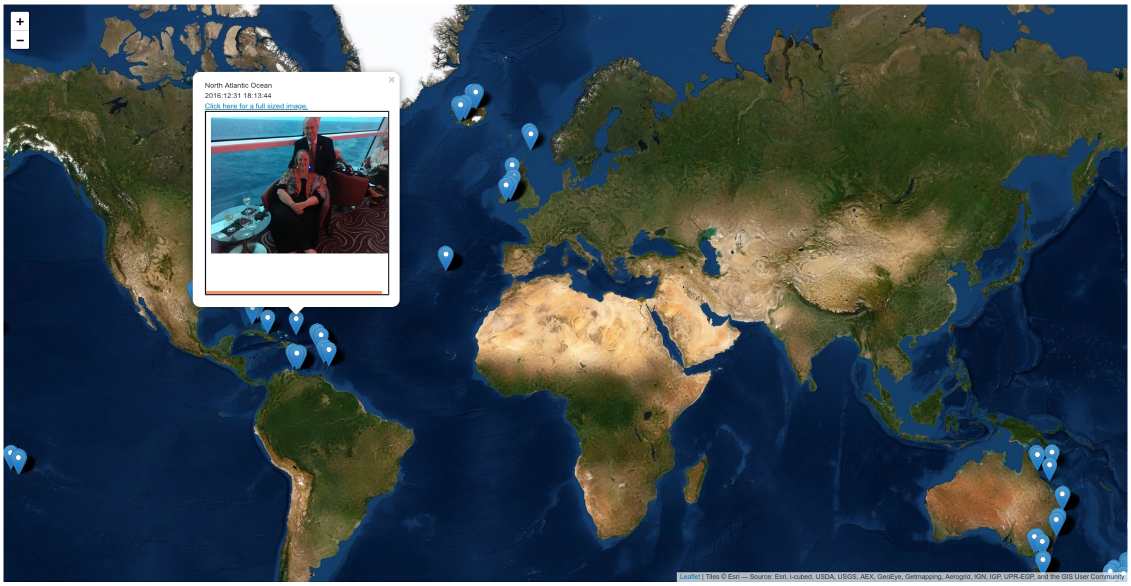


Exploration into Exif Data

Tidewater Big Data Enthusiasts
Chuck Cartledge
Developer

May 10, 2020

Contents

List of Tables ii

List of Figures ii

1 Introduction 1

2 Purpose 1

3 Exploration of data 1

3.1 Image orientation 3

3.2 Times between pictures 7

3.3 Accuracy and precision 10
3.4 Resolution 10
3.4.1 Data 10
3.4.2 Display 13

4 Data reduction **17**

5 Exif UserComment field **17**

6 Linux port usage **19**

7 Results **20**

8 Future work **24**

9 Conclusion **25**

A Miscellaneous files **29**

B References **29**

List of Tables

1 GPS related Exif tags. 2
2 Exif image orientation values. 7
3 Orientation values as reported by images. 8
4 Accuracy vs. Precision truth table. 10
5 Exif data embedded in one image. 20

List of Figures

1 Notional Exif file configuration. 3
2 Evaluating 1,592 GPS recorded time data. 4
3 Evaluating 1,592 GPS positional data. 5
4 Evaluating 1,592 time and GPS positional data. 6
5 Different Exif orientation values. 8
6 Different Exif orientation values as displayed by image viewer. 9
7 Histogram of seconds between images. 11
8 Accuracy versus precision. 12
9 Histogram of meters between images. 14
10 Dendrogram of meters between images. 15

11 Raw and clustered data on a world map. 16
12 A small area with labeled cities. 18
13 A geographically oriented picture album. 25
14 A popup from the album. 26
15 A linked image. 27
16 An image without Exif data. 28

1 Introduction

We explore how Exchangeable image file (Exif) data can be used to create a photo album using the Global Positioning System (GPS) temporal and positional data automatically embedded by many smart phones into Joint Photographic Experts Group (JPEG) images. We look at the quality of the embedded data, how the data is used "behind the scenes" by image browsers to "correct" things, and discover operating system limitations that affect how images can be processed. The end result being a web page JavaScript enabled clickable geographic oriented web page.

2 Purpose

Much of our lives are recorded and maintained on our smart phones. We will be looking at what types of data are recorded in the metadata embedded in the pictures we have taken, and those that we have received. The metadata data is called Exchangeable image file (Exif).

> *"This standard specifies the formats to be used for images, sound and tags in digital still cameras and in other systems handling the image and sound files recorded by digital still cameras."*
>
> JEITA Staff [2]

Notionally, the Exif is embedded at the start of the image file (see Figure 1). We will be focusing on the global positioning system (GPS) data (see Table 1).

The ultimate purpose of this exploration was to determine the feasibility of automating the creation of a map based clickable image showing the travels my wife Mary and I had taken based on the pictures in my phone.

3 Exploration of data

All images were downloaded from my iPhone 6s on 17 April 2020. There were a total of 1,592 JPG images. An assortment of MOV and other miscellaneous files were removed and are not part of the investigation. The images were taken mostly by my camera, some came as part of texts, while the source of others is unknown.

The first order of business was to get a feel for the "quality" of the images. Quality in this context is: can the image be used to temporally, and geographically place the image. Exif has two places where time is recorded; GPSTimeStamp and GPSDateStamp tags. GPSDateStamp is an ASCII string, GPSTimeStamp is a RATIONAL[1]. It is possible to look

[1] A RATIONAL number is computed based on two LONGs. The first LONG is the numerator and the second LONG expresses the denominator. A LONG is a 32-bit (4-byte) unsigned integer.[2]

Table 1: GPS related Exif tags. The tags in bold (ex. **GPSTimeStamp**) are the focus of our interest.

Num.	Tag	Explanation
0	GPSVersionID	GPS tag version
1	**GPSLatitudeRef**	**North or South Latitude**
2	**GPSLatitude**	**Latitude**
3	**GPSLongitudeRef**	**East or West Longitude**
4	**GPSLongitude**	**Longitude**
5	GPSAltitudeRef	Altitude reference
6	GPSAltitude	Altitude
7	**GPSTimeStamp**	**GPS time (atomic clock)**
8	GPSSatellites	GPS satellites used for measurement
9	GPSStatus	GPS receiver status
10	GPSMeasureMode	GPS measurement mode
11	GPSDOP	Measurement precision
12	GPSSpeedRef	Speed unit
13	GPSSpeed	Speed of GPS receiver
14	GPSTrackRef	Reference for direction of movement
15	GPSTrack	Direction of movement
16	GPSImgDirectionRef	Reference for direction of image
17	GPSImgDirection	Direction of image
18	GPSMapDatum	Geodetic survey data used
19	GPSDestLatitudeRef	Reference for latitude of destination
20	GPSDestLatitude	Latitude of destination
21	GPSDestLongitudeRef	Reference for longitude of destination
22	GPSDestLongitude	Longitude of destination
23	GPSDestBearingRef	Reference for bearing of destination
24	GPSDestBearing	Bearing of destination
25	GPSDestDistanceRef	Reference for distance to destination
26	GPSDestDistance	Distance to destination
27	GPSProcessingMethod	Name of GPS processing method
28	GPSAreaInformation	Name of GPS area
29	**GPSDateStamp**	**GPS date**
30	GPSDifferential	GPS differential correction

