Webology, Volume 16, Number 1, June, 2019

Efficiency of Web Crawling for Geotagged Image Retrieval

Nancy Fazal

PhD Candidate, School of Computing, University of Eastern Finland, Finland.
E-mail: fazal@cs.uef.fi

Khue Q. Nguyen

Master's degree in Information Technology (IMPIT), School of Computing,
University of Eastern Finland, Finland. E-mail: khuenq.hjc@gmail.com

Pasi Fränti

Professor, School of Computing, University of Eastern Finland, FINLAND.
ORCID: 0000-0002-9554-2827. E-mail: franti@cs.uef.fi

Received April 29, 2019; Accepted June 25, 2018

Abstract

The purpose of this study was to find the efficiency of a web crawler for finding geotagged photos on the internet. We consider two alternatives: (1) extracting geolocation directly from the metadata of the image, and (2) geo-parsing the location from the content of the web page, which contains an image. We compare the performance of simple depth-first, breadth-first search, and a selective search using a simple guiding heuristic. The selective search starts from a given seed web page and then chooses the next link to visit based on relevance calculation of all the available links to the web pages they contain in. Our experiments show that the crawling will find images all over the world, but the results are rather sparse. Only a fraction of 6845 retrieved images (<0.1%) contained geotag, and among them only 5 percent were able to be attached to geolocation.

Keywords

Location-based application; GPS; Web crawler; Location photos; Web application

Introduction

In the last decade, there has been a significant growth in mobile device usage, in which the smartphone is the most popular device. According to statistics collected by eMarketer (Statista, 2016), the total number of smartphone users is expected to increase from 2.1 billion in 2016 to more than 2.8 billion in 2020. Tracking the geographical location is considered as one of the most useful feature of a smartphone in comparison to all other features (Khan et al., 2013). It is used for navigation, neighborhood search, security, traffic updates, weather alerts, exercise monitoring, prevent people from getting lost, serve as a mobile tourist guide for cultural heritage sites (Panou et al., 2018), improve the safety of women (Krishnamurthy et al., 2017), avoid motorcycle crashes (Naranjo et al., 2017) and predict the risk of wildlife attacks (Ruda et al., 2018). It is considered so important feature that efforts are even made to have a location available indoors (Kuo et al., 2014) and on-road (Wang et al., 2014) when GPS satellites are not reachable.

The ability to get location by a smartphone has the consequence that more and more pictures will have location embedded automatically. The location is either explicitly stored as side information or saved directly to the image's *metadata* (Rahman & Nayeem, 2018). While several types of image metadata exist, the most commonly used is *Exchangeable Image File Format* (*EXIF*). This format is not limited for storing the coordinates, but several other factors are also stored, which can be helpful to determine whether a picture is authentic (Alvarez, 2004), (Baggili et al., 2014), help to provide text summarization of the image content (Lloret et al., 2008), and copyright protection (Huang et al., 2008). Figure 1 shows an example of EXIF metadata with location information included in a photo. For technical details of the EXIF, we refer to (Orozco et al., 2015).

EXIF DATA:	
Camera make:	Apple
Camera model:	iPhone 4
Data and time:	4.9.2011 12:51:11
Image size:	800 × 598
Shutter speed:	1/3016
Focal length:	3.9mm
File Type:	JPEG
GPS Latitude:	38 deg 54' 35.40" N
GPS Longitude:	1 deg 26' 19.20" E
GPS Altitude:	0 m Above sea level
Light Value:	14.9

Figure 1. Example of an image and its EXIF metadata

The process of adding geographic information to the metadata is called *geotagging*. Geographical coordinates, known as *latitude* and *longitude*, are the minimum required location information needed for geotagging.

Applications that are based on the geotagging are called *location-based applications* (Quercia et al., 2010). We consider two types of applications: location-based service and location-based game. *Location-based service* (LBS) is a position dependent service that can be easily found by its described location (Rainio, 2001) and is usually about the places and objects for practical reasons like business and tourism. *Location-based game* (LBG) is a type of pervasive game whose gameplay evolves and changes the game experience based on the location. Most modern location-based games are implemented for a mobile device with the extensive use of GPS sensor to determine the location. Some notable games on the market recently are *Pokémon* GO, *CodeRunner*, *Ingress* and *Zombies Run*.

(Fränti et al., 2017) introduced O-Mopsi, which is a location-based game based on the classical concept of orienteering. In O-Mopsi, a game is created by specifying a set of targets in the form of geotagged photos of real-world locations, for the user to visit in order to complete a game (Tabarcea et al., 2013). Photos are an important clue for the users to identify the targets. The mobile client plots the targets on a map, displays compass data and gives audio clue in different pitch and frequency based on the distance between the player and the target. The main challenges for the players are planning the tour, and then navigating through the targets in the real environment (Sengupta et al., 2018).

O-Mopsi requires its players to move around the real-world locations. To create game scenarios, the biggest challenge is to collect geotagged photos to be used as targets in the game. Currently the game relies on its users and system administrators for uploading geotagged photos manually, which is time-consuming. To automate the process of collecting the geotagged photos, the following strategies can be considered:

- Crowdsourcing
- Use existing datasets of geotagged data
- Photo sharing web sites (API's)
- Collect material by web mining.

Manual collection is slow and would be better done by crowdsourcing. However, there should be enough users willing to contribute by uploading material for others to play. Quality control of the uploaded material would then also be needed.

The second alternative is to use existing datasets of geotagged photos. Some of the notable datasets introduced by research community includes Div150Cred, MIRFlickr-1M, MIRFlickr25000, NUS-Wide, Paris500K and Yahoo Flickr Creative Commons 100 Million Dataset covering different regions of the world within certain time spans. Lots of information is readily available on the Web but no collected publicly available database exists that would cover the entire world.

Third alternative is to use API support from different photo sharing web sites such as Flickr,

Unsplash, Instagram, SmugMug, Pexels, Pixabay and Shutterstock. O-Mopsi currently uses its own manually collected photos, and data from *ViewOnCities*, which has a good quality geotagged photo collection, but it covers only a few dozens of cities mostly in Europe.

In this paper, we focus on the fourth alternative: web mining. For this purpose, we have developed a system called *Mopsi Image Crawler* (MIC) to collect geotagged photos on the Internet. The system is based on the concept of a *web crawler*, a software application that systematically browses the Web looking for suitable content.

We study the architecture and working mechanism of MIC in detail. We report experimentally, how successful this approach can be by analyzing its performance when run with three different algorithms. To find missing location information of images using geo-information retrieval (GIR) techniques is also studied.

