Design of a Recommender System for Mobile Tourism Multimedia Selection

Robert P. Biuk-Aghai Simon Fong Yain-Whar Si
Department of Computer and Information Science
Faculty of Science and Technology
University of Macau
Taipa, Macau
{robertb, ccfong, fstasp}@umac.mo

Abstract—**Applications that deliver multimedia content to and display such content on mobile devices have become increasingly common in recent years. When faced with a large amount of content, unfamiliar users can make use of each others' recommendations, through recommender systems, to find content of interest to them. As a case in point we present the design of a recommender system that can be used by tourists to request a travel itinerary, and subsequently browse multimedia content for each recommended tourist spot. The techniques combined in our recommender system include genetic algorithms and fuzzy logic. Recommendations are chosen to match a user profile based on a user's personal preferences. Our system design has wide applicability in multimedia systems where the user requires assistance in content selection.**

Keywords-recommender system; tourist application; multimedia; genetic algorithm; fuzzy logic

I. INTRODUCTION

Recent advances in ubiquitous computing and wireless networking technologies facilitate efficient delivery of rich multimedia content to mobile devices. With the availability of high speed communication networks, handheld devices are now able to perform tasks ranging from viewing of on-line video clips for personal entertainment to conducting financial transactions on the Internet. From a large number of applications offered by ubiquitous computing, recommender systems for the tourism industry [1] are highly successful since in addition to assisting users in making scheduling decisions, they also allow mobile users to re-plan their itineraries and retrieve revised recommendations during their trips. Such capability is crucial since a traveler may wish to obtain a new recommendation depending on the latest situation/availability of the tourist spots. In addition, these recommendations can be combined with rich multimedia contents such as photographs, audio and video clips, and maps.

In this research, we have developed a recommender system for visiting tourist attractions in Macau. This system is an extension of the MacauMap system [2][3] which is a highly popular handheld and web-based tourist map guide system for Macau. The recommender system aims to provide a suggested travel itinerary to a tourist based on input about the tourist's preferences, official rating of each tourist spot, and all user feedback for each spot. Tourists use their mobile device such as mobile phone or PDA to indicate their preferences for different types of tourist spots (temples, churches, museums, entertainment, nature, etc.), as well as their desired begin and end time and place. These requirements are transmitted as a call to a web service on a recommender server which generates a suitable travel itinerary. A genetic algorithm is used for determining the itinerary. Then a fuzzy logic [4] module calculates the suitable stay time for each spot based on the user's preferences. A schedule is completed for the itinerary including travel times by bus or on foot which are calculated for each pair of spots in the itinerary. The itinerary is sent back to the mobile device. The tourist may also modify an itinerary by reordering, adding or removing spots, which recalculates schedule and travel times and is performed locally on the device without requiring another server access.

The main contributions of our research work are two-fold; from the theoretical standpoint, we contribute to the design and development of a novel recommender system based on genetic algorithm and fuzzy logic. In particular, our proposed genetic algorithm-based tourist spot selection approach addresses the problem of finding a near optimal solution in a large solution space while maintaining acceptable system performance. In addition, the fuzzy logic-based travel/stay time calculation algorithm addresses the problem of capturing fuzziness in users' preferences. From the practical standpoint, our research opens the door to the development of mechanisms for assisting both mobile and stationary users in retrieving multimedia content based on personal preferences in various domains.

The remainder of this paper is organized as follows. Section II briefly reviews related work. Section III then introduces our travel recommender system in overview. Section IV presents the genetic algorithm we designed, and Section V discusses our fuzzy logic itinerary scheduling method. Finally, Section VI presents conclusions.

