A coding approach to localization using landmarks

Juan Carlo Rebanal
ECE, UCLA
Los Angeles, USA
jrebanal@g.ucla.edu

Yahya H. Ezzeldin
ECE, UCLA
Los Angeles, USA
yezzeldin@g.ucla.edu

Christina Fragouli
ECE, UCLA
Los Angeles, USA
christina.fragouli@ucla.edu

Paulo Tabuada
ECE, UCLA
Los Angeles, USA
tabuada@ee.ucla.edu

GLOBECOM 2020 - 2020 IEEE Global Communications Conference | 978-1-7281-8298-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/GLOBECOM42002.2020.9347982

***Abstract*—Fully autonomous vehicles need the ability to localize without external help, for instance by using visual sensors together with a pre-loaded map of landmarks. In this paper we connect self-localization using landmarks with coding theory. This connection enables to translate Hamming distance properties to probabilistic localization guarantees given a certain number of errors in landmark identification; it also enables to leverage existing polynomial time decoding algorithms for localization. We present promising numerical evaluation results by simulating vehicle traveling paths along a road network generated from real data of a region in Washington D.C.**

***Index Terms*—localization, autonomous navigation, landmarks, coding theory, OpenStreetMap, graphs**

I. Introduction

The ability to self-localize is a critical component in autonomous navigation, and accordingly, the literature on localization is rich and extensive, ranging from GPS-based approaches [1], [2], [3], [4] and approaches that use other artificial beacons [5], [6], [7], [8], to approaches that leverage natural landmarks, such as environmental structures or curvature [9], [10], [11], [12], [13], [14], [15]. In this paper, we propose a framework that connects self-localization using landmarks with coding theory.

An application of our approach is to provide a backup safety net for fully autonomous vehicles that need to self-localize under all types of conditions. Autonomous vehicles mainly rely on the global positioning system (GPS) for their everyday navigation; yet a fully autonomous vehicle should be able to also localize in GPS-denied environments, such as when GPS reception is weak due to the common presence of skyscrapers or tunnels [16], [17], [18] or when an adversary is spoofing information [19]. In such cases, the vehicle may leverage for instance visual sensors together with a pre-loaded map of landmarks, to self-localize without external help.

Using natural landmarks for localization is intuitive to humans; "Meet me by the tall tree" or some variant of this phrase has likely been encountered by the reader. It has also been extensively explored mainly within the robotics literature from several angles, such as by using semantic observations [20], building maps of landmarks [21], performing data association of sensor outputs with natural landmarks [22], and combinations of the above tasks [23].

Y. H. Ezzeldin and C. Fragouli were supported in part by NSF awards 1740047, 1824568 and UC-NL grant LFR-18-548554. P. Tabuada was also supported by NSF award 1740047.

In this paper, we consider an abstract framework that captures two main challenges: the fact that we want to use landmarks that are not unique but plentifully available (e.g., use trees v.s. the Capitol as landmarks); and the fact that there may be errors in landmark identification, either because our sensors are not accurate (e.g. fog obstructing visual sensors) or simply because the landmarks themselves change (e.g. a tree got pruned or a trash can was moved). To address the first challenge, we can consider sequences of landmark observations: a local traversal of the environment may create a unique sequence of individually non-unique landmarks. This is intuitive for humans, where the potential ambiguity: "Which tall tree?" can be resolved by an expansion of the local landmarks considered: "The tall tree next to the two fire hydrants". To address the second challenge, we leverage coding theory, where distance between sequences (that act as codewords) can help correct errors.

The fact that errors can hinder localization is already recognized, yet as far as we know the connection to coding is new to this paper. Work has looked at mitigating errors that happen during data association - that is, when mapping sensor inputs to landmarks - mainly by refining and combining several sensor inputs [22]. Yet, localization techniques can fail if data acquisition is performed incorrectly or error correction measures such as loop closure are not used [24]. Moreover, the evidence that proposed localization algorithms perform well is mostly empirical, through evaluation. Our goal is different: we accept that errors may happen, and aim to provide probabilistic guarantees on uniquely determining a position as a function of the length of the observed sequence of landmarks.

