PROOF COVER SHEET

Author(s):	Quanzeng You & John Krumm
Article title:	Transit tomography using probabilistic time geography: planning routes without a road map
Article no:	TLBS 963180
Enclosures:	1) Query sheet 2) Article proofs

Dear Author,

1. Please check these proofs carefully. It is the responsibility of the corresponding author to check these and approve or amend them. A second proof is not normally provided. Taylor & Francis cannot be held responsible for uncorrected errors, even if introduced during the production process. Once your corrections have been added to the article, it will be considered ready for publication.

Please limit changes at this stage to the correction of errors. You should not make trivial changes, improve prose style, add new material, or delete existing material at this stage. You may be charged if your corrections are excessive (we would not expect corrections to exceed 30 changes).

For detailed guidance on how to check your proofs, please paste this address into a new browser window: http://journalauthors.tandf.co.uk/production/checkingproofs.asp

Your PDF proof file has been enabled so that you can comment on the proof directly using Adobe Acrobat. If you wish to do this, please save the file to your hard disk first. For further information on marking corrections using Acrobat, please paste this address into a new browser window: http://journalauthors.tandf.co.uk/production/acrobat.asp

2. Please review the table of contributors below and confirm that the first and last names are structured correctly and that the authors are listed in the correct order of contribution. This check is to ensure that your name will appear correctly online and when the article is indexed.

Sequence	Prefix	Given name(s)	Surname	Suffix
1		Quanzeng	You	
2		John	Krumm	

Queries are marked in the margins of the proofs, and you can also click the hyperlinks below.

AUTHOR QUERIES

General points:

1. **Permissions**: You have warranted that you have secured the necessary written permission from the appropriate copyright owner for the reproduction of any text, illustration, or other material in your article. Please see http://journalauthors.tandf.co.uk/permissions/usingThirdPartyMaterial.asp.
2. **Third-party content**: If there is third-party content in your article, please check that the rightsholder details for re-use are shown correctly.
3. **Affiliation**: The corresponding author is responsible for ensuring that address and email details are correct for all the co-authors. Affiliations given in the article should be the affiliation at the time the research was conducted. Please see http://journalauthors.tandf.co.uk/preparation/writing.asp.
4. **Funding**: Was your research for this article funded by a funding agency? If so, please insert 'This work was supported by <insert the name of the funding agency in full>', followed by the grant number in square brackets '[grant number xxxx]'.
5. **Supplemental data and underlying research materials**: Do you wish to include the location of the underlying research materials (e.g. data, samples or models) for your article? If so, please insert this sentence before the reference section: 'The underlying research materials for this article can be accessed at <full link>/ description of location [author to complete]'. If your article includes supplemental data, the link will also be provided in this paragraph. See <http://journalauthors.tandf.co.uk/preparation/multimedia.asp> for further explanation of supplemental data and underlying research materials.
6. The **CrossRef database** (www.**crossref**.org/) has been used to validate the references. Changes resulting from mismatches are tracked in red font.

AQ1 Please provide the missing publisher location for reference 'Brush et al. (2010)'.

AQ2 Please provide the missing publisher location for reference 'Davics et al. (2006)'.

AQ3 Please provide the missing publisher name/publisher location for reference 'Dias (2004)'.

AQ4 Please provide the missing publisher location for reference 'Edelkamp and Schrödl (2003)'.

AQ5 Please provide the missing publisher name/publisher location for reference 'Graham and Stephens (2012)'.

AQ6 Please provide the missing publisher location for reference ‘Haklay and Weber (2008)’.

AQ7 Please provide the missing publisher name/publisher location for reference ‘Laurila et al. (2012)’.

AQ8 Please provide the missing publisher location for reference ‘Lin et al. (2010)’.

AQ9 Please provide the missing publisher name/publisher location for reference ‘Markoff (2008)’.

AQ10 Please provide the missing publisher location for reference ‘Paek et al. (2010)’.

AQ11 Please provide the missing publisher location for reference ‘Pajor (2009)’.

AQ12 Please provide the missing publisher name/publisher location for reference ‘Rios (2013)’.

AQ13 Please provide the missing publisher location for reference ‘Shen et al. (2013).’

AQ14 Please provide the missing publisher location for reference ‘Szalay et al. (2005)’.

How to make corrections to your proofs using Adobe Acrobat/Reader

Taylor & Francis offers you a choice of options to help you make corrections to your proofs.

Your PDF proof file has been enabled so that you can edit the proof directly using Adobe Acrobat/Reader. This is the simplest and best way for you to ensure that your corrections will be incorporated. If you wish to do this, please follow these instructions:

1. Save the file to your hard disk.

2. Check which version of Adobe Acrobat/Reader you have on your computer. You can do this by clicking on the "Help" tab, and then "About."

If Adobe Reader is not installed, you can get the latest version free from http://get.adobe.com/reader/.

3. If you have Adobe Acrobat/Reader 10 or a later version, click on the "Comment" link at the right-hand side to view the Comments pane.

4. You can then select any text and mark it up for deletion or replacement, or insert new text as needed. Please note that these will clearly be displayed in the Comments pane and secondary annotation is not needed to draw attention to your corrections. If you need to include new sections of text, it is also possible to add a comment to the proofs. To do this, use the Sticky Note tool in the task bar. Please also see our FAQs here: http://journalauthors.tandf.co.uk/production/index.asp.

5. Make sure that you save the file when you close the document before uploading it to CATS using the "Upload File" button on the online correction form. If you have more than one file, please zip them together and then upload the zip file.

If you prefer, you can make your corrections using the CATS online correction form.

Troubleshooting

Acrobat help: http://helpx.adobe.com/acrobat.html
Reader help: http://helpx.adobe.com/reader.html

Please note that full user guides for earlier versions of these programs are available from the Adobe Help pages by clicking on the link "Previous versions" under the "Help and tutorials" heading from the relevant link above. Commenting functionality is available from Adobe Reader 8.0 onwards and from Adobe Acrobat 7.0 onwards.

Firefox users: Firefox's inbuilt PDF Viewer is set to the default; please see the following for instructions on how to use this and download the PDF to your hard drive: http://support.mozilla.org/en-US/kb/view-pdf-files-firefox-without-downloading-them#w_using-a-pdf-reader-plugin

Journal of Location Based Services, 2014
Vol. 00, No. 0, 1–18, http://dx.doi.org/10.1080/17489725.2014.963180

Transit tomography using probabilistic time geography: planning routes without a road map

Quanzeng You[a] and John Krumm[b]*

[a]*Department of Computer Science, University of Rochester, Rochester, NY, USA;* [b]*Microsoft Research, Microsoft Corporation, Redmond, WA, USA*

(*Received 29 August 2014; accepted 2 September 2014*)

Vehicle routing usually depends on a road map, and road maps are expensive to create and maintain. While crowdsourcing road maps from logged GPS data has proven effective, the limited availability of GPS data limits their coverage area. To overcome this limitation, we show how to location data from geotagged tweets, which cover much of the world, to compute routes directly without making a road map. We compensate for the wide spacing of tweets' latitude/longitude points by using probabilistic time geography, which explicitly models the uncertain location of someone traveling between measured locations. In our formulation, each pair of temporally adjacent tweets contributes an estimate of the driving time along hypothesised roads in a regular grid. We show how to compute these estimates as expected values based on probabilistic Brownian bridges. We can compute routes on this regular grid using traditional A* search. Our experiments demonstrate that our computed routes match well with routes computed on the actual road network using a commercial router. Furthermore, we show that our computed routes vary sensibly with changes in traffic between rush hour and weekends. We also apply the same technique to compute reasonable airplane routes.

Keywords: road maps; vehicle routing; probabilistic time geography; Brownian bridge; geotagged Twitter

1. Introduction

People often face the problem of creating a travel plan between two places. This can be a simple drive in a vehicle or a more complex plan that involves multiple modes of transportation, including a personal vehicle, walking, public transportation and flights. While such multi-modal route planners are available (Pajor 2009), a more attractive alternative is to take advantage of what other people have done. This is because people optimise for many different criteria when planning a trip (Ben-Akiva et al. 1984), and an automatic planner may not be able to represent the richness of a plan's positives and negatives, such as scenery and safety. Even if there is no record of a person making precisely the desired trip, it may be possible to piece together parts of other trips from others into a complete plan. Because it is based on previously recorded trips, the plan

*Corresponding author. Email: jckrumm@microsoft.com would naturally incorporate collective preferences that may be hard to capture in a traditional trip planner.

One barrier to creating a collective trip planner is the lack of a large, publicly available collection of recorded trips. There are focused sets of GPS logs available, but these are usually limited to a small area (e.g. Brush et al. 2010; Laurila et al. 2012). GPS loggers tend to exhaust the battery of a cell phone in around 6 h (Lin et al. 2010; Paek et al. 2010), which makes it difficult to collect continuous location data from regular people. Taxis and package delivery companies often collect GPS data, but these data are limited to a single mode of transportation, and it is sometimes expensive.

