Towards IP-based Geolocation via Fine-grained and Stable Webcam Landmarks

Zhihao Wang
Institute of Information Engineering
Chinese Academy of Sciences
School of Cyber Security, University
of Chinese Academy of Sciences
Beijing, China
wangzhihao@iie.ac.cn

Qiang Li*
School of Computer and Information
Technology
Beijing Jiaotong University
Beijing, China
liqiang@bjtu.edu.cn

Jinke Song
School of Computer and Information
Technology
Beijing Jiaotong University
Beijing, China
jikesog@gmail.com

Haining Wang
Department of Electrical and
Computer Engineering
Virginia Polytechnic Institute and
State University
Arlington, VA, USA
hnw@vt.edu

Limin Sun
Institute of Information Engineering
Chinese Academy of Sciences
School of Cyber Security, University
of Chinese Academy of Sciences
Beijing, China
sunlimin@iie.ac.cn

ABSTRACT

IP-based geolocation is essential for various location-aware Internet applications, such as online advertisement, content delivery, and online fraud prevention. Achieving accurate geolocation enormously relies on the number of high-quality (i.e., the fine-grained and stable over time) landmarks. However, the previous efforts of garnering landmarks have been impeded by the limited visible landmarks on the Internet and manual time cost. In this paper, we leverage the availability of numerous online webcams that are used to monitor physical surroundings as a rich source of promising high-quality landmarks for serving IP-based geolocation. In particular, we present a new framework called *GeoCAM*, which is designed to automatically generate qualified landmarks from online webcams, providing IP-based geolocation services with high accuracy and wide coverage. GeoCAM periodically monitors websites that are hosting live webcams and uses the natural language processing technique to extract the IP addresses and latitude/longitude of webcams for generating landmarks at large-scale. We develop a prototype of GeoCAM and conduct real-world experiments for validating its efficacy. Our results show that GeoCam can detect 282,902 live webcams hosted in webpages with 94.2% precision and 90.4% recall, and then generate 16,863 stable and fine-grained landmarks, which are two orders of magnitude more than the landmarks used in prior works. Thus, by correlating a large scale of landmarks, GeoCAM is able to provide a geolocation service with high accuracy and wide coverage.

*Corresponding author: Qiang Li, liqiang@bjtu.edu.cn

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380216

CCS CONCEPTS

• **Information systems** → *Information extraction*; *Web crawling*;
• **Networks** → *Location based services.*

KEYWORDS

Internet of Things, Data Mining, Webcam, Information Extraction, IP Geolocation, Landmarks

ACM Reference Format:
Zhihao Wang, Qiang Li, Jinke Song, Haining Wang, and Limin Sun. 2020. Towards IP-based Geolocation via Fine-grained and Stable Webcam Landmarks. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3366423.3380216

1 INTRODUCTION

IP-based geolocation is used to determine the real-world geographic location of an Internet-connected host [20], which is valuable for many Internet applications, including targeted advertising, restricted content delivery, and online fraud detection. IP-based geolocation involves in mapping an IP address (or a domain name) to a country and region (city), as well as the corresponding pair of latitude/longitude. However, commercial geolocation databases only achieve less than 95.8% on the country-level and various disagreements on the city-level [6]. Nowadays, there is no official source of geolocation databases achieving high accuracy and wide coverage for Internet users.

Providing an accurate geolocation service is heavily dependent on the number of high-quality landmarks. Given sufficient landmarks, geolocation services would provide high accurate mappings between IP addresses and geographical locations. However, efforts to gather such mapping information are impeded by the limitations of available landmarks on the Internet. Those landmarks are community-based [9, 17, 25], leading to a limited scale in terms of their number and coverage. For instance, PlanetLab [25] only includes 420 available landmarks, and most of them are located in

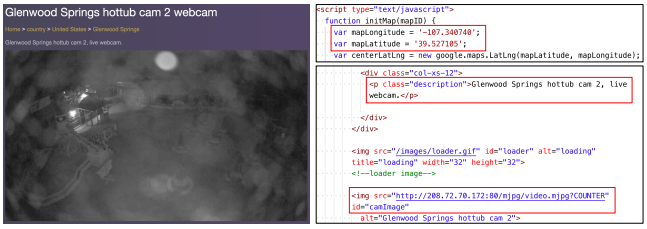
Europe and North America. They are mainly located on academic networks, with reduced availability on residential or commercial networks. Prior works proposed to increase the number of landmarks by mining web services [8, 31]. They assumed that web services are hosted locally. However, due to the ever-increasing popularity of cloud services and content delivery networks (CDNs), this assumption is no longer valid. In addition, dynamic IP addresses of those local web services would also lower their effectiveness.

As the pervasiveness of Internet-of-Things (IoT) systems, we have witnessed the significant increase of online webcams widely deployed for monitoring physical surroundings around the real world. Those online webcams are inherently associated with IP addresses and geographical locations, and more importantly, they are relatively stable over a long period, becoming ideal candidates for being used as landmarks. Thus, by leveraging the availability of numerous online webcams as the promising landmarks, we are able to successfully address the challenging problem, i.e., the lack of high-quality landmarks, in IP-based geolocation.

In this paper, we propose a framework called *GeoCAM* that is able to generate high-quality landmarks by automatically extracting the IP addresses and latitude/longitude of online webcams at large scale. We first need to search for those websites that have gathered webcams and exposed their live streams to the public. Specifically, we utilize unique features in live streams of webcams to find those websites and select top 100 sites as the target websites for GeoCAM. After the selection of target websites, GeoCAM periodically monitors these websites via crawling and uses the machine learning algorithm to determine whether a webpage includes a live webcam. Once a live webcam is found, we utilize the natural language processing technique to extract the latitude/longitude and geographical information of the device. Thus, the webcam landmarks generated by GeoCAM include the information of IP addresses and corresponding pairs of latitude/longitude. Based on webcam landmarks, we can approximately pinpoint the geolocation of an individual Internet host with high accuracy and wide coverage.

To validate the efficacy of GeoCAM, we build a prototype and conduct real-world experiments. Our results show that GeoCAM can detect webcams from webpages with 94.2% precision and 90.4% recall. For IP-based geolocation, our approach achieves higher accuracy and wide coverage than prior works. In total, we collect 282,902 webcams and generate 16,863 webcam-based landmarks, which are two orders of magnitude more than the landmarks used in prior works. These webcams cover around 170 countries, 6,450 cities, and 2,880 ASes.

The rest of this paper is organized as follows. Section 2 introduces the background of live webcams landmarks for IP geolocation. Section 3 presents the architecture of GeoCAM. In Section 4, we perform experiments on 100 aggregator websites to find webcam landmarks and measure the sheer number, stability, and coverage of these landmarks. In Section 5, we describe GeoCAM geolocation performance, and compare webcam landmarks with commercial geolocation databases and open-source landmarks. Section 6 discusses the limitations and privacy concerns of our approach. Section 7 surveys related works of IP geolocation, and finally, Section 8 concludes the paper.



(a) Live webcam on websites (b) IP address and latitude/longitude

Figure 1: An example of live webcam on websites.

2 BACKGROUND AND MOTIVATION

In this section, we present the background of live webcams available from online websites and the landmark generation based on those online webcams for IP-based geolocation.