Our contributions in this paper are as follows. First, we make a connection between the problem of landmark-based self-localization and coding theory, by introducing a special type of code that groups multiple sequences as a single codeword. Leveraging this connection, we translate Hamming distance properties to probabilistic guarantees on the worst-case length of landmark sequences that need to be sensed for localization, given a number of expected errors. Our approach is, as far as we know, the first enabling the derivation of localization guarantees through offline preprocessing of a map and thus can be used to guide the selection and density of landmarks to achieve a desired performance[1]. Finally, this connection allows us to leverage existing efficient "decoding"

[1] We do not explore the latter direction in this paper, but refer to [25].

algorithms for localization. We argue that our approach can be easily deployed, by constructing a landmarks map using data currently available online, such as the Open Street Map (OSM) [26] data. In our numerical evaluation, we find that even with quite a generic set of landmarks (such as curbside hydrants, trash cans, etc.), it is feasible for a vehicle to fast self-locate within a large area of Washington D.C.

We want to highlight that although in the rest of the paper our approach is expressed in terms of self-localization for vehicles, the connection to coding theory is an abstract concept and is true for all self-localization applications.

Related Work. There is extensive literature on localization using GPS and other artificial signals [6], [7], [8]; our approach is complementary to these, focusing on autonomous localization. Autonomous vehicle self-localization is a well-researched field, including work with roots in localization in robotics [9], [27]. Landmark detection-based methods have been explored with both natural and artificial landmarks [5], [15], [28]; a currently popular approach is simultaneous localization and mapping (SLAM) [29], that faces the challenge of scaling for larger (outdoor) environments [23]. Dealing with errors is mainly addressed at the level of associating landmarks with sensing inputs (e.g., by increasing the amount of sensing), while position identification uses Bayesian inference algorithms such as particle filtering [16], [10].

Paper Organization: Section II introduces the problem statement and develops the connection to coding theory; Section III discusses probabilistic localization; Section IV connects decoding algorithms to localization; Section IV presents numerical evaluation results using real street-level data; and Section V concludes the paper.

II. From Self-Localization to Coding

In this section, we build a connection between the self-localization problem and decodability of a channel code.

A. Self-Localization Problem

We assume the availability of an offline large-scale map of the area in which self-localization should be achieved[2]. In particular, we are given a labeled, directed graph $G = (V, E, h(E))$, with vertex set V, edge set $E \subseteq V \times V$, and a labeling function $h : E \to \mathcal{A}$, where: the edges in the set E represent different street segments; the vertices in the set V represent the legal transitions between street segments (for instance: intersection, fork-outs, etc.); $\mathcal{A} = \mathcal{B}^m$ denotes landmark vectors of length m with entries from $\mathcal{B}$, where $\mathcal{B}$ is a finite alphabet of types of landmarks (such as: tree, traffic light, fire hydrant, trash can, etc.); and $h(e)$ describes a set of landmarks of size at most m that would ideally (in the error-free case) be sensed locally by a vehicle traversing the street segment $e \in E$. A path[3] p of length n on G is a function $p : \{0, 1, ..., n\} \to V$ satisfying $(p(i), p(i+1)) \in E$ for $i = 0, \ldots, n-1$. Each path p of length n in G defines an element of $\mathcal{A}^n$ given by the sequence of labels (landmarks) $[h(p(0), p(1)), \ldots, h(p(n-1), p(n))]$ which we denote by y_p (we can equivalently think of y_p as a length mn vector with elements in $\mathcal{B}$) . When traversing p in real-life, the vehicle will detect a sequence of landmarks $\hat{y}_p \in \mathcal{A}^n$, which may not necessarily be equal to y_p, due to any source of sensing errors.

[2]Such a map can be constructed using data currently available online, such as the geographical landmark dataset in [30] and the Open Street Map (OSM) [26] data, as we do in our numerical evaluation.

[3]Our definition of path does not require that edges or vertices in the path are distinct, and coincides with the definition of a walk in graph theory textbooks [31].

B. Coding Formulation

We here provide a high level intuition of how we map the self-localization problem to a coding problem, and give the formal notation and definitions later in this section. In the following, we use the standard definition of Hamming distance, that measures the number of positions where two sequences of the same length differ [32]; we also use the well known result that a code allows to correct up to t errors if and only if the Hamming distance between any two codewords is at least $2t + 1$ [33].

