LOCATION-AWARE MOBILE SOCIAL NETWORKING FOR VIDEO SHARING

by

Li Ma
B.Eng, Zhejiang University, 2008

a Thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science
in the
School of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Summer 2011

APPROVAL

Name: Li Ma

Degree: Master of Science

Title of Thesis: Location-Aware Mobile Social Networking for Video Sharing

Examining Committee: Dr. Lou Hafer
Chair

Dr. Jiangchuan Liu, Senior Supervisor

Dr. Qianping Gu, Supervisor

Dr. Jie Liang, SFU Examiner

Date Approved: 22 August 2011

Abstract

Location information improves the routing effectiveness and facilitates the development of diverse novel applications in mobile networking. Although more people enjoy Internet and multimedia via their cell phones, most of the services for online video sharing are still restricted to traditional PC users. The existing systems of mobile video sharing have yet to effectively explore mobility decently or address the energy and bandwidth limitations of mobile devices. Furthermore, while such location information can lead to better user experiences, given privacy concerns and hardware constraints, a mobile user often exposes a limited number of locations only. We are thus interested in the *Trajectory Exposure Problem* in this context, i.e., to what degree that the user's trajectory (i.e., its route) is exposed? Furthermore, can the user adaptively control the exposure of its trajectory and yet offer useful information for location-based services?

In this thesis, we first propose MoViShare (Mobile Video Share), a universal video sharing platform that will provide anytime anywhere video browsing and publishing services for mobile devices. MoViShare targets to create and maintain location-aware mobile social networks, and to apply video abstraction technique for saving bandwidth and energy. We extend it with another project Scoop as the enhancement for professional uses.

We then explore Gaussian Process Regression, an effective tool to re-construct the trajectory of the mobile user with selected exposed locations. We examine how the re-constructed trajectory differs from the real trajectory, i.e., evaluating the *exposure rate*. We present an effective heuristic that adaptively controls the trajectory exposure rate by carefully choosing the exposed locations. We further demonstrate a practical routing protocol, MoRPTE, which, controlled by a single parameter, utilizes location information flexibly and adaptively in the spectrum from zero knowledge to full knowledge to fit the applications' demands.

Acknowledgments

First and foremost I offer my sincerest gratitude to my senior supervisor, Dr. Jiangchuan Liu, who has guided and supported me throughout my thesis with his patience and knowledge. This thesis would not have been completed or written without his encouragement and effort.

It is an honor for me to also thank my supervisor Dr. Qianping Gu and my thesis examiner Dr. Jie Liang, for reviewing this thesis and providing suggestions to improve the quality of my thesis. Thank Dr. Lou Hafer for taking the time to chair my thesis defence. Meanwhile, I would also like to extend my appreciation to the faculty and staff in the School of Computing Science of Simon Fraser University. It is really a pleasure to thank all who made this thesis possible.

I am indebted to many of my colleagues and friends to support me throughout my master study. Especially thank Zonggao Jia, who conducted the project MoViShare with me together. And yes, it is indeed such a nice experience working with you for the project. Thanks a lot for your greatest help and suggestions. Also, I would like to thank Feng Wang, Xu Cheng and Haiyang Wang, for their priceless suggestions and professional support on my research.

And finally, I thank my parents for their support and encouragement in completing this accomplishment.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

With the great penetration of GPS and other tracking services (e.g., through WiFi or cellular based stations), the geographical location is now readily available in numerous mobile devices. Such information improves the routing effectiveness and facilitates the development of diverse novel applications and social networks, making mobile users better interact with their ambient environment [47][40]. Many of their properties and issues have been analyzed through both theoretical and experimental studies; however, one aspect yet to be fully explored is the trajectory exposure and recovery of mobile users.

1.1 Location-aware Mobile Social Networks for Multimedia Sharing

Recent seven years have witnessed an explosion of networked video sharing, e.g., YouTube [14], as a new killer Internet application. Their great achievements lie in the combination of the content-rich videos and, equally or even more importantly, the establishment of social networks. The videos are no longer stand-alone units but are hyper-linked to each other based on their owners' social relationships, which has greatly motivated individuals' participation. It is interesting to point out that the clients of video sharing sites and that of advanced mobile devices are largely overlapped, with a majority of them belonging to the young generation. Unfortunately, most of the video sharing services are generally customized to traditional Internet users, even though more and more mobile devices are multimedia-ready. According to a recent report from Veeker Ltd. [13], only about 50 percent of U.S.

mobile phone subscribers with built-in cameras actually snap photos, not to mention videos. Only recently have we seen preliminary attempts that explore mobile video sharing, e.g., by YouTube, Veeker, and iTouch [5].

Comparing to PCs, mobile devices have weaker processors and smaller screen sizes. Most of them are also suffered from battery-energy limitations. Wireless mobile networks have much narrower bandwidth as compared to the broad-band Internet. Therefore, simply launching a mirror of the wired video sharing site and opening existing sources to mobile users can hardly offer satisfying services quality.

Mobile-specific social networks like airG [1] and MocoSpace [8] have started attracting clients worldwide. These mobile social networking services emphasize the activities people tend to do on their mobile devices, such as locating friends via GPS, and using camera phones and cellular data networks to get data. Unfortunately, these systems have so far focused on mobile-based communications or simple user-based social networks. Most of them have very limited location-wise considerations when building the social networks, not to mention combining them with video sharing.

To fill the gaps, we design and implement MoViShare (Mobile Video Share), a universal video sharing platform that accommodates mobile accesses. MoViShare is targeting a seamless combination of social networking and multimedia sharing for wireless mobile clients. It creates and maintains location-aware social networks among the clients. It will explore their mobility by effectively utilizing the location information and activating context-sensitive location-aware video browsing and sharing. The incorporation of mobile clients also enriches the traditional video sharing sites, given that the video contents can now be captured and published anywhere anytime. In addition, to address the issue of bandwidth and energy limitations of the mobile devices, MoViShare generates a video abstraction for every video stream. The abstractions, containing a series of key frames, are representative and intuitive, yet their lengths are only 1% to 5% of the original video clips. In mobile environment, these unique features will greatly benefit clients when using MoViShare system [42].

Furthermore, in order to meet the increasing requirements of technical professional uses, we design and implement a system called Scoop, a fast, simple and mobile social communication platform for creating comments, and tracking issues, problems and pursuits as an enhancement of workforce.

1.2 Trajectory Exposure Problem

While these new location-aware applications and routing designs can lead to better user experience, the exposure of personal location nevertheless involves privacy concerns as well. The concern is particularly severe given that the broadcast nature of a wireless channel. Recent survey [7] shows that 22% of the adult users use location based services at least once a week. However, there're still around 78% simply disable the location services in their iPhone devices or yet do not use such services often, based on privacy and other concerns. In some more critical cases, for example battle fields, exposed location would even lead to disastrous consequences [39].

Fortunately, most of the common people do not have such a stringent requirements as in battle fields. To benefit the improved performance and convenience, they often do not mind a few location points to be divulged, as long as the their visited locations are not always exposed. In other words, the exposed data should not enable an accurate re-construction of the whole route. This is also true for commercial organizations where the daily routes are strategically valuable data, e.g., for Taxi companies. We have used the taxi location data from a major vendor in the city of Beijing in the performance evaluation of our work. The companies do not mind exposing some location information to improve service or vehicle-to-vehicle communications. Yet they do not expect the daily route of their taxis to be exposed, which would greatly benefit their competitor's strategic plans. In fact, most of their taxi drivers are not comfortable if their daily routes are fully exposed, either.

Besides privacy, other concerns might involve the consideration of battery energy and the low sample rates that GPS can provide. Thus a user cannot expose all its locations continuously in location-based services and routings. Both location-oblivious and location-based services have been extensively studied in the literature. Yet services in between these two extremes have been largely unexplored. Considering the above concerns, we thus ask the following two critical questions:

If a mobile user exposes a limited number of locations only, what is the exposure rate of its trajectory (i.e., its route) ?

How can we adaptively control the exposure of a mobile user's trajectory (i.e., its route) and yet offer useful information for location-based services?

In this thesis, we explore Gaussian Process Regression, an effective tool to re-construct the trajectory of the mobile user with selected exposed locations. We examine how the re-constructed trajectory differs from the real trajectory, i.e., evaluating the *exposure rate*. We present an effective heuristic that adaptively controls the trajectory exposure rate by carefully choosing the exposed locations. In state-of-the-art location-based routings, the location information is periodically updated and not real-time. We substitute it with predicted location which performs better than historical advertised location. We further demonstrate a practical routing protocol, MoRPTE, which, controlled by a single parameter, utilizes location information flexibly and adaptively in the spectrum from zero knowledge to full knowledge to fit the applications' demands. Our performance evaluation based on both the real Beijing Taxi data and synthetic data demonstrate the flexibility and efficiency of our solution. It also facilitates the understanding of the fundamental utility and impact of location information exposed at various degrees.

1.3 Organization of the Work

We introduce some background and related works to ours in Chapter 2, and then go into some details about the location-aware mobile network projects MoViShare and Scoop in Chapter 3. After studying our mobile sharing platforms, we then raise the trajectory exposure problem in Chapter 4, presenting our motivation and introduce our proposed method of evaluating and the adaptive control of exposure rates. We further demonstrate a practical routing protocol, MoRPTE, in Chapter 5, followed by the performance evaluation in Chapter 6 and eventually the conclusion in Chapter 7.

Chapter 2

Background and Related Works

Though there have been many recent research projects investigating specifically the content sharing in mobile social networks, most have focused on the mobile-based communications or simple user-based social networks. Such data transmissions require advanced routing mechanisms specified for mobile devices, in which users' geographical locations might be exposed and trajectory reconstructed. In this chapter we will present some background and related works in such areas.

2.1 Mobile Video Sharing and Mobile Social Networking

2.1.1 Mobile Video Sharing

In May 2006, YouTube started a service that allows users to upload homemade clips via their mobile phones or PDAs [16]. Later in same year YouTube Mobile site (http://m.youtube.com) has been launched to open its video repository to cell phone users. In the early stage there are only very small amount of video clips with special format can be watched via cell phone. In recent two years, YouTube collaborated with giants of the industry such as Nokia, Verizon, Apple, Vodafone, and extended the mobile video service to 16 other countries outside USA and 10 other languages than English [15]. For example, during the collaboration with Apple Inc., they offered a feature in Apple's products, iTouch and iPhone, that allows user to browse YouTube website through a single click on one icon.

Unfortunately there are still several fundamental issues to be addressed. The performance can be compromised due to the restricted hardware conditions of mobile devices.

Comparing to PCs, mobile devices have weaker processors and smaller screen sizes. Most cell phones are also suffered from power limitations when watching videos. Comparing to wired network, mobile phone network has averagely narrower bandwidth in terms of Internet data transferring. Therefore, simply launching a mirror of the original video sharing site and opening all sources to mobile users may not give satisfying services as expected. Mobile users may face difficulties to find a wanted video, or experience long buffering time and lagging performance when playing back. We believe that more efforts need to be made to overcome the above obstacles in mobile video sharing business.

2.1.2 Mobile Social Networking

There has been several mobile-specific social networks like AirG, MocoSpace and Nokia's Ovi attracting clients worldwide in recent several years. AirG's mobile community [2] has more than 20 million unique users worldwide and is interconnected to more than 100 mobile operators in over 40 countries. AirG's mobile community solutions are proven to increase customer affinity and brand recognition for leading mobile operators and media companies globally. MocoSpace is a social network specifically designed for use on a mobile device [9]. The features of the site are similar to other social networking sites including mobile chatting, instant messaging, photo and video sharing, as well as forums. MocoSpace is the largest mobile social network in the U.S. with over 6 million registered users and over one billion monthly page-views. Nokia's Ovi is marketing as "personal dashboard" where users can share photos with friends, buy music and access third-party services [10]. It has some significance in that Nokia is moving deeper into the world of Internet services. So far, Ovi has successfully pitched to European operators [33].

To catch up the trend, traditional Internet social network providers poured money into mobile services as well. Facebook Mobile, Yahoo! Mobile and MySpace Mobile are the most successful cases in this category. Google also launched a new service called "Latitude" which allowed mobile users to see where friends were and what they were up to. Taking advantages of their existing gigantic social networks, various web2.0 services and rich contents, they got double results with half efforts.

These mobile social networking services emphasize the activities people tend to do on their phones, such as chatting, instant messaging, friends locating, and multimedia source sharing. Those features create groups for consumers, and a new profit center for carriers. eMarketer forecasts that over 800 million people worldwide will be participating in a social network via their mobile phones by 2012, up from 82 million in 2007 [3].

2.2 Two Extremes in Routing Trends

2.2.1 Location-oblivious Routing

Traditional wireless routing protocols seldom utilize nodes' geographical information. Comparing with position based routings, they can be categorized into topology based routings. There are several sub-categories, while they can be further divided into proactive, reactive and hybrid approaches; and some of them specifically can be categorized as flow-oriented or power-aware approach, and etc.

For instance, wireless routing protocol WRP uses an enhanced version of the distance-vector routing protocol. It maintains an up-to-date view of the whole network. The complexity of maintenance of table updating requires nodes' large memory and great processing power. Suffering from scalability and mobility, it is not suitable for highly dynamic networks. Other similar protocols such as DSDV, Virtual Ring [24], TBRPF [23], OLSR [29], are proactive routings, or data driven routings which maintain lists of destination and routes by periodically updating routing tables. They keep records of available paths both currently used and not used. The disadvantages of such protocols are about maintenance and failures or reconstructions, especially when the topology changes frequently.

Other protocols, like SENCAST [20] which perform reactive routings figure out a route on demand by flooding the requesting packets. Then, like Virtual Ring [24], DSR [27] and Reclaim Caching [49], they maintain only routes that are currently in use. However, without location information, the main disadvantage is high latency in routes finding.

Location-oblivious routings do not suffer from privacy issues, but they miss the opportunity of taking advantage of location information and thus to decrease the routes finding latency or the data for maintenance, which can significantly accelerate data transmissions due to the high speed and low latency requirements nowadays.

2.2.2 Location-based Routing

Another trend in an extremely opposite way is geographical routing. This type of protocols acknowledges the influence of geographical distances as significant to network performance.

They fully utilize location information and distribution of nodes in the networks, such as Self-adaptive Ondemand Geographic Routing (SOGR) [47], Simple Forwarding over Trajectory (SiFT) [25], Partial Unit Delaunay Triangulation (PUDT) [40] , and [46]. They have been proposed to cope with the highly dynamic networks. Taking advantages of mobile devices' physical position information, routing efficiency has increased significantly.

