information

Article

Travel-Time Estimation By Cubic Hermite Curve

Owen Tamin [1,2], Badrul Ikram [1], Ahmad Lutfi Amri Ramli [1], Ervin Gubin Moung [2,*] and Christie Chin Pei Yee [2]

[1] School of Mathematical Sciences, Universiti Sains Malaysia, Gelugor, Penang 11800, Malaysia; owentamin1996@gmail.com (O.T.); badrulikram31@gmail.com (B.I.); alaramli@usm.my (A.L.A.R.)
[2] Faculty of Computing and Informatics, Universiti Malaysia Sabah, Kota Kinabalu, Sabah 88400, Malaysia; peiyee@ums.edu.my
* Correspondence: ervin@ums.edu.my

Abstract: Travel time is a measure of time taken to travel from one place to another. Global Positioning System (GPS) navigation applications such as Waze and Google Maps are easily accessible presently and allow users to plan a route based on travel time from one place to another. However, these applications can only estimate general travel time based on a vehicle's total distance and average safe speed without considering route curvature. A parametric cubic curve has shown a potential result in travel-time estimation through geometric properties. In this paper, travel time has been estimated using the curvature value obtained from the Hermite Interpolation curve fitted to each section of the selected road. Design speed is determined from the curvature value, and thus an algorithm for travel-time estimation incorporating initial driving information is developed. The proposed method's accuracy was compared to the existing method's accuracy using a real-life driving test. This comparison demonstrated that the proposed method estimates travel time more accurately than Google Maps and Waze. Future study can further improve the estimation by embedding traffic data into the algorithm.

Keywords: travel-time estimation; Hermite interpolation curve; curvature; design speed; geometric properties; driving information

check for updates

Citation: Tamin, O.; Ikram, B.; Amri Ramli, A.L.; Moung, E.G.; Chin Pei Yee, C. Travel-Time Estimation By Cubic Hermite Curve. *Information* **2022**, *13*, 307. https://doi.org/10.3390/info13070307

Academic Editors: Enrique Onieva and Dmitry Pavlyuk

Received: 25 May 2022
Accepted: 17 June 2022
Published: 23 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The emergence of the Global Positioning System (GPS) navigation system with cutting-edge map technology has throughout the last decade significantly impacted how road users travel. For years, road drivers have leveraged GPS applications such as Google Maps and Waze to reduce their travel time by obtaining the best route suggestion with an estimated travel time [1]. Travel time is generally a measure of time taken to travel from one place to another. The GPS applications calculate estimated travel time based on the distance between two points and the average speed limit, as well as current traffic conditions [2]. They also collect traffic data on the speed limits of major highways, freeways, and streets. Although GPS applications rely heavily on traffic data, the accuracy of the estimated travel time may be affected in locations where traffic data are limited, such as in rural areas. Prior studies have shown that the estimation of travel time in rural areas can be calculated by implementing Markov Chains, which rely on traffic flow data from microwave detectors as one of their primary sources [3]. The modified Bayesian data fusion model collects data from loop detector data, GPS data, historical data, and ground truth data [4].

Current approaches for calculating travel time based on GPS data have their own limitations. For instance, they are less effective in handling difficulties linked to GPS and spatial road network data uncertainties, such as time window length, vehicle penetration rate, sampling frequency, and vehicle coverage on the network. Furthermore, they are more likely to rely on higher-frequency data sources from specialist data providers, which can be costly and not always available [5]. GPS applications do not consider driving speed and

the curvature of the road in predicting the estimated travel time. Heavily relying on GPS alone will result in inconsistent travel-time estimation on different routes. Especially in locations where GPS data or historical travel data cannot be relied on, the road properties can give a valuable indicator of how fast a car travels on a particular road. Therefore, driving speed and road curvature can be introduced as part of the solution in overcoming the limitations faced by other existing methods in relying upon traffic data for travel-time estimation [6]. This proposed method is cheaper, more accessible and more accurate in estimating travel time.

In this paper, one type of cubic curve, the Hermite interpolation curve, is used to ensure connection and tangent continuity between segments, which is essential in visualization. Cubic curves are commonly used in graphics due to their ability to allow some flexibility in the curve design characterized by their control polygons, and compromise between flexibility and speed of computation [6]. Approximating travel time taken by an individual based on road curvature and the average speed is the aim of this study. The proposed method has the advantage of overcoming the limitation of traffic data where it is not available in certain areas, mainly rural. The proposed method can still work well in both urban and rural areas due to the reliance on calculation. Therefore, geometric properties give a better and more precise estimation of travel time than traffic data implemented in the existing system.

In summary, the significance of this study are as follows:

- The proposed method can provide a better estimation for road users in planning their journey from the starting point to the final destination with an accurate estimated time of arrival.
- The proposed algorithm uses geometric properties of the road, which is the curvature value instead of traffic data. This value will estimate the maximum driving speed for a particular section of the road to be embedded in the travel-time estimation based on the initial driving information.

The significance of the study described above is listed in the first section of this paper. The literature review for this study is discussed in Section 2, and the research methodology is outlined in Section 3. The results are then reviewed in Section 4. The conclusion and prospective future works are presented in Sections 5 and 6, respectively.

2. Literature Review

Car drivers are risk-averse when choosing a route recommended by a system with a shorter average travel time than the same route's usual travel time. On the other hand, they tend to be risk-seeking when choosing a recommended route with a longer average travel time than the expected average travel time of the same route. This trend is repeated in driving simulation research where the reference route has a broad scope of travel time while the reference range is narrower than the range of accessible paths [7].

The curvature profile is the variability of the curve as a specific point that moves along the curve. The greater the curvature, the smaller the radius [8]. Naturally, curvature is a quantity that is inversely proportional to the radius of the circle. The fairness of a curve, which is related to the concept of curvature profile, can be seen by plotting the curvature versus the arc length for a given curve [8]. Thus, the curvature curve for the fair curves can be defined. This indicates that a curve is fair if its curvature plot consists of relatively few monotone pieces.

A parametric curve, such as a Hermite curve or a Bézier curve, employs a simple formula in which the curve is defined in terms of control polygons (a set of control vertices connected to form sequences or patterns). Although the curve resembles all control polygon patterns, it always interpolates the initial and final vertices [9]. The number of control polygons is equal to the number of edges in the polynomial curve.

This description states that global control is within the formulation, which means the movement of a control vertex influences the entire curve's shape pattern. This description indicates that global control is within the formulation. As a result, the movement of a

control vertex affects the overall shape pattern of the curve. Furthermore, because it is a polynomial, the curve is infinitely different.

A continuous vector function exists in a closed curve $\boldsymbol{C}$ in Euclidean n-space H^n. The continuous vector function of period l, which is not constant in any t-interval, is denoted as $r(t) = (r_1(t), \ldots, r_n(t))$. It is much more convenient to assume any polygon as a closed curve and ignore the differences in parameterizations. Suppose that $r(t)$ is a continuous vector function. A closed curve $r(t)$ can be simplified further to $r(t_1) = r(t_2)$ if $\frac{t_1 - t_2}{l}$ produces an integer. A closed polygon P with vertices $a_1, \ldots, a_m$ is said to be inscribed in a closed curve $r(t)$ if there is a set of parameter values t_i such that $t_i < t_{i+1}, t_{i+m} = t_i + 1$, and $a_i = r(t_i)$ for all integral values of i. If P is a polygon with at least two coincident vertices, it can be represented as the limit of a sequence of polygons with distinct vertices. As a result, all that is left is to prove the lemma for P with all vertices distinct [10].

Design speed is defined as the maximum safe speed that is sustainable under ideal road conditions. This results in highway design characteristics taking precedence over a particular section of the highway. The value of design speed is ideal based on topography, neighboring land use, and highway functional classification [11]. Thus, to achieve a safe maximum speed, the design speed should take into account vehicle safety and comfortability. According to the 1936 Barnett concept of design velocity, when the urban area is cleared, the driving community of vehicle operations will tend to be faster due to the high reasonable uniform velocity. The situation will contribute to the growing crash rate when the drivers drive along the horizontal curves of the road. The main concern at the time was that the curves were designed for non-motorized or slow-moving motorized vehicles, but car manufacturers have since been able to develop automobiles capable of reaching higher speeds [12].

According to Leisch and Leisch [13], design speed is a representative operating potential velocity defined by the design and correspondence of a highway's geometric characteristics. It indicates a nearly constant near-maximum speed that a driver on the road can comfortably maintain in good weather and with little traffic. Furthermore, it also functions as an indicator for the highway's physical condition [13]. The design speed principle, as currently applied, does not prevent inconsistencies in highway alignment. The fundamental problem, especially in the range of design speeds below 90 km/h (55 mph), is the driver's propensity to accelerate and decelerate continuously. The speed-profile technique can be used to resolve the speed disparity between cars and trucks, helping to meet the standards of drivers and comply with their inherent characteristics to achieve operational consistency and enhance driving comfort and safety.

