Maximum Entropy Bridgelets for Trajectory Completion

John Krumm
jckrumm@microsoft.com
Microsoft Research
Microsoft Corporation
Redmond, Washington, USA

ABSTRACT

Location measurements from people and vehicles often have long temporal gaps between them. However, we would still like to reason about location behavior during these gaps. This paper presents a new method for filling these gaps that is both principled and data-driven. Unlike the most common method, linear interpolation, our method explicitly represents the location uncertainty in the gaps with probability. It also learns from actual mobility data. We introduce bridgelets, which are small, spatio-temporal, maximum entropy clouds that model spatial uncertainty over small gaps. Using actual trajectories, we combine bridgelets into probabilistic bridges that are specific to absolute start and end locations on the map. The resulting bridges give the probability of visiting certain in-between locations given only the start and end points. Using real trajectory data, we compare our maximum entropy bridges to a popular baseline to show how our approach is much more accurate.

CCS CONCEPTS

• **Information systems → Geographic information systems**.

KEYWORDS

GPS, trajectory, interpolation, bridge, geospatial

ACM Reference Format:

John Krumm. 2022. Maximum Entropy Bridgelets for Trajectory Completion. In *The 30th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '22), November 1–4, 2022, Seattle, WA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3557915.3561015

1 INTRODUCTION

Location trajectories of moving people, animals, and objects are often sparsely sampled due to concerns about privacy, battery life, and storage capacity. However, it is important to be able to reason about what happens between location measurements, such as inferring a visit to a certain location or understanding the probability of disease exposure.

The most common method for inferring locations between measurements is linear interpolation, assuming a constant speed. But when the known endpoints are far apart, this assumption of straight line, constant speed becomes unrealistic, given the nature of normal movement in the world. Other deterministic interpolants are no better, because we have no reason to assume motion generally follows any particular non-straight curve.

A more promising approach is to explicitly represent the inherent uncertainty in location between measurements. In the geospatial field, this has been expressed with beads [3, 6], which represent all feasible locations that could be visited between two timestamped points, given a maximum speed. While beads give just a feasible visit region, some researchers in animal tracking advocate for the Brownian bridge [5], which expresses the uncertain in-between behavior as a time-varying, spatial probability distribution. Similarly, a Gaussian process can be used in the same way [11]. However, beads, Brownian bridges, and Gaussian processes are insensitive to ground features such as paths, roads, terrain, and obstacles, effectively modeling motion as free movement on a featureless plane. The Brownian bridge has one free parameter, the diffusion coefficient, that can be trained from mobility data [7], and the Gaussian process has a handful of parameters and a kernel function to choose. These limited parameterizations are inadequate to account for the rich and varied motion induced by personal preferences and path constraints that change from place to place.

This paper introduces a new type of mobility bridge that simultaneously accounts for the uncertainty between measured points and learns from motion data. The technique works on a spatial grid, discretizing each location point to a spatial cell ($100\,\text{m} \times 100\,\text{m}$ in our case). A single trajectory is characterized by a sequence of timestamped cells, where the timestamps are also discretized into short time periods (5 s in our case). Between each pair of temporally adjacent cells, we insert a maximum entropy bridgelet. This bridgelet spans the space and time between two cells with a spatial cloud of all feasible paths between the two cells, constrained only by the time span between the cells. The bridgelet is "maximum entropy" because it makes no assumptions about the paths between the two cells, and each path has the same, uniform probability. This bridgelet-augmented trajectory represents a candidate bridge for any two points that share this trajectory's start cell, end cell, and discrete time span. If there are other training trajectories with the same start, end, and time span, we show how we can compute the mean of their associated bridges to make a final, learned bridge. The unique advantages of this approach are that it makes no prior assumptions about behavior between measured points (i.e. using maximum entropy bridgelets) and that it learns from actual mobility data to reduce the uncertainty in a principled way (i.e. taking the mean of the bridges from similar training trajectories). In the end, for each start cell, end cell, and time span, we have a bridge that properly represents both the spatial uncertainty and observed behavior between the two locations.

SIGSPATIAL '22, November 1–4, 2022, Seattle, WA, USA

ACM ISBN 978-1-4503-9529-8/22/11…$15.00
https://doi.org/10.1145/3557915.3561015

As we will show, the advantages of our approach to trajectory completion are:

- Initially unbiased representation of feasible paths via maximum entropy
- Bridges properly blended with mobility data for realistic representation of movement constraints and preferences
- Bridges maintain non-zero probability of feasible location visits even if they have not been seen in data
- Variety of probability distributions from bridges, including visit probabilities and dwell probabilities

2 RELATED WORK

The problem of inferring the location of a moving object between measured points in time has received attention in the research literature. The most commonly used solution to this problem is simply linear interpolation. For longer distances on the globe, researchers have used great circle interpolation. In each case, the moving object is assumed to move at a constant speed along the most direct path between the two endpoints. This has the advantage of simplicity, but does not admit wandering, turns, nor changes in speed. For a discrete grid in (x, y), as we use, a grid traversal algorithm such as [1] can be used to identify all the intersected grid cells between the two endpoints.