Many routing protocols use greedy forwarding, like Most Forward with R (MFR) which always tries to deliver to the neighbor who is closest to destination. Compass Routing [37] establishes a straight line between sender and destination first, and select the neighbor closest to the line, which tries to minimize the spatial distance for a packet to be transmitted. Also, random selection for the next hop is also doable to minimize the accuracy geographical information required. Combined-Greedy-Face [50] also provide recovery approaches for greedy routing failure, based on planar graph traversal.

These routing protocols, in the spectrum with full knowledge of nodes geographical information, however have to deal with privacy issues. Privacy information abuse might lead to disasters, meanwhile various people might have personal demanding of privacy regarding to their own situation. Therefore, traditional routing which fully utilizing location information lacks of scalability and dynamics.

2.3 Privacy-aware Location Based Service (LBS) and Routing

In security fields, similar problems have been issued as privacy in Location Based Service (LBS) [45]. A simple way of defining LBS is a certain service which is provided to the users based on their own locations, such as location based traffic reports and location based store finder, and etc. Solutions such as Cloaking [48] and K-anonymity, have been proposed, in which the real location is hided by providing misleading information or non-sense data for those who overhear. They usually require a centralized server as trusted proxy to gather information, handle all calculations and the deliveries for cloaking nodes' identities and encrypting data.

There are also privacy-aware wireless routings proposed. PPBR [51] tries to avoid periodically broadcasting nodes' positions to neighbors without protection, by taking dynamic pseudo ID in advertising the nodes' positions. [22] proposed a method which is practical to identify an optimal k-anonymization of real census data when the optimal one cannot be found in time, as stated in [41]. NWA [17] proposed a revised K-anonymity method for trajectory privacy, by defining a cylinder which contains k moving objects. [32] utilized two K-anonymity techniques to protect privacy through the whole time of LBS service.

However, these approaches mostly focus on encryption and anonymity, but not for benefiting routings at the same time. And they can not dynamically accommodate diverse requirements of privacy by different people's demands.

2.4 Prediction and Modeling of Object movements

A variety of methods have been proposed to deal with the prediction and modeling of objects movements. Parametric approach can restrict the class of function by, for instance, polynomials. For example, to let $y = w_0x^0 + w_1x^1 + ... + w_nx^n$. However, it will be difficult in determining the parameters in order to be more suitable fitting the pattern. There are also user mobility models proposed, such as [30][41] [36]. It is also suggested in [31] that each user regularly visits a list of hubs with probability, or can be predicted on a map-based basis. However, such schemes rely on knowledge of maps and user profiles which might not be always available. Studies [34][21] revealed that Multi-layer Neural Network and Gaussian Process Regression are the best. However, the disadvantages of Neural Network will be stated in Chapter 6.

However, our focus is more on generally utilizing location information while preserving trajectory exposure dynamically. After comparing different methods among each other, we take Gaussian Process Regression as an example in our scenario to reconstruct trajectories given incomplete location information.

Chapter 3

Building Location-Aware Mobile Social Networks for Media Sharing

Given the fact that although a large population of video sharing clients and that of advanced mobile devices are overlapped, unfortunately, most of the video sharing services are still restricted to traditional Internet users, even though more and more mobile devices are multimedia-ready.

In order to provide a universal video sharing platform that accommodates mobile accesses, and to study its properties and related issues, we designed and implemented MoViShare, a location-aware mobile social network for multimedia sharing. This work, conducted by me and another student Zonggao Jia, was supported by a Nokia University Relation Initiative Grant. And the first stage of our prototype was successfully demonstrated in conference ACM MobiCom 2009 [42].

Later on, another project Scoop was conducted as an enhancement which offers efficient and simple access for professional mobile uses. This project was supported by a NSERC ENGAGE grant, with the collaboration of local company Trusterra.

3.1 Problem Statements

All current existing mobile video sharing systems have not fully utilized the biggest advantage of the mobile devices - the mobility. While nowadays any multimedia-ready cell phone can take pictures or capture videos anytime anywhere, thus the users intuitively would expect to share their information or data with others instantly. Instead of simply gathering video clips and streaming them to cell phones, it is better to develop a network that targets on building and maintaining a mobility aware social network among massive end users by effectively utilizing the location information. Therefore the users can instantly retrieve nearby photos and videos, make new friends during trips, and share live experiences with old friends. This kind of services is still lack at present.

Besides mobility issue, another problem is that the cell phones have limited battery power and restricted network bandwidth. These problems must be addressed because they will affect the overall usability of the whole system. To search for a video, no one is willing to spend a lot of time on waiting for loading a page with searching results. Also it is not reasonable to ask the users to watch all the candidate video clips in order to find the one they are looking for, because it is quite time and energy consuming to stream a video clip through cell phone [42].

Therefore, to fill the gaps, we developed a location-aware mobile social network named MoVishare that accommodates mobility issue; and came up with the solution of giving video abstraction of the available video clips. So that users can browse quickly and have a look at the abstraction first, then they can decide whether to watch the whole video or not [33]. Furthermore, we also developed Scoop platform as an enhancement for professional communications and sharings.

3.2 MoViShare Hights

3.2.1 Location-Aware Social Network

MoViShare will create and maintain location-aware social networks among mobile clients, as shown in Figure 3.1. Specifically, it enables context-sensitive location-aware video browsing and sharing. A client can search for videos based on their current positions or expected locations. Diverse social networking features can also be easily enabled in framework, such as locating a friend, share live experiences with existing friends, communicating with others in the same event or party, advertising visualized location information to others, and etc. This kind of services are unique in the mobile environment, and will conveniently facilitate clients to interact with their environment and other users.

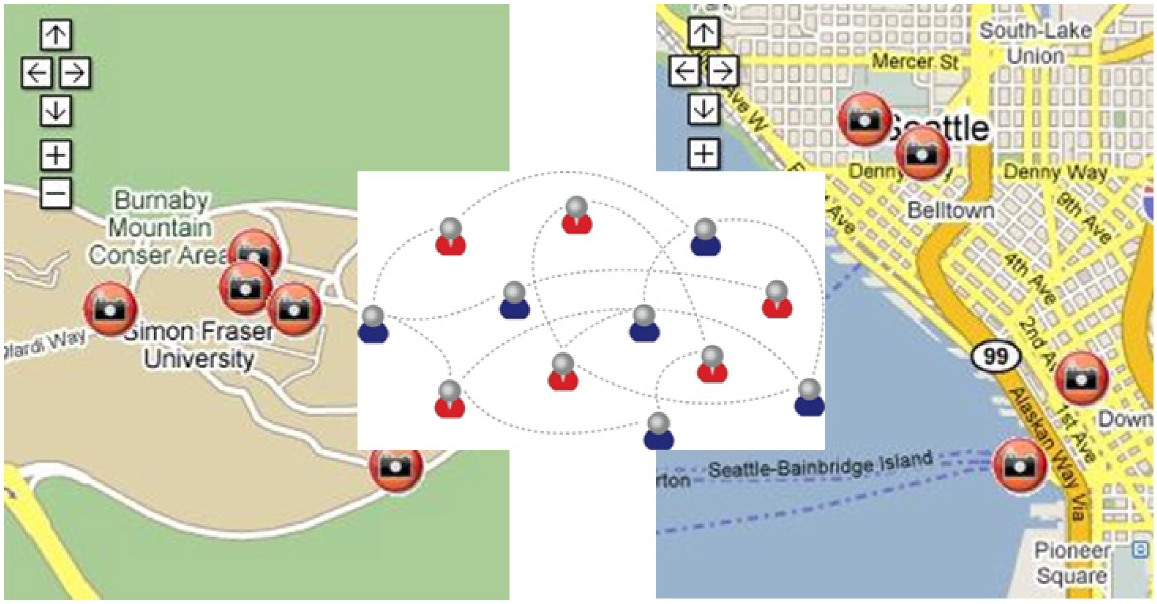

Figure 3.1: Location aware Social Network

3.2.2 Video Abstraction

Limited battery power and restricted network bandwidth remain challenges in wireless mobile networks. Thus it is not reasonable for a mobile client to download and then browse all the video clips (or even the initial parts) in order to find the right one.

MoViShare addresses the challenges through *video abstraction*. By displaying a series of key frames, it facilitates a user to immediately obtain a global impression of the whole video, such as the mainly content, place of taken, and protagonists of the video. Figure 3.2 displays a sample video abstraction, in which we can easily figure out the main content. The length of abstraction files are normally only 1% to 5% of original video clips. Therefore, fast browsing and searching features on mobile devices can be activated. With additional information provided by the system, such as the length, view times, rating and comments, a user can make a well-informed decision of whether to watch the whole video. Moreover, the video abstractions can be used by the notification system to promote video clips to selected user groups via emails or MMS messages.

There are two optional running modes for video abstraction process, at the phone and at the server. The task can be done by mobile devices themselves before uploading the videos.

Figure 3.2: An example of video abstraction

In such way there are no extra server workloads required and users have more control of the abstraction. However, there are still some problems we cannot afford to overlook in implementing this mode, such as issues with processor, memory, storage and file format, etc [42].

It is almost impossible to implement a universal application which works for all modern mobile devices. And further more, cell phones usually have no processors powerful enough to run even the simplest video abstraction algorithms. Therefore we choose to implement a centralized service for making video summaries. Such service is running at the back-end on server all the time. Once a new video is uploaded, the process is triggered to extract salient images from the source.Thus the workload on mobile devices is minimized.

3.3 MoViShare Design and Implementation

The ultimate goal of our MoViShare system is to provide a universal video sharing platform that accommodates mobile accesses. As the first stage, we target a basic prototype system demonstrating the MoViShare fundamental concepts and ideas based on Nokia's hardware and software development environments. Via Wi-Fi access available in most new Nokia mobile device models, the prototype allows users to build connections with other mobile users and share videos. Basic functions of location-aware sharing and video abstracting will be demonstrated as well.

3.3.1 System Design

Our MoViShare system consists of four major modules.

Communication Module

This is a core module which supports all the communications between the clients and the server. It manipulates clients requests, translates them, and then sends them to the server. It is also responsible for all the instructions given by clients and the feedbacks from the server. All other modules are constructed based on this module.

Social Networking Module

This module serves the social networking functionality, playing an important role in creating and maintaining the mobile social networks among the users and their friends. It provides key social network functionalities, such as commenting and rating, etc. Enhanced services can be provided through cooperation with the location management module.

Video Processing Module

This module facilitates real-time video streaming over wireless channels. While streaming has been realized in current mobile platforms, our module will generate abstractions from large video files for users to digest. Working with the location management module, location-aware video displaying, browsing, and searching will be implemented as well.

MoViShare will also explore video popularity and correlations derived from social networks. Our experience with YouTube videos has shown that there are strong clustering behaviors among videos and users, thanks to social networks [26]. This offers great potentials for effective caching and pre-fetching of videos of interest.

Location Management Module

The location management module keeps track of the client location (through GPS or Wi-Fi access point's information). More importantly, it will work together with the social network and the video processing modules to promote location-aware medias to selected groups of users.

Figure 3.3 briefly shows the location-aware video display interface, which is directly presented on a map. Other interfaces including client side main interface, uploading video interface, video browsing and searching interface can be seen in our reference website.

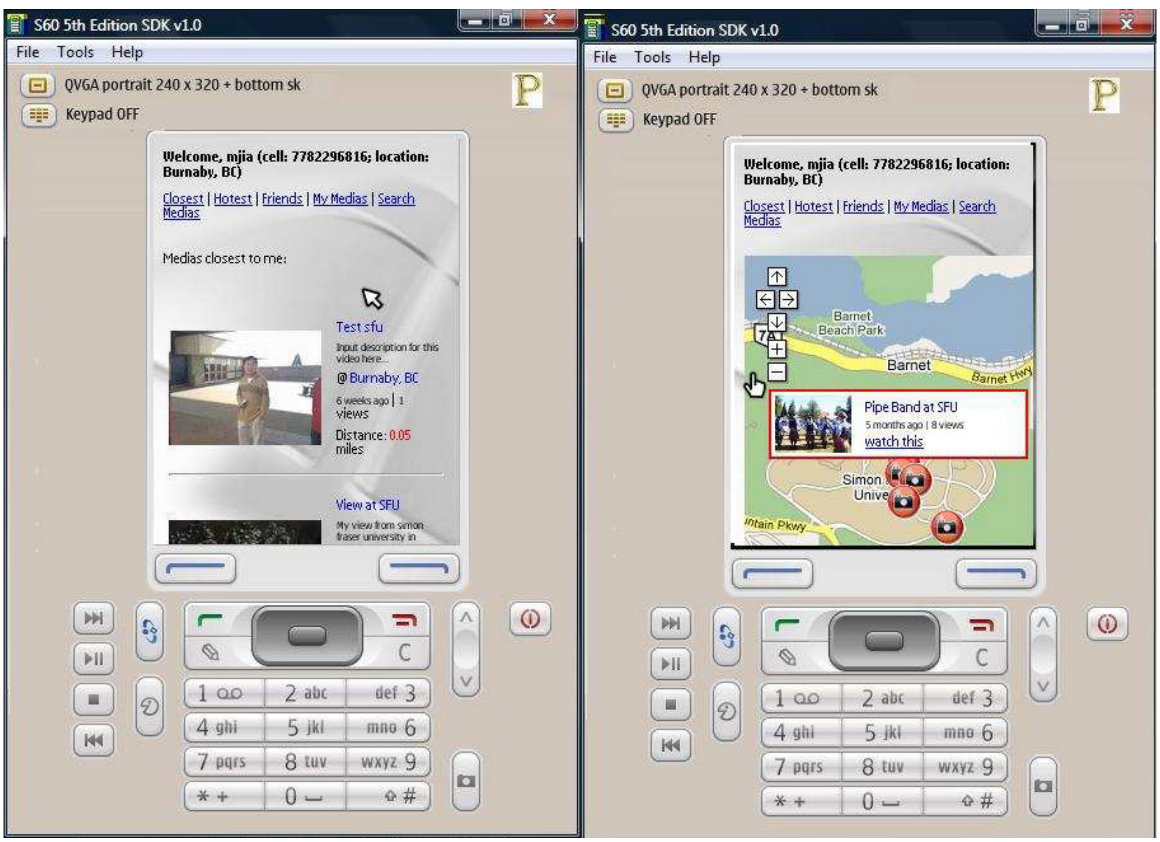

Figure 3.3: MoViShare interface with visualized location information

Users upload the media files to the system. Meanwhile, working with the integrated

GPS hardware, the location information will be uploaded as well. Therefore people can see where other users are, how far they are away from, or even a visualized location information by pictures or videos, together with other contextual information. Thus, they can further reveal nearby friends, places of interests. Also, real-time social activities can be deployed, which remain cumbersome in traditional social networking [42].

