USING SEGMENT FRACTION FOR ROAD-NETWORK LOCATIONS

Yuanyuan Pao
Lyft
185 Berry Street Suite 5000
San Francisco, CA 94107
ypao@lyft.com

ABSTRACT

Locations are often specified as (latitude, longitude) coordinates, which require specific distance methods based on spherical geometry to process. If the locations lie on a road network, then operations such as nearby search, speed computation, location prediction, or interpolation will require further processing with the road geometry. We propose augmenting the location representation with road segment ID and segment fraction for road-based locations and argue that, in doing so, commonly-used operations are greatly simplified and correspond to intuitive formulas.

***K*eywords** location · latitude · longitude · road geometry · segment fraction · map matching

1 Introduction

When dealing with locations on the road network, geographic coordinates, commonly represented in the form of (latitude, longitude), can often be difficult to use and are hard to interpret. Not only do these coordinates omit any explicit road network information, such as what road segment the location belongs to, but even simple operations such as speed computation cannot be performed without summing road distance over the segment geometry. In certain applications, map projection [1] and linear referencing [2] have been utilized to make locations easier to analyze or to associate certain attributes without compromising precision. We propose extending the concept of linear referencing to the entirety of road-based locations and show that we can perform local searches, speed computation, location prediction, and interpolation with simple, linear formulas.

2 Definition and Representation

For any location coordinate on a road segment, we propose processing locations with (road segment ID, segment fraction) instead of (latitude, longitude) coordinates.

Road Segment ID We define a road segment ID s as a unique identifier (integer or string) for a given road segment between two intersections. A road segment ID should change at any junction or decision point.

Segment Fraction The segment fraction p encodes a location as a proportion along the length of the segment. Therefore, for each segment ID, we must pre-define a start and end; this does not necessarily imply a road travel direction and can be arbitrarily chosen between the two endpoints of the segment.

With the unique id for a road segment and a segment fraction, we can associate a (latitude, longitude) coordinate with the road network using the pair (s, p). To illustrate this, an example is shown in Figure 1.

2.1 Granularity

For (latitude, longitude) coordinates, we usually require 5 or 6 decimal places to achieve approximately 1-meter granularity. With most segment lengths in OpenStreetMap [3] under 2 kilometers, segment fractions would only need 3 to 4 significant digits to achieve 1-meter granularity. For floating point accuracy, more significant digits require careful numeric representations and are subject to rounding errors.

Figure 1: A (latitude, longitude) coordinate with its corresponding road segment ID s and segment fraction p

2.2 Map Data

With locations represented as (road segment ID, segment fraction) pairs, we need access to the map data during computation. For graph traversal or shortest path queries, we will rely on services such as Open Source Routing Machine (OSRM) [4]. For segment information, we will primarily store the segment length and the start and end locations, which can be stored by road segment ID in a dictionary data structure for $\mathcal{O}(1)$ lookup. The WKT road geometry can also be stored in order to convert back into geographic coordinates.

3 Commonly-Used Operations Are Easy

This representation of road-based locations is especially beneficial for common functionality that requires knowing the road distance between any pair of locations. Typically, for (latitude, longitude) coordinates, this would require iterating over the road geometry. Segment fraction removes this burden for performing local search, computing speed, predicting, and interpolating.

In the following sections, we will represent a location using (s, p), where s is the unique road segment ID and p is the segment fraction in $[0, 1]$. We will use $l_{\mathtt{s}}$ to refer to the road segment length stored as part of our map data.

3.1 Local Search

Searching for all locations within some road distance δ of a location (s_0, p_0) is simplified with road segment fractions and segment lengths. If the search distance δ is less than the distance to the endpoints of the current segment, then the result is the set of locations on the same segment s_0 that satisfy

$$(s_0, p) : p \in \quad p_0 - \frac{\delta}{l_{\mathtt{s_0}}}, p_0 + \frac{\delta}{l_{\mathtt{s_0}}} \quad . \tag{1}$$

If the current segment is not long enough, then it is simple to recurse along neighboring segments. For example, if $\delta > l_{\mathtt{s_0}} p_0$, then the returned set should include what is returned by all local searches for neighboring segments that share location $(s_0, 0)$ and distance $\delta - l_{\mathtt{s_0}} p_0$. Similarly, if $\delta > l_{\mathtt{s_0}}(1 - p_0)$, the the returned set should include what is returned by all local searches for segments that share location $(s_0, 1)$ and distance $\delta - l_{\mathtt{s_0}}(1 - p_0)$.

3.2 Speed Computation

Given two locations (s_0, p_0) and (s_1, p_1) with corresponding timestamps t_0 and t_1, computing the average speed between these two locations is much easier than with (latitude, longitude) coordinates. If the locations share the same segment s, then we have

$$v = \frac{\Delta d}{\Delta t} = \frac{l_{\mathtt{s}}(p_1 - p_0)}{t_1 - t_0}. \tag{2}$$

Otherwise, we must first fetch the total road distance between the two locations D, which adds an additional call to OSRM for the shortest path.

Figure 2: A curved section of Interstate 5 (I-5), shown as a blue line, that we used for our numerical examples.

3.3 Prediction

Location prediction along the road segment requires only a straightforward calculation. Given a location (s_0, p_0) and a road distance δ to project (or velocity and time), if the road distance is less than the distance to the segment endpoint, we can compute the predicted location (s_0, p) where

$$p = p_0 \pm \frac{\delta}{l_s}. \tag{3}$$

Whether we add or subtract is dependent on the direction relative to the segment orientation defined by the start and end location in our road segment data lookup table. We note that, for the cases when δ extends beyond our current segment, we will also need a nearby search for all candidate segments within δ-distance.

3.4 Interpolation

Interpolation between points along the same road segment is now computed via a weighted average. Given two locations (s_0, p_0) and (s_1, p_1) with corresponding timestamps t_0 and t_1 and $s_0 = s_1$, we can write the location at time $t_0 < t \leq t_1$ as (s_0, p) where

$$p = \frac{t - t_0}{t_1 - t_0} p_1 + \frac{t_1 - t}{t_1 - t_0} p_0 = p_0 + \frac{t - t_0}{t_1 - t_0}(p_1 - p_0). \tag{4}$$

If $s_0 \neq s_1$, then, as before, we must obtain the trajectory and distance between the two locations D and compute the prediction along that path with starting location (s_0, p_0) and distance $\frac{t-t_0}{t_1-t_0} D$.

4 Numerical Examples

For our numerical examples, we will use a curvy 1-kilometer stretch of Interstate 5 in Seattle, Washington, shown in Figure 2.

- **Local Search** With $p = 0.75$ and $\delta = 100$ meters, we first lookup (or compute) the segment length, $l_s = 1068.9$ meters and then return our nearby search range as $0.6564 \leq p \leq 0.8436$.
- **Speed Computation** Say we have two locations on our segment, one with $p_0 = 0.25$ at time $t_0 = 4$ and the other with $p_1 = 0.7$ at time $t_1 = 28$. We compute the average speed for this pair of locations directly as $v = 20.04$ m/s.
- **Prediction** If we switch the known parameters from our speed example to p_0, t_0, v, we should get the same p_1 for t_1. Computing $p = p_0 + \frac{v(t_1 - t_0)}{l_s}$, we get back $p_1 = 0.7$.
- **Interpolation** For the same two known locations, we will interpolate the location at $t = 20$ as $p = 0.25 + \frac{2}{3}(0.7 - 0.25) = 0.55$.

With these quick calculations from segment fractions, we can convert the locations back to (latitude, longitude) coordinates and visualize the results as shown in Figure 3.