If linear interpolation gives the most constrained specification of the unknown path between two points, the “lifeline bead” [3, 6] gives one of the most liberal. The lifeline bead computes every possible location the object could have been between the two endpoints given a specified maximum speed and the traversal time. In spacetime, the region of possible visits is the intersection of two half-cones. For an entire trajectory of multiple points, lifeline beads are strung together to form a necklace. A lifeline bead does not specify any probability of visiting a given (x, y, t), but instead gives the points that could (and implicitly could not) be visited in light of the endpoints, traversal time, and maximum speed.

The previous two approaches are not probabilistic. A more sophisticated, probabilistic approach is the Brownian bridge. It is a conditional probability density function of location, describing the distribution of location as a function of time between two known points. The Brownian bridge is based on Brownian motion, or Wiener process, which classically describes a particle undergoing a random walk. Practically, a Brownian bridge says that the location of the particle is a time-varying two-dimensional Gaussian distribution whose variance is zero at the endpoints and maximum in the middle. A readable explanation of the Brownian bridge, applied to animal tracking, appears in the paper by Horne [5]. The Brownian bridge has been shown to be a poor representation of human mobility [7].

The Gaussian process is also a viable approach to creating a spatial probability distribution between measured points. This approach was recently used by Nguyen et al. [11] for understanding the information content of sampled trajectories. Similar to the Brownian bridge, a Gaussian process produces a time-varying, continuous probability distribution over location.

The work by Emrich et al. [2] and Niedermayer et al. [12] addresses how to do probabilistic queries on uncertain moving object data that have been subjected to a Markov model. This is similar to our proposed approach in that time and space are both discretized. While the referenced work gives a deep analysis of the Markov model and a sophisticated query structure around this model, the intellectual overlap with our work is the Markov model. Their work does not concentrate on the realism nor accuracy of the model, but rather on the use of the probabilistic model for queries. Using techniques in the paper, the Markov model could be adapted to compute statistics on in-between behavior as we do in this paper with our maximum entropy model.

The "random walk bridge" is a known mathematical entity similar to what we study [4]. It represents a sequence of moves in discrete time and space. The random walk bridge incorporates random moves at every step, while our bridgelets constitute a collection of deterministic walks between two discrete points, and we assume that one was chosen at random.

Out work is unique in that it employs the maximum entropy concept to model the inherent uncertainty between measurement points and then blends real trajectory data in a principled way to reduce the uncertainty. By explicitly enumerating all possible paths between measured points, we can compute a rich variety of useful statistics, such as visit probabilities and dwell time probabilities.

3 MAXIMUM ENTROPY BRIDGELETS

This section introduces the new concept of maximum entropy bridgelets, which we will refer to as just bridgelets. A bridgelet is a collection of all possible routes between two endpoints. A bridgelet is defined on a lattice graph, which we describe next.

3.1 Lattice Graph Representation

A lattice graph is a graph $G = (V, E)$ whose vertices V are centered on shapes in a grid that tile a part of 2D space. We use a square grid graph, an example of which appears in Figure 1. Each square is represented by a vertex of the graph, and each vertex has undirected edges in E connecting to the vertices of its square's four neighbors, as well as a self-loop back to itself. The edges represent how an entity can move from cell to cell on the graph. A “walk” in this graph is represented by a sequence of vertices $v_0 v_1 ... v_T$, where the edges between the vertices are implied, and it can begin and end at the same vertex. The temporal aspect of the walk is that there is a move along an edge every time step ΔT. Because the graph includes self-loops for each vertex, the walk may include adjacent, repeated vertices for any number of time steps. In our experiments, $\Delta T = 5$ s.

3.2 Bridgelets

A bridgelet is the set of all distinct walks between two vertices in time T. T is an integer number of time steps. Bridgelets are shift-invariant, so they can all be represented by the same starting vertex at integer grid coordinates $(0, 0)$ and time 0. There is a different bridglet for the integer coordinates of each end cell (X, Y) and each T. Each bridglet is thus distinguished by its endpoint and temporal duration, denoted by $W_{X,Y,T}$.

Two walks in a bridgelet are considered distinct if there is any difference in their ordered list of vertices $v_0 v_1 ... v_T$. There are closed

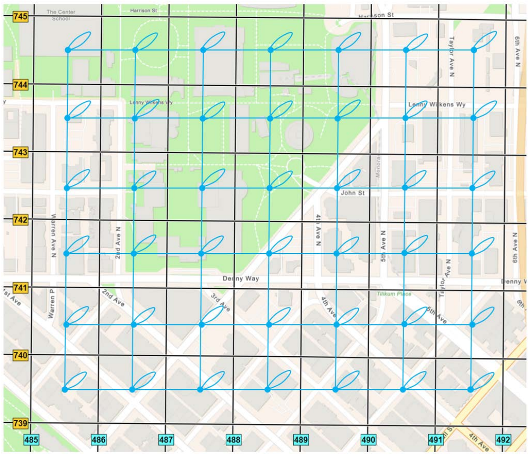


Figure 1: The black grid is part of the standard 100m MGRS grid. The blue graph has nodes at the center of each grid cell and edges for four-connected neighbors and self-loops.

form solutions for the number of "lattice paths" between two vertices in a square grid lattice graph [13], but lattice paths are restricted in various ways, such as disallowed from crossing themselves, no self-loops, and only making moves in the north or east directions. Mihailovs gives a formula (11.1) for the number of walks on a lattice graph like ours, except it does not allow for self-loops [9]. We are not aware of a closed form solution to count the number of walks as we define. We rely on an algorithm, described below, to generate bridgelets.