2.1 Live Webcam

Webcams have been increasingly connected to the Internet in the past decade. A webcam is one typical Internet of Things (IoT) device [5, 15] for monitoring physical surroundings, such as a plaza, a street, an industrial area, or a home theater. For the sake of remote access and control, webcam devices are visible and accessible through their IP addresses. In contrast to web services, webcam devices are fixed in physical places and remain relatively stable over time. Thus, online webcams have great potential to be used as promising landmarks but have not yet been explored.

Figure 1 shows one example, in which a live webcam hosted on the website of pictimo.com, monitoring the surroundings at "Glenwood Springs hottub, United States". The website utilizes the web applet to host live streams of the webcam, which is labeled with HTML tag "<img src=IP address:port/mjpg", disclosing the webcam's IP address and corresponding port. Further, the webcam's latitude/longitude pair *(-107.340740, 39.527105)* is also embedded in the HMTL file. Therefore, we can extract a webcam's IP address and geographical information to generate a landmark for IP-based geolocation services.

2.2 Landmark Generation

A high-quality landmark implies that its IP address and corresponding latitude/longitude remain stable over time. Prior works [8, 31] utilize web mining to collect landmarks from web services for providing IP geolocation. However, their landmarks have become much less available to the public, due to the wide use of CDNs and clouds for hosting web services. In addition, the mappings between IP addresses and geographical information of such landmarks based on web services are not stable over the time.

For the generation of high-quality landmarks, two fundamental properties must be held for the candidates. First, landmark candidates should be relatively stable over the time; in other words, they should be publicly available for access without changing their IP addresses and geolocations for a long time period. Webcams are usually placed in fixed positions for a relatively long period except for being uninstalled. We will periodically monitor websites to update the webcam landmarks for guaranteeing their effectiveness (see details in Section 3.2). Second, landmark candidates should include fine-grained geographical information for Internet

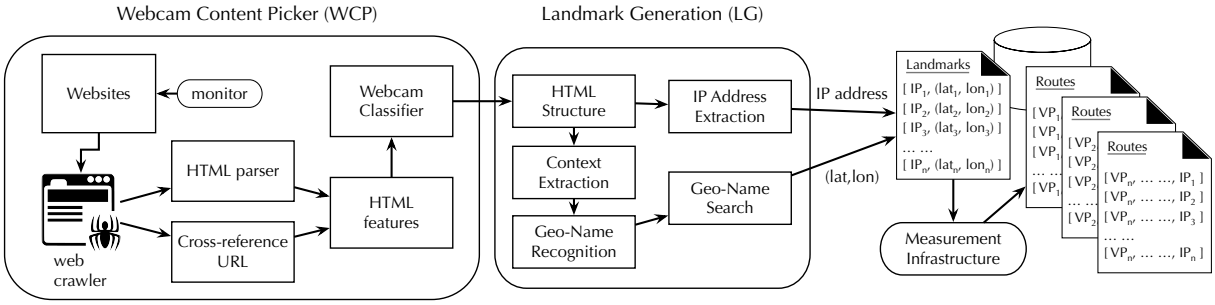


Figure 2: The architecture of *GeoCAM*.

geolocation services. Fortunately, we found many websites provide latitude/longitude and geographical information of those hosted webcams via user provision or manual inspection. Further, webcam images also disclose the surrounding physical information for fine-grained geographical information.

In this work, we propose GeoCAM to automatically extract geographical information and generate high-quality landmarks from live webcams hosted on websites. We leverage a set of existing resources and tools to address several practical problems in the process of GeoCAM, which are briefly introduced below.

Webcam Distribution Websites. Many websites collect webcam resources and distribute live streams to the public for multiple purposes, e.g., advertising beautiful scenery, monitoring traffic and weather. Considering various webpages organized by those websites, we need to filter out outlier webpages and keep useful webpages for geographical information extraction. We utilize the machine learning algorithms to build the classification model to determine whether a webpage is hosting a live webcam.

Information Extraction. For each webpage hosting a live webcam, we further extract the geographical information from the webpage, including IP address and latitude/longitude. In the Natural Language Processing (NLP) community, extracting such elements from a document is defined as Named Entity Recognition (NER) [21], which has been extensively studied. However, we cannot directly use NER to extract information because it is highly related to a specific domain. Geographical names usually are non-dictionary words, leading to low precision and recall. In this paper, we leverage a rule-based NER and local positions to extract landmarks from various webpages.

3 GEOCAM: DESIGN AND IMPLEMENTATION

In this section, we present the design and implementation of GeoCAM. Figure 2 illustrates its architecture, which consists of two major components: the webcam content picker (WCP) and the landmark generation (LG). The WCP first automatically scrapes webpages from websites, removing irrelevant content through the HTML parser. Cross-reference links are extracted to remove replicated and redundant pages among those webpages we collected. Further, the WCP utilizes machine learning algorithms to determine whether a webpage contains a live webcam. For each page considered to be webcam-relevant, the LG converts all its content (scripts, menus, and images) to texts, and preserves a snapshot of a live webcam. We use the regex to extract IP addresses, domain names, and geographical coordinates from webpages. The LG uses the rule-based NER to extract geographical names from various webpages and convert them to the latitude/longitude pairs. The LG outputs the landmark set, as a key-value format (IP, *(lat, lon)*). Note that we manually validate the effectiveness of a landmark by conducting a comparison with the stored snapshot of the corresponding live webcam in the Google Map. By creating a large number of landmarks, we can provide IP-based geolocation services. Below we elaborate on the details of GeoCAM.

3.1 Websites

GeoCAM relies on the websites that are hosting a large number of live webcams. We collected those websites using a heuristic rule: if a website distributes live streams under web applets (e.g., JPEG, MJPEG, VLC, and FFMPEG), we keep it as a candidate site. We automatically generated keywords from those web applets, ran the keywords through the open repository (Common Crawler Project [3]) and the Google search engine, and found over 3,000 websites. Then, we manually selected 100 websites (listed in Table 9) that host a large number of webcams and used them as the GeoCAM's input. One might concern that the limited number of websites could hinder the scalability of our approach. However, based on those 100 websites, we have already found 16,863 visible and accessible webcam landmarks, which are two orders of magnitude more than those used in the existing services. If a new site is available in the future, our approach can generate new landmarks with little modifications.

3.2 Relevant Webpage Detection

Aforementioned, to automatically generate landmarks from websites, we first need to scrape webpages, filter out noise and irrelevant content, and determine whether a live webcam is running on a webpage.

Web Crawler. Our web crawler is designed to periodically monitor a website to collect its webpages. Since different websites have different templates and HTML structures, we design a web crawler to scrape webpages from those sites. Given a website, the WCP first utilizes the web crawler to obtain all its pages through the breadth-first search. Specifically, we parse the website homepage to explore all its URL links, and iteratively parse the URL links to explore the next-layer pages, until no more links are found.

Table 1: Examples of regexes.

General type	Regex
IPv4	((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)(/([0-2][0-9]|3[0-2]|[0-9]))?
Domain Name	https?://[-a-zA-Z0-9._:%\+~#=]{2,256}\.([a-z]{2,6}|\d{1,3})\b([-a-zA-Z0-9._:%\+~#=?&;\(\)\[\]]*)
Geographical Coordinate	(latitude(\n|.)*?(?P<lat>[-+]?\d{1,2}\.\d+)(\n|.)*?longitude(\n|.)*?(?P<lon>[-+]?\d1,3\.\d+))|(longitude(\n|.)*?(?P<lon>[-+]?\d1,3\.\d+)(\n|.)*?latitude(\n|.)*?(?P<lat>[-+]?\d{1,2}\.\d+))

A problem here is that websites might change webpages or live webcams over time, e.g., new pages are added or old pages are removed. However, we believe that webcams are relatively stable for a long time period. The WCP periodically monitors websites/webpages by re-accessing those URL links to identify whether they are still available. In practice, the monitoring period for websites is one month and the period for webpages is one week.

