Marker-Based Localizing for Indoor Navigation

Sudarshan S. Chawathe

Department of Computer Science
University of Maine
Orono, ME 04469-5752, USA
chaw@cs.umaine.edu

***Abstract*— We present a method for assigning markers to locations to enable navigation by observing short sequences of markers. A motivating application is indoor navigation, where pedestrians can determine their location using the sequence of recently encountered markers as they walk along hallways in a building. While we may solve this problem simply by assigning a unique marker to each location, such a solution limits the granularity of localization due to limits on the number of distinct markers. We present a more efficient solution that uses a sequence of recently seen markers to determine locations. We demonstrate that our solution yields significant improvements over the naive solution even when the marker-sequences are quite short, consisting of only three of four markers each.**

I. Introduction

The method of localizing a person using markers placed in the environment has been effectively used in both indoor and outdoor settings [1]. This method is particularly promising when other methods, such as those based on GPS, are unsuitable. Although our work is applicable in several settings, we will focus on indoor navigation for concreteness, since it is a prime application for marker-based localizing due to unavailability of reliable GPS signals within large buildings. In particular, we consider settings in which a person carries a small mobile device that is capable of reading markers, such as a PDA (personal digital assistant) equipped with a camera. This mobile device uses the markers to localize the person and provide additional services, such as navigation. The specific technology used for the markers, such as optical or radio, is not important for the purpose of this paper, as our method does not make any restrictive assumptions on the marker design. However, for concreteness, we may wish to consider an optical system in which markers are based on colored geometric patterns and in which markers are detected using a camera.

In this paper, our focus is on the placement of markers in the environment in a manner that permits efficient localizing. A simple solution is to place a unique marker at each location at which localizing is desired. We shall henceforth refer to these locations as *important locations*. For example, in a large office building, the part of a hallway adjacent to each room door may be designated an important location, so that it is possible to localize a person to the nearest room door. The key drawback of this simple solution is that the number of required markers (distinct) is equal to the number of important locations. As the region within which localizing is desired grows (larger buildings, groups of buildings, entire campuses), the large number of required markers may limit the applicability of this method. For example, consider the optical marker system suggested earlier. For a given camera resolution, operating distance, marker size, and detection reliability, there is a limit on the number of distinct markers available. It is therefore useful to consider methods that make more efficient use of the available set of distinct markers, i.e., those that can localize a person to a larger set of important locations using the same number of distinct markers.

Consider a person walking inside a building using a marker-reading device such as the PDA suggested above. Instead of using only the current (most recently read) marker for localizing, our method uses a sequence of k most-recently read markers, where k is a parameter with a typical value of 3 or 4. The main idea is that we may assign identical markers to multiple important locations if we can ensure that it is not possible to arrive at both the locations after reading the same k most-recent markers. If we devise such a scheme then the required number of distinct markers is much smaller than the number of important locations. (We quantify these statements in Section IV.) Devising a scheme for placing markers that satisfies the above localizing property is the main problem addressed by this paper.

In Section II, we develop the ideas introduced informally above to yield a formal problem definition. Our solution to this problem, described in Section III, is based on a greedy algorithm that efficiently checks for potential duplicates using dynamic programming. We present some experimental results in Section IV, where we evaluate the properties of the marker-placement scheme produced by our algorithm, as well as the running time of our implementation. We discuss related work in Section V and conclude in Section VI.

II. Localizing Colorings

We model the network of indoor locations, both important locations and intermediate locations, using a graph in which edges represent locations (e.g., sections of hallways) and vertices represent intersections. (Unless indicated otherwise, we follow standard graph terminology [2].) A marker at a

This work was supported in part by the U.S. National Science Foundation with grant CNS-0426683.

location (edge) is associated with a *color* assigned to the edge. Following the scheme outlined in Section I, the color of an edge does not uniquely identify an edge (by itself). A person walking in a network of locations corresponds to a *walk* in this graph, where a walk a sequence of edges such that each successive pair of edges is incident on a common vertex. We note that we do not restrict our attention to simple walks, as is often done. The reason is that it is quite likely that a person may repeatedly traverse some edges, especially if he or she is lost, and the system should be able to localize the person in such cases. Our requirement that a person's location is uniquely determined by the k most-recently encountered markers (along any walk) is formalized as a k-step localizing coloring:

