Huang, H. & Gartner, G., 2009. Collective intelligence based mobile navigation in a smart environment. In Proceedings of the 6th International Symposium on Location Based Services & TeleCartography. Nottingham, UK.

Collective intelligence based mobile navigation in a smart environment

Haosheng Huang and Georg Gartner

Institute of Geoinformation and Cartography
Vienna University of Technology, Austria

Abstract

With more and more active or passive devices/sensors being augmented, our environment has become smarter. Also in the era of Web 2.0, the concept of "Web-as-participation-platform" has been fully adopted in the ICT society. This paper focuses on the question of how mobile navigation services can benefit from Smart Environment/Ambient Intelligence and Web 2.0/collective intelligence. After setting up a smart environment, a mobile navigation service is designed to support users' wayfinding, facilitate users' interaction and annotation with the smart environment, and collect user generated content (UGC). Based on UGC, this paper designs several collective intelligence based route calculation algorithms to illustrate the benefits of combining mobile navigation services, smart environment, and Web 2.0, such as providing "the nicest route", "the least complex route", "the most popular route", and "the optimal route". Finally, this paper concludes that mobile navigation services in smart environment can help to explicitly and implicitly collect user generated content (collective intelligence), and thus provide users with a new experience and smart wayfinding support (e.g., collective intelligence based route recommendations).

Keyword: Smart Environment, user generated content, collective intelligence based route calculation, mobile navigation service, Location Based Services

1. Introduction

With the gradual maturating of ubiquitous computing and rapid advances in mobile devices and wireless communication, Location Based Services (LBS) have gained high interest as one of the potential mobile "*killer-applications*" in the near future. Mobile navigation service is one of the most important LBS applications. It aims at providing wayfinding guidance in an unfamiliar environment. One of the successful mobile navigation systems is car navigation which has been widely used and trusted by car drivers. Recently, the increasing ubiquity of personal mobile devices (such as cell phones and PDAs) triggers a move towards *mobile pedestrian navigation systems*.

Currently, with the rapid advances in enabling technologies for ubiquitous computing, more and more active or passive devices/sensors are augmented in the physical environment, our environment has become smarter. This abundance of technologies has given place to the new notions of "Smart Environment (SmE)" and "Ambient Intelligent (AmI)". The basic idea behind SmE and AmI is that "*by enriching an environment with technology (sensor, processor, actuators, information terminals, and other devices interconnected through a network), a system can be built such that based on the real-time gathered and the historical data accumulated, decisions can be taken to benefit the users of that environment*" (Augusto and Aghajan 2009). With the increasing ubiquity of smart environments, the question of how mobile pedestrian navigation systems can benefit from SmE and AmI should be carefully investigated. However, to our knowledge, there is little work on that.

Another interesting event in the field of ICT (Information and Communication Technologies) is the gradual evolution of Web 1.0 to Web 2.0. Compared to "Web-as-information-source" in Web 1.0, Web 2.0 adopts the notion of "*Web-as-participation-platform*" (Wikipedia 2009a). In Web 2.0, users can actively contribute to the web. As a result, the term user generated content (UGC) entered mainstream usage since 2005 (Wikipedia 2009c). It refers to "various kinds of media content, publicly available, that are produced by end users". Currently, with the impetus of Web 2.0 applications, such as Facebook, Flickr, and Twitter, huge amounts of UGC are being created every hour, even every second. In this situation, the question of *how UGC can be used to generate value* becomes more and more important. Recommendation system is one of the most promising solutions for this question. UGC on the web can be viewed as users' collective intelligence. Recommendation systems can help to make collective intelligence useful. Some examples about this are "Customers who bought this item also bought" and "Best seller lists" at the Amazon website, "Most viewed" at YouTube, etc. Users can benefit from these kinds of collective intelligence based recommendations. The combination of LBS and Web 2.0 is a trend. Web 2.0 can enhance LBS with rich and real-time user generated content, which can be used to provide better services in LBS. There are some researches on this topic. They use UGC to provide event recommendations (de Spindler et al. 2006), tourist destination recommendations (Hinze and Junmanee 2006), restaurant recommendations (Dunlop et al. 2004), gas station recommendations (Woerndl et al. 2009), etc.

However, Web 2.0 has not been introduced to mobile navigation services. Most of the current mobile navigation systems are limited to provide richer, just-in-time information (navigation instructions) for users. However, a lot of users are not satisfied with simply being passive consumers, but rather want to be active contributors (Kang et al. 2008). By enabling UGC in mobile navigation services, users can share their personal experiences and feelings with other people (e.g., other navigators), which will fulfill users' *intrinsic desire* to share their experiences (with friends, or even with other people they don't really know) and thus provide users with a new experience during wayfinding. More importantly, *mobile navigation services can also be improved by integrating users' collective intelligence*. As a result, more work should be done to investigate the benefit of introducing the notions of Web 2.0 and collective intelligence into mobile navigation services, for example, providing collective intelligence based recommendations ("the most popular route"- the route which most people like).