Pre-Processing. Webpages collected by the web crawler involve irrelevant information, such as advertisements, icons, and navigation bars. The WCP utilizes the HTML parser to remove those irrelevant elements from webpage files, e.g., <ad> and <icon>. Websites usually utilize Javascript and Frame in HTML to post live streams of webcams. We keep the content of webpages together with javascript and frame for extracting webcam information. In addition, there are many cross-reference links in those webpages. Some URL links belong to a same site, while some come from different websites. If a link involves a page we have collected, we remove the duplicated one. If a link comes from different sites, we extract its domain name as a candidate website. We manually inspect those candidate sites to determine whether they host a large number of webcams. After scraping and pre-processing, we store all HTML files and use URL links as their index.

Webcam Classification. Another problem here is that some webpages do not post any live streams, while some include multiple webcams. Hence, we divide webpages into three groups: *none*, *single*, and *multiple*. In the first group ("none"), many webpages belong to user-generated-content (UGC) pages or others (e.g., contact pages, introduction pages, or login pages). We cannot extract any landmark information from those webpages that have no webcams. In the second group ("single"), webpages have only one embedded live webcam and distribute its live streams to the public. We can generate a landmark from such a webpage, including its IP address and latitude/longitude. In the third group ("multiple"), webpages might have more than one webcams. Generally, there are two kinds of HTML templates for multi-webcam pages: (1) a page displays multiple webcams under a particular topic, and (2) a page shows a single webcam together with some recommended webcams from the website owner. We can generate multiple landmarks from those webpages.

To automatically partition webpages, we leverage the observation: a live webcam is encapsulated as the specific HTML element to post its snapshot or live stream on webpages. The WCP extracts webcam features to infer the classification model of webpages as follows:

(1) IP address and domain name. To post a live stream, a website needs a webcam's source path to load its video content. Some sites directly use the webcam's IP address and port number, while others use a domain name to host the webcam's content.
(2) Snapshot. Webpages provide a webcam snapshot as its content to the public. The update rates of snapshots are different according to different website configurations. Typically, Joint Photographic Experts Group (JPEG) and Motion JPEG (MJPEG) are used to update the webcam's snapshots.
(3) Live stream. Webpages directly distribute live streams of webcams to the public. VideolAN media player (VLC) and FFMPEG are usually software libraries to load live streams.

To present those features, we use the regex matching to extract those features, as listed in Table 1. We use a binary vector indicating the presence of characters in a webpage. If the regex finds a character, the corresponding feature in the vector is set to 1, otherwise to 0. We use the machine learning algorithms to derive the classification model of the webpage. The model outputs are with three class labels: *none*, *single*, and *multiple*.

Implementation. We use BeautifulSoup [1] to parse each HTML file into a list of chunks based on the HTML tags. For each chunk, we use the regex to extract IP addresses, domain names, snapshots, and live streams. We use the scikit-learn [24] to implement Support Vector Machines (SVM) algorithms. We choose the radial basis function (RBF) as SVM kernel to learn the classification model. We deploy the classification model in GeoCAM. For single- and multi-webpages, GeoCAM further extracts geographical information for landmarks.

3.3 Landmark Information Extraction

Once a webpage involves live webcams, we need to extract relevant context information from the webpage, especially the information essential to a landmark: its IP address and latitude/longitude pair. Below we present the details of the LG component for generating landmarks.

3.3.1 ***Webcam Identification.*** As we mentioned earlier, we can generate multiple landmarks from those webpages that host multiple webcams. For multiple webcam pages, we divide those webcams into different entities for extracting geographical information.

We use Algorithm 1 to extract different webcam entities from multiple webcam pages. In the beginning, we extract all HTML tags involving relevant context information (e.g., IP address/port, domain names, and latitude/longitude) as the candidate-tag set. For each HTML tag with webcam feature, we record the tag and its parent tag. We use the dictionary set *dict* to store those tags in the format of {parent tag, children set}. If two tags have the same parent tag, we append them into the same children set. Then, we determine the type of multiple webcam pages according to the structures of those candidate tags. Specifically, we partition webcam entities based on the following rule. If all tag items have the

Algorithm 1 Multiple Webcam Extraction

Input: *HTML*, webpage with multiple webcams
Output: *Results*, IP/Geo information of webcams
Candidates ← {HTML tags within webcam features}
Results ← []
dict ← {} ▷ store (parent tag, children Set)
while *length*(*Candidates*) > 0 **do**
  for all $x \in Candidates$ **do**
    parent ← parent tag of x
    Append(x, *dict*[*parent*])
  if all candidates share same parent element **then**
    for all $x \in Candidates$ **do**
      $< ip, geo >$ ← IP/Geo from x
      Append(<ip, geo>, *Results*)
    return *Results*
  else
    for all (i, Set) pairs $\in dict$ **do**
      if $len(Set) > 1$ **then**
        remove $\forall$ tags $\in Set$ from *Candidates*
      else
        replace *Set* with the tag i

same parent tag, the multi-webcam page is probably used to show several webcam entities under a particular topic. We extract the geographical information from all webcam entities, including the IP address and geolocation information. If those tags cannot be merged into the same root, the multi-webcam page is probably used to host a single webcam together with other recommended webcams from the website owner. We do not take recommended webcams into consideration because they are duplicated with webcams in other webpages. Note that recommended webcam tags share the same parent tag in the HTML list. We repeatedly remove these tags from the candidate set until reaching the root tag in the recommended webcam list.

3.3.2 ***The Regex Extraction.*** We use the regex to extract the webcam related information from webpages, including IP address and port, domain name, and geographical coordinate. They have distinctive character features, and the regex can achieve high accuracy in practice, as shown in Table 1.

IP address and port. When an IP address and port number are directly exposed on the webpages, we use the regex to extract them from the HTML file. Once an IP address is found, the LG generates a candidate landmark that is formatted as *[IP, (null, null)]*. Note that not all webcams expose their IP addresses and ports on webpages.

Domain name. Some webcams use web services to host live streams of webcams and hide real IP addresses. Web services registered their domain names and post their URL links for webcams. The LG extracts URL links or corresponding redirection links, and sends them to the WCP for scraping HTML files. If an IP address is extracted from a new HTML file, we need to validate its effectiveness. The rule is straightforward: if their heads (<head> and </head>), or titles (<title> and </title>) are the same, the IP address is identified as the webcam address.

Geographical Coordinate. The geographical coordinate refers to a *(longitude, latitude)* pair, which is used to directly present

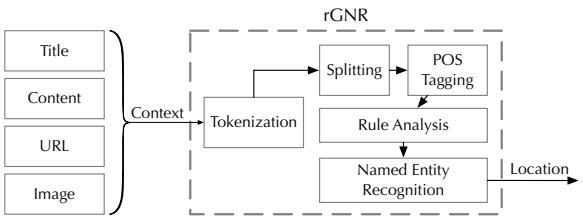


Figure 3: The rule-based geographical name recognition (rGNR)

the landmark geographical context. Note that not all webpages include geographical coordinates for webcams. There are two places to store geographical coordinates, including the HTML body and Frame. Frame is a common place to embed a webcam's geographical coordinate, e.g., Google Map. We directly use the regex (Table 1) to extract a webcam's latitude/longitude.