3.3.2 Prototype Implementation

Supported by Nokia, we have successfully built a MoViShare prototype on the N96 smartphones with the Nokia Symbian S60 development platform.

We applied the S60 Web Runtime technique [11] on the client side. The Web Runtime (WRT) extends the web browser for S60 to enable widgets and offers an optimized web experience that a user can access with a single click. On the server side, we adopt the Microsoft .Net 2.0 as the framework, using Helix server [4] to provide video streaming service. The WAP sites offer video uploading, searching and other functions to end users. We also integrated other web developing technologies such as Javascript, Google Map API to make the interface more friendly and intuitive. As such, the MoViShare can display videos directly on a map with small icons showing exact locations dynamically.

The MoViShare video processing is based on MPEG-4 standard, in which the data of information is stored in a nested box structure [6]. The Video Abstraction implementation works on several specific boxes. The Sample Table Box ("*stbl*") contains all the time and data indexing of the media samples (frames) in a track. One of its child boxes, the Sync Sample Box ("*stss*"), provides the set of the sequence numbers of all the random access points (i.e. I-Frames) within the stream. After getting the sequence numbers of a portion of I-Frames, we retrieve the corresponding chunks in Chunk Offset Box ("*stco*") and the size from the Sample Size Box ("*sts*"). This way the consumed memory and the processing time are both minimized. Considering that most of the videos captured by mobile phones are short and with a single scene only, it might not be necessary to apply shot-, cluster- or feature-based algorithms that cost much more processing time and energy [38] [43]. Yet we have started testing these advanced algorithms in our implementation as well [42].

We then covert the MPEG-4 frames to a single GIF file that is suitable for displaying in web interfaces or multimedia short messages, as shown in Figure 3.4.

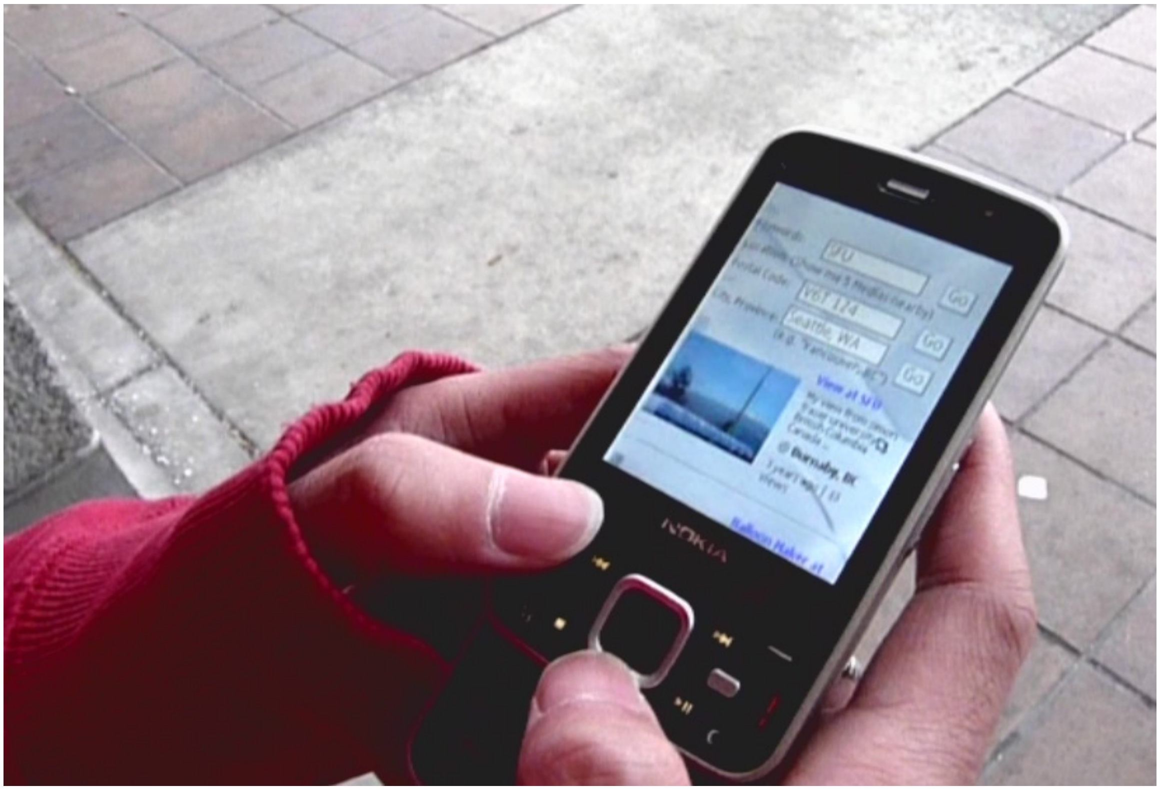

Figure 3.4: Snapshot of demonstration

3.4 Scoop Enhancements

Another project called Scoop was conducted with the collaboration of Trusterra [12]. While MoViShare targets the market of mobile social network for multimedia sharing for general uses, Scoop furthermore tries to get professional people focused on issues and pursuits that really matter. The global workforce is increasingly a mobile one, and it is predicted that by 2013, 35% of this workforce will consist of mobile knowledge professionals. An increasing number of organizations are adopting, even encouraging tele-commuting to attract employees striving for work-life balance. Lots of companies have tried to reduce the need to travel for financial and environmental reasons, while also many senior experts are typically located remotely from operational centers but desire to remain accessible and contribute. Since nowadays the mobile devices are capable of being connected to each other and multimedia ready, it is innovative to enable efficient communication, tele-presence and issue-focused collaborations.

Technical professionals are increasingly distributed and mobile. Thus they often need to collaborate across locations, company boundaries and time zones. Those companies' time is in high demand and they need to get maximum value from remote experts. But the old ways and tools does not measure up and instead have the experts run in circles with unproductive meetings, excessive travel, poorly described issues and complex user interfaces. Scoop then offers a fresh yet visual, mobile and simple approach that is in the needs of modern technical teams.

Scoop allows distributed or remote users to rapidly create issue-specific online workspaces from web and smart phone devices. Such embedded multimedia features allow users to utilize voice, pictures and video to clearly describe and document the topic at hand directly from the field. The users are capable of forming dynamic teams around each workspace and access the multimedia communication facilities to expand and discuss the topic at hand more efficiently than ever possible before [12].

3.4.1 Scoop Design and Implementation

Scoop enables a collaboration experience featuring the clarity and immediacy of an in-person or on-site meeting but without the need to coordinate same-time meetings or site visits and without the financial, environmental and personal burdens of field travel. It reduces the need to type with natural audio visual communications, and maximizes the value from remote experts, while cutting down unnecessary field costs.

The Scoop platform mainly consists of four parts.

Document and Description

Scoop leverages multimedia to describe issues and leads quickly and accurately at the source. In the system, each issue, or problem, is called a "scoop". The following Figure 3.5 shows the dashboard interface of Scoop, with the list showing all existing "scoops" that have already been assigned/shared with the current user. Figure 3.6 is the overview of a "scoop" showing all the detailed related information.

A user can create a "scoop" for the issue at hand from his mobile device. This works as a container keeping all relevant media and conversations together. There are a number of ways to describe a scoop. One or more media may be appropriate depending on each

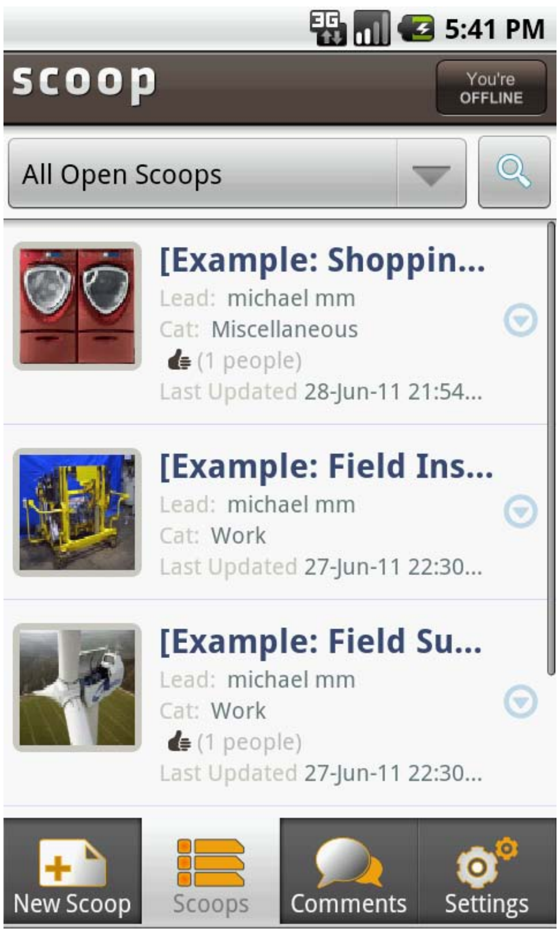

Figure 3.5: Scoop dashboard interface

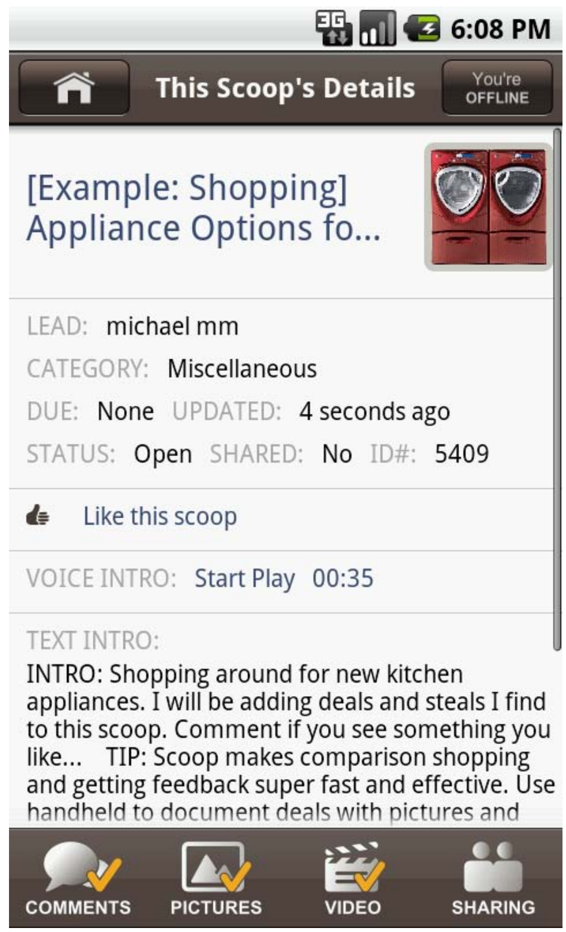

Figure 3.6: A Scoop Details

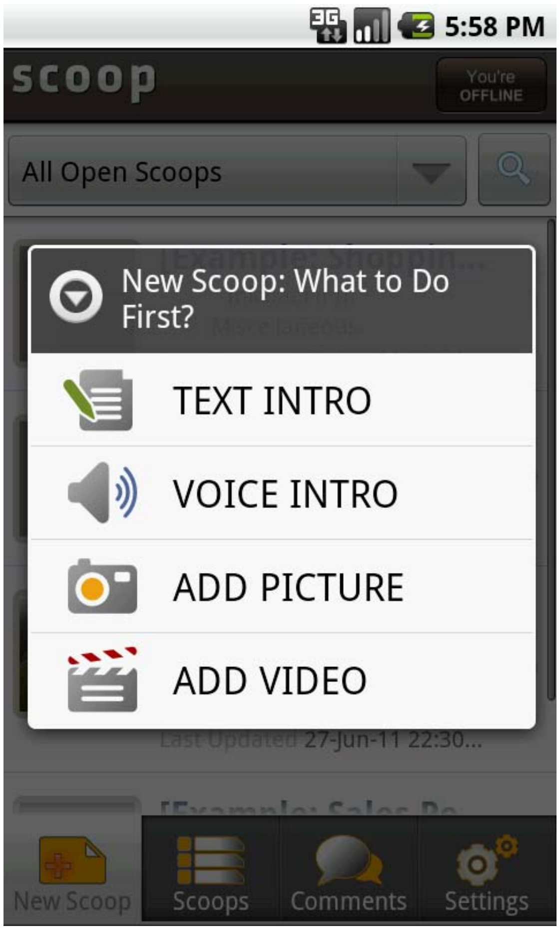

Figure 3.7: create a new scoop

Figure 3.8: pictures of one scoop

specific scenario, as shown in Figure 3.7. For example, a user can capture and add media then explain the details with voice or text captions and comments. Thus it minimizes the need of typing in the fields. Figure 3.8 shows an example of all pictures related to a "shopping scoop".