> A *k-step localizing s-coloring* of a graph $G = (V, E)$ is a surjective function $c : E \rightarrow \{1, 2, \ldots, s\}$ such that if $p_1 = e_1, e_2, \ldots, e_k$ and $p_2 = f_1, f_2, \ldots, f_k$ are walks in G with $c(e_i) = c(f_i)$ for all i in $1..k$ then $e_k = f_k$. We refer to s as the *size* of the coloring c.

Our goal of making an efficient use of the available distinct markers may then be incorporated yielding the following problem definition:

> *Minimum k-Step Localizing Coloring (k-MLC)*: Given a graph G and a positive integer k, find an integer s^* and a k-step localizing s^*-coloring c such that there is no k-step localizing s-coloring for $s < s^*$.

III. Marker Placement Algorithm

Our method for placing markers for indoor navigation in a manner that ensures k-step localization is based on a greedy algorithm that traverses a graph in depth-first order and assigns to each edge the lowest-numbered color that does not create duplicate k-sequences of colors. We describe this algorithm in more detail below. Since distinct markers may be assigned integer identifiers, we henceforth assume that the markers (colors) are integers.

We traverse the input graph in depth-first order, coloring each encountered edge as follows: Until a suitable color for the edge is found, we try all colors, in order. During each trial, we test whether coloring the edge with the trial color results in any duplicate k-sequences of colors, i.e., whether the resulting graph contains two distinct k-walks that generate identical sequences of colors. When no such duplicates are found, the color for the edge is finalized.

In the above algorithm, the test for duplicate color-sequences is the most expensive operation. All other operations are implemented efficiently using standard data structures. In order to check for duplicates, we need an efficient method for enumerating all k-walks that contain a given edge. Our solution to this sub-problem is based on precomputing (before the above depth-first traversal of the graph) an index that maps each edge e to the set of k-walks that contain e.

The bulk of the work during this precomputation is done using a dynamic-programming algorithm that computes $S(l, e, p)$ for $l = 0, 1, \ldots, k-1$, in sequence. Here $S(l, e, p)$ is the set of l-walks (walks of length l) that begin at endpoint p of edge e, and whose first edge is not e. (The edge e may appear as a subsequent edge in such walks.)

The algorithm is based on the following recurrence: For $l > 0$, $S(l, e, a)$ is the union, over all edges $f = (a, x)$, apart from e, incident on a, of the sets of l-walks obtained by prefixing f to each of the $(l-1)$-walk in $S(l-1, f, x)$. That is,

$$S(l, e, p) = \bigcup_{f=(a,x)\neq e} \; \bigcup_{w \in S(l-1,f,x)} (f) \cdot w$$

where $\cdot$ is the concatenation operator on walks: That is, $w_1 \cdot w_2$ is a walk that traverses, in order, all the edges in w_1 followed by all the edges in w_2. For the base case, we have $S(0, e, p) = \{()\}$ for all e and p. Here $\{()\}$ denotes the singleton set containing only the 0-walk $()$, i.e., the walk with no edges. This recurrence is easily converted to a dynamic programming algorithm using standard methods, so we skip the details.

Once S is computed as above, the set of k-walks containing an edge e is computed as the union of the following cross-products, for $l_a = 0, \ldots, k-1$:

$$S(l_1, e, a) \times \{(e)\} \times S(k - 1 - l_a, e, b)$$

where a and b denote the two endpoints of e and $\{(e)\}$ denotes a set containing the 1-walk (e) composed of only edge e. The cross product operator used above is similar to the standard set-theoretic operator except that the results are flattened. It combines the walks in its two operands by concatenation. That is, given sets W_1 and W_2 of walks,

$$W_1 \times W_2 = \bigcup_{w_1 \in W_1} \; \bigcup_{w_2 \in W_2} w_1 \cdot w_2$$

That the resulting coloring of the graph is k-step localizing follows directly from the test used when colors are assigned. Other properties of the resulting coloring, as well as the running time of the algorithm, are described next.

