THETA Media Unofficial Guide

Table of Contents

This document is being created to replace the Unofficial Media Guide [http://theta360.guide/community-document/community.html]. At the time of writing, the THETA S is the newest model.

Post comments and corrections to http://lists.theta360.guide in a public category or direct message @jcasman [http://lists.theta360.guide/users/jcasman/] or @craig [http://lists.theta360.guide/users/codetricity] on the lists system. You must be logged in to send a message.

This is an unofficial, community-generated guide for developing applications that use 360 media from RICOH THETA cameras. This is not authorized by RICOH and is based on publicly available information.

Get the latest news and updates on Twitter **@theta360dev [https://twitter.com/theta360dev]**

Resources

Official Information

- RICOH THETA API v2.1 reference [https://developers.theta360.com/en/docs/v2.1/api_reference/], compliant with Open Spherical Camera API level 2 [https://developers.google.com/streetview/open-spherical-camera/] from Google.
- THETA Developers Official Forum [https://developers.theta360.com/en/forums/]
- Official SDK [https://developers.theta360.com/en/docs/sdk/]
- Windows, Mac, iOS, Android apps [https://theta360.com/en/support/download/]

Unofficial Information

- Live Streaming Guide [http://theta360.guide/community-document/live-streaming.html]
- Community Discussion [http://lists.theta360.guide/]

Overview

This document focuses on still images and videos that are saved to the internal storage of the camera and transferred to a mobile phone, computer, or device such as the Sony PlayStation VR [http://lists.theta360.guide/t/new-sony-playstation-vr-integration-with-ricoh-theta/691?u=codetricity]. For information on streaming 360 video from the THETA, see the Live Streaming Guide [http://theta360.guide/community-document/live-streaming.html].

The video is a standard MP4 file. The image is a standard JPEG file. You can use tools like Photoshop, Premiere Pro, Paint to edit both the images and the videos.

The THETA has two lenses. Each lens captures half of a spheres. The two spheres are placed side by side on media internal to the camera. This image or video is referred to as dual-fisheye.

Figure 1. dual-fisheye format

Normally, you won’t see or deal with media in dual-fisheye format. You will usually deal with media in equirectangular format.

Figure 2. equirectangular format

360 players, such as the RICOH player, YouTube, Facebook, or a VR headset will take media in equirectangular format and enable navigation.

Figure 3. THETA media in 360 player

Resize

You can resize your image as long as it is in ratio of 2:1.

Figure 4. original image dimensions

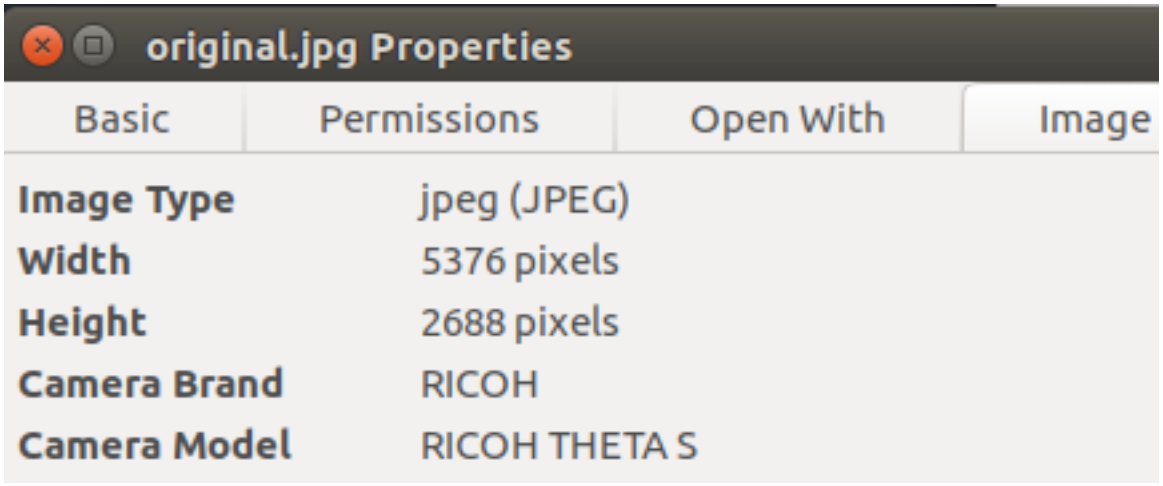


Figure 5. original file size

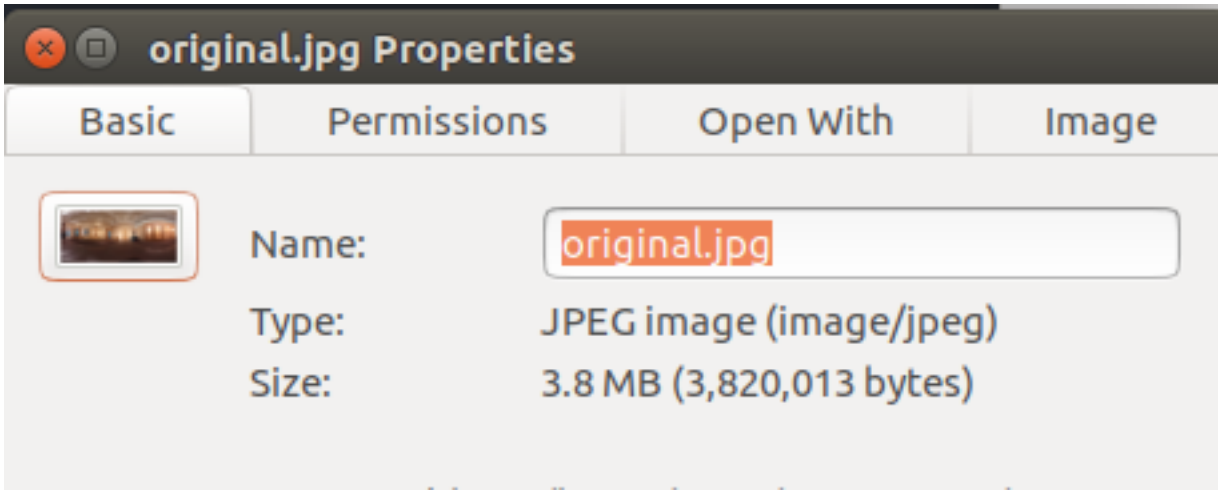
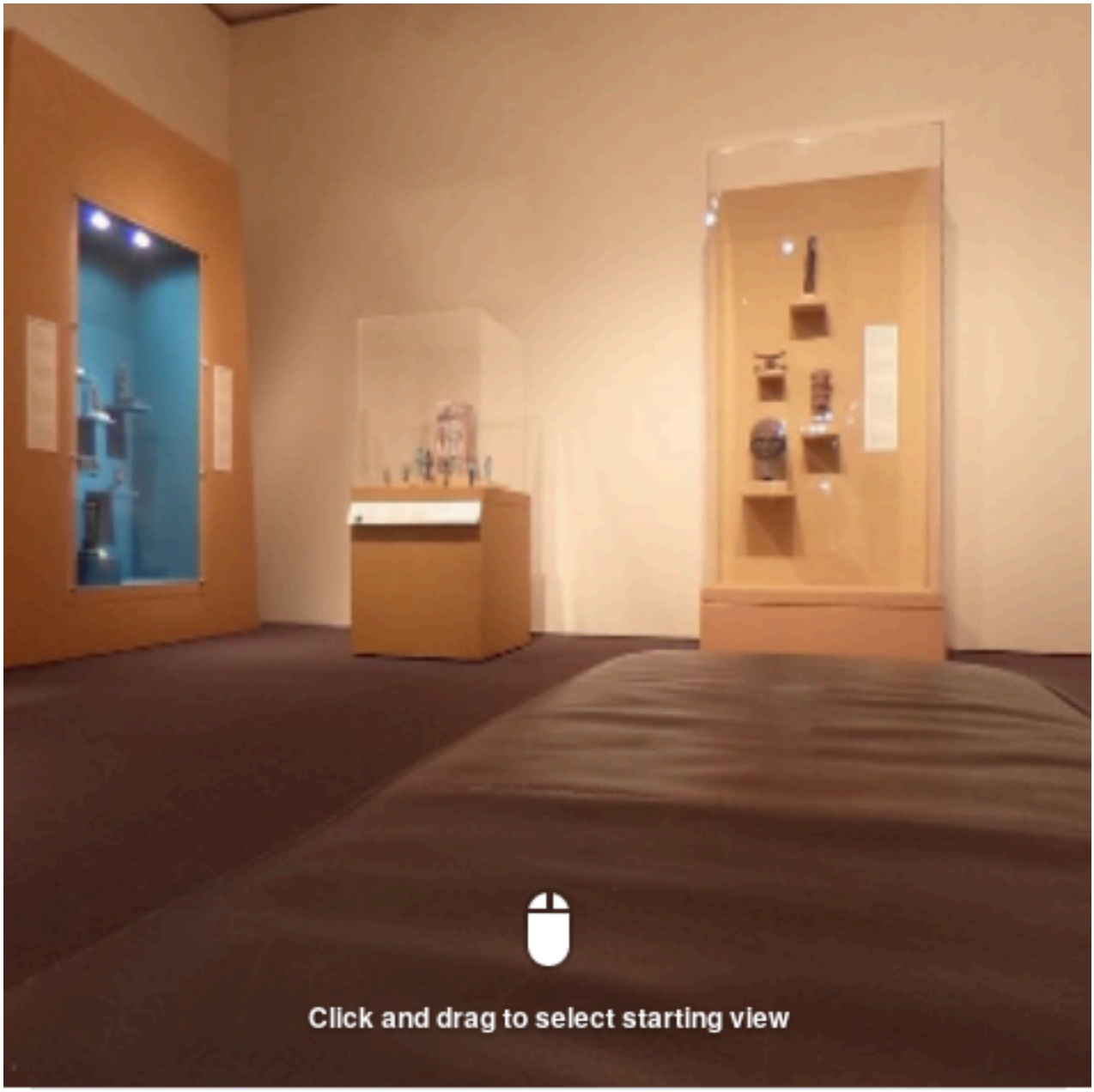


Figure 6. original image on Facebook



In the example below, I'm using the Linux utility to ImageMagick [https://www.imagemagick.org] to resize the image to a low resolution of 800x400.

```
$ convert original.jpg -resize 800x400 resized.jpg
```