A straightforward mapping of the self-localization problem to a coding problem would be to treat the sequence $\hat{y}_p$ associated with every path p in our graph as a codeword. Then the set of paths of a given length ℓ would lead to a codebook that would contain as codewords the corresponding sequences $\hat{y}_p \in \mathcal{A}^\ell$, and we could examine the Hamming distance of the code to derive error correction bounds. This approach adds unnecessary complexity to our problem (the number of paths increases exponentially with ℓ), as we observe next.

Observation 1: If a vehicle can determine that after ℓ steps it is at some vertex v_ℓ, this is sufficient to self-localize; it does not need to retrieve the path p it followed to arrive to v_ℓ.

This observation indicates that we should use as codewords groups of paths. In particular, if the vehicle moves in a graph with $|V|$ vertices for ℓ steps, the code we will consider will have $|V|$ codewords, one for each vertex; the codeword associated with vertex v will consist of all paths of length ℓ that end at v. It is among these codewords we need to distinguish to be able to localize our vehicle, and thus, *we can tolerate ambiguity among different paths that terminate at the same vertex.*

Note that, even with this approach, the worst case Hamming distance could be small: indeed, consider two paths of length ℓ that have the first $\ell - 1$ edges in common and only diverge at the last step (in the example of Fig. 1, two such paths would be $\{i, j, b, c\}$ and $\{i, j, b, e\}$). Clearly, the Hamming distance between two such paths, would be at most m (recall each edge is associated with at most m landmarks), since they may differ in at most m positions. Such a code would be able to correct $\max\{0, \lfloor \frac{m-1}{2} \rfloor\}$ errors (in Fig. 1, $m = 1$, and thus not-correctable errors can occur). That is, we will not be always able to give hard guarantees on error correction. However:

Observation 2: Given a number of expected errors, we can translate Hamming distances between codewords to probabilistic guarantees.

These guarantees would be in the following sense: assume that after ℓ steps the car is at vertex v selected uniformly at

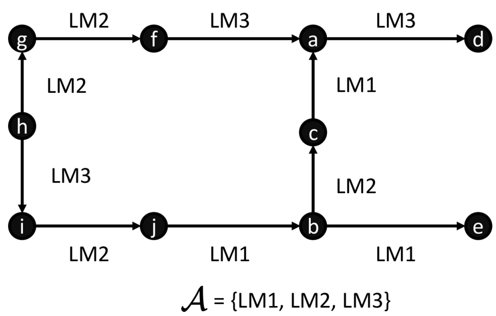

Fig. 1. Example of a labeled directed graph with landmarks sensed along each street segment (edge).

random from V; then we can calculate a lower bound on the probability that we can successfully localize at v.

C. Questions

Given the above formulation, we abstract two aspects of the self-localization problem under t sensing errors.

Question 1 (probabilistic-localization): Consider a map described by a graph G where the vehicle starts at an arbitrary $v_0 \in V$, and follows a path p of length ℓ to a vertex v_ℓ. Assume that at most t errors occur when sensing the landmark sequence $\hat{y}_p$. We ask: what (probabilistic) guarantees can we offer to uniquely determine the position of the vehicle $p(v_\ell)$ from the observed landmarks sequence $\hat{y}_p \in \mathcal{A}^\ell$ for an $\ell \leq n_t$?

Question 2 (efficient-localization): Given a map described by a graph G where the vehicle starts at an arbitrary $v \in V$, and assuming that landmark sequence $\hat{y}_p$ is observed while following a path p. If the last vertex $p(n_\ell)$ is uniquely decodable, can we identify it in polynomial-time?

Remark 1: Note that in the aforementioned abstract formulation, we are interested in how fast and effectively localization can be achieved over G (or its corresponding real-life map) given an upper bound on the number of steps n_t. On the other hand, the localization precision is a property of the graph G that depends on the definition of the vertex set V. For instance, if each vertex $v \in V$ corresponds to a street section of 1m length, then the localization precision is within 1m. Similarly, if it corresponds to 5m, that reflects on the localization precision.

D. Coding Notation

Let $\mathcal{X}$ be a finite set of elements, $\mathcal{B}$ a finite alphabet of symbols and $\mathcal{A} = \mathcal{B}^m$ an alphabet of vectors of length m with entries from $\mathcal{B}$. Given $n \in \mathbb{N}$, we denote with $\mathcal{A}^n$ the set of all possible sequences of length n using the alphabet $\mathcal{A}$ (we can equivalently think of these as sequences of length mn over the alphabet $\mathcal{B}$).