II. RELATED WORK

Recommender systems, which have been widely exploited in e-commerce, suggesting products and services to users [5], have recently found application in tourist guides. Some classical recommender system technologies [6], for travel and tourism, include expert advice platforms such as TripMatcher

and VacationCoach [1] and Me-Print [2]. By modeling the counseling interactivity provided by traditional travel agents, these recommenders offer online advice to assist users in planning their possible holiday itineraries. Technically, such recommenders among other similar systems use a content-based approach together with some proprietary languages to capture their users' expressed needs, constraints and preferences as explicit attributes. The systems then match the user preferences in a catalog of predefined itineraries. VacationCoach employs a profiling technique which demands the users to manually classify themselves into one of the profiles (e.g. "urban shopper", "nature lover" etc). Thereafter, TripHop's matching engine adjusts the importance of attributes which the user did not explicitly mention; it then combines statistics on past user queries with a prediction computed as a weighted average of importance assigned by similar users [7]. These approaches however have limited performance in terms of matching accuracy, flexibility and usability [8].

With the advent of ubiquitous computing, recommender systems evolved from being solely web-based to reaching mobile devices [9]. The second generation of mobile guides are equipped with sophisticated features such as personalization, recommendation, context-awareness, etc. A number of emerging commercial products include Do Me London [3], Mobile Travel Buddy[4] and Mobile Travel Guide[5].

Many research prototypes of mobile guides that are capable of delivering multimedia content were published in the academic literature; some typical ones are introduced here: MOBYREC [10] is a critique-based mobile recommender that allows users to specify preferences for places of interest, restaurants and hotels. Over time the recommendation improves, though many users' inputs are required. Similarly CRUMPET [11] learns user preferences over time and suggests tours with maps and other relevant information. COMPASS [12] is a mobile guiding service that works by first receiving the tourist's stated goal and preferences, and then filters out the items to be viewed on a map interface. In addition to tour route recommendation, GUIDE [13] can also access ticket reservation services. MAGITTI [14] features a scalable client-server architecture that has a load balancing facility.

Extensive surveys were conducted to evaluate and compare different mobile guides. The surveys focus on the unique features and their historical influence on the development of mobile guides [15], application designs [16] and some taxonomy of services [17].

From our observation, most recommenders take context information into account, such as the hours and busy times of requested attractions and the current location of the user. They have somewhat similar core features and functions, but differ in user interface and usage. One underlying content-based recommendation technique for most of the recommenders is case-based reasoning. In contrast, our novel recommender design taps on machine learning algorithms, specifically genetic algorithms, as well as fuzzy logic for optimal and flexible solutions respectively.

[1] www.ski-europe.com
[2] www.travelocity.com
[3] www.do-me.info
[4] www.mobiletravelbuddy.com
[5] mobiltravelguide.howstuffworks.com

III. System Overview

We have designed and implemented a recommender system that helps select tourist spots and associated multimedia data. The challenge is finding those tourist spots which match a user's interest. Once a selection has been made and presented to the user, the user can display the multimedia content of each spot to help in their decision whether to actually visit that spot, and can make adjustments to the suggested travel itinerary if needed. We have based this recommender system on MacauMap [2], a mobile tourist guide and map system that we developed several years ago. MacauMap is available for free use [3] and has been downloaded over 300,000 times over the past five years since its initial release in 2003. MacauMap is available on the Windows Mobile, PalmOS, Symbian OS Series 60 and Symbian OS Series UIQ mobile platforms, as well as through a web interface. It displays a tourist map of the entire Macau territory and its tourist spots (some 1200 streets and 120 tourist spots).

Our extended MacauMap + recommender system is implemented in a client-server system architecture. On the server side is a recommender engine which performs two main functions: (1) selecting a set of suitable tourist spots given the requirements of the user, and (2) generating an optimal schedule of the selected tourist spots consisting of suitable stay times and travel method and time between spots. On the client side is the MacauMap mobile application with which the user interacts and in which the travel itinerary is visualized. The client communicates with the server through a web server interface that the server provides: a SOAP (Simple Object Access Protocol) message containing an itinerary request is sent from the client to the server, and the server sends a SOAP message containing its reply back to the client. Because SOAP is simple, versatile and platform independent, it allows us to support multiple different client applications besides the mobile one presented here. Most importantly, it allows for easier communication through proxies and firewalls than previous remote execution technology for mobile recommender applications. The operating mode is illustrated in Figure 1.

