Next Generation of Journey Planner in a Smart City

Liang Yu, Dongxu Shao, Huayu Wu
Institute for Infocomm Research, A*STAR
1 Fusionopolis Way, Singapore 138632
Email: {yul, shaod, huwu}@i2r.a-star.edu.sg

***Abstract*—Journey planning is the key to an efficient and sustainable transportation system in a smart city. A good journey planner is expected to help commuters travel safely, comfortably and quickly, as well as keep the whole transportation network running efficiently. In modern cities, it should be able to combine a wide range of private and public transport modes, and more importantly, react to real-time events that are impactful on the topology of the transport network. In this paper, we present our multi-modal journey planner, *JPlanner* developed for the city of Singapore. *JPlanner* leverages on more comprehensive urban data, i.e., traffic network data and real-time traffic speed data, aiming to provide more accurate and effective recommendations to commuters. With respect to functionality, *JPlanner* supports the combination of multiple transport modes, such as "Park and Ride" for the switch between private car driving and public transport riding. Other travel modes supported by *JPlanner* include walking, cycling and taxi. We highlight that the key technology enabling the accurate journey planning in *JPlanner* is the Speed Fusion, which infers real-time traffic speed by fusing different data sources. Finally we use a case study to compare the journey recommendation results between *JPlanner* and the other two popular journey planners to demonstrate the advantages of our system..**

I. INTRODUCTION

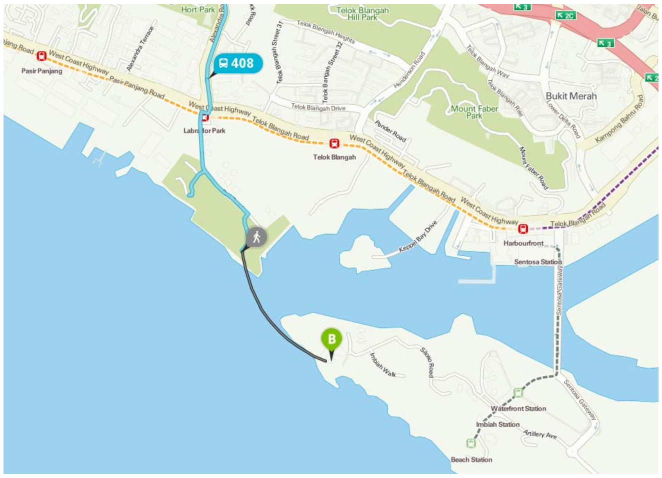
The efficiency of the transportation system in a city significantly affects the productivity of the citizens, and determines the efficiency of the city. How to improve the services in public and private transportation is always an important topic for city planners, even before people talk about smart city. Due to the trend of urbanization, most cities are getting more and more crowded. Private vehicles are no longer the most convenient transportation tool in a city, because of the increasingly serious traffic problems.

On the other hand, public transportation, e.g., metro, provides more guarantees of traveling time and can be even faster during peak hours. However, the fixed locations of public transport stations may be a headache for citizens who are not living nearby. As a consequence, people start thinking about multi-modal transportation for traveling. For example, a traveler may choose to use walking, cycling, taxi or her own vehicle for the first-mile and last-mile travels (i.e., to and from metro stations), and take the metro between the two stations.

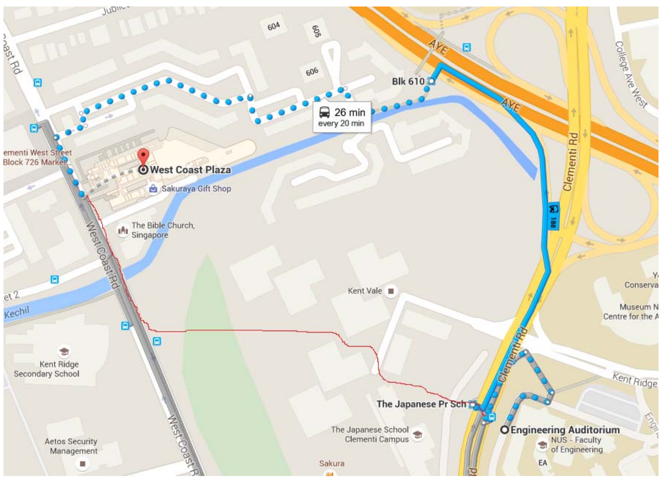
To facilitate multi-modal traveling in cities, journey planning services are extremely important. For example, if a traveler decides to drive to a metro station from home and then take the metro, a journey planner should suggest a metro station with a car park nearby, instead of simply suggesting the nearest metro station. There are a number of service providers consistently providing journey planning services. In Singapore, two popular sites are Google Maps[1] and Gothere[2]. However, the support of multiple travelling modes is weak in most existing services. Take Google Maps and Gothere as an example. The only multi-modal suggestion that can be made by these service providers is the mixture of walking and public transport tools. Nevertheless, the accuracy of the support of these two traveling modes is still not satisfiable. We list down two major problems of the existing journey planning services, including Google Maps and Gothere.