IV. Experimental Evaluation

In this section, we present an experimental evaluation of our methods, which we have implemented in Java. Experiments were conducted using Sun Java version 1.5.10, with Debian GNU/Linux 4.0 (Etch) as the operating system, running on a very modest machine (AMD Mobile Duron 800 MHz processor, 1 GB of RAM).

A. Sequences

We begin by quantifying the benefit of using a sequence of markers, instead of single markers, for localization in the simple case of a linear graph. This case corresponds to localization within a long interior hallway, for instance. We applied our method to linear graphs (sequences) of various lengths and computed the minimum number of distinct markers (colors) required by a scheme that guarantees k-step localization, for $k = 2, \ldots, 6$.

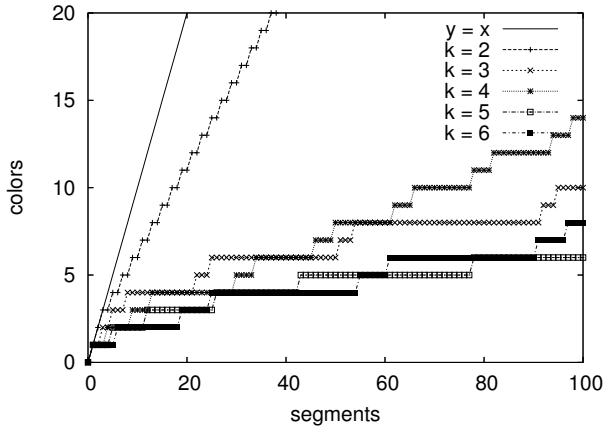

Fig. 1. Number of colors used for a k-step observable sequence as a function of the number of segments in the sequence.

Figure 1 summarizes the results of these experiments, with sequence length on the horizontal axis and the number of distinct markers on the vertical axis, for varying values of k. The figure also includes the line $y = x$ for comparison with the case when $k = 1$ and the number of distinct markers is the same as the number of locations. The figure illustrates the sharp reduction in the required number of distinct markers for k values greater than 2. It also suggests a pattern in which increasing the number of distinct markers by one increases the length of the sequences that can be localized by a large amount for $k > 2$.

B. Graph Generation

In order to evaluate our method on graphs, we need a method that produces random graphs of varying sizes. There are several standard models for random graphs. For instance, the $G(n, p)$ model, for integer $n > 1$ and real $p \in [0, 1]$ produces graphs with n vertices by creating an edge between each of the $n(n-1)/2$ pairs of vertices with probability p. While this model, and a few others, have several interesting properties and have been well studied, the resulting graphs are not suitable for our purposes because they do not resemble the networks of hallways in the interior navigation scenarios we consider. While the precise differences are difficult to quantify, they are easily visualized when the graph is depicted pictorially. One common problem is that the graphs generated by this model have too many branches and are often disconnected.

We remedy the above problem by using a two-step procedure for generating graphs for our experiments. In the first step, we generate a graph using the $G(n, p)$ model described above. In the second step, we modify the resulting graph by replacing each edge with a path of length s, where s is an integer chosen uniformly from the range $[1, S]$, S being a parameter. In what follows, order to avoid confusion between the edges of the graph produced by the first step and those of the graph produced by the second, we reserve the term edges to refer to the former and use the term *segments* to refer to the latter.