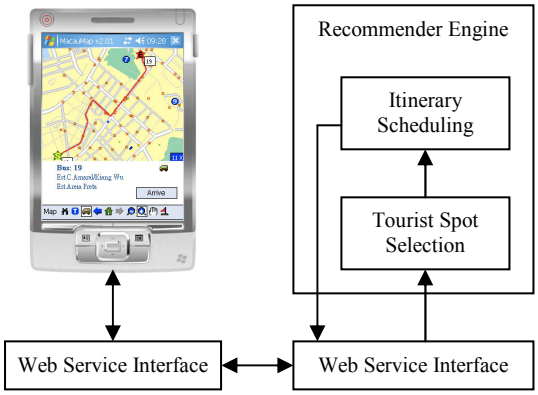

Figure 1 MacauMap recommender system operating mode

IV. GENETIC ALGORITHM-BASED RECOMMENDATION

In preparing a visiting itinerary for a location, the main challenge is to select spots which match the interest of the visitor *without* requiring the visitor to give an evaluation of their interest in each spot. In a tourist application, such as ours, we can not assume that the user is familiar with any of the spots they may like to visit, but may only be aware of some general preferences, such as a preference to visit parks, or a preference not to visit museums.

A. Recommender System

The characteristics of this situation make the use of a recommender system appropriate. A recommender system aggregates evaluations from many users on the entities (products, people, places etc.) that are to be selected, and produces a recommendation of one or more of these entities that are considered to match the user's requirements [18]. Recommender systems, also called *collaborative filtering systems*, have their origins in the Tapestry system of the early 1990s [19]. Applying the recommender system idea to tourist spot selection, feedback from other tourists on spots visited can be used to filter spots based on their popularity ranking, then combined with the user's visit preferences in order to recommend a travel itinerary.

Following the categorization of [18], the design of our recommender system provides a numeric recommendation (a ranked list of recommended tourist spots); uses both explicit and implicit source recommendations; makes anonymous recommendations; performs aggregations using personalized weighting; and uses the recommendations for selection and filtering.

As mentioned, the feedback from other users constitutes one of the main inputs for making a recommendation. In order to match other users' feedback to a given user's recommendation request, it is necessary to know this user's visit preferences. For this purpose we obtain a list of preferences from the user a-priori: users are presented with a list of tourist spot categories (historical, cultural, entertainment, nature, etc.), and asked to give each a rating on a five-point scale. This constitutes a user profile which we then use to calculate a matching rating for a certain tourist spot. After a user has finished visiting the tourist spots in a given tour itinerary, a-posteriori user feedback is collected and added to the recommender system database to provide the input for future recommendations. A preset "official" rating is also given for each tourist spot to classify the spots' visit value, and a set of random recommendations are initially generated based on these ratings to cold-start our recommender engine.

In almost all cases, preparing a travel itinerary is not a simple selection of one tourist spot that best matches the user's preferences, as the tourist will want to travel to *several* spots during a visit. Moreover, a varied travel itinerary with tourist spots from different categories is usually preferable to one with spots from only one category. Thus, preparing a travel itinerary is also not simply a case of selecting the best *n* spots from the top-rated category. A medium-rated spot in the user's top rated category may be of similar interest to a highly-rated spot in the second rated category. Finally, if a tourist has already recently visited a spot, however highly rated it may be, on the next visit to this city it may be more appropriate not to select this same spot as a spot loses its visit appeal after it has been visited. Thus the factors to take into consideration in the itinerary preparation should include:

- User's stated preferences
- User's visit history
- Official spot rating
- All users' feedback ratings

The difficulty is how to combine these factors on the potentially large amount of data – over a hundred tourist spots in the case of Macau, each with potentially very many (thousands) of user feedback ratings – to produce an optimal, or at least near-optimal, travel itinerary.

Given a potentially large volume of user feedback ratings available in the system, collaborative filtering in this tourist recommender application is suitable for making automatic predictions about the interests of a user by collecting feedback information from many users.

