Characterization of Vision-aided Indoor Localization and Landmark Routing

Aaron Ballew, Shiva Srivastava, Nikolay Valtchanov, C.C. Lee, and Aleksandar Kuzmanovic
Dept. of Electrical Engineering and Computer Science
Northwestern University
Evanston, IL

***Abstract*—GPS is the premier method of localization and wayfinding in outdoor environments. Indoor environments prevent GPS from functioning properly or at all. RF-based solutions have been proposed using varieties of radio triangulation and spectral fingerprinting. Such technically attractive methods have not seen widespread adoption in places where the demand is greatest: supermarkets, shopping malls, airports, etc. This may be due to the obstacles of funding, installing, or accessing sufficient wireless infrastructure for triangulation, as well as the scalability challenge in site-by-site fingerprinting.**

We present a framework for characterizing indoor environments. This leads to a simple, practical approach to indoor localization and wayfinding that takes advantage of visibility relationships, limited user-input, publicly available online floorplans, and lightweight processing on a mobile phone. On-site infrastructure and site-surveys are avoided. Based on this framework we find that localization precision improves dramatically as the population of landmarks grows, though few landmarks are needed during a given input cycle. We also find that indoor wayfinding is unexpectedly insensitive to hop-count, but critically dependent on path connectedness.

***Keywords*-localization; wayfinding; vision; landmark;**

I. Introduction

The Global Positioning System (GPS) has become commonplace in consumer life, most prominently applied to vehicle navigation, and increasingly found in mobile phone applications. Outdoors, GPS performs exceptionally well. A conspicuous weakness of GPS is its poor performance indoors. Typically, no connection can be made, and the mobile unit is untrackable. This has generated a body of creative, promising solutions to indoor localization. Despite their potential, none have seen widespread adoption. In broad terms, these methods are radio (RF)-based, and rely on some form of triangulation or fingerprinting [1], [2], [5], [10], [12], [13].

Triangulation operates off of radio beacons deployed within or around the indoor environment. Some proposals address this by reusing existing beacons, (e.g., wi-fi, cellular) [2], but a typical supermarket may have only one or few wi-fi APs. Indoors, cell triangulation is imprecise. RF fingerprinting compares a location's spectral characteristics to a predetermined map of radio signatures [1], [5], [10], [12], [13]. While RF fingerprinting requires no infrastructure, considerable effort is needed to build the signature map for each new environment.

In this paper, we explore the characteristics of indoor environments in the context of an application driven by user input. This leads to a simple, practical method for indoor localization and wayfinding which, running on a mobile phone, queries the user for limited information about what he sees. The application displays the likely location on a map, and the user is then directed to proceed iteratively towards the destination. The map is a processed version of an online floorplan made freely available by many businesses. Though the floorplan generally includes no reference to absolute scale, users need only enough information to become *unlost*. With this concept in mind, we evaluate the performance of localization and wayfinding under various conditions of user reliability. We find that localization precision improves rapidly as the environment becomes increasingly populated with landmarks, though few landmarks (hereafter referred to as Features) are needed from a user's input. Wayfinding naturally follows via an intuitive routing strategy. From this, we find that successful paths are consistently short, and the primary factor affecting wayfinding performance is path connectedness, rather than path length.

II. Visibility Localization

A. Environmental Model

Vision as a sensor has previously been proposed for robot localization [11]. Here, we depend on human vision for information about the environment. Since visibility requires line-of-sight, complications due to multipath can be ignored. Human vision also offers the particular advantage of being an effective noise filter. Consider that humans have little trouble identifying a static landmark, even as other people walk past it.

We propose that floorplans contain enough information, stored in the form of *intervisibility relationships*, to successfully localize on a relative scale meaningful to human users. Here, success cannot be defined as achieving absolute precision in arbitrary units. Rather, a successful localization is one which allows the user to comprehend his position relative to the environment. Precision is thus measured as a unitless ratio of the located occupied area to the total occupiable area within an indoor environment (see Definition 6).

The environmental and computational model is based on four constructs: *Isovist, Feature, Feature Vector,* and *Region*.

Definition 1 (Isovist): $V_{(i,j)}$, the visible subset of an environment as viewed from point (i, j)

Definition 2 (Feature): An identifiable object, f_v, that can obstruct or be obstructed by an element in the environment

Definition 3 (Feature Vector): $F_{(i,j)}$, an h-entry array representing the visible subset of Features as viewed from point (i, j), where h is the total number of defined Features in the environment

$$F_{(i,j)} \equiv \begin{bmatrix} f_0 & f_1 .. & .. f_{h-1} \end{bmatrix}, \; f_v = \begin{cases} 1 & \text{if } visible \\ 0 & \text{if } invisible \end{cases}$$

Definition 4 (Region): The subset of occupiable coordinates implied by a Feature Vector

The 2D floorplan is first discretized and overbounded by a $\sqrt{n}$ x $\sqrt{n}$ grid. Any coordinates falling outside the boundary of the indoor environment are invisible from all interior vantage points, and safely ignored. Then, h Features, f_v, are identified, where $v = 0, 1 \ldots (h-1)$. Variable h is used to maintain consistency with visibility literature, i.e., visibility $holes$ [9]. Features include human-identifiable characteristics, derived from labels on the floorplan. Any "feature" that cannot obstruct or be obstructed, such as an image on the ceiling viewable from all vantage points, provides no information and is ineligible to be a Feature.