We are ultimately interested not in the bridgelet's constituent walks, but in the bridgelet's statistics over the visited vertices between the two endpoints. For statistics, we use maximum entropy bridgelets where each constituent, distinct walk in the bridgelet has equal probability. This represents our unbiased uncertainty about the actual path a moving object would choose between two locations. We will show subsequently how we combine bridgelets based on actual trajectory data to build full, probabilistic bridges that reflect the actual paths in the training data.

For practical applications, we would like to know the probability that a walk passes through a certain vertex, denoted as $P(v|W_{X,Y,T})$, where v is the vertex in question and $W_{X,Y,T}$ is the bridgelet. We can compute the visit probability by simply counting, since each constituent walk has equal probability. The total number of walks is $|W_{X,Y,T}|$, and the total number of walks passing through vertex v is $|W_{X,Y,T}|_v$. Thus

$$P(v|W_{X,Y,T}) = \frac{|W_{X,Y,T}|_v}{|W_{X,Y,T}|} \tag{1}$$

Visit probabilities may be important to assess the probability of exposure to something, such as a physical sign, crime, or communicable disease.

Similarly, we can compute the probability of visiting a given vertex v for a given integer dwell time d. The number of walks that meet this condition is $|W_{X,Y,T}|_{v,d}$, so the dwell time probability is

$$P(v,d|W_{X,Y,T}) = \frac{|W_{X,Y,T}|_{v,d}}{|W_{X,Y,T}|} \tag{2}$$

Dwell probabilities may be important to assess the probability of lingering at a place such as a business or park.

By computing all distinct walks in a bridgelet, we can easily compute these and other probabilities, such as the probability of visiting two more more given vertices or the probability of visiting a given vertex multiple times. In our experiments, we concentrate on the simple visit probabilities of Equation 1.

3.3 Example Bridgelet

As an illustration, we present an example bridgelet and some of its probabilities. The example grid is shown in Figure 2. The bridgelet starts at the vertex in the origin cell "a" and ends at the vertex in cell "f" where $(X, Y) = (2, 1)$. We arbitrarily set the duration of the example bridgelet to $T = 4$, meaning it took four steps to get from "a" to "f". This bridgelet would be denoted as $W_{X,Y,T} = W_{2,1,4}$.

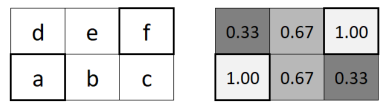


Figure 2: The example bridgelet, $W_{2,1,4}$, starts at the vertex in cell "a" and ends at the vertex in cell "f", taking $T = 4$ time steps. Because the graph is four-connected, only horizontal and vertical moves are allowed, not diagonal. The resulting visit probabilities are shown on the right.

The bridgelet itself is shown in Table 1. This is a list of all the possible walks between the end points in the allotted time. Note that a bridgelet with a length of T consists of T moves along the graph's edges. In this example, there are 12 constituent walks, all with the same start and end. The maximum entropy condition is the same as saying each row in the table has an equal probability of being the actual walk that was chosen by the moving object.

start				end
a	a	b	c	f
a	b	b	c	f
a	b	c	c	f
a	b	c	f	f
a	a	d	e	f
a	d	d	e	f
a	d	e	e	f
a	d	e	f	f
a	a	b	e	f
a	b	b	e	f
a	b	e	e	f
a	b	e	f	f

Table 1: The example bridgelet $W_{2,1,4}$ consists of these 12 possible walks between vertex "a" and "f" on the grid in Figure 2. The walks are grouped by their shape on the grid.

From the list of walks in Table 1, it is simple to count the number of walks that visit each possible cell. These counts give rise to the visit probabilities in Figure 2 and Table 2. These probabilities are computed from Equation 1, where $|W_{X,Y,T}| = |W_{2,1,4}| = 12$. Note the pleasing result that the probability of visiting the start and end vertices is 1.0 regardless of which walk may have been chosen.

vertex	count of walks with vertex	visit probability
a	12	1.00
b	8	0.67
c	4	0.33
d	4	0.33
e	8	0.67
f	12	1.00

Table 2: These are the visit probabilities for the cells in the example bridgelet $W_{2,1,4}$.

The dwell probabilities are shown in Table 3. Each row represents one vertex and one dwell time, and the probabilities are computed from the counting technique in Equation 2. Note that the marginal probabilities work out between the dwell probabilities in Table 3 and the visit probabilities in Table 2. That is

$$P(v|W_{X,Y,T}) = \sum_{d>0} P(v, d|W_{X,Y,T})$$

because $d = 0$ indicates no visit.

vertex	dwell	count of walks with vertex and dwell	dwell probability
a	0	0	0.00
a	1	9	0.75
a	2	3	0.25
b	0	4	0.33
b	1	6	0.50
b	2	2	0.17
c	0	8	0.67
c	1	3	0.25
c	2	1	0.08
d	0	8	0.67
d	1	3	0.25
d	2	1	0.08
e	0	4	0.33
e	1	6	0.50
e	2	2	0.17
f	0	0	0.00
f	1	9	0.75
f	2	3	0.25

Table 3: These are the dwell probabilities for the cells in the example bridgelet $W_{2,1,4}$ from Equation 2.