This paper attempts to introduce the notions of SmE/AmI and Web 2.0/collective intelligence into mobile navigation services. We propose that *mobile navigation systems in smart environment can help to collect (gather and accumulate) user generated content (collective intelligence) explicitly and implicitly, such as ratings, comments, feedbacks, moving tracks, durations at decision points, etc. and thus provide users with a new experience and smart wayfinding support (e.g., collective intelligence based route recommendations)* .

The rest of this paper is structured as follows. In section 2, we deploy some devices/sensors to our office building and set up a smart environment as a testbed for our mobile navigation service. Section 3 describes methods to collect different kinds of user generated content in this smart environment. In section 4, we investigate how user generated content can be used to provide collective intelligence based route recommendations, such as the nicest route, the least complex route, the most popular route, and the optimal route. Finally, section 5 draws the conclusions and presents the future work.

2. Smart environment

Smart Environment (SmE) can be viewed as "*a physical world that is richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the everyday objects of our lives and connected through a continuous network*" (Weiser 1991). Based on this understanding, we established a simple smart environment with a positioning module, which uses sensors to provide adequate positioning information, and a wireless infrastructure module, which interconnects mobile clients (such as cell phones and PDAs) and devices installed in the environment (such as servers, sensors, etc.). This section will focus on these two modules.

2.1 Indoor positioning

Most of the outdoor navigation systems employ GPS for positioning. Unfortunately, GPS can only be used outside of buildings because the employed radio signals cannot penetrate solid walls. For positioning in an indoor environment, additional installations (e.g., WiFi or sensor networks) are required.

There exist numerous different positioning techniques that vary greatly in terms of accuracy, costs and used technology. Huang and Gartner (2009) provided a survey on different positioning techniques. After comparing different positioning techniques, a *Bluetooth-based beacon positioning solution* is adopted, which uses Cell of Original (CoO) as signal metric, proximity as positioning algorithm, and adopts passive position calculation. In the smart environment, we use BlueLon BodyTag BT-002 (Bluelon 2009) as Bluetooth beacon because we can adjust the range of BodyTag BT-002 by changing its transmit power. Bluetooth beacons are placed in different places actively broadcasting their unique IDs. Mobile devices passively receive the broadcast message when they are within the range of a beacon. After receiving a beacon ID, mobile devices look up the current position from a mapping table. This mapping table can be cached in the mobile devices or accessed from a server.

After choosing the positioning technique, the sensor placement which tries to optimize the placement to balance the signal coverage and development cost has to be considered. Different applications may have different coverage requirements. For indoor navigation, complete coverage is not necessary. As decision points (areas where the navigator must make a wayfinding decision, such as whether to continue along the current route or to change direction) are essential for wayfinding (Golledge 1999), we adopt a simple placement solution: beacons are placed at every decision point. The methods suggested in Brunner-Friedrich and Radoczky (2005) are used to derive the positions of decision points. And then, in order to avoid overlapping, the range for every beacon is adjusted.

2.2 Wireless infrastructure module

The wireless infrastructure module interconnects mobile clients and devices installed in the environment. To establish a wireless infrastructure, several technological solutions are possible: IrDA, WiFi, Bluetooth, UWB, ZigBee, etc. They differ in operating frequency, range, data transfer rate, connection type, etc. After carefully analyzing and comparing different technologies, we establish a wireless infrastructure based on WiFi technology because of its highly availability, its high data rate, and its wide coverage range.

A central server is also introduced to the smart environment. It is responsible for providing indoor navigation services, gathering and recording real time messages (such as users' moving track, user generated content, etc).

Figure 1 depicts the layout of the proposed smart environment. This smart environment is very simple, but it is enough as a testbed for effectively supporting the entire indoor navigation process, including indoor positioning, route selection, and route presentation. Also this smart environment enables collecting (gathering and accumulating) different user generated content explicitly and implicitly. For other applications, different other sensors such as temperature sensors and noise sensors may be also integrated into the smart environment to facilitate context gathering.

