Defending Against GPS Spoofing by Analyzing Visual Cues

Chao Xu

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Applications

Gang Wang, Chair
Bimal Viswanath
Danfeng (Daphne) Yao

April 30, 2020
Blacksburg, Virginia

Keywords: GPS Spoofing; Location Verification; System Design

Defending Against GPS Spoofing by Analyzing Visual Cues

Chao Xu

(ABSTRACT)

Massive GPS navigation services are used by billions of people in their daily lives. GPS spoofing is quite a challenging problem nowadays. Existing Anti–GPS spoofing systems primarily focus on expensive equipment and complicated algorithms, which are not practical and deployable for most of the users. In this thesis, we explore the feasibility of a simple text-based system design for Anti–GPS spoofing. The goal is to use the lower cost and make the system more effective and robust for general spoofing attack detection. Our key idea is to only use the textual information from the physical world and build a real-time system to detect GPS spoofing. To demonstrate the feasibility, we first design image processing modules to collect sufficient textual information in panoramic images. Then, we simulate real-world spoofing attacks from two cities to build our training and testing datasets. We utilize LSTM to build a binary classifier which is the key for our Anti–GPS spoofing system. Finally, we evaluate the system performance by simulating driving tests. We prove that our system can achieve more than 98% detection accuracy when the ratio of attacked points in a driving route is more than 50%. Our system has a promising performance for general spoofing attack strategies and it proves the feasibility of using textual information for the spoofing attack detection.

Defending Against GPS Spoofing by Analyzing Visual Cues

Chao Xu

(GENERAL AUDIENCE ABSTRACT)

Nowadays, people are used to using GPS navigation services in their daily lives. However, GPS can be easily spoofed and the wrong GPS information will mislead victims to an unknown place. There are some existing methods that can defend GPS spoofing attacks, but all of them have significant shortcomings. Our goal is to design a novel system, which is cheap, effective, and robust, to detect general GPS spoofing attacks in real-time. In this thesis, we propose a complete system design and evaluations for performance. Our system only uses textual information from the real physical world and virtual maps. To get more accurate textual information, we use advanced techniques to process images and recognize text in images. We also use a neural network to help with detection. By testing with datasets in two cities, we confirm the promising performance of our system for general GPS spoofing attack strategies. We believe that textual information can be further developed in the Anti–GPS spoofing systems.

Acknowledgments

Firstly, I would like to thank my advisor Dr. Gang Wang for his patient guidance and constructive advices during this work. He gives me valuable feedbacks and inspiring suggestions, which make this work go well. I also would like to give thanks to my committee members Dr. Danfeng Yao and Dr. Bimal Viswanath for their feedbacks and comments on this work. I want to give a lot of thanks to colleagues in our lab for their support and encouragement. I am extremely grateful to my parents for their caring and support throughout these years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the development of cities, the road environment is getting more and more complicated. Thus, people are becoming more and more dependent on mobile navigation services, like the Google Map. Especially, with the development of unmanned vehicle technology, the safety of GPS navigation systems has also been challenged like never before.

GPS devices can be easily controlled when attackers inject falsified GPS signals [28]. Especially, researchers have found a way to manipulate the road navigation systems by spoofing the GPS inputs and it can deviate victims from the actual route by several kilometers [32].

So far, the research for Anti–GPS spoofing is still an open question regarding whether we can have a real-time, portable, and cheap device to detect the GPS spoofing efficiently. The problem is critical considering that GPS signals can be hardly encrypted because modifications to the existing GPS are difficult to achieve. In addition, the deployment of sensors and usage of Computer Vision techniques have very low effectiveness and uncertain robustness.

In this thesis, we propose a new solution and system design to implement a real-time detection for general GPS spoofing attack strategies. The goal is to design a system that has low cost and achieves effective and robust performance for general attack strategies. The key intuition is to use textual information along the driving routes. Like humans would remember the important textual information along the routes to estimate the current location. In the same way, we can judge whether we are on the right path when we compare a series of text

messages. Besides, although there are many places on a route that do not have the textual information, we can still estimate the position through textual information at previous data points on the trajectory. Also, it is possible to build a text-based reference dataset with a low cost of storage and computation. The reference dataset is a set of textual information with correct corresponding location coordinates.

To understand the feasibility of text-based location verification, we take 4 key steps. First, we obtain a relatively complete panorama database of several cities. Second, we design several major modules for image processing and select some of the best algorithms. Third, we perform the detection based on the correct reference data obtained in real-time. The ideal situation is that we can have a local reference dataset. Finally, we adopt the simulation method to evaluate our system in different cities.

In summary, this thesis makes two key contributions.

- We propose a novel system design for Anti-GPS spoofing. The proposed system design is extensively evaluated using the simulation system.
- We implement the text-based location verification with low-cost and promising performance. Tests from multiple cities confirm the detection ability for general spoofing attack strategies.

We hope this system design and its performance can help to raise more attention in the community to develop practical and deployable text-based Anti-GPS spoofing mechanisms to protect the massive GPS device users and emerging GPS-enabled autonomous systems.

Chapter 2

Background and Motivation

We start by introducing the background of GPS spoofing and different methods to prevent it, then we describe our research goals and approaches.

2.1 Background of GPS spoofing

GPS spoofing has been proposed for many years. It has been applied in many scenarios, ranging from ship-based GPS spoofing to small vehicle-mounted GPS spoofing. The GPS signal can be very vulnerable to spoofing attacks.

There are two main ways to implement GPS spoofing attacks. First, attackers can rebroadcast GNSS signals recorded at another place or time; Second, attackers can generate and transmit modified satellite signals. The most common GPS spoofing attacks firstly take control of victim GPS receiver to follow the spoofing signal. Then, attackers can manipulate the GPS receiver to mislead users [32]. Figure 2.1[1] shows the example how GPS spoofing attacks happen.