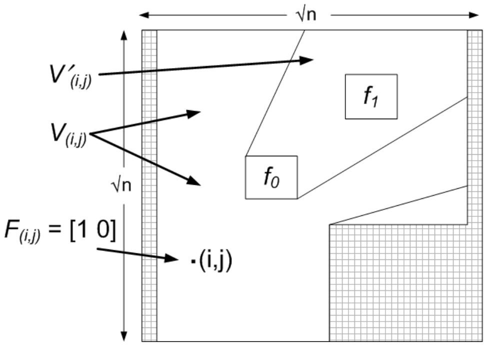


Figure 1. Floorplan with overbounding box, isovist $V_{(i,j)}$, complement $V^{'}_{(i,j)}$, $h = 2$ features, and feature vector $F_{(i,j)}$

Isovist, a term introduced in [6], represents the viewable area from a vantage point. Here we are interested in how isovists overlap. Each occupiable coordinate is encoded as a Feature Vector storing the subset of Features contained within its isovist. A user's report is itself a Feature Vector, and therefore implies a subset of coordinates that could generate such a vector. This subset of coordinates is the common Region in which the user must be located. Definition 4 is intentionally vague, as the specific interpretation of $Region$ depends on the method of deduction implemented.

Equivalently, calculating isovists from the perspective of each Feature results in a set of visible coordinates associated with each Feature. Given a user-reported Feature Vector, F_{rep}, the located Region is calculable by set-algebra. This $duality$ of intervisibility corresponds to intuition; if you can see a Feature, the Feature can see you. Likewise, if you cannot see a Feature, the Feature cannot see you.

Axiom 1 (Duality): If $f_v \in V_{(i,j)}$, then $(i, j) \in V_{f_v}$

We now present three methods of locating a user.

B. Three Methods

1) Perfect Reporting: Assume perfect reporting. For reports of $f_v = 1$, the Feature is definitely visible. For reports of $f_v = 0$, the Feature is definitely invisible. Consequently, a $Region$ is defined for this case as *a subset of coordinates with identical Feature Vectors*. At this time, it is also convenient to introduce the *Isovist-complement*.

Definition 5 (Isovist-complement): $V^{'}_{(i,j)}$, the invisible subset of an environment as viewed from point (i, j)

For all user-reported $f_p = 1$, $f_q = 0$, where $p, q = 0, 1 \ldots (h-1)$, the located Region R_{pq} is:

$$R_{pq} = \Big(\bigcap_{\forall p} V_{f_p} \Big) \cap \Big(\bigcap_{\forall q} V^{'}_{f_q} \Big) \tag{1}$$

Figure 2 illustrates the technique. A room with $h = 4$ Features is partitioned by isovists V_{f_0}, V_{f_1}, V_{f_2}, and V_{f_3}. The resulting disjoint Regions are labeled with their corresponding Feature Vectors. A user reports positive sightings of f_0, f_1, and the located Region is highlighted.

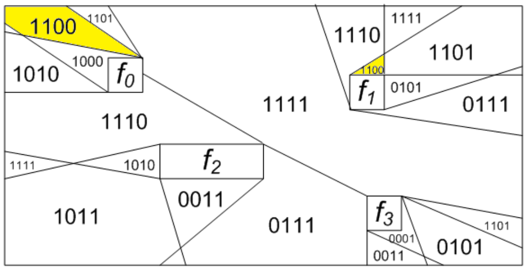


Figure 2. Floorplan with isovists V_{f_0}, V_{f_1}, V_{f_2}, and V_{f_3}. User reports $F_{rep} = [1\ 1\ 0\ 0]$

With only $h = 4$, the located Region is already very small compared to the total occupiable area. Note that a given

Region may be *discontiguous*. Such ambiguities diminish in both size and occurrence as h grows. Remaining ambiguities are resolved by the wayfinding procedure.

Each Feature Vector is a binary array of length h, implying that the maximum number of Regions possible in a given environment is 2^h. With the number of Regions growing exponentially in h, the average area per Region shrinks exponentially. This motivates how we measure the potential performance of the system, given a particular environment.

Definition 6 (Average Precision): $\bar{P}$, the ratio of the average area among all Regions and the total occupiable area, $0 < \bar{P} \leq 1$

In an environment with the maximum 2^h Regions, $\bar{P}_{min}$ is clearly $\frac{1}{2^h}$. The environment of Figure 2, however, does not generate $2^4 = 16$ Regions. Indeed, there are only 11 Regions, and this is typical of environments where $h > 3$. Note also that any environment with a convex boundary can never generate more than $2^h - 1$ Regions, since there is no vantage point from which all Features are invisible, i.e., $F_{(i,j)} = [0 \ldots 0]$ is impossible.