Our solution is to use location updates from social media such as Twitter. Advantages of these data are its wide coverage and availability. Twitter data are clearly valuable for understanding geography, as shown by the stunning heat maps made from geotagged tweets described in a Twitter corporate blog (Rios 2013), but the locations from each individual are usually widely spaced in time. This makes it difficult to infer the exact path. However, Figure 1 shows that simply connecting the dots of the geotagged tweets of many users gives a rough picture of a region's major roads. Geotagged tweets are continually refreshed, so they reflect new changes in feasible routes.

The wide spacing of these location measurements is the main challenge of using tweets to make travel plans. To manage the wide spacing in a principled way, we use probabilistic time geography, which gives a reasonable guess, and associated uncertainty, of the user's location between tweets. We refer to our technique as 'transit tomography', because tomographic reconstruction makes estimates of an interior (travel routes) from widely spaced penetrations (tweet sequences).

While our ultimate goal is to construct multi-modal trip plans, this paper has the more modest goal of computing driving routes. We do this without the benefit of a road map in order to show that we can suggest routes based only on observations of collective behaviours. Inferring driving routes has the advantage of easy ground truth comparisons in the form of regular route engines. Extrapolating to multi-modal routes, we would hope to avoid the gathering and maintenance of transportation schedules and routes, instead using the collective behaviours of what people actually do.

Our technique starts with a regular grid of candidate road segments, placed indiscriminately on the map. We then look at pairs of sequential location measurements from Twitter to estimate the traversal time of each candidate road segment. Applying simple A* search on these candidate road segments gives the route, which we compare to routes from a commercial routing program.

Besides Twitter, our techniques are appropriate for any type of widely spaced geotagged data, such as from the Chinese microblogging site Sina Weibo (http://weibo.com), geotagged photos from Flickr (http://www.flickr.com/), and location check-ins from Foursquare (https://foursquare.com/), Facebook (https://www.facebook.com/) and Google+ (https://plus.google.com/). We describe our input data from Twitter in the next section. Following that, we define our set of candidate road segments, describe how we use probabilistic time geography with tweets, evaluate our results by computing routes and conclude with a description of previous work.

2. Twitter data

Even though less than 1% of tweets are geotagged (Graham and Stephens 2012), there are so many tweets that the geotagged ones form a dense map (Rios 2013). Crucially for us,

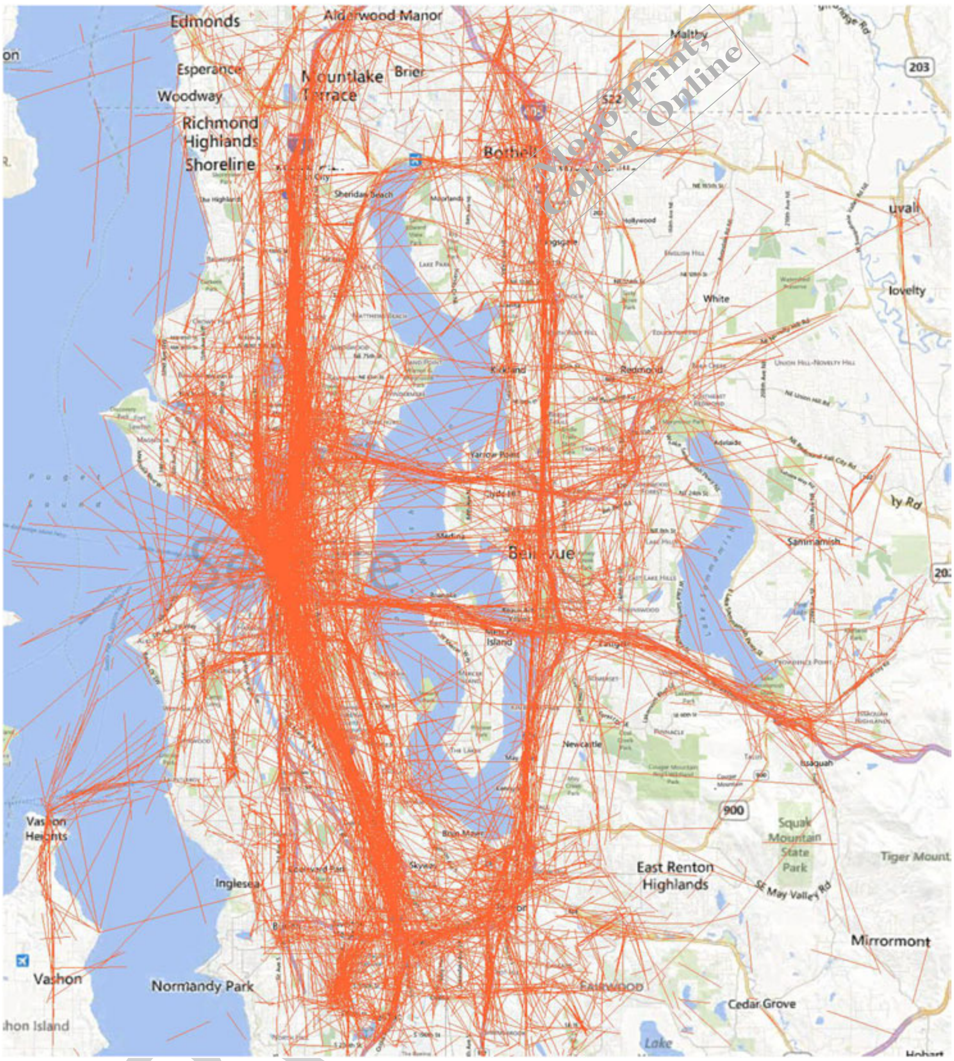

Figure 1. Drawing lines between geotagged tweets around Seattle, WA gives a rough map of the major roads.

tweets also come with a user ID and a timestamp. This means we can make a rough trajectory of each user by sorting their tweets by their post time and then connecting the corresponding latitude/longitude points with straight line segments in sequence. Our fundamental unit of data is a measurement pair, which is two geotagged tweets from the same person that are temporally adjacent. Since we are looking for pairs that correspond to roads, we keep only those pairs that are greater than 100 m apart, which helps to ensure that the difference in endpoints is due to more than just measurement noise. This value is fairly arbitrary, and we did not experiment with other values. We also keep only pairs whose speed is less than 80 miles per hour, where the speed is computed from the locations and timestamps of the pair's endpoints. This helps bias the tweet pairs to favour those from moving vehicles. Finally, we use only tweet pairs that were no more than 24 h apart.

After filtering in this way, we retain 36.5% of the original pairs that we extracted. Of course, not all the measurement pairs represent pure motion. We know that someone might linger at a location before or after tweeting, making their apparent speed slower than their actual speed. Their route between the endpoints is likely not a straight line. We designed our techniques to be robust to these conditions. The measurement pairs we used were tweets recorded in 2012 from the area around Seattle, WA, USA shown in Figure 1. This covers an area of about 1781 km^2.

3. Candidate roads

A map of tweet pairs, such as Figure 1, shows that they generally follow the major roads, especially after we filter to keep pairs representing vehicle speeds. One way to infer routes would be to link these pairs together to generate roads. However, this would make it difficult to account for the uncertainty in the subject's location between pairs of tweets. Instead of generating roads, we hypothesised a dense network of candidate roads made up of relatively short segments. Then we computed how much support each candidate segment had from the surrounding tweet pairs. We will explain this computation in the next section.