[1] https://www.orolia.com/support/skydel/vehicle-spoofing

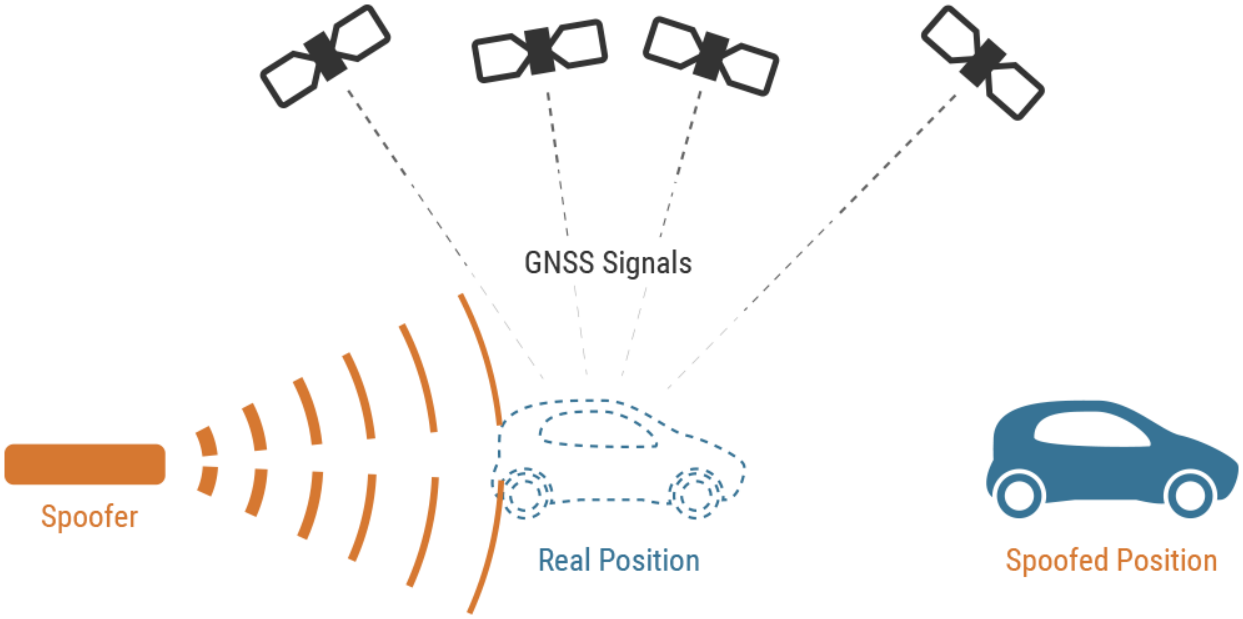


Figure 2.1: Example of GPS spoofing attacks

2.1.1 Existing Countermeasures

Generally, there are two classes of existing countermeasures based on how much they need to modify the existing GPS. Changing either the GPS satellites [12, 30], ground infrastructure [4, 10, 19, 25, 27], or the GPS receivers [9, 14, 18, 22, 33] can help to prevent the GPS spoofing. These countermeasures have high effectiveness and robustness for all general kinds of GPS spoofing attacks. However, they all require the high cost of implementation and are difficult to prevail in a short term. On the other side, some researches have tried to use cameras/LIDARs on mobile devices or vehicles to do vision-based localization [5, 23], which does not need to modify the existing GPS. SLAM (Simultaneous Localization And Mapping) can also be a method to do image-based localization [31]. However, the performance and effectiveness of these computer vision-based localization methods still need to be improved, and their robustness also needs to be further confirmed.

2.2 Related Work

Methods for localization and place recognition mainly rely on environmental images. Researchers firstly come up with sparse local features on images, like SIFT [15] and Vector of Locally Aggregated Descriptors (VLAD) [11]. However, this kind of sparse feature will be influenced by dramatic illumination changes. Then, the sequential information has be exploited in [16] [20] [21]. They are trying to use a sequence of image descriptors and build graph-based models. Since the development of DeepLearning, recent papers focus more on using deep neural networks to encapsulate some semantic information. A CNN-based image descriptor has been proposed as NetVLAD in [1]. In [6] [7], novel deep neural networks are designed and trained for image retrieval. However, the above works focus on image processing and retrieval in large image databases, making them unlikely to achieve low cost and high effectiveness for general spoofing attack strategies.

Recently, the advance of text detection and text recognition in the wild paves the way for using textual information for localization. In [29], it is the first time that textual information is proposed to navigate a robot. Hong et al. [8] propose a place recognition system and demonstrate the feasibility of utilizing the scene texts to solve the place recognition and topological localization problems in urban areas. Radwan et al. [24] associate texts to a map with landmarks and corresponding text labels and then estimate the pose of a camera. Our work differs from them in terms of verification goals and methods. Our goal is to verify if we currently have the correct GPS signal. They all focus on the estimation of the camera position. Compared to those text-based localization papers, we have made a significant contribution by proposing new system design for Anti–GPS spoofing, and more importantly proving that textual information can achieve a promising detection performance for general spoofing attack strategies.

2.3 Our Motivations

Anti–GPS spoofing is an urgent problem that needs to be addressed. Existing methods and countermeasures are not feasible for large-scale applications, or still have great uncertainty and low efficiency. Some image-based localization works try to reduce the accuracy rate to improve their performance in space and time. Even so, the cost of storage and computation for a large number of images are still extremely difficult problems. In this thesis, we seek to provide a more efficient and lightweight system to detect GPS spoofing attacks. We examine the whole dataflow in the system, from getting panoramic images, to monitoring GPS spoofing status. More importantly, for the first time, our system only uses textual information from the physical world and virtual maps to detect GPS spoofing attacks. In this way, it only takes very little cost of time and storage for computation. Furthermore, the heaviest computation in this system is the extraction of text features from panoramic images in the physical world. Ideally, we have a built-in text reference dataset so that we get rid of heavy computation on the virtual side. Instead of unstable and inefficient image retrieval, we only use textual information for our location verification. In addition, using textual information allows us to ignore dynamic things, like cars' movement along the route. It can help to reduce the uncertainty for the system to detect the GPS spoofing.

2.4 Methodology Overview

In this section, we describe our methodology to develop our system and how our system works for anti–GPS spoofing. We only describe the general idea, and we would talk more detailed design and analysis in the later sections.

Firstly, in order to extract text features from panoramic images, we design a feature engine

to automatically deblur images, detect text, recognize text, and return text feature vectors. The engine will keep extracting text and send back data nearly in real-time while it receives the images from cameras.

Secondly, to verify the location and GPS, we try to build a text reference dataset. In this thesis, we propose two ways to accomplish it. One is to use Google API to get Google Street View (GSV) and extract text features. Another is to use Azure API to get points of interest (POIs) and aggregate them as text features. We would talk more details later.

Thirdly, after feature extraction, we design a decider module to compare the textual information from the the physical world with the reference dataset. This module could have multiple choices to verify if GPS is spoofed or not. We examine them and perform a solid analysis in two cities dataset. At last, we are going to use the best model for location verification which is LSTM in the decider module.