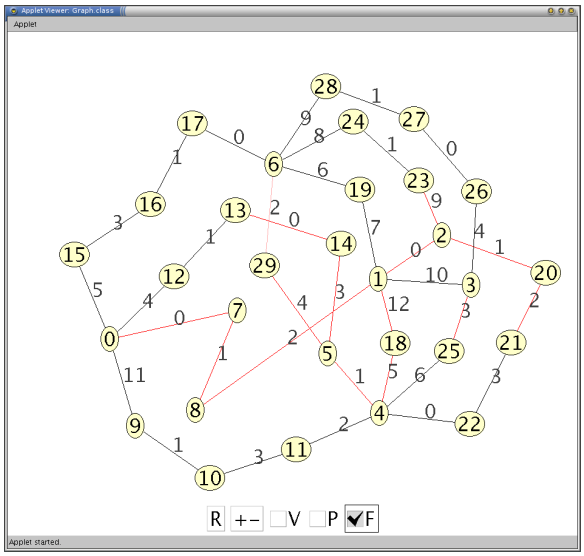

Fig. 2. Screenshot of the visualization module of our implementation

For our experiments, we used parameters that resulted in graphs that are (informally) similar to typical indoor localization environments. We tested our choice of parameters by visually inspecting the resulting graphs. Figure 2 depicts a screen-shot of this visualization part of our implementation. Although we have selected a small graph for presentation purposes, our implementation is effective for much larger ones. The visualization of larger graphs is aided by features that allow both automated rearrangement of the graph (using a ball-and-spring model) and manual repositioning of vertices on screen (using drag and drop).

C. Marker Utilization

Our main set of experiments studies the growth of the required number of distinct markers (colors) as the number of segments increases, for different values of k. In the naive solution ($k = 1$), each segment is assigned a distinct marker (color) whereas when k is larger, several segments typically share a color. Thus, the number of colors per segment provides a good metric of marker utilization, with smaller values indicating a better utilization of available markers. Figures 3–6 summarize the results of our experiments for different values of k. In each figure, the horizontal axis indicates the number of segments in the graph and the vertical axis indicates the required number of colors per segment.

Each of the Figures 3–6 also plots the function $y = x^{1/k}/x$, which is a loose lower bound on the minimum value of the number of colors per segment. That this function gives a lower bound follows by noting that there are only y^k distinct k-sequences that use y distinct markers, so that $y = x^{1/k}$ is a lower bound on the number of required markers. It is easy to observe that this lower bound is not achievable by considering a few simple examples. We plot it mainly to serve as a reference.

Perhaps the most interesting observation for the results in Figures 3–6 is that the values of the required number of

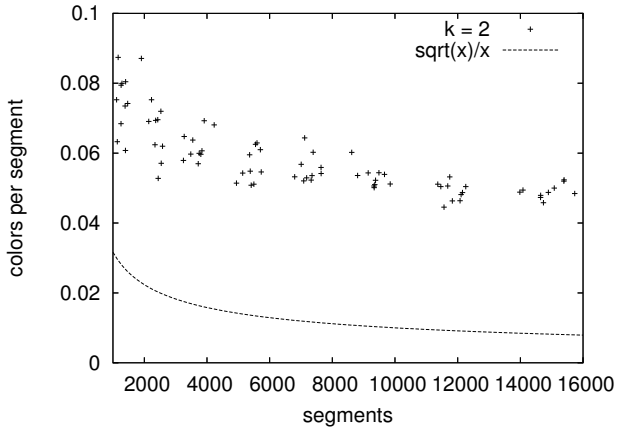

Fig. 3. Number of colors used per segment for varying graph sizes, measured as the number of segments in the graph.

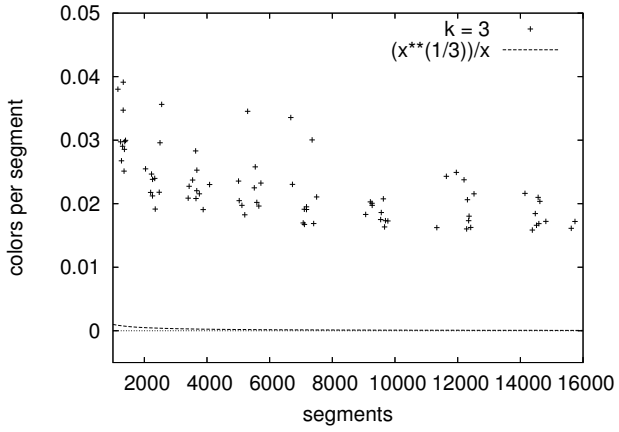

Fig. 4. Number of colors used per segment for varying graph sizes, measured as the number of segments in the graph.