The candidate roads are a simple grid made of nearly equal-sized triangles that tessellate the earth. This is the hierarchical triangular mesh (HTM) (Szalay et al. 2005). The HTM tessellates the sphere hierarchically by first dividing the north and south hemispheres into four, equal size, spherical triangles each. In each hemisphere, one corner of each triangle is at the pole, and their bases are on the equator. To generate a higher resolution grid, each triangle is tiled into four smaller triangles by connecting the mid-points of the edges of the original triangle. We used the grid shown in Figure 2, where the

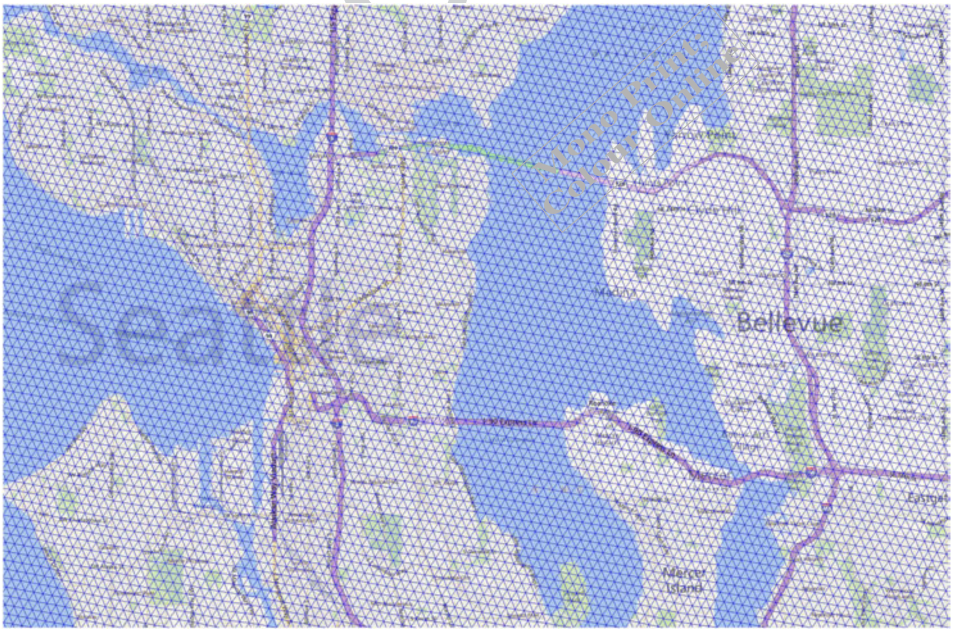

Figure 2. These are candidate roads from the HTM. Each edge is about 245 m long.

length of each edge is approximately 245 m. In the area we examined, shown in Figure 1, there are 132,884 triangular edges and 44,590 vertices. Each of these edges is a candidate road. In fact, each edge is actually two candidate roads, because we assume the edges are directional, and we treat each direction independently. In addition, we assume that edges that touch at a common endpoint are connected, meaning that a vehicle could move from the end of one edge to the beginning of any connected edge. After we set the default travel time of each candidate road to infinity, we use probabilistic time geography to compute expected travel times for candidate roads that are near tweet pairs.

4. Probabilistic time geography

A pair of location measurements taken at different times gives a clue to a person's location between the measurements, but with uncertainty. We cannot infer the actual route by simply connecting the measurements with a straight line segment. Indeed, doing this with tweets shows some paths that slash across the map unrealistically, as seen in Figure 1.

Probabilistic time geography is a way of extrapolating or interpolating a location in time from one or more time-stamped location measurements. It uses probability to represent the inherent uncertainty in an entity's location at times beyond or between the measurements. The models generally assume some type of random motion. The paper by Winter and Yin (2010) gives an introduction and recent research for probabilistic time geography where only the starting point is known, leading to an extrapolation in time. In cases where we have a pair of measurements, like ours, probabilistic time geography interpolates between the points (Winter and Yin 2010). In all cases, the resulting probability density function can be written as $p(\mathbf{x},t)$, where $\mathbf{x} = (x,y)$ are spatial coordinates of the moving person and t represents time.

A simple form of probabilistic time geography is the Brownian bridge, which assumes Brownian motion between points $(x,y,t) = (x_a,y_a,0)$ and $(x,y,t) = (x_b,y_b,T)$. This means the entity was observed at location $\mathbf{x}_a = (x_a,y_a)$ at time 0 and then again at location $\mathbf{x}_b = (x_b,y_b)$ at time T. This model is explained in Horne et al. (2007), where it is applied to animal tracking. Expressed as a probability density function, the Brownian bridge is

$$p_{\mathbf{x}_a,\mathbf{x}_b,T}(\mathbf{x},t) = \begin{cases} \frac{1}{2\pi\sigma^2(t)T}\exp\left[-\frac{(x-\mu_x(t))^2+(y-\mu_y(t))^2}{2\sigma^2(t)}\right] & \text{if } 0 < t < T \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

where

$$\mu_x(t) = x_a + \frac{t}{T}(x_b - x_a),$$

$$\mu_y(t) = y_a + \frac{t}{T}(y_b - y_a),$$

$$\sigma^2(t) = \frac{t(T-t)}{T}\sigma_{\mathrm{m}}^2.$$

For every $t \in [0, T]$, this is a Gaussian density function with mean location $(\mu_x(t), \mu_y(t))$. The mean location is a linear interpolation from (x_a, y_a) at time 0 to (x_b, y_b) at time T with constant speed. The variance of the Gaussian, $\sigma^2(t)$, is zero at times 0 and T, rising to a maximum of $\sigma_{\mathrm{m}}^2 T/4$ at time $T/2$, and staying non-negative over $[0, T]$. The diffusion coefficient, σ_{m}^2, controls the spread of the density function, and can be estimated from data. Figure 3 shows an example of Brownian bridge over (x, y) with the time variable integrated out, i.e. $p_{\mathbf{x}_a,\mathbf{x}_b}(\mathbf{x}) = \int_0^T p_{\mathbf{x}_a,\mathbf{x}_b,T}(\mathbf{x}, t)\mathrm{d}t$. The Brownian bridge represents our high confidence in the person's location at the measured endpoints, but also represents our uncertainty in the middle. It is intuitively appealing because the uncertainty in the middle of the trajectory is larger when the points are farther apart in time, representing the possibility that the person has used the time to roam off the straight-line path. While normal human travel may not conform to Brownian motion, the model is a general way to express the uncertainty in location between measurements, especially since we assume we have no knowledge of the road network.

5. Road integrals

At this point, we have a grid of candidate roads and a group of Brownian bridges from measured location pairs. Based on the data, we want to investigate each candidate road. We start by computing a weight that indicates how much support the candidate road has from the Brownian bridges of nearby location measurement pairs.

Each candidate road slices through the Brownian bridges, as illustrated in the example in Figure 4. This shows a Brownian bridge from a pair of location measurements along with a nearby candidate road. Since the road is near the Brownian bridge, we believe the person might have used the road to travel between the two measured locations. A candidate road that is positioned near many, high-density Brownian bridges is more likely to be an actual road. We can quantify this by computing the probability that the person responsible for the Brownian bridge moved from one end of the candidate road segment to the other over some time period between the two location measurements. As an example computation, we can compute the probability that at any time the person was in a

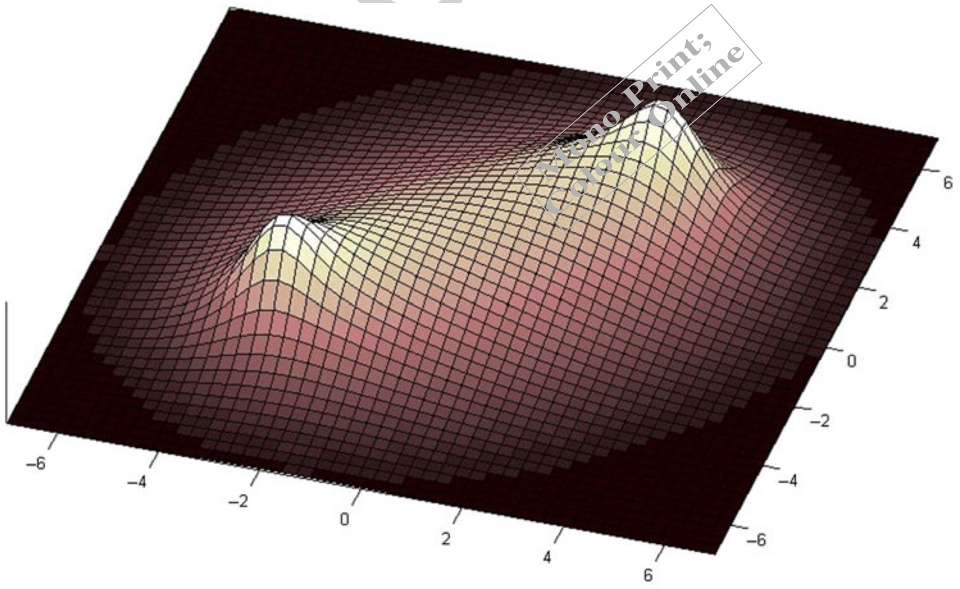

Figure 3. This is a Brownian bridge density function with time integrated out. The two (*x*, *y*) endpoints are at the two peaks: (−3, −3) and (3, 3).