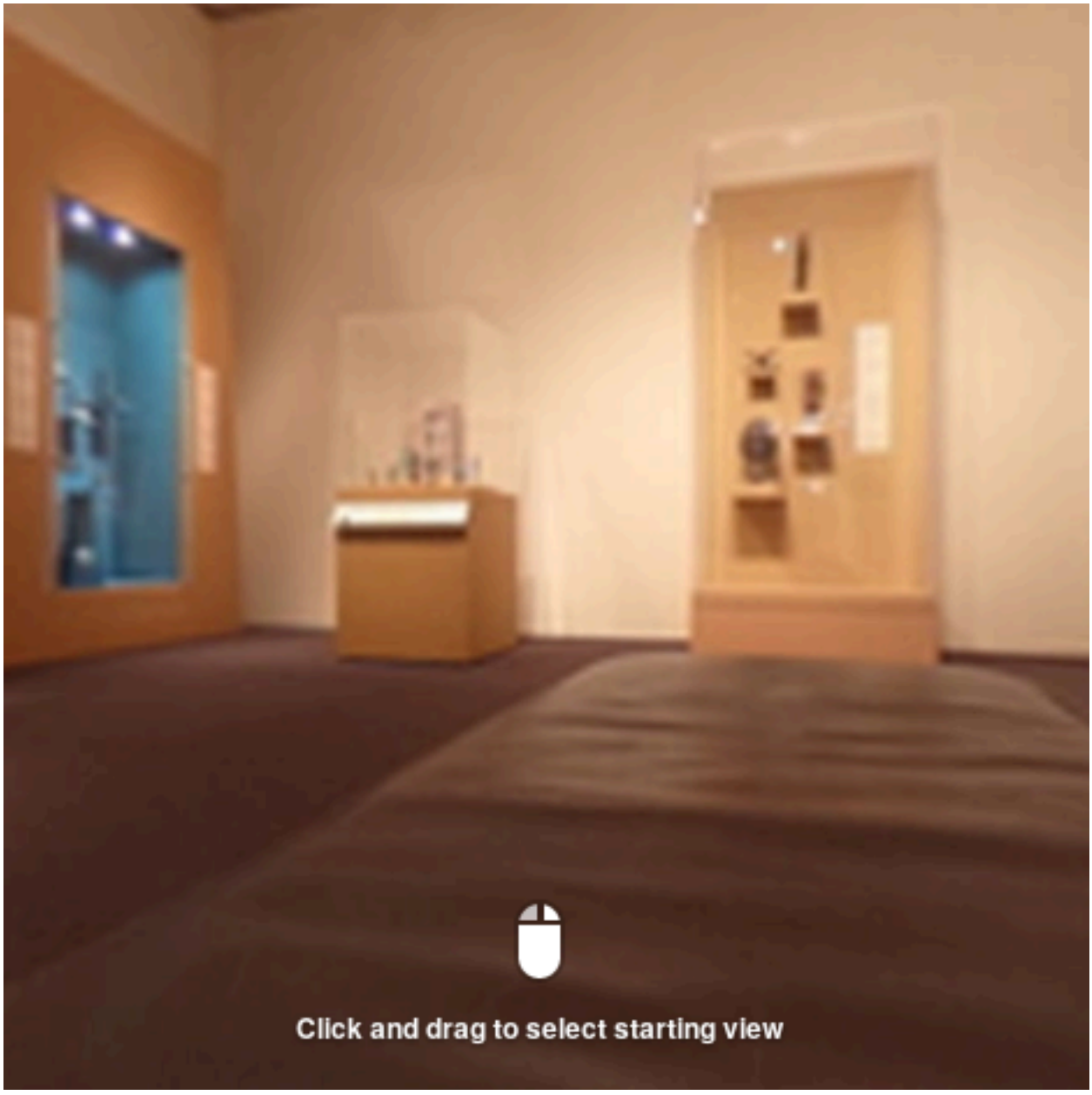
I'll verify that the file size was reduced from 3.8M to 0.2M.

```
$ ll
-rw-r--r-- 1 craig craig 3820013 Oct 23  2016 original.jpg
-rw-r--r-- 1 craig craig  205001 Jun  1 13:51 resized.jpg
$
```

Tip

In the example above, I've aliased `ls -l` to `ll`.

Figure 7. resized low-res image on Facebook



Navigation works in the resized image.

Figure 8. navigation works in resized image

I’m using exiftool [http://www.sno.phy.queensu.ca/~phil/exiftool/] to verify the dimensions of the file.

```
$ exiftool resized.jpg |grep 'Image Size'
Image Size                      : 800x400
```

Color

You can apply color transformations to the THETA image and retain the 360 features.

In this example, I’m using ImageMagick from the command line to change the image into a black and white 360 image.

```
$ convert original.jpg -colorspace Gray  -emboss 0x.5  gray.jpg
```

Figure 9. gray filter applied

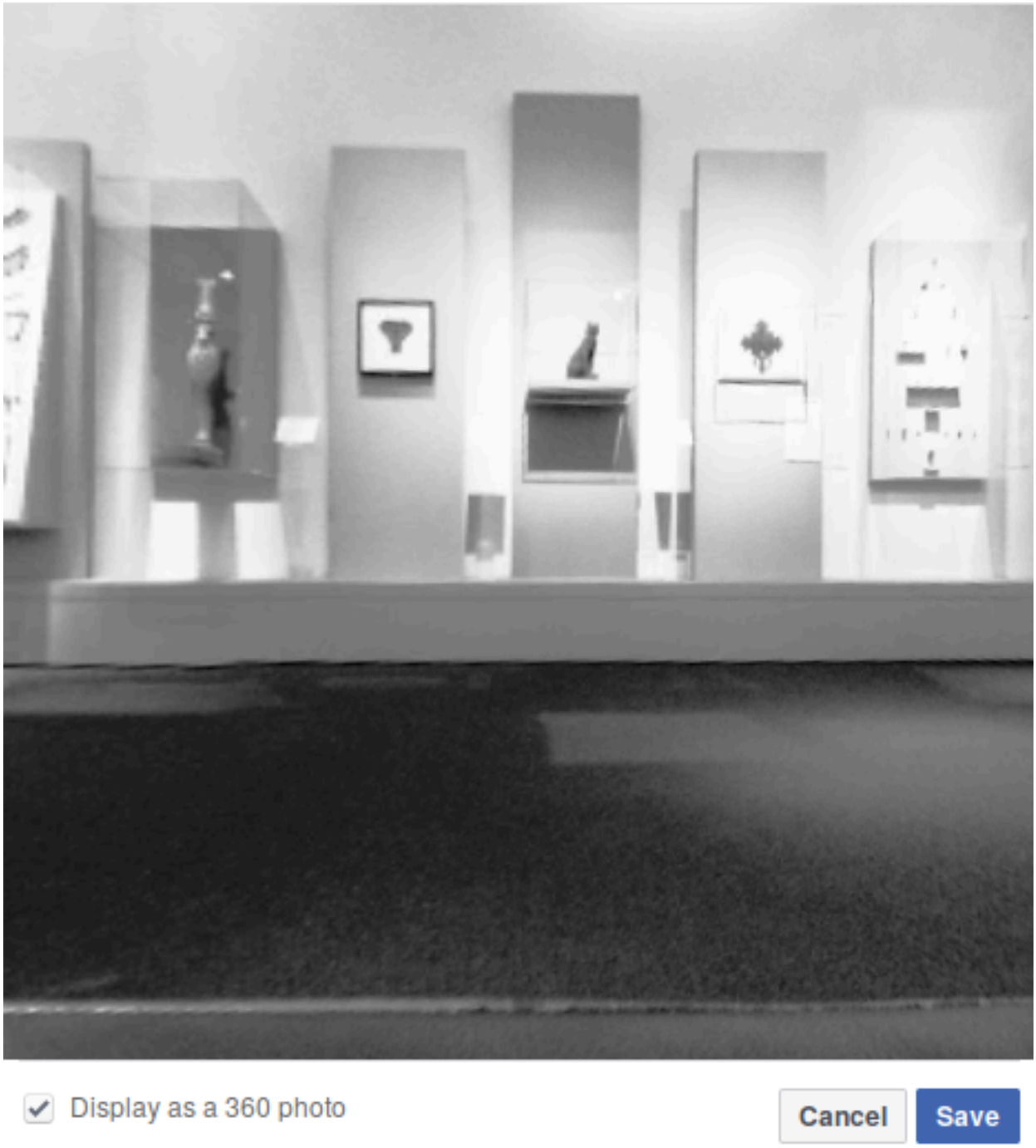


Figure 10. 360 navigations works after changing color

This example shows a command line sketch transformation applied to the 360 image.

Figure 11. pencil sketch applied to THETA image

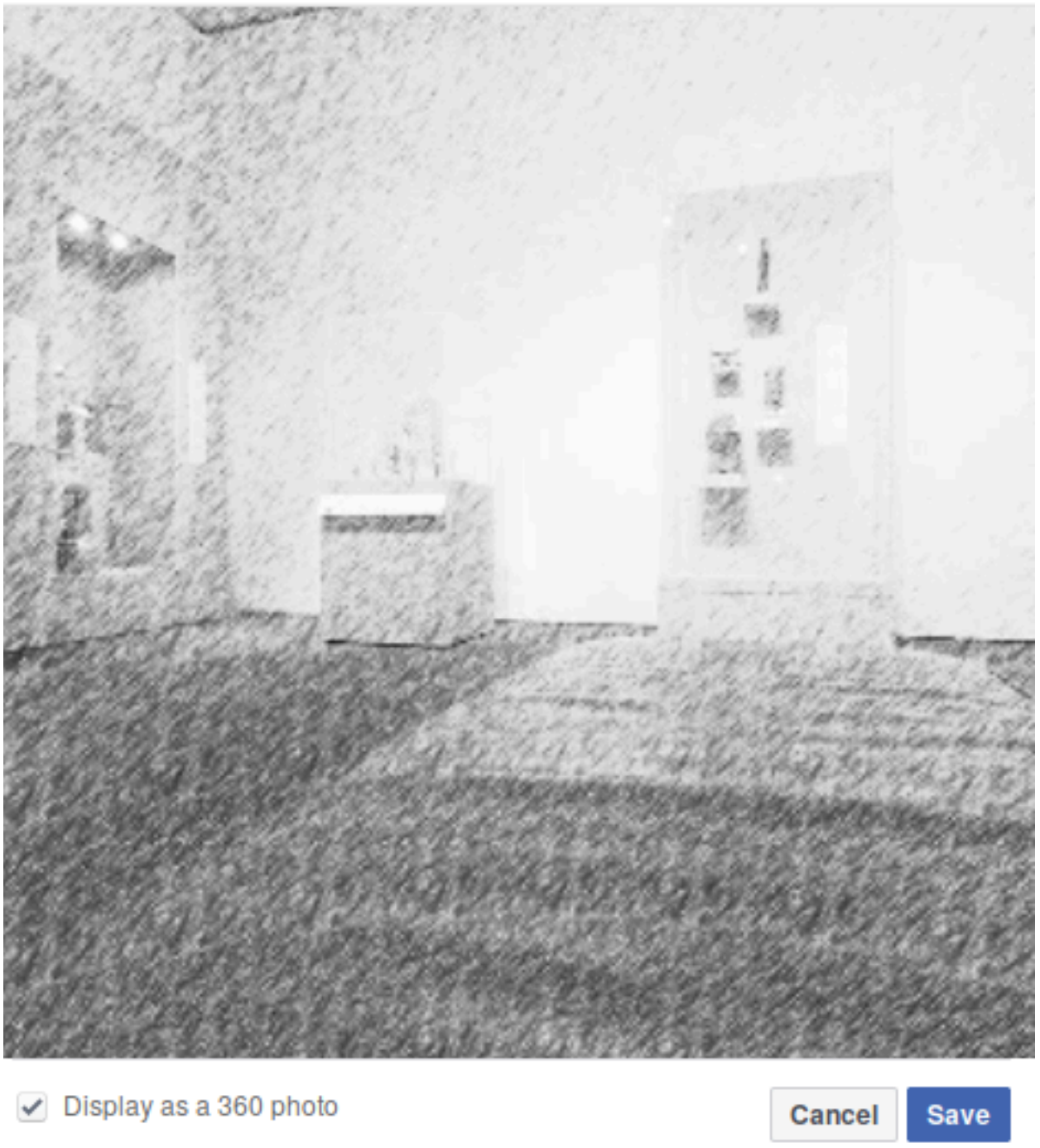


Figure 12. navigation works with pencil sketch

The actual command is somewhat complex, but you can just copy the command and replace the file name.

First make a pencil tile. You just need to do this once to generate a small tile for the pencil.