Teaming and Sharing

The system offers a way to rapidly form a micro-team for given issue. Therefore, the companies can minimize distraction to the rest of the whole team while getting the best suitable people focused on given issue. The following Figure 3.9 and Figure 3.10 are the screenshots of sharing module. An issue creator can add invitees to his scoop from contacts

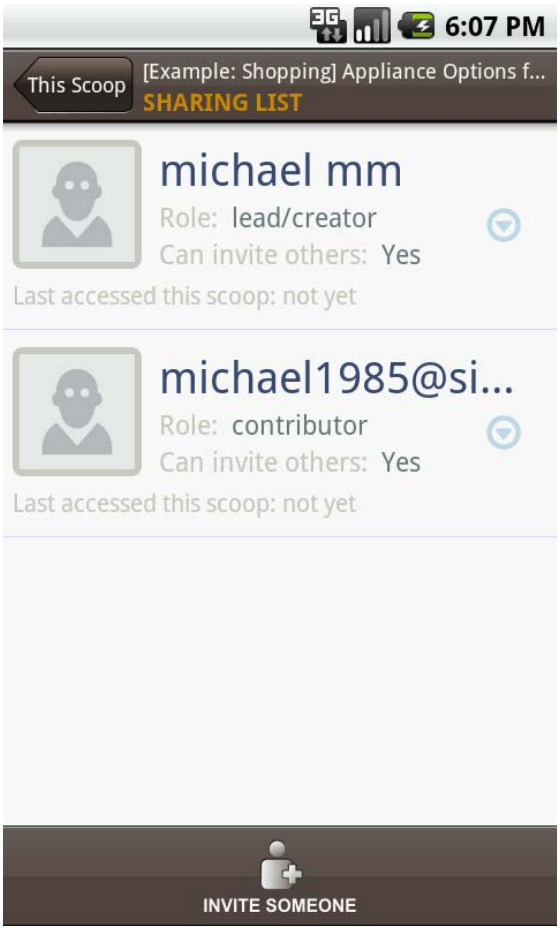

Figure 3.9: Sharing list of a scoop

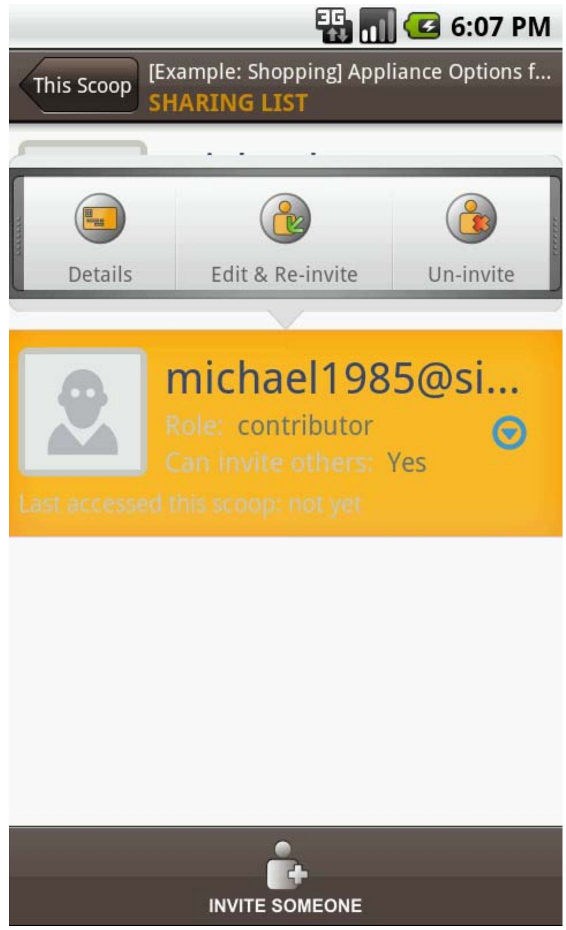

Figure 3.10: quick manipulations on invitees

or simply type in their email addresses, and can let his invitees invite others, while controlling different levels of access by assigning different team roles.

Discussion and Resolution

With network connected mobile devices, a user can comment on specific details using text or voice, and get notified of the latest updates for each issue, which increases the communication speed and accuracy. The following Figure 3.11 shows the always up-to-dated comments list about the issues available to the current user. Comments, or a simple like/unlike attitude can also be made on pictures, videos and voice clips, which is shown in Figure 3.12.

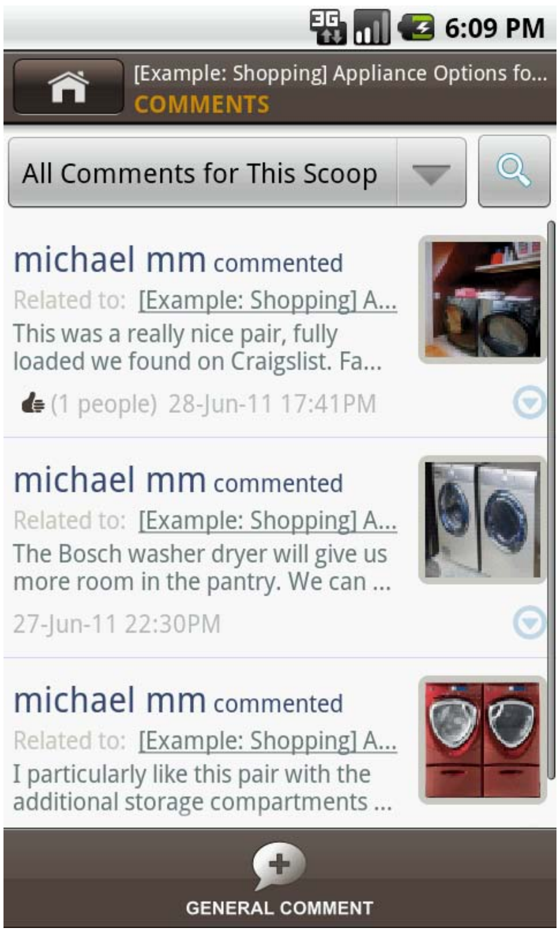

Figure 3.11: Sharing list of a scoop

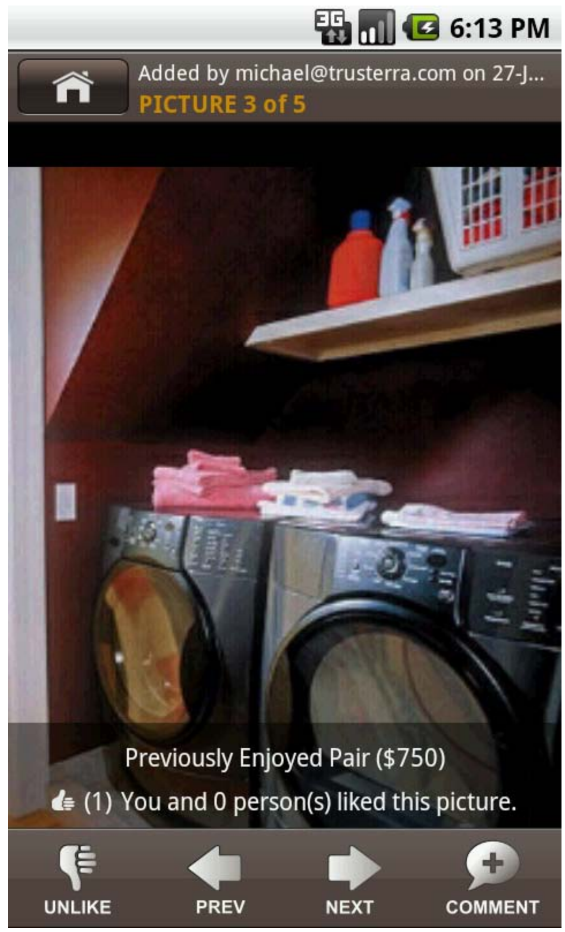

Figure 3.12: quick manipulations on invitees

Synchronization and Contribution

With the synchronization module, a user can access the latest data at his office or on the road. It is convenient to turn the mobile device into a mobile command center, by utilizing continuous 2-way synchronization.

Chapter 4

Trajectory Reconstruction and Evaluation

After explored and studied location-aware mobile social networks with two systems MoViShare and Scoop built, we found a series of properties of mobile network and issues it might be facing. One important problem of such mobile networks is the trajectory exposure problem.

4.1 Motivation

We now discuss the motivation toward the *Trajectory Exposure Problem* in mobile routing in more details. Clearly, if a mobile user exposes every location it visits, the whole trajectory of its route can be accurately reconstructed. On the other hand, if no any location is exposed, then the trajectory is basically hidden as well. While existing systems implements one or the other, between the two extremes, the mobile user indeed may expose some of the locations only. Besides the obvious privacy concerns, users might have different concerns. Considering the limited battery energy, the user (or even the mobile devices itself) might not always turn on GPS devices. Even when the GPS is turned on, depending on its implementation, the location update rate is often low. The first "real-time" tracking device New WorldTracker GPS can only update every 15 seconds.

Fig. 4.1 provides an example where three locations along a route are exposed by the mobile user. Given these incomplete *location samples*, an adversary can predict the trajectory of the user. With limited location samples, the predicted trajectory clearly will deviate from the real one. More importantly, the deviation largely depends on the quality of the exposed location samples, while not necessary their amount, as illustrated in Fig. 4.1.

What we are interested in this thesis, as motivated by the above example, is to understand the basic relations between the location exposure and the trajectory exposure. More explicitly, how can we evaluate the prediction deviations and how can the locations to be exposed be selected so that the risk of trajectory exposure be minimized? Beyond this, we look further into a practical protocol design that can utilize the location information flexibly and adaptively. With a single control parameter (the *exposure rate* of a trajectory), the protocol will work in the whole spectrum from zero location knowledge to full knowledge, so as to fit diverse applications' demands.

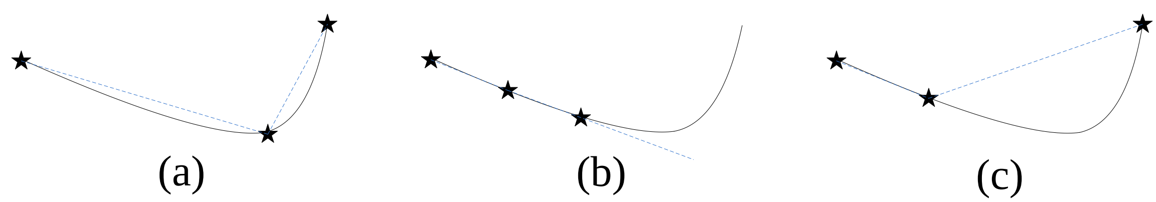

Figure 4.1: Different trajectory predictions

4.2 Trajectory Reconstruction and Evaluation

In this thesis, we will focus on the trajectory prediction with *Gaussian Process Regression (GPR)* [44]. The Gaussian Progress Regression has been widely used in different fields. For trajectory prediction, it incurs much lower overhead and yet achieves excellent accuracy as compared with other state-of-the-art tools e.g., multi-layer neural networks [19]. Our solution framework for the trajectory exposure problem however are generally applicable with other prediction tools, and we will also compare with other classical tools in later chapters.

We now briefly review GPR; more details can be found at [44]. We will also derive the trajectory exposure rate under the GPR model in this chapter. Before reviewing, we first give some notations helpful for understanding, in the following Table 4.1.

Table 4.1: Term Notations

x	random variable
$f(x)$	target value of variable x
$f_*(x)$	predicted target value of variable x
f	vector of sampled data sets, and f $= [f_1, ..., f_n]$ on sampled data set X
f_*	vector for target value of the estimate data set
$m(x)$	mean function of random variable distribution, often considered as 0
$k(x, x')$	general form of covariance function
X	matrix of the sampled data set, where $X = [\mathrm{x}_1, ..., \mathrm{x}_n]$
X_*	matrix of the estimate data set
$\overline{f_*}$	mean value for estimated target value on estimate data set
y	y is $f(x) + noise$, which is the real target value plus noise

4.2.1 Gaussian Process Regression (GPR)

A Gaussian Process [44] is defined as a collection of random variables, any finite number of which have joint Gaussian distributions, and is fully specified by a mean function and a covariance function. The inference of continuous values with a Gaussian process prior is known as Gaussian process regression. The mean is often set to zero for notational simplicity, and hence the Gaussian Process Regression only includes determining covariance function. The covariance function can be defined by a Radial Basis Function (also called Squared Exponential) with some hyper-parameters inside the function. The GPR has a natural Bayesian interpretation and has various desirable properties, e.g., ease of obtaining and expressing uncertainty in predictions and the ability to capture a wide variety of behaviors through a simple parameterization [21].

Consider x as a general random variable, a Gaussian process can be written as:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')), \tag{4.1}$$

where $m(x)$ is the mean function of the distribution of x, and $k(x, x')$ is the covariance function. The latter specifies the covariance between pairs of random variables. When observations are noise-free, given two samples of variable x, namely x_p and x_q, we have

$$\mathrm{cov}(f(\mathrm{x}_p), f(\mathrm{x}_q)) = k(\mathrm{x}_p, \mathrm{x}_q) = \exp(-\tfrac{1}{2}|\mathrm{x}_p - \mathrm{x}_q|^2), \tag{4.2}$$

as one possible situation. As mentioned, for notational simplicity, the mean function $m(x)$ is generally set to 0. However, considering noises, we assume additive independent identically distributed Gaussian noise ε, with variance σ_n^2, such that $y = f(x) + \varepsilon$. And thus the prior is

$$cov(y) = K(X, X) + \sigma_n^2 I. \tag{4.3}$$

The specification of the covariance function implies a distribution over functions. By choosing a number of points X_* to estimate, we can generate a random Gaussian vector with this covariance matrix into a Gaussian Distribution,

$$\mathrm{f}_* \sim \mathcal{N}(\boldsymbol{0},\ K(X_*,\ X_*)). \tag{4.4}$$

If the vector f contains sampled data outputs of function f and we would like to predict the value of f_* on estimate points X_*, considering noise, the joint distribution according to the prior will be

$$\begin{bmatrix} \mathrm{y} \\ \mathrm{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0},\ \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right). \tag{4.5}$$

The posterior distribution over functions can be obtained by restricting the joint prior distribution to include the one which agrees with the sampled data

$$\mathrm{f}_*|X, \mathrm{y}, X_* \sim \mathcal{N}(\overline{f_*}, cov(f_*)), \text{where} \tag{4.6}$$

$$\begin{aligned} \overline{f_*} &\triangleq \mathbb{E}[f_*|X, \mathrm{y}, X_*] \\ &= K(X_*, X)[K(X,X) + \sigma_n^2 I]^{-1}\mathrm{y} \end{aligned} \tag{4.7}$$

$$cov(f_*) = K(X_*, X_*)[K(X,X) + \sigma_n^2 I]^{-1} K(X, X_*). \tag{4.8}$$

Thus f_* which corresponds to X_* can be sampled from the joint posterior distribution by evaluating the mean and covariance matrix from Equation 4.6, and it gives the predicted mean value $\overline{f}(X_*)$ and an prediction uncertainty related variance $\mathbb{V}[\mathrm{f}_*]$ [44][28].

4.2.2 Trajectory Reconstruction

We then focus on how an estimated trajectory can be constructed, using the Gaussian Process Regression with a selection of sampled location points.

A trajectory is the path a moving object follows through space as a function of time. Specifically, we have sampled location points from x_1 to x_n, referring as the *location samples* in Section 4.1. The location points are represented by two dimensional points $\mathrm{x}_i = (x_i, y_i)$, and X is the sampled data set for all (x_i, y_i)s. Trajectories will be represented by generated Gaussian Process Regression functions which is determined by a covariance function and a mean function. Therefore we hope to learn f such that it can be mapped to the changes in location: $\Delta\mathrm{x}_{i+1} = \mathrm{x}_{i+1} - \mathrm{x}_i$.

