A Wearable Virtual Usher for Vision-Based Cognitive Indoor Navigation

Liyuan Li, *Member, IEEE,* Qianli Xu, *Member, IEEE,* Vijay Chandrasekhar, *Member, IEEE,*
Joo-Hwee Lim, *Member, IEEE* Cheston Tan, *Member, IEEE,* and Michal Akira Mukawa *Student Member, IEEE,*

***Abstract*—Inspired by progresses in cognitive science, artificial intelligence, computer vision, and mobile computing technologies, we propose and implement a wearable virtual usher for cognitive indoor navigation based on egocentric visual perception. A novel computational framework of cognitive wayfinding in an indoor environment is proposed, which contains a context model, a route model, and a process model. A hierarchical structure is proposed to represent the cognitive context knowledge of indoor scenes. Given a start position and a destination, a Bayesian network model is proposed to represent the navigation route derived from the context model. A novel Dynamic Bayesian Network (DBN) model is proposed to accommodate the dynamic process of navigation based on real-time first-person-view (FPV) visual input, which involves multiple asynchronous temporal dependencies. To adapt to large variations in travel time through trip segments, we propose an online adaptation algorithm for the DBN model. Hence, the proposed DBN model becomes a self-adaptive DBN (SA-DBN). A prototype system is built and tested for technical performance and user experience. The quantitative evaluation shows that our method achieves over 13% improvement in accuracy as compared to baseline approaches based on HMM. In the user study, our system guides the participants to their destinations, emulating a human usher in multiple aspects.**

***Index Terms*—Indoor Navigation, Wayfinding, Dynamic Bayesian Network, Egocentric Vision, Wearable Vision System.**

I. INTRODUCTION

When people visit an unfamiliar building, they occasionally have difficulties finding their ways, partly due to the lack of a mental representation of the environment [1]. Since GPS signals are usually unavailable inside buildings, researchers have investigated an array of sensor-based localization technologies, such as infrared, WiFi, RFID, USID, Bluetooth, and Barcodes [2], [3]. Among them, WiFi fingerprinting has been extensively used for indoor positioning [4], [5]. Sensor-based technologies rely on special building infrastructure and comprehensive database, which presents a few challenges for indoor environments [3], [4]. These include the high cost for infrastructure construction, the accuracy of localization, the delay of position computing, and the reliability of signals. Furthermore, sensor-based localization methods typically provide a user's location on a 2-D map (i.e. allocentric information). It requires extra mental effort on the user to interpret the map and associate it with his/her egocentric perception of the environment [6], [7], largely owing to the different mental mechanisms for processing allocentric and egocentric information [8].

If a human usher guides a visitor to a destination in a building, he/she usually gives the directions based on egocentric representations [9], [10], *i.e.* cognitive knowledge for wayfinding and egocentric visual perception of the current environment, e.g. "*Go ahead to the intersection and turn right*". Such instructions are easier for the visitor to follow [11]. As indicated in cognitive psychology, the egocentric perspective is very efficient for navigation [12], because both the usher and visitor do not rely on precise 3D metrical reconstruction or detailed floor plans. Rather, they communicate on common cognitive concepts and egocentric perceptions (*cognitive navigation*), which does not depend on a 2D map.

In the near future, wearable cameras and computing devices may enable new technologies for online personal services based on first-person-view (FPV) input [13], [14]. In this paper, we propose a wearable virtual usher for indoor navigation using a wearable camera driven by cognitive models of wayfinding [6], [15]. The basic idea is to provide *aided wayfinding* [15], whereby the user follows a set of verbal instructions from the wearable system, thus incurring less cognitive load and concerns of being lost.

Many researchers have shown that computational models inspired by biological or cognitive mechanism are both effective and efficient for artificial perception [8], [16], [17], [18]. We propose a novel computational framework of cognitive wayfinding for indoor navigation based on egocentric visual perception. Under this framework, a knowledge representation model is proposed to support wayfinding using the cognitive concepts of indoor scenes, rather than relying on geographical locations in the physical world, as did by most existing methods. Further, we introduce probabilistic models for route representation to handle the uncertainties in scene recognition during navigation. The hierarchical context model is built according to the cognitive strategy of *coarse-to-fine* heuristic concepts of indoor scenes [19], [20]. The Bayesian Network model is employed to represent the route model, taking into account variations in travel time. To handle the flexibility of cognitive representation and uncertainties associated with image classification, a Self-Adaptive Dynamic Bayesian Network (SA-DBN) is proposed to accommodate the dynamic process of wayfinding according to the route model and real-time scene recognition based on the FPV input. Besides the performance evaluation, we conduct a user study to compare our system with two other modes of navigation, *i.e.* a 2D map and a human usher. The results show the potential of wearable devices for

L. Li, Q. Xu, V. Chandrasekhar, J.-H. Lim, C. Tan and M. Mukawa are with the Institute for Infocomm Research, Singapore, 138632, e-mail: {lyli,qxu,vijay,joohwee,cheston-tan,stumam}@i2r.a-star.edu.sg

Manuscript received April 19, 2005; revised December 27, 2012.

indoor navigation in unfamiliar environments.

The main contributions of this paper can be summarized as follows: (a) A computational framework of cognitive wayfinding for indoor navigation based on egocentric visual perception, incorporating a hierarchical context model, a graphical route model, and a DBN-based process model; (b) A novel DBN model, namely SA-DBN, that achieves over 13% improvement in accuracy for indoor wayfinding based on real-time FPV input. It includes a multi-stream asynchronous DBN model and an online self-adaptation algorithm to adapt to the large variations in travel time of different users.

II. Related Work

The mental abilities of humans in wayfinding have been studied over several decades [15], [21]. Previous research has focused mainly on the exploration of cognitive knowledge representations and inference mechanisms for wayfinding [22], [23], [24]. Theories have been proposed to explain how a human user constructs a cognitive map of the physical world, how the cognitive map is represented in the brain, and what strategies are used in wayfinding based on the cognitive map. The models (*e.g.* a simulated cognitive map) are applied to solve route-planning tasks in large-scale spaces such as landscapes, cities, and campuses [11], [22], [25], [26]. Another cluster of cognitive models focuses on modeling and simulating the human wayfinding process [1], [27]. Wayfinding is formulated as a route-following process, which is represented as a sequence of action pairs: landmark detection and action execution to move to the next landmark [1], [15], [28], [29]. Inspired by past work in the area of wayfinding [1], [4], [29], [30], [31], [32], this research is concerned with the learning and representing scenes, routes, and survey knowledge of an environment and aims to aid humans in wayfinding. The route-following process can be represented as a graph model, such as a finite state machine. Existing computational models for wayfinding are artificial intelligence (AI) models based on formal language, which barely account for the uncertainties of human perceptions and spatial measurements. Recently, bio-inspired models and systems are proposed to simulate the human pereception and cognitive process using visual information [16]. Similarly, inspired by human cognition, we propose a hierarchical context model to represent the human mental model of an indoor environment, and use probabilistic models (*i.e.* Bayesian Networks) to handle uncertainties in perception and prediction in route representation and wayfinding. In existing research, the representation model and process model for wayfinding are investigated separately. In comparison, our computational framework is an integration of the knowledge representation model and the process model of indoor wayfinding.

Vision-based indoor localization and navigation has attracted attention of both computer vision and robotics researchers. Existing approaches can be classified into two categories: (1) image-based localization [33], [34], [35] which requires a large image database for every location from different viewing angles; (2) route-following [36], [37] which requires the memory of a sequence of frontal views along a route. Both categories are based on image-matching, *i.e.* a distance based on either pixel-level comparison [36] or local image feature matching [35]. Omnidirectional images are employed to reduce the ambiguities of indoor scenes [38], [39]. The most challenging issue is matching reliability for indoor scenes where the differences between images are often based on small local features [13]. This issue is particularly important for indoor scene recognition [40] as compared to outdoor applications [41]. In [42], Torralba *et al.* investigate an HMM-based approach to perform sequential place and scene category recognition on a recorded indoor sequence. Image similarity is measured using a GMM representation of GIST features. Simple first-order Markov HMMs are used for place and scene category recognition independently. The authors predict that better performance can be achieved if the place-category dependency is exploited. Applying forward-backward algorithm to HMM with an epitomic image matching, Ni *et al.* achieve improved performance on the same dataset [43].

SLAM (Simultaneous Localization and Mapping) is a popular technique in robotics research [44], [45], [46]. SLAM can be used by a robot to build a map of an environment and perform navigation tasks by keeping track of robot's location. Castle *et al.* [47] extend SLAM to track visual objects on a wall for hand-held and wearable cameras. However, as mentioned in [38], SLAM technique might not be applicable for human navigation with wearable cameras. Besides the reasons listed in [38], there are two main difficulties for SLAM in indoor navigation. First, at different positions in a commercial building, the feature points (salient features) in the images are similar due to the homogeneity of interior design, (*i.e.* repeated architectural elements) [46]. Second, a user may frequently look around by rotating his/her head in an unfamiliar building, which results in significant changes of viewing angle, irregular image motion, and severe motion blur. The Kalman filter and particle filter based feature point tracking and visual odometry (*e.g.* used in RatSLAM [48]) algorithms may not be robust enough to tackle these difficult conditions [46], [49].