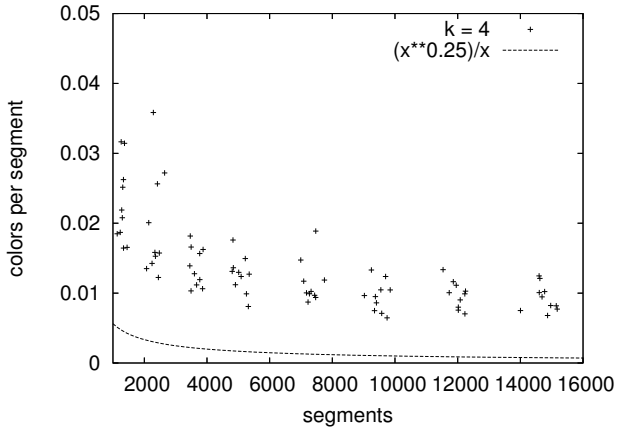

Fig. 5. Number of colors used per segment for varying graph sizes, measured as the number of segments in the graph.

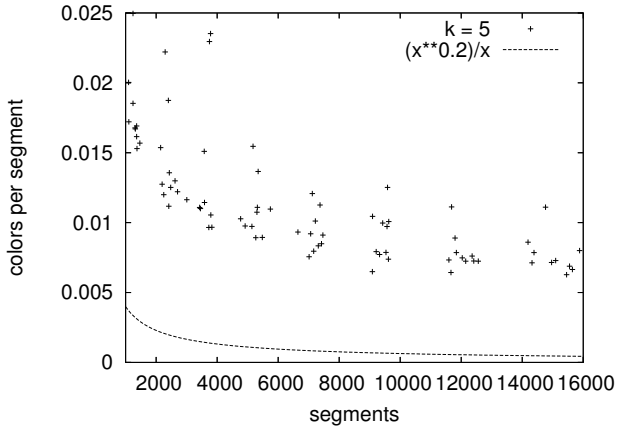

Fig. 6. Number of colors used per segment for varying graph sizes, measured as the number of segments in the graph.

colors per segment are all very low in an absolute sense. For example, Figure 4 suggests that a typical case for $k = 3$ is 0.025 colors per segment, which corresponds to each distinct marker being used by 40 segments on average. This number is a significant improvement over the naive case of $k = 1$ with one distinct marker per segment.

Figures 3–6 also suggest a slight downward trend in the required number of colors per segment as the number of segments increases (i.e., as the input graphs get larger). The variance in values of the required number of colors per segment also decreases as we move to larger graphs.

D. Sensitivity Analysis

Our next set of experiments studies the sensitivity of the above results to some key problem parameters. We first study the effect of the parameter p of the random-graph model $G(n, p)$ that is used in the first step of our graph-generating method (Section IV-B). Figure 7 summarizes the results of this study, with the horizontal axis indicating the values of p used in the graph-generating process. As above, the vertical axis indicates the required number of colors per segment.

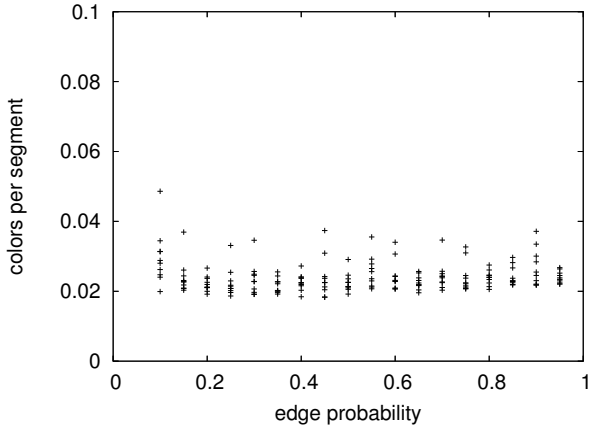

Fig. 7. Number of colors used per segment for varying edge probabilities.

