Extended Web Direction Service to Avoid Obstacle on Road using Turf.js

Kampanart Piyathamrongchai

Naresuan University, Faculty of Agriculture, Natural Resources and Environment,
Department of Natural Resources and Environment.
99 Moo 9 Tambon Tha Pho, Amphoe Muang Phitsanulok, Thailand, 65000

ABSTRACT

Web routing service is one of the most fashionable geospatial functions which are often used to find the best or the fastest route from one point to another point nowadays. The complex routing algorithm has been applied and extended to several web map service platforms, for example, Google Direction API, Graphhopper direction, Mapbox direction and Openrouteservice. The most advantages of using these platforms are not only map data updated but also the Javascript API has been designed to be easy to code. However, most web routing service platforms cannot add points or areas to specify avoidable roads, which is necessary in some routing situations, e.g., inundation and accident on the road. This paper attempts to implement a trial web routing application which integrate direction API with advanced spatial analysis for browser Javascript package called Turf.js. To implement, Google direction API was used to generate routes by defining origin and destination points and avoided road specified by point based on interactive Google map platform. A set of spatial analysis functions in Turf.js was used to examine whether or not all alternative routes are intersected with obstacles. This paper represents examples of using this extended web routing service to several situations. It shows good results that the system can suggest other routes if the first best route is obstructed. This provides better results for planning and making decision in several routing problems using google or other direction service platforms, i.e., finding an alternative route when there is an accident on a major road, and choosing the suitable hospital in emergency medical service.

1. INTRODUCTION

Web routing service is one of the most useful tool for users that can access to the map service via web browser and smartphone nowadays. It is the service that is often used to find the best routes from an origin location to destinations. There are several web map service platforms that provide the routing service to end users for example; Google Direction API, Grasshopper direction, Mapbox direction and Openrouteservice. The vital problem of constructing routing applications is scarce ready-to-use transportation network data which needs huge budget and time to collect and manage. However, these services basically provide up-to-date map data which is ready to apply routing algorithms to construct the direction and other routing services especially Google Map. Moreover, these services provides well-designed Javascript APIs which are ease to program as well. These are the reasons why web map service platforms are most currently used in many web and smartphone applications.

Nevertheless, most of web routing service platforms cannot add features to specify avoidable roads, which is necessary in some routing situations, e.g., inundation area or accident incident on the road. The advantage of constructing a routing application, using our own data and developed algorithm such as pgRouting, is to modify the programming code to create obstacles on the road to find alternative routes. Whereas most of web map service platforms cannot be customised to do so.

This paper attempts to implement a web routing application which integrate Google direction API with advanced spatial analysis for browser Javascript package called Turf.js. Nedkov (2011) built a tool based on Google map and direction service to avoid road obstacle but it was necessary to build complex spatial analysis algorithm. The Turf.js is, on the other hand, a client-sided package which provides several popular spatial analysis tools, e.g. buffer analysis, spatial query tools and overlay analysis etc. This can be used to extend the capacity of web map service platform to add more tools to analyse the data spatially. This paper represents how to integrate Turf.js with Google direction API to find alternative routes in case of the fastest route are blocked with the obstacle.

2. OBJECTIVES

To use of the advanced spatial analysis for browser Javascript package called Turf.js in order to extend the capability of Direction API service to find alternative routes when roads are blocked.

3. FRAMEWORK

Figure 1 represents the framework of the application. This study applied google map and google direction API as a prototype to integrate Turf.js to create the routing application that can add the obstacle to get the best route. The spatial functions in Turf.js, e.g. buffer analysis and overlay analysis were applied to the routes suggested by the direction API in order to select the best route. The routes that intersect to the obstacle zone will be erased and the application will recommend alternative routes to reach to the destination.