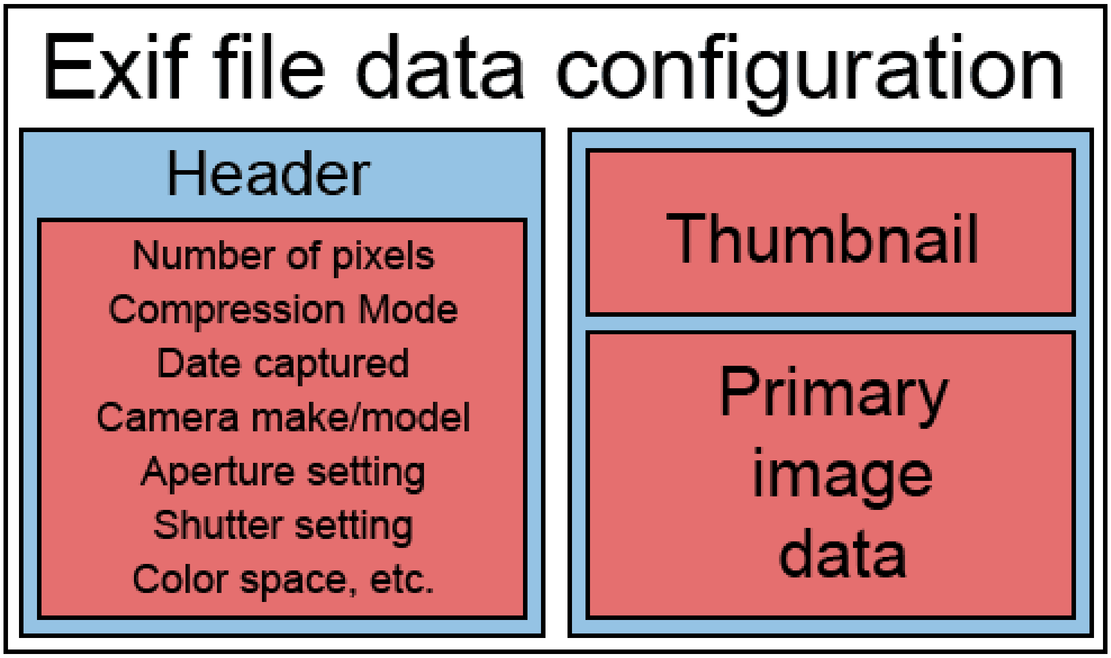


Figure 1: Notional Exif file configuration. Image from [5].

at the bytes in a JPG image and read the GPSDateStamp directly. Software must be used to locate and compute the GPSTimeStamp.

Analysis of the images looking at the presence, or absence of GPSTimeStamp and GPSDateStamp tags was performed (see Figure 2). Approximately 20% of the images had data that was either inconsistent, or incomplete. Analysis of the images looking for the presence, or absence of latitude and longitude data (GPSLatitude and GPSLongitude) was performed (see Figure 3). Positional data in the images was totally absence approximately 19% of the time, there were not any cases where either latitude or longitude was presence, and the other was absent. The presence or absence of positional data will be used to reduce the number of images to be considered. Combining the idea of eliminating images based on lack of positional data, and the presence or absence of time data, gives additional insight into the images (see Figure 4). Of the 1,592 total images, 1,259 (about 79%) have both valid positional and time data.

3.1 Image orientation

A smartphone is not limited in the way it must be orientated to take a picture. Exif records the camera's orientation when a image is created[2]. The orientation value (see Table 2) is used by some image browsing software to "correct" the displayed orientation so that the presented top-left corner is "up." Sample images were taken to demonstrate different values

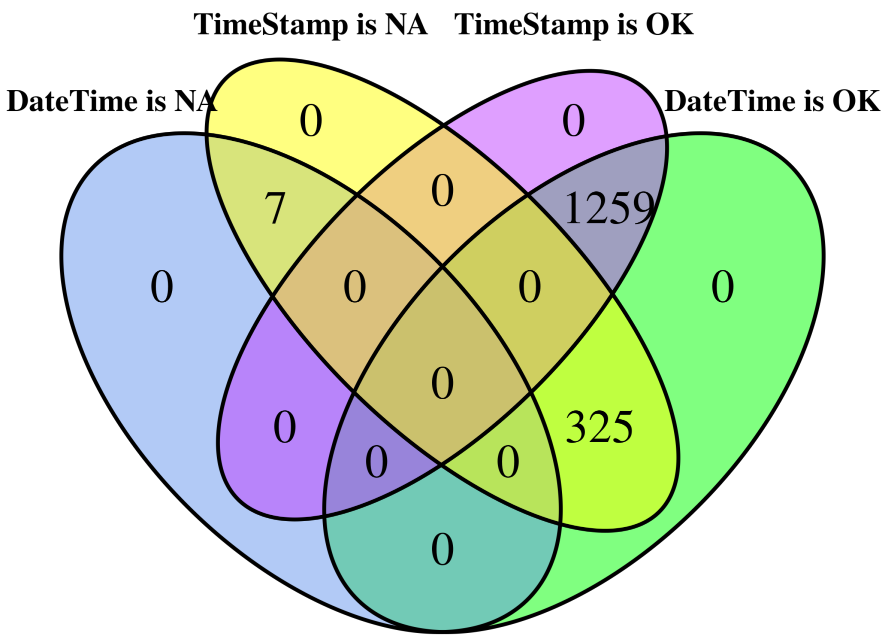


Figure 2: Evaluating 1,592 GPS recorded time data. There are 7 cases where both GPSTimeStamp and GPSDateStamp tags are not available (NA). 1259 cases where both tags are OK. 325 cases where GPSDateStamp is OK and GPSTimeStamp is NA. Approximately 20% are bad.

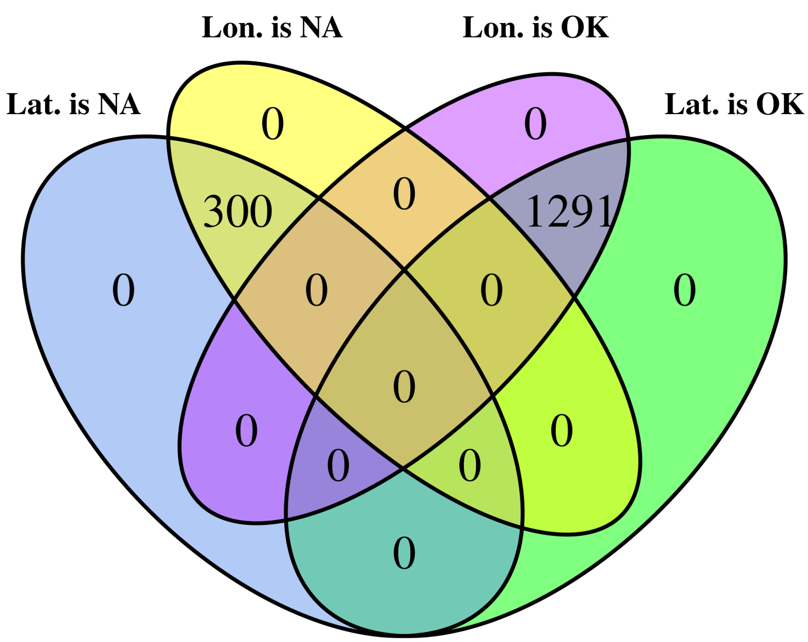


Figure 3: Evaluating 1,592 GPS positional data. Of the 1,592 total images, 300 images did not have positional data. Approximately 19% are bad.