3.3.3 ***Geographical Name Recognition.*** Many webpages only expose geographical names for webcams and their latitudes/longitudes are not available. For example, the live stream on site pictimo.com (Figure 1) shows the geographical name "Glenwood Springs hottub, United States". We need to extract those geographical names, and convert them into the latitude/longitude pairs. There are two problems for directly using NER for identifying geographical names, leading to low precision and recall. First, webcam entity may have several geographical contexts, but they expose at different coarse levels, including non-normalized and general forms. Second, many geographical names use non-dictionary and non-English words, creating various name entities.

We propose the rule-based geographical name recognition (rGNR) to extract those geographical names from webpages. Figure 3 illustrates the overview of rGNR in the process of GeoCAM. The rGNR leverages the observation that many name entities are stored in fixed positions, including <title>/<meta>, URL, and images. The <title> and <meta> of a webpage usually contain a location description of a webcam. URL is an interesting place, where a webpage indicates the location information as a part of the URL. The reason is that a website might organize its webpages in a hierarchical structure based on the geographical name. Images or snapshots of webcams have been embedded with relevant location and timestamp information, e.g., the example in Figure 1. We only extract the content from those places as the rGNR input.

We convert sentences from those places into tokens. They are connected through *delimiters*, include "\" and webcam characters. We utilize the POS tagger to split sentences into different segments. POS taggers have four types of context, including capitalized characters, webcam characters, location words, and delimiters. Domain names and titles might be capitalized characters. Webcam characters usually appear together with geographical names in the same sentence, e.g., "webcam", "kamera"(Cezch, Indonesian), and webcam vendors. Location words are the textual descriptions before or after a specific geographical name, e.g., park or harbor. After segmenting, we apply the name entity recognition (NER) [30] to recognize those geographical names.

Once a geographical name is recognized, we convert the geographical name into the corresponding latitude/longitude pair. Here, we use the public geoname database to find its geographical coordinate. For providing IP-based geolocation services, the LG generates the landmarks that are formatted as the [IP, (lat, lon)].

Implementation. For webcam images, we use an optical character recognition engine Tesseract [29] which recognizes the text from an image. However, its performance is low due to low quality of images and non-dictionary words under different languages. For textual descriptions on webpages, we use the POS tagger tool [30] to obtain segments from webpages. We further use rGNR to recognize geographical names on webpages. We use open source geoname database provided by OpenStreetMap [22], a free online map service with over 20 million names and corresponding coordinates, to build geoname-coordinate mappings.

3.4 Geolocation Service

We employ the constraint-based geolocation [7] (CBG) method to derive a coarse-grained region for a target IP address based on a large number of webcam landmarks from GeoCAM. More specifically, the CBG method first estimates the geographic distance between the probe and the target. Then, CBG uses the geographical distance and multi-lateration to locate a target host. CBG outperforms the previous measurement-based geolocation techniques by reducing errors caused by inflated latencies and indirect routers.

In our implementation of CBG, we select 10 vantage points (VP) across the U.S. and Europe, and 16,863 webcam landmarks as the probe for locating a target host. We use the maximum likelihood estimation to further derive the latitude/longitude of the target host. (1) First, we calculate the conditional probability for the geographical distance and latency. (2) Then we use all the relative latencies of landmarks to infer the conditional probability of distances. For each conditional probability, we utilize the log-likelihood function to present the maximum likelihood function. We choose the maximum likelihood probability to present the latitude/longitude of the target host.

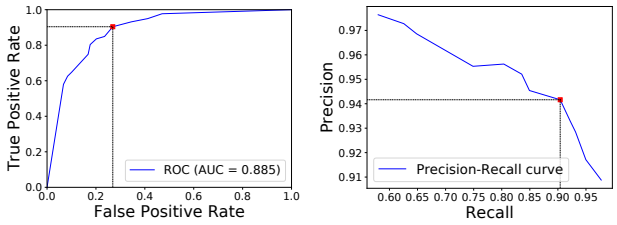
4 LANDMARK EXPERIMENTS

In this section, we validate the effectiveness of webcam landmarks created by GeoCAM based on two datasets.

4.1 Settings

We implemented a prototype of GeoCAM and ran it to automatically analyze 1.9 million webpages, using Ubuntu 14.04 on an AWS server with two 3.1GHz Intel Xeon Platinum 8175 vCPU, 8GB memories, and 10Gbps bandwidth. Here we detail the datasets used in this study.

Datasets. (1) We used *a labeled dataset* for training/testing the classification model and our geographical name recognition. This dataset contains 2,300 webpages and their corresponding webcams. These live webcams were manually collected from websites such as *pictimo*, *goowebcams*, *racamera*, and *insecam*. For every page, we manually provided its label, including *none*, *single*, and *multiple*. About 1,600 of them are tagged as *single*, 300 of them are tagged as *multiple*, and the rest (400) are tagged as *none* web pages. Each of them was manually checked to ensure that they were accurately extracted by GeoCAM. (2) We used a large-scale dataset from 1.9 million webpages to further validate the effectiveness of webcams for being used as landmarks. We manually extracted these popularity websites that are hosting live webcams on the Internet (see Table 9 in Appendix for details). On those websites, we ran the GeoCAM that scraped 1.9 million webpages between June 1st, 2018 and June 8th, 2018.



(a) ROC curve. **(b) Precision-Recall curve.**

Figure 4: ROC and PR curve of webpage classification.

Table 2: Compare geolocation entity recognition results of GeoCAM and CoreNLP.

Website	Language	False Negative Rate	
		GeoCAM	CoreNLP
goowebcams	English	4%	53%
racamera	Russian, English Polish, etc.	0%	76%
insecam	English, Chinese Russian, etc.	0%	48%
pictimo	English	0%	84%

4.2 Performance

Webpage Classification. We first evaluate the classification model's performance using the labeled webcam dataset. For the multi-class classification, the SVM algorithm with RBF kernel derives the boundary between one class to other classes. We divide the dataset into the training set (1,610 webpages) and the test set (690 webpages). We use false positive rate (FPR), precision, and recall to measure its performance, where FPR is the ratio of $|FP|/|FP+TN|$, the precision equals to $|TP|/|FP+TP|$, and the recall is $|TP|/|TP+FN|$. TP is the number of true positives, FN is the number of false negatives, FP is the number of false positives, and TN is the number of true negatives. Figure 4a shows the ROC curve of the TPR and FPR, and Figure 4b illustrates the precision-recall (PR) curve of the webcam classification performance. Our prototype achieves a precision of 94.2%, a recall of 90.4%, and a FPR of 21.4% in determining whether a webpage has a webcam. In practice, GeoCAM performance is acceptable for classifying web pages with webcams.

Webcam Location Extraction. We then evaluate the webcam location extraction of GeoCAM. We use two methods: (1) our proposed GeoCAM and (2) a general NER method called CoreNLP [30]. For each webpage, we manually extract the geographical location

Table 3: The number of webpages collected by GeoCAM over 100 websites.