3.4 Computing Bridgelets

We compute bridgelets of duration T by programmatically generating all possible walks of length T. Given the connectedness of our graph, there are five possible ways to move from a vertex: the four connected neighbors and the self-loop. In a walk of length T, there are T different moves along the edges. Thus it is convenient to index all possible walks of length T by a base-5, T-digit whole number. We can arbitrarily assign move directions to each base-5 digit, where our assignments happen to be $0 \Rightarrow stay$, $1 \Rightarrow east$, $2 \Rightarrow north$, $3 \Rightarrow west$, and $4 \Rightarrow south$. There are then 5^T distinct possible walks of length T.

Since bridgelets are not sensitive to features on the ground nor measured movement patterns, they are shift-invariant in both space and time. We arbitrarily say each bridgelet starts at an origin vertex with integer coordinates $(0, 0)$ and time 0. As we index through all the possible walks of length T, we compute the endpoint v_T of each computed walk and append it to the list of walks in $W_{v_0=0,v_T,T}$, where $v_0 = 0$ indicates the origin vertex. In actual practice, we do not even keep the list of walks, but instead just accumulate the relevant counts so we can, in the end, compute the visit probabilities from Equation 1.

Computing bridgelets this way is relatively slow. We computed bridgelets for $T \in \{1, 2, ..., 15\}$. At $T = 15$, there are $5^{15} \approx 3.05\times10^{10}$ distinct walks. There are likely efficiencies to exploit that we have ignored, such as spatial and temporal symmetries of the bridgelets.

4 FROM BRIDGELETS TO BRIDGES

A single bridgelet describes the possible routes between two locations. If we have a sequence of multiple locations, i.e. a trajectory, we can concatenate bridgelets to form a full bridge along the whole sequence and compute statistics along the whole sequence.

4.1 Bridges

We represent the trajectory as the vertices $v_1v_2...v_N$ with corresponding grid cell coordinates (X_i, Y_i) and integer timestamps T_i, where $T_i > T_{i-1}$ for temporal order. This is illustrated in Figure 3. There is a bridgelet W_i associated with each temporally adjacent pair of vertices. These bridgelets are shifted versions of the zero-based bridglets in Section 3. We introduce a shift operator $E^{X,Y,T}$ that takes a zero-based bridgelet and shifts it such that its starting coordinates are (X, Y) and its starting time is T. Then $W_i = E^{X_i,Y_i,T_i} W_{X_{i+1}-X_i,Y_{i+1}-Y_i,T_{i+1}-T_i}$. The full bridge then becomes $W_1W_2...W_{N-1}$, as shown in Figure 3.

4.2 Bridge Visit Probabilities

We are interested less in the bridge itself than in its statistics. For the purposes of this paper, we are particularly interested in the visit probabilities of the bridge induced by the sequence of points. For a vertex v, we want to compute its visit probability over the sequence of bridgelets, i.e. $P(v|W_1W_2...W_{N-1})$. Each bridgelet W_i gives independent visit probabilities $P(v|W_i)$ as computed with Equation 1. The probability of *not* visiting vertex v on bridgelet W_i is $1 - P(v|W_i)$. Since these probabilities are independent, the probability of never visiting v over the whole bridge is $\prod_{i=1}^{N-1} \left(1 - P(v|W_i)\right)$. Thus the visit probability is

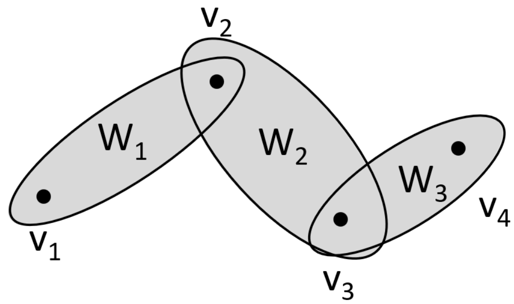


Figure 3: A full bridge on a sequence of points v_i consists of a combination of bridgelets W_i.

$$P(v|W_1W_2...W_{N-1}) = 1 - \prod_{i=1}^{N-1} \left(1 - P(v|W_i)\right) \quad (3)$$

4.3 Example Bridge

To convey the building of a bridge from bridgelets, we present an example, illustrated in Figure 4. Here the six (X_i, Y_i, T_i), $i = 1...6$ trajectory points are $(0, 0, 0)$, $(5, 4, 12)$, $(12, 4, 22)$, $(15, 0, 30)$, $(9, 0, 40)$, $(9, 7, 52)$. The corresponding bridgelets are

- $W_1 = E^{X_1,Y_1,T_1} W_{X_2-X_1,Y_2-Y_1,T_2-T_1} = E^{0,0,0} W_{5,4,12} = W_{5,4,12}$
- $W_2 = E^{X_2,Y_2,T_2} W_{X_3-X_2,Y_3-Y_2,T_3-T_2} = E^{5,4,12} W_{7,0,10}$
- $W_3 = E^{X_3,Y_3,T_3} W_{X_4-X_3,Y_4-Y_3,T_4-T_3} = E^{12,4,22} W_{3,-4,8}$
- $W_4 = E^{X_4,Y_4,T_4} W_{X_5-X_4,Y_5-Y_4,T_5-T_4} = E^{15,0,30} W_{-6,0,10}$
- $W_5 = E^{X_5,Y_5,T_5} W_{X_6-X_5,Y_6-Y_5,T_6-T_5} = E^{9,0,40} W_{0,7,12}$

