# A Navigation engine for ubiquitous computing environments

Dikaios Papadogkonas, George Roussos, Mark Levene
School of Computer Science and Information Systems
Birkbeck College, University of London
London WC1E 7HX, U.K.
dikaios, george, mark @dcs.bbk.ac.uk

May 22, 2006

## 1 Abstract

The problem of navigation and wayfinding has developed while moving from physical to virtual spaces. Now with ubiquitous environments coming to connect those two spaces new navigation requirements appear. We propose a navigation engine based on the suffix tree data structure for identifying navigation patterns and it's performance is discussed in 2 different test cases.

## 2 Introduction

Because of the human desire of exploration, navigation and wayfinding has been an issue since the early stages of the human civilaization. Compasses, maps, signs and other equipment where discovered to satisfy this need. Later on, the world wide web created a new, virtual space. The human desire for knowledge created a new navigation problem, this time related to the huge amount of data stored and the need to get to as fast as possible to the desired information. Now that ubiquitous computing managed to connect physical and virtual space, navigation in environments where both spaces coexists have raised new challenged, because people are unable to cope with the huge amount of information and because of constructional faults in the physical space's navigation system.

Here, we propose a navigation engine used to give as a trace of a user's experience, an understanding of the area usage and to help people navigate in unfamiliar environments by previously recorded user trails. We define the notion of landmark as the position of significant object in a landscape. In the ubiquitous computing environment landmarks can be identified by a variety of things captured by sensors, such as physical objects, sound, smell or vision. This engine consists of a probabilistic tree data structure which represent the sequence of visits at different landmarks, a framework for the definition of metrics used to calculate the best route under different circumstances during navigation; and associated mechanisms that can calculate such routes efficiently using the constructed network representation.

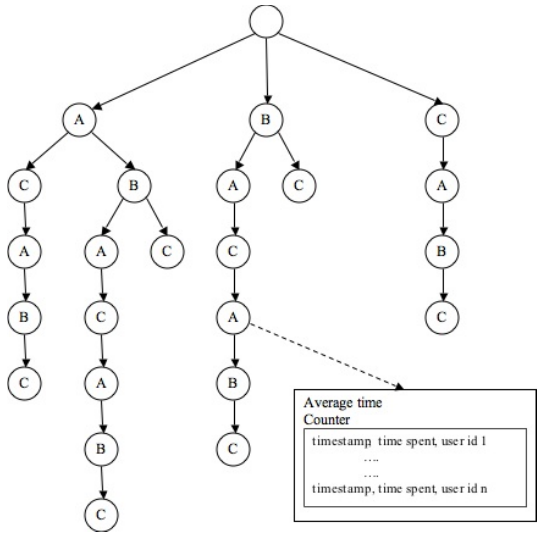


Figure 1: Suffix tree for the ABACABC trail with an example of the information associated with a node.

## 3 Navigation Engine

As a basis for our tree data structure we use the suffix tree. The suffix tree is a radix tree proposed by [Weiner] for solving the longest common substring problem by inserting in the tree all the suffixes of a given string. In our case user trails are represented as a sequence of characters where each character represents a landmark. Each node of the tree contains additional information about the position of the landmark in the trail, specifically the amount of visits the landmark had during a specific trail (the amount of people followed the trail so far), the average time spent on the trail so far, the date and time of each interaction along with the user id number who did this interaction. All user trails are inserted in the tree adding their additional information to the overlapping nodes. This way we can get very basic information about the amount of overall visits per landmark, or average time spent on a landmark just by visiting the first level nodes. The main disadvantage of the suffix tree is that it takes too much memory. So the information related to individual interaction with landmarks are stored in a separate SQL database.

From the information stored in the nodes, we are able to extract specific user trails, we can extract trails that occurred during specific periods of time. We can then measure the importance of the trail by different measures, like the frequency of a trail, the time spent on the trail the relevance of the landmarks in a trail, or all these measures combined and weighted. We can also get trails that passed through a set of landmarks, trails that started and ended at the same landmark, e.t.c. The type of queries are dependent to the investigating data.