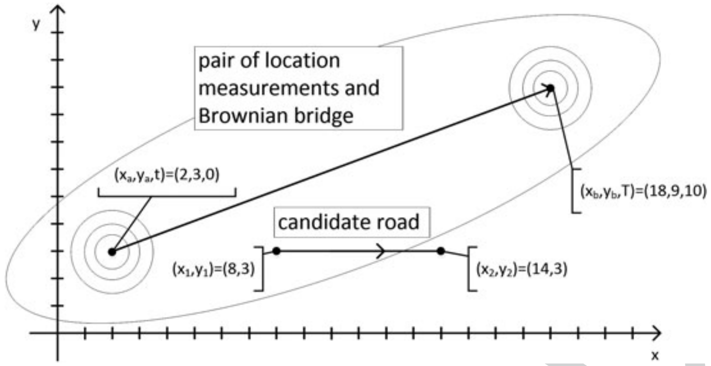
y
x
pair of location measurements and Brownian bridge
(x_a,y_a,t)=(2,3,0)
(x_b,y_b,T)=(18,9,10)
candidate road
(x₁,y₁)=(8,3)
(x₂,y₂)=(14,3)

Figure 4. This is an example of a candidate road segment and a nearby Brownian bridge.

small 2D region A_1 around the start of the road segment, which is at $\mathbf{x}_1 = (x_1, y_1)$:

$$p_1 = \int_{x_1 \in A_1} \int_0^T p_{\mathbf{x}_a,\mathbf{x}_b,T}\left(x_1, y_1, t\right) \mathrm{d}t \, \mathrm{d}\mathbf{x}_1. \tag{2}$$

This example computation integrates over all time $t \in [0, T]$ between the two measurements, and it integrates over the small region A_1 to give a scalar probability based on the Brownian bridge.

Moving beyond this example, we are actually interested in computing the probability that the person was at the start of the candidate road segment at $\mathbf{x}_1 = (x_1, y_1)$ at some time $t_1 \in [0, T]$ and then at the end of the road segment at $\mathbf{x}_2 = (x_2, y_2)$ at some time later $t_2 \in [t_1, T]$. The probability density function for this event is

$$\begin{aligned} p_{\mathbf{x}_a,\mathbf{x}_b,T}(\mathbf{x}_1, t_1, \mathbf{x}_2, t_2) &= p_{\mathbf{x}_a,\mathbf{x}_b,T}(\mathbf{x}_1, t_1) p_{\mathbf{x}_a,\mathbf{x}_b,T}(\mathbf{x}_2, t_2 | x_1, t_1) \\ &= p_{\mathbf{x}_a,\mathbf{x}_b,T}(\mathbf{x}_1, t_1) p_{\mathbf{x}_1,\mathbf{x}_b,T-t_1}(\mathbf{x}_2, t_2 - t_1). \end{aligned} \tag{3}$$

The second multiplicand of Equation (3) is a Brownian bridge that starts at $\mathbf{x}_1$ and ends at $\mathbf{x}_b$, with a duration of $T - t_1$. Recall that $\mathbf{x}_a$ and $\mathbf{x}_b$ are the ends of the Brownian bridge, and $\mathbf{x}_1$ and $\mathbf{x}_2$ are the ends of the candidate road segment. We show in the appendix that Equation (3) is a proper probability density function.

To compute the numerical probability of this traversal from $\mathbf{x}_1$ to $\mathbf{x}_2$, we look at the case where the person starts in a small region A_1 centred at $\mathbf{x}_1$ and later arrives in a small region A_2 centred at $\mathbf{x}_2$. This probability is

$$p_{12} = \int_{x_1 \in A_1} \int_0^T \int_{x_2 \in A_2} \int_{t_1}^T p_{\mathbf{x}_a,\mathbf{x}_b,T}\left(\mathbf{x}_1, t_1, \mathbf{x}_2, t_2\right) \mathrm{d}t_2 \, \mathrm{d}\mathbf{x}_2 \, \mathrm{d}t_1 \, \mathrm{d}\mathbf{x}_1. \tag{4}$$

We begin simplifying this integral by assuming that the Brownian bridges are nearly constant in regions A_1 and A_2, giving

$$p_{12} \approx |A_1||A_2| w_{12}, \tag{5}$$

where $|A_i|$ is the area of A_i and

$$w_{12} = \int_0^T p_{\mathbf{x}_a,\mathbf{x}_b,T}\left(\mathbf{x}_1,t\right)\int_{t_1}^T p_{\mathbf{x}_1,\mathbf{x}_b,T-t_1}\left(x_2,y_2,t_2-t_1\right)\mathrm{d}t_2\,\mathrm{d}t_1. \tag{6}$$

Using the numbers in the example from Figure 4, and $\sigma_\mathrm{m}^2 = 100$, numerically integrating Equation (6) in MATLAB® gives $w_{12} = 1.8573 \times 10^{-6}$.

Since our ultimate goal is to compute routes by minimising travel time, we can use integrals of these types to compute the expected travel time from region A_1 at one end of the candidate road to region A_2 at the other end. This expected value is based on the probability density function of the trip event from Equation (3) normalised by the total probability of making this trip, i.e.

$$p^*_{\mathbf{x}_a,\mathbf{x}_b,T}(\mathbf{x}_1,t_1,\mathbf{x}_2,t_2) = \begin{cases} p_{\mathbf{x}_a,\mathbf{x}_b,T}(\mathbf{x}_1,t_1,\mathbf{x}_2,t_2)/p_{12} & \text{if} \quad \mathbf{x}_1 \in A_1, \mathbf{x}_2 \in A_2 \text{ and } t_2 > t_1 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The normalisation with p_{12} ensures that $p^*_{\mathbf{x}_a,\mathbf{x}_b,T}(\mathbf{x}_1,t_1,\mathbf{x}_2,t_2)$ integrates to one over all $(\mathbf{x}_1,t_1,\mathbf{x}_2,t_2)$. Recall that p_{12} represents the probability of starting in A_1 at $t_1 \in [0,T]$ and later arriving in A_2 at $t_2 \in [t_1,T]$.

The travel time is $t_2 - t_1$, so the expected travel time from one end of the road segment to the other is given by Equation (8).

$$\begin{aligned} E[t_2-t_1] &= \int_{x_1\in A_1}\int_0^T\int_{x_2\in A_2}\int_{t_1}^T (t_2-t_1)p^*\left(\mathbf{x}_1,t_1,\mathbf{x}_2,t_2\right)\mathrm{d}\mathbf{x}_2\,\mathrm{d}t_2\,\mathrm{d}\mathbf{x}_1\,\mathrm{d}t_1 \\ &= \frac{1}{p_{12}}\int_{x_1\in A_1}\int_0^T p_{\mathbf{x}_a,\mathbf{x}_b,T}\left(\mathbf{x}_1,t\right) \\ &\quad \times \int_{x_2\in A_2}\int_{t_1}^T (t_2-t_1)p_{\mathbf{x}_1,\mathbf{x}_b,T-t_1}\left(x_2,y_2,t_2-t_1\right)\mathrm{d}\mathbf{x}_2\,\mathrm{d}t_2\,\mathrm{d}\mathbf{x}_1\,\mathrm{d}t_1 \\ &\approx \frac{|A_1||A_2|}{|A_1||A_2|w_{12}}\int_0^T p_{\mathbf{x}_a,\mathbf{x}_b,T}\left(\mathbf{x}_1,t\right)\int_{t_1}^T (t_2-t_1)p_{\mathbf{x}_1,\mathbf{x}_b,T-t_1}\left(x_2,y_2,t_2-t_1\right)\mathrm{d}t_2\,\mathrm{d}t_1 \\ &\approx \frac{1}{w_{12}}\int_0^T p_{\mathbf{x}_a,\mathbf{x}_b,T}\left(x_1,t\right)\int_{t_1}^T (t_2-t_1)p_{\mathbf{x}_1,\mathbf{x}_b,T-t_1}\left(x_2,y_2,t_2-t_1\right)\mathrm{d}t_2\,\mathrm{d}t_1. \end{aligned} \tag{8}$$

Here, we have made the same assumption about the flatness of the Brownian bridge in regions A_1 and A_2. If we evaluate the integral in Equation (8) numerically using the example in Figure 4 and $\sigma_\mathrm{m}^2 = 100$, we get $E[t_2 - t_1] = 2.0176$. This is a reasonable value, since the total travel time between the two location measurements is $T = 10$, and the road segment is about 35% as long as the distance between the two measurements. We note that we can reverse the indices on the road segment's endpoints to get a travel time going in the opposite direction on the candidate road.

The above explains how we compute an expected travel time for one candidate road segment from one pair of location measurements. We can do this computation for every road and measurement pair, leading to robust estimates of travel time, as we explain in the next section.

6. Travel time and route computation

For our test region around Seattle, WA, USA, we have over 500,000 measurement pairs from Twitter and 265,768 candidate road segments. (Each of the 132,884 line segments forming the triangular grid actually represents two candidate roads, one in each direction.) Ideally, every Brownian bridge from each measurement pair can be used to compute a travel time for each candidate road, giving multiple travel time estimates. The numerical integrals for computing w_{12} (Equation (6)) and $E[t_2 - t_1]$ (Equation (8)) run slowly, however, so we computed travel times for each road segment from a maximum of 300 nearest measurement pairs that are within 4200 m of the road segment. We further increased the speed using the spatial index built into SQL Server®, and we ran our MATLAB® numerical integrals on a cluster of 200 CPU cores. From the list of expected travel speeds for each road candidate, we recorded the minimum, maximum, and the 98th, 95th and 50th percentiles. The high percentiles are attractive, since they represent faster movement from one end of the road to the other. These likely come from tweet pairs where the user did not linger long between departing and arriving. Taking less than the maximum speed (e.g. 95th percentile) helps eliminate speeds from Twitter bots that appear to move arbitrarily quickly between locations. None of the candidate road segments are eliminated. Each one is a candidate for routing. For candidate segments that are not in range of any measurement pairs, we set their default time cost to infinity, so they will not be chosen as a part of a route. Algorithm 1 summarises the steps we use to compute the traversal probabilities w_{12} and expected travel times.