In recent years, Bayesian models have been extensively investigated to tackle the dynamics of complex systems and decision processes [8], [49], [50]. Among them, DBN is a powerful probabilistic model to describe dynamic processes and has several advantages over HMM [51]. Many DBN-based methods for event and activity recognition have been proposed [52], [53]. They have been proven to be effective in learning and representing the complicated logic and temporal dependencies for complex dynamic processes, with improved robustness in vision recognition. Our model is distinctive from existing DBN models in three aspects. First, existing DBN models are defined as a t-slice first-order Markov model and unrolled for T consecutive time slice to represent a complete event. Hence, they are synchronous DBN models. In [54], an asynchronous DBN model is proposed where the message may be propagated at different time steps. Our model represents a dynamic system that evolves at different temporal streams. Second, existing DBN models are used in an offline manner, *i.e.* they are used to classify a time sequence (event or action) after it has been completed. The forward-backward

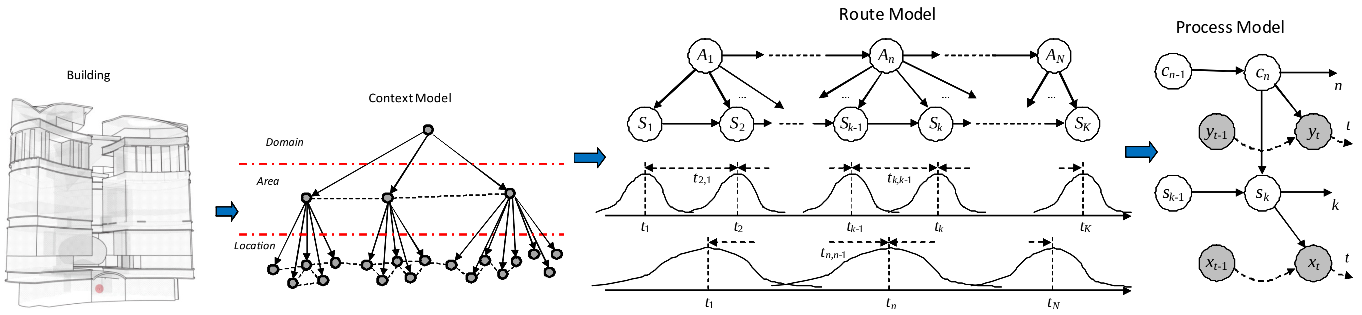


Fig. 1. The framework for computational wayfinding and navigation on FPV input, where a hierarchical context model is designed to represent survey knowledge for wayfinding in a complex building, a graphical route model derived from the context model is used to represent the route knowledge for a navigation task, and a DBN-based dynamic model is proposed to perform the wayfinding process according to the route model and real-time scene recognition on a wearable camera.

passes through the whole sequence are employed iteratively to improve the state estimation. In contrast, our DBN model is applied in an online manner. It performs online state inference as the event progresses. Third, existing DBN models are time-invariant. The term “dynamic” is used to refer to a dynamic system, not a network that changes over time [51]. Our model is self-adaptive in the sense that it updates its parameters based on the current visual input and system status, *i.e.*, when the navigation task is in progress.

III. The Approach

According to cognitive models of wayfinding, three levels of cognitive knowledge are required, namely (a) knowledge about the environment (*survey knowledge*), (b) knowledge about the path (*route knowledge*), and (c) knowledge about the destination (*destination knowledge*) [15]. In addition, a model of a logical sequence of *landmark-action* pairs is used to represent the wayfinding process [1].

We propose a computational framework of cognitive wayfinding in a complex indoor environment based solely on egocentric visual perception. The framework incorporates both knowledge representation and process models for wayfinding. It consists of (a) a hierarchical context model to represent survey knowledge based on cognitive concepts of indoor scenes, (b) a Bayesian Network based route model to represent a navigation route, and (c) a DBN-based process model to perform online wayfinding according to the route model and real-time FPV input. Figure 1 shows a diagram of the framework.

A. The Context Model

In spatial cognition, researchers focus on learning a cognitive map to represent the mental spatial model of an environment [55]. However, the format of cognitive maps is still not formally defined [11], [56], [57]. There is evidence to argue that people do not need complete knowledge of a space in order to move about effectively [58]. Hence, instead of trying to build a mental map of the physical world, we propose using cognitive concepts of scenes to represent the physical world for wayfinding and navigation since such knowledge can be easily incorporated into human navigation in an unfamiliar environment. We propose a hierarchical model to represent the context knowledge of an indoor environment. For purpose of illustration, we build our model using a typical commercial building as shown on the left of Figure 1.

The context model is a tree structure of three levels. It is designed according to the cognitive strategy of *coarse-to-fine* [19], [20], [21] on conceptual representation of indoor areas and locations, rather than physical regions in existing models. At the top level, a root node is used to represent a specific building. At the next level, the nodes represent typical areas (zones) in the building. There are three area-nodes for three conceptual categories: (a) Public area (*Mall*), (b) Transition area (*LiftLobby*), and (c) Working area (*OfficeZone*). The nodes at the bottom level indicate the relative locations in each area, which makes the description of a route in the building easy to understand. For example, the locations in *Mall* are *MainEntrance*, *Lobby*, *Shop*, *Gate* to *LiftLobby*, etc.; the locations in *LiftLobby* are *Lobby*, *InLift*, *Entrance*, etc.; the locations in *OfficeZone* are *Entrance*, *Junction*, *Corridor*, *Exit*, *Cubicle*, *MeetingRoom*, *Pantry*, *etc.* We use the label *Locations* to denote the set of the nodes in this level. The nodes in each level may be connected if there is a direct path between them, as indicated by dash-lines in Figure 1. For example, in the second layer, the nodes of *Mall* and *LiftLobby* are connected, and, in the third layer, the *Entrance* of *LiftLobby* is connected to the *Entrance* of *OfficeZone* through a glass door. The hierarchical context model represents the *survey knowledge* of the building based on cognitive concepts of scenes and locations from egocentric observations. This can be easily adopted to other buildings as it is based on generic cognitive concepts.

At each node (except the root node), we train a SVM classifier for scene recognition [59]. For real-time FPV input, we first perform “Area” level scene recognition. The winner SVM represents the area scene (y_t) and the node is activated. The next step of scene recognition is performed on the child nodes of the activated node. The winner SVM represents the recognized location (x_t).

It is worth noting that the scene is defined based on egocentric perspectives. On a conceptual level, indoor scenes can be classified into three categories, namely *area*, *line*, and *point*, depending on the spatial relationship between the observer and the environment [60], [61]. When a person observes a scene of an area (*e.g. Mall*), it means that the person is inside the area. When one observes a view of a line scene (*e.g. Corridor*), in most cases he/she is walking along the line. However, when one observes a point scene (*e.g. Junction, Gate*), it does not mean that he/she is at the location now, but is facing and approaching the location. The common concepts of egocentric observations can benefit the design of route directions for cognitive navigation. When the user is inside an area, the virtual usher may have to guide him/her to the point connecting to the next area. If the user just enters a long line scene, the virtual usher may have to provide a prompt to confirm that he/she is moving along the right direction. When the user is close to the end of a line scene or approaching a point scene, the virtual usher ought to give a direction to the next scene.

B. The Route Model

In existing models of wayfinding, a route is represented as a sequence of landmarks [1], [29]. However, in indoor environments with multiple levels and similar interior constructions, there are less distinctive landmarks compared to outdoor environments. With the cognitive representation of indoor scenes and locations as described above, a complete route for an indoor navigation task can be represented by a sequence of trip segments, where a trip segment is a route with a single transportation mode, *e.g. walking along a Corridor*, *turning left at a Junction*, and *waiting for lift at LiftLobby*. The location node in the hierarchical context model is used to represent a trip segment and the area node can be considered as a large trip segment.

Once the start point and the destination of a trip are determined, a corresponding route model (as shown in Figure 1) for the navigation task can be generated from the hierarchical context model. The route model consists of two levels of node chains. The upper level is a chain of area nodes (*i.e.* $A_n \in Areas$) which provides a brief description of the route, *e.g.* a route from *Mall Area* through *Lift Lobby* to *Office Area*. The lower level is a chain of location (or trip segment) nodes (*i.e.* $S_k \in Locations$) for a detailed representation of the route. The area nodes are connected to their child nodes in the next level according to the hierarchical context model. The two-level context representation links the scenes of areas and locations, or place-category mentioned in [42].

Depending on the individuals' preference and the conditions of the route, it may take different time to go through a trip segment. The Gaussian-shaped curve under each location node illustrates the variation in travel time for a trip segment, which can be obtained from a few data collection experiments that give the average travel time for each trip segment. Based on these, for a navigation route, each location node (*e.g.* node S_k) can be described using four parameters: the mid-time point (t_k) of the travel time, the transition time ($t_{k,k-1}$) from S_{k-1} to S_k, the temporal scale (σ_k) of the travel time (estimated as half of the travel time), and the minimum travel time (d_k) for the trip segment, i.e. $(t_k, t_{k,k-1}, \sigma_k, d_k)$. Similarly, the curves at the bottom level describe the variations in travel time through the corresponding areas. The temporal properties of an area node can be described by the parameter set $(t_n, t_{n,n-1}, \sigma_n, d_n)$.

The Bayesian Network based route model represents the *route knowledge* for a navigation task. The image recognition in each node of the route model are copied from the corresponding nodes in the context model.

C. The Process Model

Existing models of wayfinding process (*i.e.* View-Action [1], [15], [28], [29]) can be considered as a finite state machine with deterministic logic conditions for state transition, which do not account for the uncertainties in perception. Since spatial cognition consists of both perceptual and conceptual processes [62], the wayfinding process model needs to integrate perception and cognition in real world scenarios. Therefore, the system must accommodate the uncertainty, nonlinearity and dynamics of the process [49]. We propose a novel DBN model to integrate visual perception and cognitive route model to perform navigation considering its power in handling uncertainties in a dynamic process.

1) Asynchronous DBN Model: The user is expected to have a basic level of mental and kinematic abilities to maneuver in indoor environments. Following online guidance, he/she will go through the trip segments in a sequence as described by the route model. An area node can be considered to be a large trip segment which is formed by merging trip segments represented by its child location nodes. According to the graphical model for a navigation route, the dynamic model for navigation can be expressed as an Asynchronous Dynamic Bayesian Network (A-DBN) model (refer to the right end of Figure 1). Following conventions, we use the shaded nodes to denote observations and the white nodes for hidden states. In the model, nodes c_n and s_k are hidden state variables, and y_t and x_t are observations. The node c_n indicates which area the user is at time t, *i.e.* $c_n \in \{A_1, \cdots, A_N\}$. The node s_k represents the location in the area, or the trip segment the user is going through at time t, hence, $s_k \in \{S_1, \cdots, S_K\}$. The nodes y_t and x_t represent the two-level scene recognition results, so that $y_t \in Areas$ and $x_t \in Locations$. The A-DBN model indicates that the observations from the user's wearable camera depend on which area and location he/she is in according to the route model.

Different from conventional DBN models where one particular state of a system is represented by a t-slice DBN and a complete event is represented by an unrolled DBN for T consecutive time slices, our asynchronous DBN model describes a system that is dynamically changing or evolving for different temporal streams. In A-DBN model, the observations evolve at time step t, the state s_k evolves on the transition between locations (or trip segments), and the state c_n changes on the transition between areas, according to the route model. Hence, the A-DBN model includes different evolving streams over time, as indicated by the horizontal links with different labels in the figure. Similar to a conventional DBN, the proposed A-DBN can be considered as a time slice and can be unrolled along time to represent a dynamic system. The key difference is that the first-order Markov models are applied at different temporal scales for different streams.