By sets of extensive performance evaluation experiments conducted by Breese et al. from Microsoft Research [20], Bayesian networks have been shown to have smaller memory requirements and allow for faster predictions than a memory-based technique such as collaborative filtering. However, the Bayesian methods examined in [20] require a learning phase that can take up to several hours and results in a lag before changed behavior is reflected in recommendations. Hence, this extra phase required by Bayesian networks makes it an undesirable technique for an interactive application such as the one presented here.

B. Genetic Algorithm

A genetic algorithm is a search technique aimed at finding optimal solutions in a large search space, and is inspired by nature, using concepts of inheritance, mutation, selection and crossover [21]. We propose that the large amount of data and the multiple factors to be used in making an itinerary selection in our case make the problem well-suited for being solved by a genetic algorithm.

The basic unit in a genetic algorithm is termed a *chromosome*. The collection of all chromosomes is termed the *population*. Each chromosome has an associated *fitness*, which is a measure of how well it matches the target requirement.

Our chromosomes consist of a pool of tourist spot ids. The number of spots in the pool depends on the time available for travel, which is an input given by the user.

In outline, our method has following steps:

1. Initialize population
2. Evaluate fitness of each chromosome in population
3. REPEAT
   i. Select parents for crossover operation
   ii. Mutate new generation

iii. Evaluate fitness of each new chromosome

iv. Replace worst part of old population with new generation

4. UNTIL termination condition reached

Details of this are explained next.

We initialize the population by creating 50 chromosomes with several randomly chosen tourist spot ids in each. We then evaluate the fitness of each chromosome according to a fitness function, discussed below. Here is an example of a chromosome:

{1, 8, 23, 91, 112}

This example chromosome contains five tourist spot ids. Each id is the number of a tourist spot in the tourist spot database. The number of tourist spots (five in this example) is chosen so as to fill the available travel time, plus leave a few extra spots as backup in case the user wishes to add more spots to the itinerary or finishes the itinerary in less time than planned (discussed further below).

1) Fitness function

The choice of a suitable fitness function is crucial in the design of a genetic algorithm-based solution. To accommodate all the itinerary preparation factors identified earlier, we define a set of three fitness functions that are combined to calculate the total fitness value of a given chromosome.

The *base fitness* takes only the official spot rating and user preferences into account. These are very simply calculated from the official spot rating, weighted with the user's stated preference of the category of the spot, as shown in (1) below:

$$Base\ Fitness: \sum_{i=1}^{N}(R_i * P_i) \quad (1)$$

where:

N: number of spots in the pool; range: (0, ∞)

R_i: official rating of spot i; range: (0, 1]

P_i: user preference of category of spot i; range: (0, 1]

The range of this base fitness function is (0, N].

The second fitness function takes all users' feedback ratings into account, in addition to the given user's preferences. We average the ratings given by all users for the given spot, and weight this average rating with the user's stated preference of the category of that spot. All of these weighted averages are summed up to become the additional fitness function 1, as shown in (2):

$$Addition\ Fitness\ 1: \sum_{i=1}^{N}[Average(F_i) * P_i] \quad (2)$$

where:

N: number of spots in the pool; range: (0, ∞)

F_i: spot feedback of all users of spot i; range: (0, 1]

P_i: user preference of category of spot i; range: (0, 1]

The range of the first additional fitness function is (0, N].

The only remaining factor to take into account is the user's visit history. This calculation is slightly more complicated. We calculate this as a weighted time factor, with months as the time unit and considering the number of visits made, and including both user preferences and all users' feedback, as shown in (3):

$$Addition\ Fitness\ 2: \sum_{i=1}^{N}[1 - \frac{D_i}{T_i * M_i} * (1 - P_i) * (1 - F_i)] \quad (3)$$

where:

N: number of spots in the pool; range: (0, ∞)

D_i: time discount rate of spot i; range: (0, 1]

T_i: total number of visits of spot i; range: (0, ∞)

M_i: number of months since previous visit of spot i; range: (0, ∞)

P_i: user preference of category of spot i; range: (0, 1]

F_i: spot feedback of all users of spot i; range: (0, 1]

The range of the second additional fitness function is (0, N].