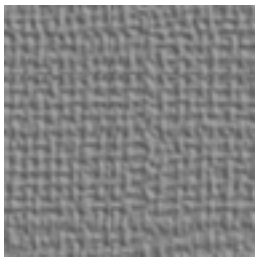
```
convert -size 256x256 xc:  +noise Random  -virtual-pixel tile  -motion-blur 0x20+1
```

Then apply the command below.

```
convert original.jpg -colorspace gray \( +clone -tile pencil_tile.gif -draw "color
```

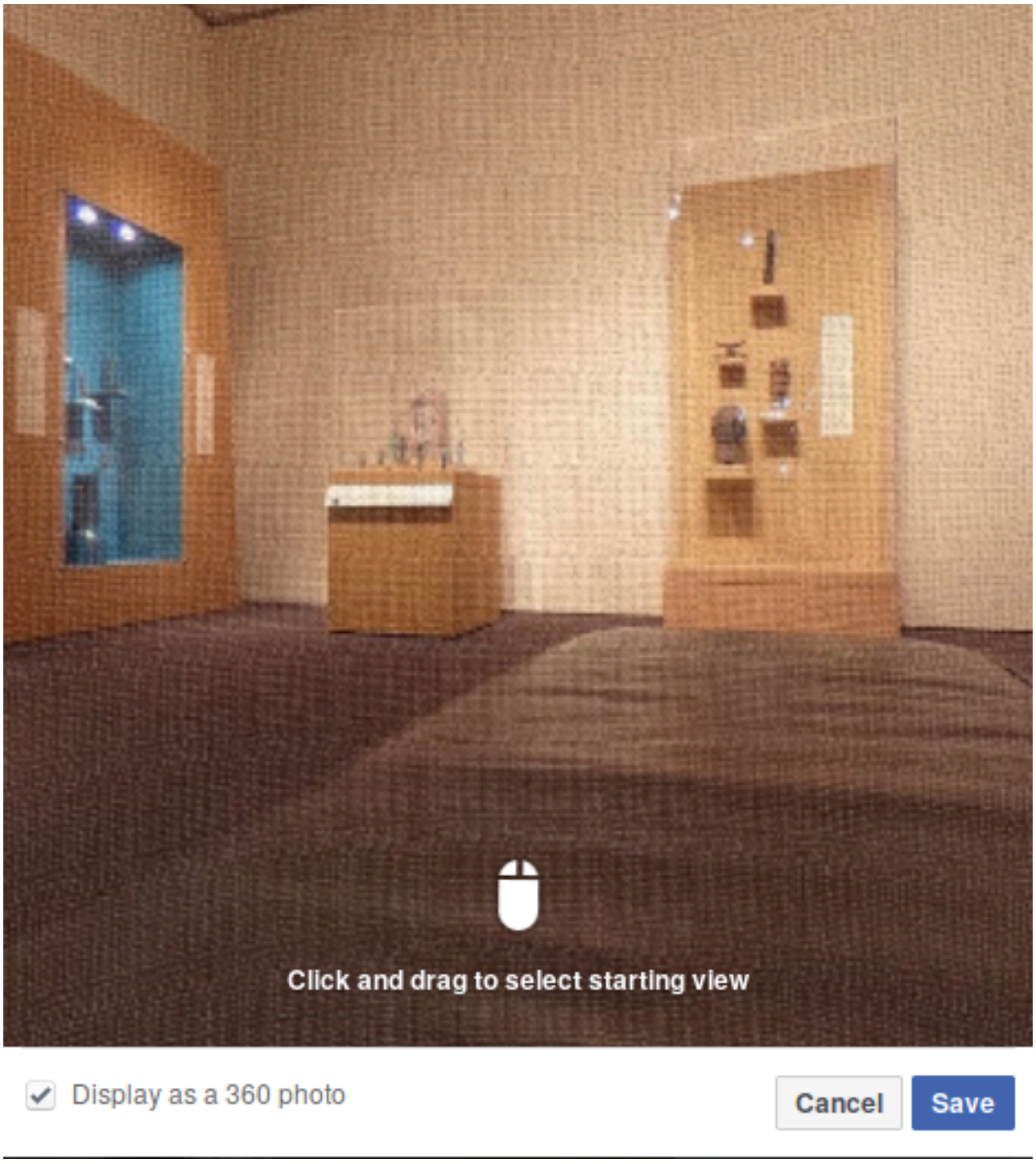
In this example, I’m applying a photo texture with this graphic.

Figure 13. fabric texture tile applied to THETA image



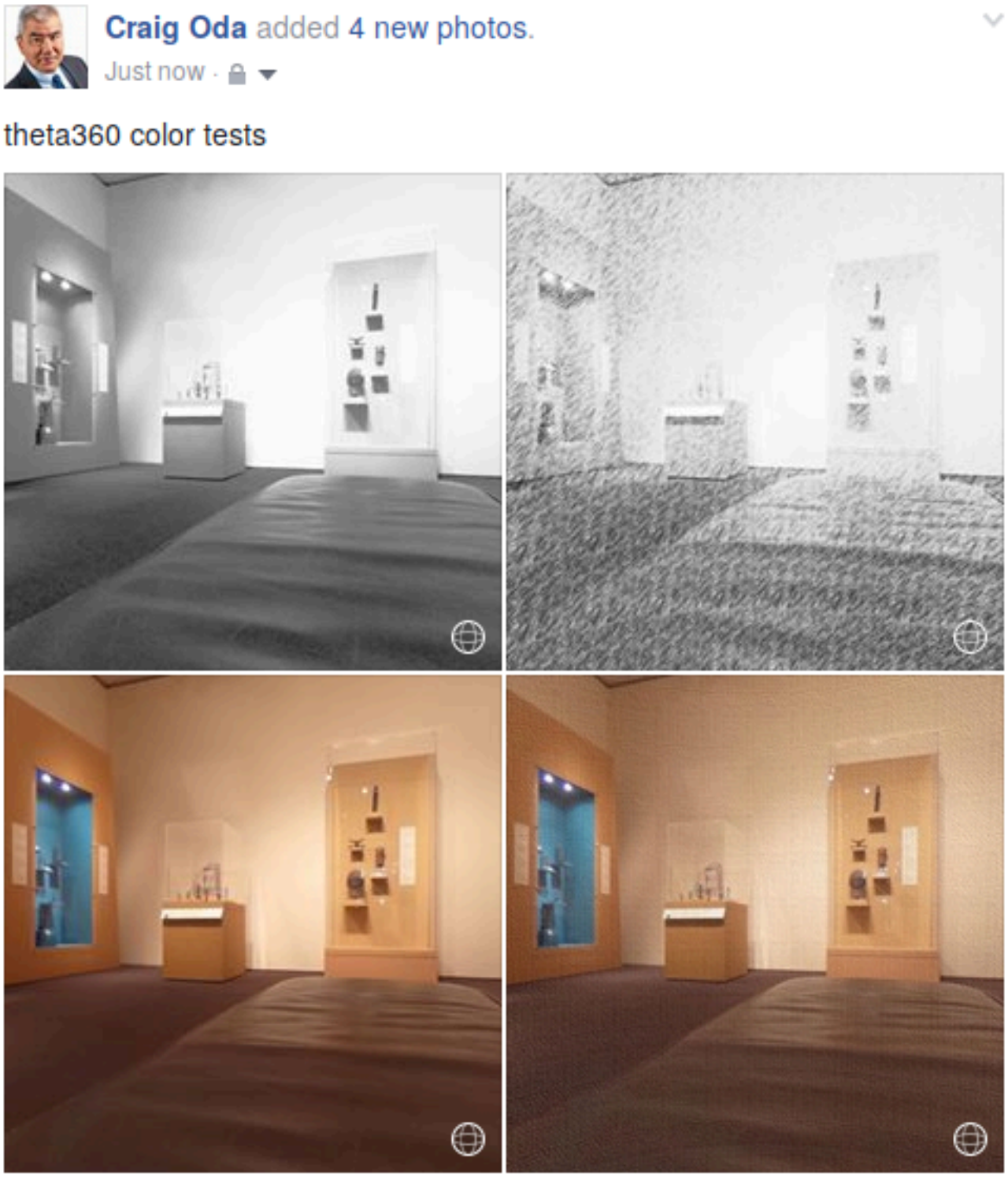
The resized image retains 360 navigation.

Figure 14. THETA image with texture effect



Here’s what the images look like on Facebook.

Figure 15. 360 image color tests



metadata

RICOH THETA images have EXIF [https://en.wikipedia.org/wiki/Exif] and XMP [https://developers.google.com/streetview/spherical-metadata] metadata embedded into the image files

What is Metadata?

360 images have hidden text embedded into the image data. This data is called metadata. It is not designed for humans. A computer reads in the metadata.

Photo Sphere XMP Metadata [https://developers.google.com/streetview/spherical-metadata] is a Google standard taken from the Adobe XMP standard. You can edit this data with an image editor or a text editor. Most 360 image viewers such as Facebook will support XMP data. As 360 images are still new, not all applications support XMP data. If the application does not support XMP data, you will need to edit the entire image with something link Hugin [http://lists.theta360.guide/t/hugin-howto-adjust-theta-image-tilt-and-centering/1270] or Photoshop.

Why Use XMP Data?

Editing the XMP data uses less CPU and memory resources. Your application can change the text string for orientation with almost no delay for the user of your application. Editing the entire image orientation with image editing techniques is going to place a heavy load on the mobile or desktop application.

What Can You Edit?

You can edit both the XMP data embedded in the THETA image as well as add new XMP data tags that comply to the Photo Sphere XMP Metadata standard. For example, your application can add GPS data as well as initial view data to the image. The data can then be used by 360 viewing applications such as Facebook.

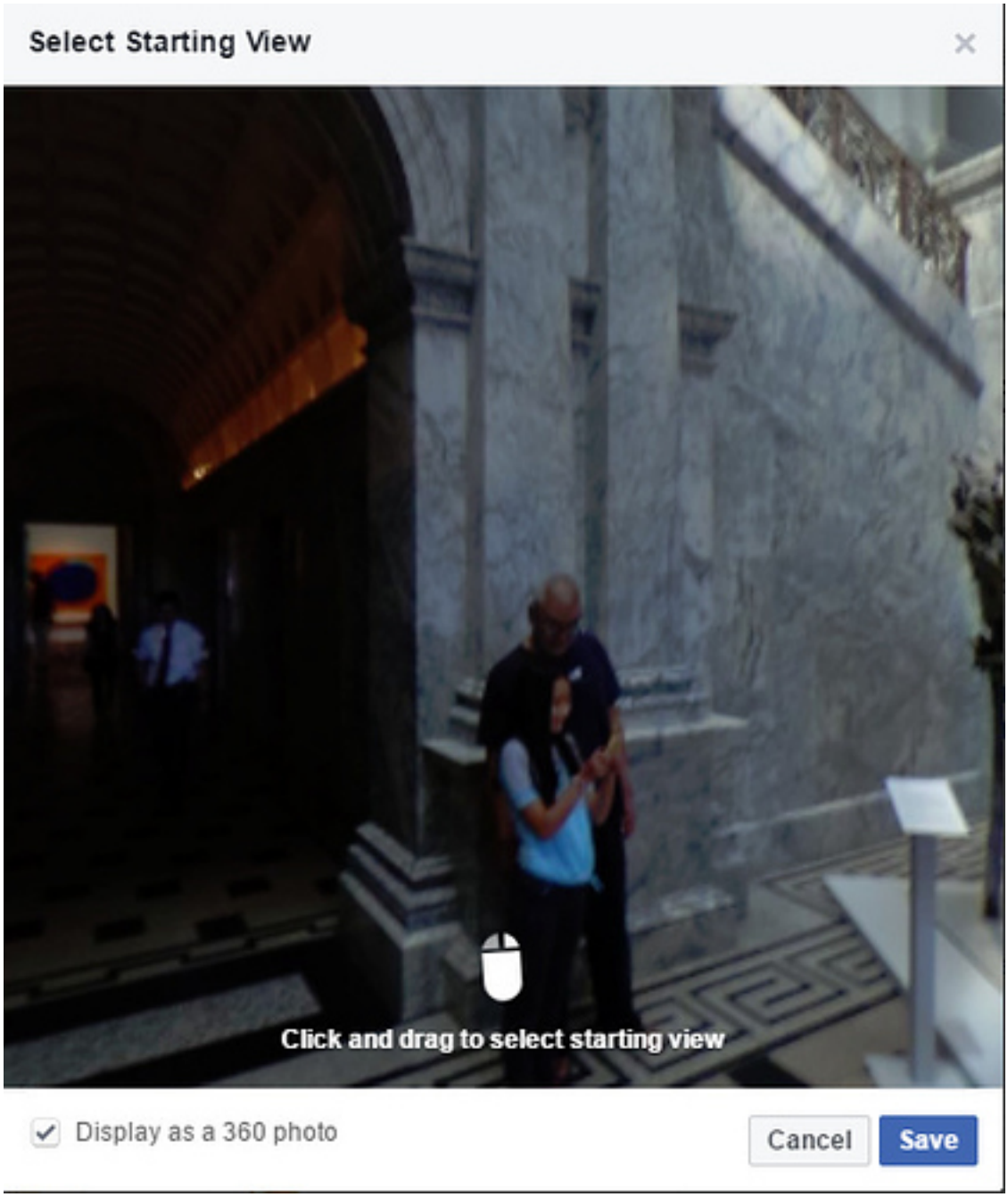
Overview of Functionality using Facebook as Viewer

This is the same image shown on Facebook with no mouse editing. The orientation was changed by only using XMP data strings.