In Figure 4, the first five images depict the visit probabilities of the five shifted bridgelets, each spanning a pair of temporally adjacent points in the full trajectory. The last image in the figure shows the visit probabilities combined using Equation 3. Note that the visit probabilities of the trajectory's measured points remain 1.0, as expected. This example trajectory crosses itself, and the visit probability at the crossing point is correspondingly high, even though there is no measurement there.

The example in Figure 4 illustrates an advantage of this method over a traditional machine learning approach. The constituent bridges of a trajectory represent all possible paths between the measurement points, even if those paths are never observed in training. This allows us to tolerate minimal, sparsely sampled training data while still maintaining the possibility of physically reachable paths that move to cells not visited in training.

4.4 Combining Bridges

From the training data, we may see several trajectories with the same start cell, end cell, and temporal duration T. Each of these leads to a bridge and associated visit probabilities on the grid, as described above. We combine the visit probabilities of these similar, single-trajectory bridges into an aggregate bridge by taking the mean of the visit probabilities in spatially corresponding cells. This has the effect of representing all the observed behaviors for this set of similar trajectories. Figure 5 shows an example of aggregating the visit probabilities of five bridges (each composed of bridgelets) into an aggregate bridge. Each constituent bridge has the same start and end cell as the others, and these two cells maintain a visit probability of 1.0 in the aggregate bridge.

5 EXPERIMENTS

This section describes our experiments on trajectory data from people, showing how our bridges perform compared to other bridges for inferring in-between locations. Laying out bridges on real mobility data means the combined bridges will represent actual routes that people use, which implicitly accounts for route preferences and physical obstacles.

5.1 Experimental Data

Our trajectory data came from Safegraph, which is a company that aggregates and sells anonymized, personal location data. Each record in their data has a pseudonymous ID indicating the user, a timestamp, and a latitude/longitude measurement. For computing our bridges, we used data from the first week of April, 2022 in the Seattle region described below. After eliminating unsuitable trajectories, we used data from 1,727,763 different users. Since our computed bridgelets do not go beyond $T = 15$, we split trajectories at temporal gaps larger than $15 \times \Delta T = 15 \times 5\,\text{s} = 75\,\text{s}$.

We discretized space with the standard Military Grid Reference System (MGRS) [14]. This grid of square cells covers the earth, with geographic offsets to account for the planet's shape. It offers multiple square cell sizes, and we used $100\,\text{m} \times 100\,\text{m}$. Centered on the downtown of Seattle, Washington USA, our grid subset covered approximately $235\,\text{km}^2$ on an approximately square grid of 154×153 cells. The coverage area is shown in Figure 6, and a close-up of the grid cells is shown in Figure 7.

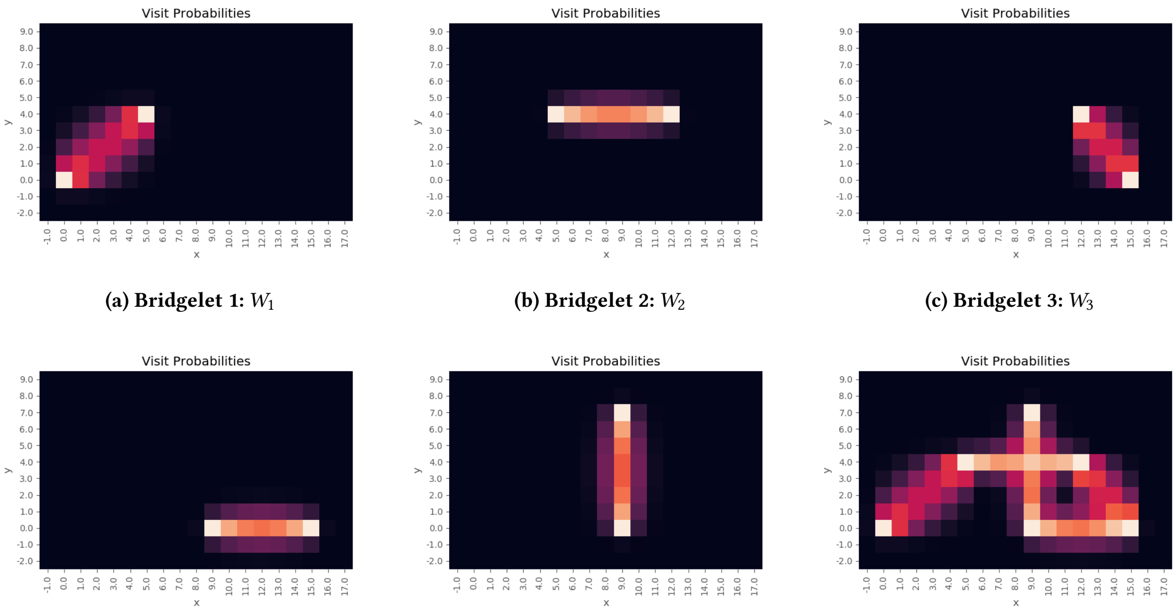
We discretized time into $\Delta T = 5\,\text{s}$ increments. Thus for every continuous time span t in seconds, the discretized, integer time span is $T = \lfloor \frac{t}{5} \rfloor$.