Roads are constructed based on a suitable and safe design speed, which is the same for every given path in the roadway. The vehicle's mass is assumed to be constant in the given situation, and the vehicle must be able to travel safely and comfortably at the given path of the road regardless of the bend [14]. The process of road velocity design reflects the correlation between curvature and velocity, in which the basic principle of horizontal curve design is derived from kinematic equation applications. In the 1930s, the United States designed the roadway's vertical and horizontal alignments with design speed as the main consideration. By taking design speed into consideration, the design criteria for a new roadway were proposed to ensure the development of an appropriate horizontal curve radii, superelevation rates, and vertical curve elements.

The design criteria were suggested to develop appropriate horizontal curve radii, superelevation rates, and vertical curve elements for new roadways [11]. Simultaneously, the use of statistical analysis of the independent vehicle speeds on the freeway was introduced. There are areas in which the stated velocity limit centered on the 85th percentile speed above the roadway's recommended design speed due to variations in design and activity requirements.

It is possible to predict general travel time for routes using traditional navigation systems. For instance, regardless of the route's metric system, a travel time can be calculated by dividing the path's distance by the speed limit. However, such calculations may

contain errors due to road conditions, traffic congestion, driving habits, the accuracy of the navigation system, and other variables that navigation systems do not account for [15]. The real-time measurement for travel time of a single route consists of a set of multiple road sections. The development of the sections is essential for the needs of traffic control, searching for driving directions, ride-sharing, and taxi dispatch [16]. The use of loop sensors as a new technique would tell individuals the travel speed of a particular path section instead of the travel time of a full journey.

Furthermore, the use of Vehicular Ad-hoc Network (VANET) technology, optimal sensor location model, Cellular Floating Vehicle Data (CFVD), Travel-Time Variability (TTV) and Urgent-Gentle Class (UGC) traffic flow model will enable effective and efficient transportation in the city [17–21]. The UGC traffic flow model estimates travel time through a ring road with viscoelastic and ramp effects. This method applies mathematical modeling in travel-time estimation without traffic data. Results show that average travel time increases when the initial ring-road density increase.

Several traveler information systems have recently been developed to provide travelers with real-time traffic information. In a typical travel information system, a road map is divided into route segments. The time required for a vehicle to travel along a particular route segment is predicted using historical and real-time sensor data. On the road, travel-time predictions are based on the relationship between the distance between two points and the time required to complete the journey [22].

Over the last decade, there has been a motivation for the development and deployment of Intelligent Transportation Systems (ITS) in travel-time estimation due to the numerous advantages that these systems can primarily provide, using Bluetooth. These systems are designed to provide users with information regarding pre-trip or en route travel to help them choose the most efficient mode of transport and to ensure an optimal control strategy. Currently, most of the states in the United States provide travelers with information on current road conditions, such as speeds, travel time, incidents, and lane closures. Drivers can then be informed of their travel time via dynamic message signs, the Internet, and cell phones [23]. By analyzing information from a large sample of vehicle trips within a city, it is possible to remodel the city's traffic pattern [24].

There may be a few factors affecting travel-time estimation based on the route traveled between two points in the network. The consistency of the range of average travel time value along the pathway is one of the factors. A study was conducted in which the movement and position of the probe vehicles within a specified interval on the road were used as the primary observation for the purpose of predicting travel time. The sampling rate is assumed to be lower due to the scale required for measuring travel time, which is shorter than the distances between consecutive records. Instantaneous speed data would not be accessible because of the communication bandwidth limitations. Within these communication bandwidth constraints, the only data available is the distance traveled and the estimated travel time [25]. Disregarding the influence of the initial travel time will undoubtedly influence the identification of the shortest route and traffic data [26].

3. Materials and Methods

The algorithm that was used to compute the Hermite interpolation function will be adapted to construct the Hermite interpolation curve using coordinates of two given routes. Furthermore, the algorithm for estimating travel time based on initial driving data will also be adapted to provide a theoretical concept for calculating curvature, design speed, and travel time estimation. Additionally, this section discusses the five stages involved in estimating travel time, as well as the overall flowchart.

3.1. Research Methodology

This section proposes a suitable algorithm for estimating travel time. An algorithm for computing the Hermite interpolation function was presented. This is followed by another

algorithm that specifically estimates travel time based on initial driving information and individual driving speed profiles.

3.2. Overall Flowchart

Figure 1 shows the overall flowchart of the proposed method. The overall flowchart consists of five phases; (i) Phase A involves obtaining the route's coordinates, (ii) Phase B details the process of calculating the curvature values and curvature radius, (iii) Phase C details the process for calculating the arc lengths of the routes, (iv) Phase D describes the process of calculating the design speed that is capped by the average speed, and finally (v) Phase E describes the process of estimating travel time. Each phase will be explained in detail using theory and a flowchart, which will be included in the following section.

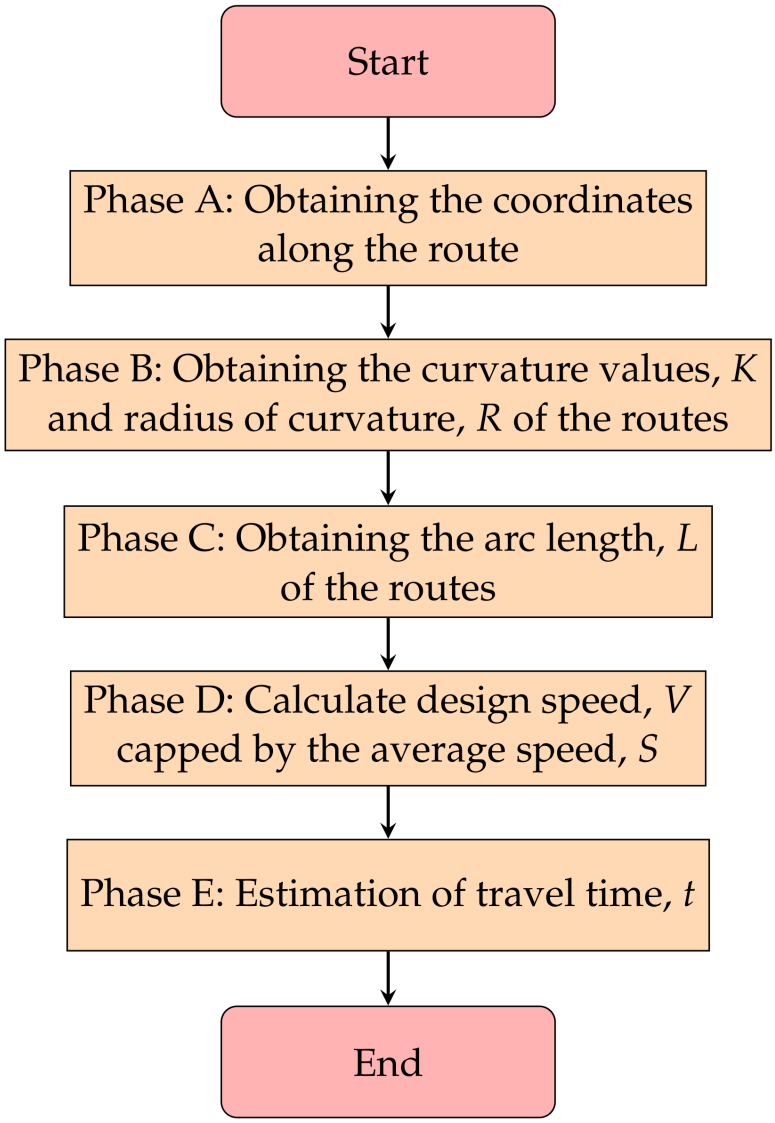


Figure 1. An overall flowchart of the proposed method.

3.3. Algorithm in Computing Hermite Interpolation Function

The procedure for estimating travel time is shown in Algorithm 1.