Our main observation here is that although there is some

variation, as is to be expected for inputs based on random graphs, there is no strong dependence of the result on the parameter p. Intuitively, this result indicates that our earlier results are not greatly affected by whether the indoor environment is sparsely or densely connected.

Next, we study the effect of the parameter k in the definition of the kMLC problem. That is, we study the effect of increasing the number of sequential marker observations that are required for guaranteed localization. Figure 8 summarizes the results, with k on the horizontal axis and the number of colors per segment on the vertical.

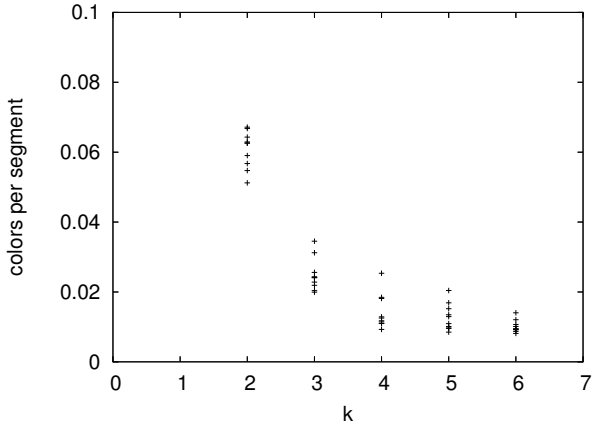

Fig. 8. Number of colors used per segment for varying number of steps required to identify a segment (the k parameter of the k-step observable graph problem).

As before, we note that the absolute values on the vertical axis are low when compared to the value of 1 for the naive $k = 1$ case, which is not shown. The figure also confirms the downward trend in the required number of colors that is suggested by Figures 3–6. Although larger values of k result in a better utilization of distinct markers, they require a larger number of observations for localization. In this regard, it is useful to note that a very large fraction the benefit of larger k values (say, $k = 6$) over the naive case ($k = 1$) is obtained by using very small k values (say, $k = 3$). Thus, it does not seem necessary to complicate the run-time localization process with k values larger than 3 or 4.

Our last set of sensitivity experiments study the effect of the segmentation parameter S used in the second step of our graph-generation procedure. (Recall, from Section IV-B, that each edge of the random graph generated by step one, using the $G(n, p)$ model, is replaced by a path of length s, where integer s is selected uniformly from the range $[1, S]$.) Figure 9 summarizes the results for two values of the probability parameter p, with S on the horizontal axis. As before, the vertical axis indicates the required number of colors per segment. We observe that, excluding very low S values, there is a low sensitivity to this graph-generation parameter.

E. Running Time

Our final set of experiments quantifies the running time of our method. Figures 10–12 summarize the results for

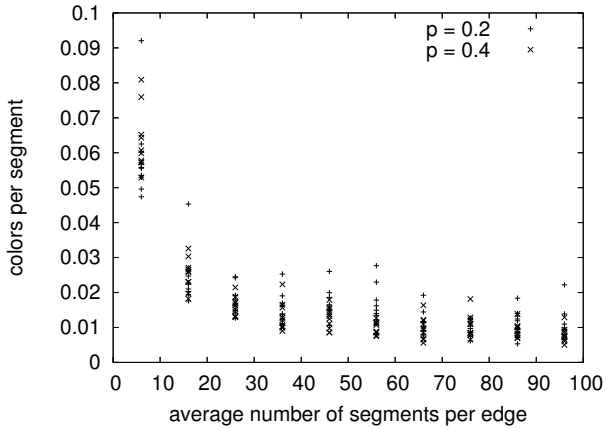

Fig. 9. Number of colors used per segment for varying values of the segmentation parameter.

different values of the parameter k from the definition of kMLC. In each figure, the horizontal axis indicates the size of the graph, measured as the number of segments, and the vertical axis indicates the measured running time, in seconds. Time was measured by simply subtracting the wall-clock time at the beginning of program execution from the time at program termination, without any corrections for concurrently running processes. In each figure, we also plot the fitted curve of the form $y = ax^{2.5}$ as a reference.