Stage	Webpage Number
Before filtering out	1,913,277
After filtering out	282,920
None-webcam hosting	1,630,357
Single webcam hosting	256,210
Multi-webcam hosting	26,692

Table 4: Relevant information extraction from webcam-related webpages.

Item	Number
Live streams of webcams	216,974
[Latitude, Longitude]	57,909
Geographical names	187,119
IP addresses of webcams	378,899

as the ground truth. Table 2 lists the performance of GeoCAM and CoreNLP. The CoreNLP has a high false negative rate (FNR) on geographical name extraction. For instance, its precision is only 24% on the site racamera.com. By contrast, GeoCAM achieves very promising performance for extracting geographical names.

4.3 Webcam Landmark Validation

Landscape. Table 3 lists the number of webpages collected by GeoCAM over 100 websites. In total, we collected 1,913,277 webpages. After filtering out unnecessary pages by WCP, there are 282,902 webpages remaining. Among those webpages, there are 256,210 pages hosting single live webcam, and 26,692 pages hosting multiple webcams. We observe that nearly 1.5 million webpages are not associated with live webcams, and thus we drop them out from the candidate set.

The Number of Webcam Landmarks. Table 4 lists the amount of relevant webcam information extracting from webpages. In total, the GS component extracts 378,899 IP addresses, 187,119 geographical names, 57,909 latitude/longitude pairs. It finally has 216,974 webcam candidates with IP addresses and geographical information (either latitude/longitude or geographical name). A landmark involves an IP address and its latitude/longitude, which we use a key-value pair to store. Note that those webcams might be overlapped with one another. The reason is that websites might scrape webpages from others. For instance, web-online24.ru and tvway.ru both collect webcams from hotel-novoros.ru. We use their IP addresses to filter out the duplicated ones. In other words, if two webcams have the same IP address, we remove the duplicated one. We obtain 16,863 landmarks with unique IP addresses and accurate latitude/longitude information.

Webcam Landmark Stability. To validate the stability of webcam landmarks collected by GeoCAM, we measure the number of available webcam landmarks and webpage URLs along with time. We illustrate their dynamic changes in Figure 5 using the datasets collected on May 11, 2019 and October 5, 2019, as well as

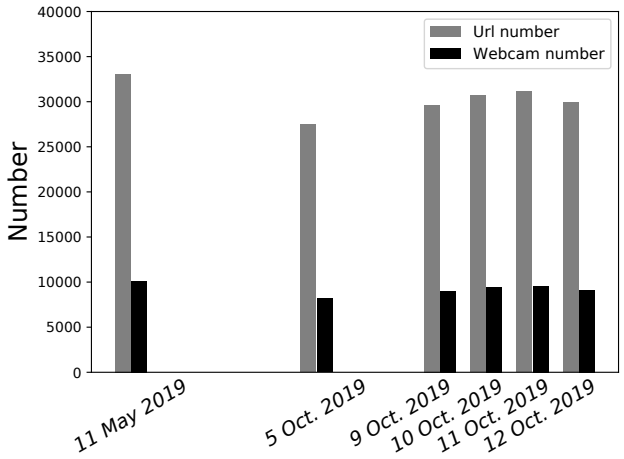


Figure 5: Dynamic changes of URL/webcam landmark along with time.

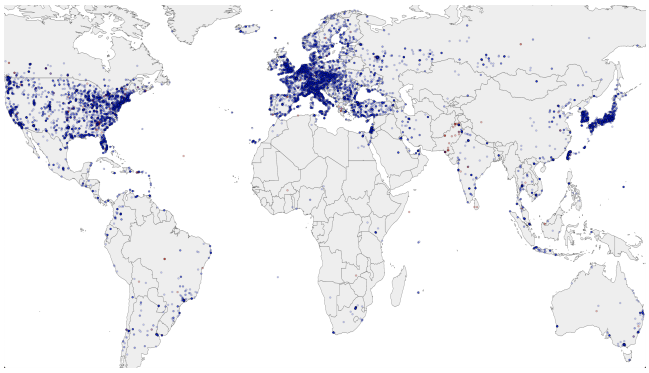


Figure 6: Geographical distribution of webcam landmarks over the globe.

the most recent datasets collected from October 9, 2019 to October 12, 2019 for four consecutive days. We can see that the number of available webcam landmarks remains stable even after 5 months (80.45%), with much less variations within one week, indicating their long-term stability.

Due to the dynamic nature of IP addresses, a webcam might not be always tied to a specific IP address. However, we observe that only 8.55% webcam landmarks change their IP addresses every 24 hours. This is because the most common default IP lease time is 8 days, and it could be further renewed for weeks or months or even longer. Even though a few webcam landmarks may change, GeoCAM can perform a periodic update to keep tracking of those websites and webcams.

Landmark Coverage. We conduct the analysis on webcam landmark coverage, including geographical distribution, AS distribution, and domain name distribution.

(i) Geographical Coverage. We first compare the geographical coverage of our landmarks with others. Figure 6 depicts the world map, where the blue dots represent our webcam landmarks, and

Table 5: The Top 10 countries of webcam landmarks.

Country	Number	Country	Number
USA	4,277	Turkey	584
France	1,001	UK	529
Italy	947	South Korea	424
Japan	864	Czech	392
Germany	631	Netherland	387

Table 6: The Top 10 cities of webcam landmarks.

City	Number	City	Number
Woods County	172	New York City	55
Sant'Agnello	110	Forlì del Sannio	53
Shintoku	105	Seoul	47
London	65	Shinjuku	44
Kostanay	59	Bangkok	41

the red dots are the landmarks from other platforms (Planetlab [25], PingER [17], and PerfSONAR [9]). As an example, Figure 10 depicts a more detailed geographical distribution of landmarks in the UK and Ireland, which is shown in the Appendix. We can see that our webcam landmarks almost cover all geographical places of existing landmarks from others with much higher density. Europe and North America are the traditional regions that open source landmarks cover well, but the other regions are rarely covered. By contrast, our webcam landmarks well cover much more geographical areas, including Russia, South America, Turkey, India, and China. Interestingly, Japan has the highest density of webcam landmarks.

Here we briefly describe the country/city distribution for geographical coverage of webcam landmarks. In total, webcam landmarks cover 170 countries and 6,448 cities. About 25% of webcams are from North America and Europe. Table 5 lists the top 10 countries that webcam landmarks cover, and those countries are from North America, Europe, and Asia. We also list the top 10 cities covered by webcam landmarks, as shown in Table 6.

(ii) AS Coverage. We also analyze the autonomous system (AS) coverage of webcam landmarks. We build *block-asn* mappings based on the existing BGP routing table analysis data [2] supported by APNIC. In total, we find that webcam landmarks cover nearly 2,875 ASes, much more than open source landmarks that are centered on academical networks. Table 7 lists the top 5 ASes covered by webcam landmarks, where most of webcams are from residential networks.

(iii) Domain Name Coverage. Domain names of hosts are usually utilized to infer geolocation information. For instance, from a hostname of *admin.umass.edu*, one may infer that the corresponding node belongs to the University of Massachusetts at Amherst. Therefore, we further analyze the domain name coverage of webcam landmarks. We use reversed DNS lookups to obtain domain names associated with IP addresses from our webcam landmarks. We resolve webcams' IP addresses to collect their pointer records (PTR). Note that not all IP addresses have a reverse entry for PTR. For 16,863 landmarks, there are 11,950 IP addresses with PTR records

Table 7: Top 5 ASes covered by webcam landmarks.

