An Indoor Trajectory Comparison Framework for Android Smartphones

Christos Laoudias†, Costandinos Costa‡, Demetrios Zeinalipour-Yazti‡ and Christos G. Panayiotou†
†KIOS Research Center for Intelligent Systems and Networks, University of Cyprus, Email: laoudias@ucy.ac.cy, christosp@ucy.ac.cy
‡Department of Computer Science, University of Cyprus, Email: cs07cc6@cs.ucy.ac.cy, dzeina@cs.ucy.ac.cy

***Abstract*—We present a powerful distributed framework for comparing trajectories and answering queries of the form: *"Report objects (i.e., trajectories) that follow a similar spatiotemporal motion to Q, where Q is some query trajectory"*. Each trajectory resides in its entirety *in-situ*, which is cheaper and more efficient for mobile devices. Our SmartTrace algorithm then deploys a search algorithm that exploits distributed similarity measures to derive the most relevant answers to Q quickly and efficiently. We extend this approach to enable trajectory similarity services indoors by using location dependent information, such as the magnitude of signals emitted by WLAN Access Points.**

I. INTRODUCTION

Nowadays due to the widespread deployment of smartphone devices featuring location capabilities there is a proliferation of innovative services, including Google Latitude, Foursquare, Gowalla, etc., that enable users to track the locations or places they and their social network have visited. In this direction advanced applications have emerged that enable users to record their trajectories, i.e. sequences of successive locations, and allow them to search for similar trajectories that other users or authorities have shared.

The notion of *similarity* captures the trajectories that differ only slightly from the given search query Q, e.g. *"Find whether there is a cycling route from the Metropolitan Museum of Art in Manhattan, through central park to the Juilliard School"* or *"Find if a new bus route is similar to the trajectories of K car users"*. There are already centralized trajectory search services such as GeoLife, GPS-Waypoints, ShareMyRoutes to perform this kind of querying. However, these services store user's trajectories on a centralized or cloud-like infrastructure. On the other hand, the techniques utilized by our SmartTrace algorithm are decentralized and maintain the data *in-situ*, i.e. on the smartphone that generated the data. The proposed scheme performs well both with respect to response time and energy, but also does so without ever revealing the complete user trajectories to the query processor.

The SmartTrace algorithm is very effective for trajectories collected outdoors that contain GPS locations. However, GPS signals are attenuated or blocked inside buildings, thus failing to provide accurate location information. An alternative solution for supporting trajectory similarity services in indoor environments is required. One option is to use an indoor positioning system [1], however this implies high costs both for the installation and maintenance of the relevant infrastructure, while scalability is also challenging. We address this issue by modifying our algorithm appropriately to exploit location dependent information, such as Received Signal Strength (RSS) readings from WLAN Access Points (AP), in order to conduct the similarity search. Experimental results indicate that this approach works quite well in practice.

II. THE SMARTTRACE ALGORITHM

Let $\{A_1, \ldots, A_m\}$ denote a collection of trajectories, where $A_i,\ i \leq m$ is defined as a sequence of l tuples $\{a_1, \ldots, a_l\}$ and each tuple $a_j,\ j \leq l$ is characterized by two spatial dimensions and one temporal dimension, i.e. (x_j, y_j, t_j). Given a query Q, itself expressed as a spatiotemporal trajectory, the objective is to retrieve the K most similar trajectories to Q, denoted as top-K trajectories. The query Q is initiated by some querying node QN or alternatively at some smartphone that propagates its Q towards QN. QN then disseminates Q to all active smartphone users in a pre-specified spatial boundary.

The SmartTrace algorithm [2], [3] relies on an *in-situ* data storage model, where location data is recorded locally on smartphones. When a query emerges, we collect a set of scores from participating smartphones and derive the answer intelligently based on these scores only. Our algorithm deploys an efficient top-K query processing algorithm that exploits distributed similarity measures, resilient to *spatial* and *temporal* noise, to derive the most relevant answers to Q quickly and efficiently in an iterative fashion, without ever unveiling any of the trajectories to QN.

A. Application in Indoor Environments

In the indoor scenario, a trajectory $A_i,\ i \leq m$, denoted as RSS trajectory, is a sequence of l multidimensional tuples $\{a_1, \ldots, a_l\}$. Each tuple a_j is characterized by n spatial dimensions and one temporal dimension, i.e. $(S_j^1, S_j^2, \ldots, S_j^n, t_j),\ j \leq l$, where t_j denotes the timestamp on which the tuple was recorded and $S_j^k,\ k \leq n,\ j \leq l$ is the RSS value from the k-th AP. The similarity between two RSS trajectories can now be computed using the distributed similarity measure on individual dimensions and combining them linearly. We found that this technique works particularly well, as this is documented in the following example which is part of our experimental evaluation inside an office building.