Chapter 3

System Design

We start by introducing our design of the whole system. We assume that we have a start time for the system because we have to use panoramic images at previous data points. It depends on how many previous points we use. In this thesis, we use 50 points for each location verification.

Figure 3.1 shows the infrastructure of our system. It mainly contains two layers: image processing, and spoofing verification.

3.1 Images Processing

There would be two sources for data input: cameras (physical world) and online virtual maps. In the analysis, we use an existing panoramic image dataset to mimic real cameras' input. In this layer, we design several modules to process a sequence of panoramic images. We aim to collect textual information from each panorama image. To extract the accurate textual information, we design the image deblurring module and Scene-OCR module.

3.1.1 Image Deblurring

As we know, the motion would render images blurry which would lower the accuracy when we perform the OCR on them. Thus, we deploy the DeblurGAN [13] to obtain clearer images

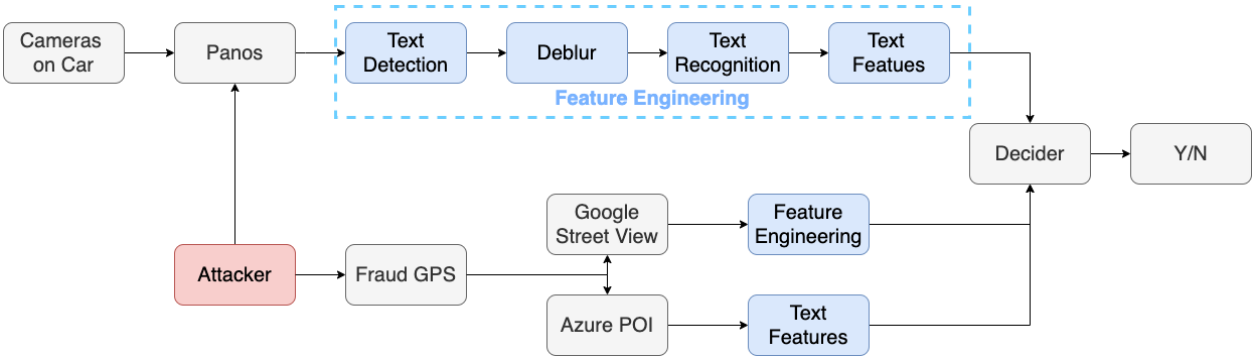


Figure 3.1: System Infrastructure

for text recognition. Figure 3.2 shows the example of text boxes before deblurring and after deblurring. We can see that the text in images becomes sharper, which is more suitable for text recognition.

Figure 3.2: Example of text deblurring

3.1.2 Scene-OCR

Intuitively, we use OCR to extract textual information from images. The most well known OCR tool is Tesseract Open Source OCR Engine[1] developed by Google. However, it has very poor performance in the real-world panoramic images. Thus, we try to build an OCR engine that can have good performance in the wild. We decide to combine the state of the art and the most popular methods for text detection and text recognition. We call this OCR engine as Scene-OCR. This module consists of two sub-modules: text detection and text recognition. Character Region Awareness for Text Detection (CRAFT) [3] is deployed for text detection. It will produce text boxes with corresponding positions from a panoramic image. The text recognition is realized using the combined model "TPS-ResNet-BiLSTM-Attn" [2]. They both have very good performance in the real-world scenery. Table 3.1 shows the performance comparison for Tesseract OCR and our Scene-OCR. To further illustrate that our Scene-OCR module performs much better than Tesseract OCR, we randomly select 1000 text boxes we collect from panoramic images and feed them into our Scene-OCR and Tesseract OCR respectively. We find that our Scene-OCR can recognize more than 70% of all text boxes with at least 0.6 confidence. On the other hand, Tesseract OCR only can recognize 18% of all text boxes and get some text from them. In addition, we look into 10 text results from Tesseract OCR and find that more than half of them are unreadable or not accurate textual information in corresponding text boxes. Thus, we prove that our Scene-OCR has much better performance than Tesseract OCR in the wild. It is important for our system to have accurate textual information.

[1] https://github.com/tesseract-ocr/tesseract

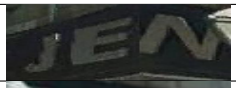
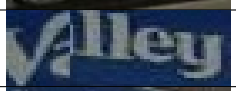
Text boxes	Sence-OCR (confidence score)	Tesseract OCR
	jen (0.92)	–
	lly (0.15)	–
	exchange (0.99)	–
	ges (0.98)	–
	alley (0.64)	i/jiley
	nationalbank (0.63)	–

Table 3.1: OCR results from text boxes

3.2 Spoofing Verification

In this layer, our decider module would have two inputs. One is textual information from cameras' images and another one is textual information from the reference dataset. The decider module is designed for comparing the current textual information from the physical world with textual information from the reference dataset and deciding whether the car is under the GPS spoofing attack or not. The internal mechanism of the decider module is a binary clarification where 0 indicates "non-attacked" and 1 indicates "attacked".

Chapter 4

Data Collection

4.1 Camera Panoramic Images

To mimic the real driving environment, our experiment relies on panoramic images from Streetlearn and the street view graph inside it. Streetlearn dataset [17] contains 56k street panoramas in Manhattan and 58k in Pittsburgh each having a resolution of 1632×408 pixels. The street view graph consists of panoramic images that are all linked to their adjacent points' street view. This makes it more convenient for us to simulate driving routes in two cities. Streetlearn dataset is serialized by protocol buffer [1]. We firstly deserialize all data and extract panoramic image IDs, the bytes of images, location coordinates, heading angles, pitch angles, and roll angles. These information will also be used for collecting a reference dataset. To avoid the misleading of text on vehicles, we also cut off 1/4 at the bottom of panoramic images.

4.2 Reference Dataset

In addition to the camera panoramic images, we still need a reference dataset (standard reference) to determine whether we are currently being attacked. For example, if we compare two panoramic images with the same coordinates and they are mostly the same, it means that

[1] https://developers.google.com/protocol-buffers/

our GPS coordinates are correct. Otherwise, the GPS is under a spoofing attack. Instead of comparing two images, we only use the textual information. In this way, we have to build our own reference dataset. Considering that we are focusing on textual information, we look into two ways to get real-time data sources: Google street view API [2] for panoramic images and Azure Map API for point of interests (POI) [3].