Web Crawler

A *Web crawler* is defined as a system that starts from a set of input web pages, called *seeds*, and then downloads all the other pages linked from it. As a link, we mean a URL that contains either http:// or https://. Other links such as *FTP* and *SMB* are excluded from the search. All the new pages are analyzed to gather information, and the process then continues using the obtained pages as new seeds, which are stored in the queue (Mirtaheri et al., 2013). The crawler basically works on the principle of a simple graph search algorithm, such as *breadth-first search* (BFS) and *depth-first search* (DFS) (Singh et al., 2014), assuming that all the web pages are linked together and that there are no redundant links (Markov & Larose, 2007).

Web crawlers are almost as old as the Web itself (Najork, 2009). They have been written since 1993 and hold a very interesting and long history (Mirtaheri et al., 2013). They were originally designed to collect statistics and information about the Web. Four different web crawlers were introduced in 1993: *World Wide Web worm*, *World Wide Web wanderer*, *jump station* (McBryan, 1994) and *RBSE spider*.

A web crawler is known by different names such as *web spider*, *ant*, *automatic indexer* (Kobayashi & Takeda 2000), or *web scutter* in *friend of a friend* (FOAF) software context. It is a system for downloading the bulk of web pages (Olston & Najorak, 2010) or a program that retrieves web pages (Cho et al., 1998). It is further defined as "*a tool to download web pages or their partial content in an automated manner*" (Elyasir & Anbananthen, 2012). A web crawler can be classified into two main types based on its functionality:

- Generic web crawler, and
- Focused web crawler.

Even though both crawler types have a similar working mechanism they are fundamentally

different in the order of web pages they choose to visit. Such order is determined by the algorithm used for implementing the crawler.

1. Architecture of web crawler

The general working mechanism of a web crawler is described as follows: the crawler receives a list of links as input, also known as the *seeds*, and adds them into the priority queue. Standard web crawler consists of the four main components (Castillo, 2004) illustrated in Figure 2:

- The queue
- The downloader
- The scheduler
- The storage

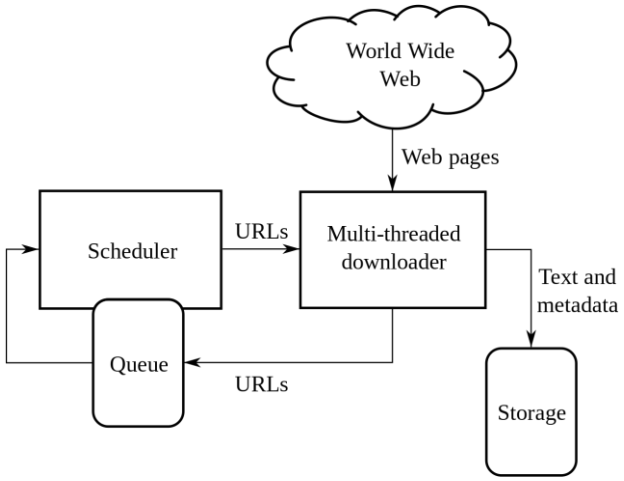

Figure 2. High-level architecture of a standard web page [Castillo, 2004]

The *queue* is a data structure that stores a list of links. In practice, the queue can either be a priority queue or a normal queue. It adds new links and retrieves the next link from queue to continue the crawling process. The actual crawling task is performed by the *downloader* and is known as the essential component of the crawling system. It is a program that carries out breadth-first search, depth-first search or similar approaches to explore and download the content.

The *scheduler* is a program which decides when the next crawling task should happen and ensures the adequate computing resources for the crawling process to continue. The *storage* is a data structure used for storing and managing result data from the crawling process. The result data can be text content, multimedia resources or metadata.

It is very likely that crawler can visit the same link multiple times, as web pages are connected interchangeably. Moreover, there also exists the possibility to have multiple links to the same web page. Thus, a crawler can maintain a cache of links or page content to check content similarity between two web pages (Markov & Larose, 2007).

2. Mopsi image crawler

In this work, we use a focused crawler called *Mopsi Image Crawler* (MIC), which is designed to download geotagged photos. The downloaded photos are then used for the content creation process of O-Mopsi game. The high-level architecture of MIC and interaction between its components is shown in Figure 3.

In general, the system starts once the scheduler has initialized some scheduled crawling task. A predefined seed is sent to the top of the queue, which is then chosen as an initial input for the downloader. The downloader requests for a web page related to the given link (URL) and receives the response data in the form of HTML document. Then, it downloads all the potential geotagged images, extracts all the links from the document and places them in the queue based on their priority. Images and their metadata information is saved to the storage. This process continues with the highest priority link from the queue and terminates once the number of visited links reaches a certain limit, or when the queue becomes empty. The crawled data is saved to the storage and it can be used later by the administrator.

