# ROUTE PLANNING FOR THE BEST VALUE FUEL

Alva Sheehy and Kenneth Dawson-Howe
*School of Computer Science and Statistics, Trinity College, Dublin 2, Ireland*

Keywords: Route Planning, Optical Character Recognition, Petrol Price Determination, Vision Application.

Abstract: The system described in this paper is an extension to the standard satellite navigation systems used in vehicles. A new route planning algorithm is developed using both shortest path and simplest path criteria and then this algorithm is extended to allow the incorporation of a visit to a petrol station along the route. The choice of petrol station is based on a combination of the relative location of the petrol stations and the cost of petrol at those stations. The price of petrol at each petrol station is constantly updated on a central server which is provided with prices by all vehicles which are using the system as they pass by. The price of petrol is determined using a camera mounted on the dashboard of the car, the images from which are processed with reasonably standard OCR software. When a vehicle requires fuel the central server provides prices for petrol stations in the vicinity of the planned route.

## 1 INTRODUCTION

Satellite Navigations (SatNav) systems are in everyday use, but are generally based on static route maps. These systems can be extended by, for example, using dynamic maps, incorporating information about road-works, accidents and traffic flow. This paper looks at a novel way of extending SatNav systems by incorporating a facility to help direct the vehicle to the best value fuel. There are two major questions for such a system. First, how does the central server get information on petrol station location and prices? Second, what does the SatNav do with this information? These questions essentially divide the system into two main parts, the acquisition of information and the use of this information.

To acquire the current fuel prices we assume that a large number of vehicles are subscribed to the system and that these vehicles are all fitted with the required navigation device which will incorporate a dashboard camera looking forwards through the windscreen. When the vehicles are being driven around the system automatically reads (using image processing) the petrol prices at each station passed and transmits this price information (along with the location of the petrol station) to the central server. See Section 2 (Image Processing).

When a vehicle requires fuel a request is made to the central server for the most up-to-date fuel prices. The SatNav system can then determine which station to incorporate into the route. This decision has to be based on (1) fuel prices at the different stations, (2) the additional financial cost of visiting each petrol station in terms of distance, and (3) the additional time and complexity cost of visiting each petrol station. In fact it is necessary for the user to explicitly decide upon the relative importance of time/complexity cost vs. financial cost. See Section 3 (Route Planning).

## 2 IMAGE PROCESSING

In order to obtain the fuel prices images from the dashboard mounted cameras must be processed to 1) locate the petrol station signs (See Figure 1) and 2) determine the price of the petrol. To locate the petrol station sign we make use of Optical Character Recognition (Cheriet 2007) by employing the Tesseract OCR software (Smith2007) to locate the keywords “Unleaded” and “Diesel”. If either word is found we proceed to attempt to locate both the other word and the prices.

The petrol prices are generally to the right of the “Unleaded” and “Diesel” keywords and so we crop a section of the image around the keywords located (See Figure 2). Note that if one of the keywords was not found in the initial search a larger region than those shown in Figure 2 is cropped (i.e. including

Sheehy A. and Dawson-Howe K. (2010).
ROUTE PLANNING FOR THE BEST VALUE FUEL.
In *Proceedings of the International Conference on Computer Vision Theory and Applications*, pages 173-178

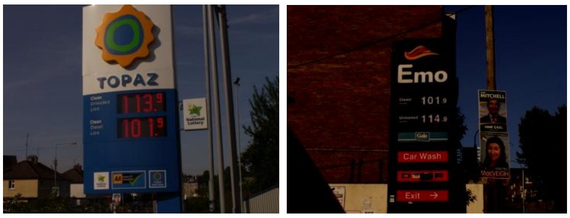


Figure 1: Example images of petrol station signs.

the area above and below the keyword found) and the keywords are searched for using Tesseract once again. They frequently allows the other keyword to be found as the area being processed does not contain much background clutter (such as that shown in Figure 1) and hence the processing will often more reliably threshold and locate the words.


Figure 2: Cropped image based on keyword location.

Many (or perhaps most) petrol station signs use a large number of LEDs to display the prices (e.g. See Figure 2). This poses a problem for most OCR packages as they cannot recognise the characters. To overcome this a morphological closing operation (i.e. a dilation followed by an erosion) is applied to the binary images (See Figure 3). In fact it was necessary to apply this process iteratively using progressively later morphological filters until the petrol price could be reliably determined. Note that for this demonstrator system it was assumed that the decimal place was always 0.9 and hence it was not processed.


Figure 3: The result of closing operations on the binary image shown in Figure 2.

The method was validated using a small sample set of 8 images of different petrol station signs.