An encoding function $f_n : \mathcal{X} \to \mathcal{A}^n$ maps an element $x \in \mathcal{X}$ to a subset $f_n(x)$ of $\mathcal{A}^n$. We will use the notation $f_n(\mathcal{S})$ to denote $\cup_{s \in S} f_n(s)$, $\mathcal{S} \subseteq \mathcal{X}$. The union $C_n = \cup_{x \in \mathcal{X}} f_n(x)$ defines a code $C \subseteq \mathcal{A}^n$.

Error correcting code. A code $C_n \subseteq \mathcal{A}^n = \mathcal{B}^{mn}$ is said to be uniquely decodable and *t-error correcting* if there exists a (decoding) map $g : \mathcal{A}^n \to X$ satisfying that, $\forall x \in \mathcal{X}$ and $\forall \sigma \in f_n(x)$:

$$d_H(\tilde{\sigma}, \sigma) \leq t \implies g(\tilde{\sigma}) = x, \quad (1)$$

where $d_H(y, z)$ is the Hamming distance between sequences y and z calculated with respect to symbols in $\mathcal{B}$. Equivalently, (1) implies that $\forall x_1 \neq x_2 \in \mathcal{X}$ with distance $d_H(f_n(x_1), f_n(x_2))$,

$$d_H(f_n(x_1), f_n(x_2)) \geq 2t + 1, \quad (2)$$

where the pairwise distance is calculated as

$$d_H(f_n(x_1), f_n(x_2) = \min_{\substack{\sigma_1 \in f_n(x_i),\\ \sigma_2 \in f_n(x_2)}} : d_H(\sigma_1, \sigma_2).$$

Self-localization as coding. Given a map described by the directed graph $G = (V, E, h)$, we define the set of messages $\mathcal{X}$ to be V, the set of symbols $\mathcal{A}$ representing the co-domain of h, the code $C^{(n)} = \{y_p : p \in P^{(n)}\}$ where $P^{(n)}$ is the set of all paths of length n on the G. The encoding function $f_n : \mathcal{X} \to \mathcal{A}^n$ is given by $f_n(x) = \cup_{p \in P_x^{(n)}} \{y_p\}$ where $P_x^{(n)}$ is the set of paths p of length n in the graph G that terminate at x, i.e., $p(n) = x$. Thus, we partition the set of paths $P^{(n)}$ into $|V|$ disjoint sets $\{f_n(x)\}_{x \in \mathcal{X}}$, each one corresponding to each vertex x.

III. Probabilistic Localization

We observe the following: To uniquely determine that the vehicle is at a specific vertex x after n steps and t errors, we require that for any other vertex y, if $p_i \in f_n(x)$ and $p_j \in f_n(y)$, then $d_H(y_{p_i}, y_{p_j}) \geq 2t + 1$. Otherwise, if the path p_i was followed to vertex x and t errors were observed, we would end up with a landmark sequence $\hat{y}_{p_i}$ that is closer to $f_n(y)$ than $f_n(x)$. If this happens, we will say that vertex x is uniquely decodable after n steps.

Example. Consider the toy graph shown in Fig. 1 where the vertices represent different street intersections and the edge labels represent the landmarks observed on the street segments connecting them. In particular, the number of possibly observed labels is $|\mathcal{A}| = 3$. In Fig. 1, if we observe only the set of paths of length 2 terminating at each vertex, we cannot resolve the ambiguity between vertices a and b even if we consider the case of having no observation errors, i.e. $t = 0$. This is due to the fact that there exist two paths terminating at a and b, respectively, that are identically labeled or equivalently, we have that $\min_{\sigma_a \in f_n(a), \sigma_b \in f_n(b)} d_H(\sigma_a, \sigma_b) < 1$.

As observed in Section IIB, unique decodability is not guaranteed, even if we allow the number of steps n to arbitrarily increase. Indeed, if there exists a common neighbor v connecting to two vertices x and y through directed edges (vx) and (vy), then error correction would not be guaranteed if $t > \max\{0, \lfloor \frac{m-1}{2} \rfloor\}$, where m is the number of landmarks associated with each edge. Instead, we can check what percentage of the codewords have pairwise Hamming distance greater than $2t + 1$. Note that as n increases, this percentage may also increase.

We can extend this observation to derive upper bounds on the pairwise Hamming distance that are independent of the number of steps n.

