Calculating Walking Distance in a Hybrid Space Model

Wen-Tai Hsieh*, Nancy Miller+, Yung-Fang Yang*,
Shih-Chung Chou* and Peter Steenkiste+

* Advanced e-Commerce Institute
Institute for Information Industry
Taipei, Taiwan, ROC

+ Department of CS and ECE,
Carnegie Mellon University
Pittsburgh, PA 15213, U.S.A.

***Abstract-* The CMU Aura system introduced a hybrid location model for space representation in ubiquitous computing environments. The hybrid location model combines hierarchical symbolic names with coordinates, with the goal of gaining the benefits of both: symbolic names offer an intuitive representation and captures spatial relationships while coordinates provide high precision. While the hybrid location model has served us well, we have so far made little use of the coordinates. In this paper we report on our ongoing work of implementing an Aura service that estimates walking distance. The availability of both symbolic names and coordinates turns out to be very advantageous for this service since it provides us both accurate distance estimates and spatial relationships that can be used to incorporate user preferences.**

I. Introduction

Many research groups are exploring are exploring context-aware computing. Xerox's Palo Alto Research Center (PARC)[1], for example, has been working on pervasive computing applications since the 1980s. In the context of Project Aura[2][3][6], Carnegie Mellon University is developing technologies that will reduce user distractions by having applications and systems automatically adapt to the user's context, including physical and computational context and user preferences.

Location information is a critical piece of context information for context-aware applications used by mobile users. Using the user's location enables services such as diverting phone calls to the receiver nearest to a user, perfecting data to the service portal near the user's path, locating interesting objects/people, navigation, etc. In order to support such applications, Aura developed the hybrid location model [3], which combines symbolic names with coordinates. The motivation for this design was that it combined the benefits of hierarchical symbolic names (accurately captures spatial relationships, intuitive for users) and coordinates (high degree of precision). In this paper, we report on our work in implementing a "walking time" service. It estimates the walking time between two points in space and it leverages the benefits of the hybrid space service.

The remainder of this paper is organized as follows. We first introduce a motivating application: a context-aware museum tour guide. We then describe the Aura hybrid space service and describe how it is used by the walking time service. Finally, we describe the status of our implementation.

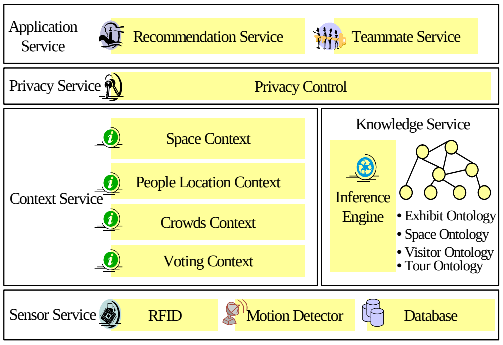
II. Context Aware Tour Guide

The Context Aware Tour Guide [4] at Institute for Information Industry is a ubiquitous computing project that focuses on giving museum visitors personalized guidance that is context sensitive. The goal is to optimize the visitors' experienced while they are visiting complex and unfamiliar environment. The target museum is the National Museum of Natural Science [5], which is one of the largest Museums in Taiwan. It has more than 3 million visitors per year and as an educational organization, they want to provide visitors with a maximum amount of useful information by providing interesting tours. We can image that with customized information and guidance, tours could be more attractive and the tourists would definitely gain more knowledge from the same exhibitions.

Traditionally, visitors of museums or art galleries are guided by guides who show them one of a set of scheduled tours. These tours typically include both exhibits that are of interest and exhibits that are not of interest. Visitors never know on which tour is most suitable for them. More, tours may not always be available because there are always more visitors than there are guides available. As a result, we started to build a Tour Guide for the National Museum of Natural Science, using context-aware technology.

Figure I shows the architecture of the context-aware tour guide. When visitors arrive at the museum, they will receive a handheld device. Before starting their tour, the visitors will enter some preferences (e.g. what type of exhibits they like) and personal information (e.g. planned duration of stay, walking speed, ..). Based on this information, the Recommendation service will either recommend a pre-defined tour that best meets the visitor's interest and schedule or the Recommendation service could create a custom tour. The tour will be displayed on a map of hand-held device and will provide the visitor step by step guidance, including in-depth information on the exhibit included in the tour. Throughout the tour, the visitor can provide input to the system about the exhibits. The Recommendation Service can then recommend changes to the tour, e.g. to adjust to user's changes in preferences or to avoid rooms that have become crowded. We believe that such a context-aware tour guide will enhance the visitor's experience and learning.