Assuming that a navigation trip takes T time steps, we denote the sequences of observable variables as $X_T = \{x_0, \cdots, x_{T-1}\}$ and $Y_T = \{y_0, \cdots, y_{T-1}\}$, respectively. At a time step t, the user is traveling through a location s_k of an area c_n. The hidden-state variables are denoted as (c_n, s_k). Let $C = Areas$ and $S = Locations$ denote the sets of areas and locations. According to the fundamental formulation of DBN [51] and the specific structure of A-DBN model shown in Figure 1, the joint distribution can be expressed as

$$\begin{aligned} P(X_T, Y_T, S, C) = p(c_0) \prod_{t=1}^{T-1} p(c_n|c_{n-1}) \prod_{t=1}^{T-1} p(y_t|c_n) \\ p(s_0|c_0) \prod_{t=1}^{T-1} p(s_k|s_{k-1}, c_n) \prod_{t=1}^{T-1} p(x_t|s_k). \end{aligned} \quad (1)$$

Let $\mathbf{s}_t = (c_n, s_k)$ denote the vector of hidden-states at time t, $\mathbf{s}_p = (c_{n-1}, s_{k-1})$ be the vector representation of previous states of $\mathbf{s}_t$, and $\mathbf{y}_t = (y_t, x_t)$ denote the observation vector. Then, the items in (1) can be combined and expressed on vector representations as

$$\left\{ \begin{array}{lcl} p(\mathbf{s}_t|\mathbf{s}_p) & = & p(s_k|s_{k-1}, c_n)p(c_n|c_{n-1}) \\ p(\mathbf{y}_t|\mathbf{s}_t) & = & p(x_t|s_k)p(y_t|c_n) \\ p(\mathbf{s}_0) & = & p(s_0|c_0)p(c_0) \end{array} \right. . \quad (2)$$

From (1) and (2), the probability distribution function (pdf) of the A-DBN (1) can be expressed as

$$P(X_T, Y_T, S, C) = \prod_{t=1}^{T-1} p(\mathbf{s}_t|\mathbf{s}_p) \prod_{t=1}^{T-1} p(\mathbf{y}_t|\mathbf{s}_t) p(\mathbf{s}_0). \quad (3)$$

The prior and conditional pdfs are defined according to the route model, as described below.

First, instead of defining state transition functions on two consecutive time slices as do in most existing DBN models, we define state pdfs on two-level trip segment transitions. This is based on the following observation. In an indoor environment, there is no clear separation between two connected areas or locations from the FPV observations. The images captured from FPV may cover both regions when the user is walking from one location to the next, *e.g.* from a lift lobby to a corridor. The user may also change her/his viewing angle drastically due to head movement even without much body movement. During such transitions in a FPV sequence, image classification is unstable. As a result, it is difficult to define a reliable state transition function based on just observations from two consecutive time slices.

At the location level of the A-DBN model, the function $p(s_k|s_{k-1}, c_n)$ represents the probability that, at time t, the user transits from trip segment s_{k-1} to s_k, given that s_k is a child node of c_n. Since s_{k-1} and c_n are causally independent [63], one can define

$$\begin{aligned} & p(s_k|s_{k-1}, c_n) = \xi_s(s_k|s_{k-1})\xi_c(s_k|c_n) \\ & = \left[\exp\left(-\frac{|(t - t_{k-1}) - t_{k,k-1}|^2}{\sigma_k^2} \right) \beta(s_k|s_{k-1}) \right] \xi_c(s_k|c_n), \end{aligned} \quad (4)$$

where t_{k-1} is the mid-point time stamp of the travel time through trip segment s_{k-1}, $t_{k,k-1}$ is the average transition time from s_{k-1} to s_k, σ_k is the time scale of s_k, $\beta(s_k|s_{k-1})$ is a logic term defined according to the route model, and $\xi_c(s_k|c_n)$ is the contribution of c_n to s_k. If s_k is a child node of c_n, $\xi_c(s_k|c_n) = 1$, otherwise, $\xi_c(s_k|c_n) = 0$. To avoid including 0 in the product pdf (3), instead of using 0 and 1 values for $\beta(s_k|s_{k-1})$, we define

$$\beta(s_k|s_{k-1}) = \left\{ \begin{array}{ll} 0.9, & \text{if } s_{k-1} \text{ is activated} \\ 0.1, & \text{otherwise.} \end{array} \right. \quad (5)$$

Another reason to use 0.9/01 values for $\beta(s_k|s_{k-1})$ is that the scene recognition accuracy is typically less than 90%. Once the navigation trip starts, the first trip segment $s_0 = S_1$ is activated, while the remaining ones are "locked". As the user passes through trip segments one-by-one, the corresponding trip segments s_k are activated in sequence. These switch functions are useful in keeping the state changing smoothly and avoiding unusual jumps in state between unconnected trip segments.

Similarly, the transition pdf function $p(c_n|c_{n-1})$ for area nodes can be computed as

$$p(c_n|c_{n-1}) = \exp\left(-\frac{|(t - t_{n-1}) - t_{n,n-1}|^2}{\sigma_n^2} \right) \beta(c_n|c_{n-1}), \quad (6)$$

where an area node c_n will be activated only after all of its child nodes (s_k) have been activated, and σ_n is the time scale of c_n, which is the sum of its child nodes' time scale values.

The observation pdfs are defined based on two-level scene classification. They are computed as:

$$p(y_t|c_n) = \left\{ \begin{array}{ll} 0.9, & \text{if } y_t = c_n, \\ 0.1, & \text{otherwise.} \end{array} \right. \quad (7)$$

$$p(x_t|s_k) = \left\{ \begin{array}{ll} 0.9, & \text{if } x_t = s_k, \\ 0.1, & \text{otherwise.} \end{array} \right. \quad (8)$$

Again, the value 0.9 is chosen as discussed above. The observation pdfs can also be obtained from a training set, but it may require a large training set which covers almost all the locations in the building evenly.

2) Online State Inference: The process of wayfinding is to correctly recognize the state progress according to the route model and egocentric observations, that is, at any time t during the trip, we want to estimate the user's state $\mathbf{s}_t$ according to the observations made so far. According to (3), this can be expressed as

$$\hat{s}_t = \arg\max_k P(X_t, Y_t, S, C). \quad (9)$$

From (3), we can obtain the log pdf as

$$\begin{aligned} Q_t &= \log P(X_t, Y_t, S, C) \\ &= \sum_{i=1}^{t} \log p(\mathbf{s}_i|\mathbf{s}_p) + \sum_{i=1}^{t} \log p(\mathbf{y}_i|\mathbf{s}_i) + \log p(\mathbf{s}_0) \\ &= \sum_{i=1}^{t-1} \log p(\mathbf{s}_i|\mathbf{s}_p) + \sum_{i=1}^{t-1} \log p(\mathbf{y}_i|\mathbf{s}_i) + \log p(\mathbf{s}_0) \\ &\quad + \log p(\mathbf{s}_t|\mathbf{s}_p) + \log p(\mathbf{y}_t|\mathbf{s}_t) \\ &= Q_{t-1} + q_t. \end{aligned} \quad (10)$$

Hence, the current state can be obtained as

$$\begin{aligned}\hat{s}_t &= \arg\max_k q_t \propto \arg\max_k [p(\mathbf{s}_t|\mathbf{s}_p)p(\mathbf{y}_t|\mathbf{s}_t)] \\ &\propto \arg\max_k [p(s_k|s_{k-1})p(c_n|c_{n-1})p(x_t|s_k)p(y_t|c_n)]. \end{aligned} \quad (11)$$

Equation (11) indicates that the estimation of state $\hat{s}_t$ depends on both the current observations and the propagation of hidden states. It is worth noting that existing approaches (*e.g.* decoding) for state inference on DBN cannot be applied here as it requires the entire sequence to be available, *i.e.* the navigation trip has been completed.

3) Self-Adaptive DBN (SA-DBN): In an A-DBN model, the parameters of a location node s_k and an area node c_n are $(t_k, t_{k,k-1}, \sigma_k, d_k)$ and $(t_n, t_{n,n-1}, \sigma_n, d_n)$, respectively. These parameters may vary significantly across different trips. For example, in one trip, the lift may take a user from level 1 to 10 directly, while in an another trip, the lift may stop frequently at different levels. Fixed parameters cannot accommodate such large variations in real-world scenarios. We design a self-adaptive algorithm to tune parameters online to adapt to each trial, while the navigation task is in progress. Hence, we call our model a Self-Adaptive DBN (SA-DBN).

For initialization, we generate a set of parameters using trip averages of 3-5 persons. For a trip that has been completed, the mid-time point of trip segment s_k can be computed as

$$t_k = \sum\nolimits_{t:s_t=s_k} q_t^k t \Big/ \sum\nolimits_{t:s_t=s_k} q_t^k, \quad (12)$$

where q_t^k, as computed in (11), is the probability of the user being in s_k at time t. Note, to provide online guidance, we need to estimate the mid-time point of current trip segment even while the trip segment is in progress.

Following the guidance, the user goes through trip segments one-by-one. At time t, if he/she is traveling through trip segment s_k, the mid-time point is computed based on existing observations as

$$\hat{t}_k^t = \sum\nolimits_{i\in[0,t];s_i=s_k} q_i^k i \Big/ \sum\nolimits_{i\in[0,t];s_i=s_k} q_i^k, \quad (13)$$

where q_i^k is larger if i is closer to the mid-time point. Meanwhile, the travel time in s_k till now can be counted as D_k^t. We define a weight measure for the time period as $w_k^t = D_k^t/(2\sigma_k^{t-1})$, where, if $w_k^t > 1$, we set $w_k^t = 1$. Then, the predicted mid-time point of s_k is updated as

$$\begin{array}{ll} t_k^t & = w_k^t \hat{t}_k^t + (1-w_k^t)t_k^{t-1}, \\ t_{k,k-1}^t & = t_k^t - t_{k-1}^t, \\ \sigma_k^t & = w_k^t(t_{k,k-1}^t - \sigma_{k-1}^t) + (1-w_k^t)\sigma_k^{t-1}, \end{array} \quad (14)$$

where if σ_k^t is less than half the minimum travel time (d_k), it is set as $d_k/2$.