- *Problem 1: lack of accurate network information*: The public transportation network and schedules in a city are rather static and usually made available to the public. Most service providers could accurately provide recommendation of public transport travels. However, for most point-to-point queries, one challenge is to find out best first-mile (i.e., walking from the origin point to a public transport station) and last-mile (i.e., walking from a public transport station to the destination point) routes and combine them with the public transport recommendations. The attempt used in most existing service providers is to use nearest stations, but the calculation of walking distance is less satisfied due to the lack of network information of walking paths. Fig. 1(a) is the route recommendation returned by Gothere. We can clearly see that the last-mile walking path is unrealistic. Google Maps can make correct recommendation for this query. However, for another query as shown in Fig. 1(b), Google Maps suggests a route with combination of walking and bus (in blue), but the walking path in red which travels through a residential area and a park will definitely be a faster and better choice and it is used by most students from the university. Since Google Maps does not have this park connector network, this recommendation cannot be made.
- *Problem 2: lack of real-time speed information*: For most point-to-point queries, there are multiple candidate routes can be chosen. Every journey planning service provider makes recommendation by choosing the best route according to travel time or any other metrics. Estimation of travel time is rather important for journey planning. Most existing service providers use static speed information on different roads and the travel distance to estimate travel time. Although some of them take different time periods (i.e., peak hours and off-peak hours) into consideration during road speed estimation, the result is still static and cannot de-

[1]https://maps.google.com
[2]http://gothere.sg



(a) Gothere example



(b) Google Maps example

Fig. 1. Examples from Gothere and Google Maps

tect the change of traffic conditions timely or accidents happened which can seriously affect the travel time. In a city situation, sometimes the difference between possible routes are not very significant. Without an accurate travel time estimation, the recommendation made may not be optimal.

In this paper, we present our work on leveraging richer data for real-time multi-modal journey planning. Compared to the existing services, our system, *JPlanner*, collects, fuses and uses broader network data and real-time traffic sensor data for better route recommendation. Due to this significant difference, we consider our system as the new generation of journey planning system. We focus on developing and deploying our system in the city of Singapore. In particular, the first contribution is that we incorporate more network data to enable the multi-modal journey planning. On top of the road network data, we also make use of park connector data with walking paths, cycle park and car park location and occupancy data, and traffic regulation data. These data make the combination of multiple travel modes, i.e., public transport (metro and bus), private car, bicycle and walking, possible during journey planning.

The second contribution of *JPlanner* is to use multiple sensor data for real-time road speed estimation. There are two speed data sources: static speed cameras which are mounted at certain points on the roads, and probing vehicles which roam in the city and report their location and speed every a few minutes. In Singapore, the probing vehicles are basically taxis mounted with GPS devices. We design algorithms to fuse the speed information from different sources and build models to estimate and report real-time speed in every road. We will demonstrate that using such real-time speed information, our system can provide more accurate journey planning services.

This project was developed based the Open Trip Planner (OTP)[3] which provides ready-to-use generic data models and algorithms so that we can focus on the implementation of new features. The rest of the paper is organized as follows. In Section II we introduce the general architecture and some functions of the *JPlanner* system. Section III focuses on the technology we used in speed fusion to get more accurate real-time speed information. Section IV demonstrates the *JPlanner* system and compares it with two popular journey planning services for a few use cases. We revisit some related work in Section V and finally conclude this paper in Section VI.

II. Overview

This section depicts the general architecture and functionalities of our system *JPlanner*. Fig. 2 describes the conceptual architecture of the *JPlanner* system.

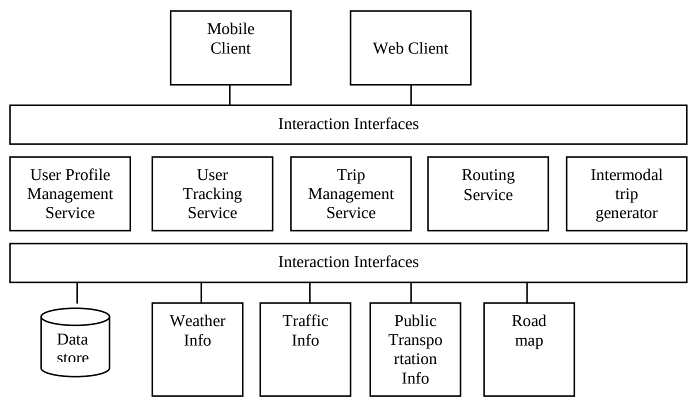


Fig. 2. Conceptual architecture of JPlanner

At the bottom, the system collects and calculates information from different sources, including road map information, public transportation information, traffic information and weather information. In particular, some traffic information is directly collected from corresponding authorities, while some is calculated from data. As mentioned, one key feature of the *JPlanner* is to use real-time traffic speed information for planning. This information is calculated by a process called Speed Fusion, from different sensor data and probe vehicle data. The details will be discussed later.

After getting different types of information, the main job of the system is to generate inter-modal trips and compute scores for final recommendation purpose. Besides this core module, the system also supports administrative operations, user recording and tracking functions and user profile management. All

[3]http://www.opentripplanner.org/

these functions will be reflected in a mobile app and a web portal for clients to use.

The user profile management and user recoding and tracking aim to know better about frequent users. On one hand, from the feedbacks from the users, we can improve the searching algorithms to make the recommendations more reasonable to most citizens. On the other hand, we can know each individual's frequent locations and provide better customized services based on some analytic methods. The user analysis module is not yet done in the current version of *JPlanner*.

Especially to be mentioned is the administrative operation support. Due to the high dynamics of urban mobility, some traffic incidents can hardly be predicted. Even if we have real-time data streaming in, in some cases, such data are still not capable to capture all the incidents. The administrative functions allow the transportation authority to manually add in incidents or connect it to an incident reporting database, so that *JPlanner* will take these incidents into account during journey planning. Fig. 3 shows an example when an administrator interrupts the system planning. The first screenshot is the current recommendation for a travel, i.e., taking Bus 188 to the destination. If there is an incident just happened in a place called Haw Par Villa along Bus 188's route, and caused the delay of 1 hour for all the vehicles on this road, the administrator will insert this incident to the system. Then the system will take it into account and recommend another route that avoids the road near Haw Par Villa, as shown in the second screen shot. In the last screenshot, it shows that even though we know Bus 188 is affected by the incident near Haw Par Villa, the system will still recommend this bus if the traveler's destination is before Haw Par Villa.