Let $\mathbb{S}_R$ be the set of all Regions generated. Then $|\mathbb{S}_R| \leq 2^h$, and

$$\bar{P} = \frac{1}{|\mathbb{S}_R|} \geq \bar{P}_{min} \qquad (2)$$

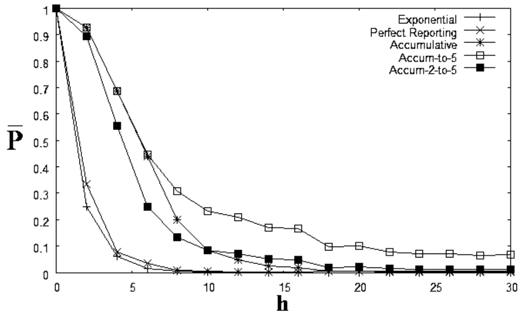
Although in general $\bar{P}$ cannot achieve the minimum, we lose very little precision in practice. Simulations using a neutral repeating pattern of Features ranging from $0 \leq h \leq 30$ show that $\frac{1}{|\mathbb{S}_R|}$ tracks closely behind $\frac{1}{2^h}$ in a quasi-exponential decay. The simulation implements an empty room for $h = 0$, two identical square Features for $h = 2$, four squares for $h = 4$, and so on. Figure 3(a):Perfect Reporting shows the resulting precision as h increases. $\bar{P}$ improves rapidly with h, stabilizing near minimum beyond $h \approx 10$. We conclude that localization performance improves as the environment becomes increasingly populated with Features.

2) Accumulative Reporting: In practice, there is a risk of reporting error. The user commits a *Type I Error* by reporting that an invisible feature is visible. The user commits a *Type II Error* by reporting that a visible feature is invisible. Limited field tests suggest Type I errors are rare, while Type II errors occur quite frequently. In a view densely populated with features, it is reasonable that a human user will overlook something.

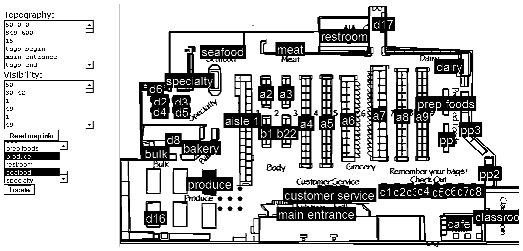
A reporting error processed by the deterministic Perfect Reporting technique ouputs not only a less accurate result, but one that is *uniquely wrong*. This is a consequence of the disjoint nature of Regions. To avoid this complication, we choose to sacrifice all information associated with a report that Feature f_v is invisible. In other words, $\Pr\big(\text{Type II} \mid f_q = 0\big) = \frac{1}{2}$. Positive sightings, however, are considered trustworthy. Let $\Pr\big(\text{Type I}\big) = 0$. It follows that a *Region* is defined for this case as *the subset of coordinates* (i, j) *such that* F_{rep} & $F_{(i,j)} = F_{rep}$, where & is bitwise AND. To illustrate,

$$F_{rep} = [1\ 1\ 0\ 0] \supseteq \left\{ \begin{array}{c} [1100], [1101] \\ [1110], [1111] \end{array} \right\}$$

F_{rep} becomes a superset implying all Feature Vectors $F_{(i,j)}$ sharing $f_p = 1$. Regions are no longer disjoint, and $\bar{P}$ should suffer. This is supported by Figure 3(a):Accumulative for small h, but surprisingly, $\bar{P}$ still decays rapidly. Beyond $h \approx 10$, $\bar{P}$ stabilizes near the precision achieved by Perfect Reporting. Even under constraints imposed by human error, localization performance recovers for sufficient h.



(a) Average Precision $\bar{P}$ vs. Features h


Topography:

```
50 0 0
849 600
15
tags begin
main entrance
tags end
```

Visibility:

```
50
30 42
1
49
1
49
```

(b) Application Interface

Figure 3. (a) Simulation precision for different methods, and (b) Real supermarket floorplan with $h = 43$, $\bar{P} = 0.005$, Accumulative Reporting

3) Accumulative with Limited Input: Realism demands acknowledgement of users' limited patience. For large h, some Feature Vectors require many positive sightings in order to be invoked, i.e., *feasible* Feature Vectors may be *implausible*. For this reason, we investigate the effect of eliminating from consideration any feasible user-reports containing > 5 positive sightings. These highly informative reports often imply the most precise Regions, so with their removal $\bar{P}$ should again suffer. Figure 3(a):Accum-to-5 shows that the resulting $\bar{P}$ underperforms the unlimited case as expected. For example, $\bar{P}$ is nearly 0.10 when $h = 20$. However, adding the reasonable constraint that a user must report ≥ 2 positive sightings brings $\bar{P}$ back in line with the unlimited case. For $h = 20$, Figure 3(a):Accum-2-to-5 shows $\bar{P} = 0.02$. This implies the system can run in the unlimited Accumulative mode since its plausible range of operation offers performance comparable to Accum-2-to-5.

C. Complexity

A discussion of complexity is warranted since the application must presumably run on a commercially available mobile phone. Calculating isovists reduces computationally to solving a *visibility graph*, of which there are many polynomial-time algorithms in $\mathrm{O}\big(n^2\big)$ or better [3]. Consistency with this notation motivates the $\sqrt{n}$ x $\sqrt{n}$ overbounding of Section 3.1. Isovists, however, are computed during offline processing prior to loading floorplans onto the mobile device, for example, as a cloud service offering processed maps for download. The details of processing floorplans are beyond the scope of this discussion. On the mobile phone itself, the worst case computation is a search through 2^h potential Feature Vectors. While 2^h grows exponentially, the number of possible Feature Vectors in a given environment is bounded by the number of occupiable coordinates, at most $\sqrt{n}$ x $\sqrt{n} = n$. Let n_h be the smallest number of coordinates occupied by h Features. For $h \geq k$, where $2^{k-1} + (k-1) < n \leq 2^k + k$,

$$\min\big[2^h,\, n - n_h\big] = n - n_h \Rightarrow \mathrm{O}\big(n\big) \tag{3}$$