Algorithm 1: Hermite Interpolation Curve

Result: Produce multiple Hermite Interpolation Curves for n dote
Define n number of points where $P_i = (x_i, y_i)$, i=1,2,3,..,n
and the number of segments is $\frac{n}{2}$
Input : Points (x_i, y_i)
Output: Piecewise Hermite Interpolation Curve
if $i = 1$ **then**
    $t_1 = p_i$, $t_2 = p_{2i+1}$, $v_1 = p_{2i} - p_i$ and $v_2 = p_{2i+2} - p_{2i}$;
else
    if $i = \frac{n}{2}$ **then**
        $t_1 = p_{n-1}$, $t_2 = p_{n+1}$, $v_1 = p_n - p_{n-2}$ and $v_2 = p_{n+1} - p_n$;
    else
        $t_1 = p_{2i-1}$, $t_2 = p_{2i+1}$, $v_1 = p_{2i} - p_{2i-2}$ and $v_2 = p_{2i+2} - p_{2i}$;
    end
end
t_1 and t_2 are the two endpoints of the control points per segment and v_1 and v_2 are the two derivative conditions.
Then, let $q = (t_1, t_2, v_1, v_2)$
Obtain $c_i = M_h.q$ where M_h is the Hermite geometry matrix
Obtain the Hermite Interpolation Function

3.4. Description of Methodology

3.4.1. Phase A: Obtaining the Coordinates of the Routes' Starting and Ending Points

Figure 2 describes the process involved in Phase A. There are several methods for obtaining the route's coordinates. One method is to manually select the points. In this case, the points are selected by obtaining the coordinates from the image in Mathematica, where scaling is required. The coordinates obtained in units based on the map scale will then be converted to meters.

Another way to obtain coordinates is using Garmin BaseCamp as shown in Figure 3. Garmin Basecamp is a map-viewing or GIS software package primarily used with Garmin GPS navigation devices. Garmin BaseCamp is used because it can detect a significant number of point locations along the selected routes, yielding positive results in the estimation of travel time. Garmin BaseCamp records all coordinate points within a 15 to 25 m interval along the route in longitude and latitude formats.

Once the coordinates are obtained, the coordinates need to be converted into (x_i, y_i) forms. Due to the fact that the coordinates obtained from Garmin BaseCamp are in longitude and latitude formats, they must first be converted to decimal degrees before being converted to meters. The coordinates can be converted to decimal degrees by using Mathematica's built-in function FromDMS. The formula of finding the decimal degrees, DD is given by Equation (1)

$$DD = degrees + (\frac{minutes}{60}) + (\frac{seconds}{3600}) \quad (1)$$

In other words, 1 decimal degrees is equal to 1° in longitude and latitude forms. For the decimal degrees to be converted to meters, we need to use the conversion factor of 10,000 km per 90 degrees due to the differences of 42 km between the earth's circumference around the equator and that around the poles. In other words, 1 m is approximately

111,111 decimal degrees. Thus, the coordinate points will be in (x_i, y_i) forms in meters metrics units.

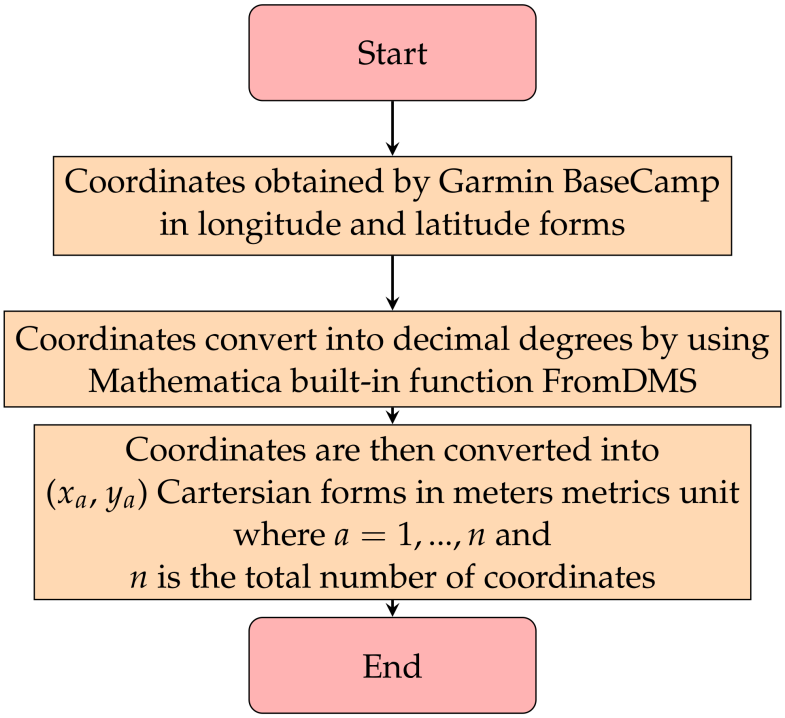


Figure 2. Flowchart for obtaining the coordinates from initial to final points of the routes.

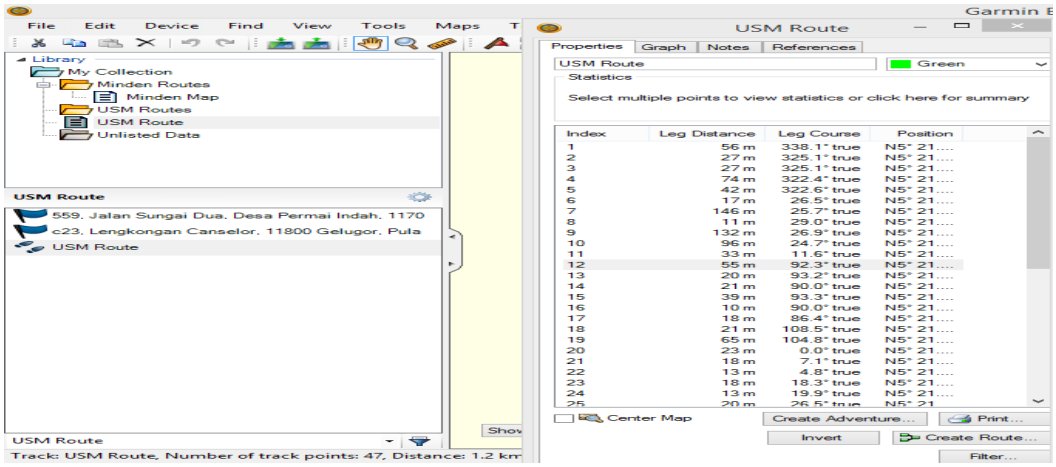


Figure 3. Obtaining the coordinates using Garmin BaseCamp.

3.4.2. Phase B: Calculating the Curvature Values and Radius of Curvature of the Routes

Figure 4 describes the process involved in Phase B. The curvature of the curve can then be derived by fitting the curves to the data. For every three consecutive control points of the curve, the curvature value denoted as K can be approximated for the second data points. The parameter, u, is set to be 0.5. This variable represents the distance traveled by the curve from its starting point as a function for parametric curves with x and y coordinates. The reason for setting u to 0.5 is to minimize the error associated with estimating the curvature value of the routes, as the curve may or may not pass through the second point.

The curvature formula can be used to approximate the curvature for the second point of each curve, where the curvature at any point $M(t) = (x, y)$ is given by Equations (2) and (3)

$$K = \frac{|x'y'' - y'x''|}{\left[(x')^2 + (y')^2\right]^{\frac{3}{2}}} \tag{2}$$

in which the curve is defined in parametric form by the equations

$$x = f(t), y = g(t) \tag{3}$$

The radius of curvature, R, is the radius of an osculating circle that touches one or more curves that has the same tangent and curvature at a given point that can be determined as in Equation (4)

$$R = \frac{1}{K} \tag{4}$$

which also known as the inverse of the curvature K of the curve at point $M(t) = (x, y)$. In other words, R is the inverse of K or K^{-1}.

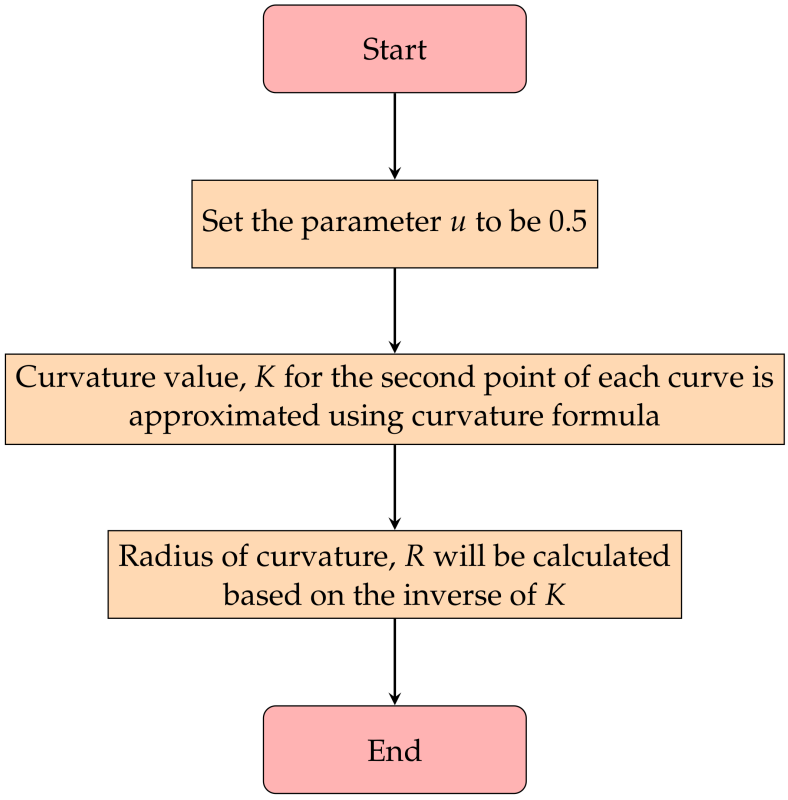


Figure 4. Flowchart for obtaining the curvature values and radius of curvature of the routes.