Figure I
Museum Tour Guide Architecture



An essential requirement for the tour guide is the ability to estimate how long it will take the visitor to walk between exhibits. Walking time estimates must be personalized: not only does it depend on a person's walking speed but it should also consider preferences such as taking stairs versus an elevator.

III. Hybrid Space Model

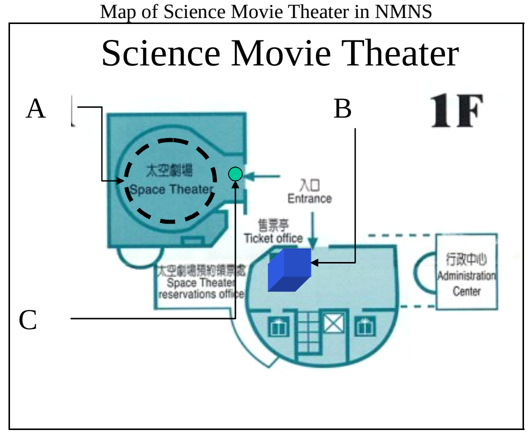
One important issue for context-aware applications is that they must model the physical environment through an appropriate location model. The prevailing location models fall into two groups: hierarchical names and coordinates [9]. Hierarchical names organize the space as a hierarchy of spaces, similar to for example hierarchical directory names or DNS. They are very intuitive and express spatial relationships but they are not very precise. In contrast, coordinates based on geometric models are simpler and precise, but they are less intuitive. Table I summarizes the differences between the two models.

TABLE I
Comparisons OF Symbolic and Geometric Space Model

Feature \ Space Model	Symbolic	Geometric
representation of spatial relationships	Good	Poor
human readability	Good	Poor
Specifying locations precisely	Poor	Good
Computing distance accurately	Poor	Good

In order to try to gain the benefits of both models, the Aura project built a Hybrid Space Model [3]. In the hybrid location model, a location is represented by an Aura Location Identifier (ALI), which consists of a hierarchical name, followed by a set of coordinates that identify the location inside the space identified by the hierarchical name. Both name and the coordinates are optional, i.e. coarse spaces can be represented by a hierarchical name only, while in some applications, only coordinates are needed (e.g. GPS).

Example I
Map of Science Movie Theater in NMNS



ALIs can be used to specify three types of location model (Example I).

- **Space** ALI represents a physical space in the hierarchy, e.g. the Science Movie Theater on the first floor of NMNS (see A in Figure II):
  ali://NMNS/SMTheater/floor1/SpaceTheater
- **Point** ALI represents the exact position of an object, e.g. the location of the entrance of Science Movie Theater of NMNS (see C in Figure II):
  ali://NMNS/SMTheater/floor1/SpaceTheater#(10,4,1)
  The coordinates are relative to the Space Theater's space coordinate system
- **Area** ALI represents a space defined by the application, e.g. the ticket office space of Science Movie Theater of NMNS (see B in Figure II):
  ali://NMNS/SMTTheater/floor1/#{(1,0),(-1.5,0.5),(0,3),(2,3.5),(3,1.5)-(1,5)}
  The first group of coordinates specify the four corners of the area while (1,5) specifies the height range of the area.

Five primitive operations are defined on ALIs:

- Distance(Ali,Ali) returns the physical distance between two ALIs.
- Contains(Ali,Ali) returns whether one location contains another.
- Within(Ali,Ali) returns whether one location is within another.

- Super(ali) returns the direct super space containing the location.
- Sub(ali) returns the list of all subspaces contained in the specified space.

The Hybrid Location model forms the basis for the Aura space service, which is part of the Contextual Information Service (CIS)[7]. The CIS collects information about the user's physical and computing context and makes it available to context-aware applications through an SQL-like API. The CIS isolates the context-aware applications from the details of how information is collected. This allows the CIS to use multiple sources of information, transparently to the user. This is for example important for location information, which can be derived from a variety of sources (e.g. RF/ID badges, calendar, wireless networks, ..). The CIS was originally designed to support the office environment [6,8], but is being ported to the represent the museum contextual information.

In order to support the Museum Tour guide, we are also implementing a Walking Time service that estimates how long it will take a visitor to walk between two points. Note that the "distance" function defined in the original space service represents geometric distance, i.e. it does not consider walls etc, so it is not a good indicator of walking time. The walking time service is interesting because it relies heavily on both the hierarchical name and coordinate features of the Aura space service.

