LANDMARK-BASED CAR NAVIGATION WITH OVERTAKE CAPABILITY IN MULTI-AGENT ENVIRONMENTS

Sirvan Khalighi[1], Somayeh Maabi[1], Mercedeh Sanjabi[1] and Ali Jahanian[2]
[1] *Department of Electronic and Computer Eng., Islamic Azad University, Qazvin Branch, Qazvin, Iran*
[2] *Department of Electronic and Computer Eng., Shahid Beheshti University, Tehran, Iran*

Keywords: Landmark-based car navigation, Wireless sensor networks (WSN), Multi-agent environment.

Abstract: Intelligent car navigation systems are planned to assist humans and route them automatically in the roads with sufficient security and correctness. Landmark-based car navigation is a widely used technique in automotive and robot navigation. In this paper, we improved a wireless landmark-based car navigation (WLCN) algorithm to operate in multi-agent (MA) environments. The extended navigation algorithm allows the cars to overtake in uni-directional real roads. Overtaking is based on the information which the cars send to each other in the road. According to this information and using a related algorithm, cars traverse each other. Analysis of accuracy and efficiency in various states, real-time RISC-based embedded system especially for high speed movements in real roads show that, the cars are navigated easily and reliable in multi-agent environments and they can successfully do overtake. In addition to reliable navigating, calculation cost of the algorithm is acceptable for real world scenarios.

1 INTRODUCTION

Automatic car navigation (ACN) system is regarded as one of the best kinds of offering solutions in intelligent transportation systems (ITS). The ACN systems are capable of doing some of the tasks (reading the maps, determining the best routes and etc.), that were performed by the driver. Recently, GPS and digital road maps are used for land vehicle navigation systems. The main drawback of using GPS is false positioning, due to the imprecise receivers and outdated maps. Therefore, GPS need more complicated map matching algorithms to state the vehicle location and navigation (Taghipour, Taghipour, 2008), which lonely can't be reliable for ACN. Most of the recent ACN algorithms are based on machine vision and artificial intelligence algorithms such as ant colony, neural network and etc. (Yashikawa, Otani, 2010). (Wu *et al.*, 2009) developed some prototypes for landmark-based car navigation using a full windshield head-up display (FWD) system. They used computer vision methods to correct distortion of FWD projection on the windshield. (Yashikawa, Otani, 2010) obtained a new routing algorithm to route on a graph embedding of the map. They proposed a combined method that integrates Tabu search and Ant colony optimization. This hybrid technique could find the shortest route when the blind alley existed in the map. Their searching algorithm is comparable with Dijkstra algorithm; But Dijkstra algorithm may not be the best solution for drivers because each driver has own definition in choosing the best route. Furthermore genetic algorithm is widely used to solve routing search and optimization problems. Kim (Kim *et al.*, 2009) considered the multi-objective mathematical formulation for ACN systems in real roads. Their method searched the road's map to solve the problem that involves the fuel cost in traffic congestion, the regulation of traffic, and the weather, etc. Hashing method was used to have a suitable selection in multi objectives route problems. In another work (Taghipour, Taghipour, 2008), proposed a correct mapping from GPS on the road network parameters. Their results show that the proposed algorithms can be effectively used for map matching. The algorithm used only location of the vehicle and vehicle speed information and database of the road network. Another car navigation system which communicated with each other through wireless LAN was developed by (Hiraishi *et al.*, 1999). The system used the traffic information from the other vehicles to perform the

Khalighi S., Maabi S., Sanjabi M. and Jahanian A..
LANDMARK-BASED CAR NAVIGATION WITH OVERTAKE CAPABILITY IN MULTI-AGENT ENVIRONMENTS.
DOI: 10.5220/0003752502340239
In *Proceedings of the 4th International Conference on Agents and Artificial Intelligence* (ICAART-2012), pages 234-239
ISBN: 978-989-8425-96-6

time-constrained search dynamically, yielding that the system can generate a new route to avoid the congestion. Significant developments and technical trends in the area of navigation systems were reviewed by (Hasan *et al*., 2009). They evaluated integration systems for obtaining a reliable and accurate navigation solution.