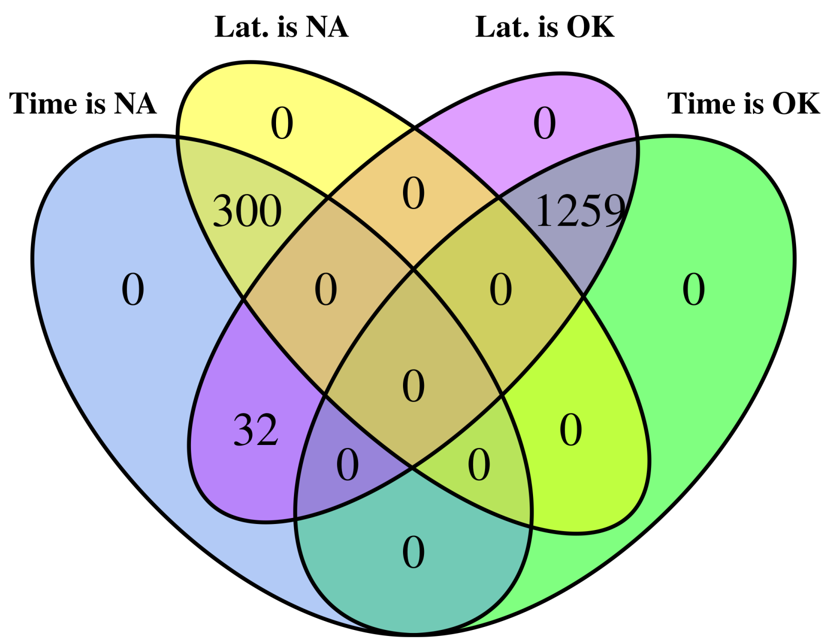


Figure 4: Evaluating 1,592 time and GPS positional data. Of the 1,592 total images, 300 images did not have positional data and were without time data. While 32 had positional data, but were without time data. The remaining images had both time and positional data. Approximately 21% are bad.

Table 2: Exif image orientation values. The orientation is defined as a SHORT. Where a SHORT is a 16-bit (2-byte) unsigned integer[2].

Value	Meaning/interpretation
1	The 0th row is at the visual top of the image, and the 0th column is the visual left-hand side.
2	The 0th row is at the visual top of the image, and the 0th column is the visual right-hand side.
3	The 0th row is at the visual bottom of the image, and the 0th column is the visual right-hand side.
4	The 0th row is at the visual bottom of the image, and the 0th column is the visual left-hand side.
5	The 0th row is the visual left-hand side of the image, and the 0th column is the visual top.
6	The 0th row is the visual right-hand side of the image, and the 0th column is the visual top.
7	The 0th row is the visual right-hand side of the image, and the 0th column is the visual bottom.
8	The 0th row is the visual left-hand side of the image, and the 0th column is the visual bottom.
Other	reserved

of Exif Orientation (see Figure 5). The same images when viewed by an image viewer are often rotated to the "correct" orientation (see Figure 6).

3.2 Times between pictures

Intuitively, the time between images would vary considerably. Even with the innumerable cat and dog images and videos on the internet, after a while a person would get bored taking pictures of the same subject again, and again, and again. We set off to see if that pattern was evident in our data. For those images that had data in the Exif DateTimeOriginal field,

1. We converted the data to seconds from January 1, 1970 (Unix seconds),
2. Sorted the data from low to high,
3. Computed the seconds between successive images, and
4. Created histogram of the time differences with 200 bins.

The resulting histogram met our expectations (see Figure 7). The histogram shows a few cases of greater than 2 months between successive pictures, and a very large number (1,299)

Table 3: Orientation values as reported by images.

Orientation value	**Count**	**%**
1	732	46.0
2	None	
3	128	8.0
4	None	
5	None	
6	718	45.0
7	None	
8	8	0.5
Not Available	6	0.4
Total	1,592	99.9

(a) Orientation value: 6 (portrait)

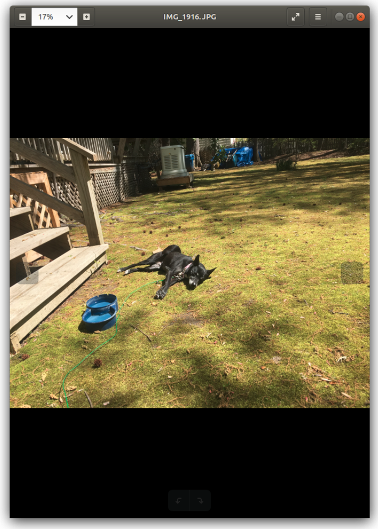
(b) Orientation value: 1 (landscape)

(c) Orientation value: 8

(d) Orientation value: 3

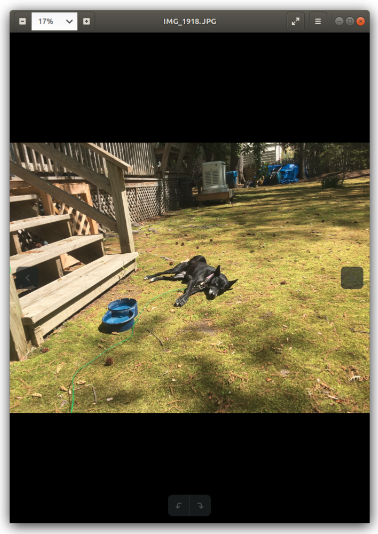
Figure 5: Different Exif orientation values. Four unrotated images taken within 1 minute. Most image viewers automatically rotate the image so that it is orientated correctly.

(a) Orientation value: 6 (portrait)



(b) Orientation value: 1 (landscape)

(c) Orientation value: 8



(d) Orientation value: 3

Figure 6: Different Exif orientation values as displayed by image viewer. Most image viewers automatically rotate the image so that it is orientated correctly.

Table 4: Accuracy vs. Precision truth table. Refer to image (see Figure 8) for visual example.

		Accuracy	
		Low	High
Precision	Low	"Shots" are striking all over the target (reference), and do not have a pattern.	Shots are striking close to the center (reference), but are not closely clustered.
	High	Shots are clustered together, but far from the center (reference).	Shots are clustered close together, and close to the center (reference).

of cases between 0 and approximately 7 hours. The data points to a long time between bursts of images, where lots of images are taken within a very short time. (The "rug" is a 1-D plot of where data exists, that might not be visible on the Y-axis.)

3.3 Accuracy and precision

Accuracy and precision are two important factors to consider when taking data measurements[3]. Both describe different aspects of data, and are important in understanding the totality of the data. Accuracy reflects how close the data is to reality. Precision reflects how reproducible the data is. Because both accuracy and precision can be qualitatively high or low, it is possible to have one attribute be high while the other is low (see Figure 8).

3.4 Resolution

In some aspects, resolution can be akin to precision. Resolution is the smallest attribute that can be resolved. Whether that is time (parts of a second), geographic position (parts of a degree), or smallest addressable part of an image (pixel). Data and displays have resolution, and they can be in conflict.

3.4.1 Data

How accurate is the GPS data as encoded by the Exif data?