Proposition 1: Consider a vertex v that is connected to vertices v_1 and v_2 through paths $p_1 = \{v, \ldots, v_1\}$ and $p_2 = \{v, \ldots, v_2\}$ both of the same length k, and let y_{p_1} and y_{p_2} denote the landmark sequences associated with paths p_1 and p_2 respectively. Then

$$d_H(f_n(v_1), f_n(v_2)) \leq d_H(y_{p_1}, y_{p_2}),$$

for all $n \geq k$.
The proof follows directly by considering two paths, one ending at v_1 and the other at v_2, that share a common initial part of length $n-k$ that reaches vertex v, and then split, one to reach v_1 and the other v_2.

Proposition 1 indicates that a method to upper bound the error correcting capabilities of a given map/choice of landmarks, is to find short length paths for pairs of vertices; we note that maps correspond to planar graphs, and thus such calculations could be computationally efficient [34]. We do not pursue further this direction in this paper; but instead calculate through brute force pairwise distances between codewords (we note that this calculation can be performed offline and once for each map).

Given the set of pairwise distances, we translate them to probabilistic guarantees as follows: we say that the probability of successful localization after n steps and t errors is lower bounded by the percentage of pairs where $d_H(f_n(v_1), f_n(v_2)) \geq 2t + 1$.[4]

IV. Localization-decoding algorithm

In this section, we present a polynomial time approach for localization based on ideas for codeword decoding in coding theory, that leverages dynamic programming (our approach is a variation of the well-known Viterbi algorithm [35]). Again we assume that at most t errors may occur.

We consider a trellis of size $|V| \times n_t$, where each of the n_t transitions in the trellis diagram represent the edge connections in the graph G. For the example given in Fig. 2, one transition of the trellis diagram is given in Fig. 2. The first layer in the trellis is connected to a common source vertex and the last layer of the trellis is connected to a common sink vertex. The decoding on this trellis diagram can be performed as follows:

(Stage 1) Given an observed sequence, calculate the costs along each transition from one vertex to another in the trellis, or the "mismatch" of traversing through a particular edge in the k-th transition in the trellis diagram. For instance, for $m = 1$, if the k-th symbol of the observed sequence $\hat{y}[i] = 7$, then for all edges from the k-th layer of the trellis diagram to the $(k+1)$-th layer, we set their cost to be zero if this edge is labeled 7, and set it to one otherwise. For $m > 1$ we count as cost the Hamming distance between the observed and edge-labeling vector in $\mathcal{B}^m$.

[4]More sophisticated translations are also possible at the cost of additional computations; we do not explore this here.

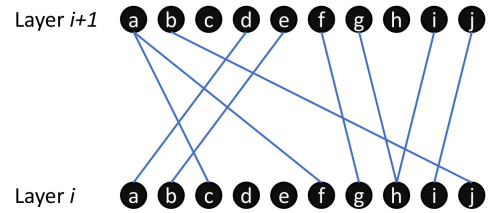

Fig. 2. Trellis transition used for the example graph in Fig. 1.

Result: Returns estimate of current position.
Let $G = (V, E, h(E))$;
$T = Trellis(G)$;
Let $\hat{y}$ be the n_t-length path codeword observed during localization;
Initialize $i = 1$;
for *symbol* $\hat{y}[i] \in x$ **do**
  for *all edges* e *in* i*-th transition in* T **do**
    Set edge weight = $\mathbb{I}(y[i] = h(e))$;
  end
  Increment i;
end
for $i \in \{1, 2, \cdots, n_t - 1\}$ **do**
  - Compute the accumulated cost at each vertex in layer $i+1$ of the trellis using the cost at layer i and edges connecting layers i and $i+1$;
  if $i \geq 2t + 1$ **then**
    if $\exists$ *unique cost minimizer* $v_*^{(i+1)}$ *in layer* $i+1$ **then**
      Return v_*
    end
  end
end

(Stage 2) Given the trellis diagram with the edge costs constructed in Stage 1, we successively compute the minimum cost of a path terminating at node $v^{(i)}$ in layer i of the trellis using any dynamic programming algorithm for finding minimum cost path. For each $i \geq t+1$ if there exists a single vertex that has a cost less than or equal to t (all other vertices have cost more than t - or more than t errors, which we assume not possible), we terminate the algorithm and return this vertex as the found location.

Remark 2: We note that a vehicle can attempt decoding at each step as it moves through the map - does not need to wait to collect sequences of symbols.