After discretizing the trajectories, each trajectory consists of a time-ordered list of grid cells and discrete time stamps. Temporally adjacent grid cells in a trajectory can be identical. From a discrete training trajectory, we extract all possible temporally adjacent sub-trajectories of at least three cells to enhance our bridge computations. For example, if a trajectory has, in order, grid cells [a,b,c,d], we trained with the original trajectory along with [a,b,c] and [b,c,d]. We do not train with trajectories of only one or two cells. A two-cell bridge is just a bridgelet. Using subtrajectories, we trained our model with 79,452,051 total trajectories.

5.2 Bead Baselines

We test our proposed bridges against two bead baselines. Our basic bead is a bridge that posits a possible visit to every cell that is accessible in time T, starting at point (X_1, Y_1) and ending at (X_N, Y_N). This is the same definition of the endpoints of our bridge that we defined in Section 4.1. The resulting set of accessible grid cells is approximated by an ellipse, and we assign a visit probability of 1.0 to each cell inside the ellipse.

The other bead, the "learned bead", gives a visit probability of 1.0 to every nonzero point in our proposed bridges. That is, it takes the visit probabilities from our proposed bridges and assigns a visit probability of 1.0 wherever the learned visit probability is non-zero.



(a) **Bridgelet 1:** W_1 (b) **Bridgelet 2:** W_2 (c) **Bridgelet 3:** W_3

(d) **Bridgelet 4:** W_4 (e) **Bridgelet 5:** W_5 (f) **Bridge from five bridgelets**

Figure 4: The visit probabilities from five bridgelets combined into a single bridge. The endpoints of the bridgelets represent sampled points of a trajectory. The bridgelets were combined into the full bridge with Equation 3. The white cells at the bridgelets' endpoints have a visit probability of 1.0.

This makes a more realistic bead, because it is informed by the training data.

5.3 Suitability of Continuous PDF Bridges as Baselines

Two obvious candidates for computing visit probabilities are the Brownian bridge [5] and Gaussian process [10]. Given two points in time and two locations, each of these gives a bridge as a continuous probability distribution as a function of time, i.e. $\big(x(t), y(t)\big) = \boldsymbol{x}(t) \sim N\big(\boldsymbol{\mu}(t), \Sigma(t)\big)$, where $\boldsymbol{x}(t)$ is the time-varying two-dimensional location whose probability distribution is normally distributed with time-varying mean and covariance.

We define visit probabilities on a discrete grid, and we can use the distribution for $\boldsymbol{x}(t)$ to compute the probability of being inside a grid cell whose corners at $(x_{\min}, y_{\min})$ and $(x_{\max}, y_{\max})$ as

$$p(t) = \int_{y_{\min}}^{y_{\max}} \int_{x_{\min}}^{x_{\max}} N\big(\boldsymbol{\mu}(t), \Sigma(t)\big) dx dy$$

This integral results in a time-varying erf() function giving the probability $p(t) \in [0, 1]$ of being inside the cell as a function of time, illustrated in [8] for the Brownian bridge. However, there is not a clear way to go from $p(t)$ to a scalar visit probability over a discrete range of time. Thus the Brownian bridge and Gaussian process are not good candidates for baseline comparisons for our proposed bridgelets. It may be possible to use simulation to compute the probabilities, but we did not pursue that approach.

5.4 Error Metric

Our error metric for each test trajectory is the mean absolute error between the predicted visited probabilities of each cell and the test trajectory, where the visited cells of the test trajectory take a value of 1.0. In equation form, the bridge assigns a visit probability p_j to each grid cell j. Each test trajectory consists of a set V of visited cells. The error is

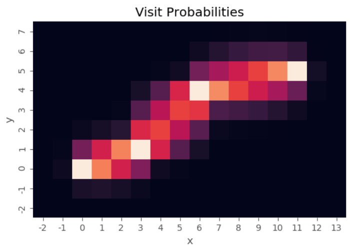
$$e = \sum_{j \in V} (1.0 - p_j) + \sum_{j \notin V} (p_j - 0.0) \tag{4}$$

This essentially says that the visit probability p_j should be 1.0 at the trajectory's visited cells and 0.0 elsewhere. Note that Appendix A explains why the visit probability of the bead bridges should be $p = 1.0$.

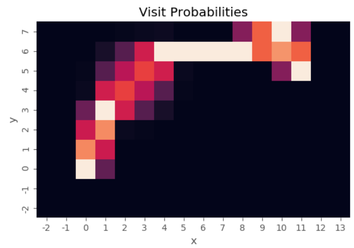
5.5 Results

We tested with trajectories from the day after our week of training data. There were 234,731 such trajectories available. However, only 75,144 had corresponding bridges from our one week of training data. Recall that a bridge is parameterized by its starting cell, ending cell, and discrete traversal time. That the training data accounted for only about 32% of the test data shows that a longer period of training is ultimately necessary for better coverage.