## 4 Case Studies

In order to evaluate our navigation engine we used two case studies. The first experiment consisted from data collected in the Computer Laboratory of Cambridge [chaintreau]. These data are trails collected from a number of imote devices embedded in artifacts or carried by

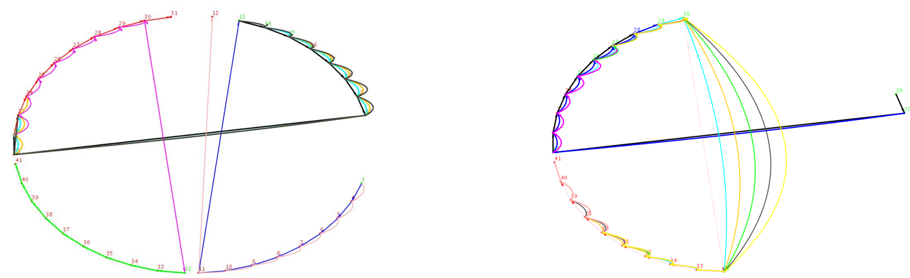

Figure 2: imotes data set, 10 most frequent trails of size 10 with 7 different landmarks and 10 ten first trails by time spent theret of size 10 with 7 different landmarks

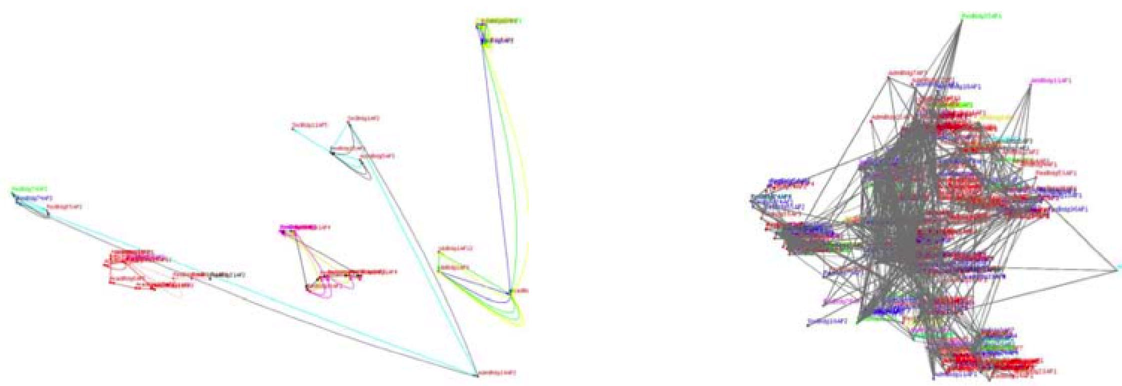

Figure 3: Dartmouth data set, 10 best trails by time and frequency combine of size 10 with 7 different landmarks (left) and the entire Dartmouth data set (right)

users. In Figure 2 we can see on the right the 10 most frequent trails, of size 10 where the user interacted with at least 7 different landmarks. Because the data where anonymised and the landmarks where not of fixed position we had to represent their location at the perimeter of a circle.

For the second case study we used data collected from the wireless network at Dartmouth University's campus which was collected as a task of the Crawdad project [kotz]. The data consist of a series of user connections at the wireless access points. From this data set we are interested to identify the use of the wireless network and patterns of the movement of the network's users. In Figure 3 we represent the landmarks depending on the physical location of the access points and this time we present (left) the 10 best trails by combining time spent on the trails and the frequency of the trail, and at the right the entire data-set of the movement in the campus (right).

## 5 Conclusions and Future Work

We presented a navigation engine able to assist user navigation by previously recorded user trails and able to give an understanding of the environments usage through previously recorded user data. The engine has been tested in two different test cases and it performed satisfactorily. Further testing is required to evaluate the performance of the engine for large data sets. Also a user interface and a query systems is required to take full advantage of the capabilities of the engine.