Analysis of previous works show the vision methods are more complex and time consume thus they are not suitable enough for real time systems. In this situation, other options such as GPS, digital roads and landmark-based maps should be explored. In addition, using new communicating technologies such as Zigbee, Wi-Fi and Wi-Max enables efficient inter-communication in a multi-agent environment. (Sanjabi *et al.*, 2009) obtained a landmark-based car navigation algorithm to route cars in the roads which are equipped with Wi-Fi landmarks. A map matching method has been used for navigating in the roads without branches. The nearest visited landmark was selected as free line. The next position will be chosen with arithmetic formulas on this free line. In another work we (Sanjabi *et al.*, 2009) discussed on navigating in multi branches routes. Every branch has its group landmarks, hence the route will be chosen according to its group landmarks IDs and car will be navigated in this route to reach the destination.

2 WIRELESS LANDMARKS

To make correct decisions during the agent's movement, agents need to know their location in the environment (e.g. localization). Therefore, agents must recognize the landmarks and communicate with them to read their internal information. To communicate between cars, there are many ways such as vision, sensor, wireless and etc. But in this project, due to the reasons which will mention in the following, just wireless landmarks have been used for navigating the agents: I) Wireless landmarks can be detected in high speed movement with low degree of error. II) In the recent years, wireless technology is dramatically improved, and many cost-effective wireless technologies (Wi-Fi, Zigbee and Wi-Max) have been commercially available. III) Information of wireless landmarks does not require any pre-processing such as other kind of signal processing. IV) Wireless landmarks make a new framework to construct a large sensor network in roads to navigate the cars. However, wireless technology consists of a variety of standards such as IEEE 802.1 (Zigbee), IEEE 802.2 (Bluetooth), IEEE 802.3 (Wi-Fi), IEEE 802.4 (Wi-Max) and some other emerging technologies such as cognitive radio. Due to significant features in terms of power consumption and maximum coverage area, Wi-Fi technology has been considered as wireless landmarks. Furthermore, the coverage area of a Wi-Fi landmark is about 50m, which is sufficient for ACN (IEEE 802 standard, 2007), (the IEEE 802.11 protocol, 2008).

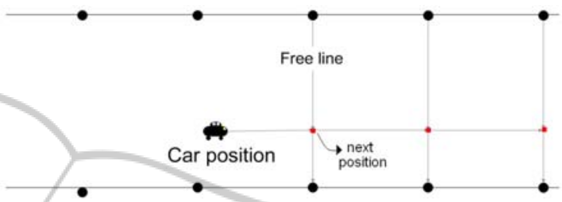


Figure 1: Place of landmarks in a road.

3 ACN USING WIRELESS LANDMARKS

To navigate the cars in multi agent environment an improvement of the WLCN algorithm (Sanjabi, 2009) is proposed. For each car, a Wi-Fi transceiver which broadcasts the car's position and receives the location of the other landmarks is considered. There are two types of wireless landmarks: Moving landmarks are the cars and fixed landmarks are the landmarks on two sides of the road (Figure 1). Due to distance limitation of wireless applications, Landmarks are assumed as Wi-Fi nodes (Wireless local networking, 2008). The maximum distance between each two Wi-Fi landmarks is a critical parameter. The suitable distance between landmarks will be evaluated in Section 5. Each Wi-Fi node propagated two essential data: A unique ID and the position of the node. When an agent reaches to a landmark covered domain, its ID and position will be visible by the moving agent. This information is used to localize the agent and navigate it along the road (Figure 2). The domain of each node maybe overlapped with the other nodes depending on the distance between the Wi-Fi nodes. As will be seen in the succeeding sections, the overlapping area has direct impact on the navigation algorithm.