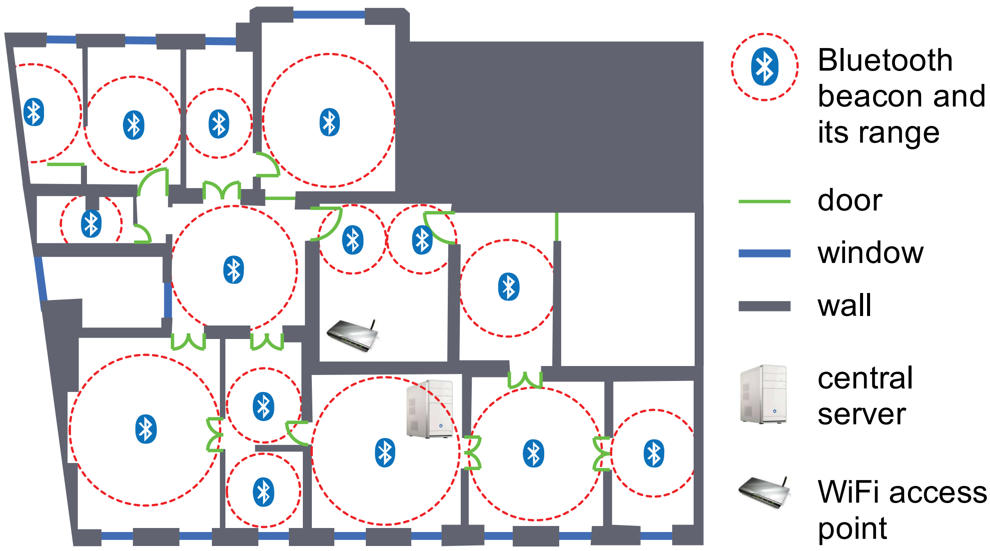

Figure 1. Layout of the proposed smart environment

3. Collecting user generated content in the smart environment

One of the great advantages of ubiquitous systems is the potentiality to directly interact with the environment. This functionality provides a basis for collecting different user generated content explicitly and implicitly. The proposed smart environment also supports this functionality.

We design a mobile navigation system to provide navigation guidance in this smart environment. During navigation in the smart environment, users receive wayfinding support which guides them to their destination. Currently, we calculate the shortest (distance) route for users according to the current context (mainly "traffic information"). For example, sometimes, some of the corridors are blocked, as a result, the system avoids guiding the user (navigator) crossing those corridors. As maps are proved to be one of the most useful presentation forms which can communicate route information efficiently, we employ schematic maps as the route presentation form. In order to enable users (navigators) to easily find their way with little cognitive load, we derive landmarks and visualize them in the route map. During navigation, if the user strays from the calculated route, the system will warn the user, and ask the user to go back. In order to protect their privacy, users can use the system anonymously.

However, the proposed mobile navigation system allows users to do more than just receive navigation guidance. They are also encouraged to contribute their own UGC (e.g., ratings, comments, feedbacks, etc.) while using the smart environment. UGC can be contributed in two ways: *explicitly* and *implicitly* (Svensson et al. 2005). The proposed system supports both of them.

3.1 Explicit collecting of user generated content

Explicit collecting means that the users have to provide information actively, for example, giving ratings, writing comments and feedbacks. These kinds of UGC, especially ratings simultaneously represent the collective intelligence of users of the system and can be used to make recommendations for other users (Ovaska and Leino 2008). The *motivations* to contribute here include not only the points (such as fun, ideology, values, understanding, enhancements, protective, career, and social) described by Nov (2007), but also the improvement of the recommendations we receive and the possibility of reaching much more relevant information (e.g., systems can learn our preferences from our UGC).

During navigation in the smart environment, users can annotate their personal preferences, comments or experiences to this environment. As the smart environment is georeferenced by the Bluetooth beacons (every beacon has an address), the user generated content posted by users can be viewed as *user generated georeferenced content*. Currently, the proposed system only supports text UGC. Multimedia UGC will be supported in the next version of the system. In default case, user generated content is dedicated for everyone (public) and has a permanent availability. Users can also specify the target person and the duration of it, for instance, this user generated content is only showed to Mary, and is only available on April Fools' Day. In order to protect the privacy, users can post their comments anonymously.

Currently, computers are hard to measure and process text information automatically. As a result, we also encourage users to give ratings. For navigation, the route users need to follow can be viewed as route segments connected by different decision points (areas). Users can give ratings for these two elements: decision point and route segment. In the smart environment, every decision point is georeferenced by a Bluetooth beacon, while every route segment is georeferenced by two Bluetooth beacons (two decision points).

At every decision point, users can give rating to identify the *level of complexity* (cost of effort) of making right decision (choosing the right road to follow) at this point. The rating value scales from 1 to 5. The more complexity, the higher the rating value. A rating for a decision point is always involved with *a pair of connected route segments* (the route segment which the user just visited, and the route segment which the user is going to visit). The current decision point is the junction of these two route segments. As a result, rating for a decision point is modeled as a *4-tuple (previous, current, next, value)* containing the previous decision point, the current point, the next decision point, and a rating value. For example, in Figure 2, rating (S, A, B, 4) is the rating for decision point A, and the user came from point S, and is going to visit B. It is important to note that for the same user, the complexity (effort cost) at a specific decision point to its neighborhoods may be different (e.g., "some routes are easy to be recognized"). For example, in Figure 2, a user may have the following ratings for decision point A: (S, A, B, 4), (S, A, C, 2), (S, A, E, 5).