3.4.3. Phase C: Calculating the Arc Length of the Routes

Figure 5 describes the process involved in Phase C. Arc length is the distance between two points along a section of a curve. The arc length value between each curve determines the distance along the curved line that forms the arc. The arc length value can be obtained using the arc length with parametric equations given by Equations (5) and (6)

$$L = \int_{\alpha}^{\beta} \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} dt \tag{5}$$

for the domain of

$$\alpha \leq t \leq \beta \tag{6}$$

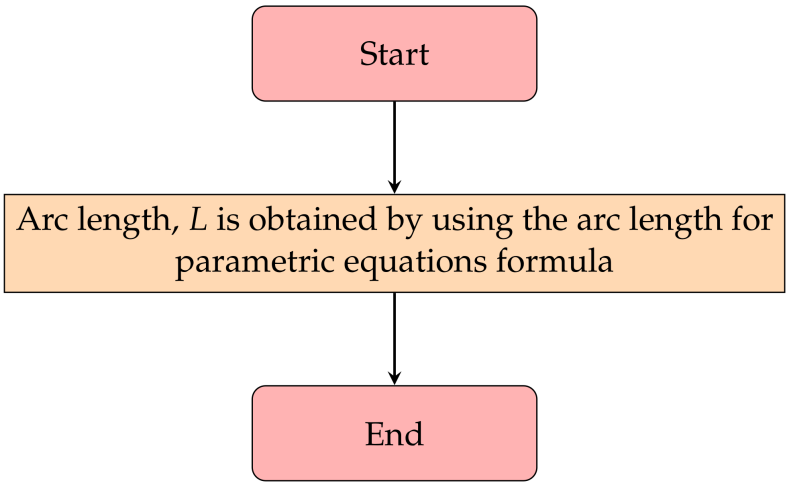


Figure 5. Flowchart for obtaining the arc length of the routes.

3.4.4. Phase D: Calculating a Design Speed That Is Capped by the Average Speed

Figure 6 describes the process involved in Phase D. The minimum radius refers to the curvature value that cannot be exceeded at a given design speed. It depends on the maximum rate of superelevation and the maximum allowable side-friction factor. The minimum safe radius can be determined using the standard curve equation denoted by Equation (7)

$$R_{min} = \frac{V^2}{127(e+f)} \tag{7}$$

where

$R_{min} = minimum\ radius\ of\ circular\ curve(m)$
$V = design\ speed\ (km/h)$
$e = maximum\ super\ elevation\ rate$
$f = maximum\ allowable\ side\ friction\ factor$

Therefore, the design speed in kilometers per hour can be calculated given by Equation (8)

$$V = \sqrt{127R_{min}(e+f)} \tag{8}$$

The maximum superelevation, e, is dependent on weather conditions, type, and the surface of roads. The ideal value of e is 0.06 for plain terrain (flat land) which matches the state of the route used in this study. Hence, the value of e is set to be 0.06.

According to the 1994 American Association of State Highway and Transportation Officials (AASHTO), the comfortable side-friction factor, f of 0.15, is an ideal value of comfort that falls in the speed range of 55–80 km/h. In this paper, 0.15 is chosen as the value for f. Once the design speed has been set, the speed will then be capped at average speed, S, and the new speed will be $v_i = \min\{V_i, S_i\}$ for $i = 1,\ldots,\frac{n}{2}$. Thus, the travel-time estimation can be computed.

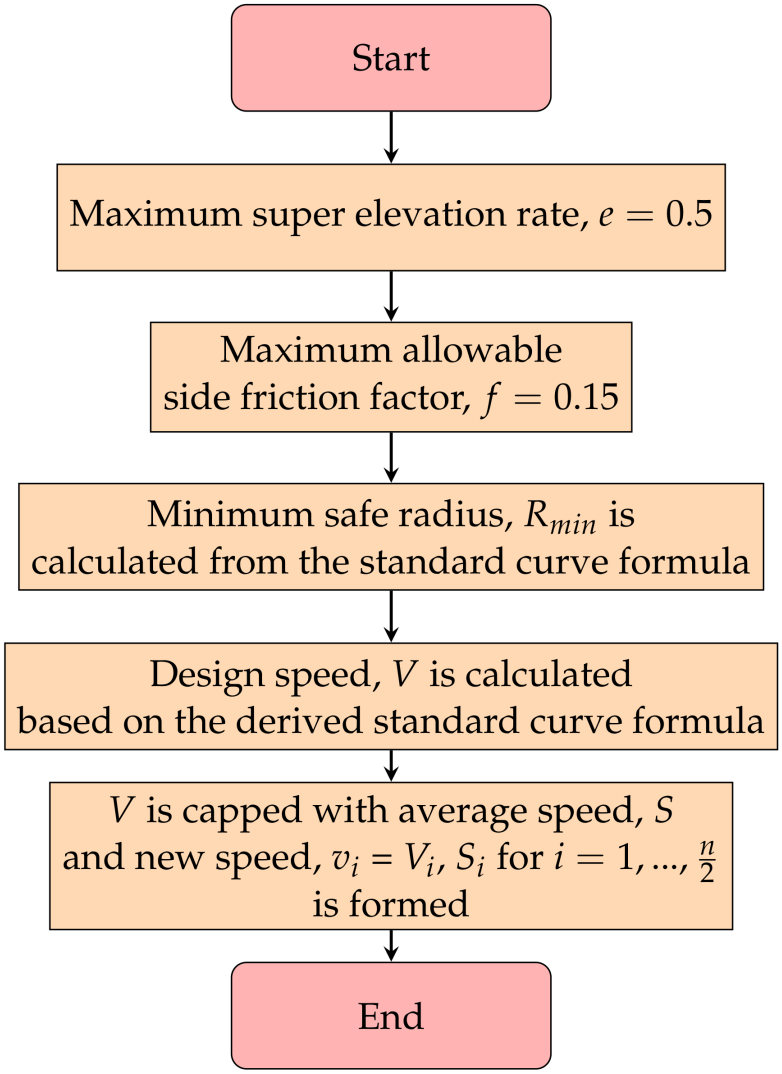


Figure 6. Flowchart for calculating design speed capped by the average speed.

3.4.5. Phase E: Estimation of Travel Time from Initial to Final Points of the Routes

Figure 7 describes the process involved in Phase E. The estimation of travel time is based on the midpoint of arc length of each curve segment, m, and the average speed, S. The midpoint of the arc length of each curve segment of the curve, denoted as m, will be calculated given by Equation (9)

$$m_i = \frac{L_i}{2} \tag{9}$$

where $i = 1, ..., \frac{n}{2}$.

The travel time, t (in seconds), can be calculated for every curve segment as seen in Equation (10)

$$t_i = \frac{3.6(m_i - m_{i-1})}{S_i} \tag{10}$$

where $i = 1, \ldots, \frac{n}{2}$. The travel-time estimation is generated by summing the time, t, for each curve segment given by Equation (11)

$$\sum_{i}^{\frac{n}{2}} \frac{3.6(m_i - m_{i-1})}{S_i} \tag{11}$$

where $i = 1,\ldots,\frac{n}{2}$.