Once the parameters of s_k are updated, those of the following trip segment nodes are revised accordingly. The mid-time point and transition time are updated as

$$\begin{array}{lll} t_l & = & t_{l-1} + \sigma_{l-1} + \sigma_l, \\ t_{l,l-1} & = & t_l - t_{l-1}, \end{array} \quad (15)$$

for $l = k+1, \cdots, K$, where σ_l is initial value for s_l. In addition, once node s_k is updated, its parent node c_n is also updated. Let C_n be the set of c_n's child nodes, then, the time scale of c_n is computed as

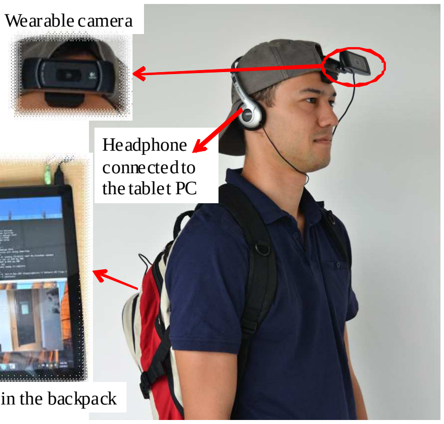

Wearable camera
Headphone connected to the tablet PC
Tablet PC in the backpack

Fig. 2. The prototype of our wearable vision system with the camera placed on forehead.

$$\sigma_n^t = \sum\nolimits_{k:s_k\in C_n} \sigma_k^t, \quad (16)$$

and t_n and $t_{n,n-1}$ are updated in the same way as (15). The area nodes following c_n are updated accordingly.

With online updating, the SA-DBN model can adapt to individual users on the fly. When the user enters a new trip segment, the estimated mid-time point is first reduced since $\hat{t}_k^t$ is much smaller than the initial t_k value. The weight w_k^t controls the rate of reduction. If the user spends more time in a given segment, the estimated mid-time point is increased gradually, till he/she leaves the segment. If the user walks faster than average, t_k is moved forward and σ_k is decreased. If the user stays in the trip segment longer than average (*e.g.* stays in lift), t_k is pushed back and σ_k increases. The parameters of the subsequent trip segments are updated accordingly.

IV. Experimental Evaluation

To evaluate the proposed method, we built and tested a prototype system. The prototype wearable system, i.e. the virtual usher, consists of a wearable camera and a tablet. A web camera (Logitech HD1080p) is placed on the user's forehead as shown in Figure 2. It is connected to a Samsung Series 7 Slate tablet (1.86GHz Core i5 processor and 4GB of RAM) with a USB cable. The system is implemented using MS VC++ 2005 in Windows 7. It processes about 8 fps at an image resolution of 320×240 pixels. It should be noted that the current prototype is still a bit heavy to be considered as "wearable". However, we believe it has the essence of a wearable system with a FPV vision device and a mobile computing device. Considering the progress in hardware miniaturization, it is possible to connect a "smart glass" with a smart phone for the task.

The online procedure for navigation guidance can be summarized as the following steps:

1) Captuing incoming image from the wearable camera;
2) Computing integral images of quantized gradients;

3) Extracting randomly sampled dense SIFTs;
4) Area and location recognition to generate y_t and x_t;
5) SA-DBN-based tracking;
6) Audio instruction.

More details on implementation, as well as performance evaluations and user studies, are described in the following subsections.

A. Implementation of Context Model

We tested our system in a complex commercial building which comprises a large shopping area, lift lobbies, and office areas spanning more than ten levels. A context model of the building is constructed as described in Section III-A. Figure 3 shows some example images of the scenes corresponding to the *Area* and *Location* level nodes of the hierarchical structure, which can provide some details on the implementation of the hierarchical context model illustrated in Figure 1. The first row lists the three area nodes of the context model. In the second row, the location nodes of each area node are listed, where the red texts denote area nodes and the green texts represent location nodes. Below the text rows are example images of the scenes from each area, where, in each image, the red text overlapping the image indicates the area and the green text denotes the location.

In our implementation, each node is realized using frame-like representation [64], where the slots are used to store the name of the node (area or location), pointers to parent and child nodes, connections to neighboring nodes, and the SVM model for scene recognition. The images used to train the SVM classifiers for the area and location nodes are also stored in the same tree structure as the context model. The training sample images are collected randomly from accessible places in the building with a wearable webcam by different persons at different time. The whole training image set is then divided into three subsets, according to the area where the images are captured, *i.e.*, *Mall*, *LiftLobby* and *OfficeZone*. For each subset corresponding to one area (*e.g. Mall*), the images are further divided into child location nodes and stored in the folds at the bottom-level nodes (*e.g. MainEntrance*, *Shops*, *Gates* and *Mall(Lobby)*). The SVM classifiers for the nodes are trained locally according to the tree structure of the context model. For an area node (*e.g. Mall*), the positive training images come from the folds of its all child nodes, while the negative training images are the collection from the child nodes of the other two area nodes. As for a location node, the positive training images come from the fold of the node, while the negative training images are the collection from its all sibling location nodes under the same area node.

B. Efficient Vision Recognition

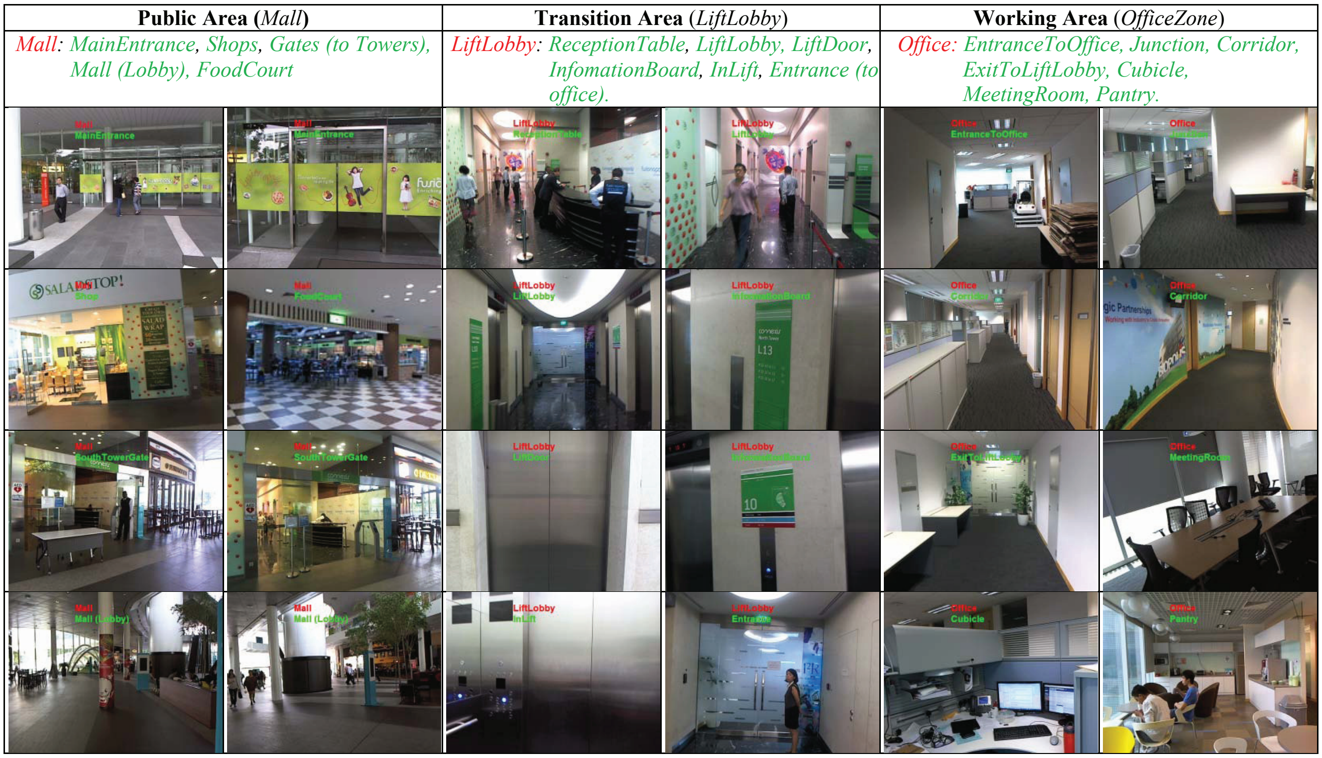
The bag-of-words (BOW) representation has been shown to be effective for image classification [65], but it may not be practical for real-time tasks. We have proposed an efficient BOW-based method for scene recognition on portable computers [59]. Unlike conventional methods which build the bag-of-words on the raw local features *e.g.*, SIFTs, our method first maps the extended SIFT features (which encode corresponding spatial information) into an embedded low-dimensional manifold subspace by an improved Spectral Regression, called Expended Spectral Regression (ESR), and then generates the visual feature dictionary (BOW) in the embedded feature space. SVM classifiers are trained on such BOW. In our method, the dimension of the visual word vector is about half of the original SIFT feature, and there is no need to employ SPM (Spatial Pyramid Matching) to encode spatial information. Hence, it requires much less memory and computational resource. In this system, the same BOW-based image representation is used for both scene and location recognition.

We extract dense SIFTs for image classification since the dense SIFTs can clearly improve the performance of BOW-based image classification [65]. To achieve real-time processing on dense SIFTs, we generate integral images of gradients quantized on orientations. On integral images of gradients, we can extract multi-scale dense SIFTs very efficiently. Though the multi-scale dense SIFT can improve the image classification, the number of them is too large for real-time processing (1882 SIFT features from an image of 320×240 pixels [59]). Two speed-up procedures are proposed for real-time computing in mobile devices.

First, from each image, we randomly select about 20% of multi-scale dense SIFTs for classification, which impacts the performance of image classification by up to 5% compared with that on all dense SIFT features from an image. Besides, in training, we can generate 10 training samples from each image by randomly select 20% multi-scale dense SIFTs each time, which means, we can generate a stable and robust SVM classifier on a small training data set which reduces the labor burden of training significantly.