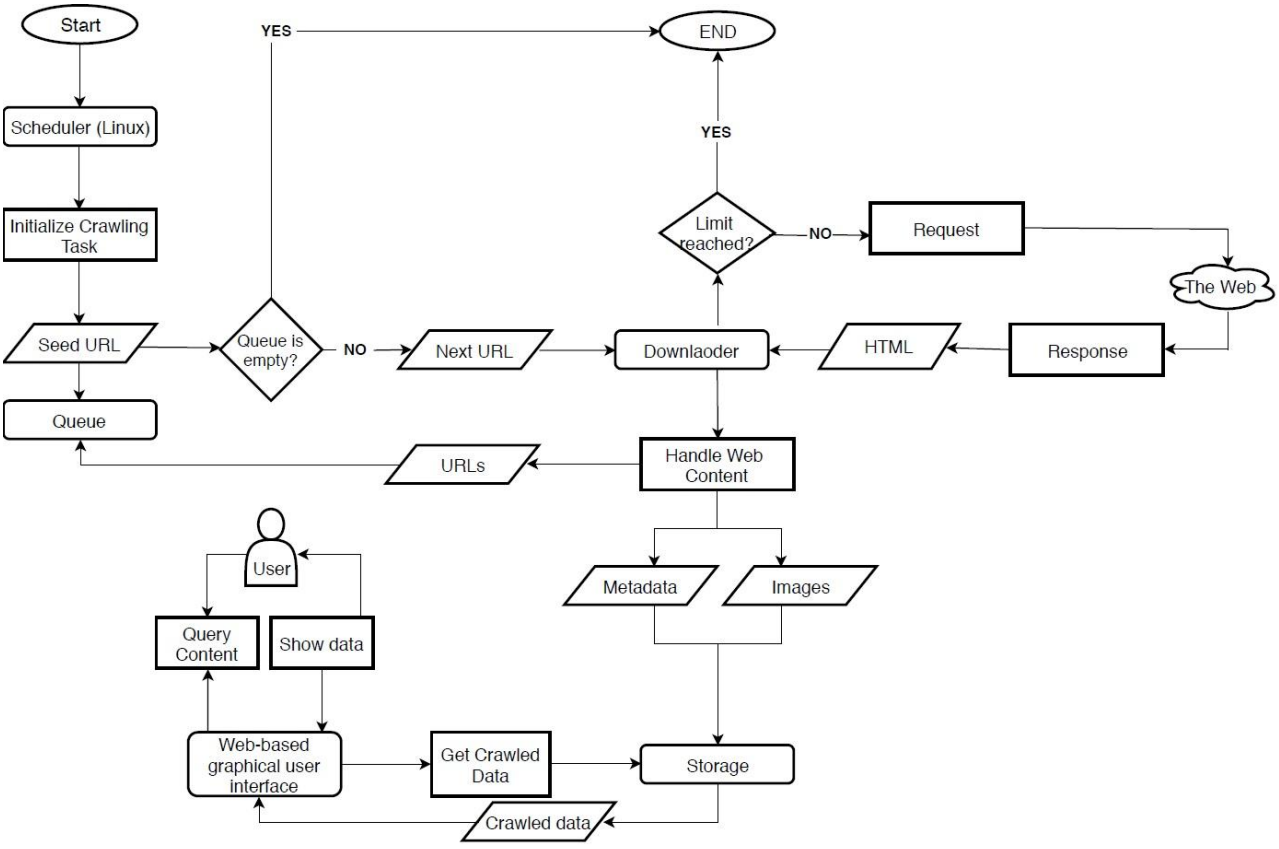

Figure 3. System architecture of Mopsi image crawler and interactions between the components

The Queue is implemented as a priority queue using *binary heap* (Cormen et al., 2009), where each node is supposed to have two child nodes and all levels of the tree should be fulfilled. The only exception is the last level of a tree. If incomplete, it is filled with empty nodes from left to right. The queue uses a web server's memory unit for storing the links and support two main operations:

- Add link
- Get the most relevant link

The Downloader is a component responsible for handling the actual crawling process. It receives a link as input and scans through the web page to download digital photos and extract their metadata. Before processing a web page, it converts the HTML content into a tree-like structure called *document object model* (DOM). The conversion process is handled using the built-in library of the *Symfony* software development framework used for the development of MIC.

The Scheduler is a part of the server's operating system. As we have implemented the MIC system on a Linux operating system, we use Linux's *Cron scheduler*. It is a background program known as a *daemon*, which schedules and executes the programs in the system. The Cron Scheduler runs at the time of the operating system boot. In every minute, it examines the *cron table (crontab)* file, which holds a list of the scheduled *cron jobs*. The instructions for scheduling execution of programs must follow a strict syntax consist of five fields: minutes, hour, day of the month, month and day of the week followed by the program command to be run.

The Storage component organizes the downloaded image files into different directories that are named under the domain names of the web sites containing the photos. It comprises of two sub-components called file storage and metadata database. The file storage is a partition of the web server's hard disk used for storing downloaded images and data management functionalities are provided by the Linux operating system. The metadata database aims to keep a cache of records of information about the downloaded web resources.

Determining the relevance of links

Our system aims to download a large number of geotagged images from as many web sites as possible. Thus, we do not require that all the visited web pages should be highly related to the overall crawling result set or to the initial seed page sent as input. However, we assume that the web pages in the crawling result set are related because of two reasons:

1. If two web pages belong to the same web site, their content is more likely related as they are connected through the same host domain.
2. Even when web pages do not belong to the same web site, they should still be related in order to satisfy *search engine optimization* (SEO) metrics for an external link, which determines the ranking of modern web sites in a search engine.

MIC prefers links to the web pages of the same web site over links to the external web sites as they are more likely related and should be given higher priority. Thus, we determine the relationship between a web page, and another connected to it without downloading its content to increase the crawling speed.

We have designed our own heuristic method for the BEFS algorithm to determine the link relevance to the web page containing it. According to the W3C standards, links on modern web sites need to be created with descriptive text title. We can take advantage of such title for calculating the relevance score. A link is considered relevant to the web page if it satisfies one of the following two criteria:

- External relevance; and
- Internal relevance.

External relevance means that both the web page containing the link and the web page associated with link should be of the same web site through similar hostname.

Internal relevance influences the degree of relevance between the web page and link. We look for the text description of link surrounded by HTML anchor tag from the DOM tree and extract some significant keywords from this description called *keywords* using the method based on candidate keyword extraction framework described in (Gali & Fränti, 2016). For each extracted keyword, we calculate the number of times it occurs in web page content; this is called term frequency (TF). The higher the number of high-frequency keywords, the more relevant a link is to the web page. As keywords in link's description occur at least once in the web page content, only keywords that occur more than once (TF > 1) contribute to the degree of relevance of the link.

1. Extract keywords from the Link's text description

Since extracted text description of a link may contain stop words such as *a*, *the*, *an*, or special characters, which are high frequency but less descriptive and should not be included while calculating link relevance score. We apply our own version of generic keyword extraction method described in (Gali & Fränti, 2016) to extract only meaningful keywords. Our method has the following three steps:

1. Normalize
2. Purify
3. Tokenize

In *normalize* step we simply trim off all the white spaces before and after the extracted text description of a link and convert all the alphabet characters into lower-case characters.

In purify step, we replace all the stop words and special characters with white spaces, as they do not carry any explicit meaning but can happen frequently. Without removing them, our method can prioritize irrelevant links whose text description contains more stop words. To be able to

detect as many stop words as possible we import a pre-defined list (Google sites, n.d.) of 7439 stop words from 28 different languages. After the purify step, the remaining words in a link's text description are considered as keywords and are ready to be separated into individual words, called tokens. We use regular expressions to detect all the white spaces and use them as delimiters to separate our keywords into individual words. Figure 4 demonstrates the overall process of the keyword extraction.

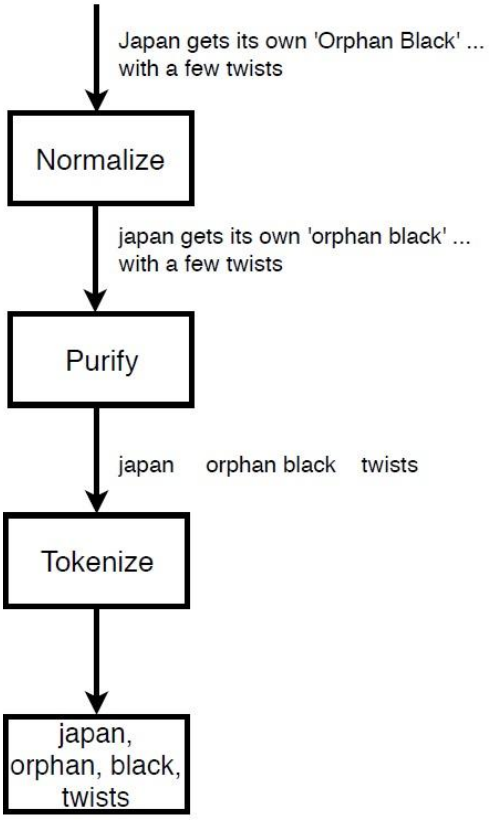

Figure 4. Demonstration of the keyword extraction method

2. Calculate keyword relevance score

The general equation for calculating the relevance score of a link is as follow:

$$R = \frac{h + \sum_{TF(k)>1} k}{1 + \sum k}$$

Where *h* takes value 0 or 1 depending on whether the link has the same hostname as the web page. Here the first summation is the total number of keywords with TF>1, and the second summation is the total number of all keywords. As an example, we consider a web page entitled *Explore Moon to Mars* from NASA web site (NASA, 2018), with links as shown in Table 1.