> *"It depends. . . . For example, GPS-enabled smartphones are typically accurate to within a 4.9 m (16 ft.) radius under open sky. However, their accuracy worsens near buildings, bridges, and trees. High-end users boost GPS accuracy with dual-frequency receivers and/or augmentation systems. These can enable real-time positioning within a few centimeters, and long-term measurements at*

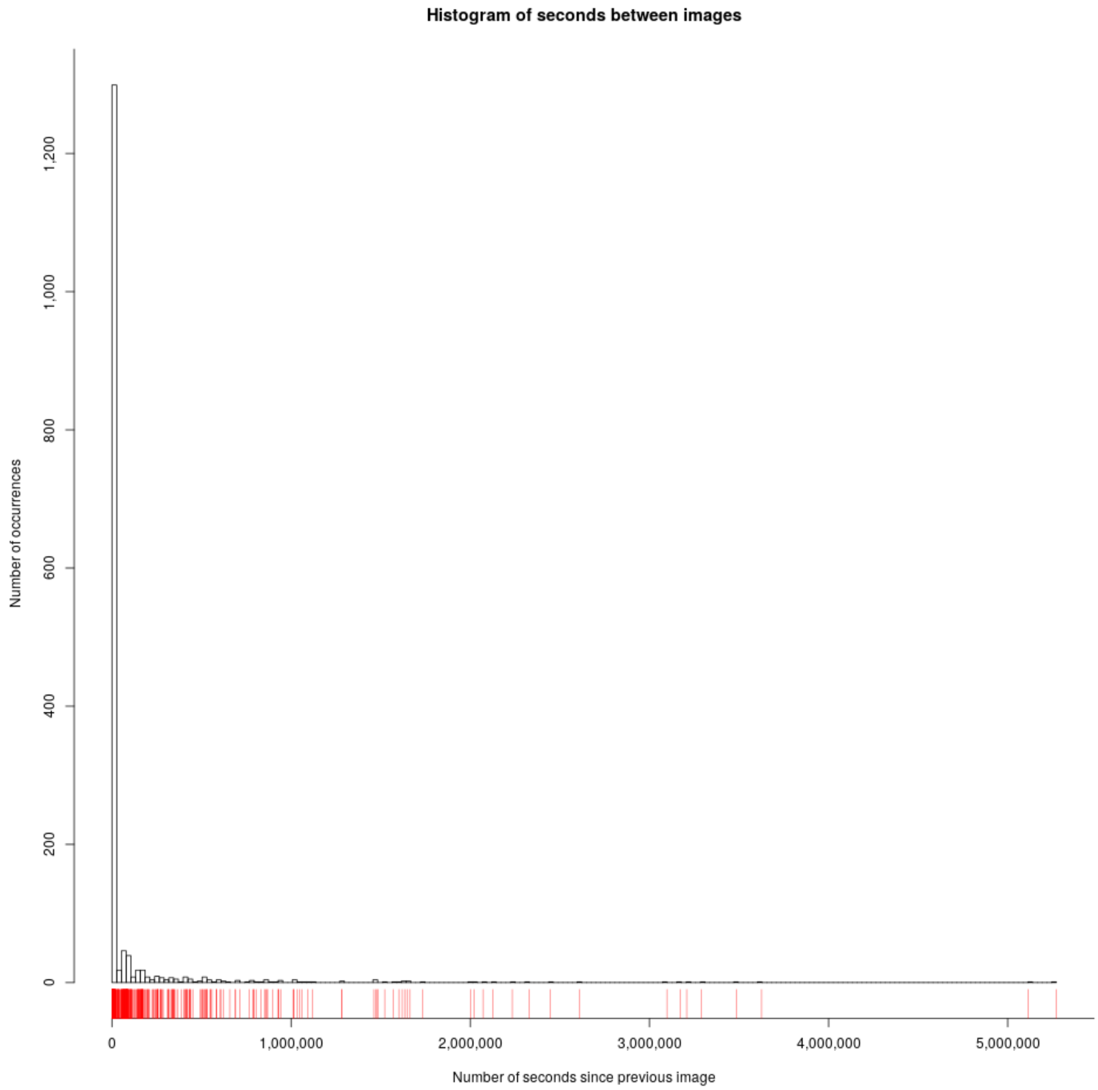


Figure 7: Histogram of seconds between images. The histogram shows a few cases of greater than 2 months between successive pictures, and a very large number (1,299) of cases between 0 and approximately 7 hours. The data points to a long time between bursts of images, where lots of images are taken within a very short time. (The "rug" is a 1-D plot of where data exists, that might not be visible on the Y-axis.)

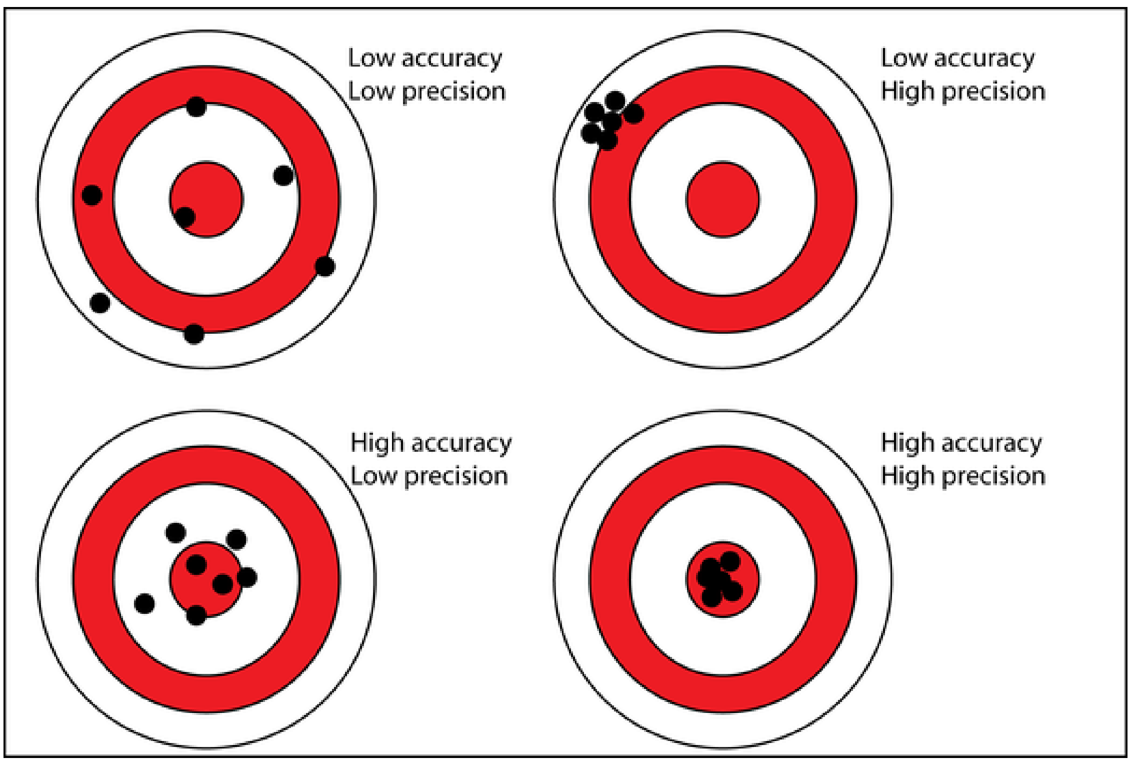


Figure 8: Accuracy versus precision. Image credit[3].

the millimeter level. ... the government commits to broadcasting the GPS signal in space with a global average user range error (URE) of ≤7.8 m (25.6 ft.), with 95% probability."

GPS Staff [1]

In practice, accuracy is how close a measured value is to the actual (true) value, and precision is how close the measured values are to each other[3]. GPS positional data is reported as a RATIONAL number of degrees relative to the Equator, or Prime Meridian. The conversion from degrees to meters along the parallels of latitude is a function of the sine of the latitude, and so will vary from one value at the equator. One degree of latitude, called an arcdegree, covers about 111 kilometers (69 miles)[4]. One degree of longitude is approximately the same length. The earth is not a perfect sphere, so one degree along a longitudinal line will vary in length depending one where it is measured relative to the equator.