To generate a BOW-based histogram, we have to match each feature with each word in the dictionary. Suppose the feature is a k-dim vector, *i.e.* $\mathbf{s} = (s_1, \cdots, s_k)$. If we obtain M features from an image and there are N words in the dictionary, the computational cost is kMN. Since the best match is the one of the minimum distance between two feature vectors, *i.e.* $\mathbf{s} \in \mathbf{w}_n$ if $n = \arg\min_j d(\mathbf{s}, \mathbf{w}_j)$ with $d(\mathbf{s}, \mathbf{w}_j) = \sum_{l=1}^{k}(s_l - w_{jl})^2$, we can exploit the previous matching results to speed up the following matching operation. Suppose d_m is the minimum distance obtained by matching $\mathbf{s}$ with the first $(n-1)$ words $\mathbf{w}_1, \cdots, \mathbf{w}_{n-1}$. Now, we want to compute the distance between $\mathbf{s}$ and $\mathbf{w}_n$. Instead of going through the full length of the feature vector to obtain the distance, we move forward one-by-one and check if the distance is larger than the minimum distance d_m. That is, for $l = 1, \cdots, k$, we calculate $d_l(\mathbf{s}, \mathbf{w}_n) = \sum_{l'=1}^{l}(s_{l'} - w_{jl'})^2$ or $d_l(\mathbf{s}, \mathbf{w}_n) = d_{(l-1)}(\mathbf{s}, \mathbf{w}_n) + (s_l - w_{jl})^2$. Once $d_l(\mathbf{s}, \mathbf{w}_n) \geq d_m$, the matching operation is terminated, and the matching $\mathbf{s}$ with next word $\mathbf{w}_{n+1}$ will be started. If the computation for $d(\mathbf{s}, \mathbf{w}_n)$ is completed and it is less than d_m, the d_m value is replaced by $d(\mathbf{s}, \mathbf{w}_n)$ and we move on to the next matching. In this way, on average, the computational cost can be reduced by half (*i.e.* $0.5kMN$).

The image set for training SVM classifiers are collected in two batches. The first batch comes from our early dataset



Public Area (*Mall*)	**Transition Area** (*LiftLobby*)	**Working Area** (*OfficeZone*)
Mall: MainEntrance, Shops, Gates (to Towers), Mall (Lobby), FoodCourt	*LiftLobby: ReceptionTable, LiftLobby, LiftDoor, InfomationBoard, InLift, Entrance (to office).*	*Office: EntranceToOffice, Junction, Corridor, ExitToLiftLobby, Cubicle, MeetingRoom, Pantry.*

Fig. 3. The implementation of the hierarchical context model. Top row: three area nodes; Mid-row: area node (red) and its child nodes of locations (green); Images: sample scene images of the locations from the corresponding area, where in each image the overlapping red text indicates the area and the green text denotes locations in the area.

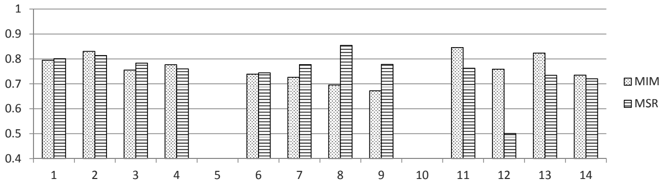


Fig. 4. The accuracy rates of image recognition by two compared methods on the 12 test sequences of three routes.

to investigate indoor scene classification [59]. The second batch is collected for investigating indoor navigation. Two experimenters used a head-mounted camera as shown in Figure 2, and walked along several routes from outside the main entrance of the building to a few meeting rooms at the different levels. The training images are randomly selected from the videos. Totally, a set of about 1500 sample images are collected to train the SVMs for scene recognition (areas and locations), which is far less than 80,000 images used for an indoor office environment in one floor described in [13], [34]. Exploiting the cognitive concepts of indoor scenes can alleviate the need of an exhaustive survey of the building to define the survey knowledge because the indoor scenes of the same category in a building are similar.

The evaluation of vision recognition is performed using a test set. To build the test dataset, we ask four users to wear our system and collect the image sequences along three challenging routes for navigation (see explanation later in subsection IV-D). The speeds and viewing angles along the routes from different users vary significantly, which is close to a practical wayfinding scenario. In the experiment, we compared the SVM-based scene recognition (MSR) with a state-of-the-art baseline method on image matching (MIM). In MIM, we use a standard CBIR pipeline with SIFT features as in [35]. We find a short list of candidates with an inverted file system, followed by geometric consistency checks with RANSAC on top matches. If no image passes RANSAC, the top match from the IFS is declared to be a match [66]. The images in training set for training the SVM models are used as the image database. In MSR, we train 18 SVM classifiers for each locations (*i.e.* the leaf nodes in context model) independently for location recognition.

We run the two methods on the 12 test sequences. For each frame, MIM compares it with all the images in the database and selects the best matching image to represent the location, and MSR applies the 18 SVM classifiers to the image and selects the winner class to represent the location. The results are compared with the ground truth, which is derived based on manual annotation. It is observed that there is no clear separation between connected locations from the first-person-view in the indoor environment. Hence, we design a matching measure between the recognition and ground truth. Let t_s^l and t_e^l be the start and end frames of the segment of location l in a test sequence, respectively, and d be a parameter to represent the length of transfer areas in the sequences ($d = 20$ frames in this experiment). Then, if a frame I_t is recognized as location l but it is out of the ground truth segment and the closest frame from the segment is t_s^l, its distance to the segment can be measured as $r_t = e^{-|t-t_s^l|^2/d^2}$. If I_t is within the segment, r_t is set as 1. The accuracy rate is computed as the average of r_t over the whole sequence. The complete evaluation on 12 sequences are shown in Figure 4. On average, the accuracy of MIM is 76.29% and that of MSR is 75.25%, respectively. It can be seen that accurate indoor scene recognition is very challenging.

C. Evaluations on Dynamic Model

The proposed SA-DBN is intended to estimate a user's location in a trip according to the route model and provide real-time context-aware guidance for indoor navigation. In such an application scenario, maintaining high accuracy is crucial. Otherwise, the system might provide irrelevant voice instructions. A few public datasets of indoor image sequences were built for scene recognition and image retrieval [34], [35], [42]. They are not used to evaluate navigation performance because no route model can be established. We evaluated our SA-DBN model on three challenging routes for navigation (see explanation later in IV-D). In this experiment, we evaluated the dynamic models on the test dataset aforementioned. The speeds along the routes from different users vary considerably. For example, the longest traveling time is over 1.5 times of the shortest one for each route.

To compare with the state-of-the-art, we benchmark our method against two baseline methods based on HMM, which can be considered as the probabilistic version of existing wayfinding process models [1], [28], [29]. Baseline approach 1 (B1:IM-HMM) is based on image matching where MIM is employed, and Baseline approach 2 (B2:SVM-HMM) is based on multi-class SVM classification where the 18 SVM classifiers are used. The method B2 is similar to [42]. To evaluate the benefit of online self-adaption in SA-DBN, we also compare it with A-DBN where the parameters are fixed.

Evaluations are carried out on the 12 test sequences. For each sequence, we apply MIM and MSR to each frame image first. Then, we obtain four sequences of location labels (1 to 18) for each test route. We use the leave-one-out cross validation to train and test the HMM models for each route. That is, for each test, we select 3 sequences to train the HMM and use the remaining sequence to test the HMM. The estimated state sequence generated by the trained HMM is compared to the ground truth. The test is repeated for all possible arrangements. Our system is tested on all 12 test sequences. For each frame, our method performs two-level area/location recognition according to the hierarchical context model. Next, the SA-DBN is used for state estimation, and its parameters are updated. Unlike HMM-based methods, we do not train the SA-DBN on the dataset for each specific route.

Figure 5 shows the performance of four approaches. On average, the accuracy of SA-DBN is 92.9%, and those of B1, B2 and A-DBN are 79.5%, 78.5%, and 71.0%, respectively. Note that the HMMs are run in an offline mode while DBNs are run in an online mode. A HMM model is fed the full test sequence after the image recognition step. The Viterbi algorithm performs forward and backward propagation to refine the state estimation iteratively. Hence, the results can be considered as an upper bound on performance of HMM based approaches provided that only the forward inference is applied. A DBN just performs forward inference as the sequence progresses. Therefore, it provides online state estimation. When the parameters are fixed in A-DBN, a large variation of travel time duration in any node (*e.g.* staying longer in the lift lobby to wait for an available lift), results in a large shift of mid-time points of subsequent nodes, which in turn results in failures in state estimation.

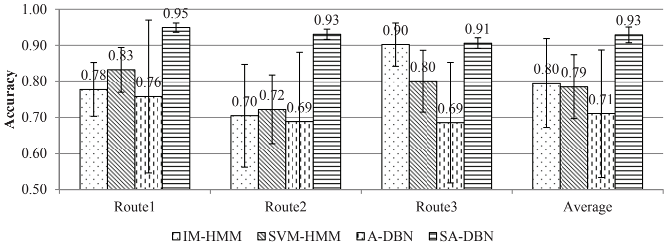


Fig. 5. The accuracy rates of state estimations of four compared methods on each test route and the average over all test routes.

To have a better understanding on our model, evaluations are also performed on some specific scenarios such as abrupt head movements, great variations of walking speeds, lighting variations and crowded scenarios.

One distinctive advantage of SA-DBN model is self-adaption to the dynamic procedure. Walking very fast or halting somewhere will not cause significant performance loss on state estimation since SA-DBN online tunes the duration and variance of the trip in a segment as described by equations (13) to (16). To have a close examination on the self-adaption operations of SA-DBN, we show the variations of the parameters in two extreme conditions, namely a very short and a very long trip duration along the same trip segment (i.e., from outside the Main Entrance to the lift lobby after the glass door). The first person spent about 8s to complete the trip segment and the second person used about 23s, almost 3 times longer than the first one. The plots of the predicted mid-time point t_k^t and variance σ_k^t of two trips are displayed in Figure 6. It can be seen that, when the user enters the segment, the predicted duration decreases from the average value, so that if the user walks fast, the state may transform to the next trip segment quickly, whereas if the user stays at the segment, the predicted duration and variance are increased gradually till the user leaves the segment. Hence, the walking speed of the user will not cause problems for state estimation by SA-DBN.