Table 1. Example links for relevance calculation

Link	Weblink	Link's Text description	Keywords
L_1	https://www.nasa.gov/moon	Earth's moon	earth's, moon
L_2	https://www.nasa.gov/mission_pages/mars/main/index.html	Mar's today	mars, today
L_3	https://www.nasa.gov/topics/solarsystem/index.html	Solar system and beyond	solar system
L_4	https://trek.nasa.gov/moon/index.html	Explore the Lunar Surface	explore, lunar, surface
L_5	https://www.nasa.gov/topics/technology/index.html	Space Tech	space, tech
L_6	https://www.nasa.gov/topics/moon-to-mars/preparing-to-go	Preparing People to go	preparing, people, go

Among the links in Table 1, all links point to the same web site in consideration except L_4. Thus, a hostname relevance score is 1 for all links except L_4. While calculating the TF value of each keyword, we count the number of keywords having TF value greater than 1 to produce keyword relevance score as shown in Table 2.

Table 2. Keyword relevance score for the sample links

Links	Keywords extracted from link's text description	Term Frequency (TF)	Number of Keywords having TF > 1	Relevance
L_1	earth moon	3 11	2	1.00
L_2	mars today	12 1	1	0.60
L_3	solar system	2	1	0.60
L_4	explore lunar surface	1 3 1	1	0.25
L_5	space tech	7 1	1	0.60
L_6	preparing people go	1 1 1	0	0.33

Finally, we divide the sum of the relevance score of the hostname and keyword relevance score by the total number of keywords plus one, to get the final relevance score R_1. For example, the score for L_1 is (1+2)/(1+2) = 1. In this example, the most relevant link is L_1 with the maximum relevance score of 1. The least relevant link is L_4 with a score of 0.25 since it belongs to a different hostname. The link L_6 is somewhat relevant because of its matching hostname, though it does not have any keywords with TF > 1.

3. Rules for downloading an image

Images on a web page can be classified into five groups based on their functionality (Gali et al., 2015) as representative, logo, banners, advertisement and formatting and icons. Since these five groups are overlapping, our system classifies the images into two groups only known as representative and non-representative.

- *Representative*: Images whose size meets the standard aspect ratio, measured as the ratio between the width and height of the photograph.
- *Non-representative*: Small images such as logo, banner and advertisements whose size does not meet the standard photographic aspect ratio or those not directly related to the content of a web page.

Table 3 specifies the rules we used for classifying images, and Table 4 lists the standard aspect ratios our system supports. The idea of using aspect ratio is already introduced in (Gali et al., 2015), for the detection of logo and banner images. We aim to use aspect ratio as a base for detecting representative images that are potentially photographs.

Our method is based on the following assumption: digital cameras or smart phone's built-in cameras output image size subjects to the international standards (ISO, 2015), whereas, images created manually by humans such as icons or banners follows different standards or no standards at all, thus making them unlikely photographic images.

MIC considers images of the representative type only because of the two reasons. Firstly, if the image is an actual photograph, it is more likely to contain location information in its metadata. Secondly, even if the image does not contain location information in its metadata but relates to the content of a web site, we can still determine its relative location by analyzing the text content of the web page contains it.

Table 3. Rules for categorizing web images

Category	Features	Keywords of an Image
Representative	Width > 400px Height > 400px	
Non-representative	Width < 400px Height < 400px	free, ads, now, buy, join, click, affiliate, adv, hits, counter, sprite Logo, banner, header, footer, button

Table 4. Supported standard aspect ratios of a digital image, grouped by image orientation

Orientation	Aspect Ratio	Decimal	Image resolution in pixels (width × height)
Rectangle	1:1	1.00	480 × 480, 512 × 512, 1024 × 1024
Landscape	4:3	1.33	640 × 480, 800 × 600, 832 × 624
	5:4	1.25	600 × 480, 1280 × 1024, 1600 × 1280
	3:2	1.50	960 × 640, 1152 × 768, 1440 × 960
	5:3	1.67	800 × 480, 1280 × 768
	16:9	1.78	960 × 540, 1024 × 576, 1280 × 720
	3:1	3.00	1200 × 400, 1500 × 500, 1800 × 600
Portrait	1:3	0.33	700 × 2100, 800 × 2400, 900 × 2700
	3:4	0.75	720 × 960, 768 × 1024, 864 × 1152
	3:5	0.60	480 × 800, 768 × 1280
	4:5	0.80	1280 × 1600, 1440 × 1800, 2048 × 2560
	9:16	0.56	900 × 1600, 1080 × 1920, 1440 × 2560
	2:3	0.67	1280 × 1920, 1440 × 2160, 1824 × 2736

Geo-information retrieval

Extracting location information embedded in the metadata of an image is the easiest way. However, we found that most of the images do not have location information in their metadata. We have recognized the following potential reasons why EXIF metadata is rarely found in web images:

1. Size limitations,
2. Vulnerability of the format,
3. Privacy concerns,
4. Web site performance optimization.

First, although the size of EXIF in JPEG is limited only to 64 kB, many devices exploit this space and store an extensive amount of less relevant information that is not really needed. This is referred to data over-collection problem (Dai et al., 2017). The size itself seems small but it directly contradicts the goal of using reduced size photos for efficiency when publishing on the Web.

Second, EXIF specifications have multiple anomalies, which can cause serious problems in the extraction of the metadata, including interoperability problems among different devices (Orozco et al., 2015). These can confuse both the forensic analysis and practitioners. For example, compressed images can store extended data into multiple segments so that data can be spread anywhere within a file. It is, therefore, possible that the image editors either damage or remove

the EXIF metadata by accident. Since image processing is by default practice before publishing the images on web, the corruption or removal of the EXIF is likely to happen.

Third, privacy is one of the main concerns in location-based services (Huang & Gartner, 2018). People are aware that their location can be shared automatically, and therefore, often explicitly disallow to use the location. Surprising results (Henne et al., 2014) showed that 33 percent of the people did not simply want to delete their metadata but would prefer encryption to restrict the access of others to the metadata. Since such options are not widely available, disallowing to share the metadata is the more likely option people choose.