III. Visibility Wayfinding

A. Environmental Model

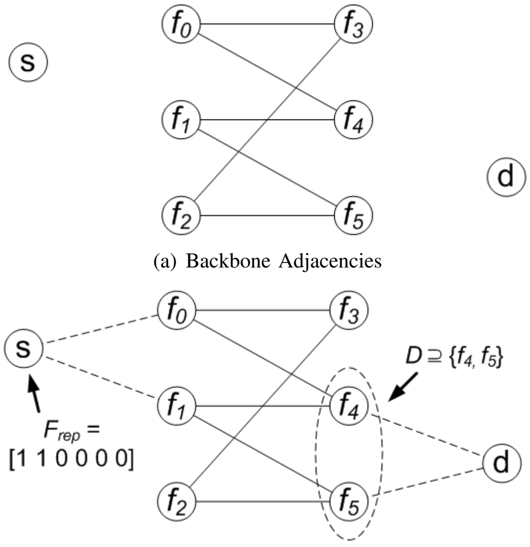
Graph-based modeling of architectural space is widely supported [8]. Existing treatments on human navigation have considered how to generate effective semantic route descriptions along collision-free paths [4]. This strategy can be likened to turn-by-turn driving directions where the user is instructed to “Turn right at Main Street.” Indoors, this might materialize as “Walk down the hallway, turn right at the bookcase.” In both cases, success relies on each element of the route description being comprehensible and identifiable on the fly. A missed turn can long go unnoticed.

We aim to streamline routing instructions by incorporating only those elements which have been positively sighted by the user. Any sighted element can serve as a potential *next-hop* toward the destination. A sequence of such sightings and next-hops comprises a path. The resulting navigation procedure is easy to follow.

Let $G(h, e)$ be a graph of h nodes and e edges, where h is the number of Features in the environment. An edge appears between a Feature and any other Feature within its isovist. Such an edge is undirected, by the following corollary to Axiom 1.

Corollary 1 (Duality): If $f_i \in V_{f_j}$, then $f_j \in V_{f_i}$

$G(h, e)$ serves as a *backbone* network, since the origination and destination points are not necessarily colocated with Features, i.e., they may not be *on-net*. Figure 4(a) illustrates the concept as a bipartite graph with off-net source and destination. Bipartiteness is not required, and chosen here only to reinforce the notion of a backbone with an access layer.



(a) Backbone Adjacencies

(b) Source and Destination

Figure 4. (a) Backbone adjacencies among intervisible room features, and (b) F_{rep} source initialization, with destination supernode $\mathbb{D}$

To initialize, a user at source location *s* must enter the backbone. Figure 4(b) shows this is conveniently achieved using F_{rep}, the input to the original localization process. Any visible Feature f_v contained within F_{rep} is a feasible access point. Note that the *output* of the localization process, i.e., the *Region*, is not needed. Consequently, the occasional localization ambiguity associated with discontiguous Regions is completely resolved during wayfinding.

Although the destination *d* is typically a backbone node, we preserve generality by allowing off-net destinations. In particular, envision a “meet-up” application where the destination is not a Feature, but another user who has entered his own F_{rep}. Leading the first user through the backbone to any *egress* node sighted in the second user’s F_{rep} puts the two users in mutual line-of-sight. If for any reason *d* still cannot be found, the wayfinding procedure can resume. This helps to formalize the routing goal.

Definition 7 (Prior-Hop Destination): $\mathbb{D}$, a supernode containing the set of Features within the isovist of destination *d*

Collapsing a subset of egress nodes into supernode $\mathbb{D}$ effectively reduces the number of vertices in graph $G(h, e)$. The number of edges may also be reduced, but this is not guaranteed since the elements of $\mathbb{D}$ are not required to have edges among themselves. A node linked to multiple elements in $\mathbb{D}$ now has multiple parallel links to $\mathbb{D}$. Also, a corresponding ingress supernode cannot exist in general, since each access point may have different next-hop options within the backbone. Letting $n = h - |\mathbb{D}| + 1$ results in the new graph $G(n, e)$, with $n \leq h$.

All links in $G(n, e)$ are assigned a cost of 1. The overriding expense at each hop is the requirement of user interaction with the application. Other factors like physical distance do not strictly increase or decrease the cost. For this reason, hop count is the metric selected to minimize the number of user iteractions. $G(n, e)$ can then be represented as a binary adjacency matrix $\mathbf{Adj}_G$. The adjacency matrix of Figure 4(b) follows,

$$\mathbf{Adj}_G = \begin{array}{c} \\ f_0 \\ f_1 \\ f_2 \\ f_3 \\ \mathbb{D} \end{array} \begin{array}{c} \begin{array}{ccccc} f_0 & f_1 & f_2 & f_3 & \mathbb{D} \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

The final component of the wayfinding model is link failure. At each hop through the network, adjacent next-hops may go unreported. The user cannot be directed toward an unreported next-hop, so this link is said to have *failed*. Only the link, and not the next-hop node itself, has failed. That node may later be sighted, reported, and subsequently visited from a different vantage point within the network. A Feature within the user's isovist goes unreported for any number of reasons, including: distance, size, lighting, another dominant or distracting element, errors in the underlying map, user fatigue, and even obstruction by the local Feature the user is departing from. Collecting all sources of error into a single independent probability of link failure, let $q = \Pr\big(\text{Type II Error}\big)$. Then, the probability a user successfully reports a Feature within his isovist is $p_v = 1-q$.