As normal vision systems, the training images are collected from a range of normal viewing angles to the indoor scenes. If the user observes a scene from unusual viewing angles, e.g. when looking up at the ceiling, looking down at the ground plane, or looking at a wall in a close distance, the performance

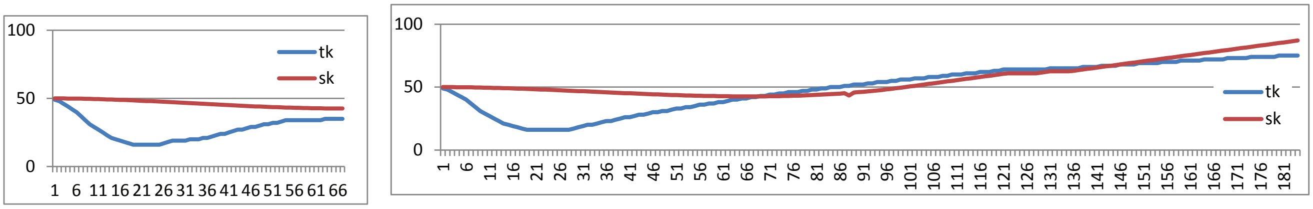


Fig. 6. The plots of adaptive parameters of SA-DBN model for extreme short and long trips passing through the MainEntrance segment, where ‘tk’ indicates t_k^t and ‘sk’ represents σ_k^t in (14).

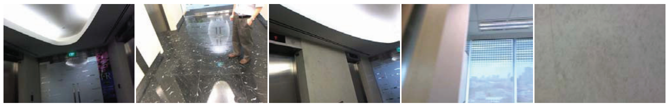


Fig. 7. Examples of unusual views of scenes. From the left to the right: looking at ceiling, looking at ground plane, looking at the corner of ceiling and wall, looking at the upper corner of window, looking at wall closely.

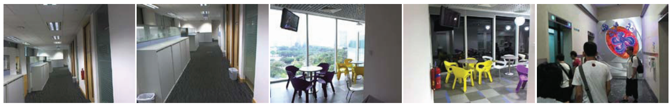


Fig. 8. Examples of scenes from day, night and crowded scenarios. First two images: corridor in the day and in the nigh; next two images: pantry in the day and in the nigh; right image: crowded scene in lift lobby.

of scene recognition would drop since such input images are not distinctive and informative for scene identification. However, the errors caused by unusual viewing angles appear as random noises. SA-DBN model exploits the knowledge of the dynamic procedure and previous history to constrain the state estimation. It is very robust to random errors. In the evaluation on user study data in the next subsection, especially on Route 3, one can observe such advantage. For a strict evaluation, we especially performed a test on Route 3 in which the user frequently moved his head and intentionally turn his viewing points to ceilings, ground surfaces, walls, corners and close objects, *etc*. A few example images of such unusual views of indoor scenes are displayed in Figure 7. Our system succeeded to guide the user to the destination. The accuracy rate of scene recognition drops to 68.91%, whereas the accuracy rate of state estimation remains at 91.08%, similar to the other tests on Route 3.

Similar to the problem of viewing angles, when building a vision system, it is required to collect the training samples of different lighting conditions and crowded scenarios. In testing, if the input images are different with training samples greatly, the recognition rate would drop. Collecting exhaustive training images will be a big labor burden for system build-up. For our system, there are four factors which can alleviate such burden. First, indoor areas of a commercial building are illuminated by artificial lighting, which are almost constant from day to night. Second, our scene recognition is based on gradient features (*e.g.* SIFT), which is robust to lighting changes [67]. Third, the scene classifiers are trained for scene categories, so that there is no need to collect images of different viewing angles and lighting conditions at every positions in the building. In Figure 8, the first four images show the examples of indoor scenes in the building captured in the day and in the nigh. Even though our training set does not include images captured in night, the trained SVM model recognizes the scene correctly. The fourth factor is that, the FPV captures a vantage point to the scene aligned with the wearer’s view. If there are crowded people in front of the user in corridor and hallway, normally less than 50% of the scene might be occluded in the FPV images. Hence, the performance of the scene recognition would not degrade too much. One example of crowded scenes in lift lobby is shown in Figure 8. In these cases, the trained SVM classifiers work normally. On the other hand, even though large lighting changes and over crowded scenarios might cause a drop of scene recognition performance, the DBN model is robust enough to mitigate from such errors in scene recognition.

D. User Study in Practical Tasks

To evaluate our system in real-world application scenarios, user studies are conducted with subjects who are new to the building. The users are asked to find their way to multiple destinations in the building using different navigation supports: a 2D map (MAP), a wearable virtual usher (WVU, *i.e.* our system), and a human usher (HUR). The purposes of the user study are twofold: (1) to investigate whether the wearable virtual usher can work in real-world tasks of indoor navigation, and (2) to compare its usability to two existing navigation modes. Three routes traversing three different areas in the building are used in the test. Each participant is required to go to three destinations in sequence: (1) from main entrance outside of the building (denoted as $M0$) to a meeting room (denoted as $M1$) in the office area, (2) from $M1$ to another meeting room (denoted as $M2$) in the office area, and (3) from $M2$ to a meeting room (denoted as $M3$) in a meeting facility area. $M0 \sim M3$ are located at different levels and in different parts (towers) of the building. In each route, the user traverse over 10 *locations* in 3 *areas*. The complexity level of the three routes is similar. The topological illustration of Route 1 is presented in Figure 9.

From the hierarchical context model, a route model is generated for each test route. The complete route model of Route 1 is illustrated in Figure 10. Since each route has a destination, a virtual area node of *destination* is introduced as node A_4 which has only one child location node of destination instance (S_{10}).

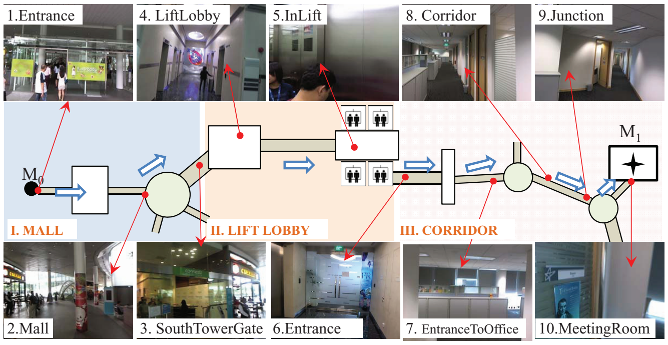


Fig. 9. The topological illustration of test Route 1.

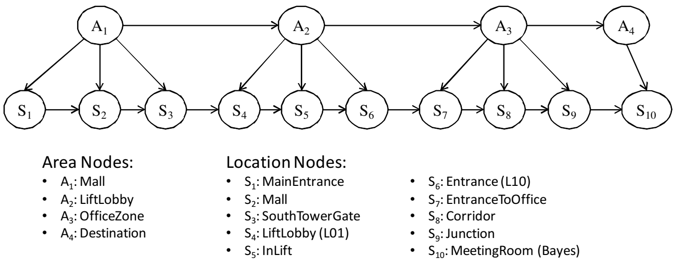


Fig. 10. The complete route model of test Route 1.

TABLE I
VERBAL INSTRUCTIONS FOR NAVIGATION ALONG ROUTE 1.

Location	Instructions
S1: MainEntrance	V1.1: Welcome. Lets begin. We will go to Bayes at level ten. Please walk through the glass doors. V1.2: Please go into the glass doors.
S2: Mall	V2.1: Please turn left towards the South Tower gate. V2.2: Please walk to the South tower gate.
S3: South-TowerGate	V3.1: You are facing south tower gate. Please go ahead to the lift lobby. V3.2: Please go to the lift lobby and take a lift.
S4: LiftLobby	V4.1: Take a lift on the left side; and go to level ten. The lift on the right does not go to level ten. V4.2: Take a lift to level ten. Please look at the top right of the lift door to see if the lift stops at level ten.
S5: InLift	V5.1: Please remember to alight at level ten. V5.2: Please remember to alight at level ten.
S6: Entrance	V6.1: Please stand in front of the information board and check if you are at level ten. V6.2: Proceed to the glass door and get in.
S7: Entrance-ToOffice	V7.1: Turn right after you enter the glass door. V7.2: Please make a right turn after the glass door.
S8: Corridor	V8.1: Go straight ahead for about 30 meters. V8.2: Please go ahead until the junction.
S9: Junction	V9.1: We are close to Bayes. It is the room on the left side of the junction. V9.2: Bayes room is around. Please check the name on the door.
S10: MeetingRoom	V10.1: Here we are. You have reached meeting room Bayes. Congratulations! V10.2: You have reached meeting room Bayes. Congratulations!

Note: V means verbal instruction or voice prompt; Vi.1 means the first verbal instruction when the user is entering or approaching the trip segment S_i, and Vi.2 means the repeat voice prompt if the user has stayed in the trip segment S_i over the half of the duration to travel through the trip segment.

Our system tracks the user's location and gives online context-aware guidance as the user travels through each trip segment, *e.g.* when he/she is approaching a junction, the system guides the user to turn left and go to the next location using a voice prompt. The voice prompt is repeated till the user leaves the segment. In this way, our system turns the navigation mode from *unaided wayfinding* to *aided wayfinding* [15]. The full list of verbal instructions for navigation along Route 1 are presented in Table I.

Two batches of user studies have been conducted. In the first batch, we recruited 12 participants. The results were reported in [14]. We extended the user study by recruiting 20 more participants. The combined results are reported below. Totally, 32 participants (19 male; aged between 18 and 61 years) are involved in this study. The participants have never been to the building and have no prior knowledge of its internal layout. Each subject was required to perform the three navigation tasks sequentially using three navigation modes, *i.e.* MAP, WVU, and HUR, respectively. The order of the modes was selected randomly and counter-balanced against the order of routes for each subject. The procedure of the experiment is as follows. After a brief explanation and informed consent, the subject is led to the starting point. The wearable system is attached to the participant, who performs the three navigation tasks as described above. Upon completion of each individual task, a questionnaire is given to the subject to collect his/her feedback on the navigation mode. The questionnaire is designed to evaluate the user's experience on six measurements: *usefulness* (S-Use), *ease-of-use* (S-Eou), *enjoyment* (S-Enj), *stress level* (S-Stress), *intelligence* (S-Int), and *trust* (S-Trust).

The virtual usher successfully guided the user to the destinations in all tests. We examine the accuracy of the WVU by checking the number of errors it made. These include inappropriate or wrong instructions. An inappropriate instruction refers to one that is not synchronized well with the user's action. For example, a user may act faster than expected so that he hears "*Please enter the glass door and turn right*" when he is already in the door. In the experiment, users could easily cope with such inconsistencies and were not misled. A wrong instruction refers to one that guide a user to the wrong direction. This did not happen in this study.