4 WLCN-MA ALGORITHM

An algorithm is proposed to navigate the cars in a multi-agent road where two sides of the road are equipped with Wi-Fi sensors. The algorithm is

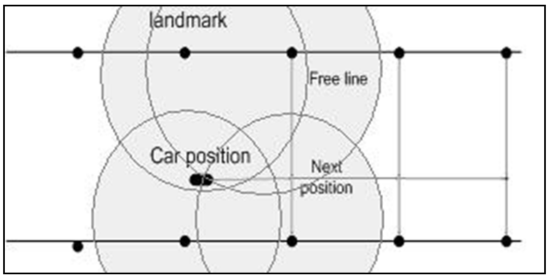


Figure 2: Covering domain of each sensor node.

named wireless landmark-based car navigation in multi-agent roads (WLCN-MA).It summarized in the flowchart which is shown in Figure 3. More details of each step will be described in the following.

4.1 Inputs of WLCN-MA algorithm

Inputs of the proposed algorithm consist of four parameters:

List of landmarks: The landmarks data extract from the various maps and then save as a text file (It is named file1). The algorithm reads the files and extracts the required data.

Other cars information: There is another file which contains information for navigating of the other cars that is generated in the road (It is named file2). This information will be used in overtaking algorithm. In addition, the other cars are separately navigated in the same map and their situation in each time slot is dumped into a file. This file is used as another input given to the WLCN-MA algorithm.

First position of car: Initial location of the car at start of the road.

End points of the route: End of the route is the position of two last landmarks which will be visible at the end of the route. Obviously, the algorithm will stop when visit them and process their data.

As summarized in Algorithm 1, WLCN-MA algorithm initialized using the start position of the agent in the route. In each iteration of the algorithm, percepts (location of the landmarks) are extracted from the environment (e.g. from input file). Current map of the road is updated based on visited percepts. Then, if there is a car in front of us (near than 50m), ACN system decides to do overtake; Otherwise, a free line (Figure 1) between the last visited percepts is constructed.

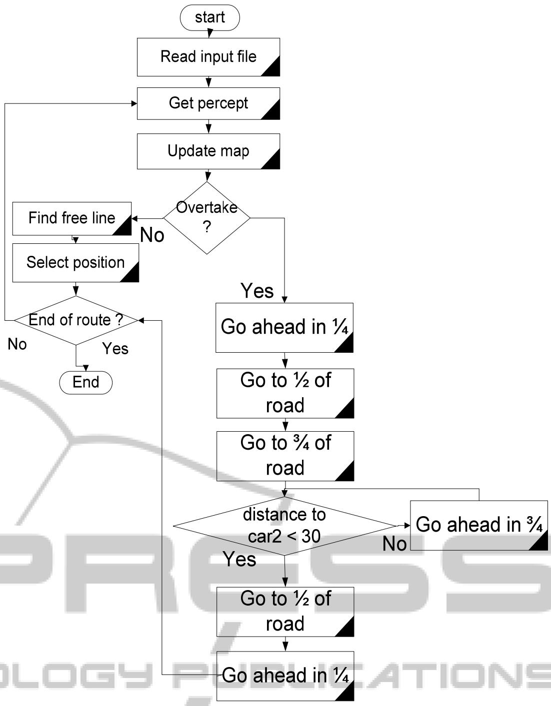


Figure 3: Flowchart of WLCN-MA algorithm.