The three fitness functions are finally combined into a total fitness value according to (4):

$$Total\ Fitness: \ Base\ Fitness + \frac{(Addition1 + Addition2)}{N} \quad (4)$$

where:

N: number of spots in the pool; range: (0, ∞)

The range of the total fitness function is (0, N + 2].

2) Selection operator

Once fitness of each chromosome has been computed, two chromosomes should be chosen to be parents for the next generation. There are many different selection operators. We have experimented with three of them: best selection, random selection and roulette wheel selection (also called fitness proportionate selection). Best selection tends to choose the same chromosomes, and may result in a next generation with a worse fit. Random selection overcomes these problems but does not utilize the fitness value of each chromosome. Roulette wheel selection, which uses the fitness value of each chromosome while still including a degree of randomness, turned out to produce the best results so our implementation uses this selection operator.

In roulette wheel selection, the fitness values of all chromosomes are initially summed up and normalized to fall in

the range [0, 1]. Then a roulette array is created where each chromosome has one slot with a value corresponding to its normalized fitness. A random real number in the range [0, 1] is generated and the chromosome closest to that value as a ceiling is selected. This process is performed twice to select two parents.

3) Crossover operator

After the selection operator has chosen two parents, the crossover operator creates a new generation chromosome from the parents. For each value (i.e. tourist spot id) in the new generation's chromosome it needs to choose which parent to copy the value from. We initialize a random bit array with the length equal to the number of values in the longer of the two parents. Then for each bit in the array, if the value is 1 copy the corresponding value from parent 1, else copy it from parent 2, unless that value already exists in the child in which case it is skipped.

As an example, consider that parent 1 has the chromosome {1, 8, 23, 91, 112} and parent 2 has the chromosome {6, 20, 42, 66, 91, 120}. The longer of the two is parent 2 with six values, so we generate a random bit array of length six, for example [101100]. Thus the child chromosome will be {1, 20, 23, 91, 120}. Note that in this example the duplicate value 91 corresponding to bit array position 5 was skipped.

4) Mutation operator

In order to allow for variation that could generate a more optimal solution, the result from the crossover operation should be mutated in a small number of cases. Mutation is usually a very small probability, in our case we found that a probability of 0.01% worked well.

The mutation operator initially generates a random bit array whose length is equal to the number of values in the new chromosome. Then it processes the chromosome: if the corresponding slot in the random bit array contains a 1, the value of the chromosome is mutated to a different, random value, else it is kept.

For example, if the child chromosome is {1, 20, 23, 91, 120} and this chromosome has been chosen for mutation (i.e. it is one of the 0.01% of chromosomes to be mutated), a random bit array of length five is generated, for example [10010]. Thus the chromosome should be mutated in its first and fourth values, for example to the new chromosome {7, 20, 23, 60, 120}.

5) Replacement policy

The replacement policy decides whether to add the new chromosome to the population or to discard it. If added, it will replace one of the existing chromosomes.

Firstly, we determine the fitness value of the new chromosome. Then we find the chromosome with the lowest fitness value in the existing population. If its fitness value is lower than that of the new chromosome, it is discarded and the new chromosome is added to the population in its place, else the new chromosome is discarded. This policy ensures that the population improves with each new generation, or at least does not degenerate.

6) Termination condition

The process of producing new generations is repeated until the termination condition is reached. Here a tradeoff between performance and solution quality has to be made. We have experimented with the number of rounds made and finally settled on terminating when the top chromosome's fitness value is stable over 500 rounds, or a maximum of 700 rounds has been reached. That is, between 500 and 700 rounds will be run. Generation of a recommended itinerary constitutes the bulk of server processing. Our current implementation, on a standard PC, requires under one second to produce a recommendation, which is acceptable for online use. A higher-end server configuration can ensure that this performance level can be maintained even on high-traffic websites.

The final output of the genetic algorithm is the chromosome with the highest fitness value. The chromosome contains a number of tourist spot ids which are used to prepare a travel schedule, as explained in the next section.