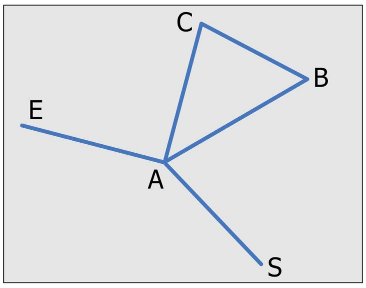

Figure 2. A road network

Rating for a route segment reflects users' *level of interest* for the route segment. For example, users may like route segment SA (in Figure 2) very much because of the nice view along it. The rating value scales from 1 to 5. The more the interest, the lower the rating value. Rating for a route segment is a *3-tuple (start, end, value)* containing the start and end decision point of the route segment, and a rating value. For example, a user like the route segment SA very much, and give the following rating: (S, A, 1).

When submitting the UGC, users only need to write their comments or give the rating values. The smart environment figures out the related positions from the positioning module (section 2.1), and stores the comments or ratings to the central server via the wireless infrastructure module (section 2.2).

3.2 Implicit collecting of user generated content

Implicit collecting means that users don't have to do anything other than using the system (Ovaska and Leino 2008). The system tracks users' actions and behaviors to detect their preference.

During navigation in the smart environment, a user's current position is recorded by the system every second, such as (userA, 2009-6-20 15:23:40, placeA), (userA, 2009-6-20 15:23:41, placeB). This sequential position information forms the user's moving track during her/his current navigation. In order to protect her/his privacy, the system uses a pseudo name (e.g., randomly generated by computer) to represent the user. Based on this real-time tracking, our system also provides a function to guide users to a person's current position (e.g., staff), which may be not her/his usual office (Huang et al. 2009).

For every moving track, some statistical data about the current navigation can be obtained: *moving duration at every decision point*, and *error point*. Similar to the explicit ratings for decision points, these two parameters may also reflect the complexity of decision points. For example, if the user stays more time at decision point A (duration is too long), and doesn't do anything (e.g, posting comments, giving ratings), it is reasonable to consider that the user has troubles in choosing which way to follow, as a result, the complexity of this decision point is high. Durations are recorded as *4-tuple (previous, current, next, value)* containing the previous decision point, the current point, the next decision point, and a duration (measured by second). Currently, we assume that, most of the time users spend during navigation is either on walking or on posting comments/ratings. As a result, if the user posts comments/ratings at the current decision point, the system will not record the duration for this decision point. The system also doesn't record the duration for error points. Durations have to be *standardized* to a cost scale (similar to ratings for decision points). In order to make this standardization, some field experiments should be carried out to find out some referenced durations.

The proposed navigation system continually checks whether the user is on the right route to her/his destination. If the user strays from the calculated route, the system will warn the user, and ask the user to go back. As a result, the system can identify the error points where a user makes wrong decisions by the following method: if the user moves in the sequence of A-B-A-C, point A will be the error point for this user because s/he made some wrong decision at A. For error point A, the following rating (previous, A, C, 5) is assigned.

4. Collective intelligence based navigation services

Currently, huge amounts of UGC are being created every hour, even every second. Recommendation systems can help to make UGC (collective intelligence) useful. Some examples about this are "Customers who bought this item also bought…" and "Best seller lists" in the Amazon website, "Most viewed" and "Most discussed" in YouTube, "Most popular tags" in Flickr, "The most popular bookmarks" in Del.icio.us, etc. These kinds of collective intelligence based recommendations can be very useful for the users of these services. Also these kinds of recommendation methods can help to achieve the center goal of Web 2.0 services: *the more they are used, the better they get* (Musser et al. 2006).

Inspired by the "most popular (viewed, discussed)…" like recommendations, we design several algorithms to illustrate *how our mobile navigation service can benefit from the user generated content* collected in section 3. We name these algorithms as collective intelligence based algorithm because they are based on users' collective intelligence. These algorithms use UGC (collective intelligence) to calculate different routes, such as route with minimal route segment rating (the nicest route), the least complex route, the most popular route, and the optimal route. As a result, we can provide collective intelligence based navigation service.

4.1 The nicest route: route with minimal route segment rating

The goal of this algorithm is to compute the route with minimal route segment rating between origin and destination. As described in section 3.1, rating for a route segment reflects users' level of interest on the route segment. The route with minimal route segment rating can be viewed as "the nicest route".