4.9 meters represents 0.004% of a degree of latitude along the equator, or along a longitudinal line, so changes in the fifth decimal point of the longitudinal degrees of at least 4 represent a real change in location.

GPS positional data is 2 dimensional, so plotting it as a standard histogram is almost meaningless (see Figure 9), because distances are between each point and every other point. In reality, we would expect the positions to be clustered. By changing our thought processes to more closely match the positional data, we can construct a dendrogam of the positional data, and cut the tree at what we consider to be a reasonable height. The height represents how close the positions would be to be considered to be "close enough" to be the same place (see Figure 10). The positional points are ordered so that entries along the X-axis are points that are close to each other. From the close points a notional center point is computed that represents all of its close points. Then the process is repeated for all the notional points, until there is only one point. The vertical line represent how far the points below it are from each member in the cluster. The red line is an arbitrary height where the tree is cut. Points below the cut line are in the same cluster, and are assumed to be "close" to the same location.

The idea of clustering the data is appealing, but abstract. We would have greater confidence in the process, if we could see some results. So, we plotted both raw and clustered data (see Figure 11). The clustered data fits well with our notional ideas.

3.4.2 Display

An additional limitation dealing with resolution and accuracy is the way in which the data will be displayed. Specifically, it is the mapping of GPS positional data to the pixels where the data will be displayed. Added to that is the resolution of how data will be selected from the display. In theory, a display pointer should have single pixel resolution, but the human operator/viewer does not have that level of acuity. Additionally, there are often "markers"

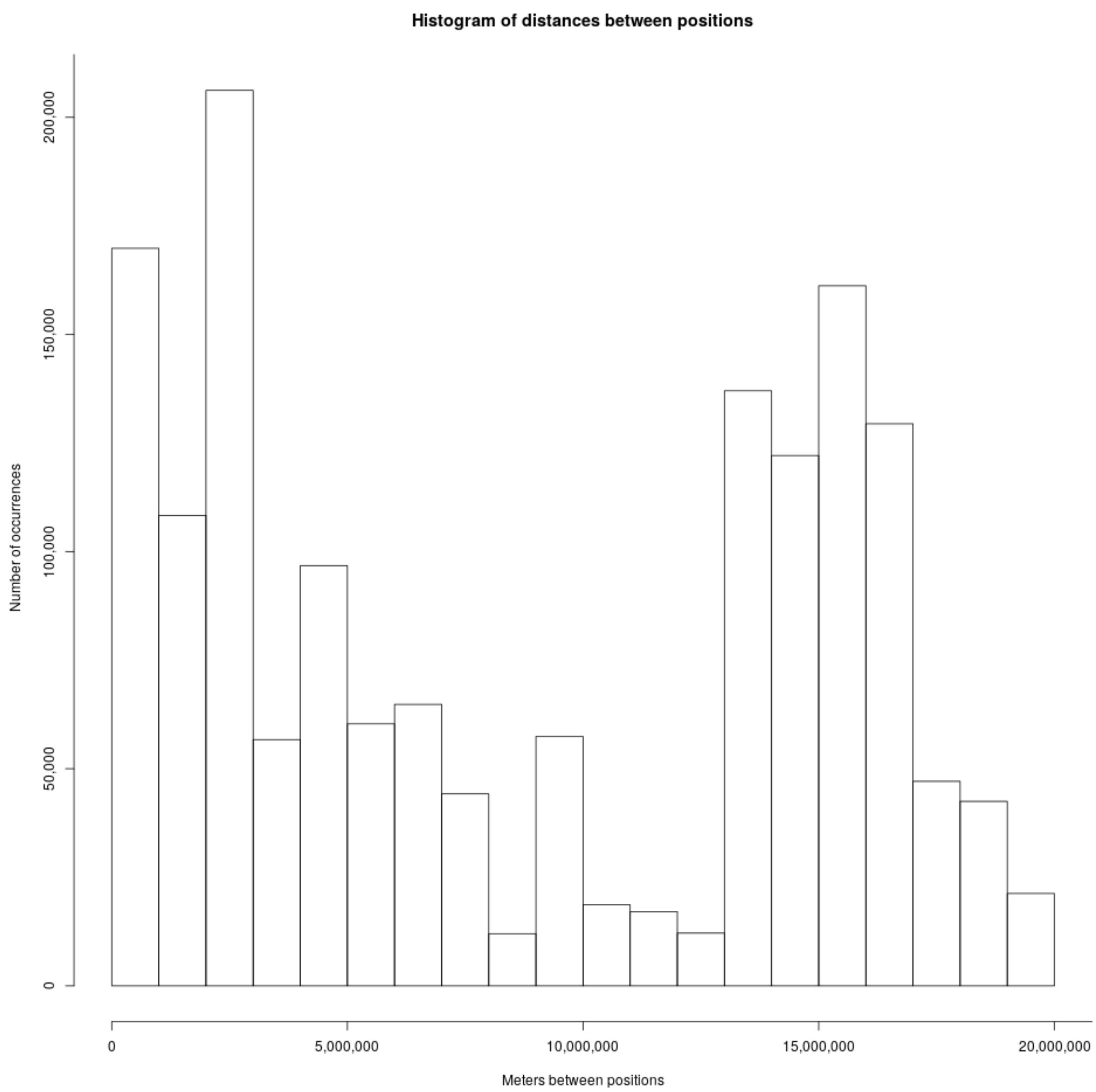


Figure 9: Histogram of meters between images.