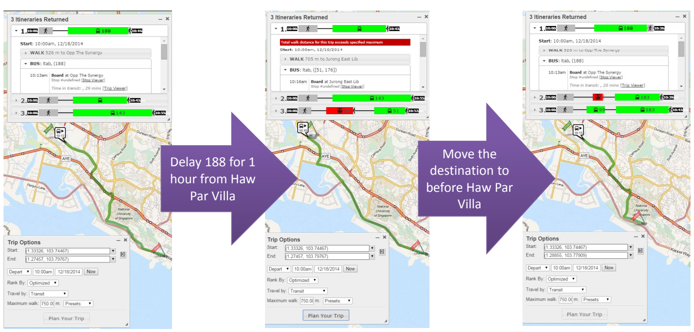


Fig. 3. Planning with an incident

By using a mobile app, if a user has already planned a journey but not started yet, the system would evaluate the impact of a new reported incident on the journey. If it is likely to affect the planned journey, the user will be notified and given an option for re-routing (see Fig. 4).

III. Speed Fusion

The traffic condition changes all the time in Singapore. For a journey that takes a long enough time, it is necessary to consider these changes happening while the driver is moving. The travel time for each single road traversing is computed by the predicted speed at the time the driver is supposed to arrive, rather than only the speed at the starting time of the journey. To achieve this, we need to build a predictor from the historic

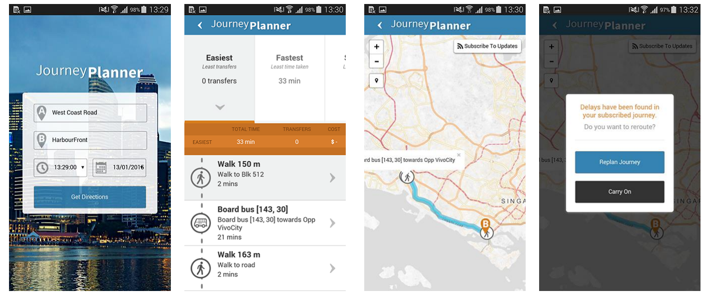


Fig. 4. User notification in case of an incident

data. In Singapore, the main methods of monitoring the road speed are stationary speed cameras and probing vehicles. The coverage and reliability of a single source may not be good enough. Thus, we developed a fusion method to incorporate both datasets to estimate the road link level speed.

The limitation of fusing multiple observation for speed estimation stems from the confidence of the raw observations within a certain context. Fig. 5 shows a simple example where the readings from two speed cameras and one vehicle report are used to estimate the average speed of the road segment highlighted in red. It can be seen that each observation only partially overlaps with the target space/time range for which the estimation should be given. Understanding how and where the data were generated and processed can help us assess the confidence of each individual observation, which can improve various algorithms if being taken into consideration. The uncertainties are not fixed for specific data sources, instead, it depends on the context where the data are produced in a case-by-case manner.

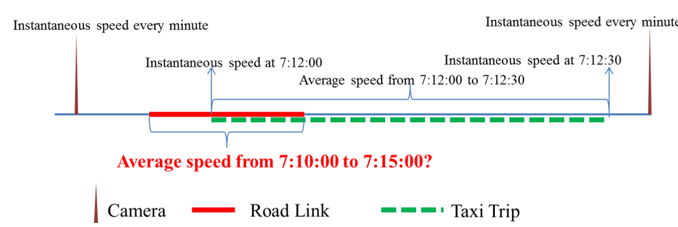


Fig. 5. A speed data fusion example

In our work, we evaluate the confidence on different observations from different sources, and apply such information to speed fusion.

A. Speed Interpolation

The observations in Fig. 5 are not exactly representative for the target road segment in a specific period. Also, those values are discrete rather than continuous, such that it introduces uncertainties when used to represent the average speed for a continuous space/time region. Our solution is to interpolate a continuous speed/time curve so as to compute the expected speed values by solving the curve function with some parameters. There are two principles for speed interpolation:

1) To minimize the overall error. In [1], the authors have proved the quadratic interpolation outperforms other less complicated methods. We adopted a more

generic polynomial based approach and tried to use different combinations of data sources for comparison and evaluation.

2) To make physical sense. The interpolated result should match the distance/time ground truth if provided, e.g., the overall travel distance, the on-spot instantaneous speed, the overall travel time, etc.

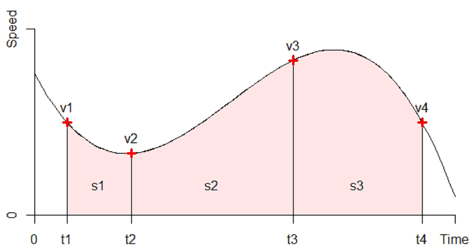


Fig. 6. Lagrange interpolation

Polynomial based interpolation is the most widely used method in this area. According to the Lagrange interpolation formula, the degree of the polynomial is *n*-1 given *n* is the number of input points, e.g., a straight line is interpolated for only two points. Fig. 6 depicts an example of using readings from four speed cameras with fixed locations. Given t_i and v_i (*i*=1, ..., *n*) are known, the simulation function can be easily computed using the Lagrange formula:

$$P(x) = \sum_{j=1}^{n} P_j(x) \tag{1}$$

$$P_j(x) = y_i \prod_{k=1, k \neq j}^{n} \frac{x - x_k}{x_j - x_k} \tag{2}$$

However, for the camera case, only the distances between the cameras are known, i.e., s_1,..., s_n, which are the red areas under the curve in Fig. 5. Using the integral form of the polynomial we can solve the t_1,..., t_n, i.e.,

$$S_i = \int_{t_i}^{t_{i+1}} f(t)dt \tag{3}$$

For the probing vehicle case, to ensure that both the travel time and distance are met by the polynomial, its degree needs to be increased by 1. Fig. 7 shows the possible forms of such interpolation where the interpolation of two points could be a quadratic function: (a) a constant acceleration case which is very rare in reality; (b) a case that a driver tends to drive at low speed; (c) a case that a driver tends to drive at high speed such that the travel distance is longer than others.