Fourth, the authors of many modern web sites may also use image editing application to remove the metadata simply to reduce the image size for performance optimization. When there is a large amount of metadata, the file size becomes bigger which in turn increases the browser loading time. An independent experiment online recently reported that removing the metadata lead to about 8.5 percent smaller image size (Short Pixel Blog, 2017). We performed a similar but a smaller scale experiment with images from https://www.locationscout.net/ and from *Flickr*. We used software called ExifPurge to remove the EXIF metadata. Our results showed that an image size was reduced by 32.1 percent, on average.

To have an idea of how often the images on the Web have geotag in their EXIF metadata, we performed another small-scale experiment as follows. We input 8 seeds from different web sites and let the crawler visit 100 web pages for each web site. The crawler downloaded 6845 images in total. The results are summarized in Table 5 and show that only 14 images had location information embedded in their metadata. This corresponds to 0.2 percent frequency, which implies that the geodata within the EXIF is actually very rare finding on the Web.

Table 5. Number of images found (left), and the number of images having geotag in EXIF (right)

Web site	Images	Geotagged
https://www.locationscout.net/	922	5
http://www.foxnews.com/travel.html	150	1
http://www.lonelyplanet.com/	303	5
http://businessinsider.com/travel/	1714	2
http://www.vogue.com/living/travel/	4	-
http://www.dailymail.co.uk/travel/	3449	1
http://www.bbc.com/travel/	217	-
http://www.visitfinland.com/	86	-
Total	**6845**	**14**

Because of the insufficient number of geotagged images found, we needed a better method to find the location information of images based on the information associated with them. This information can be obtained from the text description of images found on web pages they belong to, using *geographic information retrieval techniques* (*GIR*), which acquires geographic information from a resource collection, particularly a collection of text (Manning et al., 2009).

The process in which geolocation information is determined and extracted from a text resource is called *geoparsing* (Richter et al., 2017). The *geoparsing* process takes an unstructured text description of places as input and produces geographic coordinates (latitude and longitude) as output (Gelernter & Zhang, 2013). For instance, the string of text: "*Cherry Blossom Heerstrasse, Bonn. Photo by Anirban Chakraborty*" produces latitude value of 50.723920 and longitude value of 7.103690, which correspond to Heerstrasse Avenue in Bonn, Germany.

The idea of geoparsing text for determining address has been discussed by (Tabarcea et al., 2017) within the framework for a *location-aware search engine*. One of the key components in this framework is the address detector, used to search and verify the validity of the postal addresses on the web page. The idea of address detector is used to identify individual address elements such as street, city and zip code, and then aggregate them to build an address candidate. Then, gazetteer data from OpenStreetMap data is used to validate each address candidate.

We use a third-party geoparsing service (Geocode.xyz API, 2015) which serves as the *geoparser*. It performs the following actions whenever it founds an image with missing geolocation information in its metadata.

1. Extract image description
2. Request geolocation information from a geoparser service
3. Save the extracted geolocation information to the database.

We start by describing an image using the text found in the DOM structure. According to the standards defined by the World Wide Web Consortium (W3C), an image should have its "*alt"* attribute defined with meaningful text content. However, many web pages do not define the value of this attribute. Therefore, the information found from the "*alt*" attribute is rarely useful.

Consequently, the image description found in the "*alt*" attribute is rarely useful for the geoparser. We have observed that the useful text description of an image is likely to come from these three sources: the alt attribute, the file name, and one of the nearest DOM elements to the image element called *caption elements*, matching either one of the following HTML tags:

1. The anchor tag <a/>
2. The heading tag <h1/> to <h5/>
3. The paragraph tag <p/>

Our method is simple: we use white space as the delimiter and concatenate each of the texts

found in the DOM element attribute, the caption element (with inner HTML tags stripped out) and the image file name (without a file extension) to produce a new text string. Figure 5 shows the method of producing the image description.

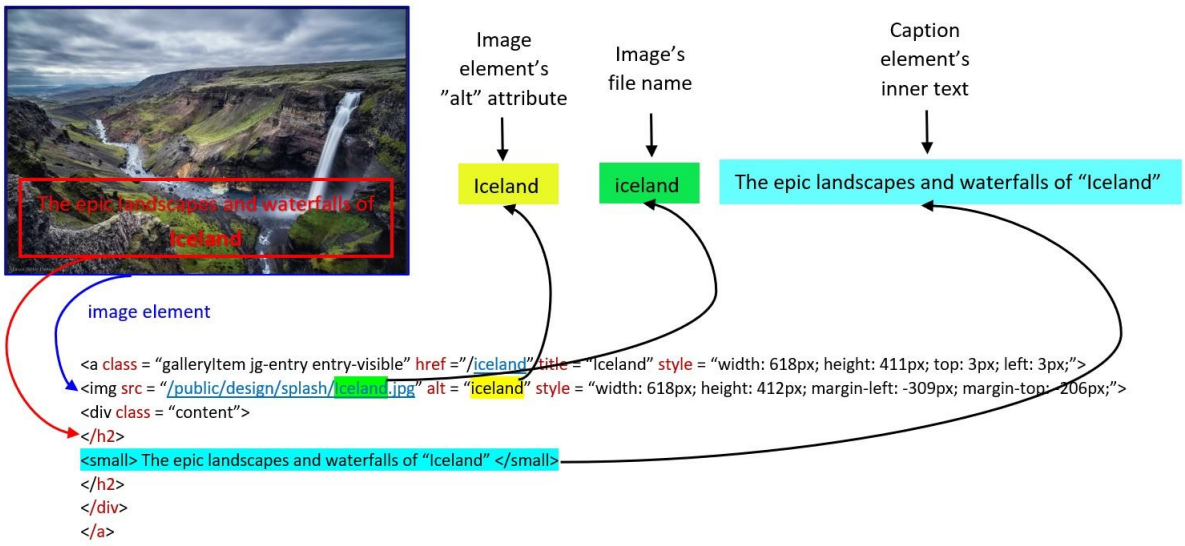

Figure 5. Illustration of the method for producing image description

The advantage of our method is that it guarantees the richness of the information as we collect text from different elements of the DOM structure. A drawback is that the method might not work so well with webs pages that have a smaller amount of text content and with dynamic web sites where JavaScript is used to generate the text content.

In the next step, we generate an HTTP request to the geoparser and receive JSON responses that provide determined address and the corresponding location coordinates. A typical request URL string is composed of two parts: the address of the service, and the input HTTP query parameters which are a free-form text, and a flag for receiving JSON data as output. Figure 6 provides a sample request to the geoparser.