IV. Walking Time Estimation

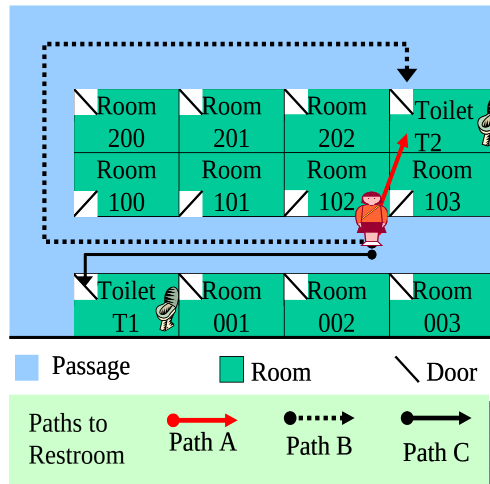
We discuss the design of the walking time service in two steps. We first discuss walking between two points on the same floor and then we look at walking between floors or buildings. First, we introduce the concept of a path.

Walking Path

The original space service operates in terms of areas and points in areas. To estimate walking times, we also need to introduce the concept of a "path". For instance, we want to find a nearest restroom for visitor Kelly (Figure I). A "nearest bathroom" service based on the distance function in the ALI Space Model will return T2 (Path A) since it is the closest bathroom in space. Unfortunately, Path A between Kelly's location and T2's location passes through a wall. The nearest bathroom service instead should return T2, which can be reached using Path B.

This simple example shows that estimating walking time requires us to first calculate feasible (walking) paths between two points. A path simply consists of a sequence of points, such that it is possible to walk between consecutive points. In our model, two consecutive points can be connected by a straight line, called a segment, so, given the coordinates of the two points, it is easy to calculate the distance between the two points (or the length of the segment). The length of the path is simply the sum of the lengths of the segments in the path.

Figure I
Visitor Routing Map

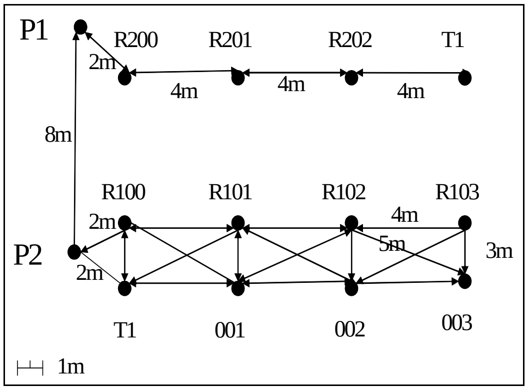


Walking Time on a Floor

On a single floor, we expect that for most people, walking time will be roughly proportional to distance, i.e. we can estimate walking time by multiplying distance with a user-specific walking speed. An important point is that "distance" is an objective metric, so we can use the same algorithm for all visitors. There are some exceptions to this rule. For example, visitors with strollers or wheelchairs may not be able to use all doorways. As we will see below, it is pretty easy to incorporate these constraints in our model by simply removing certain "connections" from the model for those visitors.

The next question is how we can identify feasible paths between two points and calculate their distance. The original space service provides information about spaces, including their physical size. However, it does not provide information about connectivity between spaces, i.e. it does not specify what rooms were connected or where the doors might be located. The first step is extend the space service to provide this information. Figure II, for example, shows the connectivity information for the space shown in Figure I: each dot represents a connection. Connections can be doors, e.g. connecting rooms to a corridor, or open entrances (i.e. "doors" that cannot be closed), or simply locations where two spaces flow into each other, e.g. the locations marked P1 and P2 show where the corridors in Figure I connect.

Figure II
Connection Information and Transfer Cost



Specifically, we need to add the following information to the space service:

1. Connectivity information : for each node in the space hierarchy (i.e. space), we add connectivity information, i.e. what other spaces is it connected to.
2. Information about the nature of the connection, e.g. its location (as a set of coordinates), possible constraints (e.g. one way only, no strollers, ..).

The space service is implemented using a database [3]. To add the connectivity information, we simply add a table. The primary columns in the new table are a "from" space and a "to" space, both in the form of an ALI, and a "coordinate" column. An entry simply means that it is possible to walk from the "from" space to the "to" space using some form of entrance located at the specified coordinates; the coordinates are specified relative to the coordinate system of the floor. Additional columns can specify constraints, e.g. fire exit only.

Given this connectivity information, we can now easily formulate the problem of finding feasible paths and of calculating walking distances as a graph problem in the following way:

- The nodes in the graph correspond to the connectivity points in the connectivity table.
- We add two nodes to the graph corresponding to the visitor's current location and target destination, both represented as a set of coordinates.
- Edges in the graph correspond to the spaces connecting the points. Figure II shows for example all the edges connecting the connectivity points. The weights of the edges correspond to the length of the segments they represent, i.e. the distance between the two points; this can be easily calculated based on the coordinates associated with each point.