Generally, graphs are a standard data structure for representing road and transportation networks. A graph G consists of a set of vertices V and edges $E \subset V \times V$ connecting the vertices. In a route network, every intersection is represented as a vertex, and each road (route segment) is represented as an edge (Duckham and Kulik 2003). Edges can be assigned with weights (cost), for example, Euclidean distance of this edge, travel time, or travel fares. For our case, G is an Undirected Graph. The shortest (cost) route from origin A to destination B can be viewed as the path in graph G with least cost. *Dijkstra's algorithm* can be used to solve this problem (Dijkstra 1959). The basic idea of Dijkstra's algorithm is to assign some initial distance values and try to improve them step-by-step. Figure 3 depicts the pseudo code of the algorithm.

```
 1  function Dijkstra(Graph, origin, destination):
 2      for each vertex v in Graph:          // Initializations
 3          dist[v] := infinity,    previous[v] := undefined
 4      dist[origin] := 0                    // Distance from origin to origin is 0
 5      Q := the set of all nodes in Graph // All nodes in the graph are unoptimized - thus are in Q

 6      while Q is not empty:                // The main loop
 7          u := vertex in Q with smallest dist[]
 8          if dist[u] = infinity:     break                      // There is no route from origin to destination
 9          if u = destination:        break                      // reach the destination
10          remove u from Q
11          for each neighbor v of u:        // where v has not yet been removed from Q.
12              alt := dist[u] + cost_between(u, v)
13              if alt < dist[v]:            //If the distance is less than the previously recorded distance
14                  dist[v] := alt,      previous[v] := u

15      //read the shortest path
16      S := empty sequence
17      u := destination
18      while previous[u] is defined:
19          insert u at the beginning of S
20          u := previous[u]

21      return S
```

Figure 3. The pseudo code of Dijkstra's algorithm (adapted from (Wikipedia 2009b))

For calculating the route with minimal route segment rating from origin to destination, the rating for each route segment (road) is assigned to its corresponding edge in graph G. The rating for route segment (s,e) based on collective intelligence is calculated as:

$$R\_E(s,e) = \begin{cases} 3, & \text{if no ratings, use default value} \\ \dfrac{\sum R_i(s,e)}{n}, & \text{others} \end{cases} \qquad (f-1)$$

Where $R_i(s,e)$ is user *i*'s rating for route segment (s,e), and n is the total number of ratings for (s,e).

In order to use the Dijsktra's algorithm, line 12 of Figure 3 should be adapted to *"Alt:=dist[u]+R_E (u,v)"*.

4.2 The least complex route

The goal of this algorithm is to compute route with least complexity between origin and destination. As described in section 3.1, ratings for decision points reflect the complexity of each decision point. A route from origin to destination includes a series of decision points. As a result, the least complex route can be viewed as route with the minimal ratings for decision points.

Ratings for decision points are modeled as 4-tuple (previous, current, next, value). The rating value can be viewed as a cost assigning for a pair of connected route segments. For example, rating (S, A, B, 4) in Figure 2 can be viewed as the cost of negotiating the path from S to B through decision point A. The cost of navigating from node $previous$ to node $next$ through node $current$ based on collective intelligence is calculated as:

$$R\_DP(previous, current, next) = \begin{cases} 3, & \text{if no ratings, use default value} \\ \dfrac{\sum R_i(previous, current, next)}{n}, & others \end{cases} \qquad (f-2)$$

Where $R_i(previous, current, next)$ is user i's rating for route segments $(previous, current)$ to $(current, next)$, and n is the total number of ratings on this pair of route segments.

When introducing cost of pair of connected edges, the least cost route may include cycles. For example, in Figure 4, the route with minimal ratings from S to E is (S, A, B, C, A, E), which crosses the center node A twice.

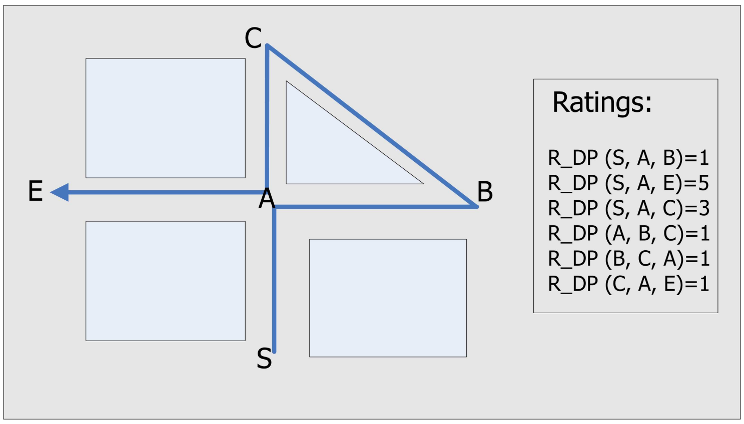

Figure 4. The route crosses the center node twice.

As a result, it is impossible to simply adapt the cost function in line 12 of Figure 3 to "Alt:=dist[u]+R_DP (previous[u], u, v)", and run the Dijkstra's algorithm. Because at the 2nd step of the main loop (lines 6-14) in Figure 3, node A will be removed from Q (list of unvisited nodes), and not be checked anymore. And then the Dijkstra's algorithm will report that "there is no route from S to E".