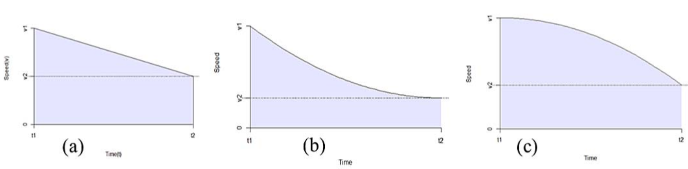


Fig. 7. Speed trajectory of probing vehicles

B. Evaluating the Confidence of Observations

Most existing methods assume the confidences for all observations are constant, which may not be true in reality. As mentioned, the novelty of our work is to evaluate different confidence values for different observations. As can be seen from Fig. 5 that there are some geometric correlations between the measured objects and the target road segment, which can be used as evidence for computing the confidences.

1) Confidence for Camera Interpolation: Geostatistic interpolation methodologies regard that the correlation between a known point and unknown point will decrease when the distance between them increases. For example, Kriging interpolation [2] uses the covariance as the indicator and a function to approximate the relation between the covariance and the distance. According to this function, the covariance value will reach a limit when the distance increases, which means the two points are totally irrelevant. We assume the same rationale holds when we interpolate the speed using polynomial, i.e., the confidence is approximately inversely proportional to the distance between the known point and the estimating point.

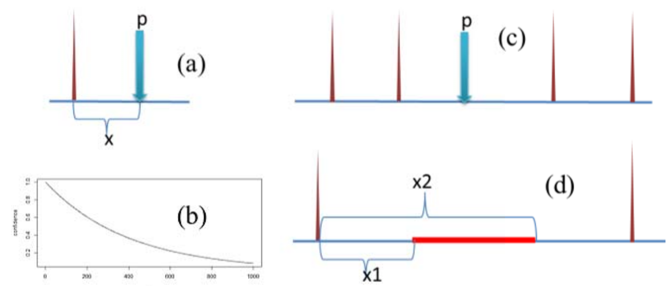


Fig. 8. Confidence model

As depicted in Fig. 8(a), the confidence of the estimation on a single point *i* can be computed by an exponential function as shown in Fig. 8(b):

$$f^i(x) = e^{-x/\alpha} \tag{4}$$

The parameter α can be decided by the covariance study on existing data in a similar way to Kriging. When multiple known values are used for interpolation, as depicted in Fig. 8(c), the confidence is the weighted aggregation of all single confidence values. The weightage value is computed by the same Inverse Distance Weighting (IDW) method.

$$f(x) = \sum w_i f^i(x), where \sum w_i = 1 \tag{5}$$

To compute the confidence of the interpolation on a continuous road segment as shown in Fig. 8(d), an integral result from the single-point confidence function is used.

$$\overline{f(x)}_{x_1}^{x_2} = \frac{\int_{x_1}^{x_2} f^*(x)dx}{x_2 - x_1} \tag{6}$$

According to this equation, once the point detection devices are set up at fixed locations, the confidence of the estimation for a specific road link is determined.

2) Confidence for Probing Vehicles Interpolation: The probing vehicle data has at least two sources of confidence:

1) Interpolation confidence. It is the same with the confidence for the point detection data. Let us denote it as C_I.
2) Coverage confidence. The coverage confidence can be measured by the ratios of the overlapped length respectively to the lengths of the road segment and the trajectory, denoted as O_1 and O_2.

The final confidence of a vehicle's trajectory is then the product of the three variables, i.e., C_I, O_1 and O_2. In our study, the reporting frequency for the probing vehicle data is about 1-3 minutes. If the traffic condition is good, a vehicle can easily travel several kilometers within 3 minutes. Assuming a road segment a car travels on is only 200 meters, the confidence of the estimation is then very low. Another very rare but ideal case is that if a car reports exactly at the two end points of a road segment, the confidence is very high.

There are some other uncertainties which are not touched in this paper but worthy of mentioning here. For example, an actual data fusion process might involve other steps such as map-matching, which can introduce some uncertainties as investigated in [3], [4]. If we are not sure whether a car has really passed a road, the confidence should be decreased. For the point detection data, if there is a traffic light between two speed cameras, the confidence will also decrease.

C. Speed Integration

For a certain road segment, the fused speed under each type of source (i.e., point detector and probe vehicle) can be computed by taking consideration of both the interpolated speed values and confidence values, as described above. The final step is to integrate the speed values from these source types.

In our work, we adopt a simple linear fusion method to integrate speed values. Suppose $\hat{v}_p$ is the average speed calculated from point detectors, and $\hat{v}_t$ is the average speed calculated from probe vehicles. Two confidence values for the two types of data sources are represented by f_p and f_t respectively. Then the final average speed fused from the two values is:

$$\hat{v} = \alpha_1 f_p \hat{v}_p + \alpha_2 f_t \hat{v}_t \tag{7}$$

Note that α_1 and α_2 are used to control the importance of each type of source.

D. Experiment

We selected a segment of arterial road in Singapore for the experiment. Fig. 9 shows the different views of that area. The road is bidirectional but our focus is the left side, i.e., from south to north. The speed cameras on the left side are named (the last two digits) with even numbers in a descending order. The target road segment for this experiment is highlighted with the rectangle in the "Road Link" view. The "taxi trips" view shows the starting points (blue) and ending points (red) of each taxi trip that overlaps with the target link.