AS ID	AS network	Number
AS7922	COMCAST, US	774
AS9121	TTNET, TR	507
AS4766	KIXS-AS-KR Korea Telecom, KR	393
AS3269	ASN-IBSNAZ, IT	390
AS22394	CELLCO, US	364

but 4,913 of them without PTR records. We observe that 311 PTR items have geographical clues, where 309 have the edu TLD and 9 have the gov TLD. When a record ends with a distinguishable TLD (edu or gov), we can infer their location information.

5 GEOLOCATION PERFORMANCE

In this section, we use the webcam landmarks to provide geolocation services for approximately pinpointing the geolocation of an Internet host. Further, we compare our GeoCAM with prior geolocation algorithms, open-source landmarks, and commercial geolocation databases.

Geolocation Comparison. We compare two geolocation approaches: (1) GeoCAM based on webcam landmarks and (2) CBG [7] based on open source landmarks. Since the number of open source landmarks is limited, we only implemented a CBG approach, similar to SLG [31], which uses multilateration to shrink the region and pinpoint a target to the node with the smallest relative latency.

We first compare two approaches using the dataset collected from residential networking communities. We selected 200 targeted hosts in Europe from our webcam dataset, where most of they are from residential communities. We manually inspected those snapshots of live streams of webcams, and searched relevant geographical contexts in Google Map to obtain their ground truth latitude/longitude information. Figure 7 illustrates the CDF of the geolocation errors on the residential hosts. The X-axis is the error distance (kilometer) between the ground true data and the estimated data. We observe that GeoCAM achieves 82.02% higher performance than CBG. Under GeoCAM with webcam landmarks, 40% of hosts are less 50KM error and 80% of hosts are less than 120 KM error. Our webcam landmarks have hosts from both academic and residential networking communities. Hence, our approach is more stable in various networking environments. Due to the lack of landmarks from residential networks, CBG with open source landmarks has the worst performance.

We also compare two approaches in academic institutions. Figure 8 shows the cumulative distribution function (CDF) of the geolocation errors on the academic hosts. Those targeted hosts (150) come from Planetlab [25], PingER [17], and PerfSONAR [9]. Every host has an IP address and corresponding latitude/longitude information. We observe that two approaches perform differently in the geolocation accuracy, where our approach is close to 40% of hosts with less 25KM error, and 80% of hosts with less than 130KM error. Although open source platforms usually provide the coverage for academic institutions, it is clear that our webcam landmarks also have a high coverage for academic networking communities.

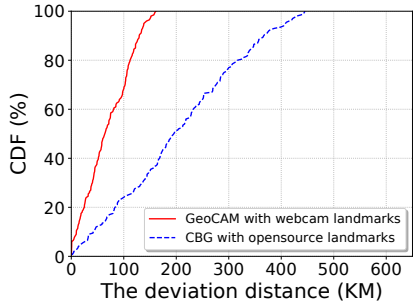


Figure 7: Residential networking communities.

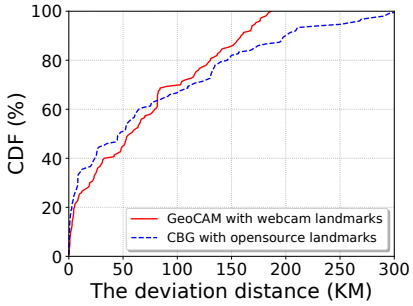


Figure 8: Academic networking communities.

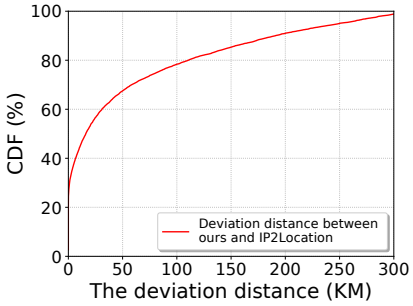


Figure 9: Compared with the commercial geolocation database.

Table 8: The number of visible and accessible landmarks.

Data source	Num.	Scope
GeoCAM	16,863	Academic & Residential
PlanetLab [25]	420	Academic
PerfSONAR [9]	642	Academic
PingER [17]	127	Academic
RIPE Atlas [27]	458	Residential

Open Source Landmarks. We compare our webcam landmarks with open source ones, including Planetlab [25], PingER [17], PerfSONAR [9], and RIPE Atlas [27]. Table 8 lists the number of visible and accessible landmarks. Note that we only incorporate those visible and accessible landmarks. For instance, PerfSONAR [9] states that it has 2,143 nodes; however, most of them has become unavailable, due to usage restrictions on these nodes, and only 642 of them are still accessible. Moreover, those landmarks are from academical communities, rarely from other communities. Our webcam landmarks distribute in various types of places with a high geographical resolution, including residential and other environments. Overall, our webcam landmarks are two orders of magnitude more than the landmarks used in prior works.

Commercial Geolocation Database. We further compare our webcam landmarks with IP2Location [11], a popular commercial geolocation database. Figure 9 shows the CDF curve of deviation distances between our landmarks and IP2Location. There are 44% webcam landmarks having the deviation less than 10km, compared with IP2Location. Note that the commercial databases cannot provide ground truth labels for our landmarks because those databases only provide coarse mappings between physical locations and IP hosts.

6 DISCUSSION

Our study shows that GeoCAM makes a first step towards fully automated landmark generation based on webcam devices. On the other hand, our current design is still preliminary. Here, we discuss the limitations and concerns of GeoCAM.

Coverage Limitation. In the data collection, we crawled 100 different websites and retrieved 1.9 million webpages. However, we acknowledge that GeoCAM cannot exhaustively gather all live webcams on the Internet. Besides those 100 websites, some less popular websites (e.g., blogs or forums) might also distribute webcams to the public. In our future work, we plan to crawl more websites to find a even larger number of webcams for being used as landmarks.

Privacy Concerns. Since we generate landmarks based on webcams, user privacy could be a concern. However, we only scrape webpages from websites and extract location information of candidate landmarks, which are already publicly accessible on the Internet. If a landmark violates user privacy or permission, we can directly remove it from the candidate set. More importantly, for those webcams used as landmarks, we never attempted to log into or control them.

Webcam Image Recognition. As aforementioned, we used the Tesseract [29] to recognize the text from a webcam image. Here, we only focus on embedded characters on the webcam image (geographic names and timestamps). With only a few exceptions, geographic names derived from webcam images are already included in the geolocation names from HTML elements. Overall, only a small portion of texts extracted from images are unique and most of them are timestamps.

7 RELATED WORK

IP-based geolocation has two categories: client-dependent and client-independent. The client-dependent approach relies on the client-side support, such as GPS and WiFi signals. GeoCAM is a client-independent approach that does not require any client-side support. In this section, we survey the previous client-independent works.

Data mining-based approach. Given geographical information in public webpages/datasets, this approach derives IP-location relationships using data mining techniques. Liu et al. [16] utilized check-in data from social networks to generate IP-user-location relationships. They mined check-in patterns from login logs and inferred users' geolocation and IP addresses (such as *home*, *office*, and *others*). Such a method requires private login logs and only works for active users. Padmanabhan and Subramanian [23] proposed geographical clusters based on address prefixes in the border gateway protocol (BGP) tables, and pinpointed a host using the landmarks within the same cluster. Huffaker et al. [10] leveraged geographical hints from hostnames to geolocate a large set of routers on

the Internet, but geo-hints of hostnames are limited and probably inconsistent with actual locations.