B. Routing and Simulation

1) Routing Strategy: At each hop, an underlying routing strategy determines which next-hop to recommend. Reminiscent of mobile ad-hoc wireless networks, visibility wayfinding must cope with high link failure rates and relatively local information. However, like traditional wired networks, a topology map is available in the form of $\mathbf{Adj}_G$.

We propose a simple greedy strategy, hereafter referred to as *Hop-by-Hop Shortest Path First(SPF)*. The occupied node j experiences selected user-induced failures of its *directly-connected* links. Any remaining edges of $\mathbf{Adj}_G$ are assumed intact, since we cannot predict what future link failures will occur. Dijkstra's shortest path algorithm is then invoked between node j and destination $\mathbb{D}$. After proceeding 1 hop, the newly occupied node is redefined as j.

Note that concatenating each Hop-by-Hop SPF decision may not reduce to the same route as an end-to-end shortest path calculation in which all network-wide failures have already occurred. The latter case amounts to having foreknowledge that is unavailable to Hop-by-Hop SPF, so its resulting path can be considered *Optimal*.

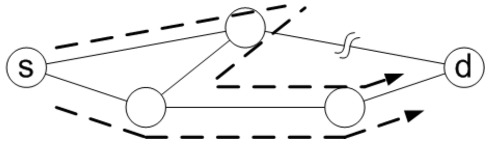


Figure 5. Hop-by-Hop SPF (top) vs. 3 hop Optimal path (bottom)

We evaluate the expected performance of Hop-by-Hop SPF by simulating room adjacency matrices of varying node density, edge density, and failure rate. Comparison against Optimal routing, both with and without link failures, provides two levels of lower-bound. *Random routing*, implemented with hop-by-hop iterations as in Hop-by-Hop SPF, provides a form of upper-bound performance, i.e., the worst result without actively avoiding the destination. The simulations reveal parameters for which Hop-by-Hop SPF closely tracks Optimal routing.

2) Simulation: In $G(n, e)$, e is replaced by p, the independent probability that an edge appears between any two nodes. p is equivalently the edge *density* of the graph. Erdős-Rényi [7] random graphs $G(n, p)$ are generated for all combinations of $n = \{10, 20, 40, 80\}$ and $p = \{.125, .25, 0.5\}$. Every instance of $G(n, p)$ is evaluated using Optimal routing, Hop-by-Hop SPF, and Random routing, under conditions of link availability according to $p_v = \{.25, 0.5, .75\}$. An additional case of Optimal routing immune to link failure, i.e., $p_v = 1$, serves as a lowest-bound benchmark. Each test case is repeated on 1000 random graphs and the results averaged.

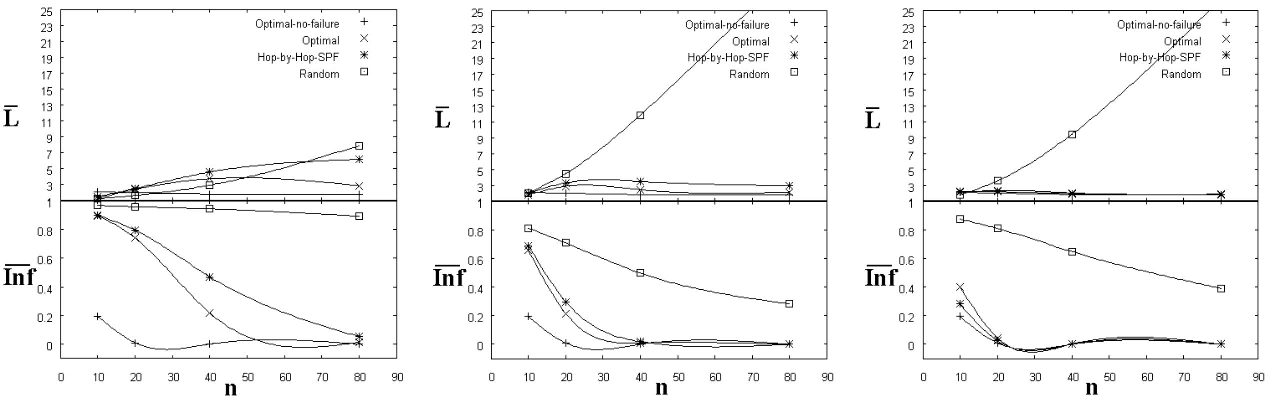
The performance measurement of each routing strategy on $G(n, p, p_v)$ is provided by,

Definition 8 (Characteristic Distance): $\bar{L}$, the average distance, in hops, among the feasible paths between all node pairs, resulting from a given routing strategy

Our interest is in both the distance of successful paths as well as *whether* paths are successful. As $\bar{L}$ is an average, it does not consider infinite paths. Link failures and randomness in the graph-generation process may result in critical node isolations, i.e., no feasible path, for a particular node pair. Therefore, every $\bar{L}$ is accompanied by $\overline{Inf}$, the average occurrence of infinite paths expressed as a ratio to the total $\binom{n}{2}$ potential paths. A compact approximation of $\bar{L}$ is found in [14],

$$\bar{L} \approx \frac{\ln n}{\ln k} \tag{4}$$

where k is the average degree per node in a graph.