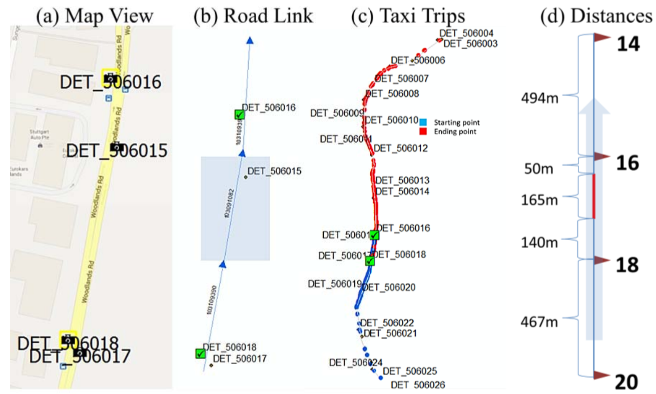


Fig. 9. Selected area for the experiment

Four nearest cameras are used, i.e., numbered with 20, 18, 16 and 14. The distances between the cameras and the link are detailed in Fig. 9(d). Note that the red line is the target link which is only 165 meters. To compare the difference introduced by using different numbers of cameras, they are grouped into 4 sets: two nearest cameras (18 and 16), first 3 cameras (20, 18, 16), last 3 cameras (18, 16, 14), and all 4 cameras.

The detailed interpolation polynomials for speed cameras within the 9:35 - 9:40 window are shown in Fig. 10. The red areas are the periods when a vehicle is driving on the target road segment according to the interpolated polynomials.

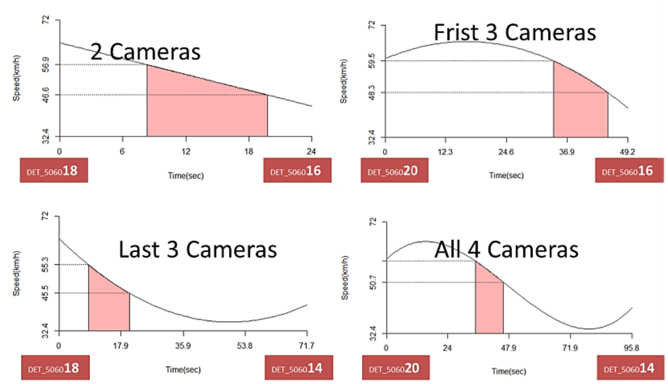


Fig. 10. Interpolation for the speed camera data

On the other hand, there are 5 taxi trips overlapping with the target road segment between 9:30 and 10:05. Fig. 11 shows the interpolation result for 4 of them. For trip 1 and 3 the drivers tended to drive at low speed, while in 2 and 4 at high speed. According to the curves, the average speeds for the target road segment (red areas) are very different from those of the whole trips.

Finally, we use Formula 7 to integrate the speeds returned from speed cameras and taxis. Although the computed confidence values of taxi are very low, they are still more trustful than the camera data because the travel time is measured while the camera is not able to trace a vehicle. We set the ratio between α_1 and α_2 to 1/4 in this case.

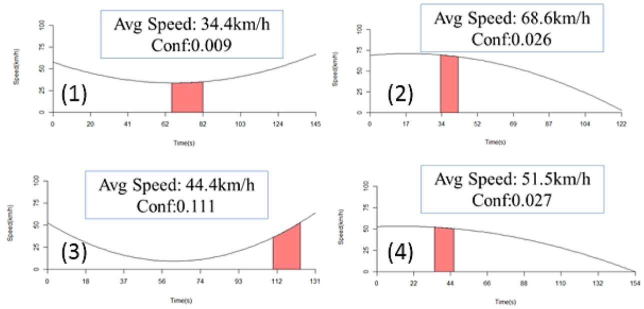


Fig. 11. Interpolation for the probing vehicle data

IV. JPLANNER DEMONSTRATION

In this section, we demonstrate the key features provided by the *JPlanner* system, with use cases. Note that in the client app, as well as all the demonstration screenshots, we use the colors of purple, red, green, blue and grey to represent different travel modes of driving and taxi, metro, bus, cycling and walking respectively.

A. Single-modal Planning

In this section, we showcase how *JPlanner* does single-modal route recommendation. As mentioned, *JPlanner* use more accurate real-time speed information, which makes recommendation quite dynamic by considering the traffic conditions.

The first example is about driving route recommendation. In Singapore, during peak hours and off-peak hours, the traffic conditions on different roads are quite different. Fig. 12 shows an example that at different hours the system recommends different routes for the same pair of origin and destination, due to the different traffic conditions on different roads at different times. Actually, during different time period, the road speeds on highways and normal roads may vary significantly. The optimal routes based on shortest travel time should also be different.

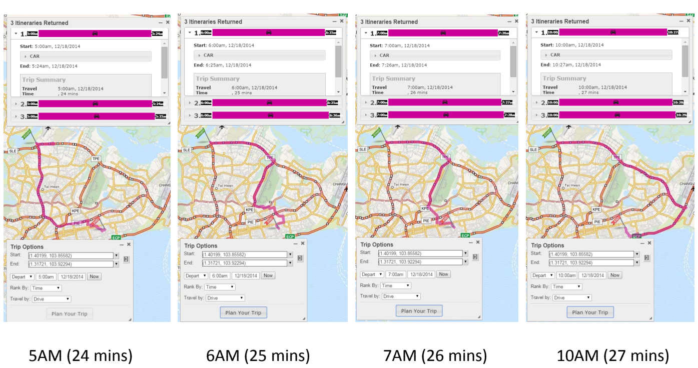


Fig. 12. Time-dependent driving route recommendation

Furthermore, there are many parks in Singapore, where people prefer cycling and jogging inside. Since we also collect park connector network data, we can recommend paths in/through parks. Fig. 13 shows an example that the *JPlanner* recommends a route that travels through some parks for a bicycle traveler. *JPlanner* appreciates safety more than any other factors for bicycle riders. As a result, the park connector is always prioritized as long as it is available, compared to other roads with vehicles.