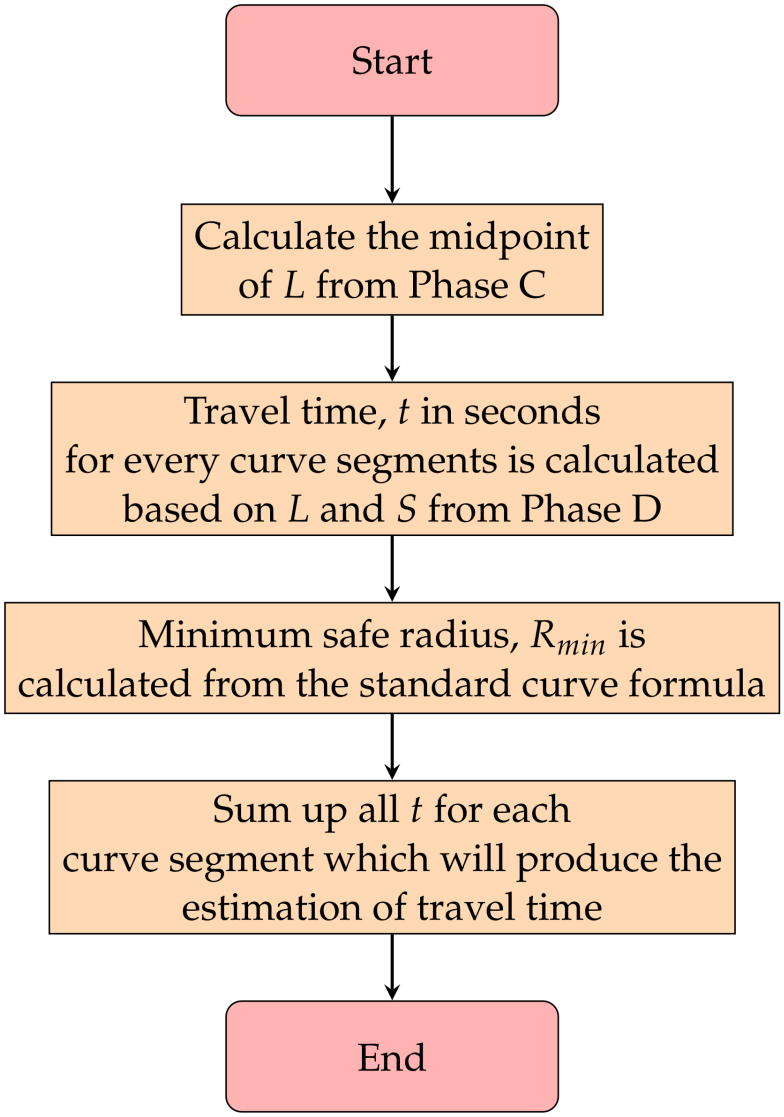


Figure 7. Flowchart for estimating of travel time from initial to final points of the routes.

3.5. Algorithm in Estimating Travel Time Based on Initial Driving Information

The procedure for estimating travel time is shown in Algorithm 2.

Algorithm 2: Estimation travel time

Result: Produce Suitable Estimation Travel Time

1 Set route at the initial points, P_0 and the ending points, P_n
2 Obtain coordinates along the road
3 Fit the data as in Algorithm 1 and obtain $curve_i$ for $\frac{n}{2}$ segment
4 Estimate curvature at each segment k_i, $i = 1,\ldots,\frac{n}{2}$
5 Initial driving information by obtaining average speed, S for first few seconds
6 Calculate speed capped by the average speed, S for $v_i = \min\{V_i, S_i\}$, $i = 1,\ldots,\frac{n}{2}$
7 Compute estimation of travel time, t

3.6. Relative Error

Relative error, R_E, is the ratio of difference between measure value, M_v, with expected value, E_v, to expected value, E_v, which can be written as in Equation (12)

$$R_E = \frac{M_v - E_v}{E_v} \tag{12}$$

This type of error analysis was chosen as one of the indicators for evaluating the precision and performance of the proposed method in this paper. Since this paper's proposed method focuses on travel-time estimation, the units for M_v and E_v will be in minutes.

3.7. Assumptions

There are a few assumptions made in this paper when estimating travel time for the experiment. First, the road must be traffic-free, which means no traffic light along the desired routes. Encountering heavy traffic or traffic lights will undoubtedly affect the result in time estimation of time travel. Second, driving speed is assumed to be almost constant. Inconstant driving speed will affect the estimation of travel time.

4. Results

This paper-proposed algorithm was put to the test in two different locations in Penang for the purpose of estimating travel time. Additionally, driving tests were conducted along the routes of the two sites to determine the accuracy of the travel-time estimation of the proposed algorithm, Google Maps, Waze, as well as previous research that uses the urgent-gentle class (UGC) traffic flow model.

4.1. Case 1: Travel-Time Estimation in Minden

The first Penang map as shown in Figure 8 is the Minden route with an approximate length of 1.68 km. Hermite interpolation curves are used to construct the curve of the route in Minden as shown in Figure 9.



Figure 8. Minden Route from Garmin BaseCamp.

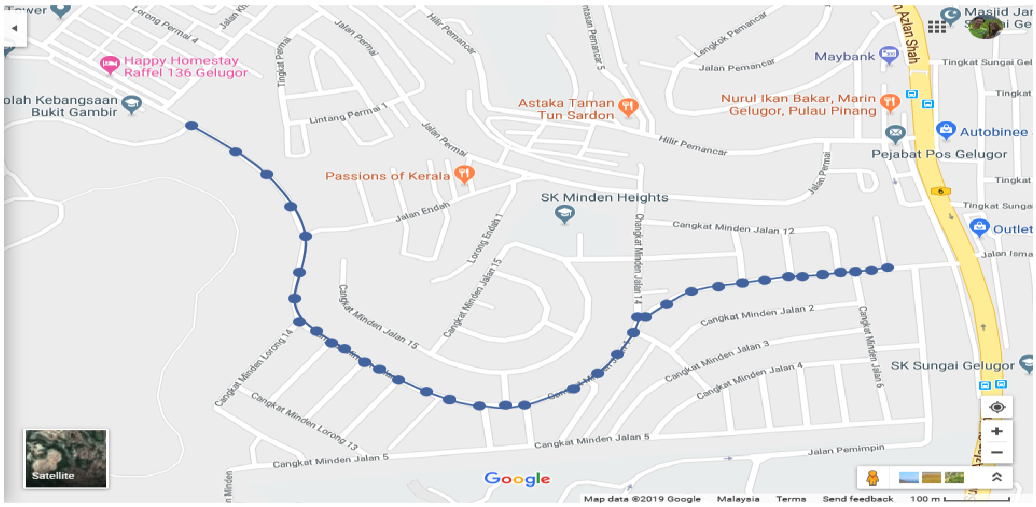


Figure 9. Minden Route Curve Constructed with Hermite Interpolation Curve.

A total of 102 coordinate points were obtained in longitude and latitude forms from Garmin BaseCamp along the desired routes. The obtained coordinates were then fitted into the algorithm to calculate the curvature value and the travel-time estimation. From the 102 coordinates that was picked from the BaseCamp, a total of 51 curves was produced for this paper. Afterwards, by referring to Section 3.4.1, the longitude and latitude coordinates are then converted into decimal degrees and meters. Finally, by referring to Section 3.4.2, to calculate the value of curvature, the curvature value for each 51 curve is obtained (see Table A1). Using the built-in function ListLinePlot in Mathematica, the curvature value graph for each curve can then be plotted as shown in Figure 10.

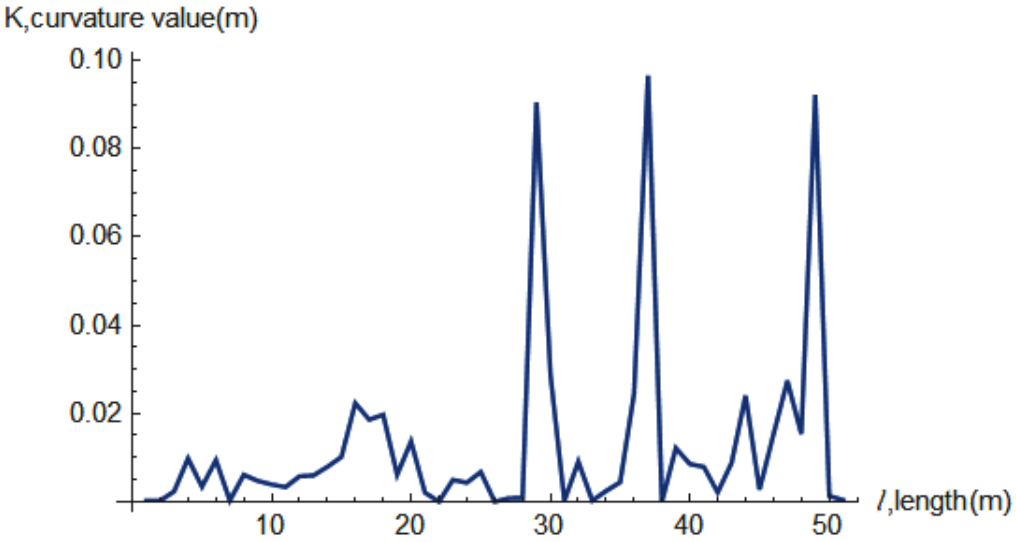


Figure 10. Curvature value graph.

By referring to Section 3.4.3, the value of the arc length for each 51 curve and the radius of curvature can be obtained (see Tables A2 and A3). In addition, by referring to Section 3.4.4, the design speed values (km/h) capped by an average speed of 90 km/h can also be obtained (see Table A4).

The calculated design speed capped by an average speed of 90 km/h is determined to be the reasonable highway speed limit, as shown in Figure 11. As a result, the estimated travel time is approximately 99.1073 s.