We adopt Gaussian process Regression reviewed in the previous Section 4.2.1 to estimate trajectory. By deriving the conditional distribution in Equation 4.6, if we let $K = K(X, X)$, $K_* = K(X, X_*)$,to denote the vector of covariance between the estimate points and the n location samples, and $k(x_*) = k_*$ in case there's only one estimate point x_*. Therefore, compactly we can get the scenario of one estimate point case, where Equation 4.7 and Equation 4.8 can be shorten as

$$\overline{f_*} = K_*^\top (K + \sigma_n^2 I)^{-1}\mathrm{y} \tag{4.9}$$

$$\mathbb{V}[f_*] = k(x_*, x_*) - k_*^\top (K + \sigma_n^2 I)^{-1} k_*. \tag{4.10}$$

On obtaining Equation 4.9 and 4.10, we can predict the changes in two trajectories at future times. The mean of the prior Gaussian process is normally determined to be zero. And for covariance function, we can use bias term and noises added version of original, as shown in Equation 4.3. After combining them, we get the following Algorithm 1 for prediction. Here X is the data set, y is the target value, k is covariance and σ_n^2 is the noise, while x_* is the data for estimation. And $\overline{f_*}$ is the mean value, $\mathbb{V}[f_*]$ is variance, while $logp(y|X)$ is marginal likelihood. In the algorithm, $cholesky(K + \sigma_n^2 I)$ is the Cholesky decomposition on the matrix of $K + \sigma_n^2 I$.

About the hyperparameters in the noisy squared-exponential covariance function,

$$k_y(x_p, x_q) = \sigma_f^2 exp(-\frac{1}{2l^2}(x_p - x_q)^2) + \sigma_n^2 \delta_{pq}, \tag{4.11}$$

setting different values for length-scale l, signal variance σ_f^2, and noise variance σ_n^2 might get different trajectories, but some under-fits data and some over-fits. We can get the values by

Algorithm 1 Predictions(X, y, k, σ_n^2, x_*):

$L = cholesky(K + \sigma_n^2 I)$
$\alpha = L^\top \backslash (L \backslash y)$
$\overline{f_*} = k_*^\top \alpha$
$v = L \backslash k_*$
$\mathbb{V}[f_*] = k(x_*, x_*) - v^\top v$
$logp(y|X) = -\frac{1}{2} y^\top \alpha - \Sigma_i log L_{ii} - \frac{n}{2} log 2\pi$
$return(\overline{f_*}, \mathbb{V}[f_*], logp(y|X)$

maximizing the hyperparameters (l, σ_f^2, σ_n^2) through optimizing the log marginal likelihood $logp(y|X)$ in Algorithm 1. To obtain this, we do a partial derivatives of the marginal likelihood w.r.t. the hyperparameters, such that

$$\begin{aligned} \frac{\partial}{\partial \theta_j} logp(y|X, \theta) = \frac{1}{2} y^\top K^{-1} \frac{\partial K}{\partial \theta_j} K^{-1} y - \frac{1}{2} tr(\\ K^{-1} \frac{\partial K}{\partial \theta_j}) \\ = \frac{1}{2} tr\left((\alpha\alpha^\top - K^{-1}) \frac{\partial K}{\partial \theta_j} \right), \end{aligned} \tag{4.12}$$

where $\alpha = K^{-1} y$. A More detailed explanation can be referred to [44].

4.2.3 Exposure Rate Evaluation

To calculate the exposure rate of a trajectory, we compare how much corresponds the estimated trajectory is towards the original trajectory, in which we calculate the deviation $(\overline{f_*} - y)$ at each location point, and summarize along the whole path. We introduce a measurement Exposure Rate (ER), which ensures the overall consistency limited within r%, as in Equation 4.13, where the exposure rate gets 1 when $\overline{f_*}$ has value the same with y, and gets a minimum of 0 when the deviation of $\overline{f_*}$ towards y is $|y|$. When the deviation is too large and exceeds threshold, the exposure rate will be counted as 0, although it is indeed negative.

$$ExposureRate(ER) = \frac{1}{n} \sum_{i=0}^{n} \left(1 - |\frac{\overline{f_{*i}} - y_i}{y_i}| \right) * 100\%. \tag{4.13}$$

We then can ensure location trajectory exposure to be maintained within a percentage $r\%$ using the above equation. If the computed result is less than r%, then the differences between the original trajectory and the predicted trajectory is great enough, so that the exposure is safe and trajectory exposure is maintained under a Exposure Rate ER. Otherwise the exposure is beyond the limitation and it is over exposed.

4.2.4 Controlling Exposure Rate

We now address the problem of how to control the trajectory exposure rate $r\%$ through a careful selection of exposed locations. As we have illustrated in Fig. 4.1, exposing different location samples, even with the same quantity, can lead to dramatically different trajectory reconstruction, and hence different exposure rate. A natural question therefore is: *given a threshold r%, which locations can be exposed by a mobile user, such that the exposure rate of its trajectory does not exceed r%?*

Clearly, an optimal solution of this problem for general trajectories involves the enumeration of different subsets of location samples. It is known that state-of-the-art GPS devices updates the location information at a rate of $15s/update$. That is, for a constantly moving user, 4 samples can be generated in one minute, and the total number quickly accumulates over time. As such, an optimal solution is computational infeasible, particularly for computation-power-limited and memory-storage-limited mobile devices.

We now present an effective heuristic that tries to capture the key features of a trajectory. We define the *critical value* of each location sample as the criticalness when the location is exposed. The higher the value is, the more critical it should be towards the trajectory. The points representing features mostly occur when there is a direction turning in the trajectory, as the middle point in Fig. 4.1 and Fig. 4.2(a). Thus the middle point in Fig. 4.2(b) is not critical. We define the location critical value by comparing a location sample with its neighbor points, to determine if it represents some feature of trajectory, in Equation 4.14:

$$CriticalValue = |\frac{\Delta y_{left} + \Delta y_{right}}{2}|. \tag{4.14}$$

X values can also be used in the substitution of y values.

The heuristic solution works as the following: First, the critical value of each location sample of the whole trajectory is evaluated. Then the location samples are sorted in ascending order by critical values. Every iteration we choose a point with minimum critical values, so on and so forth, and apply our proposed mechanism to do prediction and see if $r\%$ is maintained. Thus, given the threshold of $r\%$, we can determine the maximum number of location samples that can be exposed and which these points are, and all possible ways of different location points' combinations.

However, there are certain percentage of people that do not always move based on a routine. Their trajectory are not periodically repeated and a future trajectory can not be decided. Based on historical location samples, we perform a practical implementation with progressive trajectory exposure. We can periodically update the sender's historical location points, generate a new up-to-date trajectory periodically. Then based on historical records and the *Already Exposed Location Points*, we can progressively choose new points to expose, after assuring such new selection will not result in a predicted trajectory exceeding the limit, with the using of GPR to calculate the Exposure Rate, and only expose if the selected location samples will have the exposure rate less than $r\%$.

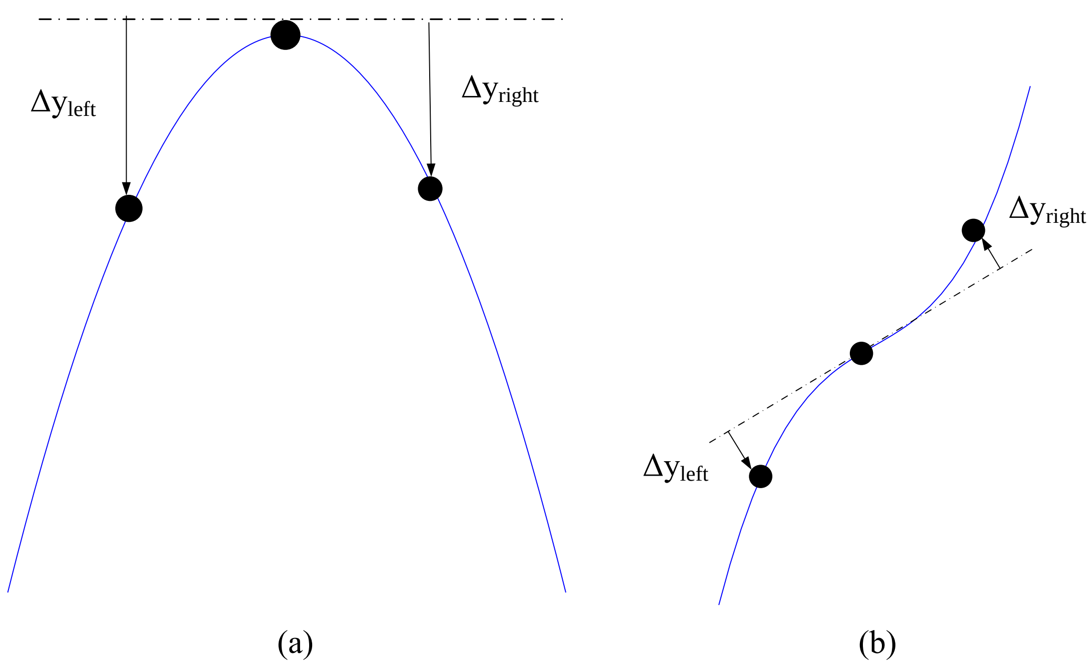

Figure 4.2: Different cases in determining location critical value

Chapter 5

Mobile Routing with Partial Trajectory Exposure (MoRPTE) — A Case Study

We now proceed with a case study using our trajectory exposure control tool for mobile routing. A conventional wisdom for location-based mobile routing is to decouple the location advertisement and path determination [47][40]. First, a location advertisement module enables each node to broadcast its locations to others with which they can use in routing. Such advertisement can be done by periodically broadcasting its location information to others that spreads out the whole network or to some server which keeps records of all nodes' up-to-date location data. Such information will then be used by the path determination module to locate the best route.

However, such routing schemes assume nodes' location advertisements real-time, which is not in real world. A user may have different concerns as mentioned in Section 4.1. Therefore realistically, a user cannot expose its trajectory at a exposure rate of 100%. In the state of art location-based routings, all locations of nodes are assumed to be up-to-dated. But the location of a user's neighbor might be updated a period of time ago, leaving its current location unknown. We target SOGR [47] and present our MoRPTE (Mobile Routing with Partial Trajectory Exposure) which utilizes predicted location in substitute of the previously advertised locations, in order to make routings more efficient, while also considering trajectory privacy.

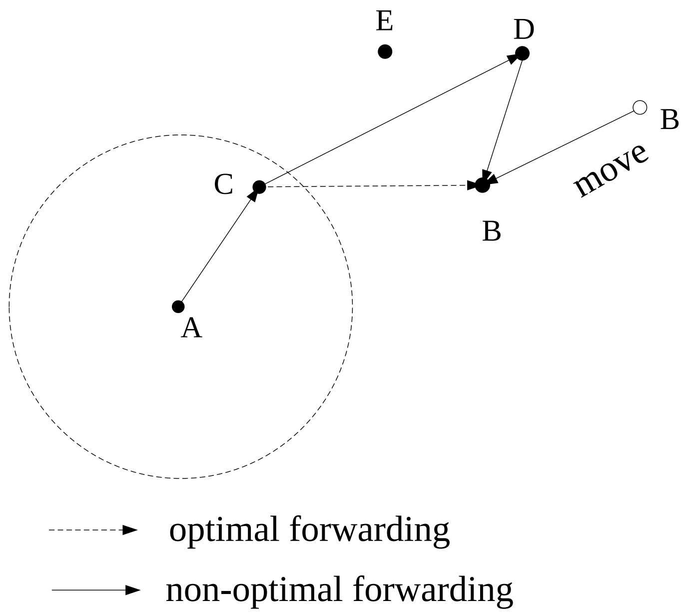

Figure 5.1: Less precise prediction occurs

In the original SOGR protocol, an intermediate node makes packet forwarding decisions based on its knowledge of the neighbors' positions and the destination's position. It uses a geographic-based greedy forwarding that forwards data greedily to a neighbor that progresses the most towards the destination. However, the locations of intermediate nodes are advertised periodically and not up-to-date, though it is assumed to be real-timed. We adopt a predicted location in substitute of original location, so as to deliver the packets efficiently, while the rest of SOGR routing protocol is remained the same, including back-off replies and route optimizations.

The advertisement of locations is thus more flexible, adopting the proposed GPR prediction in Section 4.2.2. Predicted locations are now used for calculation to determine next-hops. Thus, when a node doesn't advertise its location for some time, an estimated location will behave better than last-updated location. SOGR offers a simple estimation during the phase of validity estimation, in which simply last two recorded locations are used to form a simple line. Our prediction methods behaves more realistic, considering the randomness and realities of node movements.

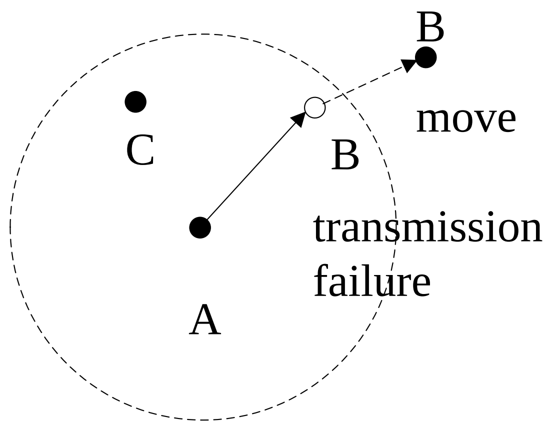

Figure 5.2: Handling inaccurate destination location problem

Although it seems that locations are simply substituted by predicted ones, there exists a certain problem, that prediction might sometimes bring non-optimal paths. Different nodes might obtain different historical data of each other, which will bring different predicted future locations of intermediate nodes. Thus during path calculation, a prediction not precise enough might be generated and a non-optimal next-hop might be selected. As the Fig. 5.1 shows, A wants to deliver a packet to the destination B based on the predictions of current locations of its neighbors and the location of B, it selects C as its next-hop. However, C might not be able to predict the real-time current location of B, and will select D as its optimal next-hop. D predicts that B will move to the current new location, it will send the packet to B and thus brings a longer path than directly from C to B. We solve the problem with the help of other overhearing nodes. During the forwarding, if a node E which possesses more information about the movement of B and can predict its current location, E will send a correction message to C, notifying C to change its next-hop as B, when $dis_{(C,B)} < dis_{(C,D)} + dis_{(D,B)}$. Our mechanism can also handle the inaccurate destination location problem. As shown in Fig. 5.2, if B moves out of the transmission range of A but A doesn't predict that, the transmission will fail. However another node C can predict B's movement and correct A's next-hop so as to fit the current location of B.