After modeling the connectivity on a floor as a directed graph, we can find feasible paths as any sequence of edges. We can find the path representing the shortest walking distance by simply applying a shortest path routing algorithm to the graph. If some inter-space connections are not usable for some visitors, those nodes can simply be removed from their graphs.

Our results distance calculation algorithm is as follow:

1. UserLocation = GetUserLocation(UserID)
2. StartPoint =
   FindNearestConnectionPoint(UserLocation)
3. Dist_UL_to_StartPoint =
   Distance(UserLocation,,StartPoint)
4. Path = Get the shortest path between StartPoint
   and Destination
5. TotalDistance =
   Path.Distance + Dist_UL_to_StartPoint

Walking between Floors

While walking time on a floor is typically proportional to the distance between the two points, determining the best way to walk between two flows is more complex. The reason is that we have to consider the visitors' preferences with respect to using stairs or elevators. Also, we have to be able to consider the personal tradeoff between going to a different floor versus walking further on the same floor. Let us consider some example:

- Some people cannot use stairs and they always need an elevator. Others (e.g. elderly) will strongly prefer elevators, while others prefer stairs because they dislike the wait associated with an elevator. These tradeoffs may depend on the number of floors that has to be crossed.
- Given a choice between a destination (e.g. a restroom) on the same floor or on a different floor, people will have different preferences. Some people will be willing to walk an extra 50 meters to avoid using stairs/elevators, while for others, the cutoff may be only 10 meters.

Note that these preferences could also be context dependent. For example, they may depend on how crowded the museum is (will impact the waiting time at the elevator).

The hybrid space service provides two options for implementing a multi-floor walking time service. The first option is to extend the single floor solution: we build graphs for individual floors and use information about the location of stairs and elevators to connect them. We then have to calculate personalized weights for the edges representing stairs and elevators reflecting the user's preferences and possibly context information and solve the routing problem. This is conceptually fairly straightforward, although the graph could become quite large for large spaces and incorporating personal preferences may become tricky.

An alternative is to apply a hierarchical solution based on space hierarchy. We first solve per-floor problems to find the shortest distance between both the user's location and the destination and the stairs and elevators of interest. We then solve a small "inter-floor" optimization problem. This could be done by exhaustively evaluating the cost of all possible paths between the two floors, since the number of options will be small. This approach is more complex, but it maybe easier to incorporate personal preferences. We plan to implement both algorithms and compare them both in terms of performance and flexibility.

With both approaches, we need the user's preferences. The easiest format to use is "time", e.g. a per-floor elevator cost of 1 minute and stair cost of 2 minutes would reflect a preference for elevators and also a relative preference to walking a certain distance. Note also that we can use the same two approaches to estimate the time to walk between two buildings.

In important point is that out path finding approach uses both the coordinate and symbolic aspects of the hybrid space model. We use the symbolic information to specify what spaces are (directly) connection while the coordinate system is used to automatically calculate distances between different connections for the same room. If either aspect were missing, some information would have to be imported from outside the model.

V. Implementation

We are in the process of implementing the "walking distance" service. The format of the "connectivity" table that is added to the space service is shown in Table II. Additional columns will be added later to reflect constraints and properties of the connection.

Table II
Relation Definition of Connection Point in Database

Field	Type	Notes
ConnectionPointID	Integer	Door
FromSpaceID	Varchar	ALI
ToSpaceID	Varchar	ALI
Coordinates	Varchar	ALI

To find the shortest path between two connection points (Figure III), we implemented the Floyd All-Pairs-Shortest-Path algorithm [10]. It uses a dynamic-programming methodology to solve the All-Pairs-Shortest-Path problem. The algorithm runs in $O(N^3)$ time, so it is potentially expensive. If necessary, the results for common queries (e.g. to find restrooms) could be cached since the information is static. The distance between the user's location and the nearest connection point is calculated on the fly using the coordinate system.

Implementing the "walking time" service, once the database has been set up, is clearly straighforward. The most time consuming part of the project is adding the relevant connectivity information to the database. The Aura space service includes a GUI that allows users to enter data into the space service based on maps in the AutoCad format. We are currently in the process of extending this tool to also allow users to specify the connectivity information. Once that is completed, we will be able to run full scale tests of the Walking Time service.