The pseudocode of the algorithm discussed above is given in Algorithm 1. The complexity of Algorithm 1 is dominated by (Stage 2), where the minimum cost path is found. There are a number of algorithms that can be employed in this step as subroutines, for instance, Viterbi's algorithm which can perform the task in $O(n_t(|E| + |V|))$ time, which drives the computational complexity of Algorithm 1.

V. Experimental Evaluation

For our performance evaluation, we simulate transitions performed in a real-world map of Washington D.C. We create the graph $G = (V, E, h(E))$ describing the map, by employing the geographical landmark dataset used in [30] in addition to Open Street Map (OSM) [26] data. In particular, we study a subset of the Washington D.C. area constrained by the coverage of the dataset [30] with an area of 10.097 km^2. Next, we give a brief description of how the graph G and the labeling function h were constructed in our experiments (we refer to reader to the extended version [36] for details).

Structure of graph G. The OSM data provides a graphical representation of the street map in terms of intersections and street segments connecting them. The set of intersections were used as the vertex set V and the street segments were used as the edge set E.

Landmarks. The landmark object types considered were fire hydrants, street lights, traffic lights, trash cans, and traffic signs, which each satisfy properties of desirable landmarks [15], [12]. Together with the graph structure, this allows for a topological map-like representation [37] of the regions studied.

Labeling function $h(E)$ of G. To construct the labels for each street segment, a rectangular contour was constructed along the street segment and only the considered landmarks in [30] that fall within the street segment contour are considered part of the label. In addition to that, we added three additional landmarks that can be observed by an autonomous vehicle: the absolute bearing of a street segment (e.g. the onboard compass reading) binned to 1 of 8 equal bearings, the lengths of street segments (e.g. an odometer equipment reading a street segment is 50m before the next turn) with bins every 2m, and the bidirectionality of the street as a bit. For each street segment, a vector of landmarks, e.g. $[c_1, c_2, \ldots, c_8]$, was created from individual landmarks c_i. Therefore, a path of length ℓ has an associated label of 8ℓ symbols that are each used in calculating Hamming distances.

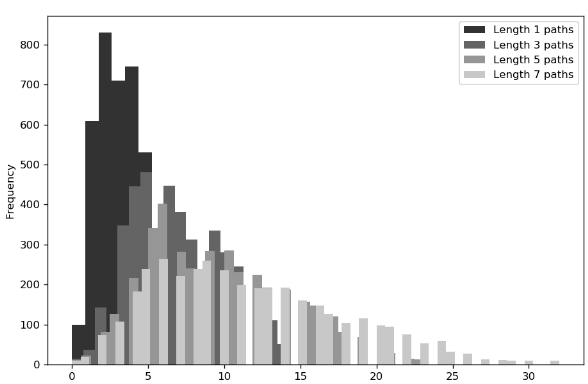

Fig. 3. Overlapping histograms of minimum Hamming distances of path groups for different path lengths.

Table I provides probabilistic guarantees, as a function of the path length (columns) the vehicle traverses, and the number of errors (rows) it experiences.

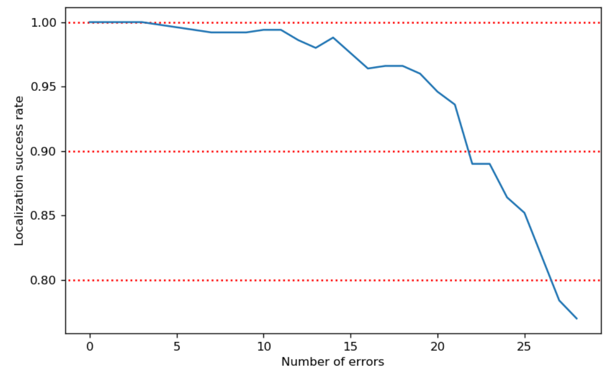

Fig. 4. Localization success rate with different amounts of errors introduced for length-7 paths. Red dotted lines show thresholds of 1.0, 0.9, and 0.8.

TABLE I
Proportions of localizable locations for D.C. dataset.