Example: The experimentation area is roughly 560m^2 containing office rooms, open plan workstations and meeting rooms connected with corridors. We have installed 3 WLAN APs, which cover the whole area adequately. We collected 5 RSS trajectories along with the respective location coordinates (to plot the trajectories on the flooplan map for verification). The average length of the trajectories is 30 tuples and each tuple contains the MAC addresses and RSS levels of all 3 APs. We then conducted a top-2 query for one of the RSS trajectories (Q) and the trajectories T_2 and T_3 were correctly identified as the top-2 answers; see Fig. 1.

III. Android Implementation

We have developed a prototype system that realizes the proposed framework; see Fig. 2. Our client-side software is developed around the Google Map API and its installation APK package has a size of 510KB. Our code is written in JAVA and consists of approximately 4,500 lines-of-code (LOC). In particular, our server-code uses ∼1,500 LOC and runs over JDK 6 and Ubuntu Linux, while our client-code uses ∼2,500 LOC plus ∼250 lines of XML elements and runs over the Dalvik VM.

The client-side GUI allows a user to configure a wide range of parameters, such as K, and then query other devices by example. In the case of GPS trajectories the user may plot and iterate through the responses using various presentation functions (Fig. 2, Left), while for RSS trajectories the identities of the users with the highest similarity scores are returned (Fig. 2, Middle). Our prototype finally enables a user to switch between Offline and Online Mode (Fig. 2, Right) in order to simulate movement (i.e., the trajectory file can be stored on the device, as opposed to be collected in real time). This feature helps with playing back recorded scenarios.

IV. Demonstration Settings

For the demo at the conference venue we will use 5 HTC Desire smartphone devices running Android 2.1 (Eclair). These devices are equipped with a Qualcomm Snapdragon QSD 8250 1 GHz processor and provide 512 MB of Flash ROM, as well as 512 MB of RAM. The server will be running on an Asus eeePC netbook with Ubuntu Linux, while the HTC devices will connect to the server through the available WLAN hotspot. We will use a projector along with a display export utility to present the interactions on a smartphone directly on a wall, so that attendees will be able to follow the interactions.

A. Demonstration Scenarios

Interactive: Our first objective is to demonstrate the effectiveness of the proposed algorithm in identifying users that are moving similarly indoors when RSS trajectories are used. We will hand out the 5 smartphones to selected participants and ask them to start moving around for a few minutes, by enabling the "Online" (WiFi) mode from within the GUI (Fig. 2, Right). In this mode the user's RSS trajectory will be logged to the flash card. Two arbitrary users **X** and **Y** will be asked to move within proximity during the largest interval of their walk. At the end we will connect **X**'s device, through USB, to a laptop that will project **X**'s screen on a wall and conduct a top-1 query through **X**. This should reveal that the RSS trajectory of user **Y** has the highest similarity (Fig. 2, Middle). This feature will enable several interesting applications in the future; for instance, an attendee might be able to determine other attendees that have participated in common sessions, in order to initiate new discussions and collaborations.

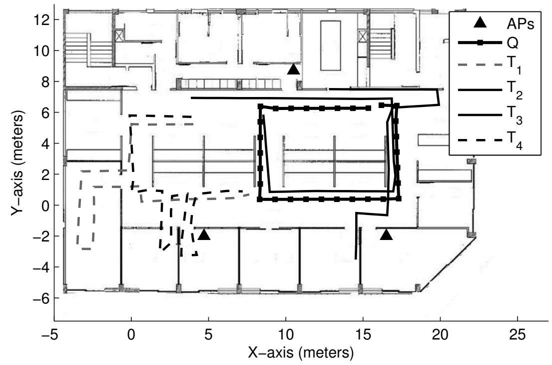

Fig. 1. Application in an Indoor Environment using RSS Trajectories.

Fig. 2. **Screenshots of the prototype system** Left: The *matched* GPS trajectory displayed using Google Maps; Middle: A Query Response Message for RSS trajectories; Right: Parameter configuration and mode selection panel.

Trace-driven: We will also demonstrate how our algorithm would apply outdoors by allowing attendees to select among a number of available GPS traces (collected a priori and stored on the smartphones). Then, a top-K query will be initiated on one device and the K most similar trajectories will be displayed along with the query trajectory for verification using Google Maps (Fig. 2, Left). Our tests have shown that such an action is taking only a few seconds, even for large trajectories, so we expect this feature to be of particular interest.

Acknowledgment

This work is supported by the Cyprus Research Promotion Foundation and in part by the third author's Startup Grant, funded by the University of Cyprus between 2010-2011.

References

[1] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 1, pp. 13–32, 2009.

[2] Zeinalipour-Yazti D., Laoudias C., Andreou M., Gunopulos D., "Disclosure-free GPS Trace Search in Smartphone Networks," *IEEE MDM*, 2011.

[3] Costa C., Laoudias C., Zeinalipour-Yazti D., Gunopulos, D., "SmartTrace: Finding similar trajectories in smartphone networks without disclosing the traces," *IEEE ICDE*, 2011, pp.1288–1291.