Algorithm 1**: WLCN-MA**

1. Read the start position.
2. ***While*** (*TRUE*)
3. P_n = Get percept.
4. Update map.
5. ***If*** existing car in front of us ***then***
   (a) Decide to overtake,
   (b) Do overtake.
   else
   Find free line.
   [End of ***If*** structure]
6. Select the best next position *(NP).*
7. Move to new position.
8. ***If*** (*NP* is the end of route) ***then***
   Finish.
   [End of ***If*** structure]

[End of ***While*** structure]

[End of **WLCN-MA**]

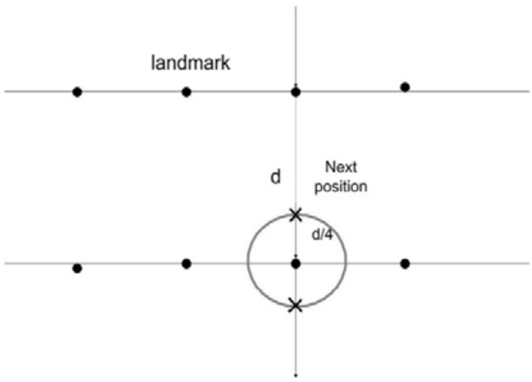


Figure 4: Select position based on a geometrical method.

Algorithm 2**: Overtaking**

1. Give the cars position.
2. Decrease our car speed.
3. *Call* **Free-line** algorithm.
4. Select new position in 1/4 right side of read.
5. *Call* **Free-line** algorithm.
6. Select new position in 1/2 right side of read.
7. Increase our car speed.
8. ***While*** (*our distance is less than 30m*)
   Call Free-line algorithm.
   Select new position in 3/4 right side of read.
   [End of ***While*** structure]
9. *Call* **Free-line** algorithm.
10. Select new position in 1/2 right side of read.
11. *Call* **Free-line** algorithm.
12. Select new position in 1/4 right side of read.
13. Return to ***main algorithm.***

[End of **Overtaking**]

Algorithm 3**: Free-line**

1. P_n = Get percept.
2. Update map.
3. Find free line.
4. ***Return*** Free-line.

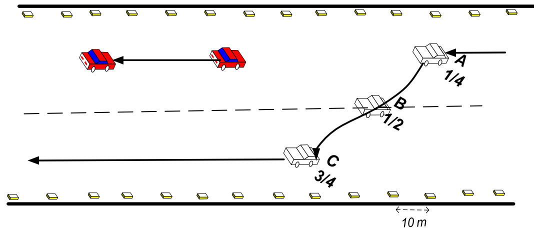
[End of **Free-line**]

After estimating the free line, the best point of the free line is selected as the candidate point for the next position of the car according to the distance formula (1) (figure 4).

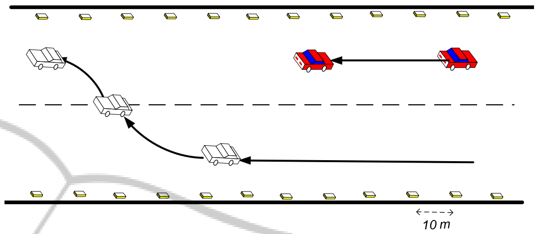
$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

Finally, the next point is calculated and the current position of the agent will be updated. If the new current position is on the free line between the last landmarks (end of the route), the algorithm is finished. Otherwise, the algorithm goes ahead to get the next percepts from the input file. As it can be seen from the figure 4, the center of the circle is right landmark of free line and its radius is *d/4* where *d* denotes the road width. This crossing leads to two positions but one point which is inside of the road is the next position. It is worthwhile to consider some criteria points in the selection of suitable next position. The proposed algorithm can be easily modified to consider any new metric.

The overtaking Algorithm: As it is shown in the Figure 5-(a) and summarized in the algorithm 2, algorithm 3, when one car (the red car) with lower speed and inadequate distance, is visited in front of our position (the white car), the main algorithm decides to do overtake. Therefore, the car's speed is decreased down to the front car’s speed (the red car), then the car's movement is continued for one step in 1/4 right side of the road, where step means passing states of get percept, update map, find free line and



(a)



(b)

Figure 5: (a) Steps of overtake process, (b) Overtake steps (backing).