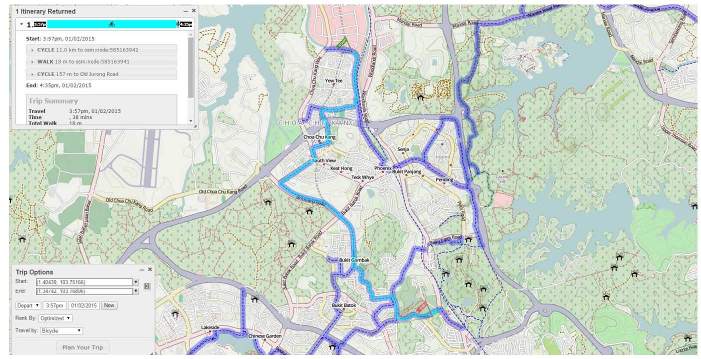


Fig. 13. Cycling route recommendation

B. Multi-modal Planning

The first multi-modal journey planning case we would like to demonstrate is the combination of metro and taxi. Actually this is a very common mixed travel mode for citizens in Singapore. During peak hours, it is difficult to get a taxi from suburban areas to city center, because there are a lot of taxi demands from the suburban residential areas. Even if one is able to book a taxi, the traffic speed on the major roads from suburban to city is quite slow. Thus the metro is the best choice during peak hours. Once arriving in the city area, one can easily find many available taxis that just drop off passengers in city. If her office is out of walking distance from the metro station, she can easily flag down a taxi from the metro station in city. In another situation, when the travel distance is long and the traveler is not rushing, due to cost concern, many people may choose metro. If the start point or end point is not near a metro station, she may need a taxi as the first-mile or last-mile transportation tool.

Fig. 14 shows how *JPlanner* recommends routes by combining taxi and metro. In these cases, the start location (first screenshot) or the destination (second screenshot) or both (third screenshot) are far away from any metro station, and there is no convenient bus services as well. In this case, the system will intelligently suggest combined route by taxi and metro.

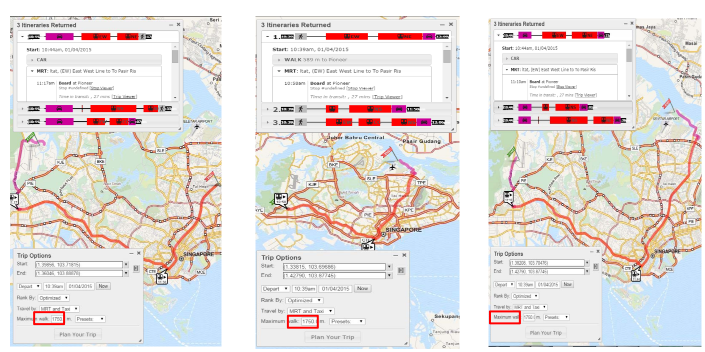


Fig. 14. Multi-modal route recommendation with metro and taxi

Note that the system considers the taxis' availability when suggesting a taxi-transit switch. Currently it is indicated by the taxi stands around the stops/stations. In Singapore, all train stations have taxi stands inside or nearby, while bus stops might not. The taxi availability could also incorporate more factors such as the number of free taxis around and the street hail restriction. Although not implemented yet due to the lack of related data, such customization would be easy under the system framework so that we leave it to the future work.

The second use case is about park and ride. In this case, a traveler may choose to drive to a certain car park, park her car, and take the public transit to the final destination. When *JPlanner* does this recommendation, it not only considers the traffic conditions, but more importantly it needs to consider the parking facilities of different metro stations and bus stops. The system collects the information of all the car parks across the Singapore island, and incorporates this information into journey planning. Fig. 15 shows the interface that a user may choose which car park is of her convenience for park and ride, given she has a season parking package bind to some car parks. Of course, if the user does not mind which car park she can use, the system will choose the best parking location for her considering a set of factors such as walking distance, overall travel time, car park availability, etc.

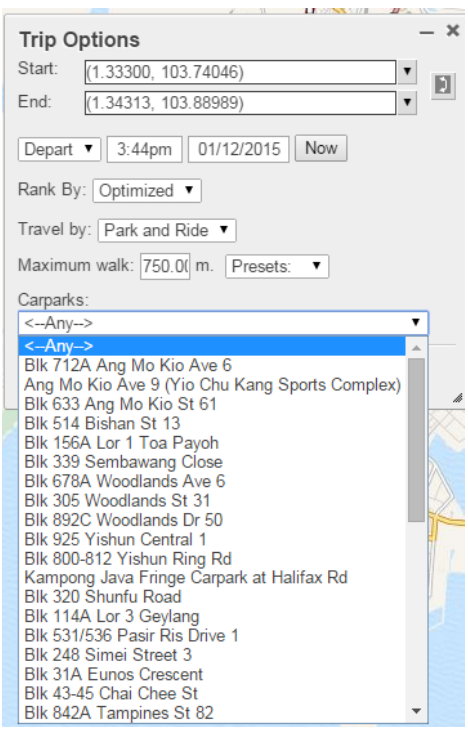


Fig. 15. Interface for car park selection

Fig. 16 shows the two cases that the user does not choose and chooses a car park, when *JPlanner* helps her do the planning. We can see that the system can make different metro and bus services for the user based on different requirements.