Therefore, all locations for calculation are predicted and thus fit more of the real-time locations, rather than the ones advertised before. It brings more efficient and precise data forwardings, also guarantees the exposure rate during location advertisements.

Chapter 6

Performance Evaluation

To examine the performance of our proposed mechanism, we have conducted a series of experiments, both with real traces and synthetic data. The traced data we use is the real trajectory correspondent data of taxis in Beijing, China. Such data is administrated by Beijing government and is hard to obtain. The data reflects real situations of taxis' movements throughout the whole city, and one single of all data files contains $10,050$ taxis' trajectories with size over 100MB for one day. The data is collected periodically, with each taxi's identity, time, GPS location, velocity, direction information. The capacity of data is too huge so we picked a number of typical ones for performance evaluation. "Typical" means the taxi is mostly moving rather than mostly staying at a same location.

6.1 Performance of MoRPTE Routing when Having Different Trajectory Exposure Rates

For instance, the trajectory of taxi A73198 from $20:38$ to $23:56$ is shown in Fig. 6.1, in which the coordinate has been normalized. There are 100 nodes randomly deployed in the area, which represent intermediate nodes. For clarity, here we only show 15 of them, which are used in the following comparisons. By MoRPTE routing, the following figures show the routing paths when setting the trajectory exposure rate parameter r to 10%, 50%, 100%, as in Fig. 6.2, Fig. 6.3 and Fig. 6.4, where different colored lines denotes different routing paths.

We can find in the graph that if the parameter is set to be 100% as shown in Fig. 6.4,

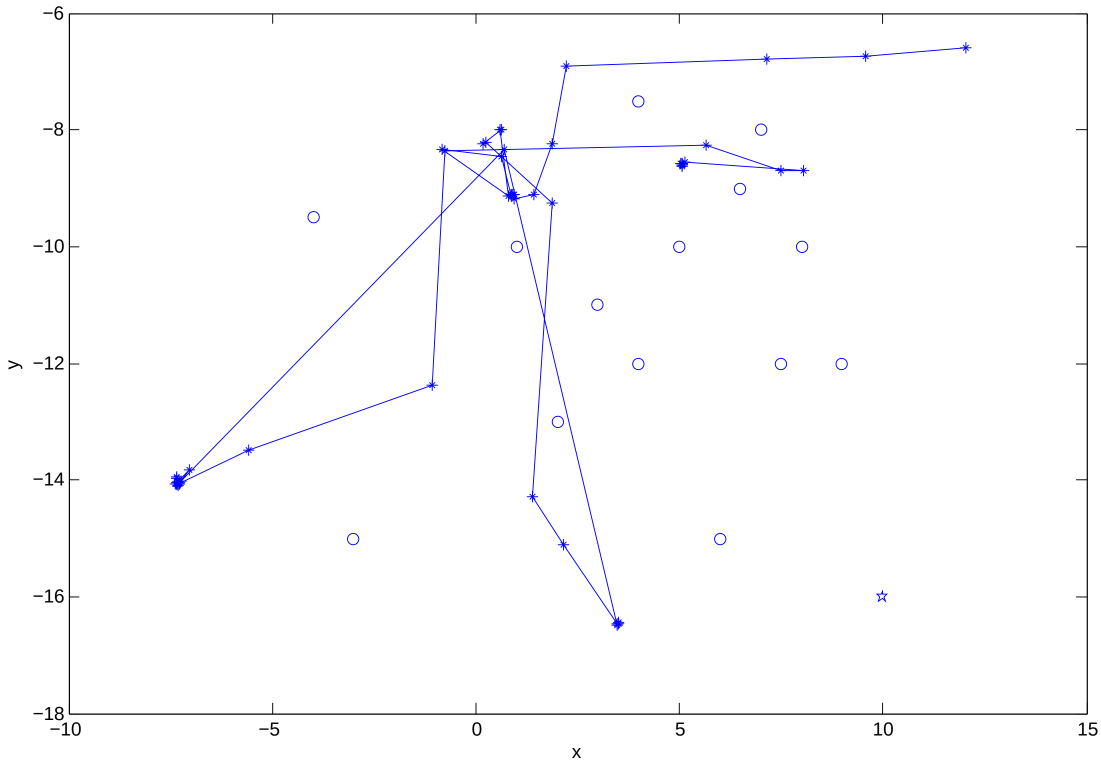

Figure 6.1: Taxi A73198 trajectory path sample

the source node does not care about exposure of its trajectory information; therefore the routing paths are the same as when using MFR. Fig. 6.5 shows the trajectory prediction progress of one intermediate node. We generate prediction using another taxi BG0544 data which has 200 more entries.

We can see from the figure that, at some nodes the prediction consistency is low comparing with other points, which is due to data latency and loss. Therefore, using integration does not help to determine the consistency of prediction. Because of discontinued data sampling, we achieve the judging by Equation 4.13 instead. The previous case study shows the performance of MoRPTE routing, which dynamically controls users' trajectory exposure adaptively in the spectrum from zero knowledge to full knowledge, limiting the exposure rate within $r\%$.

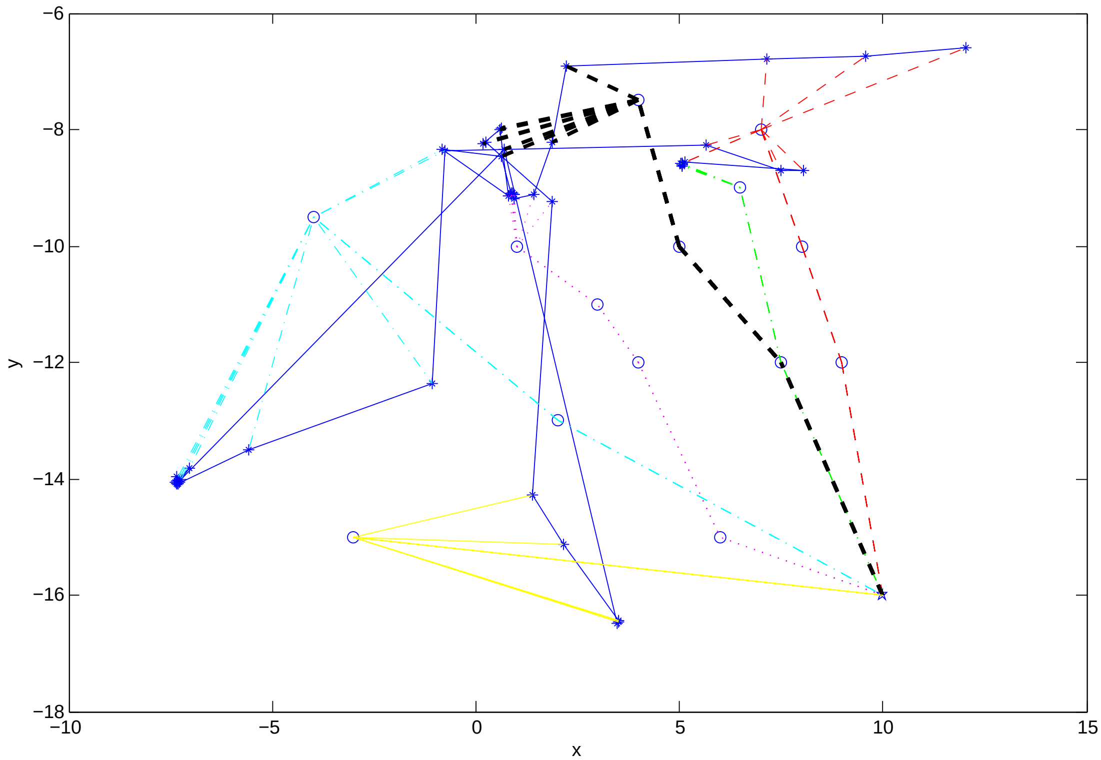

Figure 6.2: Routing Paths when r%=10%

6.2 Performance on Predicted-location-based Data Forwarding

We simulated our mechanism MoRPTE on 1000 random nodes to compare with SOGR-HR which use last-advertised position as node's current location; AODV which do not use location information and GPSR [35]. We set SOGR parameters $Ref_{backoff}$, $Inc_{backoff}$, $Dis_{timeout}$ and $[Min_{timeout}, Max_{timeout}]$ to be 10ms, 2ms, 300m and [10s, 30s], with the Max_{hops} as 2. The movement models follows random way point model, setting the maximum speed to be 0m/s to 50m/s. Also, an advertisement option is added randomly, for each node to choose whether or not to advertise its current location. We study the same metrics as in [47]: packet delivery ratio, which is the ratio of packets delivered; control overhead, that the total number of control messages sending over each hop divided by the total number of

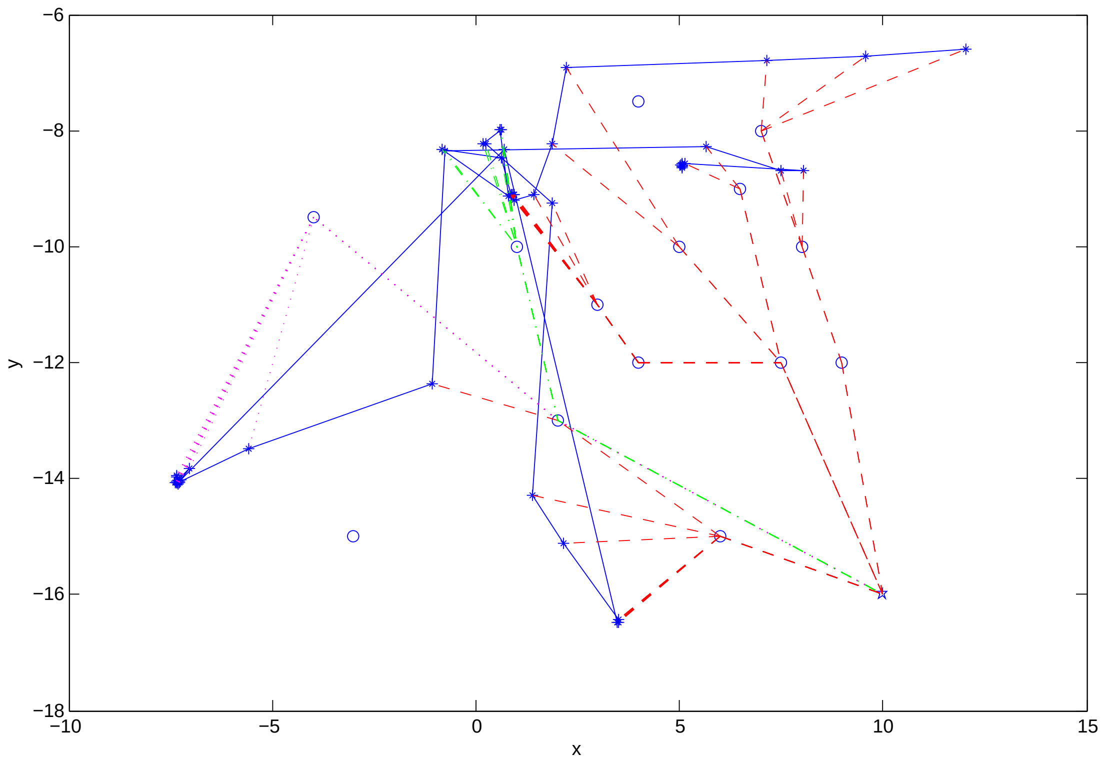

Figure 6.3: Routing Paths when r%=50%

packets received; and the total number of data packet forwarding accumulated from each hop over the total number of data packets received.

The scalability of AODV is limited, which is because of limited network range and restricted flooding. Therefore, when the networks are dynamic, the routes can be easily broken and packets are dropped. On the contrary, location-based routings such as MoRPTE are more scalable and robust. The following Fig. 6.6 shows the packet delivery ratios of different routing protocols with different maximum moving speeds. The location-oblivious routings have lower delivery ratios than location-based routings. MoRPTE behaves stable even when the nodes move fast, with the help of GPR location prediction of intermediate nodes. SOGR only use advertised locations and thus when nodes do not advertise its location for a period of time, the performance will be affected. Also the performance of SOGR decreases than the experiments conducted in [47], which is due to the random advertising

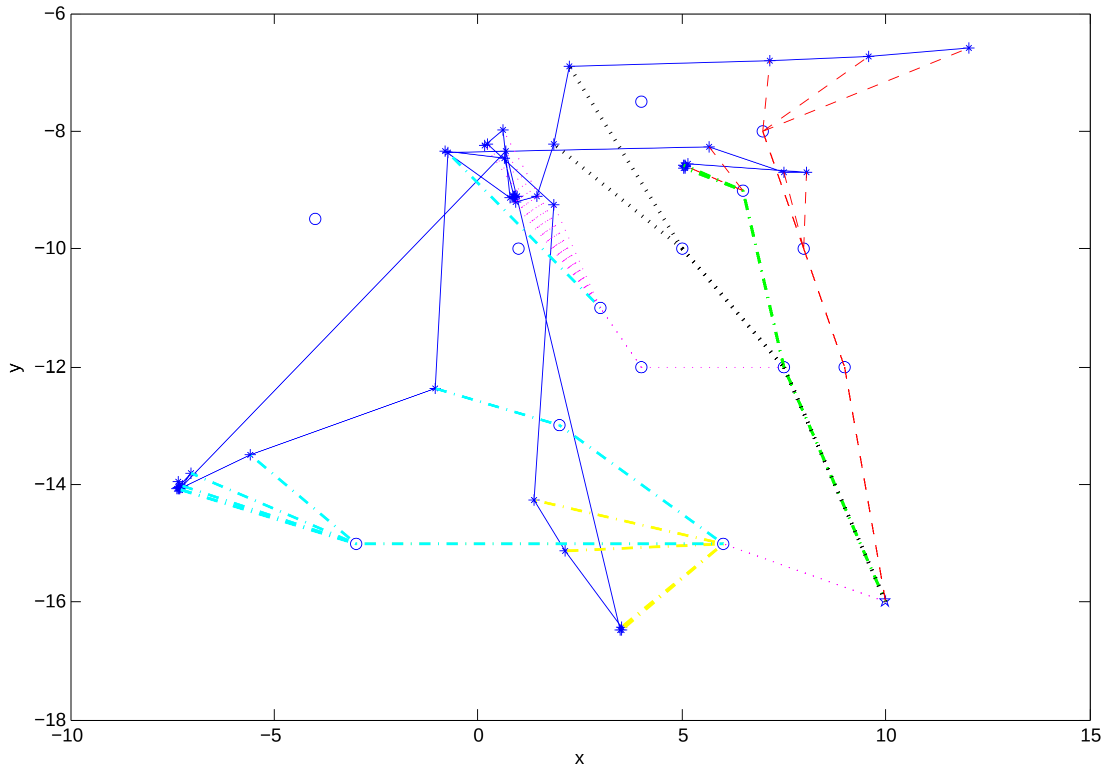

Figure 6.4: Routing Paths when r%=100%

and de-advertising of locations by intermediate nodes.