The evaluation results on the questionnaires and the comparison between three navigation modes are shown in Figure 11. We conducted Friedman test with post-hoc analysis using Wilcoxon signed ranks test with Bonferroni correction (a significant level set at $p < .017$) for evaluating the subjective measures. We found significant differences in all six subjective measures across three conditions. Post-hoc analysis showed a positive effect of the CNGs context-aware capability on the user experience as compared to the MAP, except for the measurements of enjoyment, stress, and trust (Figure 11). In particular, users perceived the cognitive navigation guide to be more *useful* than the 2D map [S-Use (CNG, MAP): $z = 2.466, p = .014$]; it was *easier-to-use* [S-Eou (CNG, MAP): $z = 2.795, p = .005$]; and it was considered to be more *intelligent* [S-Int (CNG, MAP): $z = 2.608, p = .009$]. However, the *enjoyment* level was identical in both conditions [S-Enj(CNG, MAP):$z = 1.053, p = .292$]. No statistical

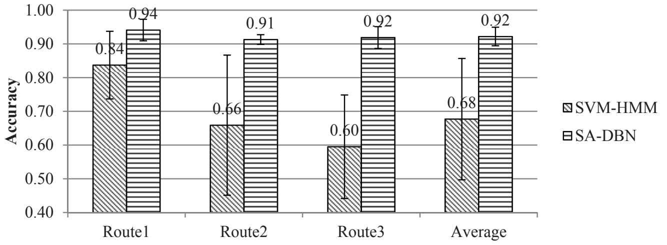


Fig. 12. The accuracy rate of state estimation on the test data set from the real-world user study.

difference was found with respect to the *stress* level [S-Stress (CNG, MAP): $z = 1.22, p = .222$] and *trust* [S-Trust (CNG, MAP): S-Trust: $z = 1.582, p = .114$]. As compared to the human guide, the CNG achieved equivalent performance in terms of *usefulness* [S-Use (CNG, HUG): $z = 2.003, p = .041$] *ease-of-use* [S-Eou (CNG, HUG): $z = 2.133, p = .033$], and the *enjoyment* level [S-Enj (CNG, HUG): $z = -2.384, p = 0.019$]. It was inferior to a human guide in terms of *perceived stress* [S-Stress (CNG, HUG): $z = 3.056, p = .002$] and *trust* [S-Trust (CNG, HUG): $z = 3.070, p = .002$], and *intelligence* [S-Int (CNG, HUG): $z = 4.298, p = .000$].

It can be observed that the users consider the wearable virtual usher better than 2D map on *usefulness*, *ease-of-use* and *intelligence*. It is close to a human usher with respect to *usefulness*, *ease-of-use* and the *perceived enjoyment* level. The scores for *perceived stress*, *intelligence* and *trust* are lower than the HUR condition because WVU is still a new paradigm, that is not familiar to the user.

In this real-world user study, participants were completely unfamiliar with the building environment. As a result, there was a much higher variation in viewing angles, motion blur, trip times, *etc.*, in comparison to the data set collected in testing experiments with staffs in this building. The large difference in the way of observing the environment by different users makes the tests more practical. In the first batch, we also recorded the image sequences during automated navigation using WVU. A quantitative evaluation of SA-DBN on the 12 recorded sequences is performed and compared with B2:SVM-HMM, where the HMM models are trained using the first dataset in IV-C. The results are shown in Figure 12. On average, our system achieves 92.2% accuracy for state estimation while that of SVM-HMM drops to 67.7%. Comparing Figure 12 and Figure 5, one can find that the performance of our method remains stable while that of Baseline 2 drops significantly. This shows the robustness of our SA-DBN model in practical real-world scenarios. Videos of the tests are available at (*http://perception.i2r.a-star.edu.sg/Navigation/Navigation.htm*).

E. Discussion

In this research, no geographic information (such as floor plans) is exploited. The wayfinding and navigation are conducted solely based on the cognitive concepts of indoor scenes observed from FPV. Apparently, if more information on spatial location is employed, the performance of indoor localization and navigation can be further improved.

Nowadays, WiFi-based positioning is most popular for real-world deployment because the infrastructure is available in many public buildings [5], [68]. The typical indoor position accuracy of WiFi localization is within 2 to 7 meters [5], [68], [69]. To further improve the accuracy to meet commercial requirements, expensive initial deployment is required, *e.g.* adding more access points, building and maintaining a huge database of geographic locations of WiFi access points, and performing a comprehensive survey to record WiFi signals.

On the other hand, we believe that vision- and sensor-based techniques may complement each other for indoor navigation [8]. In a broad sense, vision itself is a sensory modality that is characterized by the rich information and ubiquity in various environments. This study has shown that, exploiting visual perception on FPV images and cognitive knowledge of wayfinding, as well as simple speech instruction on the view aligned with the user, it may be easy to achieve natural navigation without the requirement of accurate localization. Integrating with other types of sensors, such as WiFi, infrared, auditory, *etc.*, one may develop practical cognitive systems that changes its perceptual sensitivity according to task requirements. For example, one can exploit cognitive vision based visual spatial perception and WiFi localization to develop low-cost, practical systems for navigation and other location based services (LBS) without requiring high accuracy on each modality.

V. Conclusions

In this paper, we present a wearable vision system for real-time cognitive indoor navigation based on FPV image recognition. Since the user wears the camera on his/her forehead, the captured FPV images align with the egocentric view of the user. Our cognitive models for navigation are formulated based on egocentric perspective observations which are directly linked to the perceptual experience of the user [12]. Thus, the system can be naturally incorporated into the human wayfinding process.

The proposed method relies on a computational framework for cognitive wayfinding and a DBN-based process model. Under the proposed framework, both the knowledge representation and wayfinding process models are integrated. A hierarchical context model for survey knowledge is built on the cognitive concepts of indoor scenes. The probabilistic models are introduced to represent the route and process models. A Bayesian Network model is employed to represent the route knowledge. The proposed SA-DBN model plays an important role in achieving high performance in the challenging navigation tasks. The SA-DBN model combines cognitive knowledge for wayfinding, visual perception, and an online-tuning mechanism to realize reliable navigation in dynamic scenarios. Detailed evaluations show the superior performance of our model over state-of-the-art vision approaches. The results of the user studies show the effectiveness of our system for indoor navigation in real-world scenarios. In future, we plan to extend

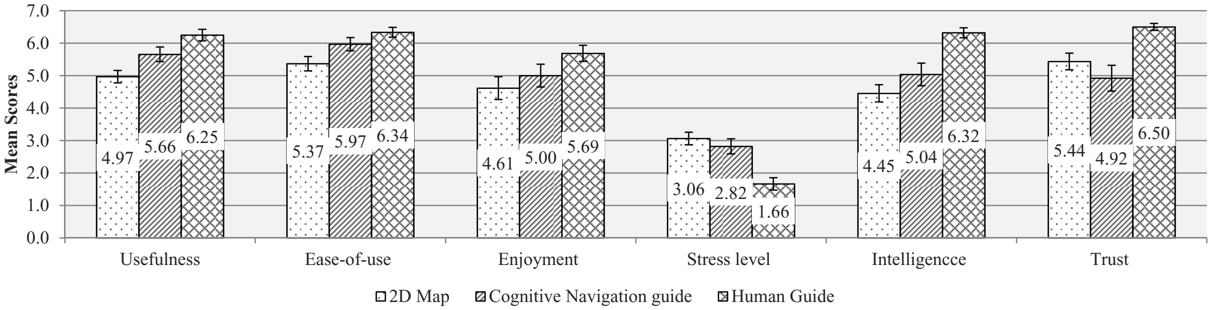


Fig. 11. Statistics of the survey on users' experience of using three assistant approaches for indoor navigation.

the context model to involve more spatial information for localization, and improve the process model to incorporate the scenario where the user does not follow the guidance and deviates from the planned route. Meanwhile, to make the system truly wearable, we intend to use smaller devices with adequate computing power, e.g. the proposed system can be deployed using smart phone with a wearable camera such as Google Glass. Finally, we plan to investigate the effectiveness of the system to help children and senior citizens, to better understand how people with varying cognitive abilities interact with such a system.

REFERENCES

[1] M. Raubal and M. Worboys, "Human wayfinding in unfamiliar buildings: A simulation with cognizing agents," in *International Conference on Spatial Cognition: Scientific Research and Applications (ICSC)*, 2000.

[2] H. Huang and G. Gartner, "A survey of mobile indoor navigation systems," in *Lecture Notes in Geoinformation and Cartography*. Springer, 2010, pp. 305–319.

[3] N. Fallah, I. Apostolopoulos, K. Bekris, and E. Folmer, "Indoor human navigation systems - a survey," *Interacting with Computers*, vol. 25, no. 1, pp. 21–33, 2013.

[4] P. Heiniz, K.-H. Krempels, C. Terwelp, and S. Wuller, "Landmark-based navigation in complex buildings," in *Int'l Conf. Indoor Positioning and Indoor Navigation (IPIN)*, 2012, pp. 1–5.

[5] D. Schneider, "You are here," *IEEE Spectrum*, December 2013.

[6] C. Freksa, A. Klippel, and S. Winter, "A cognitive perspective on spatial context," *Dagstuhl Seminar Proceedings 05491*, pp. 1–16, 2005.

[7] N. Giudice, J. Bakdash, K. Legge, G. Folmer, and R. Roy, "Spatial learning and navigation using a virtual verbal display," *Interacting with Computers*, vol. 7, no. 1, pp. 3:1–3:22, 2010.

[8] J. F. Ferreira, J. Lobo, P. Bessière, M. Castelo-Branco, and J. Dias, "A bayesian framework for active artificial perception," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 699–711, 2013.

[9] R. Wang and E. Spelke, "Updating egocentric representations in human navigation," *Cognition*, vol. 77, pp. 215–250, 2000.

[10] S. Creem-Regehr, "Remembering spatial locations: The role of physical movement in egocentric updating," in *In Gary L. Allen (Ed.), Human Spatial Memory: Remembering Where*. Erlbaum, 2004, pp. 163–190.

[11] H. Taylor, T. Brunye, and S. Taylor, "Spatial mental representation: Implications for navigation system design," *Reviews of Human Factors and Ergonomics*, vol. 4, no. 1, pp. 1–40, 2008.

[12] S. Werner, B. Krieg-Bruckner, H. Mallot, K. Schweizer, and C. Freksa, "Spatial cognition: The role of landmark, route, and survey knowledge in human and robot navigation," in *Informatik aktuell*, 1997, pp. 41–50.

[13] T. Kanade and M. Hebert, "First-person view," *Proceedings of IEEE*, vol. 100, no. 8, pp. 2442–2453, 2012.