4.2.1 Google Street View API

We developed a tool to get panoramic images for target location coordinates. Unfortunately, Google Street View API does not allow us to directly derive 360° panoramic images. We request 3 times for each location coordinate and horizontally merge three 120° panoramic images into a 360° panoramic image. Those panoramic images are stored locally and prepared for subsequent processing.

4.2.2 Azure Map API

We choose Azure Map "search nearby" API to collect POI for all location coordinates. We mainly look for top 5 POI' names within 50 meters of the target point. The API would return each POI's information and we store their names in a string with the corresponding coordinates.

[2] https://developers.google.com/maps/documentation/streetview/intro
[3] https://docs.microsoft.com/en-us/rest/api/maps/search/getsearchnearby

Chapter 5

Experiment Design and Evaluation

We are aiming to build a text-based anti–GPS spoofing system. In this section, we introduce the way we design the system and how to evaluate its performance against different spoofing strategies. We start by introducing three main steps of our experiment: text extraction, text feature engineering, and location verification.

5.1 Data Preparation

5.1.1 Text Extraction

One of the most important things in our experiment is text extraction from panoramic images. We use state of the art techniques like image deblurring, text detection, and text recognition. They are playing a great role in our system to get more accurate textual information from panoramic images. The input image would firstly go through text detection. The results are multiple text boxes in the image. Then, we deblur images contained in these text boxes. This processing would make text boxes images sharper so that text recognition would be more accurate. After the image deblurring, we send all text boxes into the text recognition module. For each text box, we would have a result consisting of a confidence score, position coordinates, and a text string.

In order to set up a threshold for filtering unreadable text boxes, we randomly pick a

panoramic image and look into its text boxes. Figure 5.1 shows the example where an original panoramic image results in text information. Considering that text information is still nearly accurate when the confidence score is large than 0.6, we set up the threshold as 0.6 for filtering unreadable text.

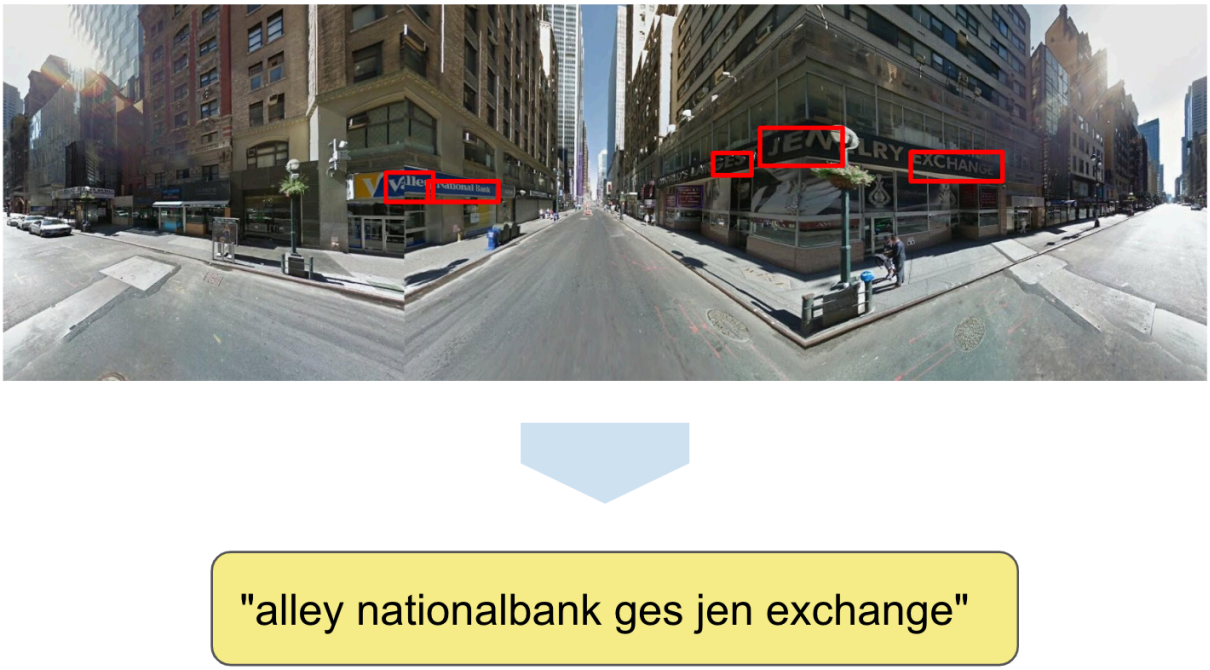


Figure 5.1: Example of text extraction

5.1.2 Text Feature Engineering

When we collect text from a panoramic image, we intuitively know that closer text boxes would be more relevant. Thus, we use Spectral Clustering to group text boxes and merge these text strings into a sentence. Given the sentence, we can have a vector with a size of 1×765 from the sentence embedding with BERT and XLNet [26]. These vectors would represent the points in the physical world and would be compared with the reference dataset

in the decider module. Additionally, we transfer the position information of text boxes to the vector. As shown in Figure 5.2, we divide a panoramic image into 6 parts. In each part, we examine the center position of text boxes and count the number of text boxes in each part. These numbers would construct a 1x6 vector, representing the position information of text boxes.