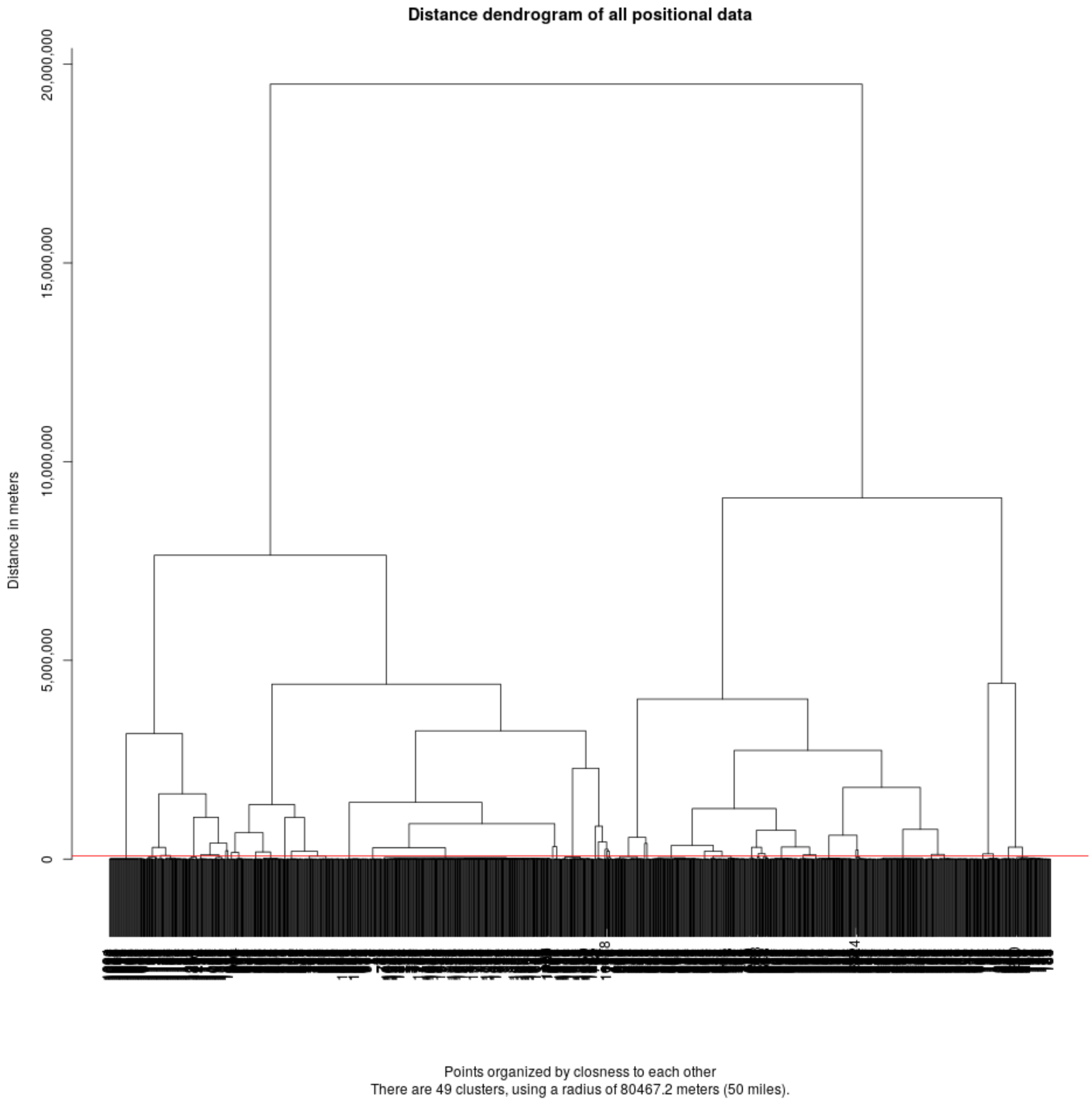


Figure 10: Dendrogram of meters between images. The positional points are ordered so that entries along the X-axis are points that are close to each other. From the close points a notional center point is computed that represents all of its close points. Then the process is repeated for all the notional points, until there is only one point. The vertical line represent how far the points below it are from each member in the cluster. The red line is an arbitrary height where the tree is cut. Points below the cut line are in the same cluster, and are assumed to be “close” to the same location.

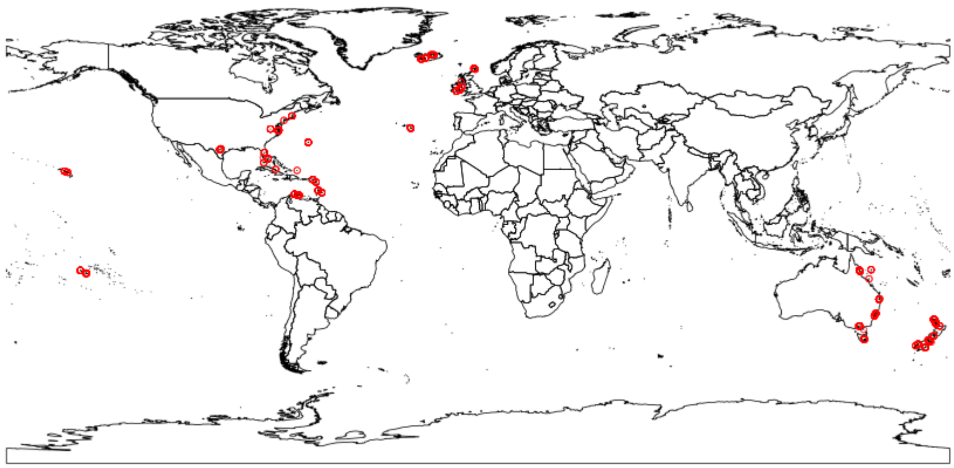


Figure 11: Raw and clustered data on a world map. The purpose of the plot is to provide confidence that the algorithms are preforming reasonably, and in accordance with our feel for the data. Raw image GPS points are plotted in red. Clustered data is plotted as green filled polygons.

added to the display to highlight different things, and these markers are usually much much larger than one pixel. At the limit, different GPS positions may map to the same pixel, markers overlaid on the display will obscure the pixel where the data lives, the mechanics of the pointer pixel selection will return only one marker, so the other obscured markers can not be seen, nor selected.

4 Data reduction

Clustering is a technique to reduce the number of raw data points to something that is more manageable. Clustering was used in the previous section to identify and display positions that were "close" to each other. While those clusters are important, and work well; they don't have the type of attributes that humans are used to dealing with. Positional data was clustered using a 5 mile threshold. The cluster centers were used to identify nearby cities. These representative cluster cities were plotted on a geographic display (see Figure 12).

The conversion from GPS positional data to city names was accomplished via the use of two free services:

- Photon via the `revgeo` library. A free service, so access to it was throttled to be polite. It was able to return cities for most of the positions.
- opencage_reverse via the `opencage` library. The service requires a free key to throttle service. In the background, `opencage` claims to use a version of memoize to limit the number of calls to their service.

Photon was used for the first cut of names for the cluster positions, and stored locally for later use. Unnamed cluster positions were then labeled with the results from opencage. opencage was able to identify locations like the Coral Sea, which is not near any city.

5 Exif UserComment field

The Exif standard supports a field named "UserComment." The type of data is listed as undefined, and must be less than 255 bytes. On a Linux distribution, the Exif data can be viewed using this command:

exiftool IMG_1915.JPG

The UserComment Exif data can be updated using this command:

exiftool -m -UserComment="Created by Chuck Cartledge"
IMG_1915.JPG

Figure 12: A small area with labeled cities.

The UserComment field has the potential to be very useful. Here are a few examples of how the UserComment field can be used/exploited:

- Hiding Malware Inside Images on GoogleUserContent (at `https://blog.sucuri.net/2018/07/hiding-malware-inside-images-on-googleusercontent.html`)
- Malware Hidden Inside JPG EXIF Headers (at `https://blog.sucuri.net/2013/07/malware-hidden-inside-jpg-exif-headers.html`)
- Return of the EXIF PHP Joomla Backdoor (at `https://blog.sucuri.net/2015/11/exif-php-joomla-backdoor.html`)
- Attackers concealing malware in images uploaded to Google servers (at `https://www.scmagazineuk.com/attackers-concealing-malware-images-uploaded-google-servers/article/1488518`)

6 Linux port usage

The R implementation running under Linux appears to have a problem with ports. Then again, it may not be viewed as a problem, just a fact of life because a port can be bidirectional and should remain open until explicitly closed. When the using the `revgeo` function, it appears that a port is opened to the web site to pass and retrieve data. This port is not closed when the function call is completed. So, the port is considered open, even when there is no expected future use for that same port during the program execution.

Under the Linux Operating System (at least Ubuntu 18.04 that I am using), the number of file descriptors that an application can have open at any one time is 1,024. By default, the number is found by using the "`ulimit -n`" command at the command line interface. This value can be changed using different tools either permanently, or for the current session. The important things is that there is a fixed number per session, and there is a potentially unknown number of file descriptors that will be needed by the `revgeo` function.

The solution to this problem is to partition the number of times `revgeo` will be called per process to something lower than the limit returned by the `ulimit` command. That is what the `revgoWorker.R` and `revgoWorkerSub.R` scripts do. `revgoWorker.R` is a standalone script called in the background by the main script (`test05.R`) and given a full list of positions that need servicing. `revgoWorker.R` partitions that list into blocks of at most 200 positions, and passes them onto the second standalone script `revgoWorkerSub.R`. `revgoWorkerSub.R` uses up to 200 file descriptors (i.e., ports) while it is running, and when it dies, those file descriptors (ports) are returned to the OS. `revgoWorkerSub.R` never approaches the `ulimit` limit even though it is called multiple times. From the main script's perspective, it passes a list of positions, and receives back a list of positions.