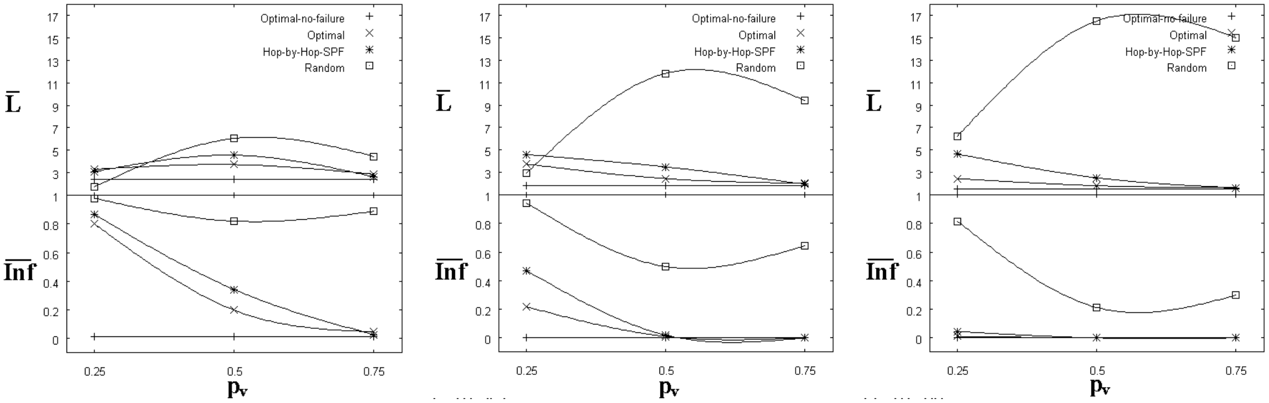


(a) $p = .25$, $p_v = .25$

(b) $p = .25$, $p_v = 0.5$

(c) $p = .25$, $p_v = .75$

Figure 6. Characteristic Distance $\bar{L}$ vs. n nodes for $p = .25$ and various p_v



(a) $n = 40$, $p = .125$

(b) $n = 40$, $p = .25$

(c) $n = 40$, $p = 0.5$

Figure 7. Characteristic Distance $\bar{L}$ vs. p_v for $n = 40$ and various p

Eq.(4) can be adapted as a convenient estimate of $\bar{L}$ for Optimal routing in $G(n, p, p_v)$. Given $k = \frac{2e}{n}$, let e be the number of edges remaining in $G(n, p, p_v)$ after all network-wide link failures have occurred. Then, $e = p\binom{n}{2}p_v$, and substituting, $k = pp_v(n-1)$. Restating Eq.(4),

$$\begin{aligned} \bar{L} &\approx \frac{\ln n}{\ln k} && (5) \\ &\approx \frac{\ln n}{\ln\left(pp_v\left(n-1\right)\right)} && (6) \\ &\approx \frac{1}{1 + \log_n(pp_v)} && (7) \end{aligned}$$

As Eq.(7) estimates $\bar{L}$ for Optimal routing, it also estimates the lower-bound of $\bar{L}$ for Hop-by-Hop SPF.

Figure 6 plots $\bar{L}$ and $\overline{Inf}$ as n increases. For space and clarity, density p is held constant at .25, the center of the test range. Maintaining constant density as n varies also reflects reality. Adding a new Feature to a room immediately offers up to n new edges. Thus, the growth in edges has a proportional relationship with n. Random routing sets a quasi-linear upper-bound on $\bar{L}$ for the test range, implying that Hop-by-Hop SPF should perform sublinearly. Figures 6(b) and 6(c) show that for $p_v \geq 0.5$, not only does Hop-by-Hop SPF offer sublinear growth, $\bar{L}$ tends to *decrease*. In fact, for $n \geq 20$, Hop-by-Hop SPF closely tracks Optimal routing in both $\bar{L}$ and $\overline{Inf}$.

Consider $G(n = 20, p = .25, p_v = 0.5)$. Eq.(7) predicts a lower-bound of $\bar{L} = 3.27$ hops. The simulation results in $\bar{L} = 2.87$ for Optimal routing, and $\bar{L} = 3.31$ for Hop-by-Hop SPF, representing a stretch of only 15%. Increasing n to 40 brings $\overline{Inf}$ below 2% for both cases. This suggests that Eq.(7) reasonably approximates the lower-bound, Hop-by-Hop SPF is an effective routing strategy, and the performance of visibility wayfinding improves as the environment becomes more populated with Features.

Figure 7 plots $\bar{L}$ and $\overline{Inf}$ vs. p_v. Here density p is varied between poorly connected and well connected networks. For space and clarity, n is held constant at 40, the center of the test range. This value of n has particular relevance due to its agreement with real floorplans we have studied. Recall that h represents the number of Features in an indoor environment, and $n = h$ when supernodes are ignored. The supermarket of Figure 3(b) has $h = 43$, and the convention center of Figure 9 has $h = 38$.