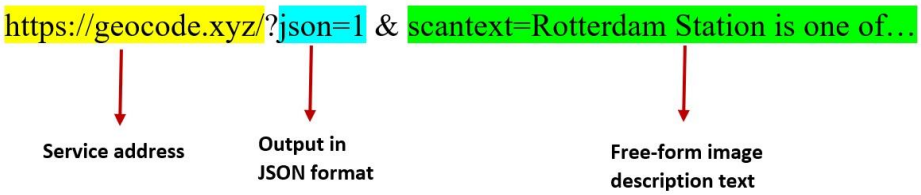

Figure 6. Geoparser request URL parts explained

The parameter *json* shows that the output data should be produced in JavaScript object notation (JSON) format and parameter *scantext* is the description of an image. Once a successful HTTP request is established with the service, it returns the output data in which we can find a list of matching locations. To simplify the geoparsing process, we configure our system to select only first matching item as it provides the most relevant geolocation information. Sample output from

the service is shown in Figure 7.

Of all the output information, our system uses only the latitude, longitude and the location values, and then updates the metadata information of the downloaded image. In case the service fails to determine the geolocation information of an image, our system facilitates users to update geolocation information through GUI frontend.


```
{
    "longt": "-105.23046",
    "matches": "1",
    "match": [
        {
            "longt": "-105.23046",
            "location": "GOLDEN,US",
            "matchtype": "locality",
            "confidence": "0.3",
            "MentionIndices": "",
            "latt": "39.76407"
        }
     ],
"latt": "39.76407"

}
```


Figure 7. Result output of the geoparser service

Experimental results

We setup the MIC system on a personal computer running Ubuntu Linux distribution with the technical specifications as shown in Table 6. For experimental results, we studied and analyzed three different aspects of the crawling results:

1. Quality of the results
2. Performance of the crawler
3. Impact of the seed selection

The quality of the crawling results is measured by the number of geotagged images over a total number of images our system retrieves. According to the official U.S. government information about the GPS, its accuracy is very high and reliable (GPS Accuracy, n.d.). We, therefore, do not consider the accuracy of the GPS coordinates, but we merely assume that the GPS information is always accurate.

Table 6. Technical specifications of the computer used for experiment purpose

Resource:	**Specification:**
Processor:	7th Generation Intel Core i7 7200U
Memory:	16Gb DDR4-2400
Storage:	256Gb SSD
Download speed	50Mbps

We compare the crawling results obtained from three different crawling algorithms: DFS, BFS, and BEFS. In the experiments, we set the limit of 1000 links in total for the crawler to visit by executing 10 consecutive crawling tasks, each task stops after visiting 100 links. All experiments are performed with Locationscouts (https://www.locationscout.net/) as a seed. We did not specify any theme or section from the web site but used the main page as such.

We collected four pieces of information from each crawling task results:

1. Execution time
2. Memory usage
3. Total number of images
4. Number of geotagged images

Table 7 shows the comparison of execution time, memory usage and average time to process a single web page by three crawling algorithms.

Table 7. Crawling task statistics of the three crawling algorithms

Algorithm	Avg. execution time (min)	Avg. memory usage (MB)	Avg. time to process a single web page (s)
BFS	10	63	9.23
DFS	14	81	6.52
BEFS	26	128	9.41

We found that the execution time of the crawling tasks using the BEFS algorithm is slower and not as stable as that of DFS and BFS. This is because of its performance dependency on the size of content used for calculating the relevance score between the web page and its links. Similarly, BEFS consumes more memory because it performs more calculations, which store the values to the memory.

However, the results revealed that downloading the images and handling their metadata takes most of the time. This process consumes about 86 percent of the overall processing time, with average time ranging from 5 to 8 seconds per page, which contains 10 images, on average.

The other processes include the processing of web page's text content, database reading and writing, link extraction and queue operations which take about one second on average, for all the algorithms. Table 8 summarizes the proportion of execution times for the steps involved in the process of a single web page.

Table 8. Proportion of the execution times of the processing steps for a single web page

Process	Time taken (s)		
	BEFS	**BFS**	**DFS**
Download and handle image metadata	8.1	8.9	5.6
Relevance Calculation	< 0.1	0	0
Other	1.2	1.0	0.8

We conclude that the overall performance of BEFS is slightly worse than the BFS and DFS. It takes about 1.5 milliseconds to calculate the relevance score of a link whereas BFS and DFS do not have this step. The size of the text content and the number of links in a web page affects the BEFS performance. Although the speed of BEFS is slower than that of BFS and DFS, it helps significantly to retrieve more geotagged images according to the statistics shown in Table 9. BEFS discovered about 2 percent of the total images as geotagged. This percentage is 1.3 percent for BFS and only 0.4 percent for DFS.

Table 9. Crawling result statistics of the three crawling algorithms

#	BEFS		BFS		DFS	
	Images	**Geotagged**	**Images**	**Geotagged**	**Images**	**Geotagged**
1	645	23	811	21	972	3
2	222	-	45	-	589	-
3	1272	23	407	-	470	3
4	1170	15	440	5	261	-
5	1295	27	383	10	817	4
6	1280	57	816	5	388	3
7	1360	18	12	-	1	-
8	1390	30	420	9	6	-
9	18	-	306	1	0	-
10	1180	8	258	1	72	-
Total	9832	201 **2.0%**	3898	52 **1.3%**	3576	13 **0.4%**

The better crawling result of the BEFS algorithm comes from the fact that it considers the relationship between an individual web page and its associated links, which is not the case with DFS and BFS.

With reference to the distribution of representative and non-representative images in the overall crawling result, we found that approximately 53 percent of the web images belong to the non-representative category. We further studied the classification of images based on their aspect ratio as defined in Table 4. Using our crawled dataset of 687 geotagged images collected from

different seeds, our results showed that most of the geotagged images had aspect ratio of 3:2 (68%), while no images found belonging to the aspect ratio of 3:1, 1:3 or 3:5, as shown in Table 10.

Table 10. The number of images found versus the aspect ratio

Orientation	Aspect ratio	Results
Landscape	3:2	464
	16:9	96
	4:3	33
	5:3	5
	5:4	2
	3:1	-
Rectangle	1:1	14
Portrait	2:3	66
	3:4	5
	4:5	1
	9:16	1
	1:3	-
	3:5	-

Next, we study the impact of a seed on the crawling result using 27 manually selected web sites of different topic domains. For each of the seeds, we run only one crawling task that was limited to visit 100 links. The results in Table 11 shows that the web sites about photo search and sharing service, and travel news have a higher chance to have geotagged images. In contrast, web sites about blogs and business usually contain no geotagged images. We conclude that the seed plays a vital role in the crawling performance.

We further studied the correlation of the topic/keywords to the success of finding geotagged images. For example, knowing the title of a web page, how far we can conclude if the page would have geotagged images? For this purpose, we extracted the keywords from different web sites (seeds) that provided geotagged images using keyword extraction tool ("Northcutt," n.d.).

The results in Table 11 show that the seeds contributing geotagged images have more often keywords like *travel*, *photo*, *image*, *photography*, *world*, *trip*, *spots*, and *landscapes*. Web sites containing such keywords have a higher probability of geotagged images. In other words, we can also say that the web sites about travel resources and photo search are a good source of geotagged photos.