C. Comparison

In this section, we use one use case to demonstrate the difference between JPlanner and two service providers, i.e., Google Maps and Gothere, on the effectiveness of journey planning. Since the two service providers does not support multi-modal planning well, we only compare single-modal journey planning.

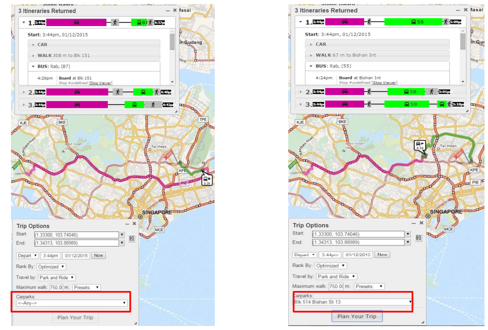


Fig. 16. Multi-modal route recommendation for park and ride

We sent the request to different services with the origin and destination of Sunset way (1.32346, 103.76894) and Goodwood Hill (1.31361, 103.83383) respectively. The recommendation result from the three services are listed in Fig. 17.

	JPlanner	Gothere	Google Maps
Option 1	Bus Service 75 and then transfer to Bus Service 105 **75→105**	Bus Service 151 or 154 and then transfer to Bus Service 105 or 132 or 190 or 972 **151/154→105/132/190/972**	Bus Service 151 or 154 and then transfer to Bus Service 190 **151/154→190**
Option 2	Bus Service 74 or 74e and then transfer to Bus Service 105 **74/74e→105**	Bus Service 151 or 154 and then transfer to Bus Service 48 or 66 or 67 or 170 or 171 **151/154→48/66/67/170/171**	Bus Service 7 with no transfer in between. **7**
Option 3	Bus Service 151 or 154 and then transfer to Bus Service 171 **151/154→171**	Bus Service 75 and then transfer to Bus Service 105 **75→105**	Bus Service 75 and then transfer to Bus Service 105 **75→105**
Option 4	NIL	NIL	Bus Service 74 or 151 or 154 and then transfer to Bus Service 171 **74/151/154→171**

Fig. 17. Recommendation result for the three services

We can see that there are two common sequences among the three systems, i.e., 75 $\rightarrow$ 105 and 151/154 $\rightarrow$ *. 75 $\rightarrow$ 105 is ranked as the first choice by *JPlanner* but the third by the other two, while 151/154 $\rightarrow$ * is ranked third by *JPlanner* but the first by the others. With a deeper investigation, we notice the fundamental reason that causes the differences is the plan for walking, of which we are using more detailed data and algorithms. This can be illustrated better in Fig. 18.

We can see Gothere only uses main roads and always assumes a direct walk from one point to another, where encourage jaywalking or reckless pedestrian crossing of a roadway. Google Maps follows the roads at origin and destination, but there is still a 'jump' at the transfer stage which is illegal and misleading to commuters. Also, the approximation will significantly influence the recommended routes because it

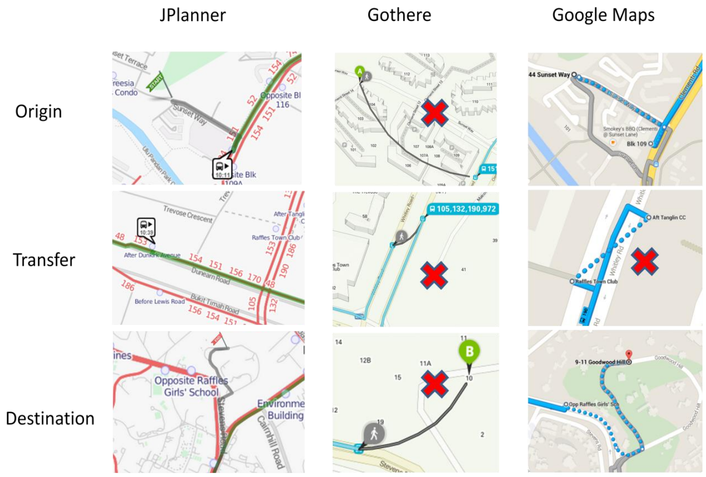


Fig. 18. Segmented routes for Sunset Way to Goodwood Hill

takes much less time to "jump" than walk safely and legally. According to our test, most of the differences between our results and others' are due to this. *JPlanner* incorporates OpenStreetMap with the full level of details, and assume all the walking should be on the road and discourage the illegal crossing the roads.

V. Related Work

A. Speed fusion

As mentioned, there are two major sources for speed observation: (1) point detectors such as loop detectors or speed cameras which capture the instantaneous speed of each vehicle passing through a certain check point, and (2) probe vehicles, which also measure the instantaneous speed with timestamps along its trajectory. The average speed for the probe vehicles between each pair of consecutive points can be inferred after the real path on the road is found out using map-matching algorithms. In our work, the speed sensor network deployed in Singapore provides the point detection data and the taxis act as the probe vehicles.

Either source can be used alone to derive the road speed [5], [6], although fusing them to derive more trustful results have interested many researchers [7], [8], [9], [10], even in Singapore [11]. Most of the existing methods focus on statistical or machine learning methods such as Bayes fusion, evidence theory, neural network, etc [12], [11]. The root causes of the uncertainties for each different data source have not been well studied, which could be essential inputs for those training models. For example, the data used in those research are assumed to be complete and have the same spatiotemporal context, while in reality, the data can be very sparse and do not fit well with the road network.

B. Multi-modal journey planning

The first problem to solve for multimodal journey planning is to integrate transport networks for different modes. Horn presented a framework for integrating fixed schedule services such as bus and trains, and so-called demand responsive services such as taxi. It resolves each user request as single or multiple-leg journeys which are allocated to different planning models. Zhang et al. [13] proposed a model which connects the networks by using transfer links for which pedestrian network play an important role.