For $n = 40$, Hop-by-Hop SPF closely tracks Optimal routing in $\bar{L}$, regardless of p or p_v. In $\overline{Inf}$, however, only for $p_v \geq 0.5$ does Hop-by-Hop SPF become relatively low and near-Optimal. This implies that link availability p_v's effect is strongest on *whether* the user reaches the destination, rather than how long the path is. Revisiting Figure 6, the same behavior is apparent. Further, the assumption of independence for p and p_v implies that performance should be similar for graphs with the same product pp_v, even if p and p_v are interchanged.

Inspecting the full dataset of Hop-by-Hop SPF for all $G(n, p, p_v)$ (not illustrated) reveals consistently low $\bar{L}$, but distinct regions of very high and very low $\overline{Inf}$ (see Fig. 8). The $\overline{Inf}$ data can be separated into 2 subsets: poor performers with minimum value 24% and good performers with maximum value 5%. This represents a sharp transition, with low values appearing when $pp_v > \frac{\ln n}{n}$. This corresponds to a well-known threshold beyond which random graphs are connected *almost surely* [7]. We conclude that $\overline{Inf}$, moreso than $\bar{L}$, is the limiting factor as to whether visibility wayfinding is effective. Therefore, for indoor environments where $pp_v > \frac{\ln n}{n}$, we predict good performance in both $\bar{L}$ and $\overline{Inf}$.

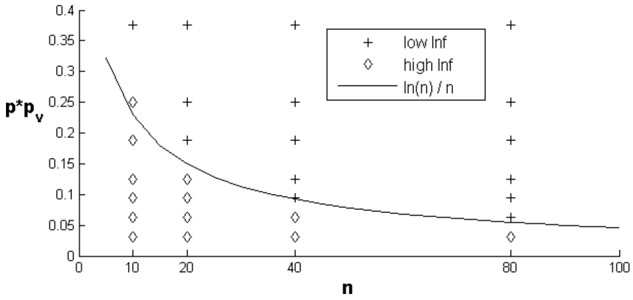


Figure 8. Sharp transition between low $\overline{Inf}$ good performers and high $\overline{Inf}$ poor performers

C. Complexity

For the graph $G(n, e)$, with n nodes and e edges, Dijkstra's shortest path algorithm runs in $\mathrm{O}\big(e \log n\big)$. Hop-by-Hop SPF requires, at worst, $n-1$ iterations of Dijkstra if the sequence of failures forces a user through all nodes. Therefore, Hop-by-Hop SPF results in $\mathrm{O}\big(ne \log n\big)$. However, each iteration need not operate on the entire $G(n, e)$ if routing loops are prohibited. At each hop, the previous node and at least 1 edge are removed from the graph. Let $c \leq 1$ be a constant fraction. After cn iterations, at worst $(1-c)n$ nodes and $e - cn$ edges remain. The remainder of the computation is then $\mathrm{O}\big((1-c)n(e-cn)\log((1-c)n)\big) \Rightarrow \mathrm{O}\big(ne \log n\big)$. Therefore, the full computation over all iterations is still $\Omega\left(ne \log n\right)$. However, most of the time the worst case does not occur. Typically, each iteration removes more than 1 edge, so insight is gained by considering the typical number of edges per node. Let $k = \frac{2e}{n}$ be the average degree per node in $G(n, e)$. The average computation over all iterations is,

$$\mathrm{O}\big(\sum_{i=0}^{n} e_i \log(n-i)\big) \tag{8}$$

where $e_0 = e$, and $e_i = e_{i-1} - \frac{2e_{i-1}}{n+1-i}$. Intuition also tells us that typically, all n nodes are not visited. Let $m \leq n$ be the true number of nodes visited, occuring with probability $\frac{1}{n}$. Eq.(8) becomes,

$$\mathrm{O}\big(\frac{1}{n}\sum_{m=1}^{n}\sum_{i=0}^{m} e_i \log(n-i)\big) \tag{9}$$

IV. Field Test

We have conducted a limited on-site user study at a large convention center using a real online floorplan. The application prototype was implemented on a commercial mobile phone, the HTC Android Dev 1. Due to space limitations, full details of the prototype and testing are omitted in favor of our main observations.

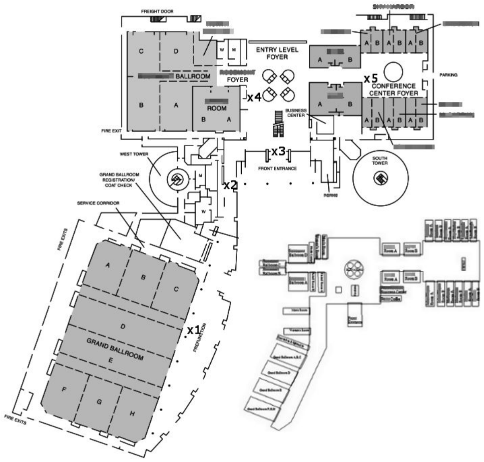


Figure 9. Real floorplan of test site, computational map (inset), and test positions x_i, identifying details blurred

Figure 9 shows the original floorplan, as well as the streamlined version used for computation. Maps are anonymized by blurring any obvious identifiers.