finally select next position. Then, the car must go to 1/2 and more to 3/4 from right side of the road in two steps. After increasing our speed for passing the front car, we regularly check our distance with the red car by calling a move function. A move function compares current time with the existing times in the file2 and finds the closest time to extract the red car’s position. This process is for simulating concurrent movement of two cars. Then distance of two cars calculates with the distance formula (1). If we passed the red car with a suitable distance, we back to 1/4 right side of the road, otherwise, remaining in 3/4 one with passing just one step. For checking the distance we need to know passed time according to movement. This time is calculated by

$$T = X/V \quad (2)$$

Where V, X and T denote the car's speed, car's movement and current time respectively, and then is fed into the move function (V is assumed as fixed value). To finish the overtaking process, the distance of our car and the red car is needed to be known. Thus, distance of two cars calculates using move function and the distance formula (1). For backing, the car first goes to 1/2 and then 1/4 from right side of road with passing two steps. The algorithm will be finished if new position is located on or after the end point of the route which is extracted (algorithm 2) (Figure 5-(b)).

5 EXPERIMENTAL RESULTS

The WLCN-MA algorithm has been implemented

Table 1: Experimental results in terms of error rate.

Benchmark map	Visible landmarks	NP deviation	D	Error rate (%)
1	16	0	16	0%
2	16	2	16	2%
3	16	1.5	16	1.5%
4	16	0.5	16	0.5%
Average	**16**	**2**	**16**	**2%**

on a MIPS-based embedded system to evaluate the algorithm in real world conditions. Algorithm was developed in C programming language and compiled on *MIPS* architecture using a standard *gcc* cross-compiler with optimization level *O3*. The algorithm is evaluated with five road maps that are extracted from the real maps.

Table 1 shows the experimental results of different maps in terms of error rate. These maps are shown in Figures 6, 7. The column of ***visible landmarks*** shows the average number of visited landmarks. Column ***NP deviation*** represents the average deviation of calculated next position and ideal position for the car. It is noted that the error-free position is calculated manually to compute the error rate. Furthermore Column ***D*** shows the length of the free line and finally, the column ***Error rate*** shows the deviation of estimated route in percent. As can be seen from Table 1, the average number of visible landmarks is almost 16. As the number of visible landmarks increases, the reliability of the method improves. Therefore, we need to choose appropriate distance between the landmarks. The minimum visible landmarks that are needed to navigate are two, but some other problems such as effect of noise, car speed, Wi-Fi limitations or even road conditions enforce to require more than 2 landmarks in each step. The results of implementing the WLCN-MA algorithm on MIPS-based embedded system are shown in the Table 2. In this table, Frequency's column shows the operational frequency of embedded system and Landmark distance's column shows the average distance between landmarks. Column Total runtime shows

Table 2: Experimental results in terms of allowable maximum speed.

Frequency	Landmark distance	Total runtime	Step runtime	Max. possible speed (m/s)
300 MHZ	10	13.3 ms	4 ms	250
300 MHZ	20	13.3 ms	4 ms	500
300 MHZ	30	13.3 ms	4 ms	600
100 MHZ	10	40 ms	12 ms	83.33
100 MHZ	20	40 ms	12 ms	166.66
100 MHZ	30	40 ms	12 ms	250

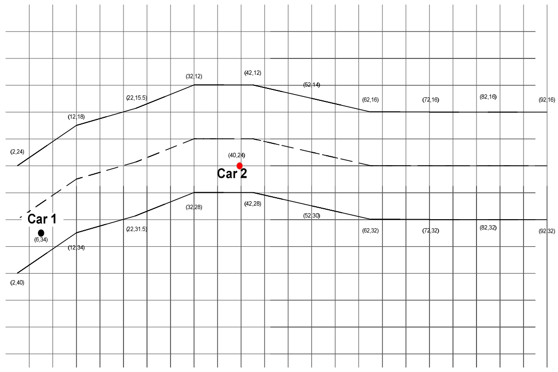


Figure 6: The road No. 1 (first benchmark in Table 1).