In Fig. 6.7, we find that MoRPTE generates least control messages. SOGR generate reasonably more messages as mobility increases. In MoRPTE, predicted location fits real-time location better than previously advertised data. GPSR brings unnecessary overhead when the movements are slow. Fig. 6.8 shows that GPSR has the maximum overhead, due to non-optimal routing when topology is changed dynamically. SOGR's optimization process helps with the route changing and thus performs better. MoRPTE handles the situations too when in-precise prediction and detour occurs.

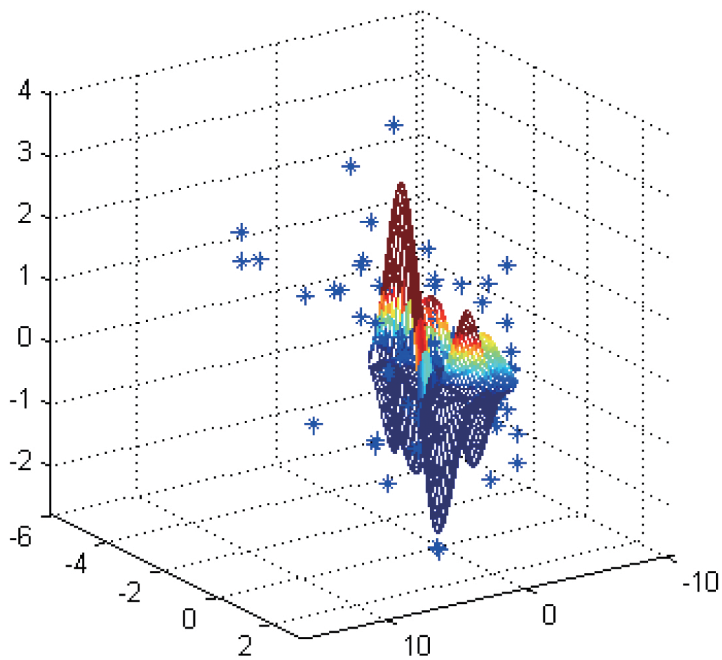

Figure 6.5: The prediction at one intermediate node

6.3 Performance of General Estimation Model with Gaussian Process Regression for Trajectory Exposure Control

Next we'll show the performances compared with other predicting methods. We carry out synthetic-data-based experiments in a large simulation system. The nodes and users' trajectories are randomly deployed. The nodes move following the model of Random Way Point, with the speed between 1 and 20 m/s, and a pause time 30s. As Fig. 6.9 shows, we set the network size to be 50, 100, and 500, in order to examine the impact of network size on the mechanisms' performances by the metrics of consistency rate and deviation.

The more precise one can reach, the more consistent the predicted trajectory should be with the original trajectory, the greater consistency rate it should be, while the error should

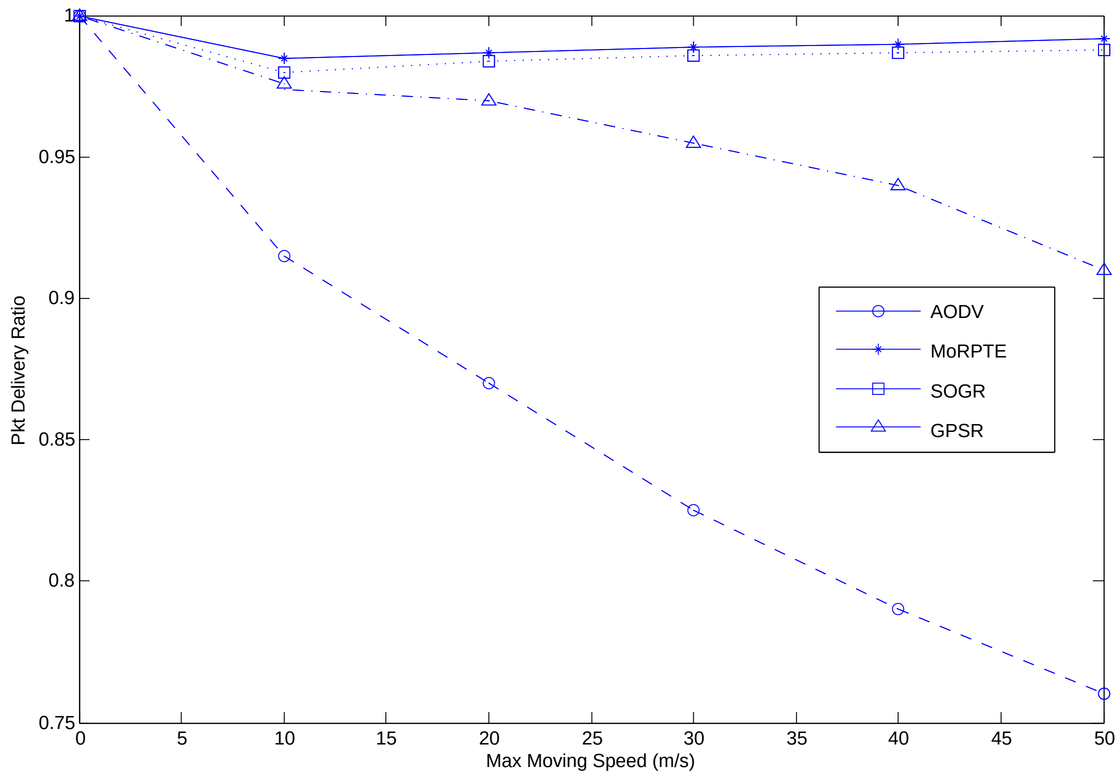

Figure 6.6: packet delivery ratio

be smaller which stands for the deviation between prediction and the original.

6.3.1 Comparisons with other prediction mechanisms

Upon network size, we also conduct other parameters such as $r\%$, and errors between predicted and original trajectories. We compare MoRPTE with the parametric approaches with exponent level 1 to 3, and multi-layer neural networks, which is one of the best prediction mechanisms mentioned in Chapter 2.

Fig. 6.11 shows the result of neural network, where we chose a model with 1 hidden layer and 2 neurons for the hidden layer for not to incur too much overheads in generating the model. However it can only have outputs in a certain interval and is not possible to predict values outside the interval. Besides, another disadvantage as stated in [18], is that to determine the correct model, complexity and to build a model with desired complexity

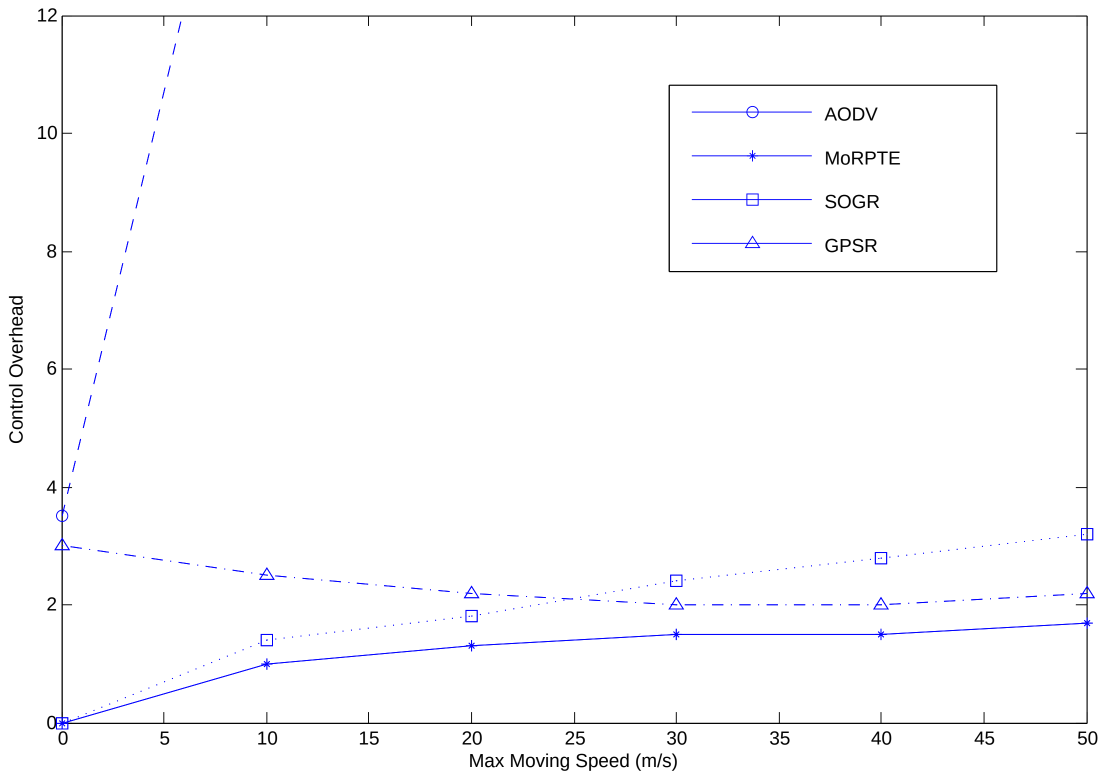

Figure 6.7: control overhead

are often expensive.

Fig. 6.10 shows the different performances of the three when network size is different. We can find that generally polynomial has the most significant deviation towards the real trajectory. Neural Network performs better, while the proposed GPR the best as expected. As the network size grows, the polynomial with different exponent levels perform more different. This is because with the increase of network size and trajectories, the inconsistency rate will be amplified. Also, the error rate, which means the ability of prediction by corresponding methods, differs and our approach performs the best among the three, considering complexity and overheads.