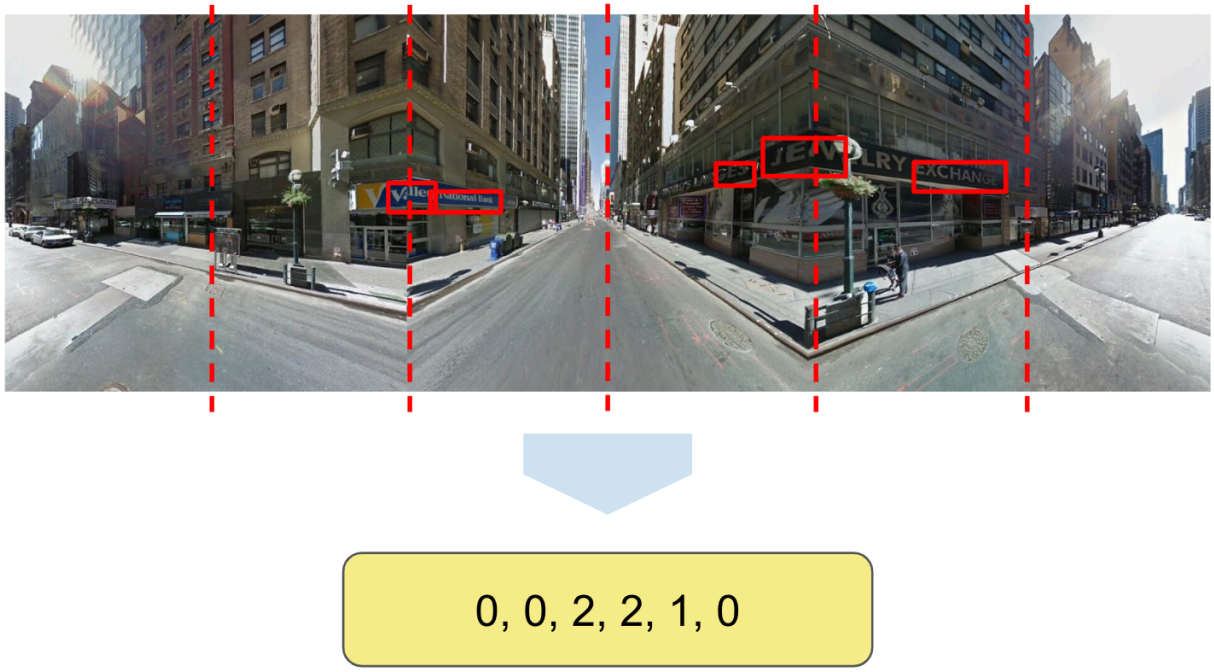


Figure 5.2: Text position information

5.2 Detect GPS Spoofing

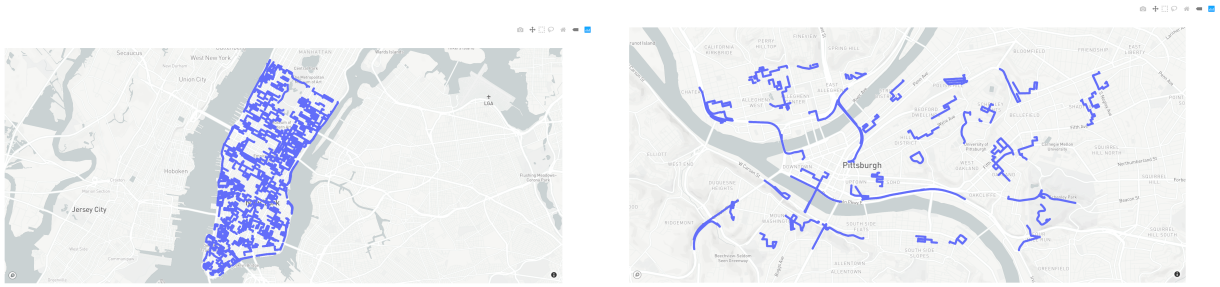
The design for the decider module in our system is that it can notice the difference between two inputs so that it is able to judge whether we are under GPS spoofing or not. Two inputs are text feature vectors from the physical world and reference dataset. It is a binary classification and we are going to use textual information at previous data points on the

trajectory. To handle this kind of sequence data, we decide to use the LSTM to build a binary classifier. In our system, for each current test point, we use 50 consecutive points' text features to construct a sequence data. We label the point as attacked or non-attacked based on whether we give it a fraud GPS coordinate. We use labeled training dataset to train such an LSTM. Then, we use the trained LSTM binary classifier to predict whether we are under attack or not. To avoid the information leak, the training dataset and testing dataset would be sampled from two cities. In our experiment, we firstly use the Manhattan dataset as a training dataset and Pittsburgh dataset as a testing dataset.

5.3 Performance Evaluation

5.3.1 Detection for Single-point Attacks

First, we conduct the experiment on single points that are attacked with random location coordinates. We will see if it is possible to detect the spoofing attacks when we only examine the panoramic image of the current place. We perform text fuzzy matching on the textual informal from camera input and the reference dataset. The reference dataset is GSV and the threshold for similarities is 0.8. If there are no textual informal in both side, we think it is a failure of detection. We test 500 data points, where half are attacked and another half are not. The detection accuracy is almost 0.2. Besides, we can only have 0.4 detection accuracy even if we set up the threshold as 0.5. The performance is poor as the result shows and can hardly be improved. The reason is that there is actually no textual information in most of the panoramic images. Thus, we need to use textual information at previous data points on the trajectory for the spoofing detection at a single point.



(a) Driving routes in Manhattan area (b) Driving routes in Pittsburgh area

Figure 5.3: Driving routes for training and testing

5.3.2 Continuous Attack Along the Driving Route

Next, we mainly test the scenario that the car is subjected to continuous GPS spoofing during travel. We will see whether the system has good capability to detect GPS spoofing attacks. In the beginning, we randomly selected 200 routes in the Manhattan area for generating training data and 40 routes in the Pittsburgh area for generating testing data. Figure 5.3 shows these driving routes simulation in the Manhattan and Pittsburgh area. We can see that these routes in Figure 5.3a basically cover the Manhattan area evenly and routes in Figure 5.3b are also distributed evenly in the Pittsburgh area.

The average length of routes is about 1km. We use 100 routes in the Manhattan area as cameras' input and another 100 routes' GPS coordinates as a spoofing attack. To prevent the classifier from only learning the difference between two places, we use the same driving routes for both labels to generate the training dataset. There are 99 points with panoramic images on each road. At each point, we have a concatenated vector of the camera data vector and the reference data vector which are derived from Section 5.1.2. Then, we set a window with a length of 50 points. Starting from the 50th point, we construct a data point which consists of the current point and previous 49 points. In this case, by sliding the window, we generated 50 data points on each route.

There are two options for the reference data: Google Street View (GSV) and Azure Map POI. We test both of them.

GSV for Reference

The size of the data at each point on the route is 1×1548. Therefore, the size of each data point in the training dataset is 50×1548. Also, each data point would have the corresponding label with a 0 or 1 indicating whether it has been attacked. Since we have a total of 100 routes used for generating attacked and non-attacked training data, there are a total of 10000 data points for the training data. We feed the training dataset into the LSTM to train the binary classifier in the decider. We have 40 driving routes in the Pittsburgh dataset. In the same way as the generation of training data points, we can get 2000 data points for testing. Through our trained model, we get a test accuracy of 0.9385. This shows that our model has a promising performance for the general attack strategy when we use GSV as reference data.

POI for Reference

The size of the data for each point on the route is 1 × 1542 because there is no position vector of text boxes in the POI information. Therefore, the size of each data point is 50 × 1542. Also, each data point would have the corresponding label with a 0 or 1 indicating whether it has been attacked. Similarly, we feed these 5000 data points into the LSTM to train the binary classifier in the decider. For the 2000 sets of testing data generated on the Pittsburgh dataset, our test accuracy is only 0.4875. The performance is very poor and unstable.