Default View of Image Before Editing

The original image looks like this on Facebook with no editing. Although Facebook allows you to set the orientation with their Facebook app, let’s imagine that you’re building a better Facebook or building a new application to support XMP data. Imagine that the mouse is not there.

Figure 16. default starting view



This is the default XMP data that we'll edit for our tests.

```
<GPano:PosePitchDegrees>-0.9</GPano:PosePitchDegrees>
<GPano:PoseRollDegrees>-0.4</GPano:PoseRollDegrees>
```

Pose Pitch Increased by 180 Degrees

Here, I'll increase the PosePitchDegrees by 180 degrees to illustrate the orientation change. Image the camera is upside and you need to correct the image orientation.

Figure 17. pitch rotated 180 degrees


```
<GPano:PosePitchDegrees>179.1</GPano:PosePitchDegrees>
<GPano:PoseRollDegrees>-0.4</GPano:PoseRollDegrees>
```

Pose Roll Increased by 180 Degrees

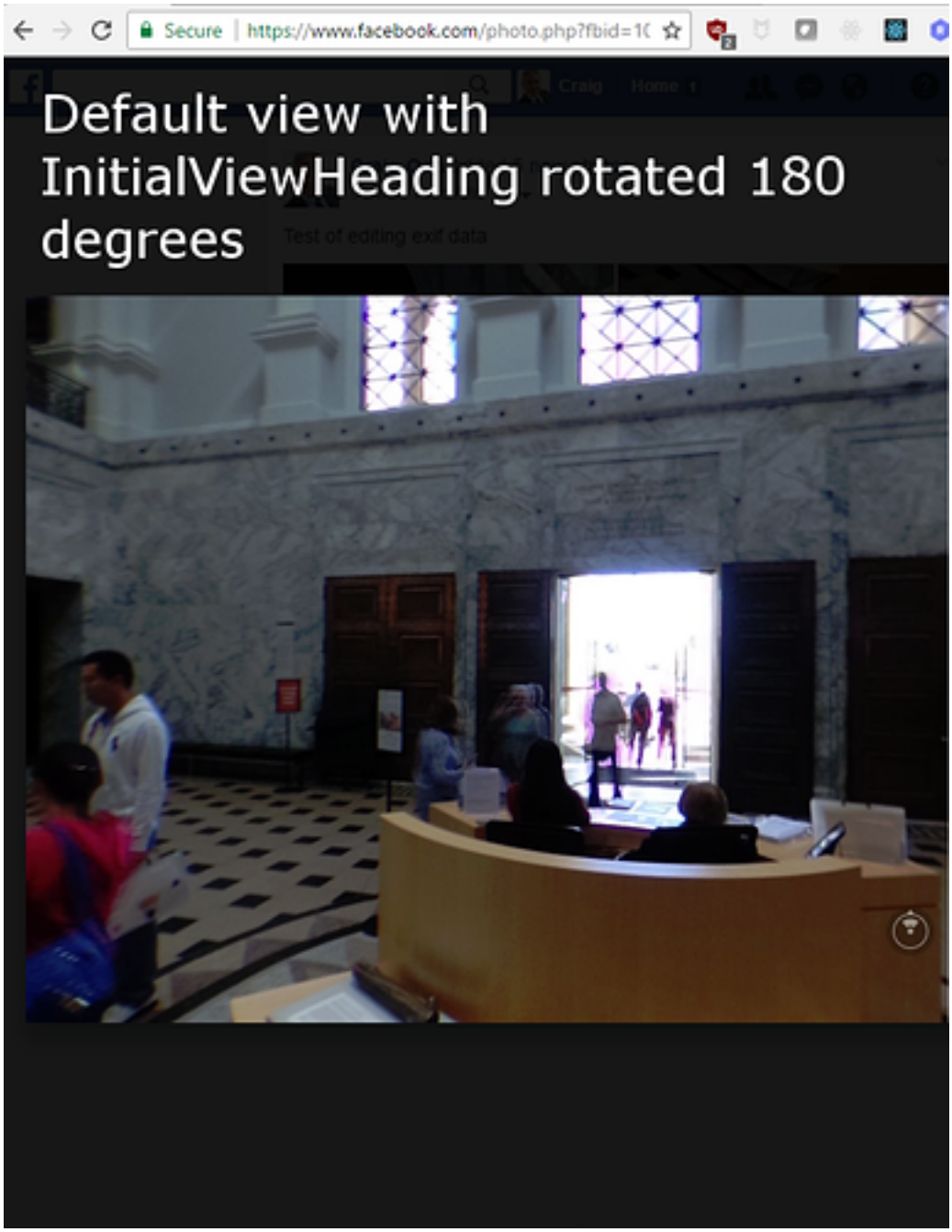
In the next example, I’ve increased the Roll by 180 and reset the Pitch to the default value. image::img/metadata/roll180.jpg[role="thumb" title="roll rotated 180 degrees"]

```
<GPano:PosePitchDegrees>-0.9</GPano:PosePitchDegrees>
<GPano:PoseRollDegrees>179.6</GPano:PoseRollDegrees>
```

InitialViewHeadingDegrees Decreased by 180

In the next example, I've added a new metadata tag for *InitialViewHeadingDegrees*. I got the name of the tag from the XMP standard. I've rotated the orientation by 180 degrees to show the back of the image sphere.

Figure 18. initial view heading rotated 180 degrees




```
<GPano:PosePitchDegrees>-0.9</GPano:PosePitchDegrees>
<GPano:PoseRollDegrees>-0.4</GPano:PoseRollDegrees>
<GPano:InitialViewHeadingDegrees>135</GPano:InitialViewHeadingDegrees>
```

Metadata Tools

There are a number of free tools and libraries to access the metadata.

- ExifTool by Phil Harvey [http://www.sno.phy.queensu.ca/~phil/exiftool/]
- ExifToolGUI [http://u88.n24.queensu.ca/~bogdan/]
- the eXif.er [https://www.thexifer.net/]

Figure 19. ExifToolGUI

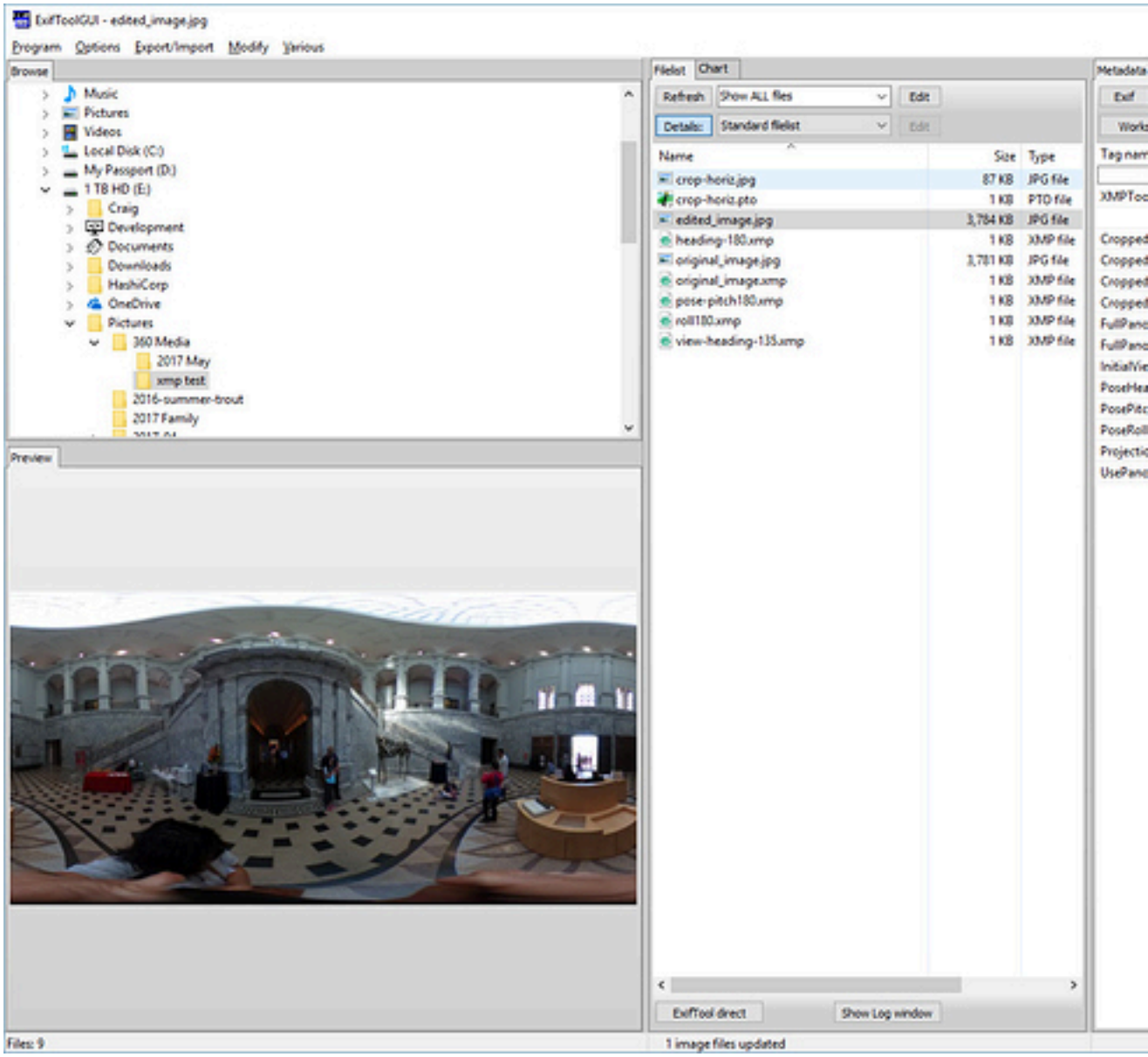
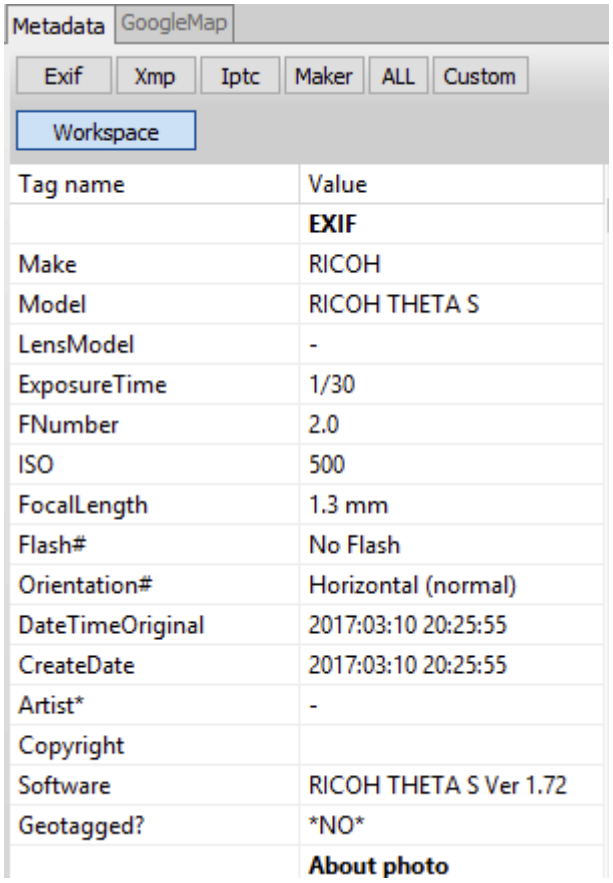