# 3 ROUTE PLANNING

This section deals with how the navigation system chooses routes. Please note that what is presented here is a summarised version of the route planning which was developed. For further details including more motivation behind the developed cost functions see (Sheehy 2009).

## 3.1 Background

### 3.1.1 Simplest Route vs Shortest Route Cost Functions

Shortest-path methods can take into account a number of different criteria when computing an “optimal” path (sequence of roads) between an origin and a destination. The cost function associated with any particular route can be weighted to take into account such criteria as distance travelled or time taken to travel a route. Penalties assigned to particular roads can also be taken into account. By applying weightings to these criteria, combinations of route finding criterion can be used in determining an optimal path between two points. The criteria used with shortest-path methods are usually “concrete” physical criteria such as the physical length of a particular road or edge.

Often the driver would prefer to be given a simple, easy path to follow in preference to a path which is more complicated but is distance-wise, shorter. This is especially true if the shortest path would involve many turns and smaller roads etc. A number of papers (Riesbeck 1980, Elliott 1982, Streeter 1986) dealing with cognitive studies suggest that the overall complexity of a route is just as important in developing a good navigation system as being able to find routes with shortest times or distances. (Riesbeck 1980) emphasises the importance of clear, concise directions while (Elliott 1982) studies the directions given by people when asked how to get from a particular origin and destination. (Streeter 1986) in particular discusses the important factors which users consider in choosing a route. The number of turns as well as the number of traffic lights on a particular route was given importance by the users, just as was the “classical” idea of the time taken and the distance travelled. This all suggests that successful route planning systems must take all such criteria into account.

(Mark 1986) conjectures that the simplest routes to direct would also be the simplest for the driver to navigate, thus minimising the chance of errors due to navigational problems. The algorithm put forward in (Mark 1986) takes into account both the length of the route and the complexity of its description. He does this by classifying intersections by how difficult they are to guide a user through and adjusting the weights of each route accordingly.

An improvement suggested in (Duckham 2003), would be to take into account the type of roads in the network. Their algorithm relies purely on evaluating the complexity of navigating a particular route. As such there is no account made for how quickly a particular road can be traversed, or indeed the speed limit or the number of lanes on the road. It would also be good to direct the algorithm to more important roads. That said, the authors note that the simplest route is often already directed to the more important roads.

### 3.1.2 Search Algorithm

In order to find an optimal route from a given origin to a given destination a search algorithm must be used. While Dijkstra's algorithm (Dijksra 1959) and some of its modified versions (Fu 2004, Sanders 2007) are widely recognised as some of the best route finding algorithms (as they are guaranteed to give the optimal result), they are very labour intensive both in terms of time and memory usage. This is due to the fact that they blindly searches for routes, looking in all directions equally.

To overcome these problems the A* or Goal-directed search algorithm (Guzolek 1989, Ikeda 1994) can be used. This algorithm incorporates an estimate of the remaining cost into the metric (i.e. $f(n) = g(n) + h(n)$ where $g(n)$ is the actual cost from the origin to node n and $h(n)$ is an estimate of the cost from node n to the destination. In order to guide the search and in order to guarantee an optimal solution, the value of $h(n)$ must be a lower bound for the remaining cost from the current node to the destination. When using shortest path methods, the value of $h(n)$ is generally based on the Euclidean distance and the maximum speed limit for the road network as this would give the minimum time to get to the destination. However, it is not as easy to use the A* algorithm when simplest path criteria dictate in the cost function. This is due to the fact there is no way to adequately determine a lower bound for the simplest path criteria from node n to the destination as these criteria generally do not depend on the relative position between node n and the destination. The only lower bound which could be chosen which would ensure a minimum bound would be zero (which in most cases would correspond to node n being on a road which itself leads to the destination without any instructions needed along the way). Having $h(n) = 0$ however, gives the Dijkstra Algorithm itself and so there would be no speed-up or increase in efficiency as the search would not be guided in any way.

## 3.2 Combined Simplest Route & Shortest Route Cost Function

We have developed a cost function which combines both the simplest route and the shortest route costs. To compute the shortest path we use the following formulas.

$$c(R) = \Sigma^n c_i = \Sigma^n (\mathrm{TravelTime}_i + \mathrm{RoadDelay}_i + \mathrm{TurnDelay}_i) \quad (1)$$

$$\mathrm{TravelTime} = \mathrm{Distance} / \mathrm{SpeedLimit} \quad (2)$$

where Distance is the distance of the road, SpeedLimit is the legal speed limit of the road, RoadDelay is the time delay in traversing a particular road (due mainly to traffic), and TurnDelay is the time needed to complete a specific turn. Note, the value for TurnDelay is the time delay in travelling from the previous road (road i-1) to the current road (road i).