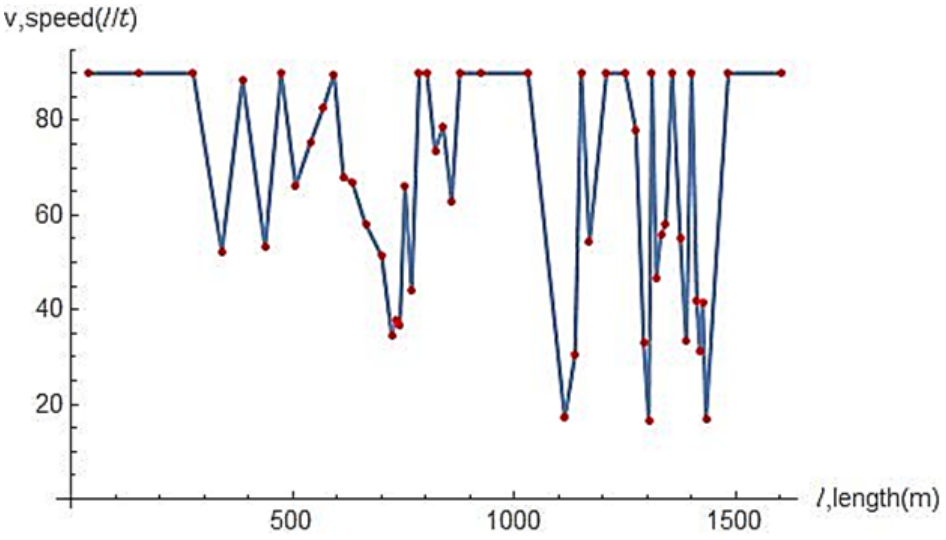


Figure 11. Design speed capped by average speed of 90 km/h

4.2. Case 2: Travel-Time Estimation in Universiti Sains Malaysia

The second Penang's map, as shown in Figure 12 is the Universiti Sains Malaysia (USM) route with an approximate length of 1.18 km. The Hermite interpolation curve is used to construct the curve of the route in USM as shown in Figure 13.

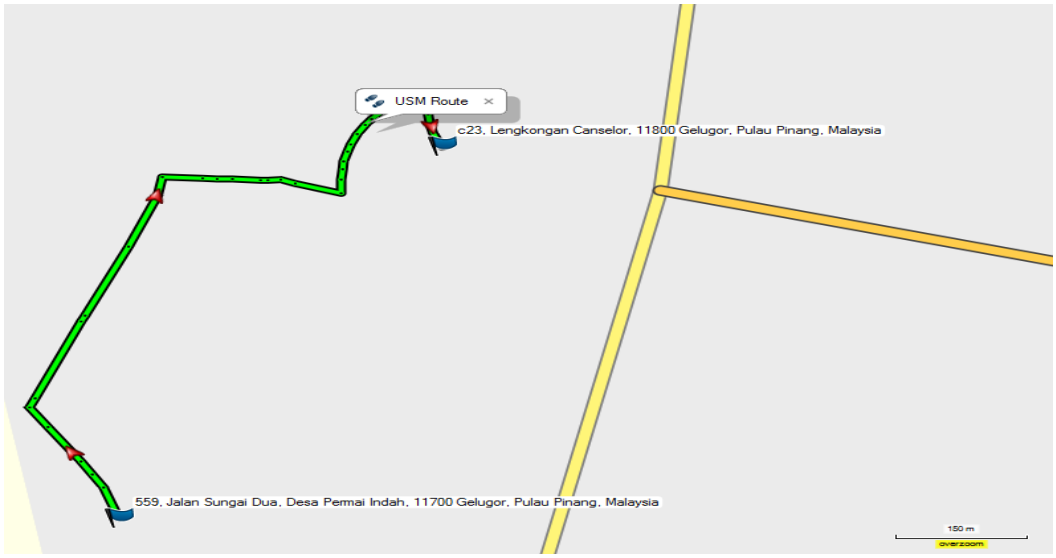


Figure 12. USM Route from Garmin BaseCamp.



Figure 13. USM Route Curve Constructed with Hermite Interpolation Curve.

As in Algorithm 2, Garmin BaseCamp is used to obtain 47 coordinate points in longitude and latitude forms along the desired routes. The obtained coordinates were then fitted into the algorithm to calculate the curvature value and the travel-time estimation. From the 47 coordinates that was picked from the BaseCamp, a total of 23 curves was produced for this paper. Afterwards, by referring to Section 3.4.1, the longitude and latitude coordinates are then converted into decimal degrees and meters. Finally, by referring to Section 3.4.2, to calculate the value of curvature, the curvature value for each 23 curve is obtained (see Table A5). Using the built-in function ListLinePlot in Mathematica, the curvature value graph for each curve can then be plotted as shown in Figure 14.

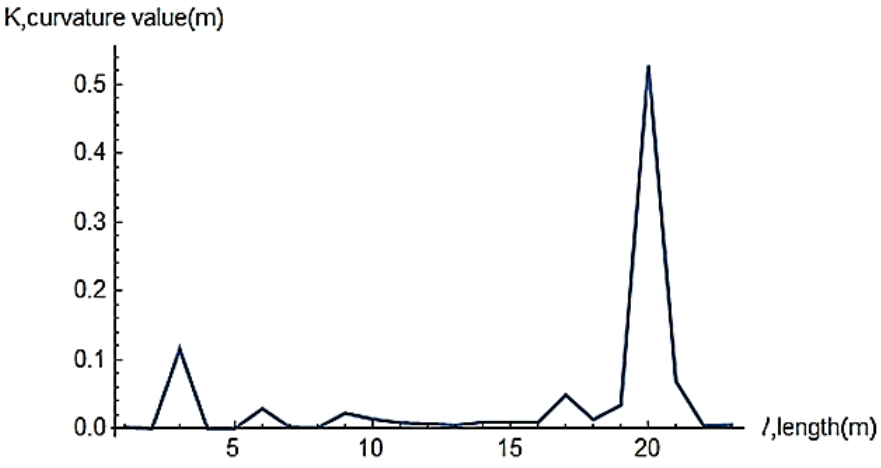


Figure 14. Curvature graph.

By referring to Section 3.4.3, the value of the arc length for each 23 curve can be obtained (see Table A6). The value of the radius of curvature is shown in Table A7. In addition, by referring to Section 3.4.4, the design speed values (km/h) capped by 90 km/h can also be obtained (see Table A8).

The calculated design speed capped by the average speed of 90 km/h is determined to be the reasonable highway speed limit, as shown in Figure 15. As a result, the estimated travel time is approximately 92.8345 s.

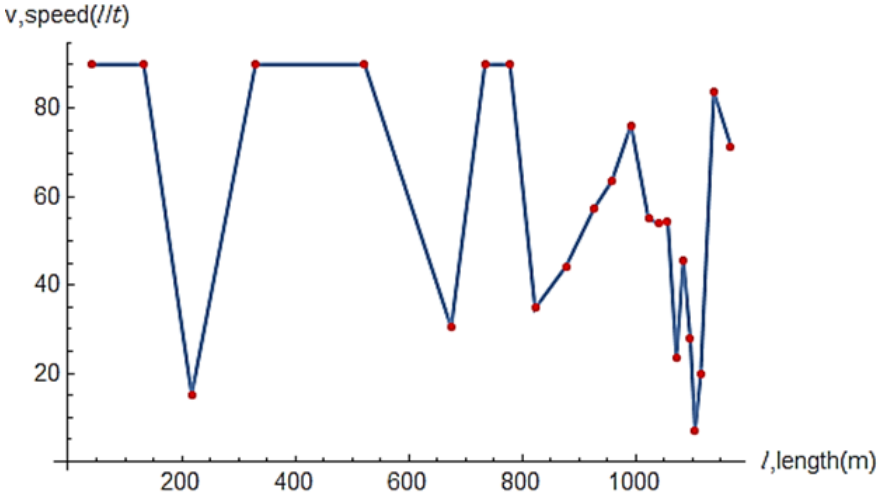


Figure 15. Design speed capped by average speed of 90 km/h.

5. Discussion

A Comparison of the Proposed Work to Google Maps, Waze, Here WeGo, TomTom, and Existing Work

The travel-time estimation of the proposed algorithm is then compared to four GPS navigation applications, namely Google Maps, Waze, Here WeGo, and TomTom, for Minden and USM routes shown in Figures 9 and 13. Additionally, the travel-time estimation of the proposed algorithm is also compared to the UGC traffic flow algorithm by Zhang et al. [21]. According to Kay Ireland, Google Maps estimates the travel time for users based on the distance between two points and the average speed limit between those points [2]. Additionally, Google Maps also considers existing traffic conditions, which often lengthen travel time. For this paper, it is assumed that Waze, Here WeGo, and TomTom use the same algorithm as Google Maps.