Figure 20. metadata viewed with ExifToolGUI



Tag name	Value
	EXIF
Make	RICOH
Model	RICOH THETA S
LensModel	-
ExposureTime	1/30
FNumber	2.0
ISO	500
FocalLength	1.3 mm
Flash#	No Flash
Orientation#	Horizontal (normal)
DateTimeOriginal	2017:03:10 20:25:55
CreateDate	2017:03:10 20:25:55
Artist*	-
Copyright	
Software	RICOH THETA S Ver 1.72
Geotagged?	*NO*
	About photo

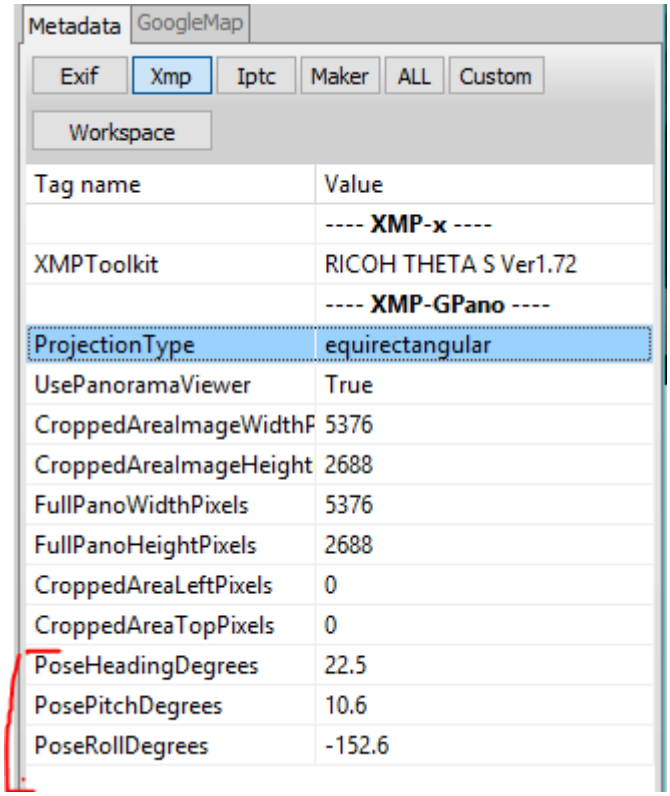
You can immediately see useful data in the metadata tags. Many sites, including Facebook will look [https://facebook360.fb.com/editing-360-photos-injecting-metadata/] for a **Make** of **RICOH** and a **Model** of **RICOH THETA S**. You can use the metadata **Make** and **Model** to see if the image file your application is opening is a 360 image.

Important

The THETA image must also be in a ratio of 2:1. Your application should check for this ratio in case the image was edited incorrectly in post-processing.

Clicking on the Xmp tab, we'll now get to the real exciting data.

Figure 21. Exciting THETA XMP Data



For starters, you can check for the **ProjectionType** of **equirectangular**. Your application can easily identify an equirectangular image this way.

Below this, you can see the **PoseHeadingDegrees**, **PosePitchDegrees**, and **PoseRollDegrees**. The THETA camera has internal sensors to get the heading, pitch and roll.

Caution

The user of your application may have edited the image and deleted the metadata from the image. The solution is to review their post-processing workflow and then inspect the metadata manually or your application can check it.

Metadata can be injected into a 360 image that has lost the data by graphic editing tools. Programming libraries can do this, or you can use exiftool in a script.

```
exiftool -ProjectionType="equirectangular" photo.jpg
```

Here’s an article [http://lists.theta360.guide/t/getting-360-images-to-work-after-resize-exif-technique/1066] on getting the images to work after resizing the image by copying the original XMP data and copying over the resized image.

Edit XMP Data In Text File

Figure 22. Use ExifToolGui to export XMP data as a text file

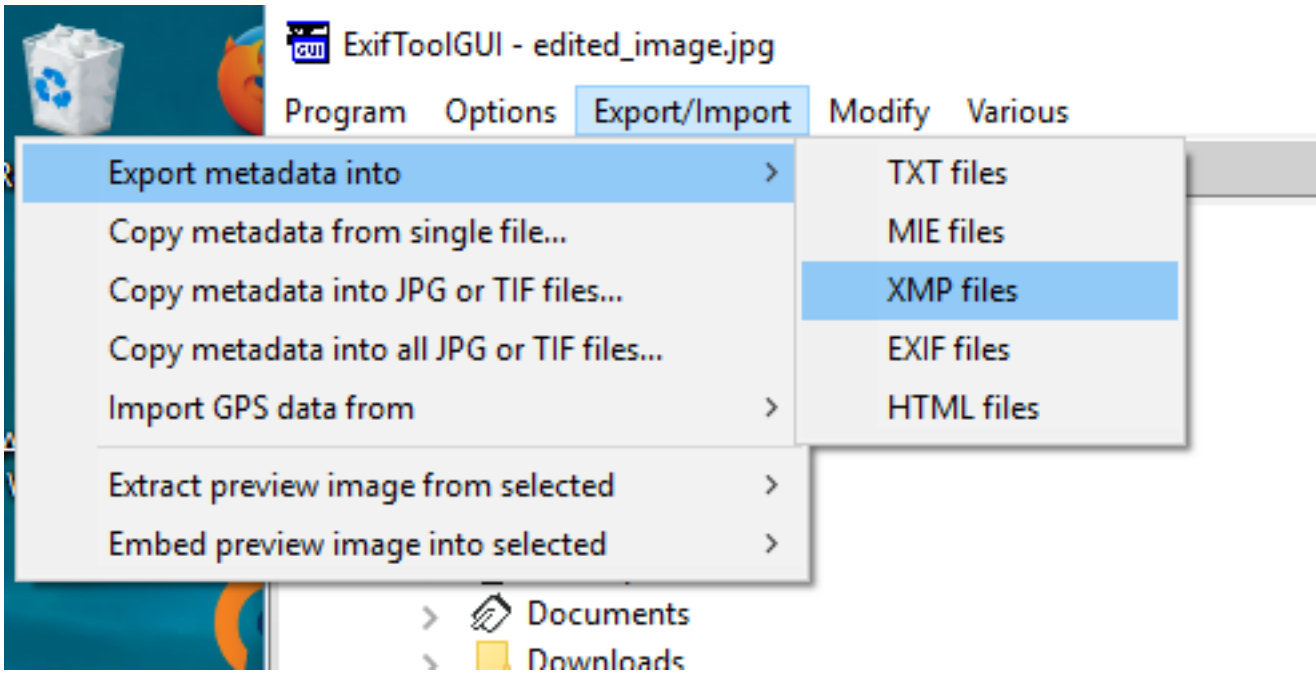
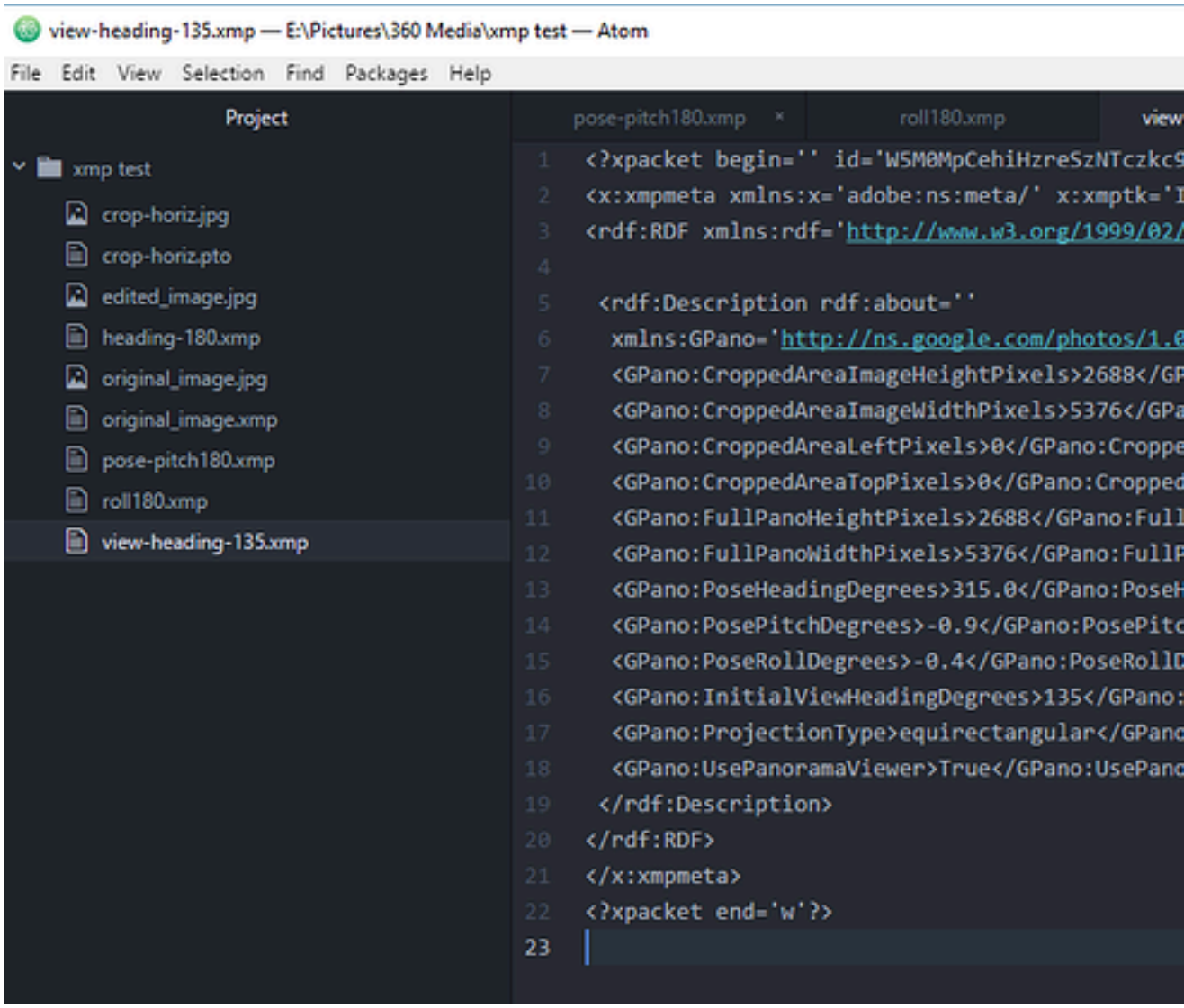
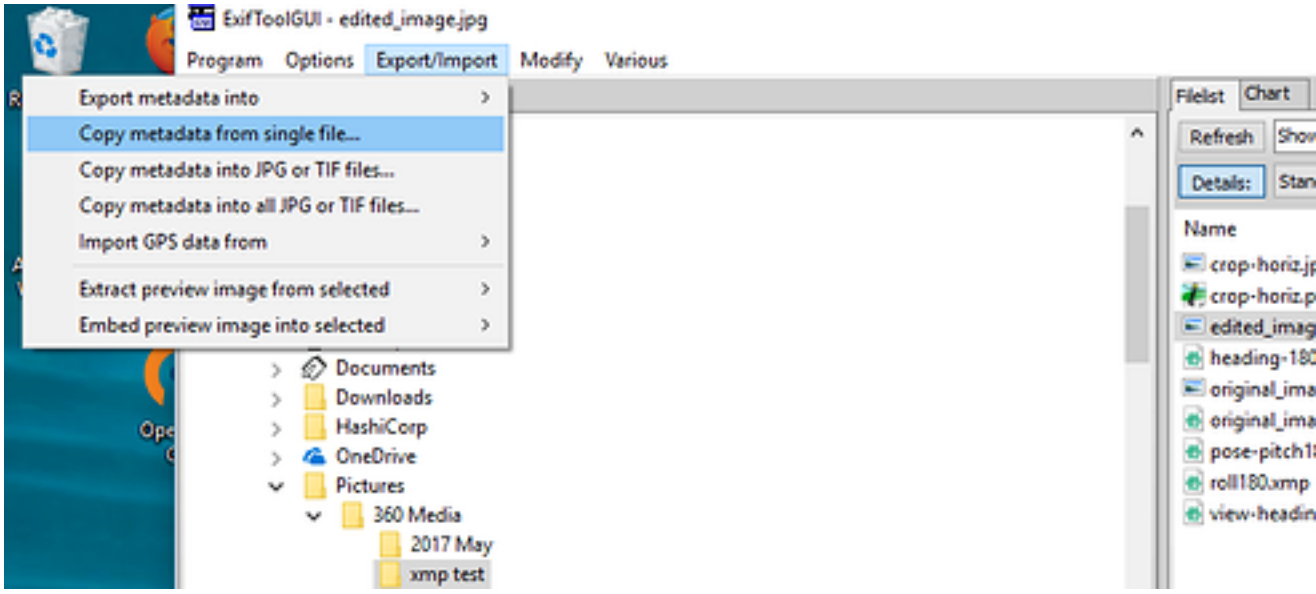


Figure 23. XMP data can be edited in a text editor like Atom



Import XMP data into THETA image. With the THETA image you want to edit selected, copy metadata from single file.

