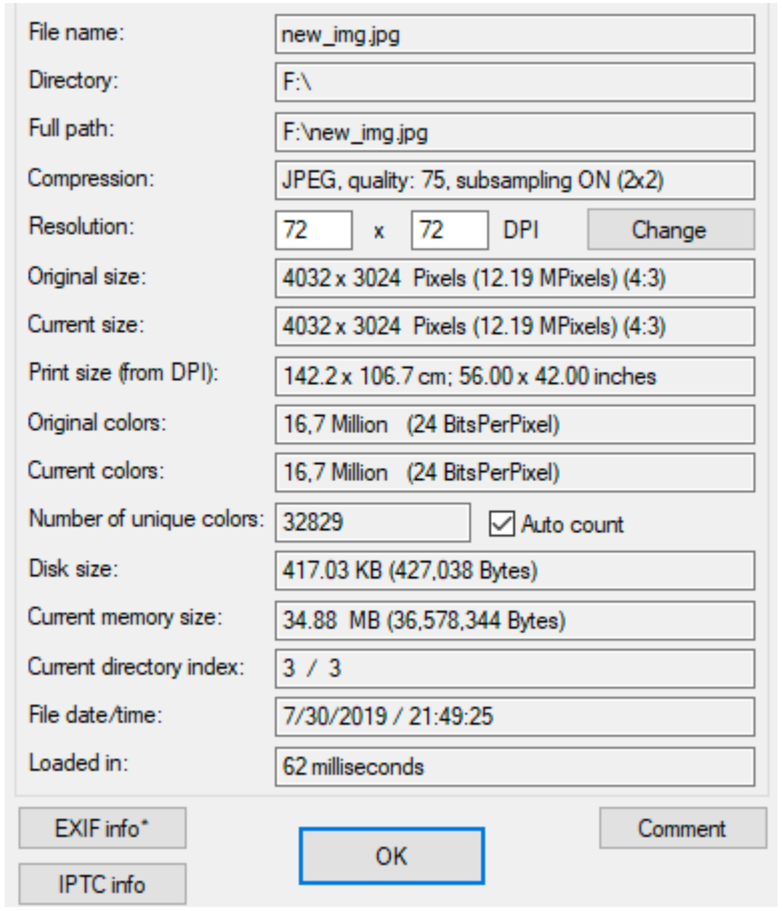
Now come to the issue in the beginning of this post. According to this article, Windows systems before Windows 8 does not take into account the Exif orientation flag and show the images as is, i.e., the raw, un-rotated images are shown instead of the properly rotated images. Since the images my colleagues gave me are shown correctly on their Windows 7 machine, we can draw a conclusion that the raw images are in correct orientation. Somehow, the smartphone is reporting the wrong orientation flags for some of the photos. When I show these photos on my Windows 10 machine, since Windows 10 respects the orientation flag, some of the images will be shown as rotated due to the false orientation flag.

On the other hand, if you see correct photo on Windows 10 but rotated one on a Windows 7 machine, that is because the raw image is in rotated position and Windows 7 does not respect the orientation info in Exif.

Read and Write Exif Info

IrfanView

IrfanView is a great image viewer on Windows, which respects the image Exif info. To view the image Exif info, open an image and click `Image -> Information`. If the image contains Exif info, you can then click the `EXIF info` button at the bottom left of the popup window to check the image Exif info.



File name:	new_img.jpg
Directory:	F:\
Full path:	F:\new_img.jpg
Compression:	JPEG, quality: 75, subsampling ON (2x2)
Resolution:	72 x 72 DPI
Original size:	4032 x 3024 Pixels (12.19 MPixels) (4:3)
Current size:	4032 x 3024 Pixels (12.19 MPixels) (4:3)
Print size (from DPI):	142.2 x 106.7 cm; 56.00 x 42.00 inches
Original colors:	16,7 Million (24 BitsPerPixel)
Current colors:	16,7 Million (24 BitsPerPixel)
Number of unique colors:	32829 ☑ Auto count
Disk size:	417.03 KB (427,038 Bytes)
Current memory size:	34.88 MB (36,578,344 Bytes)
Current directory index:	3 / 3
File date/time:	7/30/2019 / 21:49:25
Loaded in:	62 milliseconds

Disable Auto-rotate for IrfanView

By default, IrfanView respects the Exif info and will auto-rotate the image based on its orientation flag. To disable this behavior, go to `Options -> Properties/Settings` , click `JPG/PCD/GIF` and uncheck the box `Auto-rotate image according to EXIF info (if available)` .

Pillow and Exif Info

If you read an image with Pillow and show it or save it again, Pillow will not respect the Exif orientation tag. You may see such issues here and here. There is a pull request to address this issue, which has been merged now.

Pillow is able to read the image Exif info, but it is not able to edit the Exif info. A sample script to show the image Exif info is shown below:

```
from PIL import Image
from PIL. from PIL.ExifTags import TAGS

img = Image.open('test.jpg')

exif = img.getexif()

for k, v in exif.items():
    print('{}: {}'.format(TAGS[k], v))
```

In the above script, We use the `Image.getexif()` [3] to retrieve the image Exif info. `TAGS` is a dictionary mapping the tag number to a descriptive name.

Piexif

The Python package piexif can be used to both read and write the image Exif info.

Based on the example on its documentation site, I show an example of change the image Exif orientation flag and save a new image using the new Exif info.

```
from PIL import Image
import piexif

img = Image.open('test.jpg')
if "exif" in img.info:
    exif_dict = piexif.load(img.info['exif'])

if piexif.ImageIFD.Orientation in exif_dict['0th']:
    exif_dict['0th'][pixeif.ImageIFD.Orientation] = 3

    # quick and dirty work around to avoid type error
    exif_dict['Exif'][41729] = b'1'

    exif_bytes = piexif.dump(exif_dict)

img.save('new_img.jpg', exif=exif_bytes)
```

The primary info of Exif are stored in the `0th` key of `exif_dict`, which is also a Python dictionary. It seems that piexif did not check the value type of the Exif dict so that we may encounter ValueError when we try to dump the exif_dict. After reading the source code of piexif, I use the following line of code to work around this issue just for now.

```
exif_dict['Exif'][41729] = b'1'
```

After that, you should be able to dump the `exif_dict` without errors.

References

- An online service which reads and writes Exif info
- Windows 10 auto-rotate images
- Why images do not always appear correctly rotated
- Disable auto-rotate in IrfanView
- Derotate image with exiftool
- JPEG rotation and EXIF orientation
- How does smartphone decide orientation
- JPEG Orientation
- Rotate image based on Exif orientation in Pillow
- Can not dump exif dict as bytes
- Read image exif with Pillow
- Exif orientation tag

---

1. Flag 9 means *undefined.* ↩

2. Image source: https://www.daveperrett.com/articles/2012/07/28/exif-orientation-handling-is-a-ghetto/ ↩

3. This method is added in Pillow 6.0.0. ↩

Related

Display Image with Pillow inside Ubuntu on Windows
16 February 2019 · Updated: 31 March 2020 · 268 words · 2 mins
Note PIL Windows Python X11 WSL Ubuntu

Two Issues Related to ImageFont Module in PIL
4 December 2018 · Updated: 13 February 2020 · 387 words · 1 min
Python PIL Windows Font 字体 Unicode Python

Add A Custom Search Engine for Vimium
18 April 2023 · 280 words · 2 mins
Technique