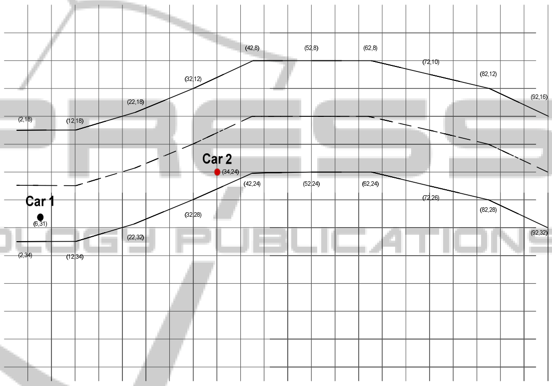


Figure 7: The road No.2 (second benchmark in Table 1).

the total time needed for navigating the agent from start to end of the path and Step runtime represents the runtime for one step of the algorithm. And finally, the column Maximum possible speed represents the maximum speed of each car during the navigation process. Experiments are performed on all benchmarks with three different landmark distances and three various operational frequencies. As can be seen in this table, the WLCN-MA algorithm can be used in high speeds with accessible frequencies and very small error rate because of its light computations.

6 CONCLUSIONS

This paper improved the pervious car navigation algorithm based on the landmarks to work in multi-agent environments. In fact, the overtaking capability was the main improvement of it. Decision of the overtaking is according to cars transactions with each other. The experimental results show that the proposed algorithm can be used for real-time and high-speed agents even in low operational frequencies. As future research, we are going to consider other moving agents as obstacles using more intelligent algorithms.

REFERENCES

A technical tutorial on the IEEE 802.11 protocol, Available on htt://sss-mag.com/pdf/802_11tut.pdf, 2008.

Apolloni B., et. al, October 2005. Machine learning and robot perception, *Springer*.

Hasan A. M., Samsudin Khairulmizam, Ramli Abd Rahman, Azmir Raja Syamsul, and Ismaeel Salam A., 2009. A review of navigation systems (Integration and Algorithms), *Australian Journal of Basic and Applied Sciences*, pp: 943-959

Hiraishi Hironori, Ohwada Hayato and Mizoguchi Fuinio, 1999. Intercommunicating car navigation system with dynamic route finding, *International Conference on Intelligent Transportation Systems*, pp: 284-289

"IEEE 802 standard," Available on http: //en.wikipedia.org /wiki/ IEEE_802, 2007.

Kim Byung-Ki, Jo Jung-Bok, Kim Jong-Ryul and Gen Mitsuo, 2009. Optimal route search in car navigation systems by multi-objective genetic algorithms, *International Journal of Information Systems for Logistics and Management*, Vol. 4, No. 2, pp: 9-18

Olson C. F., *2002*. Selecting landmarks for localization in natural terrain, *In Autonomous Robots Manufactured in the Netherlands*

Sanjabi M., Maabi S., Jahanian A. and Khalighi S., 2009. A light-weight car navigation algorithm for high speed agents using wireless landmarks, *IEEE International Conference on Information and Automation*, pp: 1028 - 1033

Sanjabi M., Maabi S., Esmaeili Z., Jahanian A. and Khalighi S., 2009. A landmark based navigation system for high speed cars in the roads with branches, *International Journal of Information Acquisition*, Vol. 6, No. 3

Taghipour Sara, Taghipour Ali, April 2008. An algorithm for map matching for car navigation system, *The International Conference on Information & Communication Technologies*, pp: 7-11

Wu Wen, Blaicher Fabian, Yang Jie, Seder Thomas, Cui Dehua, 2009. A prototype of landmark based car navigation using a full windshield head up display system, *Workshop on Ambient media computing*

"Wireless local networking based on 802.11standards," Available on http: //www.Wi-fi.org, 2008.

Yoshikawa Masaya, Otani Kazuo, 2010. Ant colony optimization routing algorithm with Tabu search, *Directory of open access journals*, pp: 2104-2107, VOL. 2182