Fig III
Floyd algorithm

```
Floyd(int L[1..n][1..n])
{
   D = L;
   P = 0;
   for (k=1; k<=n; k++)
     for (i=1; i<=n; i++)
        for (j=1; j<=n; j++)
           if (D[i][k]+D[k][j] < D[i][j]) {
              D[i][j] = D[i][k]+D[k][j];
              P[i][j] = k;
           }
}
```

Track the shortest path

VI. Related Work

Car navigation systems and trip planning services such as MapQuest have to find the "shortest path" between two points. While there metric is driving distance (or time), the goal is similar to ours and they use routing solutions similar to ours. The main difference is that finding paths inside a building is more complicated because spaces can be connected in many different ways, e.g. door, elevator, stairs, … Roads tend to be more uniform.

A number of groups have observed that coordinate based (geometric, Euclidian, physical, ..) and symbolic (semantic, topological, …) models for space representation have complementary benefits [9,11,12,13]. Several of these groups also observed that the term "distance" has a different meaning in these two types of models. In coordinate-based systems, distance tends to refer to the Euclidian distance. In symbolic systems, the concept of distance is less clearly defined, but it is typically related to the connectivity between spaces, as it is for example expressed at the different levels of a space hierarchy [12,13]; this representation can more easily be used to represent the walking distance.

The space model and navigation solution presented in [13] is the closest to our work. It consists of a topological model (similar to our hybrid space representation) and an exit hierarchy (similar to our connectivity information). The main difference with our work is that the hybrid nature of our space model makes it possible to automatically calculate the "primitive distance" between two directly connected exits while this information has to be obtained from outside the model in [13]. Moreover, our representation of exits/connectivity is simpler, probably because our connectivity information is defined relative to the hybrid space model.

VII. Conclusion

In this paper, we proposed a path calculation model based a hybrid space model. The solution uses both the hierarchical name and coordinate information provided by the hybrid space model. The space service is enhanced with space connectivity information. This allows us to formulate the problem of finding the path with the shortest walking distance as a simple routing problem.

Our solution allows the selection of paths that are customized to individual users. This is especially important for when walking between floors or buildings. In those cases, walking time is not simply proportional to distance, but we have to consider people's ability and preferences with respect to dealing with stairs and elevators.

ACKNOWLEGEMENT

This research was in part funded by the Institute for Information Industry and the Ministry of Economic Affairs, Department of Industrial Technology, ROC, and in part by the National Science Foundation under award number CCR-0205266.

REFERENCES

[1] Publications on PARC webpage: http://www.parc.xerox.com/

[2] Project Aura, http://www-2.cs.cmu.edu/~aura/

[3] Changhao Jiang1, Peter Steenkiste "A Hybrid Location Model with a Computable Location Identifier" In Proceedings of Ubiquitous Computing. 246-263

[4] Context aware tour guide webpage : http://140.92.140.46/MiningTeam/context/scenario.htm

[5] NATIONAL MUSEUM OF NATURAL SCIENCE webpage: http://www.nmns.edu.tw/

[6] Garlan, D., Siewiorek, D., Smailagic, A., and Steenkiste, P. Project Aura "Towards Distraction-Free Pervasive Computing." IEEE Pervasive Computing 1, 2 (April-June 2002), 22–31.

[7] Judd, G., and Steenkiste, P. Providing Contextual Information to Ubiquitous Computing Applications. Technical Report CMU-CS-02-154, Department of Computer Science, Carnegie Mellon University, July 2002.

[8] Next Wave Technology and Research Conference webpage: http://www.itpilot.org.tw/921216/index.htm

[9] Domnitcheva, S. "Location modeling: State of the art and challegnes." In Workshop on Location Modeling for Ubiquitous Computing (2001).

[10] Zhenlin Wang " Introduction to Algorithms" webpage: http://www.csl.mtu.edu/cs4321/www/Lectures/floyd.ppt

[11] Barry Brumitt and Steven Shafer, "Topological World Modeling Using Semantic Spaces", Workshop on Location Modeling for Ubiquitous Computing, held in conjunction with Ubicomp 2001, September 2001, Atlanta.

[12] Christoph Schlieder, Thomas Vogele, and Anke Werner, "Location Modeling for Intentional Behavior in Spatial Partonomies", Workshop on Location Modeling for Ubiquitous Computing, held in conjunction with Ubicomp 2001, September 2001, Atlanta.

[13] Haibo Hu and Dik-Lun Lee, "Semantic Location Modeling for Location Navigation in Mobile Environments", 2004 IEEE International Conference on Mobile Data Management, January 2004, Berkeley.