To incorporate the simplest path with this metric we compute:

$$\mathrm{COST}(R) = (\Sigma^n (\mathrm{TravelTime}_i + \mathrm{RoadDelay}_i + \mathrm{TurnDelay}_i) * \mathrm{RoadTypeWeight}_i * \mathrm{LaneNumberWeight}_i) * \Pi^n \mathrm{TurnWeight}_i \quad (3)$$

where $\mathrm{RoadTypeWeight}_i$ is the road type weight which is assigned to road i. Take the following example: a minor road has a weighting of 1.1, a major road (of the same length) a weighting of 1 Each road has an initial cost function (total time of travel) of 4. When taking road type weighting into account the cost value ci for the minor road is now 4.4 while the cost value ci for the major road is now 4. Because the major road results in a lower cost function, an algorithm which favours lower cost functions will thus favour routes with major roads. $\mathrm{LaneNumberWeight}_i$ is a similar weighting factor which gives more cost to roads with single lanes (as more major roads facilitate easier overtaking of slow vehicles and are generally better signposted). The TurnWeight takes into account the complexity associated with navigating a turn. For example a right turn off a busy road without the aid of traffic lights is a much more stressful and complicated turn to take for the driver than a simple slip road off to the left.

$$COST_{A*}(n) = \left\{ \begin{array}{l} \sum_{i=1}^{n} (TravelTime_i + RoadDelay_i + TurnDelay_i) * RoadTypeWeight_i * LaneNumberWeight_i \\ + \left( \frac{D_{n,dest}}{v_{max}} + RoadDelay_{min} + TurnDelay_{min} \right) * RoadTypeWeight_{min} * LaneNumberWeight_{min} \end{array} \right\} * \prod_{i=1}^{n} TurnWeight_i * TurnWeight_{min}$$

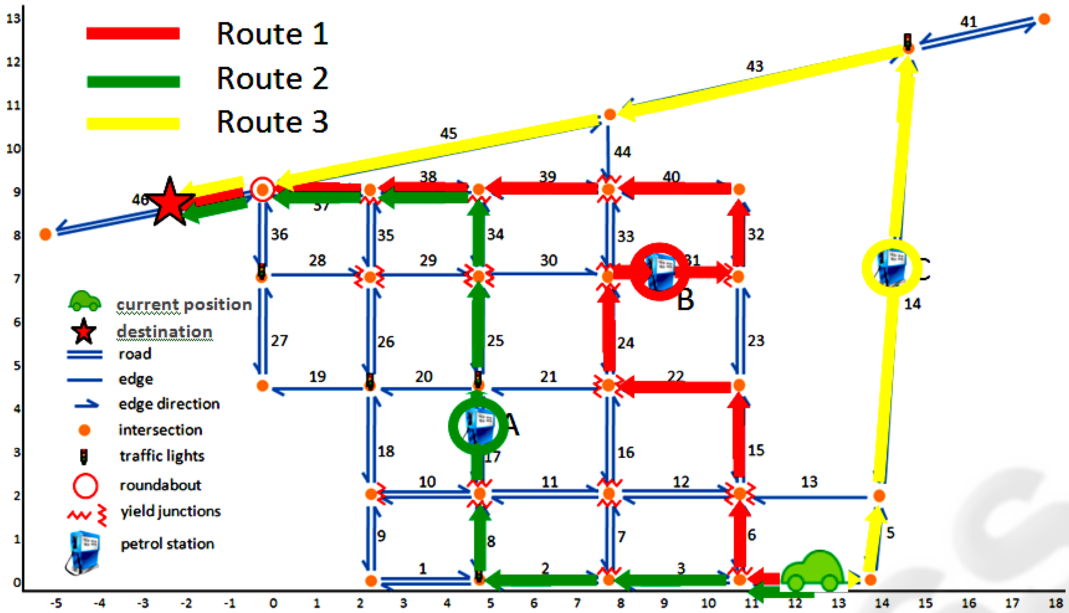


Figure 5: A road network with a car en route to a destination on which are shown three possible routes via different petrol stations. See text for explanation of route selection.

EuroCost is the cost associated with detouring to a given petrol station and w is the weighting factor which is used to weight between the two different cost functions. A weighting factor of w=1 would result in choosing a petrol station which would give the best route, regardless of the monetary cost. A weighting factor of w=0 on the other hand would choose a petrol station based on its monetary cost alone, regardless of the route to get there. (Remember that even though w=0 results in no comparison of the routes to different petrol stations, the route to get to these petrol stations are still chosen by the normal route finding algorithm and so are still the optimal routes to get to these particular petrol stations.)

In examining a particular petrol station, in order to improve on efficiency and reduce the number of petrol stations examined, the total overall cost is calculated at each iteration of the route finding algorithm. This is to cut off the search for a route to that petrol station (and hence ignore that petrol station) if the overall total cost is greater than the minimum cost found so far.