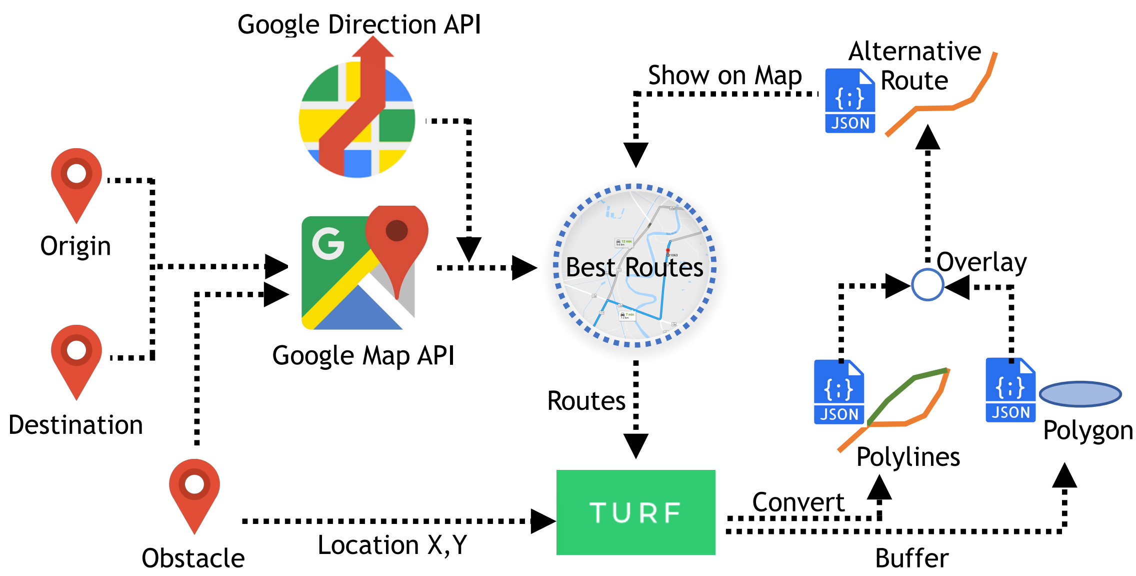


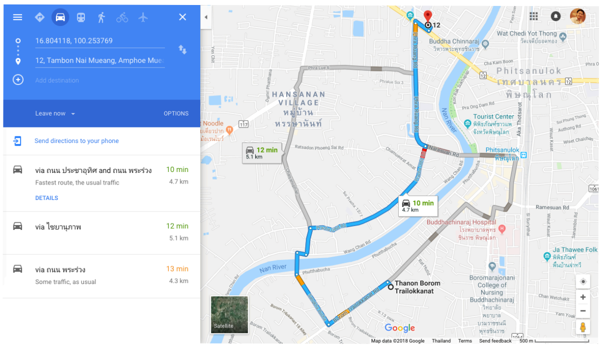
Figure 1. Conceptual Framework

4. IMPLEMENT THE APPLICATION

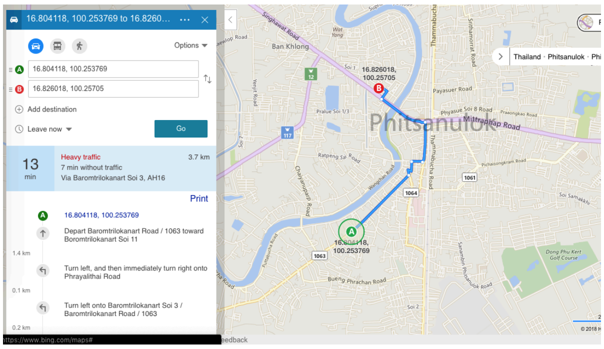
As described in the last topic, there are two main programming components, Map and Direction API, and Turf.js, to be used to develop the application. This section discusses each component in the way that were used in this study.