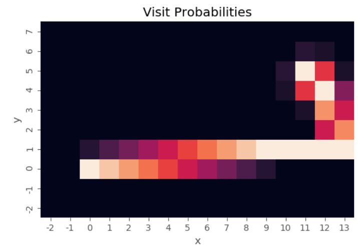
Our test results are shown in Figure 8. The bars show the median error metric (Equation 4) over all the test trajajectories for the proposed method and the two bead baselines. With a median visit probability error of 2.62, the proposed method clearly outperforms



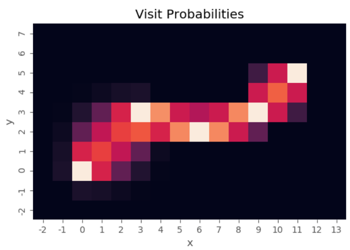
(a) Bridge from trajectory 1



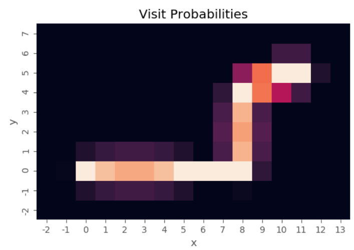
(b) Bridge from trajectory 2



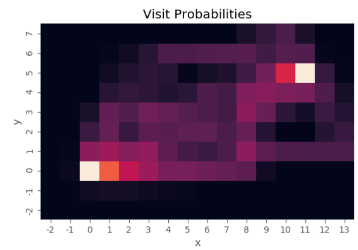
(c) Bridge from trajectory 3



(d) Bridge from trajectory 4



(e) Bridge from trajectory 5



(f) Aggregate mean bridge

Figure 5: Visit probabilities of bridges 1-5 were constructed from sampled trajectories (white points) and bridgelets spanning between the trajectory points. The aggregate visit probabilities in the lower right are the mean of bridges 1-5.

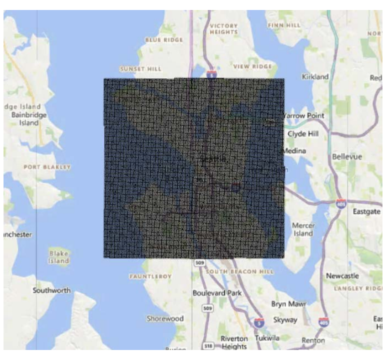


Figure 6: Trajectory data came from this area surrounding the core of Seattle, Washington USA.

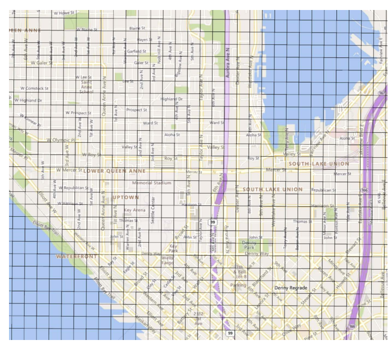


Figure 7: This close-up shows the size of the $100\,\mathrm{m} \times 100\,\mathrm{m}$ cells of the Military Grid Reference System (MGRS)

the learned beads (median 40) and elliptical beads (mean 63). We attribute this to the fact that the proposed bridges are based on training data and that the visit probabilities are represented by continuous probability values.

6 CONCLUSION

Because mobility data is often sparsely sampled, it is important to reason about location behavior between measurements. This paper proposes a new approach based on maximum entropy bridgelets to compute probability distributions of likely visit locations between measurements on a discrete spatial grid. The bridgelets represent all possible walks between two grid cells for a given traversal time. The maximum entropy constraint gives equal probability to all the walks, properly representing unbiased uncertainty by making no prior assumptions about which are most likely. Applied to a measured trajectory, these bridgelets are strung together over the measurement gaps. For computing visit probabilities and dwell probabilities, we derive the formulae for combining the probabilities of the individual bridgelets. Compared to traditional beads, our experiments show that our bridgelet based approach is much more accurate at estimating visit probabilities.

Future opportunities for extending this approach include computational efficiency. Computing the bridgelets is intensive, and there are likely spatial symmetries to exploit that we have not adequately explored. Likewise, training is computationally intensive,

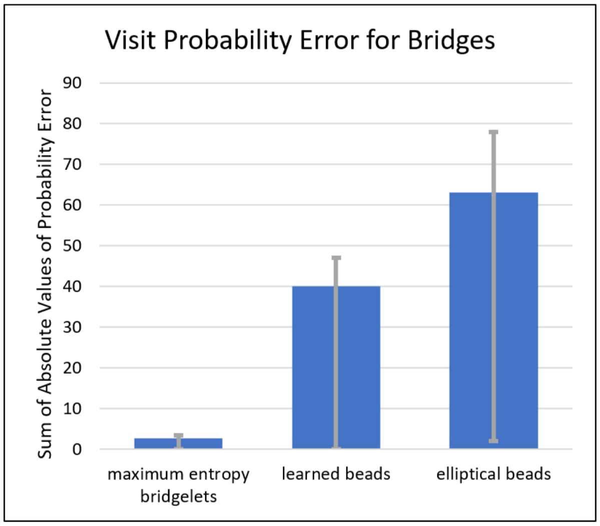


Figure 8: These bar represent the medians of the sums of the absolute visit probability errors for three different types of bridges. The error bars extend from the first to the third error quartile.

and there are efficiencies to discover to make it faster. It would also be useful to accommodate bridges that change over time as well as uncertainty in the location measurements.