The travel-time estimation of the proposed algorithm and the UGC algorithm was compared based on the same average speed and distance covered with the same condition in a free-flow traffic environment. The time shown by the UGC algorithm can be found in Tables 1 and 2.

Table 1. Estimation and Error Travel Time along the USM Route.

Method	Time (Minutes:Seconds)	Error (Minutes:Seconds)	Relative Error (%)
Driving Test (Average speed of 40 km/h)	2:14	-	-
Algorithm 2	2:08	0:06	+4.69
Google Maps	3:00	0:46	−25.56
Waze	3:00	0:46	−25.56
Here WeGo	3:00	0:46	−25.56
TomTom	3:20	1:06	−33
UGC algorithm [21]	1:46	0:28	26.42

In the first driving test on the USM route, the car was driven on 27 March 2019 at 11:30 a.m. The starting speed 40 km/h was taken, and the driving speed was almost constant. As reported in Table 1, the driving journey takes 2 min and 14 s to finish the route. Although this paper-proposed algorithm has not been embedded in any GPS-related application, the algorithm estimates a travel time of 2 min and 8 s. On the other hand, Google Maps, Waze, and Here WeGo have estimated a similar travel time, which is 3 min, and TomTom has estimated 3 min and 20 s. Finally, the UGC algorithm estimated the travel time to be 1 min and 46 s.

In terms of relative error, the proposed algorithm produces +4.69%, whereas Google Maps, Waze, and Here WeGo all produce the same relative error, which is −25.56%. The minus sign indicates that the measured value, M_v, is less than the expected value, E_v. TomTom and UGC algorithms produce relative errors of −33% and +26.42%, respectively. All this information has been tabulated in Table 1. Table 1 shows that the proposed algorithm gives the best performance with the differences of 6 seconds in time error and produces the lowest percentage in relative error (closest to zero).

Table 2. Estimation and Error Travel Time along the Minden Route.

Method	Time (Minutes:Seconds)	Error (Minutes:Seconds)	Relative Error (%)
Driving Test (Average speed of 50 km/h)	2:43	-	-
Algorithm 2	2:38	0:05	+3.16
Google Maps	3:00	0:17	−9.44
Waze	4:00	1:17	−44.17
Here WeGo	4:00	1:17	−44.17
TomTom	3:10	0:27	−14.21
UGC algorithm [21]	2:01	0:42	+34.71

For the second driving test on the Minden route, the car was driven on 17 April 2019 at 10:10 a.m. The starting speed 50 km/h was taken, and the driving speed was almost constant. As reported in Table 2, the driving journey takes 2 min and 43 s to finish the route. Although this paper-proposed algorithm has not been embedded in any GPS-related application, the algorithm estimates a travel time of 2 min and 38 s. Waze and Here WeGo both estimate the travel time to be 4 min, while Google Maps and TomTom estimate the travel time to be close to 3 min. Finally, the UGC algorithm estimates the travel time to be 2 min.

In terms of relative error, the proposed algorithm produces +3.16%, whereas Google Maps, Waze, and Here WeGo produce the same relative error, which is −44.17%. Google Maps, TomTom, and UGC algorithms produce relative errors of −9.44%, −14.21%, and +34.71%, respectively. All this information has been tabulated in Table 2. From Table 2, it can be seen that the proposed algorithm gives the best performance with the differences of 5 s in time error and produces the lowest percentage in relative error (closest to zero).

Based on the first and second driving test results shown in Tables 1 and 2, the proposed algorithm outperforms Google Maps, Waze, Here WeGo, TomTom, and the UGC algorithm for roads with no traffic.

6. Conclusions

This paper has introduced an algorithm for estimating travel time based on the curvature of the road and the initial driving information. A parametric cubic curve, particularly the Hermite interpolation curve, has great potential in estimating the travel time from the initial point to the endpoint through geometric properties. The curve is fitted to each section of the road to obtain its curvature value. Then, by capping the speed with the average speed, the new speed is determined by referring to the minimum value of a set of average speeds and the design speed. Finally, the estimated travel time is calculated based on the speed information.

The proposed algorithm has been tested on two different routes. The results show that the proposed algorithm's travel-time estimation performance is more accurate than Google Maps, Waze, Here WeGo, and TomTom. However, the proposed algorithm contains a few limitations. A small change in coordinates will affect the graph's curvature, given that the curve will not interpolate the entire points. These changes will lead to more significant impacts on the estimation of time travel.

Another limitation is the comparison of the proposed method to the classical baseline in the current literature study. Most algorithms estimate travel time using traffic data instead of this paper's proposed algorithm, which heavily relies on calculation. Therefore, the proposed algorithm's travel-time estimation performance can only be fairly compared to the UGC's recent algorithm due to implementing non-traffic data into the algorithm.

7. Future Works

As a recommendation for future work, by incorporating the proposed algorithm into existing GPS applications such as Waze and Google Maps, the proposed algorithm can provide a more accurate travel-time estimation. Since Waze and Google Maps have recorded a lot of traffic data, taking into account the properties of the road will enhance the reliability of the system. Adopting the proposed method to estimate travel time in rural areas with less traffic information is another way to test the performance of the method. Another potential future area for study is the integration of the current statistical approach that relies on GPS and travel data with the proposed method that emphasizes road geometry. Finally, introducing another method to compute the estimated curvature value instead of using cubic Hermite is another area for future study.

Author Contributions: Conceptualization, O.T., B.I. and A.L.A.R.; Methodology, O.T., B.I. and A.L.A.R.; Formal analysis, O.T. and B.I.; Investigation, O.T., B.I. and A.L.A.R.; Resources, A.L.A.R.; Data curation, O.T. and B.I.; Writing—original draft preparation, O.T., B.I.; Writing—review and editing, A.L.A.R., E.G.M. and C.C.P.Y.; Supervision, A.L.A.R.; Project administration, A.L.A.R. and E.G.M.; Funding acquisition, E.G.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the Ministry of Higher Education Malaysia through Fundamental Research Grant Scheme (FRGS/1/2021/STG06/USM/02/6). The article processing charge (APC) was funded by the Research Management Center (RMC), Universiti Malaysia Sabah, through the UMS/PPI-DPJ1 Journal Article Fund.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors appreciate those who have contributed to the success of this research. The authors also thank the Ministry of Higher Education Malaysia, Universiti Sains Malaysia, and Universiti Malaysia Sabah for supporting this study.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Minden Route

Table A1. Curvature Value.

Curve	Curvature Value (m)
1	0.000247306
2	0.000226624
3	0.00232591
4	0.00979193
5	0.00338909
6	0.00941989
7	0.000257791
8	0.00608296
9	0.00470567
10	0.00388926
11	0.00330489
12	0.00575351
13	0.00595183
14	0.00790284
15	0.0101264
16	0.022331
17	0.0185956
18	0.0197224
19	0.0061094
20	0.0136362
21	0.00204687
22	1.90843×10^{-12}
23	0.00492915
24	0.00429652
25	0.00672408
26	9.2335×10^{-12}
27	0.000761889
28	0.000858011
29	0.0905602
30	0.0287488
31	0.0002
32	0.00906519
33	0.000230414
34	0.00244053
35	0.00439855
36	0.024472
37	0.0965981
38	1.71004×10^{-11}
39	0.0121398
40	0.00853815
41	0.00788897
42	0.00209679
43	0.00880859
44	0.024034
45	0.00273701
46	0.0152398
47	0.0273923
48	0.01536
49	0.0922506
50	0.00129649
51	0.000360787

Table A2. Arc Length Value.

Curve	Arc Length Value (m)
1	74.6517
2	151.649
3	95.9045
4	32.4759
5	61.7388
6	40.098
7	32.3435
8	33.4544
9	33.0263
10	26.3306
11	21.6152
12	20.0441
13	20.7177
14	38.3122
15	35.4048
16	9.46941
17	9.28639
18	9.46621
19	13.0982
20	16.8147
21	17.5695
22	17.5682
23	18.2493
24	19.0753
25	19.0885
26	18.5185
27	78.1603
28	132.932
29	33.1105
30	13.1409
31	14.8148
32	22.3218
33	54.4978
34	29.8663
35	18.6147
36	16.9919
37	5.41516
38	9.44413
39	9.4496
40	11.7171
41	9.97549
42	19.3358
43	17.5912
44	9.29081
45	14.9337
46	8.28672
47	5.87035
48	6.68692
49	7.94336
50	93.115
51	150.694

Table A3. Radius of Curvature Value.