Failure to partition the use of file descriptors in this manner can result in interesting program behavior. If the main script uses all it's file descriptors, any attempt to use additional descriptors results in default error messages about "too many files being open" with any indication of where the file descriptors were consumed. This error message can come from asking for help about a command (e.g. `?readLines`), from attempting to read a file (e.g. `readLines(''foo.txt'')`), or saving data (e.g. `save(foo, file=''foo.rds'')`). Closing the R session, and rerunning the same code will allow the application to work, because the new session will start without the old file descriptors.

7 Results

The end result of all the analysis and data wrangling, was a JavaScript enabled clickable web page showing where the images were taken (see Figure 13). Clicking on a marker returns a popup (see Figure 14). The popup contains the image associated with that location, and a link to see a full scale image (see Figure 15). Exif data embedded in the image is presented (see Table 5). The R `leaflet` function creates the clickable page as a collection of files, and loads the HTML into the default browser. Different browsers display the pages and popups in different ways. In some browsers, the popup shows a reduced image. In others, the image is full size, and you have to pan around the image to see what it is. Some browsers do not support JavaScript, so nothing is displayed.

The R `leaflet` function does not appear to have options to make the same collection of files compatible across multiple browsers. While the raw JavaScript is available in the `leaflet` files, there wasn't enough interest in making execution compatible across all browsers.

Table 5: Exif data embedded in one image. Data as reported by ExifTool Version Number 10.80.

Exif field name	**Value**
File Size	1997 kB
File Modification Date/Time	2020:05:10 15:41:07-04:00
File Access Date/Time	2020:05:10 15:41:07-04:00
File Inode Change Date/Time	2020:05:10 15:41:07-04:00
File Permissions	rw-r–r–
File Type	JPEG
File Type Extension	jpg
MIME Type	image/jpeg
Exif Byte Order	Big-endian (Motorola, MM)

(Continued on the next page.)

Table 5. (Continued from the previous page.)

Exif field name	**Value**
Make	Apple
Camera Model Name	iPhone 6s
Orientation	Horizontal (normal)
X Resolution	72
Y Resolution	72
Resolution Unit	inches
Software	10.1.1
Modify Date	2016:12:31 18:13:44
Y Cb Cr Positioning	Centered
Exposure Time	1/15
F Number	2.2
Exposure Program	Program AE
ISO	800
Exif Version	0221
Date/Time Original	2016:12:31 18:13:44
Create Date	2016:12:31 18:13:44
Components Configuration	Y, Cb, Cr, -
Shutter Speed Value	1/15
Aperture Value	2.2
Brightness Value	-1.8708577
Exposure Compensation	0
Metering Mode	Multi-segment
Flash	Auto, Did not fire
Focal Length	4.2 mm
Subject Area	2004 1017 237 238
Run Time Flags	Valid
Run Time Value	373112635431083
Run Time Epoch	0
Run Time Scale	1000000000
Acceleration Vector	-0.9213895232 0.03489909734 -0.400359874

(Continued on the next page.)

Table 5. (Continued from the previous page.)

Exif field name	**Value**
Content Identifier	03BD167D-32D6-4BFA-96F6-4F6EA304C5BB
Sub Sec Time Original	675
Sub Sec Time Digitized	675
Flashpix Version	0100
Color Space	sRGB
Exif Image Width	4032
Exif Image Height	3024
Sensing Method	One-chip color area
Scene Type	Directly photographed
Exposure Mode	Auto
White Balance	Auto
Focal Length In 35mm Format	29 mm
Scene Capture Type	Standard
Lens Info	4.15mm f/2.2
Lens Make	Apple
Lens Model	iPhone 6s back camera 4.15mm f/2.2
GPS Latitude Ref	North
GPS Longitude Ref	West
GPS Altitude Ref	Above Sea Level
GPS Time Stamp	23:13:44.24
GPS Speed Ref	km/h
GPS Speed	10.92
GPS Img Direction Ref	Magnetic North
GPS Img Direction	114.2734694
GPS Dest Bearing Ref	Magnetic North
GPS Dest Bearing	114.2734694
GPS Date Stamp	2016:12:31
GPS Horizontal Positioning Error	10 m
Compression	JPEG (old-style)
Thumbnail Offset	2078

(Continued on the next page.)

Table 5. (Continued from the previous page.)

Exif field name	Value
Thumbnail Length	10734
XMP Toolkit	XMP Core 5.4.0
Region Area Y	0.335500
Region Area W	0.059000
Region Area X	0.497500
Region Area H	0.079000
Region Area Unit	normalized
Region Type	Face
Region Extensions Angle Info Yaw	0
Region Extensions Angle Info Roll	0
Region Extensions Confidence Level	1000
Region Extensions Time Stamp	8954696850620
Region Extensions Face ID	4
Region Applied To Dimensions H	3024
Region Applied To Dimensions W	4032
Region Applied To Dimensions Unit	pixel
Image Width	4032
Image Height	3024
Encoding Process	Baseline DCT, Huffman coding
Bits Per Sample	8
Color Components	3
Y Cb Cr Sub Sampling	YCbCr4:2:0 (2 2)
Aperture	2.2
GPS Altitude	49.9 m Above Sea Level
GPS Date/Time	2016:12:31 23:13:44.24Z
GPS Latitude	21 deg 35’ 29.36” N
GPS Longitude	69 deg 3’ 53.79” W
GPS Position	21 deg 35’ 29.36” N, 69 deg 3’ 53.79” W
Image Size	4032x3024
Megapixels	12.2

(Continued on the next page.)

Table 5. (Continued from the previous page.)

Exif field name	Value
Run Time Since Power Up	4 days 7:38:32
Scale Factor To 35 mm Equivalent	7.0
Shutter Speed	1/15
Create Date	2016:12:31 18:13:44.675
Date/Time Original	2016:12:31 18:13:44.675
Thumbnail Image	(Binary data 10734 bytes, use -b option to extract)
Circle Of Confusion	0.004 mm
Field Of View	63.7 deg
Focal Length	4.2 mm (35 mm equivalent: 29.0 mm)
Hyperfocal Distance	1.82 m
Light Value	3.2

(Last page.)

8 Future work

The current/embedded R scripts demonstrate that a clickable, JavaScript enabled photo album can be created and demonstrated. There are a number of ways that the end product can be improved, including:

1. Rework the leaflet file structure so that it is transportable. The current implementation exists only so long as the R session is active, and is destroyed when the session ends.

2. Currently, each popup window has exactly one image. It would make more sense to have multiple images based on the number of original positions that were collected into the currently selected cluster.

3. Allow the user to select different cluster sizes when creating the leaflet.

4. Order the images in the popup by time.

5. Use the Exif embedded thumbnails vice full sized images.

6. Have the size of the cluster be determined dynamically by the current geographic zoom factor. The number of clusters could remain fixed, or under user control.