Incorporating real time data for planning has been paid much attention. Li et al. [14] replaced the fixed public transit schedule with a real time transit arrival time predictor in their journey planner application. Borole et al. [15] proposed a method to incorporate real time delays into transit network to minimize the gap of the prediction model. The system requires all the buses to mount GPS devices and continuously communicate with the central server. Leibig et al. [16] used a spatiotemporal random field method to predict the future traffic condition and use it as a parameter for the route searching algorithm, where they used Gaussian Process to model the junction based traffic flow values.

Performance improvement is also a challenge especially for cities with complex transportation networks. Delling et al. [17] proposed a label constrained method to for the shortest path problem which tries to avoid the frequent and unnecessary mode switching to improve the performance. Bast et al. concluded that the number of Pareto-optimal solutions from mutlimodal planning would explode for a traditional searching algorithm. They proposed a Types aNd Threshold filter which could produce a much smaller yet representative subset of results.

VI. Conclusion

In this paper, we present our journey planning system *JPlanner*. Different from the existing journey planning service providers, our system leverages on more comprehensive urban data to guarantee the effectiveness and accuracy of journey recommendation. In particular, *JPlanner* makes use of more traffic network data, e.g., park connector network, to enable broader range of transport modes to be supported. Furthermore, a speed fusion module is used to fuse different data sources to infer more accurate real-time traffic speed in different roads. This is the core to a more accurate travel time estimation, and thus a better journey recommendation. We demonstrate the *JPlanner* system using a few use cases. Finally, we compare the journey planning result returned by JPlanner with two existing services, i.e., Google Maps and Gothere, to show the differences and prove the advantages of *JPlanner*.

References

[1] L. Sun, J. Yang, and H. Mahmassani, "Travel time estimation based on piecewise truncated quadratic speed trajectory," *Transportation Research Part A: Policy and Practice*, vol. 42, no. 1, pp. 173–186, 2008.

[2] M. L. Stein, *Interpolation of spatial data: some theory for kriging*. Springer, 1999.

[3] K. Ghys, B. Kuijpers, B. Moelans, W. Othman, D. Vangoidsenhoven, and A. Vaisman, "Map matching and uncertainty: an algorithm and real-world experiments," in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2009, pp. 468–471.

[4] M. A. Quddus, W. Y. Ochieng, and R. B. Noland, "Integrity of map-matching algorithms," *Transportation Research Part C: Emerging Technologies*, vol. 14, no. 4, pp. 283–302, 2006.

[5] B. Hellinga and R. Gudapati, "Estimating link travel times for advanced traveler information systems," in *Proceedings of the Canadian Society of Civil Engineers, 3rd Transportation Specialty Conference*, 2000.

[6] R. Zito, G. d'Este, and M. A. Taylor, "Global positioning systems in the time domain: how useful a tool for intelligent vehicle-highway systems?" *Transportation Research Part C: Emerging Technologies*, vol. 3, no. 4, pp. 193–209, 1995.

[7] F. Soriguera Martí *et al.*, "Highway travel time estimation with data fusion."

[8] C. Bachmann, "Multi-sensor data fusion for traffic speed and travel time estimation," Ph.D. dissertation, 2011.

[9] Y.-J. Byon, A. Shalaby, B. Abdulhai, and S. El-Tantawy, "Traffic data fusion using scaat kalman filters," in *Transportation Research Board 89th Annual Meeting*, no. 10-2392, 2010.

[10] M. Berkow, C. M. Monsere, P. Koonce, R. L. Bertini, and M. Wolfe, "Prototype for data fusion using stationary and mobile data," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2099, no. 1, pp. 102–112, 2009.

[11] R. L. Cheu, D.-H. Lee, and C. Xie, "An arterial speed estimation model fusing data from stationary and mobile sensors," in *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, 2001, pp. 573–578.

[12] Q.-J. Kong, Z. Li, Y. Chen, and Y. Liu, "An approach to urban traffic state estimation by fusing multisource information," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 10, no. 3, pp. 499–511, 2009.

[13] J. Zhang, F. Liao, T. Arentze, and H. Timmermansa, "A multimodal transport network model for advanced traveler information systems," *Procedia Computer Science*, vol. 5, no. 0, pp. 912 – 919, 2011, the 2nd International Conference on Ambient Systems, Networks and Technologies (ANT-2011) / The 8th International Conference on Mobile Web Information Systems (MobiWIS 2011). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050911004522

[14] J.-Q. Li, K. Zhou, L. Zhang, and W.-B. Zhang, "A Multimodal Trip Planning System With Real-Time Traffic and Transit Information," *Journal of Intelligent Transportation Systems*, vol. 16, no. 2, pp. 60–69, 2012. [Online]. Available: http://dx.doi.org/10.1080/15472450.2012.671708

[15] N. Borole, D. Rout, N. Goel, P. Vedagiri, and T. V. Mathew, "Multimodal Public Transit Trip Planner with Real-time Transit Data," *Procedia - Social and Behavioral Sciences*, vol. 104, no. 0, pp. 775 – 784, 2013, 2nd Conference of Transportation Research Group of India (2nd CTRG). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877042813045631

[16] T. Liebig, N. Piatkowski, C. Bockermann, and K. Morik, "Predictive Trip Planning-Smart Routing in Smart Cities." in *EDBT/ICDT Workshops*, 2014, pp. 331–338.

[17] D. Delling, T. Pajor, and D. Wagner, "Accelerating multi-modal route planning by access-nodes," in *Algorithms-Esa 2009*. Springer, 2009, pp. 587–598.