In order to solve this problem, we use the *restricted pseudo-dual graph* proposed by Winter (2002). The pseudo-dual graph D of the original graph G is defined as: 1) each edge e_i of G is represented as a node v_i in D, 2) each pair of connected edges (e_i, e_j) in G is represented as edge ε which connects nodes v_i and v_j in D. For example, Figure 5 depicts the pseudo-dual graph of the graph in Figure 2. Note that the pseudo-dual graph D is a Directed Graph.

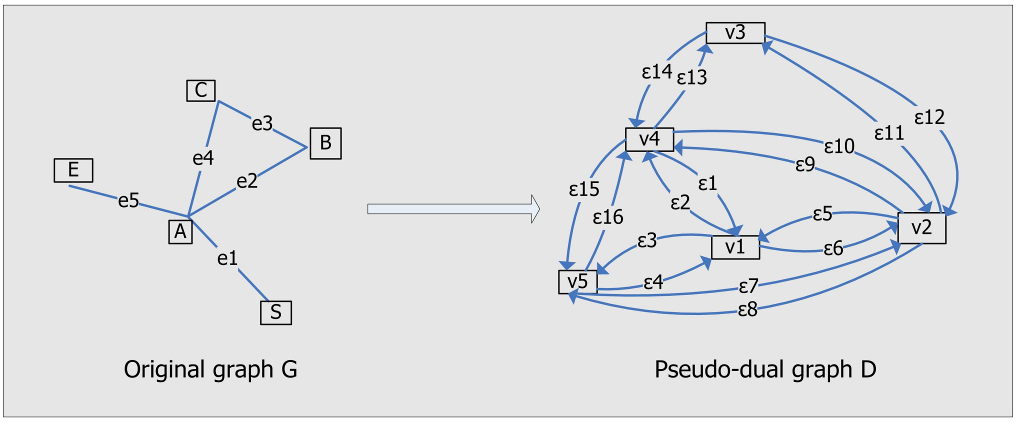

Figure 5. Original graph and its pseudo-dual graph

Winter (2002) proved that the shortest (cost) route (single-source/single target) problem in original graph G can be transformed to multi-sources/multi-targets problem in D. He reduced this problem to a single-source/single-target problem by adding a virtual source node and a virtual target node to D. In this new graph D', the shortest route can be computed by using the classical Dijkstra's algorithm.

For our case, the collective intelligence based costs for decision points (in formula 2) can be easily assigned to edges in the pseudo-dual graph D:

$$Cost(v1, v2) = R\_DP(previous, current, next) \qquad (f-3)$$

Where $v1$ in D is the edge $(previous, current)$ in G, and $v2$ in D is the edge $(current, next)$ in G.

Based on D, we adapt the line 12 of Figure 3 to *"Alt:=dist[u]+cost(u,v)"*, and use the classical Dijkstra's algorithm (Figure 3) to compute the route with least complexity between origin and destination. The result of this calculation is the least complex path in graph D. It can be easily transformed back to the route in graph G.

It is also interesting to note that these ratings for decision points can also be used to *quantitatively evaluate* the mobile navigation service. For example, for a long period of time (e.g., two months), if the average rating of a specific decision point is too high (e.g., >4.5), it is reasonable to consider that the navigation support provided at this decision point is very poor. As a result, information services provided by the navigation and smart environment at that place should be enhanced. This can be viewed as a new method for quantitative evaluation of software in software engineering.

4.3 The most popular route

The goal of this algorithm is to compute the most popular route between origin and destination. The most popular route is the *optimal trade-off* between the route with minimal route segment rating ("the nicest route" in section 4.1) and the route with least complexity (section 4.2).

In order to calculate the most popular route, we assign an optimum cost to each decision point, which depends on both the ratings for route segments, and ratings for decision points. This optimum cost is given by:

$$R\_DP_{popular}(previous, current, next) = \lambda_d \cdot R\_DP(previous, current, next) + (1-\lambda_d) \cdot R\_E(current, next) \qquad (f-4)$$

Where λ_d determines the weight of the impact for the ratings for decision points, $R\_E(current, next)$ and $R\_DP(previous, current, next)$ can be calculated by formula f-1 and f-2.

Similar to the algorithm in section 4.2, the most popular route can be calculated by the classical Dijkstra's algorithm based on the pseudo-dual graph.

In order to achieve better result, λ_d has to be calibrated. λ_d may be different for different environments. The method proposed by Haque et al. (2007) can be used to find out the optimum value for λ_d. It compares the results for different λ_d values with those obtained from the separate algorithms (e.g., route with minimal route segment rating in section 4.1 and route with least complexity in section 4.2).