REFERENCES

[1] John Amanatides, Andrew Woo, et al. 1987. A fast voxel traversal algorithm for ray tracing.. In *Eurographics*, Vol. 87. 3–10.

[2] Tobias Emrich, Hans-Peter Kriegel, Nikos Mamoulis, Matthias Renz, and Andreas Zufle. 2012. Querying uncertain spatio-temporal data. In *2012 IEEE 28th international conference on data engineering*. IEEE, 354–365.

[3] Pip Forer. 1998. Geometric Approaches to the Nexus of Time, Space, and Microprocess: Implementing. *Spatial and temporal reasoning in geographic information systems* (1998), 171.

[4] Claude Godreche, Satya N Majumdar, and Grégory Schehr. 2015. Record statistics for random walk bridges. *Journal of Statistical Mechanics: Theory and Experiment* 2015, 7 (2015), P07026.

[5] Jon S Horne, Edward O Garton, Stephen M Krone, and Jesse S Lewis. 2007. Analyzing animal movements using Brownian bridges. *Ecology* 88, 9 (2007), 2354–2363.

[6] Kathleen Hornsby and Max J Egenhofer. 2002. Modeling moving objects over multiple granularities. *Annals of Mathematics and Artificial Intelligence* 36, 1 (2002), 177–194.

[7] John Krumm. 2021. Brownian Bridge Interpolation for Human Mobility?. In *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*. 175–183.

[8] John Krumm. 2022. The Brownian Bridge for Space–Time Interpolation. In *Spatial Gems, Volume 1*, John Krumm, Andreas Züfle, and Cyrus Shahabi (Eds.). Vol. 1. Association for Computing Machinery (ACM), Chapter 9, 73–82.

[9] Aleksandrs Mihailovs. 1998. Enumeration of walks on lattices. I. *arXiv preprint math/9803128* (1998).

[10] Kien Nguyen, John Krumm, and Cyrus Shahabi. 2021. Gaussian Process for Trajectories. *arXiv preprint arXiv:2110.03712* (2021).

[11] Kien Nguyen, John Krumm, and Cyrus Shahabi. 2021. Quantifying Intrinsic Value of Information of Trajectories. In *proceedings of the 29th International Conference on Advances in Geographic Information Systems*. 81–90.

[12] Johannes Niedermayer, Andreas Züfle, Tobias Emrich, Matthias Renz, Nikos Mamouliso, Lei Chen, and Hans-Peter Kriegel. 2013. Probabilistic Nearest Neighbor Queries on Uncertain Moving Object Trajectories. *Proceedings of the VLDB Endowment* 7, 3 (2013).

[13] Wikipedia contributors. 2019. Lattice path — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Lattice_path&oldid=916383786 [Online; accessed 17-May-2022].

[14] Wikipedia contributors. 2022. Military Grid Reference System — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Military_Grid_Reference_System&oldid=1088192371 [Online; accessed 13-June-2022].

A BEAD BRIDGES SHOULD HAVE VISIT PROBABILITY 1.0

The two bead baselines consist of bridges with a set of visit probabilities that are all $p_j = p = 1.0$ everywhere a visit could occur (basic bead) or has a non-zero probability in our proposed bridges (learned bead). A natural question is whether p should be some other value in $[0, 1]$ to minimize the error metric for a more fair comparison to our proposed bridges. We will show next that $p = 1.0$ is the only reasonable value.

set name	set	trajectory value	bridge probability	error per cell
true positives	$V \cap B$	1	p	1-p
false negatives	$V \cap B'$	1	0	1
false positives	$V' \cap B$	0	p	p
true negatives	$V' \cap B'$	0	0	0

Table 4: These are the types of errors for the bead bridges.

The cells visited by the test trajectory are in a set V, and these are given a value of 1.0 in the error metric. The bridge cells are in a set B. The cells *not* in the trajectory and bridge are V' and B', respectively. The sets making up the error metric and their associated per-cell errors are shown in Table 4.

The error metric for the bead bridges is then

$$\begin{aligned} e_{bead} &= \sum_{j \in V \cup B} (1-p) + \sum_{j \in V \cup B'} 1 + \sum_{j \in V' \cup B} p \\ &= |V \cup B|(1-p) + |V \cup B'| + |V' \cup B|p \\ &= (|V' \cup B| - |V \cup B|)p + |V \cup B| + |V \cup B'| \\ &= (|V' \cup B| - |V \cup B|)p + |V| \end{aligned} \tag{5}$$

From Equation 5, the error metric varies linearly with the bridge's visit probability p with a slope of $|V' \cup B| - |V \cup B|$. If $|V' \cup B| < |V \cup B|$, then the slope is negative, so p should take on its maximum value of 1 to minimize the error. If $|V' \cup B| > |V \cup B|$, then the slope is positive, so p should take on its minimum value of 0 to minimize the error. However, $p = 0$ effectively deletes the bridge and gives no information, so $p = 1$ is the only reasonable choice for the bead bridges.