Figure 24. Using ExifToolGui, copy metadata from single file into image



Exiftool from command line

You can read and edit the XMP data directly from the command line. In the examples below, I'm using a bash shell on Windows. Many people use a bash script to control exiftool

Reading Data

```
$ ./exiftool.exe edited_image.jpg > data.txt
$ less data.txt
```

This is the metadata in my data.txt file

```
ExifTool Version Number         : 10.47
File Name                       : edited_image.jpg
Directory                       : .
File Size                       : 3.7 MB
File Modification Date/Time     : 2017:05:31 12:44:39-07:00
File Access Date/Time           : 2017:05:31 12:44:39-07:00
File Creation Date/Time         : 2017:05:31 11:31:21-07:00
File Permissions                : rw-rw-rw-
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
Exif Byte Order                 : Big-endian (Motorola, MM)
Image Description               :
Make                            : RICOH
Camera Model Name               : RICOH THETA S
Orientation                     : Horizontal (normal)
X Resolution                    : 72
Y Resolution                    : 72
Resolution Unit                 : inches
Software                        : RICOH THETA S Ver 1.62
```

```
Modify Date                     : 2016:10:23 12:25:33
Y Cb Cr Positioning             : Co-sited
Copyright                       :
Exposure Time                   : 1/125
F Number                        : 2.0
ISO                             : 100
Sensitivity Type                : Standard Output Sensitivity
Exif Version                    : 0230
Date/Time Original              : 2016:10:23 12:25:33
Create Date                     : 2016:10:23 12:25:33
Components Configuration        : Y, Cb, Cr, -
Compressed Bits Per Pixel       : 3.2
Aperture Value                  : 2.0
Brightness Value                : 3.5
Exposure Compensation           : -0.7
Max Aperture Value              : 2.0
Metering Mode                   : Multi-segment
Light Source                    : Unknown
Flash                           : No Flash
Focal Length                    : 1.3 mm
Maker Note Type                 : Rdc
Firmware Version                : 1.62
Serial Number                   : (00000000)00010093
Recording Format                : JPEG
Accelerometer                   : 359.6 -0.9
Compass                         : 315
Time Zone                       : -07:00
Exposure Program                : Auto
White Balance                   : Auto
User Comment                    :
Flashpix Version                : 0100
Color Space                     : sRGB
Exif Image Width                : 5376
Exif Image Height               : 2688
Interoperability Index          : R98 - DCF basic file (sRGB)
Interoperability Version        : 0100
Exposure Mode                   : Auto
Scene Capture Type              : Standard
Sharpness                       : Normal
GPS Version ID                  : 2.3.0.0
GPS Latitude Ref                : North
GPS Longitude Ref               : West
GPS Altitude Ref                : Above Sea Level
GPS Time Stamp                  : 19:25:31
GPS Img Direction Ref           : True North
GPS Img Direction               : 315
GPS Map Datum                   : WGS84
GPS Date Stamp                  : 2016:10:23
Compression                     : JPEG (old-style)
Thumbnail Offset                : 58496
Thumbnail Length                : 3118
XMP Toolkit                     : Image::ExifTool 10.47
Cropped Area Image Height Pixels: 2688
Cropped Area Image Width Pixels : 5376
```

```
Cropped Area Left Pixels        : 0
Cropped Area Top Pixels         : 0
Full Pano Height Pixels         : 2688
Full Pano Width Pixels          : 5376
Initial View Heading Degrees    : 135
Pose Heading Degrees            : 315.0
Pose Pitch Degrees              : -0.9
Pose Roll Degrees               : -0.4
Projection Type                 : equirectangular
Use Panorama Viewer             : True
Image Width                     : 5376
Image Height                    : 2688
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:2 (2 1)
Aperture                        : 2.0
GPS Altitude                    : 36 m Above Sea Level
GPS Date/Time                   : 2016:10:23 19:25:31Z
GPS Latitude                    : 37 deg 25' 58.58" N
GPS Longitude                   : 122 deg 10' 14.32" W
GPS Position                    : 37 deg 25' 58.58" N, 122 deg 10' 14.32" W
Image Size                      : 5376x2688
Megapixels                      : 14.5
Ricoh Pitch                     : -0.9
Ricoh Roll                      : -0.399999999999977
Shutter Speed                   : 1/125
Thumbnail Image                 : (Binary data 3118 bytes, use -b option to extrac
Focal Length                    : 1.3 mm
Light Value                     : 9.0
```

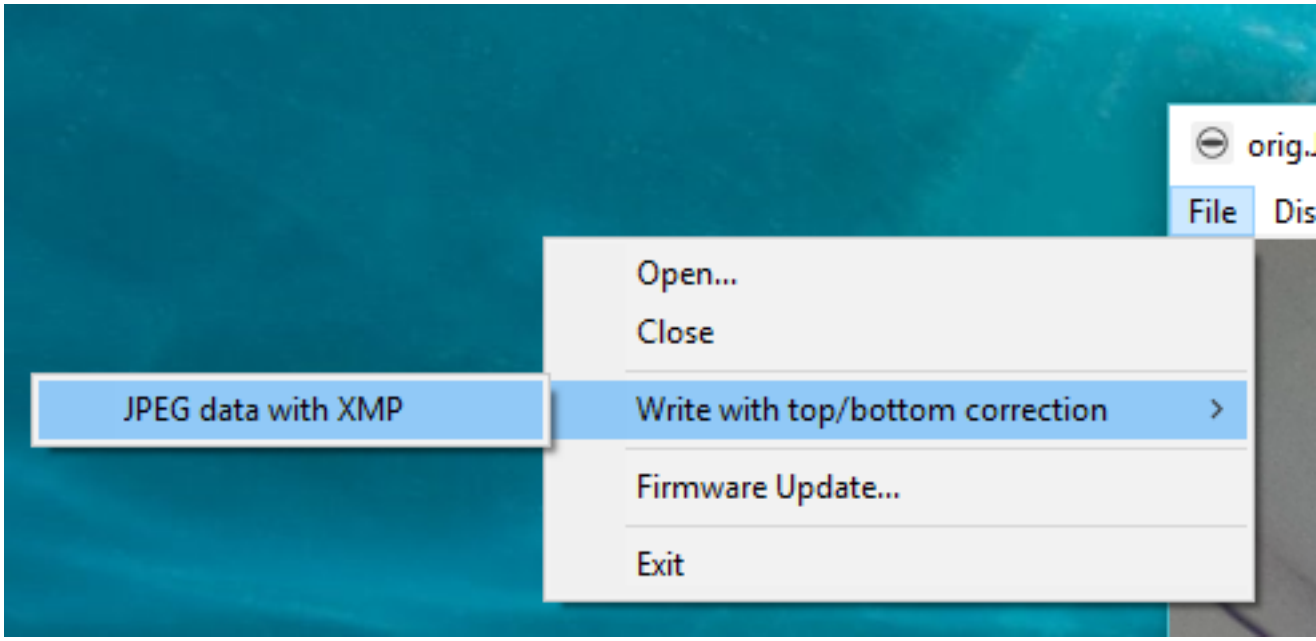
Writing Data

```
$ ./exiftool.exe -InitialViewHeadingDegrees=45 edited_image.jpg
    1 image files updated
```

Zero Out XMP Data with Desktop Application

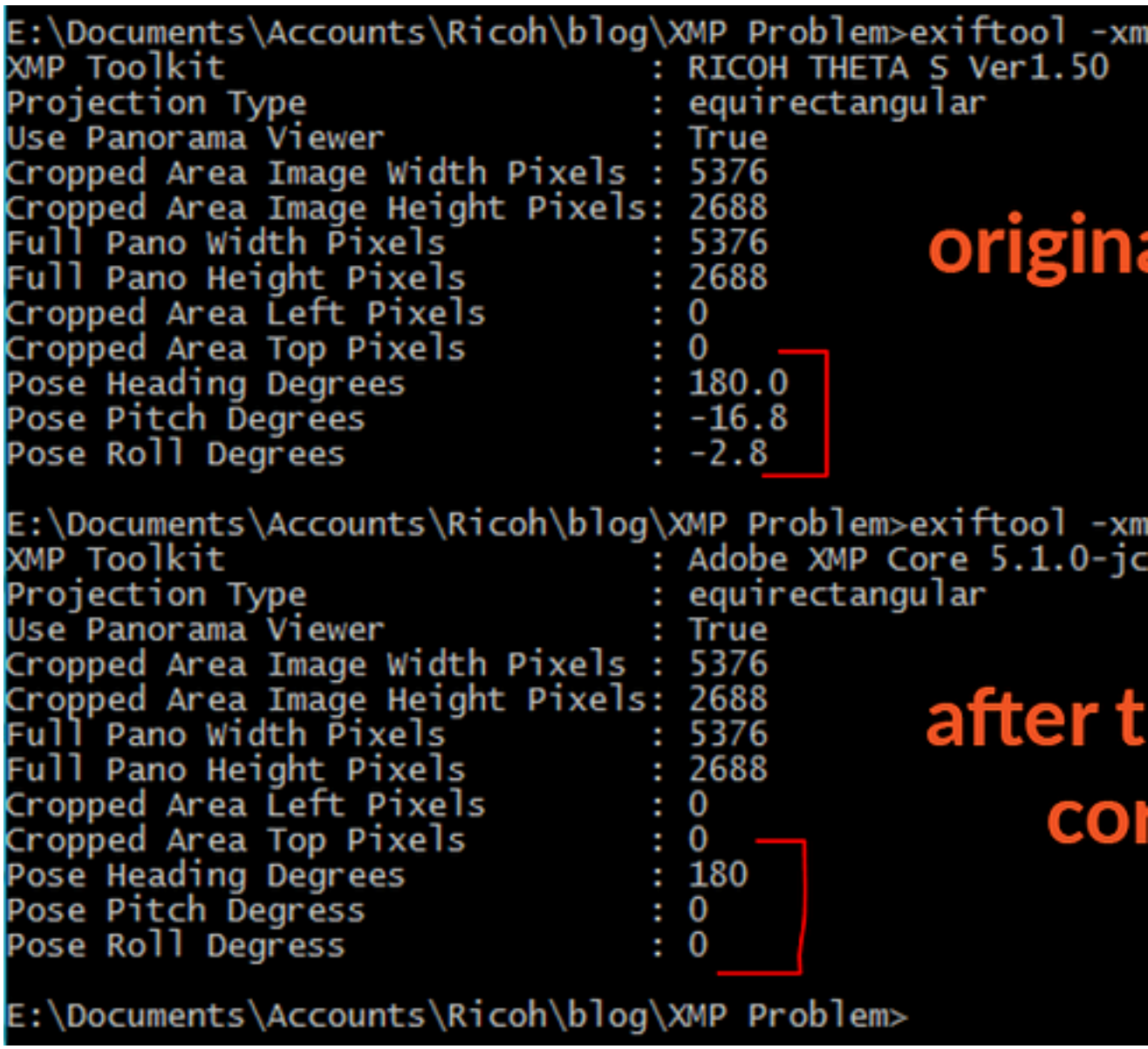
The THETA desktop application can be used to zero out the pitch and roll so that your image editing application does not get confused. Problems occur when the entire image is edited with an image editor and then an application like Google Street View applies the XMP pitch and roll correction on top of the editing.