Table 11. Impact of seed quality and correlation of the keywords with number of geotagged photos

Web Page	Description	Keywords	Images	Geotagged	%
https://www.locationscout.net/	Geo photo sharing service	*landscape, places, photo, photos, photography*	945	52	5.5
https://www.pexels.com/	photo search and sharing service	*free, viewer, October, choose, photos, English, stock*	781	14	5.2
https://imagelocations.com/	photo search service	*city, beach, photographs, lake, valley, desert, woodland, ocean*	336	21	6.3
http://www.foxnews.com/travel.html	travel news	*travel, spots, popular, places, news, natural*	129	5	3.8
https://news.zing.vn/du-lich.html	travel news	*news, world, sang, zing, du*	433	1	0.2
http://businessinsider.com/travel/	travel news	*travel, business, valley, discover, stock*	1442	02	0.1
https://www.theguardian.com/uk/travel/	travel news	*travel, guides, review, guardian, places, world, planet, photograph, holidays, landscape*	59	1	1.7
https://www.msn.com/en-us/travel/	travel news	*travel, photos, version, news, guides, please,*	336	4	1.2
http://www.vogue.com/living/travel/	travel news	*vacation, vogue, fall, spring, guide, beautiful*	59	1	1.7
http://www.dailymail.co.uk/travel/	travel news	*news, destinations, holidays, Europe, travel, travelers, country, images, world, photographs*	3296	2	0.1
http://www.bbc.com/travel/	travel news	*travel, news, world, nature, beaches, city, image, nature, world*	185	5	2.7
http://www.visitfinland.com/	Finland travel guide	*travel, guide, Finland, spot, nature, places, landscape, world, trip*	88	4	4.5
http://www.utranuittotupa.fi/	restaurant service	*private, restaurant, party rooms, Joensuu, business, new, theatre*	42	0	0
https://pkamknkirjasto.wordpress.com/tag/wartsila-talo/	blog	*knowledge, library, posts, tagged, start, already, decoration, proceeded, schedule*	29	0	0
http://pippurimylly.fi/	restaurant service	*best, peppermill, recipes, think, familiar, restaurant*	110	0	0
http://vaarakirjastot.fi/paakirjasto	location information page	*reserve, edit, libraries, sun, café, young people, remember, premises*	9	0	0
http://www.kausalanautotarvike.fi/	Car accessories	*rights, designed, copyright, car accessory, service, spare parts*	368	1	0.3
http://www.huanqiu.com/	World news	*largest, source, china, chili, jeep*	749	3	0.4
https://visitkouvola.fi/ru	Kouvola travel guide	*visitkouvola, innovation, concrete*	16	0	0
https://visitsweden.com/	Sweden travel guide	*holidays, Sweden, visit, official, tourism, travel, holiday, people, Swedish, activities*	45	2	4.4
https://www.thecrazytourist.com/	travel resource	*tourist, crazy, Alaska, England, California, Arizona, globe, best*	89	1	1.1
http://www.lahdenmuseot.fi/museot /fi/hiihtomuseo/	Lahti city museum	*museum, renovations, bay, art, poster*	397	17	4.3
http://international.visitjordan.com/	Tourism	*Jordan, tourism, visit, land, travel, beauty*	162	6	3.7
http://visitbudapest.travel/	Tourism	*Budapest, world travel, guide, attraction*	54	3	5.5
https://info.goisrael.com/en/	Tourism	*Israel, tourist, information, travel, tours, attractions*	88	4	4.5
https://www.choosechicago.com/	Tourism	*things, hotels, vacation, Chicago, international,*	120	3	2.5
https://www.tourism-lorraine.com/	Tourism	*Lorraine, tourisme, discover, destinations, natural, holiday*	61	1	1.6

Conclusions

We have studied the efficiency of web crawling for retrieving geo-tagged images on the Web. Our system targets geotagged images, which have geographical coordinates information embedded in their EXIF metadata. The geotagged images retrieved by the system can be used as a material for content creation in O-Mopsi, which is a mobile location-based orienteering game. Our experiments revealed that only <1 percent of the all images crawled were geotagged. These results are therefore quite disappointing. Therefore, in order to improve the performance, we used geoparsing tools, which increased the number of geotagged images by 5 percent, which is still not very encouraging.

We found potential reasons for the discouraging results as: Images in modern web sites might not include EXIF metadata because of size limitations, privacy concerns, and web site performance optimization. While *Flickr* and *SmugMug* have the geo-location stored in the images, most other networking web sites like *Facebook* and *Instagram* delete the metadata for privacy and security reasons.

We further conclude that the seed plays a very important role in the overall crawling results. A good seed provides significantly better crawling results while a bad seed can provide zero results. Web sites about travel news and photo sharing services serve as a good seed and are more likely to provide geotagged photos as compared to the web sites about business and blogs.

Acknowledgments

This study was partly funded by Finnish national agency for education (grant TM-17-1057-4).

References

Alvarez, P. (2004). Using extended file information (EXIF) file headers in digital evidence analysis. *International Journal of Digital Evidence*, *2*(3), 1-5.

Baggili, I., Marrington, A., & Jafar, Y. (2014, January). Performance of a logical, five-phase, multithreaded, bootable triage tool. In *IFIP International Conference on Digital Forensics* (pp. 279-295). Springer, Berlin, Heidelberg.

Castillo, C. (2004). *Effective Web Crawling (Ph. D. thesis). University of Chile*. Retrieved 2010-08-03.

Cho, J., Garcia-Molina, H., & Page, L. (1998). Efficient crawling through URL ordering. *Computer Networks and ISDN Systems*, *30*(1-7), 161-172.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms*. MIT Press.

Dai, W., Chen, L., Qiu, M., Wu, A., & Chen, B. (2017, November). A privacy-protection data separation approach for fine-grained data access management. In *2017 IEEE International Conference on Smart Cloud (SmartCloud)* (pp. 84-89). IEEE.

Elyasir, A. M. H., & Anbananthen, K. S. M. (2012). Focused web crawler. *IPCSIT*, *45*, 149-153.

Fränti, P., Mariescu-Istodor, R., & Sengupta, L. (2017). O-Mopsi: Mobile orienteering game for sightseeing, exercising, and education. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, *13*(4), 56.

Gali, N., & Fränti, P. (2016, April). Content-based title extraction from web page. In *WEBIST (2)* (pp. 204-210).

Gali, N., Tabarcea, A., & Fränti, P. (2015). Extracting representative image from web page. In *WEBIST* (pp. 411-419).

Gelernter, J., & Zhang, W. (2013, November). Cross-lingual geo-parsing for non-structured data. In *Proceedings of the 7th Workshop on Geographic Information Retrieval* (pp. 64-71). ACM.

Henne, B., Koch, M., & Smith, M. (2014, March). On the awareness, control and privacy of shared photo metadata. In *International Conference on Financial Cryptography and Data Security* (pp. 77-88). Springer, Berlin, Heidelberg.