Once we have all the travel times, we can compute a route over our road network using traditional A* search for a given start and end location on the map, seeking to minimise travel time.

Algorithm 1 Traversal probabilities and expected travel times using probabilistic time geography

```
Input: (1) $N$ measurement pairs (e.g. Tweets) $[(x_i, y_i, t_i), (x_j, y_j, t_j)]$, where $x_k$ is the latitude, $y_k$ is the longitude and $t_k$ is the post time of one tweet; (2) $P$ available compute cluster nodes; (3) $\sigma_m$; (4) edge length of triangle in grid $\alpha$; (5) threshold $k$ (to limit the number of candidate measurement pairs for each triangle edge)
Output: Traversal probabilities $W$ and expected travel times $T$ between the end points of all the triangle edges
Create triangle on the given area where each triangle edge has length $\alpha$
Build spatial index for all measurement pairs using SQL SERVER
Equally split the edges of all the triangles into $P$ disjoint groups $g_1, g_2, \ldots, g_P$
Distribute the $P$ groups on $P$ nodes
    for each $p = 1$ to $P$ do
        for $e$ in do
            Find the $k$ nearest measurement pairs to edge $e$
            Calculate $k$ pairs of expected travel probabilities $W_e$ and time $T_e$ between nodes of $e$ using Equations (6) and (8) respectively
        end for
    end for
Collect the results from $P$ nodes
Output $W$ and $T$
```

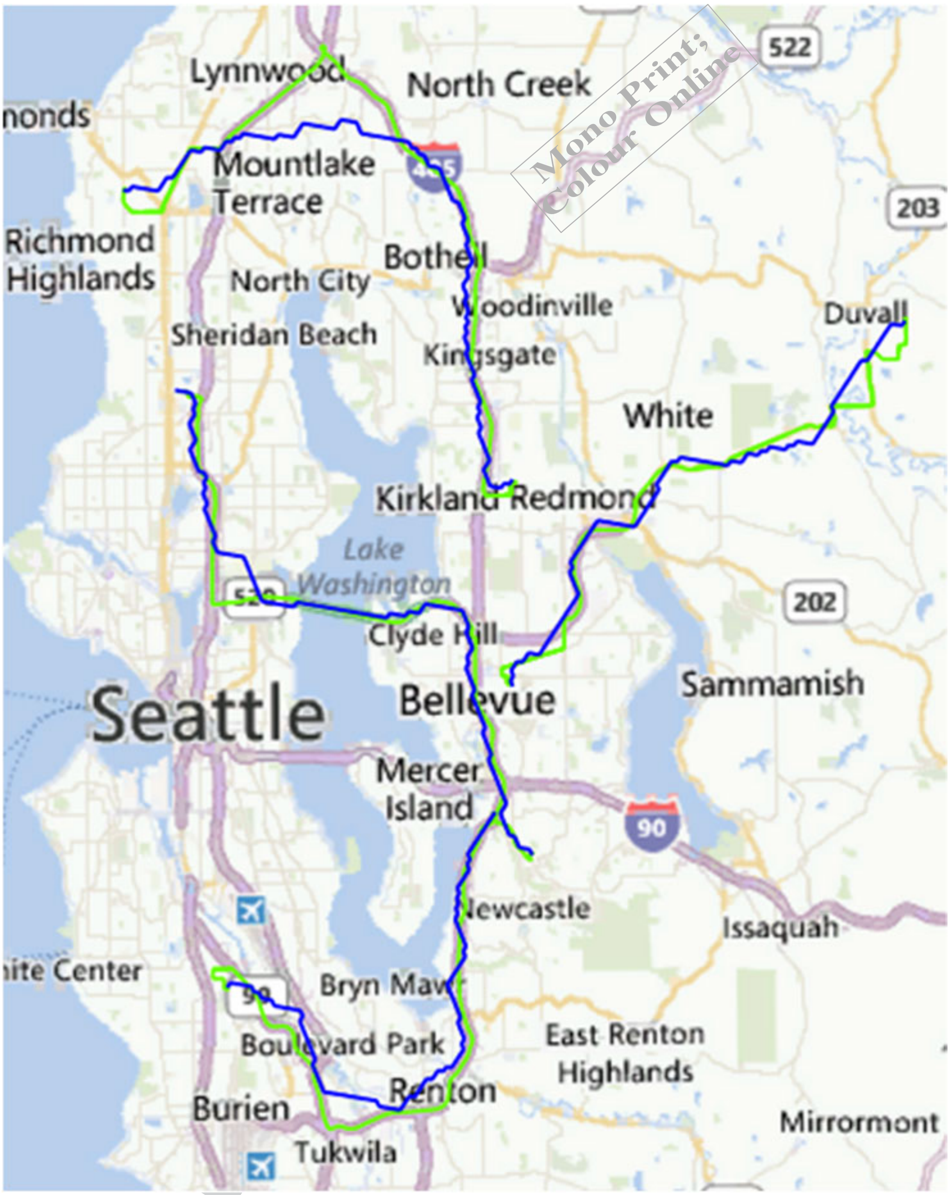

Figure 5. These are examples of the routes we computed compared to routes from a major mapping site. The green paths are from the mapping site, and the blue paths are our routes. The left side shows routes that match well, and the right side shows routes that match poorly.

One of the few ‘magic numbers’ we need for our method is the value of σ_m^2 for the Brownian bridge. This value governs the uncertainty of the vehicle’s location as it moves between the measured endpoints of its route. In their animal tracking work, Horne et al. (2007) give a principled method for determining σ_m^2. For our work, we tried various values and noticed that the best results were for $\sigma_m^2 = 100$ and fairly insensitive to changes on either side of this value. We used this value for all our experiments.

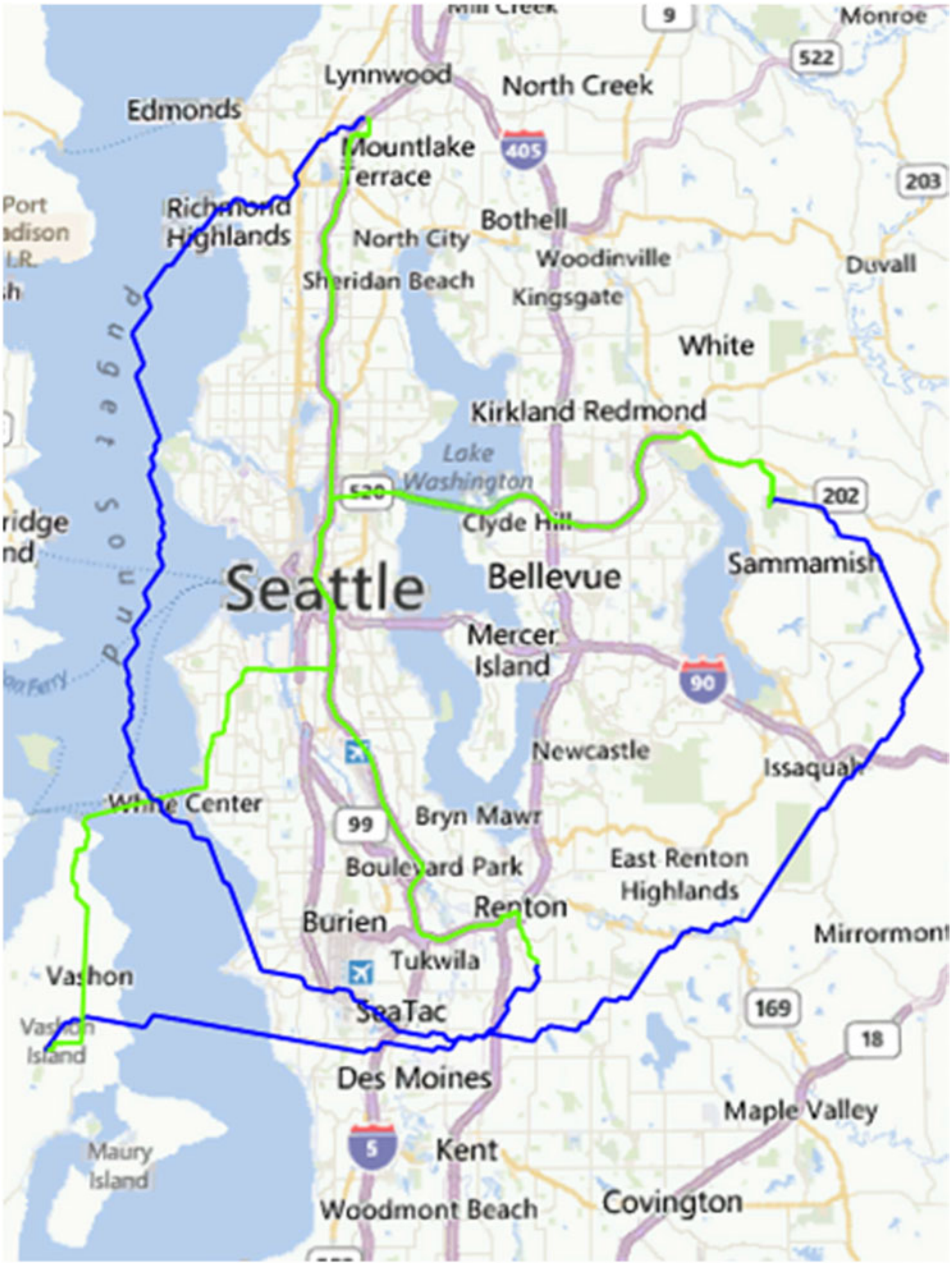

Figure 5. Continued.