We also exchange cities, Pittsburgh dataset for training, and Manhattan dataset for testing.

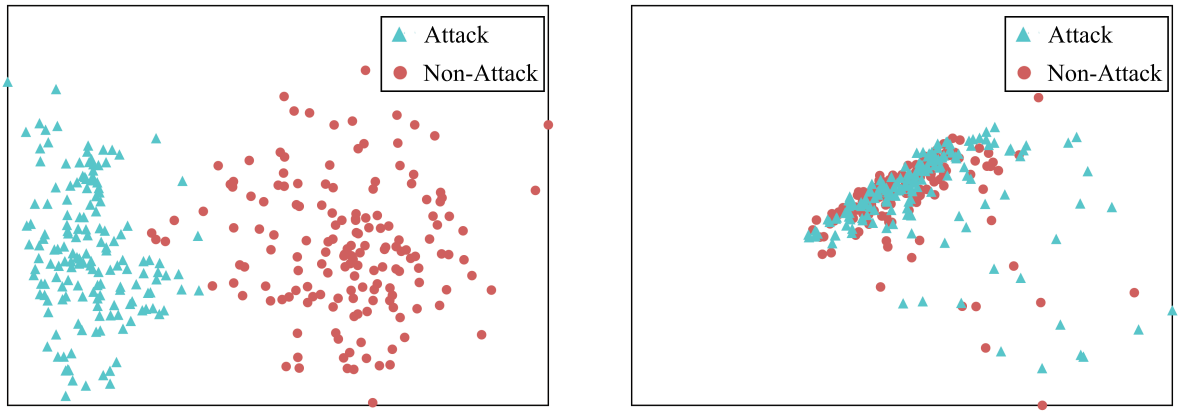
Then, we perform the same examination and get similar results. Our system achieves 0.9433 test accuracy on the GSV reference dataset and 0.5125 test accuracy on the POI reference dataset.

Considering that POIs are including all textual information around the point, there may be much more repeated and indistinguishable situations, especially when the GPS offset of the attack is not significant. Thus, when we choose POI as the reference data, the detection accuracy would be lower. In order to verify the label distinguishability of the training data, we use hypertools [1] to display our training data which has a high dimension in a 2-D figure. As shown in Figure 5.4, the left one is training data with GSV and the right one is training data with POI. The data overlap of different labels is very high when we use POI as the reference data. Furthermore, we do some case studies for failed detection with POI reference. We manually look into 10 points in different routes. We find that there are more noises in POI textual information. POIs in all 10 points contain the textual information from the physical world. However, they have super low similarities because the key textual information only occupies a little space in such a string from POI. 9 points have lower than 0.2 similarity between textual information from the physical world and POI. Also, there is 1 point in which we can not see any textual information from its panoramic image. But it still has 5 POI around it so we apparently have the wrong textual information for such a place. These cases also indicate that the current POI reference has issues to achieve a robust and effective performance for continuous GPS spoofing attacks.

5.3.3 Comparing Vector Similarity

Finally, in order to verify that our LSTM model has not been overfitted, or whether there can be a simpler model. We test when we only calculate the similarity between vectors from

[1] https://github.com/ContextLab/hypertools



(a) GSV for reference (b) POI for reference

Figure 5.4: Training data in 2-D

the physical world and reference data. In this way, we set up a threshold to distinguish attacked points and non-attacked points. As we know that each place has 50 vectors with a size of 1×765 both from the physical world and reference data. In order to calculate the similarity, we flatten these 50×765 matrices into 1×38250 vectors and calculate similarities for each pair. The result is shown in Figure 5.5. It means that if we set the bar as 0.5 where all attacked points would be detected, we will have almost 80% non-attacked points be marked as attacked points. The best bar seems to be 0.4 where 80% attacked points would be detected, but there are still 40% non-attacked points would be classified incorrectly. The performance of this method is apparently worse than our LSTM binary classifier and does not work for general spoofing attacks.

5.3.4 Partial Attacks Along the Driving Route

Considering attackers might use smarter ways to spoof GPS signals, we also control the attack frequency and test the performance of our system in this attack strategy. Along the driving routes, there are partial points under the GPS spoofing attack. We examine the

