Indian Journal of Science and Technology, *Vol 9(20), DOI: 10.17485/ijst/2016/v9i20/94704, May 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Design of Road Database for Self-Driving Vehicles

Jae Moon Lee*

Department of Multimedia Engineering, Hansung University, Korea; jmlee@hansung.ac.kr

Abstract

Background/Objectives: In recent industry, self-driving vehicles are very actively researched because they are considered as a new industrial power in most countries. In self-driving vehicles, many different fields should be researched at the same time. The research of road database for self-driving vehicles should not be indispensable in those fields. **Methods/ Statistical Analysis:** In order to design the road database for self-driving vehicles, the conventional road database should be enhanced in many sides, because it is designed as auxiliary equipment under assumption that a human is driving the vehicle. While the conventional road database are focused on the accumulation of static data, the road database for self-driving vehicles must include dynamic data such as roads that may be temporarily closed because of repairing. **Findings:** This paper proposes the design of a road database for self-driving vehicle so that it includes dynamic data as well as static data on roads. In order to provide a generality and scalability of road database, we design the database by using the well-known Entity-Relationship Model. In order to simplify the database, we also use the abstracting method and then extracted 6 entities and 10 relationships. From these entities and relationship, the entity-relationship diagram is also proposed. **Improvements:** The design road database can also be extended by using other relational methods than Entity-Relationship model.

Keywords: Database, Entity-Relationship Model, Road, Self-Driving Vehicle

1. Introduction

Many technologies on the self-driving vehicle have been studied in the past 10 years[1–3]. Self-driving vehicles are autonomous ones driving from origin to destination automatically, taking care of themselves without any human intervention. It is possible only because of the recent researches in computing technologies and various sensors related to self-driving vehicles[4]. Self-driving vehicles are important in that they significantly reduce car accidents, save energy, and help save time during commuting[5]. Especially in the aging society, an self-driving vehicle will be inevitable.

According to WHO, there are 1.2 million deaths every year caused by car accidents. If the self-driving vehicle is commercialized, we will be able to significantly reduce deaths caused by car accidents. The damage and loss caused by car accidents - not just limited to financial loss but also social loss that cannot be quantified - are immense. Thus, the invention of a self-driving vehicle that can provide safe mobility is essential for the well-being of the public. A self-driving vehicle runs at the best quality of energy status, and thus contributes greatly to energy saving. According to a report from the EnoCenter[6] of the US, it was estimated that if 10% of the cars in the US are self-driving vehicles, about 37 billion dollars could be saved. In this aging society that we live in, a self-driving vehicle is essential. As nations all around the world are rapidly growing into an “elderly society”, naturally there are more and more elderly drivers who are less nimble and agile in dealing with unexpected situations. Thus, the importance of self-driving vehicles is increasingly highlighted. Globally, the number of elderly people is expected to be on a steady increase: 7.2% in 2010, 14.3% in 2018, 20.8% in 2026. Self-driving vehicles will be essential as a means of transportation for those people. Also, as more and more women are increasingly participating in economic activity, the number of women drivers is growing

**Author for correspondence*

too. Self-driving vehicles will be more than helpful in this global trend in that it can provide a vehicle security system that will prevent parking lot crimes and vehicle theft which women are especially vulnerable.

All around the world, a fervent competition to develop better self-driving vehicles is emerging. Until now, vehicle manufacturers have taken the lead in terms of automobile technology. However, the information technology business is taking lead in developing self-driving vehicles. Google and Nvidia[1], both IT companies that have started off as search engines, are representative instances. These companies are developing self-driving vehicles based on the high quality Graphics Processing Unit (GPU) and state-of-the-art sensors that help recognize nearby objects. Conventional vehicle manufactures are also changing automobile hearts to batteries and applying various smart functions to vehicles. For instance, there are many attempts lately to connect smart mobile devices such as smart phones or smart watches with vehicles. Google is taking an outstanding lead in this competition for a better self-driving vehicle. Last 2010, Google officially announced the plans regarding the development of self-driving vehicles. The early version of a self-driving vehicle, equipped with various sensors, camera, GPS on a Japanese company vehicle, is famous. Google also released the prototype of the self-driving vehicle in the December of 2014. Google introduced the prototype as "a self-driving vehicle almost as good as a real one". Unlike the previous type, the sensors have been miniaturized. Also, various functions for convenience have been added. The sensors that are attached on the roof of the Google self-driving vehicles are called the Light Detection and Ranging (LiDAR). New devices and technology such as the Google map and GPS have also been added. The core of Google self-driving vehicle technology is utilizing various sensors to reduce the blind spot of the car.

In self-driving vehicles, the road database is considered as the most fundamental elements in the self-driving vehicle[7]. This paper proposes the design of a road database for the self-driving vehicle. A self-driving vehicle will use the road database to route itself from the point of departure to the destination. Also, it will continuously compare and control its current location monitored through GPS with its location on the road database to evaluate if it is traveling in the correct path[7]. In a decision making module, the road database is used to predict and forecast the self-driving vehicle's movement. Road database in many navigating systems is currently used universally as auxiliary equipment[8–11]. The road database only serves to help safe driving under the assumption that a human is driving the vehicle. Thus, it is insufficient enough to be utilized for self-driving vehicles. For instance, existing navigations do not need to know the specific traffic lane which the vehicle is traveling. However, in a self-driving vehicle, information regarding the traffic lane of which the car travels is essential. A self-driving vehicle must first move to the appropriate traffic lane in order to perform an order. Also, the conventional road database does not include any data for self-driving drive in an intersection. However, in order for a self-driving vehicle to perform self-driving at an intersection, it is essential to have the coordinates for left and right turn. This paper will design a road database that overcomes the limits of conventional navigation road databases and therefore is appropriate for self-driving vehicle.