Figure 25. Official THETA Desktop Application saving XMP data



After the correction, the XMP data for pitch and roll will be zeroed out.

Figure 26. Pitch and Roll are zeroed out by the THETA app



Important

The orientation of images on an application like Google Streetview will be incorrect if the user applied correction to some images and not others. They need to apply the adjustment to all images.

Libraries

- easyexif [https://github.com/mayanklahiri/easyexif]: C++ EXIF parsing library by Mayank Lahiri [http://lahiri.me/]
- Exiv2 [http://www.exiv2.org/]: C++ metadata library and tools by Andreas Huggel
- ExifLib [https://www.codeproject.com/Articles/36342/ExifLib-A-Fast-Exif-Data-Extractor-for-NET] - A fast EXIF data extractor for .NET by Simon McKenzie

End User Tools

These tools are for digital artists and support the XMP tags. Many other tools support the metadata tags.

- Flexify 2 [http://www.flamingpear.com/flexify-2.html] Photoshop plug-in
- Adobe Premiere Pro with VR support [https://helpx.adobe.com/premiere-pro/how-to/vr-video.html]

Specification

The RICOH THETA S and THETA SC take images in two different sizes: 5376x2688 and 2048x1024. The application you build can also resize the images to any width and height that is in a 2:1 ratio.

Note

The image must be in a ratio of 2:1.

dual-fisheye

Stitching

Equirectangular.

RICOH THETA lens distortion data.

Sony PlayStation VR Media Player

Sample Application

The sample application theta_rectify [https://github.com/khufkens/theta_rectify] and post [http://lists.theta360.guide/t/theta-s-auto-level-bash-script/1257] were contributed to the community by Dr. Koen Hufkens [http://www.khufkens.com/] of Harvard University.

My Virtual Forest [http://virtualforest.io] project is still running strong and generates tons of spherical images (currently ~50GB). However, the post on which the camera sits is not perfectly level. The Theta S camera normally compensates for this using an internal gyroscope which detects pitch and roll of the camera. Yet, when downloading images directly from the camera no adjustments are made and the pitch and roll data is merely recorded in the EXIF data of the image.

As such I wrote a small bash script which rectifies (levels the horizon) in Theta S spherical images using this internal EXIF data. This is an alternative implementation to the THETA EXIF Library [https://github.com/regen100/thetaexif] by Regen [https://github.com/regen100]. I use his cute Lama test images for reference. All credit for the funky images go to Regen. Below is the quick install guide to using my script. I hope it helps speed up people's Theta S workflow.

Install

Download, fork or copy paste the script from my github repository [https://github.com/khufkens/theta_rectify] to your machine and make it executable.

```
$ chmod +x theta_rectify.sh
```

Use

```
$ theta_rectify.sh image.jpg
```

The above command will rectify the image.jpg file and output a new file called image_rectified.jpg.

Figure 27. Original Image

Figure 28. rectified image

Visual comparison between my results and those of Regen's python script [http://www.regen-techlog.com/2014/06/26/python-thetaexif/] show good correspondence.

Requirements

The script depends on a running copy of exiftools, imagemagick and POVRay. These tools are commonly available in most Linux distros, and can be installed on OSX using tools such as homebrew. I lack a MS Windows system, but the script should be easily adjusted to cover similar functionality.

Code

```
#!/bin/bash
#
# Automatically levels Theta S spherical images
# depends on exiftools / imagemagick and POVRay
# Should work on most Linux installs and on
# OSX using homebrew installs or similar

# get the filename without the extension
noextension=`echo $1 | sed 's/\(.*\)\..*/\1/'`

# grab the width and height of the images
```

```
height=`exiftool $1 | grep "^Image Height" | cut -d':' -f2 | sed 's/ //g'`
width=`exiftool $1 | grep "^Image Width" | cut -d':' -f2 | sed 's/ //g'`

# grab pitch roll
roll=`exiftool $1 | grep "Roll" | cut -d':' -f2 | sed 's/ //g'`
pitch=`exiftool $1 | grep "Pitch" | cut -d':' -f2 | sed 's/ //g'`
pitch=$(bc <<< "$pitch * -1")

# flip the image horizontally
convert -flop $1 tmp.jpg

# create povray script with correct image parameters
cat <<EOF > tmp.pov
// Equirectangular Panorama Render
// bare bones script
// camera settings
camera {
  spherical // equirectangular projection
  up    y * 1
  right  x * image_width / image_height
  location <0,0,0>     // put camera at origin
  angle 360 180        // full image
  rotate x * 0         // Tilt up (+) or down (-)
  rotate y * -90         // Look left (+) or right (-)
  rotate z * 0         // Rotate CCW (+) or CW (-)
}
// create a sphere shape
sphere {
  // center of sphere
  <0,0,0>, 1
  texture {
    pigment {
      image_map {
        jpeg "tmp.jpg"
        interpolate 2 // smooth it
        once   // don't tile image, just one copy
        map_type 1
      }
    }
    rotate x * $roll   //Tilt up (+) or down (-) or PITCH
    rotate y * 0       //shift left (+) or right (-)
    rotate z * $pitch  //Rotate CCW (+) or CW (-) or ROLL
    finish { ambient 1 }
  }
}
EOF

# execute povray script and rename file
povray +W$width +H$height -D +fj tmp.pov +O${noextension}_rectified.jpg

# remove temporary files / clean up
rm tmp.jpg
rm tmp.pov
```

Explanation

This section added by Craig.

exiftool pulls the orientation from the XMP data. To get the Roll, grep for the Roll.

```
$ exiftool original.jpg |grep Roll
Pose Roll Degrees               : -0.3
Ricoh Roll                      : -0.300000000000011
```

Same for pitch.

```
$ exiftool original.jpg |grep "Pitch"
Pose Pitch Degrees              : 2.8
Ricoh Pitch                     : 2.8
```

This line uses ImageMagick to flip the image horizontally.

```
convert -flop $1 tmp.jpg
```

In this section, Koen creates a Povray file.

```
# create povray script with correct image parameters
cat <<EOF > tmp.pov
// Equirectangular Panorama Render
// bare bones script
```

The Povray configuration file is covered in this documentation [http://www.povray.org/documentation/3.7.0/t2_2.html#t2_2].

The main section of the documentation is below.

```
camera {
  location <0, 2, -3>
  look_at  <0, 1,  2>
}
```

The camera statement describes where and how the camera sees the scene. It gives x-, y- and z-coordinates. location <0,2,-3> places the camera up two units and back three units from the center of the ray-tracing universe which is at <0,0,0>. By default +z is into the screen and -z is back out of the screen.

look_at <0,1,2> rotates the camera to point at the coordinates <0,1,2>. A point 1 unit up from the origin and 2 units away from the origin. This makes it 5 units in front of and 1 unit lower than the camera. The look_at point should be the center of attention of our image.

```
sphere {
  <0, 1, 2>, 2
  texture {
    pigment { color Yellow }
  }
}
```

The first vector specifies the center of the sphere. In this example the x coordinate is zero so it is centered left and right. It is also at y=1 or one unit up from the origin. The z coordinate is 2 which is five units in

front of the camera, which is at z=-3. After the center vector is a comma followed by the radius which in this case is two units. Since the radius is half the width of a sphere, the sphere is four units wide.

Povray

This shows image adjustment with Povray.

.pov configuration file

```
#include "colors.inc"
#include "shapes.inc"

camera {
spherical
}

sphere {
  // center of sphere
  <0,0,0>, 1
  texture {
    pigment {
      image_map {
        jpeg "museum.jpg"
        interpolate 2
        once
        map_type 1

    }
   }
   rotate z * 90
   // rotate x * 90
   finish { ambient 1}
}
}
```

command line

```
$ povray +W5376 +H2688 +fj  thetasphere.pov +Othetaspherez90.jpg
```