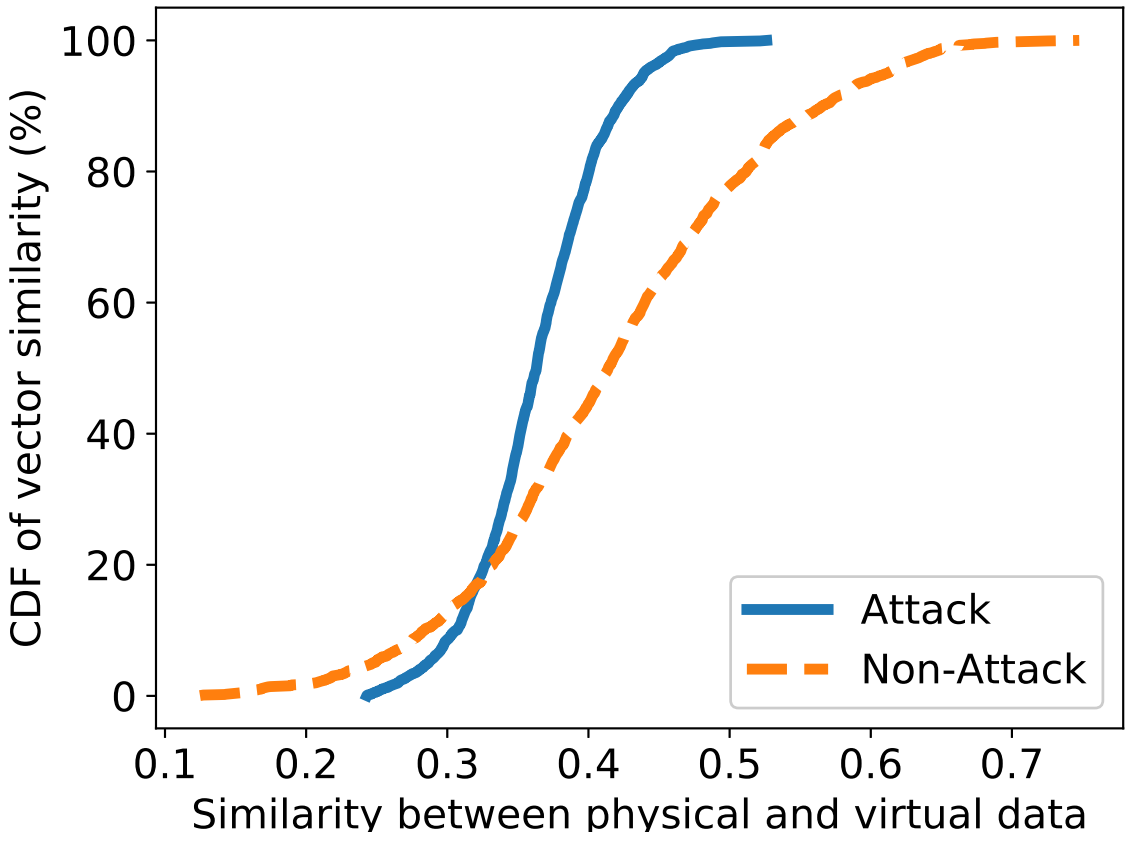


Figure 5.5: CDF of vectors similarity

performance of our system when we have different ratios of attacked points in a route, from 10% to 100%. We randomly select these points in a route and give them incorrect location coordinates. Then, we use these routes as test dataset to validate if our system could still detect the GPS spoofing attack. As shown in Figure 5.6, our system can still achieve promising performance for this spoofing attack strategy when we use GSV for reference, even under very low attack frequency. When there are about 20% attack points in driving routes, the detection accuracy can still achieve more than 80%. Also, if there are more than 50% attack points in driving routes, our system can detect the GPS spoofing attacks with nearly 98% accuracy. Unfortunately, our system has not good performance when we use POI

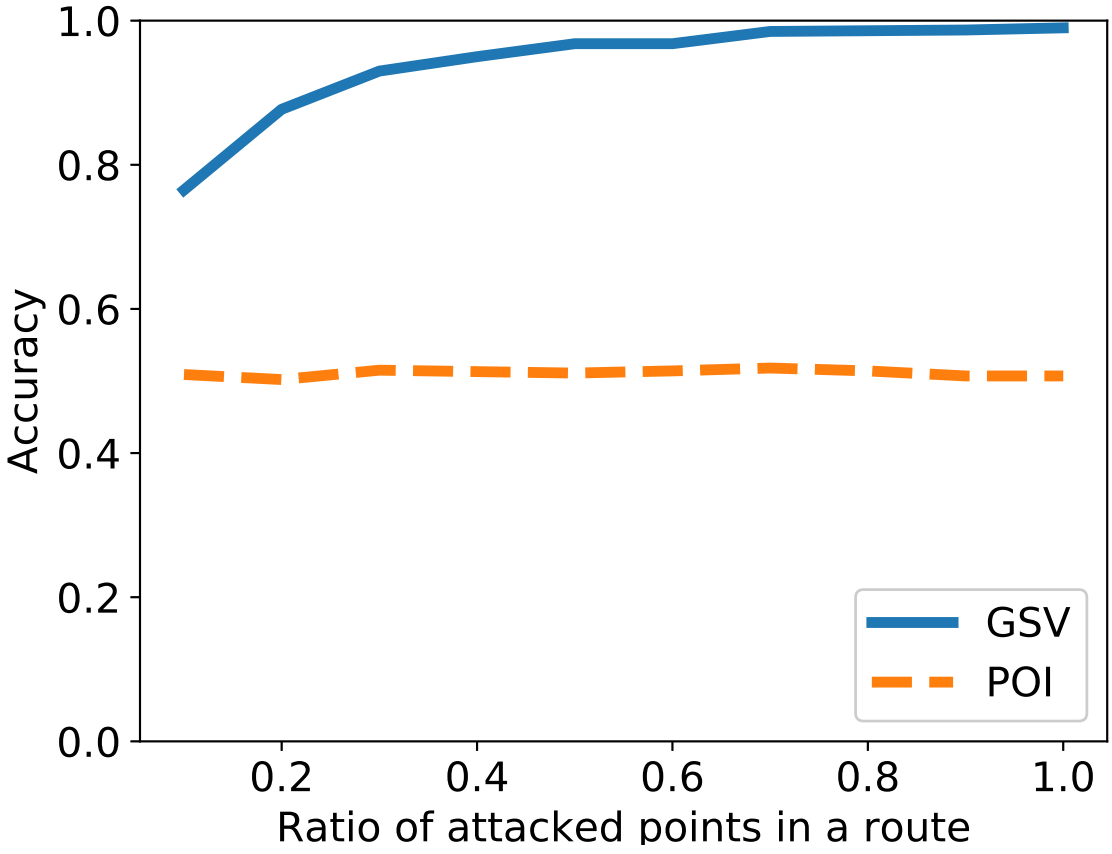


Figure 5.6: Detection accuracy versus ratio of attacked points

for reference. For all ratios of attacked points in a route, it has only about 50% detection accuracy. Comparing the performance of two reference datasets, we decide to use textual information from real-time GSV because of much more solid performance.

Chapter 6

Discussion

6.1 Limitations

There are a few limitations in our work. First, our reference dataset is textual information from real-time GSV, where indicates that we have to get all GSV and extract textual information from them. It is a extremely heavy work when we build a reference dataset for a new city. However, this is a one-time job and can be shared with all users. Also, to keep the detection accuracy of the system, the reference dataset needs to be updated frequently because the environment beside routes would be changing fast. Second, our analysis mainly focuses on routes in cities where have more textual information along the routes. We need to examine some more strategies to deal with situations like the countryside and highways. Third, we do not know if we are currently under GPS Spoofing at a single point because we use past panoramic images from past points. However, we argue that our system can detect the GPS spoofing once it happens in the routes. We assume it is sufficient for noticing victims. Fourth, even we have already designed multiple modules to improve the ability to extract textual information from the physical world, we still miss many textual information during the processing. This limitation largely depends on the development of techniques for text detection and recognition. We have proved the feasibility to use textual information for detection of GPS spoofing attacks and displayed a promising result. We hope that there would be more ways for us to improve the ability to get more accurate and sufficient textual

information from images.