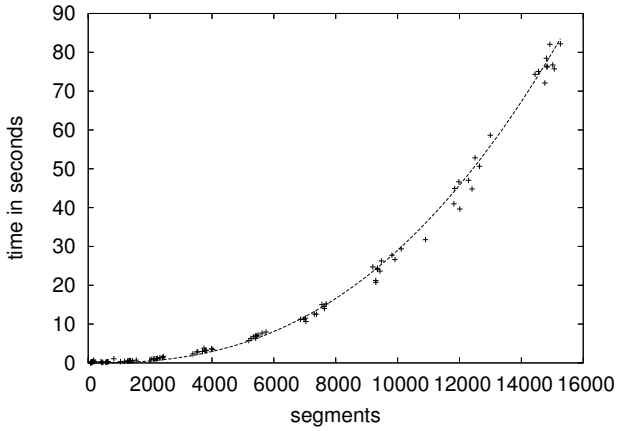

Fig. 10. Running time for $k = 2$.

V. Related Work

Our work in this paper has been motivated by work on the M-CubITS system [1], [3], [4], [5], [6]. As indicated by that work, marker-based methods are attractive in both indoor and outdoor settings, and have been validated experimentally in the field. In particular, the M-CubITS implementation using cameras that detects markers that are patterned tiles attached to the floor or sidewalk is directly relevant to our work in this paper, as discussed in Section I.

A few variants of our idea of a k-step localizing coloring (Section II) have been addressed in other domains. Perhaps the closest are the notions of trackability and observable

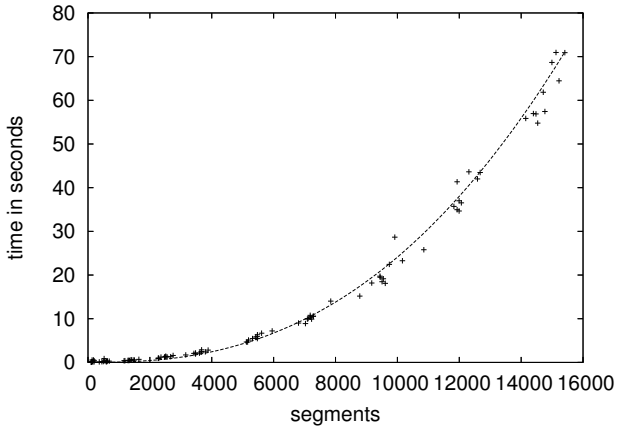

Fig. 11. Running time for $k = 3$.

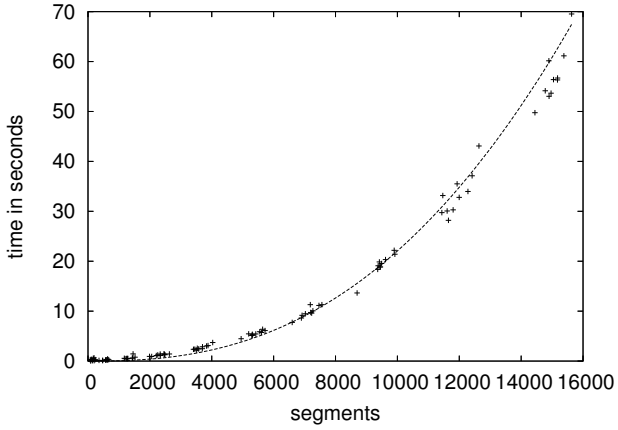

Fig. 12. Running time for $k = 4$.

graphs [8], and the Viterbi algorithm [9], [10]. However, the definition of observable graphs places no a priori limit on the length of the walk that must be traversed before a location is identified. While this variation raises some interesting questions, it is not suitable for our motivating applications in which it is important to identify a location using a small number of steps.