Errors	Length 1	Length 3	Length 5	Length 7
0	0.9802	0.9992	0.9997	0.9997
1	0.6290	0.9592	0.9775	0.9794
2	0.2738	0.7603	0.8932	0.9068
3	0.0733	0.5266	0.7037	0.7783

These guarantees are derived offline by preprocessing the map and calculating pairwise Hamming distances, as discussed in Section III. We find that with this map and landmark choices, with no errors, even after a single edge the vehicle can uniquely self-localize with probability more than 98%; with one error the vehicle needs to travel up to 3 edges to localize with probabilities 97%; while with three errors, after 7 edges the localization probability is above 77%. We underline that these guarantees are pessimistic (lower bounds); indeed in our performance evaluation through numerics (wee Fig. 4 that we discuss later), paths of length 7 led to successful localization even with 20 errors in more than 90% of the simulated cases.

Fig. 3 shows overlapping histograms of pairwise Hamming distances for each group of paths associated with an ending vertex and different path lengths taken as an average over 500 trials. As the path length increases, the histograms flatten and the tail grows further out as expected, yet there always remain some small distance cases as discussed in Section II and III.

Fig. 4 presents the average performance of the decoding algorithm, for uniformly at random selected length-7 walks, averaged over 500 trials and as a function of the number of errors. We find that even with 20 identification errors perturbing the ground truth landmark sequence, 94.6% of the cases still led to successful localization.

VI. Conclusion

In this paper, we present a connection between the self-localization problem and coding theory. By representing a road network as a graph, landmark observations can be encoded in vectors assigned to different edges representing street segments. Sequences of landmarks are analogous to codewords in coding theory, where conditions on decodability and error-correction can be applied to understand the limitation of the road network representation as well as the number of paths needed for self-localization within this map. We thus argue that through this connection, we can preprocess existing maps and provide probabilistic guarantees on fault tolerance. In our evaluations, our decoding algorithm was able to localize in over 94% of randomly chosen walks even when 35% of the walk's labels were corrupted with errors, demonstrating its robustness to observation errors.

References

[1] Z. Tao, P. Bonnifait, V. Fremont, and J. Ibanez-Guzman, "Mapping and localization using gps, lane markings and proprioceptive sensors," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 406–412.

[2] M. A. Al-Khedher, "Hybrid gps-gsm localization of automobile tracking system," *arXiv preprint arXiv:1201.2630*, 2012.

[3] M. Agrawal and K. Konolige, "Real-time localization in outdoor environments using stereo vision and inexpensive gps," in *18th International conference on pattern recognition (ICPR'06)*, vol. 3. IEEE, 2006, pp. 1063–1068.

[4] S. Nirjon, J. Liu, G. DeJean, B. Priyantha, Y. Jin, and T. Hart, "Coin-gps: indoor localization from direct gps receiving," in *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, 2014, pp. 301–314.

[5] M. L. Sichitiu and V. Ramadurai, "Localization of wireless sensor networks with a mobile beacon," in *2004 IEEE international conference on mobile Ad-hoc and sensor systems (IEEE Cat. No. 04EX975)*. IEEE, 2004, pp. 174–183.

[6] K. Ok, D. Kwon, and Y. Ji, "Bluetooth beacon-based indoor localization using self-learning neural network," in *The 3rd International Workshop on Deep Learning for Mobile Systems and Applications*, 2019, pp. 25–27.

[7] P. Gautam, S. Kumar, A. Verma, and A. Kumar, "A novel energy-efficient localization of sensor nodes in wsns using single beacon node," *IET Communications*, 2020.

[8] E. Olson, J. J. Leonard, and S. Teller, "Robust range-only beacon localization," *IEEE Journal of Oceanic Engineering*, vol. 31, no. 4, pp. 949–958, 2006.

[9] M. Liu, X. Lei, S. Zhang, and B. Mu, "Natural landmark extraction in 2D laser data based on local curvature scale for mobile robot navigation," *2010 IEEE International Conference on Robotics and Biomimetics, ROBIO 2010*, pp. 525–530, 2010.

[10] C. Ye, G. Chen, S. Qu, Q. Yang, K. Chen, J. Du, and R. Hu, "Self-Localization of Parking Robots Using Square-Like Landmarks," 2018. [Online]. Available: http://arxiv.org/abs/1812.09668

[11] N. Engel, S. Hoermann, M. Horn, V. Belagiannis, and K. Dietmayer, "DeepLocalization: Landmark-based Self-Localization with Deep Neural Networks," 2019. [Online]. Available: http://arxiv.org/abs/1904.09007

[12] A. Kampker, J. Hatzenbuehler, L. Klein, M. Sefati, K. D. Kreiskoether, and D. Gert, "Concept study for vehicle self-localization using neural networks for detection of pole-like landmarks," *Advances in Intelligent Systems and Computing*, vol. 867, pp. 689–705, 2019.