7. Vehicle routing experiments

Our goal was to compute vehicle routes from Twitter data that match routes computed from the road network. To test this ability, we created 10,000 ground truth routes between random endpoints using the Bing® Maps route engine. Specifically, we chose endpoints at random from all the road intersections in our test area. The resulting routes are given as a

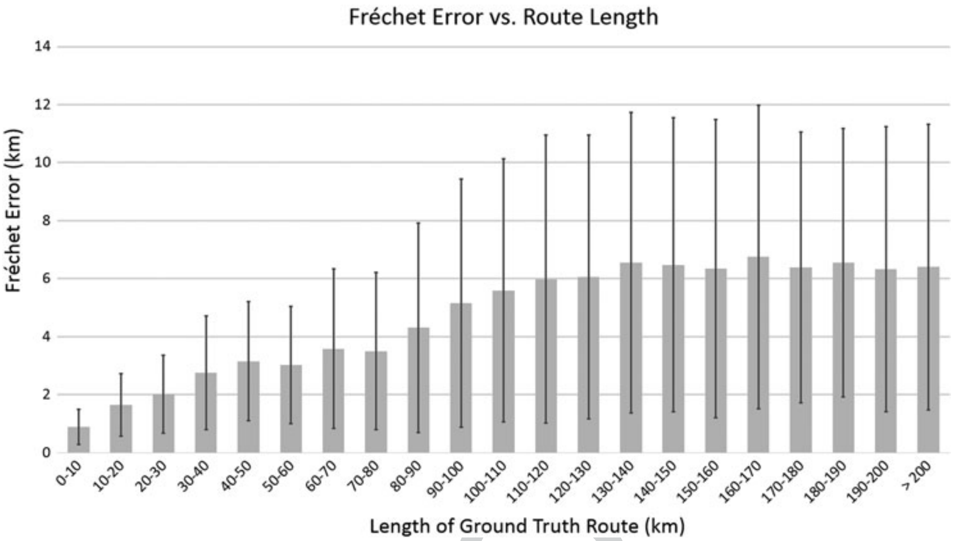

Figure 6. The Fréchet error between our computed routes and ground truth goes up with the length of the route and then stays approximately constant after a route length of 130 km. The error bars show ± 1 standard deviation.

polyline of latitude–longitude pairs. For each ground truth route, we also computed a corresponding route with the same endpoints using our candidate roads and our 95th percentile computed driving speeds. Note that we computed our test routes without any knowledge of the actual road network, relying only on our candidate grid of roads and their expected travel times computed from Twitter data.

Figure 5 shows a few of the routes we computed and their corresponding ground truth. The map on the left shows routes that matched well. At first glance, one of the remarkable features of the computed routes is that they do not cut through bodies of water, which would often give a more direct route. Instead, our routes tend to stay near the actual ground truth route. This is despite the fact that we do not explicitly eliminate any candidate roads from our network, including those that lie on the water. We also note that the original data from Figure 1 show several tweet pairs that slash across the map along no actual road, but our router was not distracted by these impossible paths.

The map on the right in Figure 5 also shows some poor matches between the ground truth routes and our computed routes. Two of the mistaken routes cut across the water where there is no ferry service, although they do seem to be taking feasible boat routes. This is because our method makes no attempt to distinguish boat traffic from car traffic, and thus our method chooses a water route as the most efficient.

To quantify the quality of our computed routes, we compared them with the ground truth routes using the Fréchet distance. This measure of similarity is commonly used to compare two geometric trajectories. It is often described as the minimum length of a leash that would connect a human walking on one trajectory and a dog on the other. Each may move at any speed or stop, but cannot go backwards. Over our 10,000 test routes, the average Fréchet distance between them and our computed routes was 4.4 km, and the median was 2.4 km. On average, the Fréchet distance was about 6.2% of the ground truth route length, and the median was 4.4%, meaning the error in our computed routes was relatively low. For routes

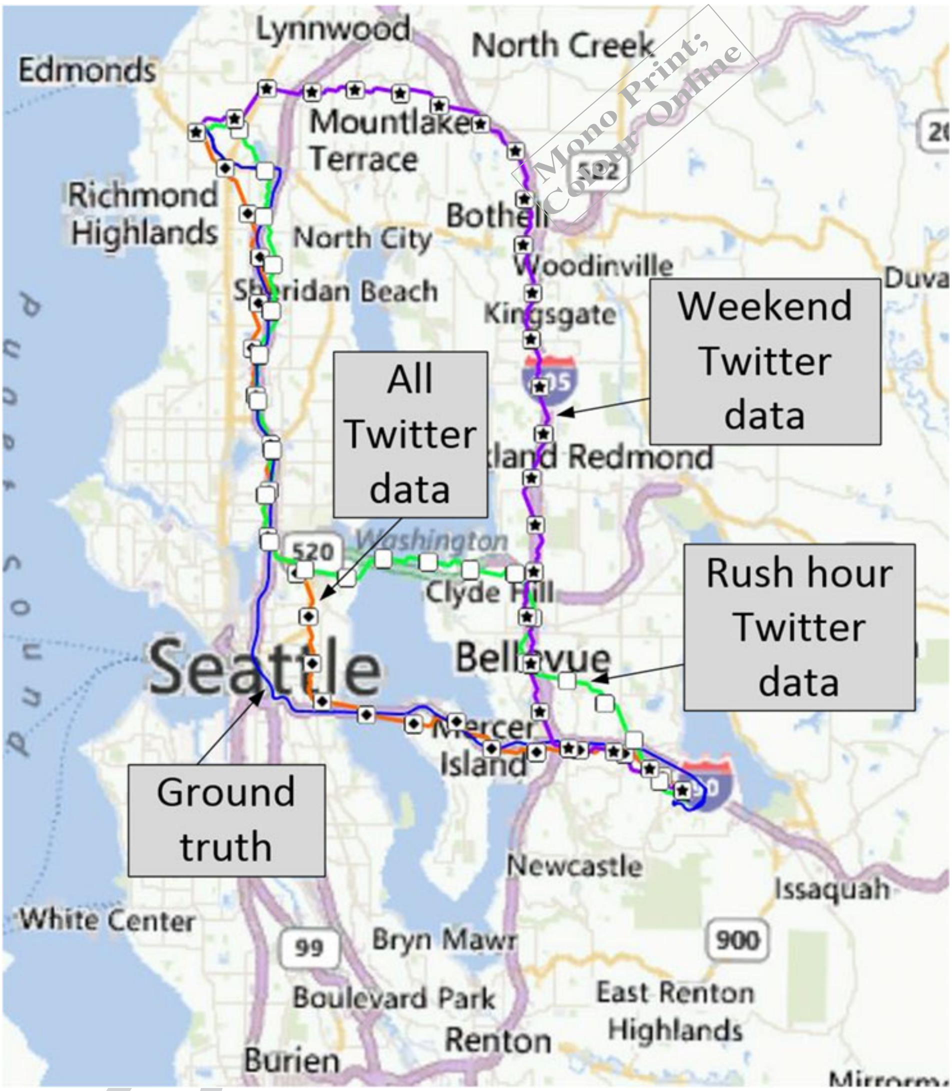

Figure 7. Our computed routes vary depending on the time periods of the Twitter data.

between 0 and 10 km, the average Fréchet distance was 0.6 km, and the error generally grew with longer routes. This trend is shown in Figure 6. We attribute this increase in deviation to the fact that longer routes give more opportunity for deviation.