Figure 29. original image

Figure 30. x rotated 90 degrees

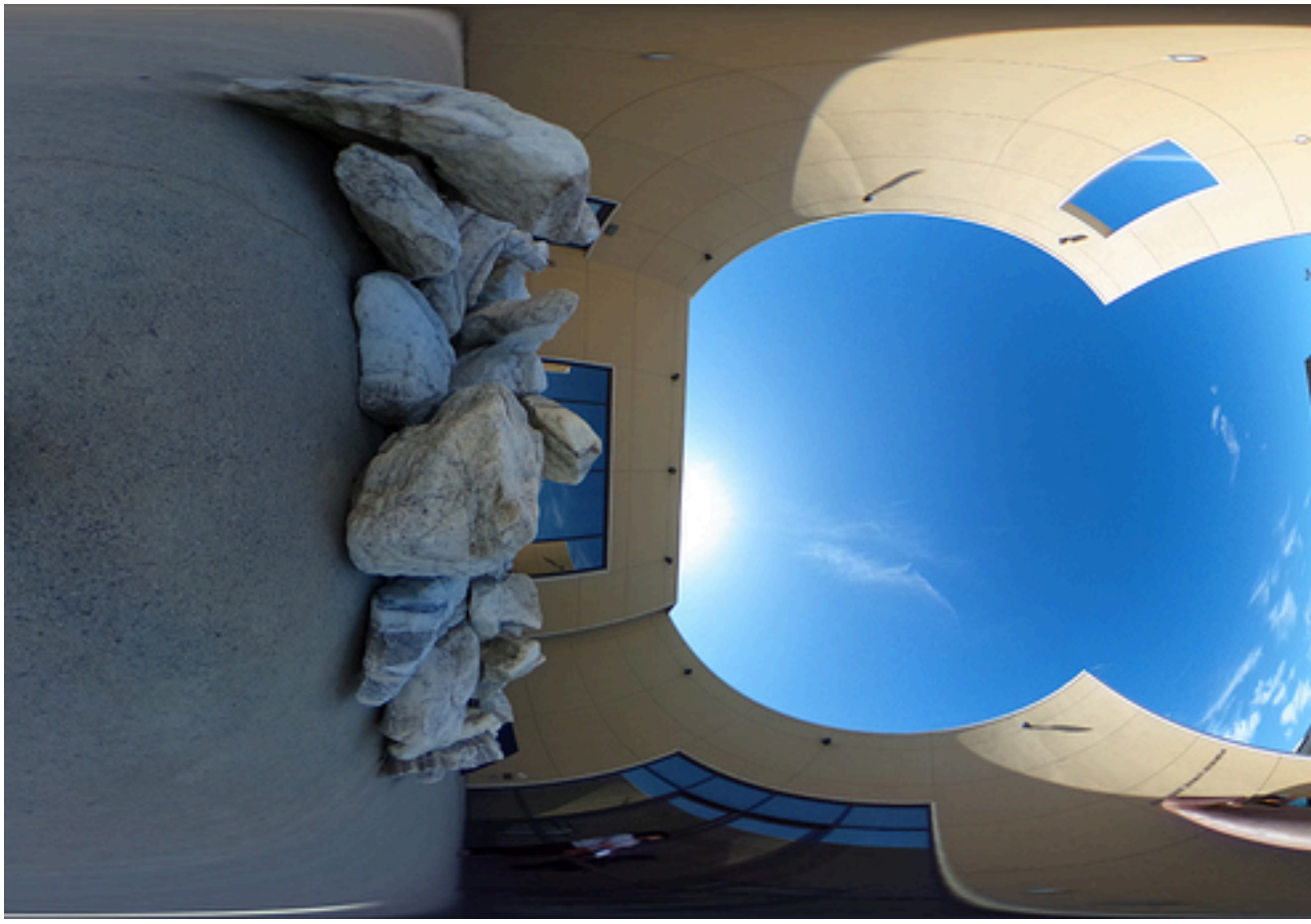


Figure 31. view in THETA desktop app with x rotation

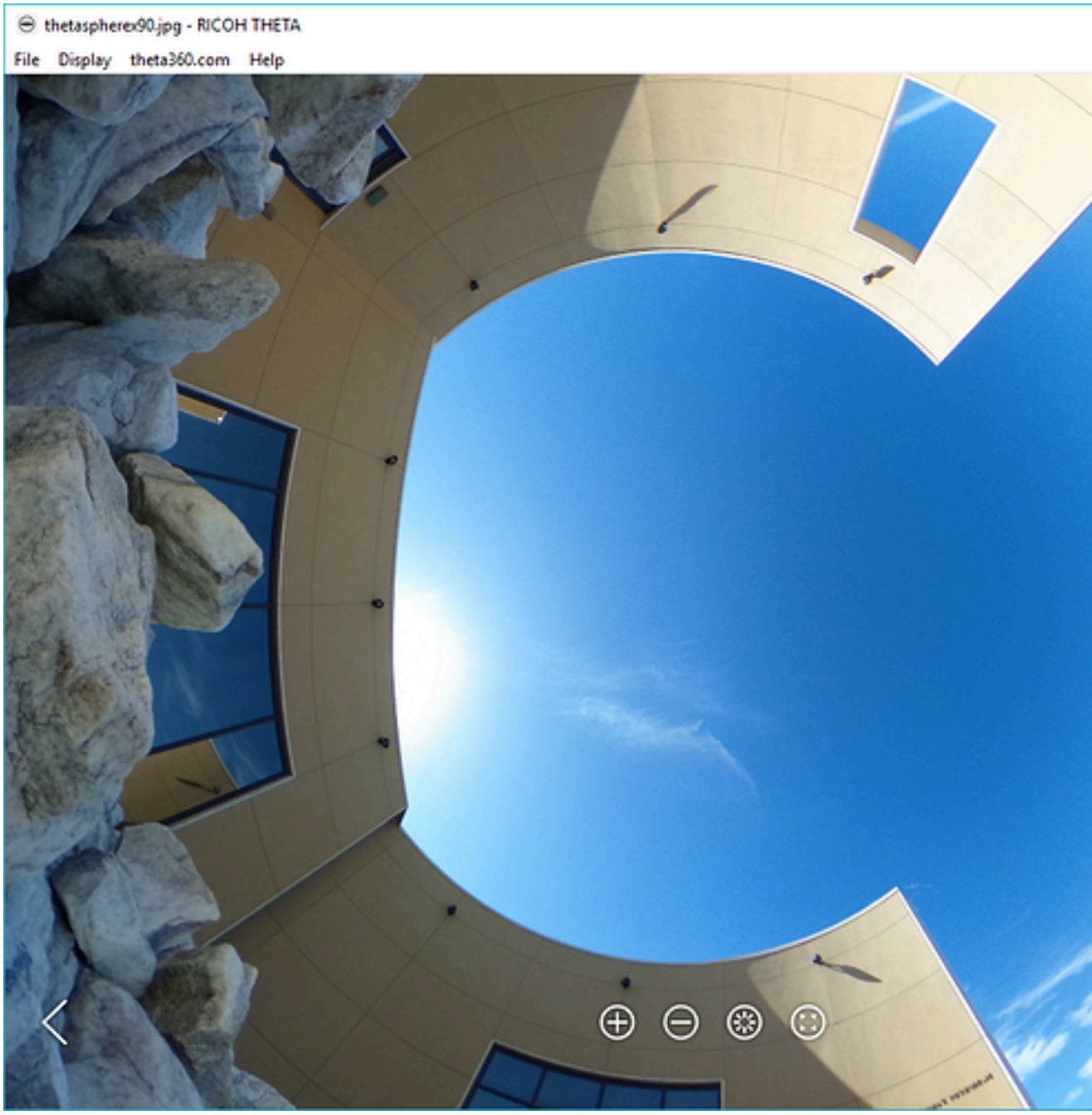


Figure 32. z rotated 90 degrees

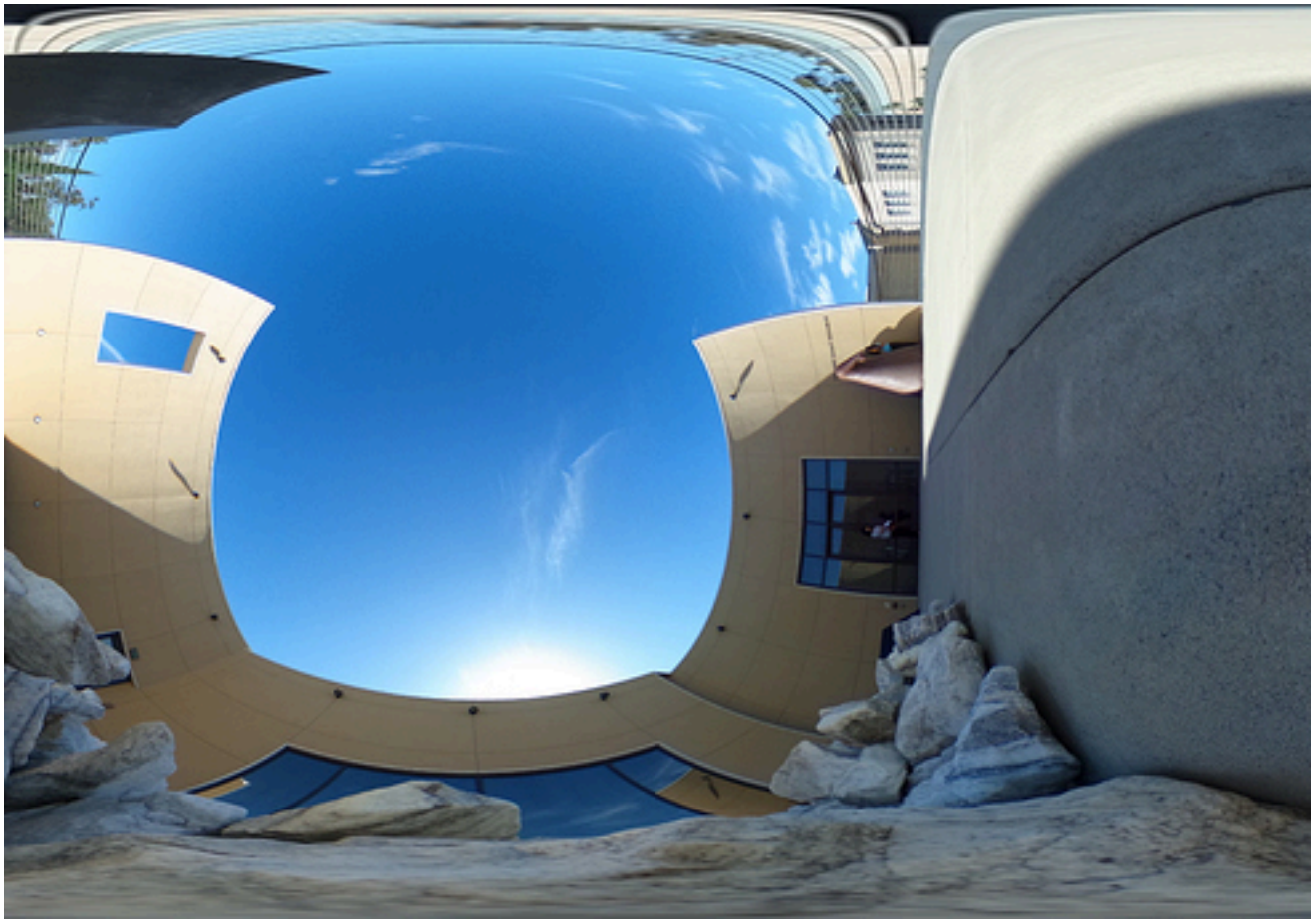
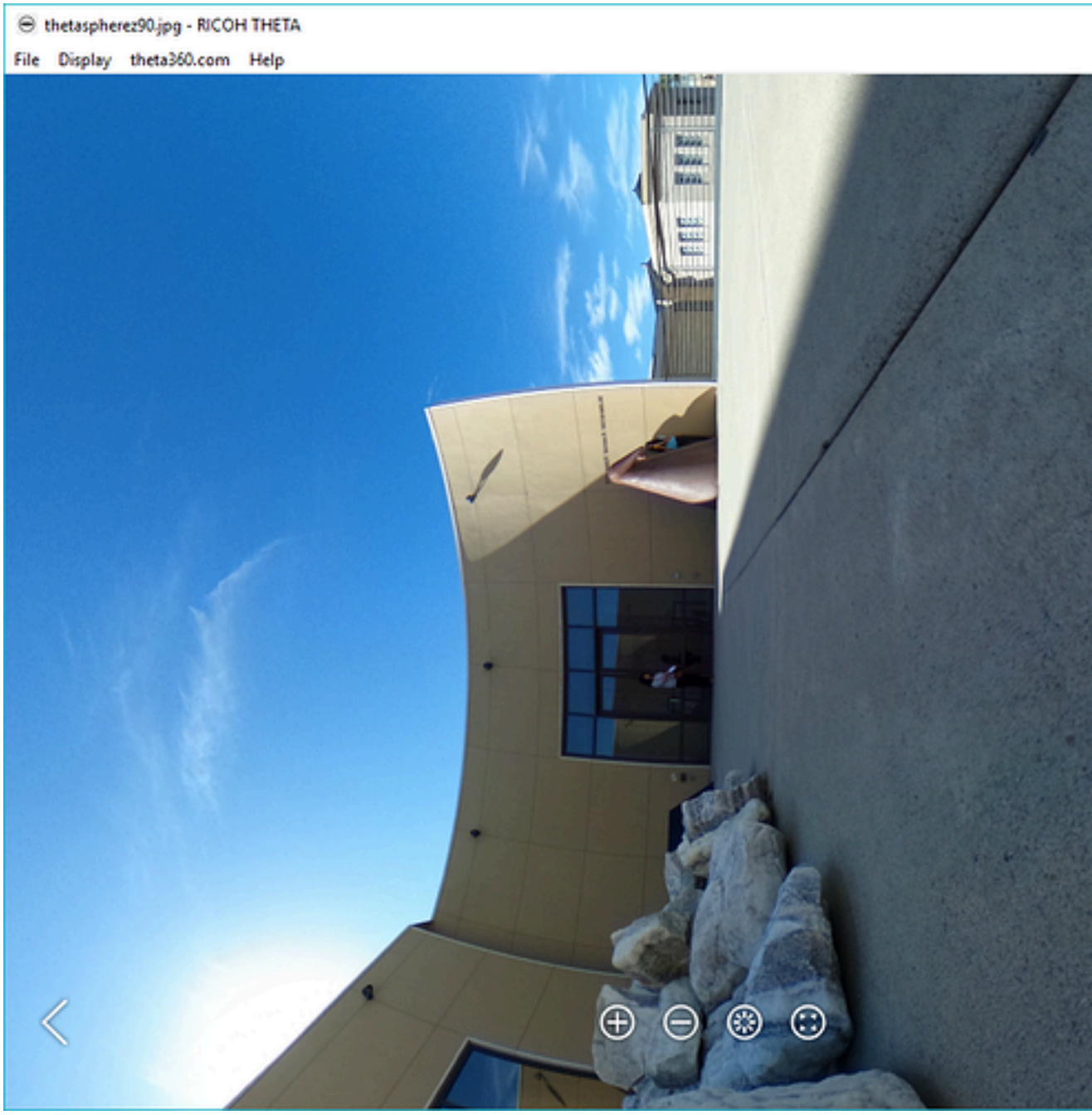


Figure 33. view in THETA desktop app with z rotation



Platforms

aframe

Google VR

Unity

Web sites