Network measurement-based approach. This method uses latencies and topologies among VPs and targets to estimate the geo-location of a target host. Gueye et al. [7] proposed a constraint-based geolocation (CBG), which uses geographical distance and multilateration to locate a target host. CBG utilizes the bestline estimation to reduce errors induced by inflated latencies and indirect routers. Katz-Bassett et al. [12] proposed a topology-based geolocation (TBG), which introduces network topology constraints for improving the performance of CBG. Wong et al. [32] proposed a generic framework called Octant that locates IP hosts by incorporating latency, network topology, and hostnames. Laki et al. [13] proposed Spotter to use the highest probability density to geolocate target hosts.

Geolocation database. There are several available IP geolocation databases, including MaxMind GeoLite2 [19], GeoIP2 [18], IP2Location [11], and NetAcuity [4]. Those databases provide physical locations to IP addresses at the country-level and city-level granularities. Prior works [6, 26, 28] have evaluated the accuracy of databases from endpoints to router geolocations. Poese et al. [26] found that the number of unique geographical locations are much less than the number of IP blocks, which cannot build accurate mappings between physical locations and IP hosts. Shavitt and Zilberman [28] demonstrated that most of databases cannot archive the accuracy they acclaimed at the country level, and perform poorly at the city level. Gharaibeh et al. [6] analyzed router geolocation accuracy across those databases, and found that databases are not reliable for geolocating routers at the city level.

Landmark. The accuracy of IP-based geolocation is heavily dependent on the density of landmarks. There are several platforms for providing landmarks with known IP addresses and accurate geolocations to the public. PlanetLab [25] is a geographically distributed network testbed, running 1,353 nodes at 717 sites, most of which are universities or research institutions. As yet, we find that only 420 nodes are still available. PerfSONAR [9] is a network measurement tookit designed to provide federated coverage of paths, and help to establish end-to-end usage expectations. PingER [17] is a monitoring infrastructure to understand network performance and allocate resources to optimize performance between laboratories.

Mining web services has been proposed to increase the number of landmarks. Guo et al. [8] and Li et al. [14] extracted geographical information from webpages and built the relationships between IP addresses of websites and their physical locations. However, the location information extracted from websites is not the actual location of websites. Those landmarks belong to the /24 subnet, only at the city-level granularity. Wang et al. [31] also leveraged geographical information on websites and searched them through online maps for generating landmarks. Unfortunately, for those landmark generation approaches based on mining web services, the mappings among domain names, IP addresses, and locations are not stable, due to the widely used cloud services and CDN. Thus, their landmarks have unstable and unverifiable issues, resulting in unavailability of landmarks. By contrast, we generate landmarks based on live webcams that are stable with time and can be verified by manual inspection on their live streams. Moreover, GeoCAM generates landmarks at the latitude/longitude granularity.

8 CONCLUSION

IP-based geolocation heavily relies on the number of high-quality landmarks. As the pervasiveness of IoT systems, an increasing number of webcams connected to the Internet have become an ideal set of candidates for being used as landmarks. In this paper, we proposed a framework GeoCAM to automatically generate webcam landmarks and provide IP-based geolocation services. Specifically, GeoCAM periodically monitors those websites that host live webcams and uses the natural language processing technique to extract the IP addresses and latitudes/longitudes of webcams. We conducted experiments to evaluate the performance and effectiveness of GeoCAM. Our results show that GeoCAM automatically detects webcams with 94.2% precision and 90.4% recall, and can generate 16,863 stable and fine-grained landmarks. These webcam landmarks are two orders of magnitude more than the landmarks used in existing geolocation services, and thus GeoCAM is capable of providing a geolocation service with high accuracy and wide coverage.

ACKNOWLEDGMENTS

We are grateful to anonymous reviewers for their insightful feedback. The work was partially supported by National Key R&D Program of China (No. 2018YFB0803402), National Natural Science Foundation of China (No. 61972024 and No. 61672092), Key Program of National Natural Science Foundation of China (No. U1766215), and Fundamental Research Funds for the Central Universities of China (No. 2018JBZ103).

REFERENCES

[1] Beautiful Soup 2004. Beautiful Soup. https://www.crummy.com/software/BeautifulSoup/. Accessed: 10-April-2019.
[2] BGP Routing Table Analysis 1999. BGP Routing Table Analysis. http://thyme.apnic.net/. Accessed: 10-April-2019.
[3] Common Crawl 2010. Common Crawl. https://commoncrawl.org/. Accessed: 10-April-2019.
[4] Digital Envoy 1999. NetAcuity. https://www.digitalelement.com/solutions/. Accessed: 10-April-2019.
[5] Xuan Feng, Qiang Li, Haining Wang, and Limin Sun. 2018. Acquisitional Rule-based Engine for Discovering Internet-of-Things Devices. In *27th USENIX Security Symposium (USENIX Security 18)*. USENIX Association, Baltimore, MD, 327–341. https://www.usenix.org/conference/usenixsecurity18/presentation/feng
[6] Manaf Gharaibeh, Anant Shah, Bradley Huffaker, Han Zhang, Roya Ensafi, and Christos Papadopoulos. 2017. A Look at Router Geolocation in Public and Commercial Databases. In *Proceedings of the 2017 Internet Measurement Conference (IMC '17)*. ACM, New York, NY, USA, 463–469. https://doi.org/10/gfvgsq
[7] Bamba Gueye, Artur Ziviani, Mark Crovella, and Serge Fdida. 2006. Constraint-Based Geolocation of Internet Hosts. *IEEE/ACM Trans. Netw.* 14, 6 (Dec. 2006), 1219–1232. https://doi.org/10/ffh2c2
[8] C. Guo, Y. Liu, W. Shen, H. J. Wang, Q. Yu, and Y. Zhang. 2009. Mining the Web and the Internet for Accurate IP Address Geolocations. In *IEEE INFOCOM 2009 (INFOCOM'09)*. 2841–2845. https://doi.org/10/cmv5nd
[9] Andreas Hanemann, Jeff W. Boote, Eric L. Boyd, Jérôme Durand, Loukik Kudarimoti, Roman Łapacz, D. Martin Swany, Szymon Trocha, and Jason Zurawski. 2005. PerfSONAR: A Service Oriented Architecture for Multi-domain Network Monitoring. In *International Conference on Service-Oriented Computing (ICSOC)*. Berlin, Heidelberg, 241–254.
[10] Bradley Huffaker, Marina Fomenkov, and kc claffy. 2014. DRoP: DNS-Based Router Positioning. *SIGCOMM Comput. Commun. Rev.* 44, 3 (July 2014), 5–13. https://doi.org/10/f6f3k8
[11] IP2Location 2001. IP2Location. https://www.ip2location.com/. Accessed: 10-April-2019.