V. ITINERARY SCHEDULING

The output of the genetic algorithm is simply a list of tourist spot ids. However, in order to prepare a travel schedule from these spot ids, the following tasks need to be performed:

- Determine the means of transport (walk or bus) and travel time from each tourist spot to each other tourist spot;
- Determine the optimal sequence for visiting the tourist spots;
- Determine the suitable visit time at each spot, based on the preferences of the user;
- Determine times to insert meal breaks (if the visit extends over a meal time).

We consider public buses to be the main means of transport, or walking if the distance is short. On the server is a database of bus route information which we use to determine the bus number and route length. Given an average travel speed we can calculate the travel time by bus. To this we add the travel time on foot from one tourist spot to the start bus stop, and from the destination bus stop to the second tourist spot. We then compare this travel time to that of walking the entire distance and choose travel on foot as the means of transport if its travel time is shorter than that of bus travel.

The optimal travel sequence for the tourist spots is determined by using the A* search algorithm [22], a standard shortest path algorithm, where travel time is used as the distance value between tourist spots.

The main challenge in scheduling is to determine the suitable visit time for each tourist spot. For example, a tourist with great interest in visiting museums may spend several hours visiting a museum, whereas another visitor with only moderate interest in museums may only wish to spend one hour in the same museum. Therefore we pre-define different visit times for each tourist spot, and use fuzzy logic [4] to determine the suitable stay time. The inputs are user preferences and user feedback which are mapped to low, medium and high value fuzzy sets. Each tourist spot also has three fuzzy sets

corresponding to a short, medium and long visit time. Then depending on the fuzzy value of user preferences and user feedback, an output fuzzy value is selected. Defuzzification, using a centre of gravity method, returns a real-numbered visit time value.

Using the travel sequence, travel time and visit time, an initial itinerary schedule is prepared. Finally, meal break times are inserted into this initial schedule according to user pre-defined meal times, resulting in the final itinerary schedule.

Once the final schedule has been transmitted to the mobile device, the user can browse the schedule and play the multimedia content associated with each tourist spot, which resides on the device. If the user wishes to change the schedule, several types of changes are possible offline on the mobile device, i.e. without requiring another connection to the server: the order of spots can be re-arranged; spots can be removed from the itinerary; and spots from the backup list can be added to the itinerary. In all cases all schedule times are automatically recalculated, as all travel times for all pairs of spots are stored on the mobile device as part of the itinerary schedule. The backup list is usually set to be equally long to the main list of spots so that the user has sufficient choice in replacing unwanted tourist spots. This allows schedule flexibility while avoiding unnecessary server access.

VI. CONCLUSIONS

This paper has illustrated the design of a recommender system for mobile multimedia selection. The recommendations on travel itineraries are formulated with geographic maps, tourist spot images, explanatory text, and short video clips. The recommender system employs a genetic algorithm for generating travel itineraries and a fuzzy-logic based module for calculating visit/stay times for each stop of the entire trip. The system also takes into account user preferences and feedback from previous recommendations to other users.

The presented design is suitable in any situation in which users are confronted with either unknown content or overwhelming amounts of multimedia and other kinds of information, and require assistance in choosing content that matches their own interests and preferences. Examples include location-based systems that allow the user to view multimedia information about nearby shopping or dining locations; cinema guide systems that assist the mobile user in selecting a movie to watch; and others.

ACKNOWLEDGMENT

We gratefully acknowledge the work of our students Witman Chan, Peter Fong and Nick Wong from the Faculty of Science and Technology, University of Macau, on implementing the MacauMap recommender system.

REFERENCES

[1] D. R. Fesenmaier, (Editor). Destination Recommendation Systems: Behavioural Foundations and Applications, CABI Publishing, 2006.

[2] R. P. Biuk-Aghai, "MacauMap: Next generation mobile travelling assistant", in Proceedings of Map Asia 2004, Beijing, China, 26-29 August 2004.

[3] MacauMap, http://www.macautourism.gov.mo/macaumap/en/index.php

[4] L. Zadeh, "Outline of a new approach to the analysis of complex system and decision process", IEEE Transactions on Systems, Man, and Cybernetics, SMC-3(1), pp. 28-44, 1973.