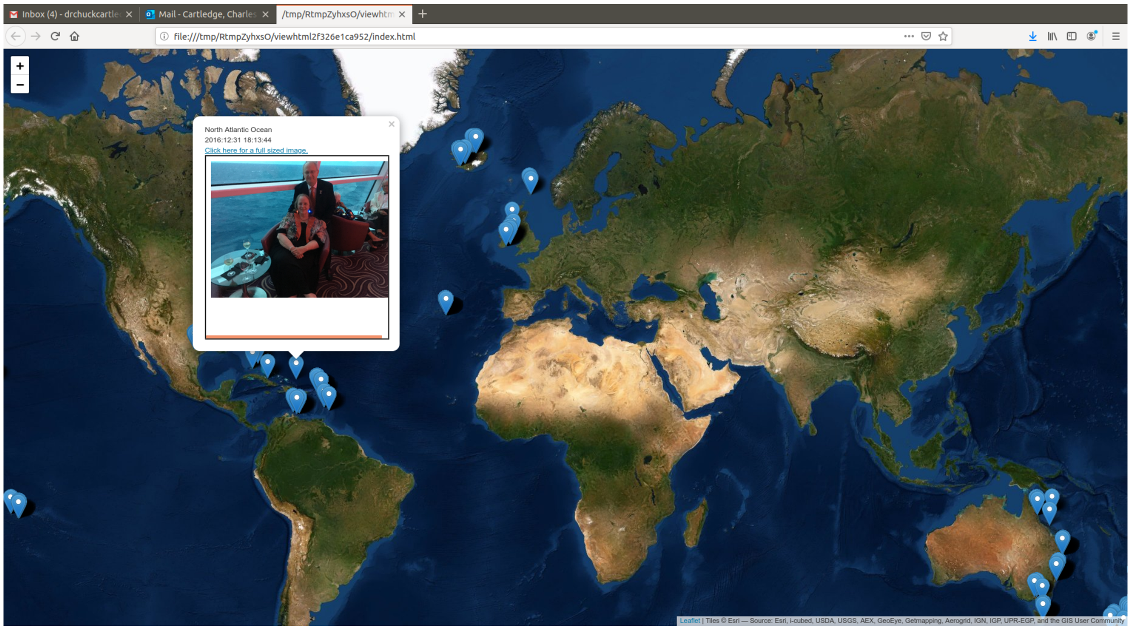


Figure 13: A geographically oriented picture album.

9 Conclusion

We investigated 1,592 images. almost all created by an iPhone 6s over the course of several years. Not all cameras insert Exif data, of any kind into the image (see Figure 16).

For all images:

- Approximately 19% had bad or non-existent GPS positional data.
- Approximately 21% had either bad or non-existent GPS temporal or positional data.
- Approximately 46% were taken in landscape mode.

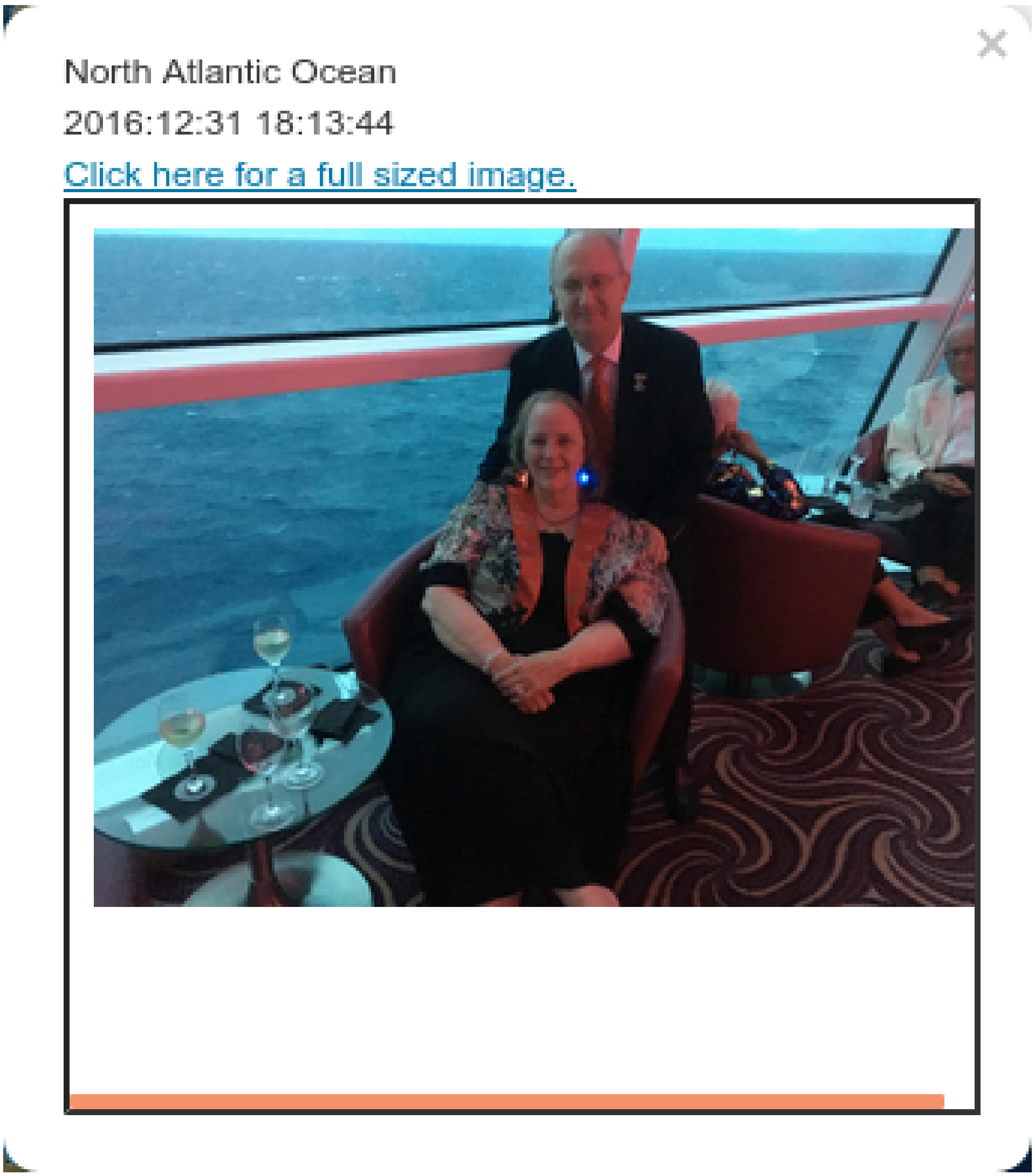


Figure 14: A popup from the album.

Figure 15: A linked image.

Figure 16: An image without Exif data. Based on the clothes and the type of location, the picture was probably taken onboard a Celebrity cruise ship either April or August 2004, in either the North Atlantic or Baltic Sea. The camera used to take the image did not support Exif data.

A Miscellaneous files

A collection of miscellaneous files mentioned in the report.

- IMG-0560.JPG – A sample image with Exif data.
- library.R – R support library file.
- P8150008.jpeg – A sample image without Exif data.
- revgoWorker.R – R script to control the `revgoWorkerSub.R` script.
- revgoWorkerSub.R – Accesses the Photon geocoder service to convert positional data to city name.
- test05.R – R script used to analyze the Exif data, and generate images.

B References

[1] GPS Staff, *Gps accuracy*, `https://www.gps.gov/systems/gps/performance/accuracy/`, 2020.

[2] JEITA Staff, *Exchangeable image file format for digital still cameras: Exif version 2.2*, Japan Electronics and Information Technology Industries Association (2002).

[3] Math is Fun Staff, *Accuracy and precision*, `https://www.mathsisfun.com/accuracy-precision.html`, 2020.

[4] National Geographic Staff, *Latitude*, `https://www.nationalgeographic.org/encyclopedia/latitude/`, 2020.

[5] PhotoMetaData Staff, *Standards: Exif*, `https://www.photometadata.org/META-Resources-metadata-types-standards-Exif`, 2020.