4.1 Map and Direction API

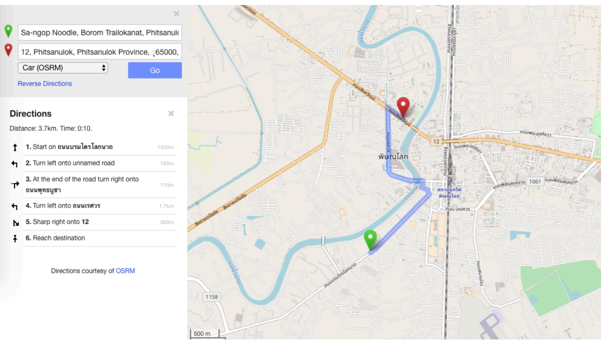
There are several web map services that can be used for free or by subscription, e.g. Google Map, Open Street Map, Bing Map and Map Box etc. Most of these services support up-to-date road network data that can be used for finding places or locations and for navigation system. However, the direction function of these services return different results. Google map provides the best routes and some other alternative routes which users are able to choose. Whereas other services gives the result as one best route only (see Figure 2.). Therefore, the google map and direction API is used as base map in this prototype application.



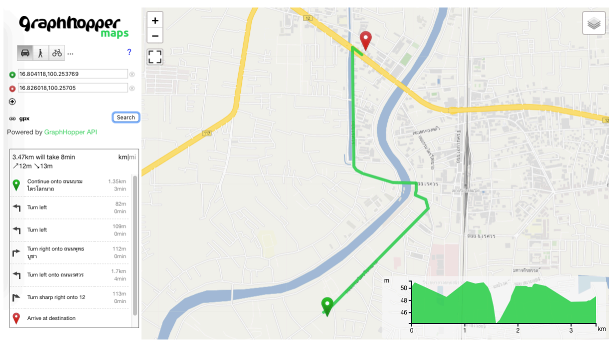
a) Google map



b) Bing map



c) Open Street map

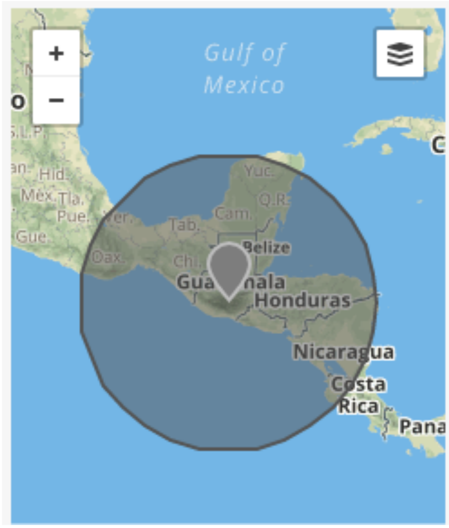


d) Grasshopper map

Figure 2. Direction from Web map services

4.2 Advanced geospatial analysis for browsers with Turf.js

Turf.js (http://turfjs.org/) is a Javascript library that composes of sets of useful spatial analysis modules such as measurement, coordinate mutation, transformation, feature conversion, booleans, and interpolation. Each module stores many spatial functions which can be used to process the spatial data, for instance; 'lineToPolygon', 'combine', 'polygonize' in feature conversion module. Turf.js is client-sided process language, then users can write the codes to apply those spatial modules directly in the web browser. The spatial data used in Turf.js is in form of GeoJSON. Users can therefore prepare their data in this form as input data, then the output data will be returned as GeoJSON after applying spatial functions.




```
var point = turf.point([-90.548630, 14.616599]);
var buffered = turf.buffer(point, 500, {units: 'miles'});
```

Source: turf.org

Figure 3. Example of using Turf.js

Figure 3 represents an example of using Turf.js to create a point on the map by only one line of code. The result of *turf.point* function will return a GeoJSON object and then collect to the *point* variable. For the next line of code, the point object created in the previous line can be used as input data to *turf.buffer* function. Then the buffer zone in polygon shape will be created as GeoJSON object and assign to *buffered* variable. Moreover, Turf.js prepares plenty of spatial functions that are able to compare and overlay two or more feature layers such as *turf.booleanWithin (*returns true if the first geometry is completely within the second geometry*)* and *turf.union* (takes two or more polygons and returns a combined polygon). Since Turf.js is convenient to code, it returns GeoJSON object, and it is client-based processes, this is suitable for extending the application that may find difficulty to customise.