[14] Q. Xu, L. Li, J. Lim, C. Tan, M. Mukawa, and G. Wang, "A wearable virtual guide for context-aware cognitive indoor navigation," in *Mobile-HCI'14*, 2014, pp. 111–120.

[15] J. Wiener, S. Buchner, and C. Holscher, "Towards a taxonomy of wayfinding tasks: A knowledge-based approach," *Spatial Cognition and Computation*, vol. 9, no. 2, pp. 152–165, 2009.

[16] H. Qiao, Y. Li, T. Tang, and P. Wang, "Introducing memory and association mechanism into a biologically inspired visual model," *IEEE Trans. Cybern.*, vol. 44, no. 9, pp. 1485–1495, 2014.

[17] P. Narayan and D. Campbell, "Embedding human expert cognition into autonomous uas trajectory planning," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 530–543, 2013.

[18] T. Kitamura and D. Nishino, "Training of a learning agent for navigation – inspired by brain-machine interface," *IEEE Trans. EEE Trans. Syst., Man, Cybern. B Cybern.*, vol. 36, no. 2, pp. 353–365, 2006.

[19] J. Wiener and H. Mallot, "Fine-to-coarse route planning and navigation in regionalized environments," *Spatial Cognition and Computation*, vol. 3, no. 4, pp. 331–358, 2003.

[20] J. Wang, H. Zha, and R. Cipolla, "Coarse-to-fine vision-based localization by indexing scale-invariant features," *IEEE Trans. EEE Trans. Syst., Man, Cybern. B Cybern.*, vol. 36, no. 2, pp. 413–422, 2006.

[21] S. Nayak, V. Mishra, and A. Mukerjee., "Towards a cognitive model for human wayfinding behavior in regionalized environments," in *AAAI Fall Symposium: Advances in Cognitive Systems*, 2011.

[22] B. Kuipers, "Modeling spatial knowledge," *Cognitive Science*, vol. 2, pp. 129–154, 1978.

[23] D. Mark, C. Freksa, S. Hirtle, R. Lloyd, and B. Tversky, "Cognitive models of geographical space," *International Journal of Geographical Information Science*, vol. 13, no. 8, pp. 747–774, 2000.

[24] R. P. Darken, B. Peterson, and B. S. Orientation, "Spatial orientation, wayfinding, and representation," in *In K. M. Stanney (Ed.), Handbook of Virtual Environments: Design, Implementation, and Applications*. Erlbaum, 2001, pp. 493–518.

[25] J. Weisman, "Evaluating architectural legibility: Way-finding in the built environment," *Environment and Behavior*, vol. 13, pp. 189–204, 1981.

[26] M. O'Neill, "A biologically based model of spatial cognition and wayfinding," *Journal of Environmental Psychology*, vol. 11, pp. 299–320, 1991.

[27] M. Raubal, "A formal model of the process of wayfinding in build environments," in *Lecture Notes in Computer Science*, vol. 1661, 1999, pp. 381–399.

[28] M.-P. Daniel and M. Denis, "Spatial descriptions as navigational aids: A cognitive analysis of route directions," *Kognitionswissenschaft*, vol. 7, no. 1, pp. 45–52, 1998.

[29] K.-F. Richter and A. Klippel, "A model for context-specific route directions," in *International Conference Spatial Cognition*. Springer, 2004, pp. 58–78.

[30] C. Heye and S. Timpf, "Factors influencing the physical complexity of routes in public transportation networks," in *10th International Conference on Travel Behaviour Research*, 2003.

[31] M. Duckham and L. Kulik, "Simplest paths: Automated route selection for navigation," in *COSIT 2003. Spatial information theory*. Springer, 2003, pp. 169–185.

[32] K. Baras, A. Moreira, and F. Meneses, "Navigation based on symbolic space models," in *Int'l Conf. Indoor Positioning and Indoor Navigation (IPIN)*, 2010, pp. 1–5.

[33] R. Sim and G. Dudek, "Comparing image-based localization methods," in *Proc. Int'l J Conf. Artificial Intelligence*, 2003, pp. 1560–1562.

[34] H. Kang, A. Efros, M. Hebert, and T. Kanade, "Image matching in large scale indoor environment," in *CVPR Workshop on Egocentric Vision*, 2009.

[35] R. Huitl, G. Schroth, S. Hilsenbeck, F. Schweiger, and E. Steinbach, "Virtual reference view generation for cbir-based visual pose estimation," in *ACM MM*, 2012, pp. 993–996.

[36] Y. Matsumoto, K. Sakai, M. Inaba, and H. Inoue, "View-based approach to robot navigation," in *IEEE/RSJ Int'l Conf. Intelligent Robots and Systems*, vol. 3, 2000, pp. 1702–1708.

[37] Y. Kaneko and J. Miura, "View sequence generation for view-based outdoor navigation," in *ACPR*, 2011, pp. 139–143.

[38] O. Koch and S. Teller, "Body-relative navigation guidance using uncalibrated cameras," in *ICCV*, 2009, pp. 1242–1249.

[39] A. Murillo, D. Gutierrez-Gomez, A. Rituerto, L. Puig, and J. Guerrero, "Wearable omnidirectional vision system for personal localization and guidance," in *CVPRW*, 2012, pp. 8–14.

[40] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *CVPR*, 2009, pp. 413–420.

[41] C. Siagian and L. Itti, "Rapid biologically-inspired scene classification using features shared with visual attention," *PAMI*, vol. 29, no. 2, pp. 300–312, 2007.

[42] A. Torralba, K. Murphy, W. Freeman, and M. Rubin, "Context-based vision system for place and object recognition," in *ICCV*, 2003, pp. 273–280.

[43] K. Ni, A. Kannan, A. Criminisi, and J. Winn, "Epitomic location recognition," *PAMI*, vol. 31, no. 12, pp. 2158–2167, 2009.

[44] H. Durrant-Whyte and T. Bailey, "Simultaneous localisation and mapping (slam): Part i the essential algorithms," *Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[45] ——, "Simultaneous localisation and mapping (slam): Part ii state of the art," *Robotics and Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[46] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. Rendon-Mancha, "Visual simultaneous localisation and mapping: A survey," *Artificial Intelligence Review*, 2012.

[47] R. Castle, D. Gawley, G. Klein, and D. Murray, "Towards simultaneous recognition, localization and mapping for hand-held and wearable cameras," in *Proc. Int'l Conf. Robotics and Automation (ICRA)*, 2007, pp. 4102–4107.

[48] M. Milford and G. Wyeth, "Mapping a suburb with a single camera using a biologically inspired slam system," *IEEE Trans Robot*, vol. 24, no. 5, pp. 1038–1053, 2008.

[49] P. Stano, Z. Lendek, J. Braaksma, R. Babuska, C. de Keizer, and A. J. den Dekker, "Parametric bayesian filters for nonlinear stochastic dynamical systems: A survey," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1607–1622, 2013.

[50] Y. Xiang and M. Truong, "Acquisition of causal models for acquisition of causal models for local distributions in bayesian networks," *IEEE Trans. Cybern.*, vol. 44, no. 9, pp. 1591–1604, 2014.

[51] K. Murphy, "Dynamic bayesian networks: Representation, inference and learning," Ph.D. dissertation, UC Berkeley, 2002.

[52] P. Natarajan, V. Singh, and R. Nevatia, "Learning 3d action models from a few 2d videos for view invariant action recognition," in *CVPR*, 2010, pp. 2006–2013.

[53] Z. Zeng and Q. Ji, "Knowledge based activity recognition with dynamic bayesian network," in *ECCV*, 2010, pp. 532–546.

[54] A. Pfeffer and T. Tai, "Asynchronous dynamic bayesian networks," in *UAI (arXiv preprint arXiv:1207.1398)*, 2005.

[55] R. Golledge, *Wayfnding Behavior: Cognitive Mapping and Other Spatial Processes*. The Johns Hopkins University Press, 1998.

[56] S. Timpf, G. Volta, and D. Pollock, "A conceptual model of wayfinding using multiple levels of abstraction," in *Theory and Methods of Spatio-Temporal Reasoning in Geographic Space*, A. Frank, I. Campari, and U. Formentini, Eds., vol. 639, 1992, pp. 348–367.

[57] A. Redish, *Beyond The Cognitive Map: From Place Cells to Episodic Memory*. Cambridge: MIT, 1999.

[58] D. Norman, *The Design of Everyday Things*. New York: Doubleday, 1988.

[59] L. Li, W. Goh, J.-H. Lim, and S. Pan, "Extended spectral regression for efficient scene recognition," *Pattern Recognition*, vol. 47, pp. 2940–2951, 2014.

[60] B. Landau and R. Jackendoff, ""what" and "where" in spatial cognition," *Behavioral and Brain Sciences*, vol. 16, pp. 217–265, 1993.

[61] S. Hansen, K.-F. Richter, and A. Klippel, "Landmarks in openls — a data structure for cognitive ergonomic route directions," in *International Conference on Geographic Information Science*, 2006, pp. 128–144.

[62] G. Allen, "Spatial abilities, cognitive maps, and wayfinding - bases for individual differences in spatial cognition and behavior," in *Wayfinding Behavior - Cognitive Mapping and Other Spatial Processes*. Johns Hopkins University Press, 1999, pp. 46–80.

[63] N. Zhang and D. Poole, "Exploiting causal independence in bayesian network inference," *Journal of Artificial Intelligence Research*, vol. 5, pp. 301–328, 1996.

[64] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Pearson Education, 2003.

[65] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2. IEEE, 2006, pp. 2169–2178.

[66] B. Girod, V. Chandrasekhar, D. Chen, N.-M. Cheung, R. Grzeszczuk, Y. Reznik, G. Takacs, S. Tsai, and R. Vedantham, "Mobile visual search," *IEEE Signal Process. Mag.*, vol. 28, no. 4, pp. 61–76, 2011.

[67] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. IEEE, 1999, pp. 1150–1157.

[68] G. Jekabsons, V. Kairish, and V. Zuravlyov, "An analysis of wi-fi based indoor positioning accuracy," *Scientific Journal of Riga Technical University*, vol. 47, pp. 131–137, 2011.

[69] G. Retscher, E. Moser, D. Vredeveld, D. Heberling, and J. Pamp, "Performance and accuracy test of a wifi indoor positioning system," *Journal of Applied Geodesy*, vol. 1, no. 2, pp. 103–110, 2007.