[12] Ethan Katz-Bassett, John P. John, Arvind Krishnamurthy, David Wetherall, Thomas Anderson, and Yatin Chawathe. 2006. Towards IP Geolocation Using Delay and Topology Measurements. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement (IMC '06)*. ACM, New York, NY, USA, 71–84. https://doi.org/10/bdnhxh
[13] S. Laki, P. Mátray, P. Hága, T. Sebők, I. Csabai, and G. Vattay. 2011. Spotter: A Model Based Active Geolocation Service. In *2011 Proceedings IEEE INFOCOM (INFOCOM'11)*. 3173–3181. https://doi.org/10/cn225w
[14] D. Li, J. Chen, C. Guo, Y. Liu, J. Zhang, Z. Zhang, and Y. Zhang. 2013. IP-Geolocation Mapping for Moderately Connected Internet Regions. *IEEE Transactions on Parallel and Distributed Systems* 24, 2 (Feb. 2013), 381–391. https://doi.org/10/gdwtgf
[15] Q. Li, X. Feng, R. Wang, Z. Li, and L. Sun. 2018. Towards Fine-grained Fingerprinting of Firmware in Online Embedded Devices. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*. 2537–2545. https://doi.org/10.1109/INFOCOM.2018.8486326
[16] H. Liu, Y. Zhang, Y. Zhou, D. Zhang, X. Fu, and K. K. Ramakrishnan. 2014. Mining Checkins from Location-Sharing Services for Client-Independent IP Geolocation. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications (INFOCOM'14)*. 619–627. https://doi.org/10/gfvgr6
[17] W. Matthews and L. Cottrell. 2000. The PingER project: active Internet performance monitoring for the HENP community. *IEEE Communications Magazine* 38, 5 (May 2000), 130–136. https://doi.org/10.1109/35.841837
[18] MaxMind, Inc 2002. MaxMind GeoIP2 Database MaxMind. https://www.maxmind.com/en/geoip2-databases/. Accessed: 10-April-2019.
[19] MaxMind, Inc 2002. MaxMind GeoLite2 Free Downloadable Databases. https://dev.maxmind.com/geoip/geoip2/geolite2/. Accessed: 10-April-2019.
[20] James A. Muir and Paul C. Van Oorschot. 2009. Internet Geolocation: Evasion and Counterevasion. *ACM Comput. Surv.* 42, 1 (Dec. 2009), 4:1–4:23. https://doi.org/10/fwhxsm
[21] David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes* 30, 1 (2007), 3–26.
[22] OpenStreetMap 2004. OpenStreetMap Wiki. https://wiki.openstreetmap.org/w/index.php. Accessed: 10-April-2019.
[23] Venkata N. Padmanabhan and Lakshminarayanan Subramanian. 2001. An Investigation of Geographic Mapping Techniques for Internet Hosts. In *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '01)*. ACM, New York, NY, USA, 173–185. https://doi.org/10/bmthnj
[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
[25] Larry Peterson, Tom Anderson, David Culler, and Timothy Roscoe. 2003. A Blueprint for Introducing Disruptive Technology into the Internet. *SIGCOMM Comput. Commun. Rev.* 33, 1 (Jan. 2003), 59–64. https://doi.org/10/cp6w9p
[26] Ingmar Poese, Steve Uhlig, Mohamed Ali Kaafar, Benoit Donnet, and Bamba Gueye. 2011. IP Geolocation Databases: Unreliable? *SIGCOMM Comput. Commun. Rev.* 41, 2 (April 2011), 53–56. https://doi.org/10/d3cbbx
[27] RIPE Atlas 2016. RIPE Atlas. https://atlas.ripe.net/. Accessed: 10-April-2019.
[28] Y. Shavitt and N. Zilberman. 2011. A Geolocation Databases Study. *IEEE Journal on Selected Areas in Communications* 29, 10 (Dec. 2011), 2044–2056. https://doi.org/10/dgb3d3
[29] Tesseract 1984. Tesseract Open Source OCR Engine. https://github.com/tesseract-ocr/tesseract. Accessed: 10-April-2019.
[30] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1 (NAACL '03)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 173–180. https://doi.org/10.3115/1073445.1073478
[31] Yong Wang, Daniel Burgener, Marcel Flores, Aleksandar Kuzmanovic, and Cheng Huang. 2011. Towards Street-Level Client-Independent IP Geolocation. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation (NSDI'11)*. USENIX Association, Berkeley, CA, USA, 365–379.
[32] Bernard Wong, Ivan Stoyanov, and Emin Gün Sirer. 2007. Octant: A Comprehensive Framework for the Geolocalization of Internet Hosts. In *Proceedings of the 4th USENIX Conference on Networked Systems Design & Implementation (NSDI'07)*. USENIX Association, Berkeley, CA, USA.

APPENDIX

Table 9: Full list of 100 webcam aggregator websites

Domain names of aggregator websites			
te.ua	cvlt.ch	krvn.com	pictimo.com
101.at	dczo.ch	ktru.org	seovisit.ru
1tv.me	ezpc.ru	mosgt.ru	turistik.cz
1vl.ru	fcdp.ro	phys.org	webkams.com
bay.tv	imao.sk	skjm.com	worldcam.eu
bye.ch	iski.it	syote.fi	earthcam.com
cic.ba	keri.ee	tveye.su	geocities.jp
nsf.se	lkvm.cz	tvjoy.ru	livecam.asia
och.nu	lszp.ch	tvway.ru	niki-surf.ru
ozi.ro	pike.jp	xa911.cn	racamera.com
r4n.it	rcnt.es	ipcams.ch	reklboard.ru
sld.uk	rftp.ru	mosday.ru	uamuseum.com
ufv.ca	seom.bg	telus.net	walltrend.ro
wff.lv	thai.mn	ttrix.com	webcambg.com
www.hr	trop.ch	dibcras.ro	camhacker.com
ycs.at	wdhd.ru	fboller.de	checkcams.com
9310.no	zvho.ch	lavrsen.dk	inchbeach.com
a2ch.ru	2ch.live	letunam.ru	opentopia.com
abgc.pl	56kb.com	sravni.org	roxy-world.ro
abvm.fr	camua.ru	webcams.bg	seo-surf.info
aset.no	etar.org	geoearth.ru	skidkamera.se
aupm.fr	fgsrb.ru	geometeo.it	skiweather.eu
bbox.ch	ilm24.ee	gobefore.me	top-kamery.cz
bowa.dk	ketry.cz	insecam.org	webkameror.se
bswr.de	kneb.com	panorama.sk	goowebcams.com

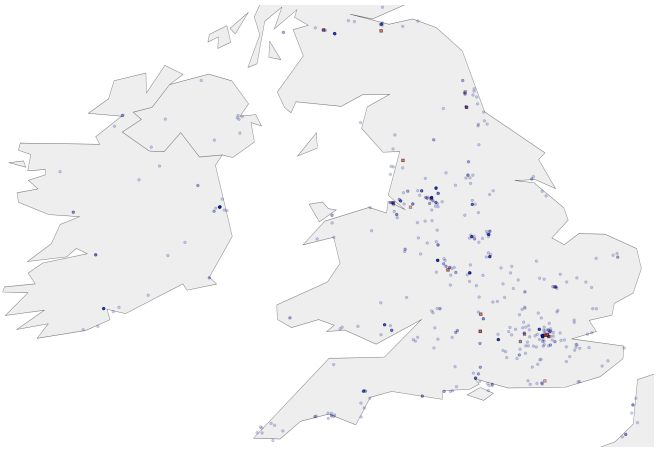


Figure 10: Geographical distribution of landmarks in the UK and Ireland.