4.3 Programming web application

This study utilised Google Map and Direction API version 3 to build web map application. Moreover, Turf.js is also Javascript library. The web application designed in this paper was therefore developed as client-sided based application. It is unnecessary to implement the server-sided database and coding.

Procedure of the application:

- User locates 3 points directly on the map including 1) origin, 2) destination and 3) obstacle respectively.
- The origin and destination locations are sent to the google map API to create

markers on the map and then are set as origin and destination locations on the google direction API.

- The direction API returns the best routes and alternative routes.
- All routes are convert to *turf.LineString* that is in GeoJSON format.
- The obstacle location defined in step 1 is used to apply *turf.buffer* function to create buffer area in GeoJSON format.
- The routes in step 4 and obstacle buffer area in step 5 are overlaid using *turf.disjoint* function to eliminate the route that intersects to the obstacle area.
- The alternative routes that do not intersect to the obstacle zone are drawn to the web map.

5. TESTING THE APPLICATION

The web application interface shown in Figure 4 represents major components of the prototype. Users can interactively define 2 points with different symbol including: origin as starting point and destination as the finishing point to calculate best routes. To keep web application simple, the road block is located interactive as another point on the map; then the 100 meters-buffer zone is constructed as road obstacle area. After defining all necessary points, the application is ready to analyse to find the best route and alternative routes.

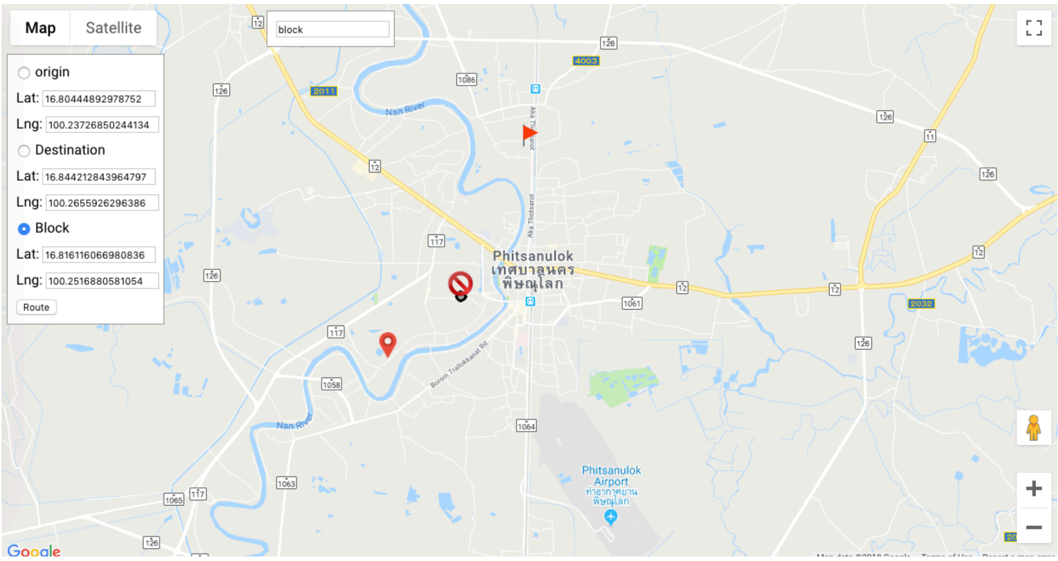


Figure 4. Web routing application interface

Figure 5 shows some examples of using extended web routing application to find routes in two cities including Phitsanulok City in Thailand and Can Tho City in Vietnam. In a trial, an origin and destination point were defined on the map. The application then returned alternative routes suggested by google direction API as shown in Figure 5a) and 5c). On the other hand, when a road block was located, some suggested route that passes over the road block was then erased. The map represents the final result as alternative routes which avoid the road obstacle from the origin to the destination as depicted in Figure 5b) for Phitsanulok City and in Figure 5d) for Can Tho City.