[5] J.B. Schafer, J.A. Konstan, and J. Riedl, "Ecommerce recommendation applications", Data Mining and Knowledge Discovery, vol. 5, nos. 1–2, pp. 115–153, January–April 2001.

[6] H. Werthner and S. Klein, Information Technology and Tourism—A Challenging Relationship, Springer-Verlag, New York, 1999.

[7] J. Delgado and R. Davidson, "Knowledge bases and user profiling in travel and hospitality recommender systems", in K. Woeber, A. Frew, and M. Hitz, eds.: Proc. 9th Int'l Conf. Information and Comm. Technologies in Tourism (ENTER 2002), pp. 1–16, Springer-Verlag, Heidelberg, Germany, 2002.

[8] K. Swearingen and R. Sinha, "Beyond algorithms: An HCI perspective on recommender systems", in Proc. Recommender Systems: Papers from the 2001 ACM SIGIR Workshop, 2001, available online at: http://cs.oregonstate.edu/~herlock/rsw2001

[9] B. Rao, L. Minakakis. "Evolution of mobile location-based services", Communications of the ACM, vol. 46, no. 12, pp.61-65, December 2003.

[10] F. Ricci and Q.N. Nguyen, "Acquiring and revising preferences in a critique-based mobile recommender system", IEEE Intelligent Systems vol. 22, no. 3, pp. 22-29, 2007.

[11] S. Poslad, H. Laamanen, R. Malaka, A. Nick, P. Buckle, and A. Zipf, "CRUMPET: Creation of user-friendly mobile services personalised for tourism", in Proc. 3G, pp.26-28, 2001.

[12] M. van Setten, S. Pokraev, and J. Koolwaaij, "Context-aware recommendations in the mobile tourist application COMPASS", in W. Nejdl and P. De Bra (eds.) AH 2004, LNCS 3137, Springer-Verlag, 2004.

[13] K. Cheverst, N. Davies, K. Mitchell, A. Friday, and C. Efstratiou, "Developing a context-aware electronic tourist guide: Some issues and experiences", Proceedings of the SIGCHI conference on Human factors in computing systems (CHI2000), pp.17-24, The Hague, The Netherlands, April 2000.

[14] M. Roberts, N. Ducheneaut, B. Begole, K. Partridge, B. Price, V. Bellotti, A. Walendowski, and P. Rasmussen, "Scalable architecture for context-aware activity-detecting mobile recommendation systems", 2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks 2008 (WoWMoM 2008), pp. 1–6, 23-26 June 2008.

[15] J. Baus, K. Cheverst, C. Kray, "A survey of map based mobile guides", in L. Meng and A. Zipf (eds.): Map-based mobile services – Theories, Methods and Implementations. Springer, Berlin, Heidelberg, New York, 2005.

[16] M. Eisenhauer, R. Oppermann, and B. Schmidt-Belz, "Mobile information systems for all", in Proceedings of the Tenth International Conference on Human-Computer Interaction 2003, pp. 354-358.

[17] Stein, P. Hawking, and P. Sharma, "A classification of u-commerce location based tourism applications", in A. Treloar (ed.): AusWeb: The Eleventh Australasian World Wide Web Conference. Southern Cross, Lismore, Qld., Australia, pp. 224-232, 2005.

[18] P. Resnick and H.R. Varian, "Recommender systems", Communications of the ACM, vol. 40, no. 3, pp. 56-58, March 1997.

[19] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry", Communications of the ACM, vol. 35, no. 12, pp. 61-70, December 1992.

[20] J.S. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering", Technical Report MSR-TR-98-12, Microsoft Research Corporation, Redmond, Washington, USA, October 1998.

[21] D. Goldberg, "Genetic and evolutionary algorithms come of age", Communications of the ACM, vol. 37, no. 3, pp. 113-119, March 1994.

[22] P.E. Hart, N.J. Nilsson, B. Raphael, "A formal basis for the heuristic determination of minimum cost paths", IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pp. 100-107, July 1968.