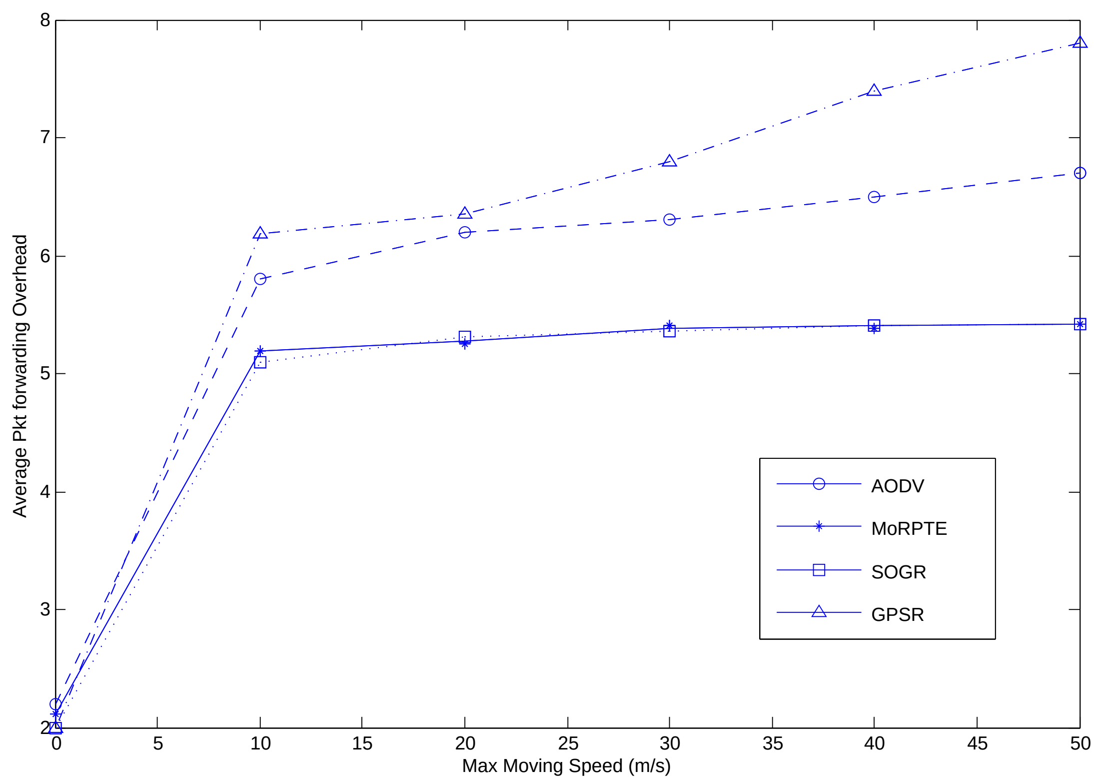

Figure 6.8: average number of data packet forwarding

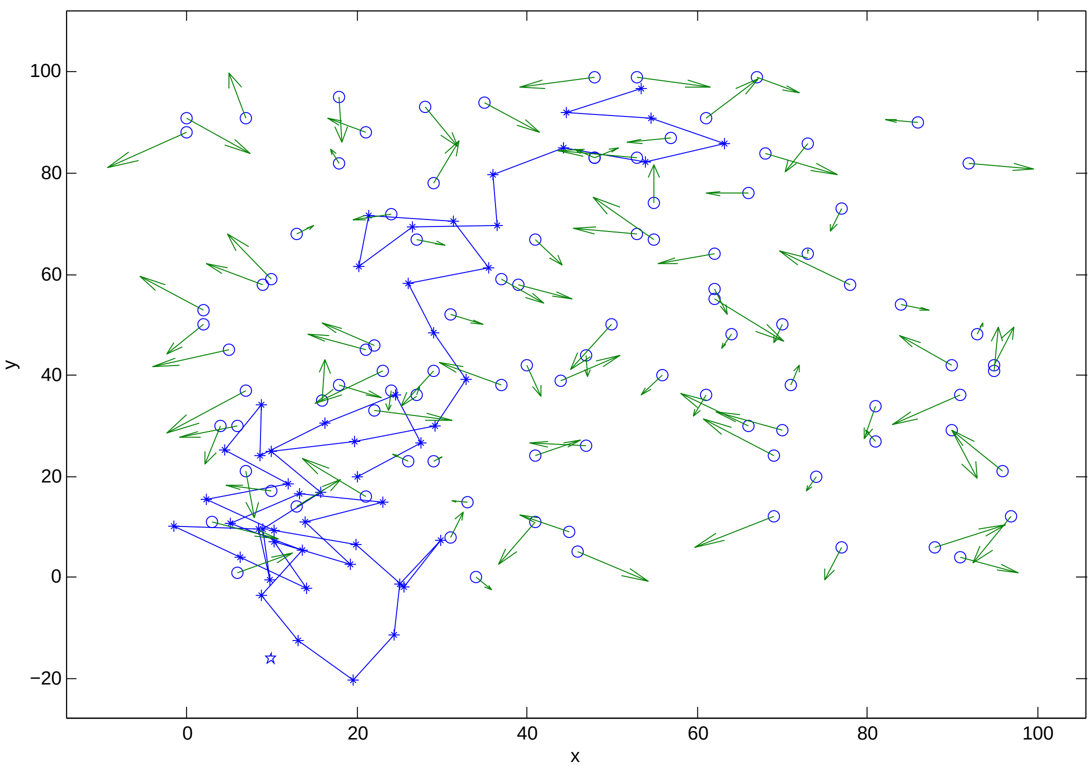

Figure 6.9: data generation in large simulation system

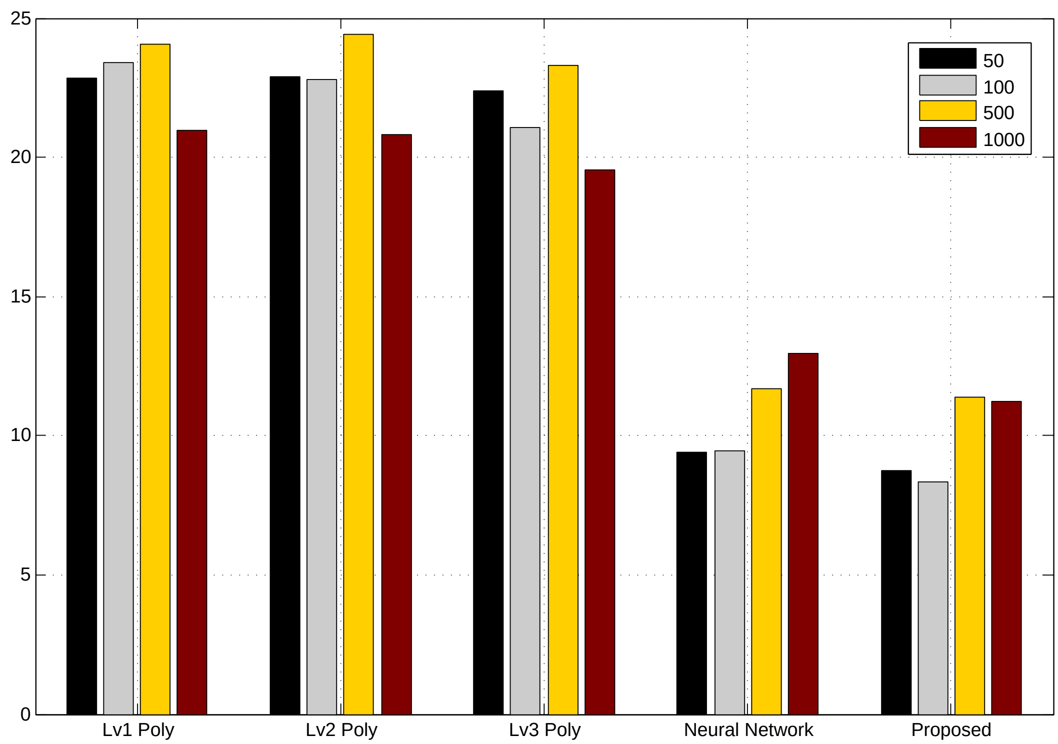

Figure 6.10: Error/Deviation Comparison among different network sizes

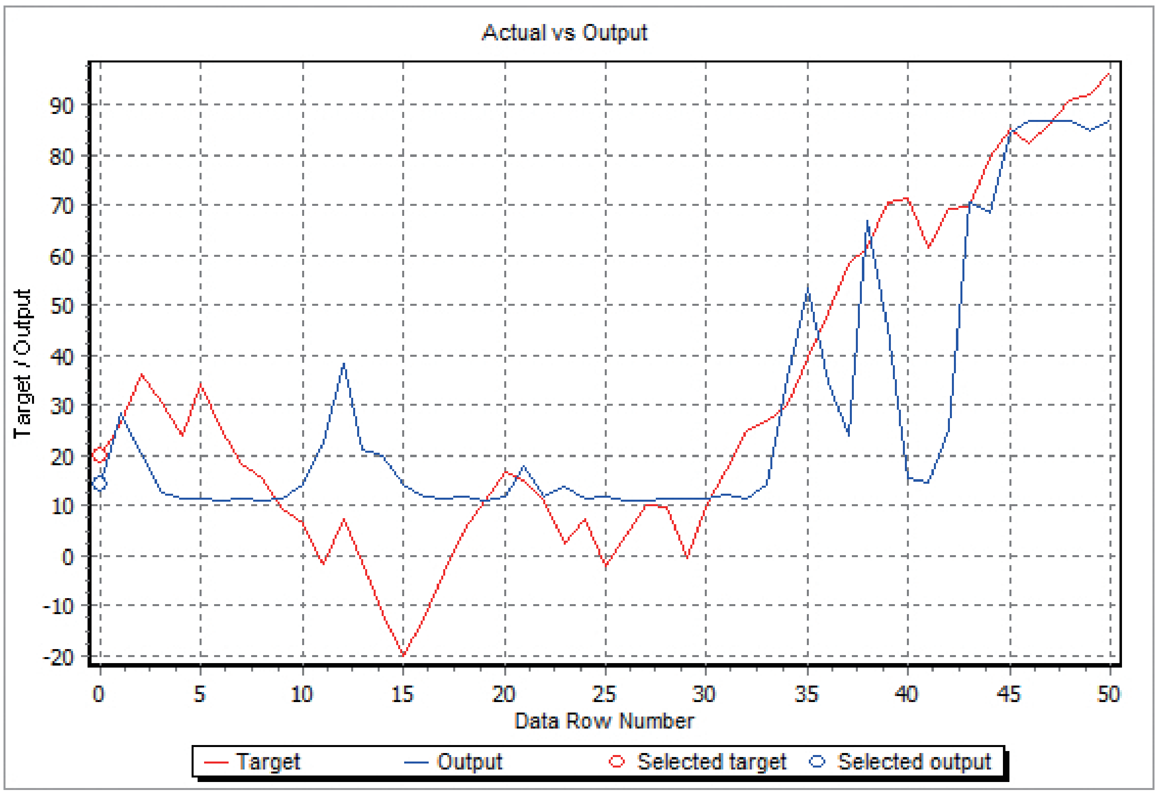

Figure 6.11: Prediction of Simple Neural Network

Chapter 7

Conclusion and Future Works

In this thesis, in order to explore the trajectory exposure problem in mobile networks, we studied location-aware mobile social networks for general media sharing and professional uses, by designing and implementing two platforms MoViShare and Scoop. we then proposed a mechanism to adaptively control the exposure of a mobile user's trajectory while offer useful information for location-based services. We further discussed the controlling of exposure rate, to specify which locations to expose. We also proposed MoRPTE that utilizes location information flexibly in the spectrum from zero to full knowledge, which makes sure no trajectory exposure rate would exceed $r\%$. We may later introduce and examine more flexible controls, for instance, user specific or heterogeneous demands.

Also we are planning to use other fitting methods, such as B-spline which is predefined by a number of de Boor points, to manipulate between predefined and dynamic controls of trajectory information exposure and recovery.

Bibliography

[1] airG. http://www.airg.com/.

[2] airG's Mobile Community. http://www.airg.com.

[3] Everyone is Talking About Mobile Social Networking. http://corp.airg.com/news/inNews/eMarketer05222008.htm.

[4] Helix Server. http://www.realnetworks.com/products/media_delivery.html.

[5] iPodtouch. http://www.apple.com/ipodtouch/.

[6] ISO/IEC 14496-12, ISO Base Media File Format.

[7] Mobile consumer briefing. The Mobile Marketing Association, http://goo.gl/BTYv4.

[8] MocoSpace. http://www.mocospace.com/.

[9] MocoSpace wiki. http://en.wikipedia.org/wiki/Mocospace.

[10] Ovi wiki. http://en.wikipedia.org/wiki/Ovi(Nokia).

[11] S60Web Runtime (WRT). http://wiki.forum.nokia.com.

[12] Trusterra. http://www.trusterra.com.

[13] Veeker. http://www.veeker.com.

[14] YouTube. http://www.youtube.com.

[15] YouTube extends mobile video sharing service. http://www.it.com.cn/f/news/081/24/541937.htm.

[16] YouTube offers mobile upload service. http://news.cnet.com/2100-1025_3-6070527.html.

[17] O. Abul, F. Bonchi, and M. Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *IEEE ICDE*, 2008.

[18] S. Akoush and A. Sameh. Mobile user movement prediction using bayesian learning for neural networks. In *IEEE IWCMC*, 2007.

[19] S. Akoush and A. Sameh. Movement prediction using bayesian learning for neural networks. In *IEEE ICSNC*, 2007.

[20] P. Appavoo and K. Khedo. Sencast: A scalable protocol for unicasting and multicasting in a large ad hoc emergency network. In *International Journal of Computer Science and Network Security*, 2008.

[21] W. Azmy, N. Gayar, A. Atiya, and H. El-Shishiny. Mlp, gaussian processes and negative correlation learning for time series prediction. In *Springer-Verlag MCS*, 2009.

[22] R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *IEEE ICDE*, 2005.

[23] B. Bellur, R.Ogier, and F. Templin. Topology broadcast based on reverse-path forwarding (tbrpf). In *RFC3684*.

[24] Matthew Caesar, Miguel Castro, Edmund B. Nightingale, Greg O'Shea, and Antony Rowstron. Virtual ring routing: network routing inspired by dhts. *ACM SIGCOMM*, 2006.

[25] A. Capone, I. Filippini, M. Garcia de la Fuente, and L. Pizziniaco. Sift: An efficient method for trajectory based forwarding. *IEEE ISWCS*, 2005.

[26] X. Cheng, C. Dale, and J. Liu. Statistics and Social Network of YouTube Video. In *Proceedings of IEEE IWQoS*, 2008.

[27] Y. Hu David B. Johnson and David A. Maltz. The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4. In *RFC 4728*.

[28] David Ellis, Eric Sommerlade, and Ian Reid. Modelling pedestrian trajectories with gaussian processes. In *IEEE International Workshop on Visual Surveillance*, 2009.

[29] P. Jacquet et al. Optimized link state routing protocol. In *RFC3626*.

[30] W. Gao and G. Cao. Fine-grained mobility characterization: Steady and transient state behaviors. In *ACM MobiCom*, 2010.

[31] J. Ghosh, M. Beal, H. Ngo, and C. Qiao. On profiling mobility and predicting locations of wireless users. In *ACM international workshop on Multi-hop ad hoc networks*, 2006.

[32] A. Gkoulalas-Divanis, V. Verykios, and M. Mokbel. Identifying unsafe routes for network-based trajectory privacy. In *SDM*, 2009.

[33] Zonggao Jia. Movishare: Building location-aware mobile social networks for video sharing. In *Thesis, Simon Fraser University*, 2009.

[34] N. Kamel, A. Atiya, N. Gayar, and H. El-Shishiny. An empirical comparison of machine learning models for time series forecasting. In *Econometrics Review*, 2009.

[35] B. Karp and H. Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. *ACM MobiCom*, 2000.

[36] M. Kim, D. Kotz, and S. Kim. Extracting a mobility model from real user traces in real world. In *IEEE INFOCOM*, 2006.

[37] E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In *Canadian Conference on Computational Geometry*, 1999.

[38] Ying Li, Ying Li, Tong Zhang, Tong Zhang, Daniel Tretter, and Daniel Tretter. An overview of video abstraction techniques. Technical report, Report HPL-2001-191, HP Laboratories, 2001.

[39] X. Lin, R. Lu, P. Ho, X. Shen, and Z. Cao. Asrpake: An anonymous secure routing protocol with authenticated key exchange for wireless ad hoc networks. In *IEEE ICC*, 2007.

[40] C. Liu and J. Wu. Efficient geometric routing in three dimensional ad hoc networks. In *IEEE INFOCOM*, 2009.

[41] C. Ma, D. Yau, N.K. Yip, and N. Rao. Privacy vulnerability of published anonymous mobility traces. In *ACM MobiCom*, 2010.

[42] L. Ma, Z. Jia, and J. Liu. Movishare: Building location-aware mobile social networks for video sharing. In *ACM MobiCom*, 2009.

[43] JungHwan Oh, Quan Wen, Jeongkyu Lee, and Sae Hwang. Video abstraction. *Video Data Management and Information Retrieval*, pages 321 – 346, 2004.

[44] C. Rasmussen and C. Williams. Gaussian Processes for Machine Learning. In *The MIT Press*, 2006.

[45] S. Steiniger, M. Neun, and A. Edwardes. Foundations of Location Based Services. *University of Zurich*.

[46] S. Subramanian, S. Shakkottai, and P. Gupta. Optimal geographic routing for wireless networks with near-arbitrary holes and traffic. In *IEEE INFOCOM*, 2008.

[47] X. Xiang, Z. Zhou, and X. Wang. Self-adaptive on demand geographic routing protocols for mobile ad-hoc networks. In *IEEE INFOCOM*, 2007.

[48] T. Xu and Ying Cai. Location cloaking for safety protection of ad hoc networks. In *IEEE INFOCOM 2009*, 2009.

[49] Shin-Jer Yang. Design issues and performance analysis for dsr routing with reclaim-based caching in manets. *International Journal of Network Management*, 2010.

[50] Yaling Yang Yujun Li and Xianliang Lu. Routing metric designs for greedy, face and combined-greedy-face routing. In *IEEE INFOCOM*, 2009.

[51] C. Zhang, X. Lin, P. Ho, X. Sun, and X. Zhan. Ppbr: Privacy-aware position-based routing in mobile ad hoc networks. In *IEEE MILCOM*, 2007.