Curve	Radius of Curvature Value (m)
1	4043.57
2	4412.59
3	429.939
4	102.125
5	295.064
6	106.158
7	3879.11
8	164.394
9	212.509
10	257.118
11	302.582
12	173.807
13	168.015
14	126.537
15	98.7516
16	44.7808
17	53.7761
18	50.7037
19	163.682
20	73.334
21	488.551
22	5.23992×10^{11}
23	202.875
24	232.747
25	148.719
26	1.08301×10^{11}
27	1312.53
28	1165.49
29	11.0424
30	34.784
31	5000
32	110.312
33	4340.01
34	409.748
35	227.348
36	40.863
37	10.3522
38	5.84783×10^{10}
39	82.3738
40	117.121
41	126.759
42	476.92
43	113.525
44	41.6077
45	365.362
46	65.6179
47	36.5067
48	65.1042
49	10.84
50	771.315
51	2771.72

Table A4. Design Speed Value.

Curve	Design Speed Value (km/h)
1	328.393
2	343.051
3	107.082
4	52.1888
5	88.7094
6	53.2094
7	321.646
8	66.2147
9	75.2836
10	82.8091
11	89.8324
12	68.084
13	66.9401
14	58.0925
15	51.3196
16	34.5587
17	37.871
18	36.7732
19	66.0712
20	44.2246
21	114.148
22	3.7383×10^6
23	73.5572
24	78.7868
25	62.9789
26	1.69953×10^6
27	187.097
28	176.305
29	17.161
30	30.458
31	365.171
32	54.2404
33	340.218
34	104.537
35	77.8676
36	33.0124
37	16.616
38	1.24885×10^6
39	46.8712
40	55.8894
41	58.1435
42	112.781
43	55.0248
44	33.3118
45	98.7127
46	41.8333
47	31.2031
48	41.6693
49	17.0031
50	143.426
51	271.885

Appendix B. Universiti Sains Malaysia Route

Table A5. Curvature Value.

Curve	Curvature Value (m)
1	0.00143936
2	0.000174761
3	0.115027
4	0.000175984
5	0.000189532
6	0.0287265
7	0.00134099
8	0.000919944
9	0.0220848
10	0.0136478
11	0.00807376
12	0.00661262
13	0.00458866
14	0.00879168
15	0.00914695
16	0.00898123
17	0.0485347
18	0.0128368
19	0.0339538
20	0.527587
21	0.0684043
22	0.00380143
23	0.00522651

Table A6. Arc Length Value.

Curve	Arc Length Value (m)
1	82.1217
2	101.855
3	66.818
4	158.21
5	226.979
6	77.8543
7	40.7627
8	48.3178
9	39.6313
10	68.6561
11	31.7625
12	29.319
13	39.1433
14	22.3346
15	13.3654
16	19.0974
17	11.3484
18	13.1197
19	11.3204
20	5.56887
21	14.3109
22	32.0223
23	28.3249

Table A7. Radius of Curvature Value.

Curve	Radius of Curvature Value (m)
1	694.753
2	5722.09
3	8.69359
4	5682.34
5	5276.15
6	34.811
7	745.718
8	1087.02
9	45.28
10	73.2717
11	123.858
12	151.226
13	217.928
14	113.744
15	109.326
16	111.343
17	20.6038
18	77.9011
19	29.4517
20	1.89542
21	14.619
22	263.059
23	191.332

Table A8. Design Speed Values.

Curve	Design Speed Value (km/h)
1	136.121
2	390.651
3	15.2269
4	389.292
5	375.12
6	30.4698
7	141.026
8	170.267
9	34.7508
10	44.2058
11	57.4743
12	63.5075
13	76.2375
14	55.0777
15	53.9975
16	54.4934
17	23.4415
18	45.581
19	28.0264
20	7.10992
21	19.7456
22	83.7603
23	71.4341

References

1. Trapsilawati, F.; Wijayanto, T.; Jourdy, E.S. Human-computer trust in navigation systems: Google maps vs. Waze. *Commun. Sci. Technol.* **2019**, *4*, 38–43. [CrossRef]
2. Ireland, K. How Does Google Maps Calculate Travel Time? Available online: https://www.techwalla.com/articles/how-does-google-maps-calculate-travel-time (accessed on 12 December 2019).

3. Tang, J.; Hu, J.; Hao, W.; Chen, X.; Qi, Y. Markov Chains based route travel time estimation considering link spatio-temporal correlation. *Phys. Stat. Mech. Its Appl.* **2020**, *545*, 123759. [CrossRef]
4. Mil, S.; Piantanakulchai, M. Modified Bayesian data fusion model for travel time estimation considering spurious data and traffic conditions. *Appl. Soft Comput.* **2018**, *72*, 65–78. [CrossRef]
5. Sanaullah, I.; Quddus, M.; Enoch, M. Developing travel time estimation methods using sparse GPS data. *J. Intell. Transp. Syst.* **2016**, *20*, 532–544. [CrossRef]
6. Farouki, R.T.; Sakkalis, T. Pythagorean hodographs. *IBM J. Res. Dev.* **1990**, *34*, 736–752. [CrossRef]
7. Katsikopoulos, K.V.; Duse-Anthony, Y.; Fisher, D.L.; Duffy, S.A. Risk attitude reversals in drivers' route choice when range of travel time information is provided. *Hum. Factors* **2002**, *44*, 466. [CrossRef]
8. Coolidge, J. The unsatisfactory story of curvature. *Am. Math. Mon.* **1952**, *59*, 375–379. [CrossRef]
9. Barsky, B.A.; DeRose, T.D. Geometric continuity of parametric curves: Constructions of geometrically continuous splines. *IEEE Comput. Graph. Appl.* **1990**, *10*, 60–68. [CrossRef]
10. Milnor, J.W. On the total curvature of knots. *Ann. Math.* **1950**, 248–257. [CrossRef]
11. Fitzpatrick, K.; Krammes, R.A.; Fambro, D.B. Design speed, operating speed and posted speed relationships. *Inst. Transp. Eng. ITE J.* **1997**, *67*, 52.
12. Barnett, J.; Haile, E.; Moyer, R. Safe side friction factors and superelevation design. In Proceedings of the Highway Research Board Proceedings, Washington, DC, USA, 18–20 November 1936; Volume 16.
13. Leisch, J.E.; Leisch, J.P. New concepts in design-speed application. *Transp. Res. Rec.* **1977**, *631*, 4–14.
14. Misro, M.; Ramli, A.; Ali, J. Approximating maximum speed on road from curvature information of Bézier curve. *Int. J. Math. Comput. Phys. Electr. Comput. Eng.* **2015**, *9*, 705–712.
15. Liu, S.Y.; Ding, Y.J.; Chen, R. Estimating Travel Time. U.S. Patent 8,712,682, 2014.
16. Wang, Y.; Zheng, Y.; Xue, Y. Travel time estimation of a path using sparse trajectories. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; ACM: New York, NY, USA, 2014; pp. 25–34.
17. Lee, W.H.; Hwang, K.P.; Wu, W.B. An intersection-to-intersection travel time estimation and route suggestion approach using vehicular ad-hoc network. *Ad Hoc Net.* **2016**, *43*, 71–81. [CrossRef]
18. Gentili, M.; Mirchandani, P.B. Review of optimal sensor location models for travel time estimation. *Transp. Res. Part Emerg. Technol.* **2018**, *90*, 74–96. [CrossRef]
19. Chen, C.H.; Chang, H.C.; Su, C.Y.; Lo, C.C.; Lin, H.F. Traffic speed estimation based on normal location updates and call arrivals from cellular networks. *Simul. Model. Pract. Theory* **2013**, *35*, 26–33. [CrossRef]
20. Durán-Hormazábal, E.; Tirachini, A. Estimation of travel time variability for cars, buses, metro and door-to-door public transport trips in Santiago, Chile. *Res. Transp. Econ.* **2016**, *59*, 26–39. [CrossRef]
21. Zhang, Y.; Smirnova, M.; Bogdanova, A.; Zhu, Z.; Smirnov, N. Travel time estimation by urgent-gentle class traffic flow model. *Transp. Res. Part Methodol.* **2018**, *113*, 121–142. [CrossRef]
22. San Filippo, S.; Tayyeb, A.; Murray, S.; Cheng, K.; Lebrun, J. Estimating Times to Leave and to Travel. U.S. Patent 9,377,319, 2016.
23. Nadeem, T.M.; Miller, D.D. Estimation of Travel Times Using Bluetooth. U.S. Patent 8,519,868, 2013.
24. Bertsimas, D.; Delarue, A.; Jaillet, P.; Martin, S. Travel time estimation in the age of big data. *Oper. Res.* **2019**, *67*, 498–515. [CrossRef]
25. Jenelius, E.; Koutsopoulos, H.N. Travel time estimation for urban road networks using low frequency probe vehicle data. *Transp. Res. Part Methodol.* **2013**, *53*, 64–81. [CrossRef]
26. Rahmani, M.; Koutsopoulos, H.N.; Jenelius, E. Travel time estimation from sparse floating car data with consistent path inference: A fixed point approach. *Transp. Res. Part Emerg. Technol.* **2017**, *85*, 628–643. [CrossRef]