Huang, H. C., Fang, W. C., & Chen, S. C. (2008, December). Copyright protection with EXIF metadata and error control codes. In *2008 International Conference on Security Technology* (pp. 133-136). IEEE.

Huang, H., & Gartner, G. (2018). Current trends and challenges in location-based services. *ISPRS International Journal of Geo-Information*, 7(6), 199.

Khan, W. Z., Xiang, Y., Aalsalem, M. Y., & Arshad, Q. (2013). Mobile phone sensing systems: A survey. *IEEE Communications Surveys & Tutorials*, *15*(1), 402-427.

Kobayashi, M., & Koichi, T. (2000). *Information retrieval on the Web*. ACM Computing Surveys (CSUR).

Krishnamurthy, V., Saranya, S., Srikanth, S., & Modi, S. (2017, August). M-WPS: Mobile based women protection system. In *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)* (pp. 1701-1706). IEEE.

Kuo, Y. S., Pannuto, P., Hsiao, K. J., & Dutta, P. (2014, September). Luxapose: Indoor positioning with mobile phones and visible light. In *Proceedings of the 20th annual international conference on Mobile computing and networking*(pp. 447-458). ACM.

Lloret Romero, N., Gimenez Chornet, V. V., Serrano Cobos, J., Selles Carot, A. A., Canet Centellas, F., & Cabrera Mendez, M. (2008). Recovery of descriptive information in images from digital libraries by means of EXIF metadata. *Library Hi Tech*, *26*(2), 302-315.

Manning, C. D., Raghavan, P., & Schutze, H. (2009). *Introduction to Information Retrieval*. Cambridge: Cambridge University Press.

Markov, Z., & Larose, D. T. (2007). *Data mining the Web: uncovering patterns in Web content, structure, and usage*. John Wiley & Sons.

McBryan, O. A. (1994, May). GENVL and WWWW: Tools for taming the web. In *Proceedings of the First International World Wide Web Conference* (Vol. 341).

Mirtaheri, S. M., Dinçtürk, M. E., Hooshmand, S., Bochmann, G. V., Jourdan, G. V., & Onut, I. V. (2013, November). A brief history of web crawlers. In *Proceedings of the 2013 Conference of the Center for Advanced Studies on Collaborative Research* (pp. 40-54). IBM Corp..

Najork, M. (2009). Web crawler architecture. *Encyclopedia of Database Systems*, 3462-3465.

Naranjo, J. E., Jiménez, F., Anaya, J. J., Talavera, E., & Gómez, O. (2017). Application of vehicle to another entity (V2X) communications for motorcycle crash avoidance. *Journal of Intelligent Transportation Systems*, *21*(4), 285-295.

Olston, C., & Najork, M. (2010). Web crawling. *Foundations and Trends® in Information Retrieval*, *4*(3), 175-246.

Orozco, A. L. S., González, D. M. A., Villalba, L. J. G., & Hernández-Castro, J. (2015). Analysis of errors in exif metadata on mobile devices. *Multimedia Tools and Applications*, *74*(13), 4735-4763.

Panou, C., Ragia, L., Dimelli, D., & Mania, K. (2018). An architecture for mobile outdoors augmented reality for cultural heritage. *ISPRS International Journal of Geo-Information*, *7*(12), 463.

Quercia, D., Lathia, N., Calabrese, F., Di Lorenzo, G., & Crowcroft, J. (2010, December). Recommending social events from mobile phone location data. In *2010 IEEE international conference on data mining* (pp. 971-976). IEEE.

Rahman, M. K., & Nayeem, M. A. (2017, May). Finding suitable places for live campaigns using location-based services. In *Proceedings of the Fourth International ACM Workshop on Managing and Mining Enriched Geo-Spatial Data* (p. 7). ACM.

Rainio, A. (2001). Location-based services and personal Navigation in mobile information society. New technology for a new century. In *International Conference FIG working Week 2001, 6 11 May, Seoul, Korea*.

Richter, L., Geiß, J., Spitz, A., & Gertz, M. (2017, September). HeidelPlace: An extensible framework for geoparsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 85-90).

Ruda, A., Kolejka, J., & Silwal, T. (2018). GIS-assisted prediction and risk zonation of wildlife attacks in the Chitwan National Park in Nepal. *ISPRS International Journal of Geo-Information*, *7*(9).

Sengupta, L., Mariescu-Istodor, R., & Fränti, P. (2018). Planning your route: where to start? *Computational Brain & Behavior*, *1*(3-4), 252-265.

Singh, A. V., & Vikas, A. M. (2014). A review of web crawler algorithms. *International Journal of Computer Science & Information Technologies*, *5*(5), 6689-6691.

Tabarcea, A., Gali, N., & Fränti, P. (2017). Framework for location-aware search engine. *Journal of Location Based Services*, *11*(1), 50-74.

Tabarcea, A., Wan, Z., Waga, K., & Fränti, P. (2013). O-Mopsi: Mobile orienteering game using geotagged photos. In *WEBIST* (pp. 300-303).

Wang, J., Ni, D., & Li, K. (2014). RFID-based vehicle positioning and its applications in connected vehicles. *Sensors*, *14*(3), 4225-4238.

Statista. (2016). *Number of smartphone users worldwide 2014-2020*. Retrieved April 22, 2018, from https://www.statista.com/statistics/330695/number-of-smartphone- users-worldwide/

Google Sites. *Download stop words - Kevin Bougé*. Retrieved March 20, 2018, from https://sites.google.com/site/kevinbouge/stopwords-lists

NASA. (2018). *Moon to Mars*. Retrieved August 29, 2018 from https://www.nasa.gov/topics/moon-to-mars

ISO. (2015). *ISO 18383:2015*. Retrieved July 22, 2018 from https://www.iso.org/standard/62322.html

ShortPixel Blog. (2017). *How much smaller are images with EXIF data removed?* Retrieved August 20, 2018 from https://blog.shortpixel.com/how-much-smaller-can-be-images-without-exif-icc/

Geocode.xyz API. (2015). *Geocoding/reverse geocoding/geoparsing & sentiment analysis API terms*. Retrieved August 29, 2018 from https://geocode.xyz/api

GPS Accuracy. *GPS.gov: GPS accuracy*. Retrieved September 06, 2018 from https://www.gps.gov/systems/gps/performance/accuracy/#how-accurate

Northcutt. *Keyword extractor tool*. Retrieved September 30, 2018 from http://northcutt.com/tools/free-seo-tools/keyword-extrac.

Bibliographic information of this paper for citing:

Fazal, Nancy, Nguyen, Khue, & Fränti, Pasi (2019). "Efficiency of web crawling for geotagged image retrieval." *Webology*, 16(1), Article 177. Available at: http://www.webology.org/2019/v16n1/a177.pdf

---