This experiment shows that our computed routes are close to routes computed in the normal way. This is a positive outcome, as long as we assume that drivers follow the routes computed by a mapping program. This may be true in general, but we know that drivers often deviate. We know, for instance, that drivers choose different routes depending on traffic conditions. One of the advantages of our approach is that we compute routes based on how people really travel in the world, based on their own opinions of what makes the best route. We checked one route to see if our method was sensitive to gross differences in time. The resulting routes are shown in Figure 7 for a route starting in the upper left of the

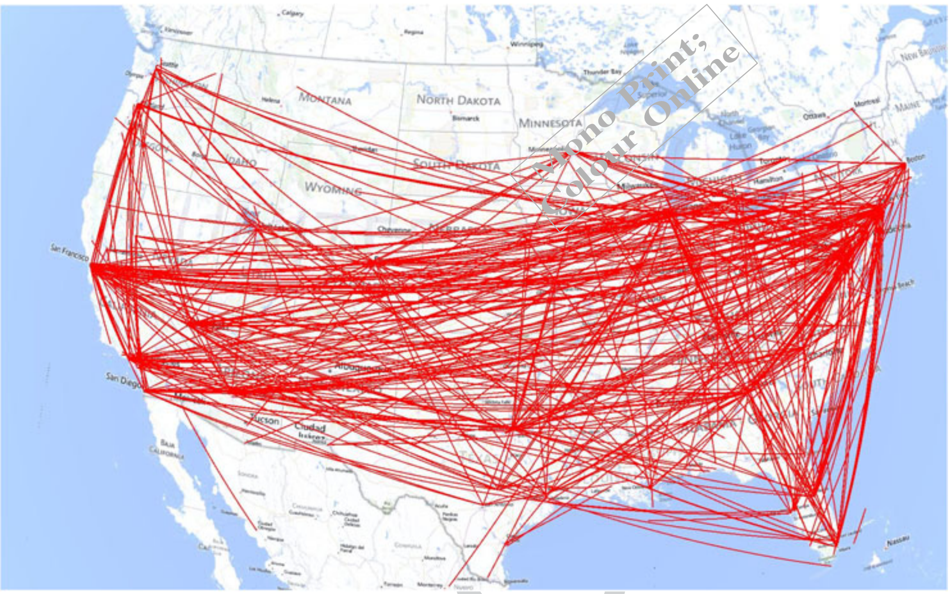

Figure 8. These line segments show tweet pairs that qualified as airplane trips based on speed and distance.

map and ending towards the lower right. The ground truth route, computed from Bing® Maps, is fairly well reproduced by the route computed from all the Twitter data. However, if we use only Twitter data from rush hours (weekdays between 6 a.m. and 10 a.m. and between 5 p.m. and 8 p.m.), we see a different route that uses a toll bridge over the central lake rather than the untolled bridge to the south. If we use Twitter data from weekends only, we get a completely different route that avoids all major bridges, curving over the north end of the lake. While we cannot yet say our computed routes match what people actually do under different conditions, we do know that our method is sensitive to the time of day and day of the week, and it seems to give reasonable routes under various conditions in this example.

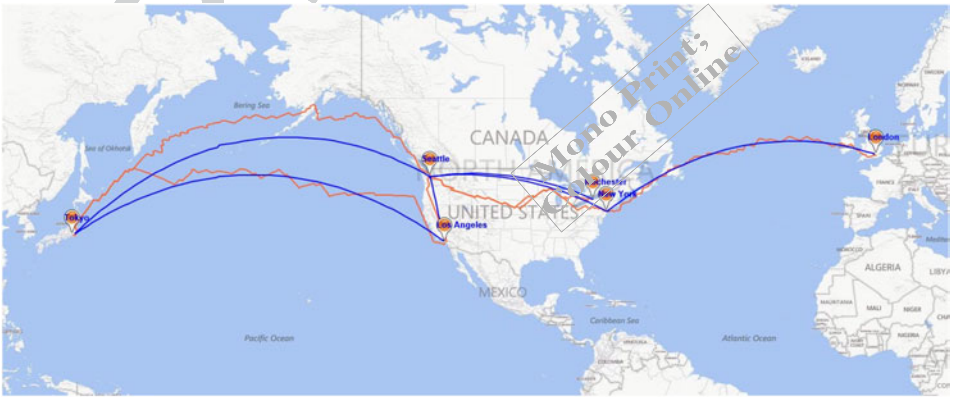

Figure 9. Our computed flight paths are close to the great circle path for cross-oceanic routes.

Up to this point, we have concentrated on ground-based transportation (and inadvertently on water-based transportation). Recalling our long-term goal of multimodal travel plans, we also tested our technique for air travel, where our goal is to infer popular airline routes. These could eventually be linked to the driving routes discovered previously to create a multi-modal plan, e.g. a terrestrial vehicle to the airport, followed by an airplane trip, and then another terrestrial vehicle to the destination. We found airline routes by making two changes to our process. The first change is that we use tweet pairs that are more likely to represent air travel. We require that the endpoints of each pair must be at least 100 km apart and that the average speed be at least 150 miles per hour. This leaves us with less than 0.15% of our original tweet pairs, but still enough to compute routes. A subset of the qualifying tweet pairs is shown in Figure 8. The other change involves our triangular grid. We enlarge it to cover the world and we make it coarser to save computation. The approximate length of a triangle edge for our road routes was 245 m, but the length was about 62 km for our coarser, worldwide air travel network. Other than this, all our computations are the same. Figure 9 shows six air travel routes computed this way: London to New York, Seattle to New York, Seattle to Rochester, NY, Seattle to Los Angeles, Tokyo to Seattle and Tokyo to Los Angeles. Most of the routes approximate the great circle route, except for the two east–west routes in the USA. These two routes are likely affected by shorter tweet pairs, which can divert the direct route towards intermediate airports.

8. Related work

Our goal is to find routes based on how people move around. One of the first mentions of this idea comes from an urban legend in Boston, MA, USA. The story is that the city chose its chaotic street layout by paving the paths that had been established by wandering cows. While the story is not true (Dias 2004), the idea is close to ours: watch travellers moving and then piece together parts of their routes into new routes. (This suggests future work where cows are using Twitter.) Animals also played a central role in the work of Horne et al. (2007), who described the use of Brownian bridges to study the movement of black bears whose location samples were usually 7 h apart. From this they could find places where bears were likely to cross a road. In the world of humans, other work has collected location data to assess the dynamics of local populations. An example is CitySense™, which shows how the popularity of different sections of a city vary with time, helping people find nightlife. Skyhook's Geospatial Insights product gives customers access to recorded population data in 100 m × 100 m tiles by hour. Their data come from users of their WiFi/cell tower location service. Our work is not aimed at where people go, but rather what routes they use to get there. Efficient routing was the goal of the ClearFlow project from Microsoft Research (Markoff 2008). It used GPS data from regular drivers to help infer traffic conditions on roads without traffic sensors. Similarly, Waze© has developed a business around gathering GPS data from its users, from which it can generate time-efficient driving routes. In our work, we compute routes without any knowledge of the road network. More closely related is Walkie-Markie (Shen et al. 2013). This project was aimed at inferring walking paths through a building. The paths were generated by dead reckoning from inertial sensors carried by the building's occupants. Since this suffers from drift, their algorithm resets the dead reckoning locations at virtual landmarks, which were the locations of signal strength maxima from the building's WiFi access points. Our work differs in that we have no data between our location measurements and that we are working outside with Twitter data from thousands of users.

Another related area is that of building road maps from GPS data. These maps have a graph representation of the road network, including edge weights that approximate the cost of driving along each road segment. Cartographers traditionally construct these maps with a combination of latitude–longitude trajectories and aerial images, both from dedicated mobile platforms. Researchers have shown how to automate this process using GPS trajectories from everyday drivers (Biagioni and Eriksson 2012; Cao and Krumm 2009; Davics et al. 2006; Edelkamp and Schrödl 2003; see Biagioni and Eriksson 2012 for an excellent survey) or publicly accessible aerial imagery (e.g. Seo et al. 2012). For GPS trajectories, automation is especially important, because the resulting maps can reflect changes in the road network in a timely way.

While automatically producing a road graph from GPS traces is feasible, efforts to date are limited in scope to relatively small areas, such as a subsection of a city. (Hybrid efforts such as Google Maps™ and OpenStreetMap (Haklay and Weber 2008) still require manual editing.) This is because the techniques developed so far use densely sampled GPS data (e.g. Cao and Krumm 2009) uses a 1-s sampling interval), and there are no publically available, dense GPS data-sets that cover much larger areas. In addition, building a map does not fully represent people's implicit travel preferences as we do with our technique.