However, in order to compute the euro savings a value for the distance of the overall route needs to be found. As the euro savings value is being computed during the route search, this is clearly not possible. An estimate is used however, in a similar idea to that of the h(n) function in the A* algorithm. A lower bounds value for distance can be computed using Euclidean distances. As such, if a route search has not reached the petrol stations yet the value for the distance of the route $dist_{Route}$ will be;

$$dist_{Route} = dist_{RouteSoFar} + est_{n,station} + est_{station,dest} \quad (8)$$

where $dist_{RouteSoFar}$ is the distance of the route from the origin to the current position n, $est_{n,station}$ is the Euclidean distance from the current position n to the petrol station and $est_{station,dest}$ is the Euclidean distance from the petrol station to the destination.

# 4 VALIDATION

The image processing developed in section 2 was tested on a limited number of test images. The route planning developed in the previous section has been tested in a simulated environment. Initially scenarios were developed to test that the route planning algorithm worked appropriately in a variety of situations.

To validate the petrol price metrics a few different scenarios were considered. In Figure 5 below a car is shown on a road network with a destination shown by the red star. It is assumed that petrol is required and three routes which include petrol stations with associated prices are considered (A is €1.02, B is €1.00 and C is €1.04). When the weighting factor, w = 0.3 petrol station B (Route 1) is chosen. This route is also chosen for w=0 which implies that this is the route with the best monetary savings. When w = 0.7 petrol station A (Route 2) is chosen. Finally when w = 0.95 petrol station C is chosen. This is also true for w = 1 so this choice of petrol station gives the best overall route.

The scenario of changing petrol prices has also been considered and examples shown where the selected petrol station (and route) will change as the prices vary.

## 5 CONCLUSIONS

A prototype system has been developed demonstrating all of the technology required for the application proposed. More importantly the mathematics concerned with route planning and with integrating the choice of petrol station based on price have been developed and a basic validation achieved.

To further this work we need to integrate a real satellite navigation system with a camera system and a mobile communications system. Practically, this will require OEM involvement.

## REFERENCES

Cheriet, M., Kharma, N., Liu, C., Suen, C., 2007. *Character Recognition Systems: A Guide for Students and Practioners*, Wiley-Interscience.

Dijksra, E., 1959. A Note on Two Problems in Connexion with Graphs. In *Numerische Mathematik*, Pages: 269 – 271.

Duckham, M., Kulik, L., 2003. “Simplest” Paths - Automated Route Selection for Navigation, *In Lecture Notes in Computer Science: Spatial Information Theory*, October, Vol. 2825, Pages: 169 - 185.

Elliott M., Lesk, M., 1982. Route Finding in Street Maps by Computers and People, In *Proceedings of AAAI-82 Conferences* 1982.

Fu, M., Li, J., Deng, Z., 2004. A Practical Route Planning Algorithm for Vehicle Navigation System, in *Fifth World Congress on Intelligent Control and Automation (WCICA)*. Vol. 6, Pages: 5326 - 5329.

Guzolek, J., Koch, E., 1989. Real-time Route Planning in Road Networks. in *Vehicle Navigation and Information Systems Conference*, Sep 1989, Pages 165-169.

Ikeda, T., Hsu, M. Imai, H., Nishimura, S., Shimoura, H., Hashimoto, T., Tenmoku, K., Mitoh, K., 1994. A Fast Algorithm for Finding Better Routes by AI Search Techniques”, In Proceedings of Vehicle Navigation and Information Systems Conference, 1994, Pages: 291 - 296.

Mark, D., 1986. Automated Route Selection for Navigation. in *IEEE Aerospace and Electronic Systems Magazine* 1 1986, 2-5.

Riesbeck, C., 1980. You Can’t Miss It’: Judging the Clarity of Direction. In *Cognitive Science 4*, Pages: 285 – 303.

Sanders, P., Schultes, D., 2007 Engineering Fast Route Planning Algorithms. In *Lecture Notes in Computer Science: Experimental Algorithms*, June 2007, Vol. 4525, Pages: 23 - 36.

Sheehy, A., 2009. *A Driver’s Assistant: Finding the Best Routes and Petrol Stations*, MSc thesis, Dept. of Computer Science, Trinity College, Dublin 2. Ireland.

Smith, R., 2007. An Overview of the Tesseract OCR Engine. In *Ninth International Conference on Document Analysis and Recognition*, Vol. 2, Pages: 629 – 633.

Streeter, L., Vitello, D., 1986. A Profile of Drivers’ Map-Reading Abilities, In Human Facts, 28, Pages: 223-239