After processing the convention center, we find $h = 38$ and $p = .283$. The most critical uncontrollable factor is p_v, the probability a Feature is reported given that it is within the user's computed isovist. Figure 10 shows the expected behavior of the convention center for various p_v, given h(or n) $= 38$ and $p = .283$. Observe the strong correspondence between the real matrix of Figure 10 and the random matrices of Figure 7(b). In Figure 7(b), $n = 40$ and $p = .25$, parameters approximating those of the real test site. The agreement between these figures suggests the simulation environment is applicable. $\frac{\ln n}{n}$ predicts pp_v should exceed .096 to achieve near-Optimal path length and path failure rate. For $p = .283$, this requires $p_v > .339$. Figure 10 suggests p_v should be higher, around 0.5. The user testing, therefore, focuses on estimating p_v.

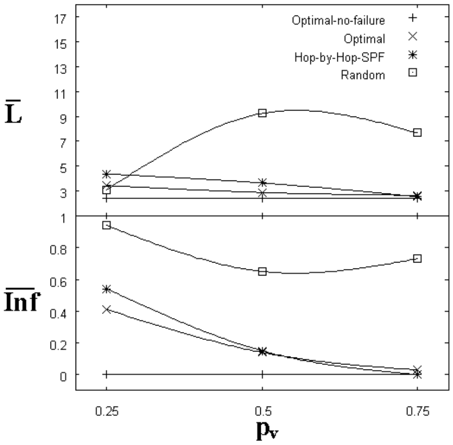


Figure 10. Expected performance on real convention center floorplan with observed $n = 38$, $p = .283$. Compare with Figure 7(b)

Test subjects consisted of 10 volunteers with no prior knowledge of the test site. They were aware the study involved an “indoor GPS” system using a mobile phone.

Using a uniform randomly generated binary matrix superimposed over the test site floorplan, 5 random coordinates within the occupiable space were chosen as vantage points. These vantage points are labeled x_i in Figure 9. Isovists V_{x_i} and Feature Vectors F_{x_i} were computed, generating lists of the Features that *should* be visible from each x_i. The various F_{x_i} range in length from 4 to 18 Features. Every volunteer was tested from every vantage point, asked simply to “Mark each Feature you see,” producing 50 readings.

The vision test resulted in $p_v = .496$. Applying the t-distribution, as is customary for small-sample studies, we find at the 90% confidence level that $p_v > .46$. This is well above .339, the predicted quality threshold. It is also close to 0.5, the value suggested by Figure 10.

V. Conclusions

We have combined theory with practicality to deliver an indoor localization and wayfinding system with a realistic chance at widespread adoption. It successfully avoids on-site infrastructure and detailed site-surveys. Precise building schematics are not required; we use only rough online floorplans that preserve visibility relationships, often with no reference to scale. Implementation requires no formal cooperation from building owners, and the software runs on a commercially available mobile phone.

Respect for realism guided all aspects of the system’s design and evaluation. Using effective computational models, we investigated the factors influencing location precision, path length, and success rate. Relationships among the number of environmental elements, their adjacencies, and human reporting ability, predict which floorplans have the potential to offer a good user experience. These insights improve our understanding of indoor environments, and provide a framework by which future results can be incorporated into useful applications.

References

[1] M. Azizyan, I. Constandache, and R. Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In *MobiCom*, pages 261–272. ACM, 2009.

[2] P. Bahl and V. Padmanabhan. Radar: an in-building rf-based user location and tracking system. In *INFOCOM*, volume 2, pages 775–784, 2000.

[3] B. Ben-Moshe, O. Hall-Holt, M. Katz, and J. Mitchell. Computing the visibility graph of points within a polygon. In *SCG ’04*, pages 27–35, USA, 2004. ACM.

[4] D. Caduff and S. Timpf. The landmark spider: Representing landmark knowledge for wayfinding tasks. In *Reasoning with Mental and External Diagrams*, 2005.

[5] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, and M. Wiseman. Indoor location sensing using geomagnetism. In *MobiSys*, pages 141–154. ACM, 2011.

[6] L. Davis and M. Benedikt. Computational models of space: Isovists and isovist fields. Technical report, USA, 1979.

[7] P. Erdös and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5:17–61, 1960.

[8] G. Franz, H. Mallot, and J. Wiener. Graph-based models of space in architecture and cognitive science - a comparative analysis. In *17th InterSymp*, pages 30–38, 2005.

[9] F. Hoffmann, M. Kaufmann, and K. Kriegel. The art gallery theorem for polygons with holes. In *SFCS*, 1991.

[10] D. Kaspar. Application of directional antennas in rf-based indoor localization systems. Master’s thesis, Swiss Federal Institute of Technology Zurich (ETH), 2005.

[11] M. Köhler, S. Patel, J. Summet, E. Stuntebeck, and G. Abowd. Tracksense: Infrastructure free precise indoor positioning using projected patterns. In *Pervasive Computing*, pages 334–350, 2007.

[12] C. Nerguizian, L. Hamza, V. Nerguizian, and M. Saad. 3d indoor geolocation with received signal strength fingerprinting technique and neural networks. In *ESPOCO*, pages 1–6, USA, 2005. WSEAS.

[13] W. ur Rehman, E. de Lara, and S. Saroiu. Cilos: a CDMA indoor localization system. In *UbiComp*, pages 104–113. ACM, 2008.

[14] D. Watts and S. Strogatz. Collective dynamics of small world networks. *Nature*, 393:440–442, 1998.