9. Summary

Location data from social media sites such as Twitter is attractive for computing travel routes because of its freshness and wide coverage. However, the logged locations are generally widely separated, making it difficult to determine the intermediate route. We use probabilistic time geography, specifically Brownian bridges, to give a probability distribution of intermediate locations. We introduced a principled method to project the probability onto a candidate grid of roads and compute expected travel times. Using the grid as a graph, we can plan routes using a conventional A* search. Our computed routes matched well with routes computed from a regular router. Furthermore, we showed how our computed routes vary between rush hours and weekends, and we also computed air travel routes.

Perhaps the most surprising aspect of our method is its simplicity. Although our graph of candidate roads covered the region indiscriminately, we did not have to adjust it by eliminating rarely used edges, such as those on the water. Instead, the computed travel times were enough to guide the routes onto nearly the correct paths without any knowledge of the underlying road network.

The simplicity of our method invites future work aimed at creating a full travel plan. We already showed that we can distinguish air travel from surface travel using speed and distance thresholds. We could likely distinguish ground and water travel with a map of rivers, lakes and oceans. Furthermore, we could distinguish different modes of travel based on the source of the data: Buses, taxis and trains often have dedicated GPS loggers on board, and there are specialised websites where people upload their walks, hikes, runs and bicycle rides. With routes from multiple modes of travel, it will be possible to plan multi-modal trips, such as a taxi to the airport, a flight, and then a train to the final destination. Another challenge is to describe a travel plan to a user. We have shown how to compute the geometry of a trip, but people need higher level directions, especially when switching transit modes.

References

Ben-Akiva, M., M. Bergman, A. J. Daly, and R. Ramaswamy. 1984. *Modeling Inter-Urban Route Choice Behaviour*. Utrecht: VNU Press.

Biagioni, J., and J. Eriksson. 2012. "Map Inference in the Face of Noise and Disparity." *Proceedings of the 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS 2012)*, Redondo Beach, CA.

Biagioni, J., and J. Eriksson. 2012. "Inferring Road Maps from Global Positioning System Traces: Survey and Comparative Evaluation." *Transportation Research Record: Journal of the Transportation Research Board* 2291: 61–71.

[AQ1] Brush, A., J. Krumm, and J. Scott. 2010. *Exploring End User Preferences for Location Obfuscation, Location-Based Services, and the Value of Location*. ACM.

Cao, L., and J. Krumm. 2009. "From GPS Traces to a Routable Road Map." *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS 2009)*, Seattle, WA.

[AQ2] Davics, J., A. R. Beresford, and A. Hopper. 2006. *Scalable, Distributed, Real-Time Map Generation*. IEEE.

[AQ3] Dias, J. 2004. *How Now? Cow Path Tale is Pure Bull*.

[AQ4] Edelkamp, S., and S. Schrödl. 2003. *Route Planning and Map Inference with Global Positioning Traces*. Springer.

[AQ5] Graham, M., and M. Stephens. 2012. *A Geography of Twitter*.

[AQ6] Haklay, M., and P. Weber. 2008. *OpenStreetMap: User-Generated Street Maps*. IEEE.

Horne, J. S., E. O. Garton, S. M. Krone, and J. S. Lewis. 2007. "Analyzing Animal Movements Using Brownian Bridges." *Ecology* 88: 2354–2363.

[AQ7] Laurila, J. K., D. Gatica-Perez, I. Aad, J. Blom, O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, and M. Miettinen. 2012. *The Mobile Data Challenge: Big Data for Mobile Computing Research*.

[AQ8] Lin, K., A. Kansal, D. Lymberopoulos, and F. Zhao. 2010. *Energy-Accuracy Trade-Off for Continuous Mobile Device Location*. ACM.

[AQ9] Markoff, J. 2008. *Microsoft Introduces Tool for Avoiding Traffic Jams*.

[AQ10] Paek, J., J. Kim, and R. Govindan. 2010. *Energy-Efficient Rate-Adaptive GPS-Based Positioning for Smartphones*. ACM.

[AQ11] Pajor, T. 2009. *Multi-Modal Route Planning*. Universität Karlsruhe.

[AQ12] Rios, M. 2013. *The Geography of Tweets*.

Seo, Y.-W., C. Urmson, and D. Wettergreen. 2012. "Ortho-Image Analysis for Producing Lane-Level Highway Maps." *Proceedings of the 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS 2012)*, Redondo Beach, CA.

[AQ13] Shen, G., Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang. 2013. "Walkie–Markie: Indoor Pathway Mapping Made Easy." In *Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI'13)*. USENIX Association 13.

[AQ14] Szalay, A. S., J. Gray, G. Fekete, P. Z. Kunszt, P. Kukol, and A. Thakar. 2005. *Indexing the Sphere with the Hierarchical Triangular Mesh*. Technical Report MSR-TR-2005-123, Microsoft Research.

Winter, S., and Z.-C. Yin. 2010. "Directed Movements in Probabilistic Time Geography." *International Journal of Geographical Information Science* 24 (9): 1349–1365.

Winter, S., and Z.-C. Yin. 2010. "The Elements of Probabilistic Time Geography." *Geoinformatica* 15: 417–434.

Appendix

Theorem. $p_{\mathbf{x}_a,\mathbf{x}_b,T}(\mathbf{x}_1, t_1, \mathbf{x}_2, t_2)$ in Equation (3) is a legitimate probability density function.

PROOF. According to the definition of probability function, we need to prove that Equation (3) satisfies both non-negativity and that the integral over all possible values equals 1.

1. Non-negativity

Since we have

$$p_{\mathbf{x}_a,\mathbf{x}_b,T}(\mathbf{x}_1,t_1,\mathbf{x}_2,t_2)=p_{\mathbf{x}_a,\mathbf{x}_b,T}(\mathbf{x}_1,t_1)p_{\mathbf{x}_1,\mathbf{x}_b,T-t_1}(\mathbf{x}_2,t_2-t_1),$$

by the definition of $p_{\mathbf{x}_a,\mathbf{x}_b,T}(\mathbf{x},t)$ in Equation (1) the two factors are both nonnegative everywhere.

2. The probability density function integrates to 1.

$$\begin{aligned}
&\int p_{\mathbf{x}_a,\mathbf{x}_b,T}(\mathbf{x}_1,t_1,\mathbf{x}_2,t_2)\mathrm{d}\mathbf{x}_1\mathrm{d}t_1\mathrm{d}\mathbf{x}_2\mathrm{d}t_2\\
&=\int_{\mathbf{x}_1}\frac{1}{T}\int_0^T N\big((\mathbf{x}_1);(\mu(t_1),\sigma(t_1))\big)\int_{\mathbf{x}_2}\frac{1}{T-t_1}\int_{t_1}^T N\big((\mathbf{x}_2);(\mu(t_2),\sigma(t_2))\big)\mathrm{d}\mathbf{x}_1\mathrm{d}t_1\mathrm{d}\mathbf{x}_2\mathrm{d}t_2\\
&=\int_{\mathbf{x}_1}\frac{1}{T}\int_0^T N\big((\mathbf{x}_1);(\mu(t_1),\sigma(t_1))\big)\int_{\mathbf{x}_2}\frac{1}{T-t_1}\int_{t_1}^T N\big((\mathbf{x}_2);(\mu(t_2),\sigma(t_2))\big)\mathrm{d}\mathbf{x}_1\mathrm{d}t_1\mathrm{d}\mathbf{x}_2\mathrm{d}t_2\\
&=\int_{\mathbf{x}_1}\frac{1}{T}\int_0^T N\big((\mathbf{x}_1);(\mu(t_1),\sigma(t_1))\big)\left(\frac{1}{T-t_1}\int_{t_1}^T\left(\int_{\mathbf{x}_2}N\big((\mathbf{x}_2);(\mu(t_2),\sigma(t_2))\big)\mathrm{d}\mathbf{x}_2\right)\mathrm{d}t_2\right)\mathrm{d}\mathbf{x}_1\mathrm{d}t_1\\
&=\frac{1}{T}\int_{\mathbf{x}_1}\int_0^T N\big((\mathbf{x}_1);(\mu(t_1),\sigma(t_1))\big)\left(\frac{1}{T-t_1}\int_{t_1}^T\mathrm{d}t_2\right)\mathrm{d}\mathbf{x}_1\mathrm{d}t_1\\
&=\frac{1}{T}\int_{\mathbf{x}_1}\int_0^T N\big((\mathbf{x}_1);(\mu(t_1),\sigma(t_1))\big)\mathrm{d}\mathbf{x}_1\mathrm{d}t_1\quad=\frac{1}{T}\int_0^T\int_{\mathbf{x}_1}N\big((\mathbf{x}_1);(\mu(t_1),\sigma(t_1))\big)\mathrm{d}\mathbf{x}_1\mathrm{d}t_1\\
&=1.
\end{aligned}$$

Thus $p_{\mathbf{x}_a,\mathbf{x}_b,T}(\mathbf{x}_1,t_1,\mathbf{x}_2,t_2)$ satisfies both properties. Therefore, it is a legitimate probability density function. □