4.4 The optimal route

Compared to the shortest (distance) route, the most popular route in section 4.3 may lead to longer distance between origin and destination. As a result, we calculate the optimal route, which takes ratings for route segments, ratings for decision points, and the Euclidean length of route segments into account. In order to calculate the optimal route, we assign an optimum cost to each decision point, which depends on the three parameters mentioned above. This optimum cost is given by:

$$R\_DP_{optimal}(previous, current, next) = (1-\lambda_o)\cdot R\_DP_{popular}(previous, current, next) + \lambda_0 \cdot Dist(current, next) \qquad (f-5)$$

Where λ_o determines the weight of the impact for the Euclidean length of route segments, $Dist(current, next)$ is the Euclidean length of route segments, and $R\_DP_{popular}(previous, current, next)$ can be calculated by formula f-4.

The calculation of the optimal route and the calibration of λ_o can use the same methods as in section 4.3.

4.5 Discussion

There are some similar papers focusing on calculating different routes for users. For example, the route with minimal number of turns, the route with minimal angle by Winter (2002); the route with least instruction complexity by Duckham and Kulik (2003); the reliable route which minimizes the number of complex intersections with turn ambiguities by Haque et al. (2007), etc. However, all of the above routes are mainly based on the geometrical characteristics of the road network. The proposed collective intelligence based algorithms are based on all users' UGC, which r*eflects users' navigation experiences* in the environment. As a result, compared to other route algorithms, our algorithms will provide results which are more suitable to the users.

From the mathematical perspective, the nicest route algorithm uses the cost (weight) of edges to compute the shortest (cost) route. The least complex route algorithm uses the cost (weight) of connected edges pair. Both of the most popular route algorithm and the optimal route algorithm combine cost of edges and cost of connected edges pair. In this paper, we use level of interest and complexity as cost. For some other applications, the cost function may differ, such as travel time, travel expenses, etc. However, the proposed algorithms can be also used to solve this kind of problems.

It is also important to note that the above algorithms of collective intelligence based route calculation make community-at-large recommendations to individual users. It is not especially made for any particular user but all get the same recommendation (Ovaska and Leino 2008). In our daily life, *these kinds of popularity-based recommendations have been proved to be very useful*. However, some of the users may have particular interests. In order to make more relevant recommendations for them, *collaborative filtering* should be introduced. The most known examples of collaborative filtering are Amazon-like "Customers who bought this item also bought" recommendations. Collaborative filtering includes two steps: 1) find out similar users (this step can be viewed as assigning the current user to a group), 2) carry out the "popularity-based recommendations" on this group of users. As a result, the proposed algorithms can also be used in the second step of collaborative filtering.

In this paper, we use indoor navigation as a testbed. However, the proposed algorithms can be also applied to outdoor pedestrian navigation services and car navigation services.

5. Conclusions and future work

Technology available today is rich. With the rapid advances in enabling technologies for ubiquitous computing, more and more active or passive devices/sensors are augmented in the physical environment, our environment has become smarter. Also, currently the ICT (Information and Communication Technologies) society has fully adopted the concept of "Web-as-participation-platform" in Web 2.0. As a result, *the combination of Location Based Services, smart environment and Web 2.0 is a trend.* This paper addressed this concern. In this paper, a smart environment with a positioning module and a wireless communication module was set up to support users' wayfinding, facilitate users' interaction and annotation with the smart environment, and collect user generated content. In order to illustrate the benefits of introducing smart environment and Web 2.0 into mobile navigation services, this paper designed several collective intelligence based route calculation algorithms to provide smart wayfinding support to users, such as "the nicest route", "the least complex route", "the most popular route", and "the optimal route".

From the above discussions, the following conclusions can be drawn: *smart environment can help to collect user generated content explicitly and implicitly (collective intelligence) during navigation, such as ratings, comments, feedbacks, moving tracks, durations at decision points, etc. By enabling UGC, mobile navigation services can provide users with a new experience and smart wayfinding support (such as, collective intelligence based route recommendations).*

According to Svensson et al. (2005), the central idea of Web 2.0 services is that: the more they are used, the better they get. As a result, our next step is to recruit more people to use our collective intelligence based navigation service. Furthermore, in the other part of our UCPNavi project, some researches are focusing on finding patterns of human wayfinding behaviors (Millonig and Gartner 2008). These patterns can help to divide users into different groups, and then be used for collaborative filtering. We will combine the result to provide more relevant recommendations.

Acknowledgements: This work has been supported by the UCPNavi project (Ubiquitous Cartography for Pedestrian Navigation, funded by Austrian FWF), which issues the problem of indoor navigation in a smart environment.

Reference:

Augusto, J. and Aghajan, H. (2009). Editorial: Inaugural issue. Journal of Ambient Intelligence and Smart Environments, 1(1), pp. 1-4.