[13] R. Spangenberg, D. Goehring, and R. Rojas, "Pole-based localization for autonomous vehicles in urban scenarios," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016-Novem, pp. 2161–2166, 2016.

[14] C. Brenner, "Global localization of vehicles using local pole patterns," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5748 LNCS, pp. 61–70, 2009.

[15] R. Madhavan and H. F. Durrant-Whyte, "Natural landmark-based autonomous vehicle navigation," *Robotics and Autonomous Systems*, vol. 46, no. 2, pp. 79–95, 2004.

[16] M. A. Brubaker, A. Geiger, and R. Urtasun, "Map-based probabilistic visual self-localization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 4, pp. 652–665, 2016.

[17] D.-V. Nguyen, F. Nashashibi, T.-K. Dao, and E. Castelli, "Improving poor gps area localization for intelligent vehicles," in *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE, 2017, pp. 417–421.

[18] J. Surber, L. Teixeira, and M. Chli, "Robust visual-inertial localization with weak gps priors for repetitive uav flights," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 6300–6306.

[19] I. Sajjad, D. D. Dunn, R. Sharma, and R. Gerdes, "Attack mitigation in adversarial platooning using detection-based sliding mode control," in *Proceedings of the First ACM Workshop on Cyber-Physical Systems-Security and/or PrivaCy*, 2015, pp. 43–53.

[20] N. Atanasov, M. Zhu, K. Daniilidis, and G. J. Pappas, "Localization from semantic observations via the matrix permanent," *International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 73–99, 2016.

[21] K. Basye, T. Dean, and J. S. Vitter, "Coping with Uncertainty in Map Learning," *Machine Learning*, vol. 29, no. 1, pp. 65–88, 1997.

[22] Y. Bar-Shalom, F. Daum, and J. Huang, "The Probabilistic Data Association Filter: Estimation in the presence of measurement origin uncertainty," *IEEE Control Systems*, vol. 29, no. 6, pp. 82–100, 2009.

[23] T. Bailey, "Mobile Robot Localisation and Mapping in Extensive Outdoor Environments," *Philosophy*, vol. 31, no. August, p. 212, 2002. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.4.1707

[24] U. Frese, "Treemap: An O(log n) algorithm for indoor simultaneous localization and mapping," *Autonomous Robots*, vol. 21, no. 2, pp. 103–122, 2006.

[25] J. C. Rebanal, "Self-localization of autonomous vehicles using landmark object detection," Master's thesis, UCLA, 2020.

[26] "Open street map," Website, https://www.openstreetmap.org/.

[27] M. Betke and L. Gurvits, "Mobile Robot Localization using Landmarks," no. May 1997, 2013.

[28] J. Levinson and S. Thrun, "RobustVehicleLocalizationinUrbanEnvironmentsUsingProbabilisticMaps.pdf," pp. 4372–4378, 2010.

[29] M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Transactions on robotics and automation*, vol. 17, no. 3, pp. 229–241, 2001.

[30] S. Ardeshir, A. R. Zamir, A. Torroella, and M. Shah, "GIS-assisted object detection and geospatial localization," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8694 LNCS, no. PART 6, pp. 602–617, 2014.

[31] J. A. Bondy, U. S. R. Murty *et al.*, *Graph theory with applications*. Macmillan London, 1976, vol. 290.

[32] R. W. Hamming, "Error detecting and error correcting codes," *The Bell system technical journal*, vol. 29, no. 2, pp. 147–160, 1950.

[33] S. Roman, *Introduction to coding and information theory*. Springer Science & Business Media, 1996.

[34] G. N. Frederickson, "Planar graph decomposition and all pairs shortest paths," *Journal of the ACM (JACM)*, vol. 38, no. 1, pp. 162–204, 1991.

[35] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[36] J. C. Rebanal, "Appendix," 2020. [Online]. Available: http://arni.ee.ucla.edu/_media/group/localization.pdf

[37] A. Angeli, S. Doncieux, J.-A. Meyer, and D. Filliat, "Visual topological slam and global localization," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 4300–4305.