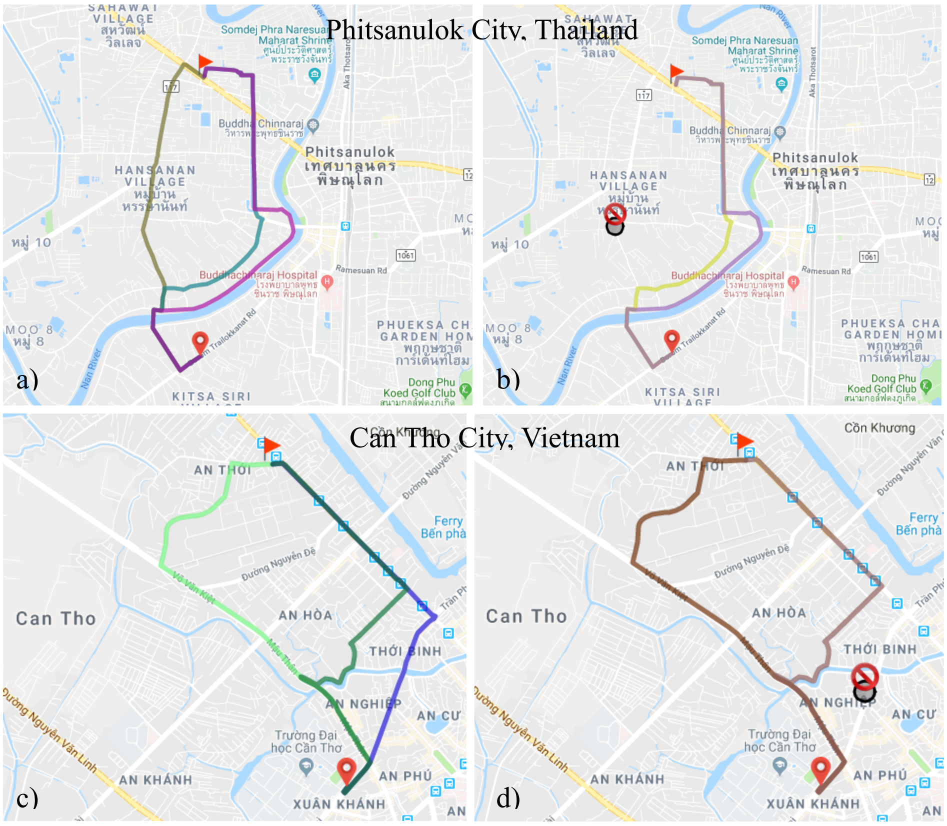


Figure 5. Examples of finding suitable route in Phitsanulok and Can tho City

6. SUMMARY AND RECOMMENDATION

This routing web application was developed to gain advantages of an existing map and direction service (Google direction API) integrating with the Turf.js, which is spatial analysis Javascript package, to extend the capability of routing service. The results clearly show how the application avoids some route that is obstructed by road block and promotes alternative routes. It is also obvious that Google map and direction service has very rich in global road network information which users can apply this application to almost everywhere on earth. This prototype can be useful in several fields that need to use routing service to analyse, e.g. emergency service (Choosumrong et.al., 2014) and food delivery service (Tobing, 2016). Users can apply this idea to other map and direction service platforms and extend the application to more specific field in the future.

7. REFERENCES

Choosumrong, S., Raghavan, V., Delucchi, L., Yoshida, D. and Vinayaraj, P., 2014. Implementation of Dynamic Routing as a Web Service for Emergency Routing Decision Planning, International Journal of Geoinformatics, 10(2), 13-20 (ISSN 1866-9298).

Nedkov, S. And Zlatanova, S., 2011. Enabling Obstacle Avoidance for Google Maps' Navigation Service. In Proceedings of 7th Geoinformation for Disaster Management, Anltalya, Turkey, 3–7 May 2011.

Tobing, R. D. H., 2016. A Food Ordering System with Delivery Routing Optimization Using Global Positioning System (GPS) Technology and Google Maps. Internetworking Indonesia Journal Vol.8/No.1, 17-21.