Bluelon (2009). BodyTag BT-002. http://www.bluelon.com/index.php?id=263, Accessed on 2009.6.

Brunner-Friedrich, B. and Radoczky, V. (2005). Active Landmarks in Indoor Environments. In Bres, S. and Laurini, R. (eds.), VISUAL 2005, Springer LNCS 3736, pp. 203-215 .

de Spindler, A., C. Norrie, M., Grossniklaus, M. and Signer, B. (2006). Spatio-Temporal Proximity as a Basis for Collaborative Filtering in Mobile Environments. In C. Norrie, M., Dustdar, S. and Gall, H. (eds.), Proceedings of UMICS'06, Luxemburg, June 5-9, 2006.

Dijkstra, E.W. (1959). A Note on Two Problems in Connexion with graphs. Numerische Mathematik.1, pp. 269–271.

Duckham, M. and Kulik, L. (2003). "Simplest" paths: Automated Route Selection for Navigation. In Kuhn, W., Worboys, M.F. and Timpf, S. (eds.), COSIT 2003. Springer LNCS 2825, pp. 169–185.

Dunlop, M.D., Morrison, A., McCallum, S., Ptaskinski, P., Risbey, C. and Stewart, F. (2004) Focussed Palmtop Information Access through Starfield Displays and Profile Matching. In Crestani, F., Jones, M. and Mizzaro, S. (eds.), Proceedings of workshop on Mobile and Ubiquitous Information Access, Springer LNCS 2954, pp. 79–89.

Golledge, R. G. (1999). Wayfinding Behavior: Cognitive Mapping and Other Spatial Processes. The Johns Hopkins University Press, Maryland.

Haque, S., Kulik, L. and Klippel, A. (2007). Algorithms for Reliable Navigation and Wayfinding. In Barkowsky, T. et al. (eds.), Spatial Cognition V, Springer LNAI 4387, pp. 308–326.

Hinze, A. and Junmanee, S. (2006). Advanced Recommendation Models for Mobile Tourist Information. In Meersman, R. and Tari, Z. et al. (eds.), Proceedings of OTM Confederated International Conferences, CoopIS, DOA, GADA, and ODBASE 2006, Springer, pp. 643-660.

Huang, H. And Gartner, G. (2009). A Survey of Mobile Indoor Navigation Systems. Report of UCPNavi Project, Vienna University of Technology, 2009.

Huang, H., Gartner, G., Schmit, M. and Li, Y. (2009). Smart Environment for Ubiquitous Indoor Navigation. In Proceedings of NISS 2009, IEEE.

Kang, Y., Stasko, J., Luther, K., Ravi, A. and Xu, Y. (2008). RevisiTour: Enriching the Tourism Experience with User-Generated Content. In O'Connor, P., Hoepken, W. and Gretzel, U. (eds.), Information and Communication Technologies in Tourism 2008, Springer, pp. 59-69.

Millonig, A. and Gartner, G. (2008). Ways of Walking - Developing a Pedestrian Typology for Personalised Mobile Information Systems. In Gartner, G. and Rehrl, K. (eds.), Location Based Services and TeleCartography II - From Sensor Fusion to Context Models, Springer Verlag Berlin Heidelberg, pp. 79 - 94.

Musser, J., O'Reilly, T., and O'Reilly Radar Team (2006). Web 2.0 Principles and Best Practices. O'Reilly Radar Report, O'Reilly Media, Inc. http://www.oreilly.com/catalog/web2report/chapter/web20_report_excerpt.pdf, Accessed on 2009.6.

Nov, O. (2007). What Motivates Wikipedians? Communications of the ACM, 50(11), pp. 60-64.

Ovaska, S. and Leino, J., 2008. A Survey on Web 2.0. http://www.cs.uta.fi/reports/dsarja/D-2008-5.pdf, Accessed on 2009.6

Svensson, M., Hoeoek, K., and Coester, R. (2005). Designing and Evaluating Kalas: A Social Navigation System for Food Recipes. ACM Transactions on Computer-Human Interaction, 12(3), pp. 374-400.

Weiser, M., 1991. The Computer for the 21st Century. Scientific American, 265(3), pp. 94-104.

Wikipedia (2009a). Web 2.0. http://en.wikipedia.org/wiki/Web_2.0, Accessed on 2009.6.

Wikipedia (2009b). Dijkstra's algorithm. http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm, Accessed on 2009.06

Wikipedia (2009c). User-generated content. http://en.wikipedia.org/wiki/User-generated_content, Accessed on 2009.06

Winter, S. (2002). Modeling Costs of Turns in Route Planning. GeoInformatica, 6(4), pp. 345-361.

Woerndl, W., Brocco, M. and Eigner, R. (2009). Context-Aware Recommender System in Mobile Scenarios. International Journal of Information Technology and Web Engineering, 4(1).