Another related idea is that of the distinguishing number of a graph [11], [12], [13], [14]. Much of this work developed from the following recreational problem posed by Frank Rubin in 1979 [15], [16]: A person must distinguish keys on a key-ring (circular) using only colored markers placed on the keys. What is the minimum number of distinct colors that can achieve this goal? Although there has been much work on variants of this problem, it is not directly relevant to our applications because it is typically permissible to use non-local information in the graph for the purpose of uniquely identifying locations. In our applications, it is important that only local information be used, since non-local markers are typically invisible.

VI. Conclusion

We have described a marker-based method for indoor localizing in which multiple locations are, in general, assigned identical markers. Locations with identical markers are distinguished on the basis of other markers that are encountered by walks leading to those locations. We have presented an algorithm for assigning markers to locations to yield a scheme that satisfies this property. The key benefit of this method is that it permits a super-linear growth in the number of identifiable locations as the number of distinct markers is increased. We have presented experimental results that quantify this benefit on synthetic environments and that illustrate the efficiency of our method. In continuing work, we are evaluating our method in real environments. Although, for concreteness, this paper has focused on localizing for indoor navigation applications, our method is applicable in several other settings, both indoor and outdoor.

References

[1] Tetsuya Manabe, Seiji Yamashita, and Takaaki Hasegawa, "On the M-CubITS pedestrian navigation system," in *Proceedings of the 9th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Toronto, Canada, Sept. 2006, pp. 793–798.

[2] Béla Bollobás, *Modern Graph Theory*, ser. Graduate Texts in Mathematics. New York: Springer, 1998, vol. 184.

[3] Je-Yeon Kim and Takaaki Hasegawa, "An experimental study on the positioning by M-CubITS," in *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Washington, D.C., Oct. 2004, pp. 977–981.

[4] Seiji Yamashita and Takaaki Hasegawa, "On the M-CubITS pedestrian navigation system by a camera-equipped mobile phone," in *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Washington, D.C., Oct. 2004, pp. 714–717.

[5] ——, "On the M-CubITS pedestrian navigation system using textured paving blocks and its experiments," in *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Vienna, Austria, Sept. 2005, pp. 121–125.

[6] Je-Yeon Kim and Takaaki Hasegawa, "On repositioning of the M-sequence lane markers system in highways and applications in intersections," in *Proceedings of the 6th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Shanghai, China, Oct. 2003, pp. 520–523.

[7] Valentino Crespi, George V. Cybenko, and Guofei Jiang, "The theory of trackability with applications to sensor networks," Dartmouth College, Computer Science, Hanover, New Hampshire, Tech. Rep. TR2005-555, Aug. 2005.

[8] Rapha el Jungers and Vincent D. Blondel, "Observable graphs," arXiv:cs/0702091v1. http://arxiv.org/, Feb. 2006.

[9] Andrew J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.

[10] Lawrence R. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[11] Christine T. Cheng, "On computing the distinguishing numbers of trees and forests," *The Electronic Journal of Combinatorics*, vol. 13, no. 1, p. R11, Feb. 2006.

[12] Karen L. Collins and Ann N. Trenk, "The distinguishing chromatic number," *The Electronic Journal of Combinatorics*, vol. 13, no. 1, p. R16, Feb. 2006.

[13] Michael O. Albertson and Debra L. Boutin, "Distinguishing geometric graphs," *Journal of Graph Theory*, vol. 53, pp. 135–150, 2006.

[14] Michael O. Albertson, "Distinguishing Cartesian powers of graphs," *The Electronic Journal of Combinatorics*, vol. 12, no. 1, p. N17, Sept. 2005.

[15] Frank Rubin, "Problem 729: the blind man's keys," *Journal of Recreational Mathematics*, vol. 11, no. 2, p. 128, 1979, solution in 12(2), 1980.

[16] Michael O. Albertson and Karen L. Collins, "Symmetry breaking in graphs," *The Electronic Journal of Combinatorics*, vol. 3, no. 1, p. R18, June 1996.