6.2 Future Work

Based on our system design, there are still some open questions for text-based localization verification. First, instead of using 360° panoramic images, we can only use 120° or even 90° images. It is possible to use the phone to record the video in the car and then it can report the detection result. Second, the real-world testing would be interesting and we can drive the car in multiple cities with real GPS spoofing attacks to examine our system. Third, to further reduce the cost of the deployment, we can build part of our system on the cloud and allow users to upload processed vectors and GPS location credentials to see if they are under attack or not. For security concerns, the image processing part would still be on the car and only encrypted data vectors will be uploaded to the cloud services. Fourth, our system can only detect whether we are under the GPS spoofing or not. A more challenging work is to use textual information to localize the car. It should be able to estimate a location coordinate in real-time and notice drivers if it is different from the current GPS coordinate. Then, it even could take over the GPS equipment and do the job of navigation.

Chapter 7

Conclusion

In this thesis, we explore the feasibility of a real- time text-based Anti–GPS spoofing system. Analysis and driving simulation tests in two cities all confirm that our system can achieve a promising performance with GSV reference dataset for general GPS spoofing attack strategies. We discuss several strategies for the text-based location verification and prove that LSTM as a binary classifier performs best among them. We hope that the system design and proposal can motivate practical mechanisms to protect the massive GPS users and GPS-enabled autonomous systems.

Bibliography

[1] Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., and Sivic, J. Netvlad: Cnn architecture for weakly supervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[2] Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., Oh, S. J., and Lee, H. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *International Conference on Computer Vision (ICCV)* (2019).

[3] Baek, Y., Lee, B., Han, D., Yun, S., and Lee, H. Character region awareness for text detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).

[4] Brands, S., and Chaum, D. Distance-bounding protocols. In *Workshop on the Theory and Application of of Cryptographic Techniques* (1993).

[5] Brubaker, M. A., Geiger, A., and Urtasun, R. Lost! leveraging the crowd for probabilistic visual self-localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013), pp. 3057–3064.

[6] Gordo, A., Almazan, J., Revaud, J., and Larlus, D. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision* (2017), 237–254.

[7] Hoang, T., Do, T.-T., Le Tan, D.-K., and Cheung, N.-M. Selective deep convolutional features for image retrieval. In *25th ACM International Conference on Multimedia* (2017).

[8] Hong, Z., Petillot, Y., Lane, D., Miao, Y., and Wang, S. Textplace: Visual place recognition and topological localization through reading scene texts. In *IEEE International Conference on Computer Vision (ICCV)* (2019).

[9] Hu, L., and Evans, D. Using directional antennas to prevent wormhole attacks. In *NDSS* (2004).

[10] Jansen, K., Schäfer, M., Moser, D., Lenders, V., Pöpper, C., and Schmitt, J. Crowd-gps-sec: Leveraging crowdsourcing to detect and localize gps spoofing attacks. In *IEEE Symposium on Security and Privacy (SP)* (2018).

[11] Jégou, H., Douze, M., Schmid, C., and Pérez, P. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2010).

[12] Kuhn, M. G. An asymmetric security mechanism for navigation signals. In *International Workshop on Information Hiding* (2004).

[13] Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., and Matas, J. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).

[14] Lazos, L., Poovendran, R., and Capkun, S. Rope: robust position estimation in wireless sensor networks. In *International Symposium on Information Processing in Sensor Networks (IPSN)* (2005).

[15] Lowe, D. G. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision (ICCV)* (1999).

[16] Milford, M. J., and Wyeth, G. F. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *IEEE International Conference on Robotics and Automation (ICRA)* (2012).

[17] Mirowski, P., Banki-Horvath, A., Anderson, K., Teplyashin, D., Hermann, K. M., Malinowski, M., Grimes, M. K., Simonyan, K., Kavukcuoglu, K., Zisserman, A., et al. The streetlearn environment and dataset. *arXiv preprint arXiv:1903.01292* (2019).

[18] Montgomery, P. Y. Receiver-autonomous spoofing detection: Experimental results of a multi-antenna receiver defense against a portable civil gps spoofer. In *Radionavigation Laboratory Conference Proceedings* (2011).

[19] Moser, D., Leu, P., Lenders, V., Ranganathan, A., Ricciato, F., and Capkun, S. Investigation of multi-device location spoofing attacks on air traffic control and possible countermeasures. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking* (2016), pp. 375–386.

[20] Naseer, T., Burgard, W., and Stachniss, C. Robust visual localization across seasons. *IEEE Transactions on Robotics* (2018), 289–302.

[21] Naseer, T., Spinello, L., Burgard, W., and Stachniss, C. Robust visual robot localization across seasons using network flows. In *Twenty-eighth AAAI Conference on Artificial Intelligence* (2014).

[22] Nielsen, J., Broumandan, A., and Lachapelle, G. Gnss spoofing detection for single antenna handheld receivers. *Navigation 58*, 4 (2011), 335–344.

[23] Nistér, D., Naroditsky, O., and Bergen, J. Visual odometry. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (2004).

[24] Radwan, N., Tipaldi, G. D., Spinello, L., and Burgard, W. Do you see the bakery? leveraging geo-referenced texts for global localization in public maps. In *IEEE International Conference on Robotics and Automation (ICRA)* (2016).

[25] Rasmussen, K. B., and Capkun, S. Realization of rf distance bounding. In *USENIX Security Symposium* (2010).

[26] Reimers, N., and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing* (11 2019), Association for Computational Linguistics.

[27] Schäfer, M., Lenders, V., and Schmitt, J. Secure track verification. In *IEEE Symposium on Security and Privacy* (2015).

[28] Tippenhauer, N. O., Pöpper, C., Rasmussen, K. B., and Capkun, S. On the requirements for successful gps spoofing attacks. In *ACM Conference on Computer and Communications Security* (2011).

[29] Wang, H.-C., Finn, C., Paull, L., Kaess, M., Rosenholtz, R., Teller, S., and Leonard, J. Bridging text spotting and slam with junction features. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2015).

[30] Wesson, K., Rothlisberger, M., and Humphreys, T. Practical cryptographic civil gps signal authentication. *NAVIGATION: Journal of the Institute of Navigation 59*, 3 (2012), 177–193.

[31] Zamir, A. R., and Shah, M. Accurate image localization based on google maps street view. In *European Conference on Computer Vision (ECCV)* (2010).

[32] Zeng, K. C., Liu, S., Shu, Y., Wang, D., Li, H., Dou, Y., Wang, G., and Yang, Y. All your gps are belong to us: Towards stealthy manipulation of road navigation systems. In *USENIX Security Symposium (USENIX Security)* (2018).

[33] Zhang, Z., Trinkle, M., Qian, L., and Li, H. Quickest detection of gps spoofing attack. In *IEEE Military Communications Conference (MILCOM)* (2012).