The chapter 2 will analyze requirements for a road database in order for it to be sufficient for self-driving vehicle. The chapter 3 will propose the data model for self-driving vehicle by using the well-known data modeling technology and give the diagram based on the proposed model. Finally, the chapter 4 will give the conclusions for the paper.

2. Requirements of Road Database

2.1 Self-Driving Vehicles and Road Database

A self-driving vehicle requires various core information technology. It requires not only the mechanical factors, but also the Advanced Driver Assistance Systems (ADAS), the Vehicle-to-Infra/vehicle/nomadic (V2X), and a detailed map database. The ADAS helps the driver to make decisions to increase the driver's ability to react to various circumstances. The V2X deduces information that cannot be detected through the sensors. It communicates with infrastructure and other vehicles to draw a conclusion about traffic information about the direction of progress of other vehicles. A detailed map database provides information about all static objects on the road. This helps the driver prepare for traffic lights or curves that are soon to come. In terms of constructing the detailed map, it is important to accumulate highly accurate regional data. In terms of utilizing the detailed map, it is important to develop a system that can match the data received from the sensor and the data already accumulated within the map[12]. Thus, the development of software that can amalgamate with the sensor data is crucial.

The electronic map for ADAS is gaining interest as a core factor the map-based ADAS. Currently most ADAS functions are based on a sensor, camera, laser. Once an electronic map that contains information of the road and geography is merged, the control accuracy is greatly increased. This is because the vehicle already knows where it is heading when it performs the detections. This is seen as the pre-phase of self-driving vehicles. Road database for an efficient traffic system has been accumulated in various ways. This is, however, focused on the efficient use and maintenance of roads. Thus, in order for this to be utilized in a self-driving vehicle, the unnecessary parts must be removed and some additional parts must be added. While conventional road databases are focused on the accumulation of static data, road databases for self-driving vehicles must focus on dynamic data such as roads that may be temporarily closed by construction as well as static data. This section discusses the requirements of a road database that can be optimized for self-driving vehicles.

2.2 Requirements of Road Database

There are lots of data regarding on roads. However, the data that is required for self-driving vehicles are as follows: Boundary lines of roads, road centerlines, crosswalks, traffic lights, bridges, intersections, multi cross roads, interchange, tunnels. These geographic features must be utilized according to the location of the vehicle in order for the vehicle to move. It is common that these features are modeled by nodes and links. In order to represent geographic features that lay ahead, nodes can be distinguished as intersection nodes, changing nodes of properties of road, facility nodes. Intersection nodes generally refer to the point where roads cross. The changing nodes of properties of road refer to points where property of a road changes. Those changes in properties of a road may be influenced by number of lanes, max speed limit, motorways, starting point of reversible lanes, end points of reversible lanes, and temporary closure due to construction. Facility nodes use nodes to be starting points and end points for all facilities that exist on the road. These facilities include high-level roads, tunnels, bridges, railroad crossings, tollgates. Link is information that connects node and node. A link is a line that represents the line of roadways. It is the line of a road connected by starting nodes and ending nodes. It has unique characteristics of each road. Thus links are distinguished by nodes. Links are determined by the definition of nodes. Links acquire a unique characteristic which may become crucial information in the flow of traffic. Information of intersection is crucial for a self-driving vehicle. An intersection is like a switch, connecting roads and roads. An intersection is considered as a dynamic road that appears and disappears. In this paper, an intersection is not limited to one node, but has as many nodes as the number of connected roads. Figure 1 is an example of an intersection and change of property of the road because of construction. There are 6 nodes and 8 (r_1~r_8) roads, where a black dot represents as a node. The read r_1 and r_5 has only one lane, while the others have two lanes. From the road r_1, it must be connected to r_2, r_4 and r_6 according to the traffic light. This paper asserts that r_9, r_{10} and r_{12} in addition to road r_1~r_8 must be accumulated in the database as a road in order for this connection to be smooth. Those additional roads are also generated for r_3, r_5, and r_7. In Figure 1, there are two nodes with related to the changing node of property of road, marked as "under construction".

Usually, a road environment is composed of intersections and roads that connect intersections. Also, in any intersection or road, there are crosswalks for pedestrians. There may be traffic lights between crosswalks and intersections. An intersection is a point where multiple roads are connected. An intersection acts as a switch that connects road to road on a certain based on traffic lights. There intersection can be assigned to each connected switch roles at different times. A road connects an intersection to another intersection. There are straight lined roads and curved roads. A road is composed for more than one lane. Each road has different lengths. Each road has the different maximum speed. Such road environments can be modeled as a graph. In the graph model, an intersection is considered as a node, and roads are considered as a link.

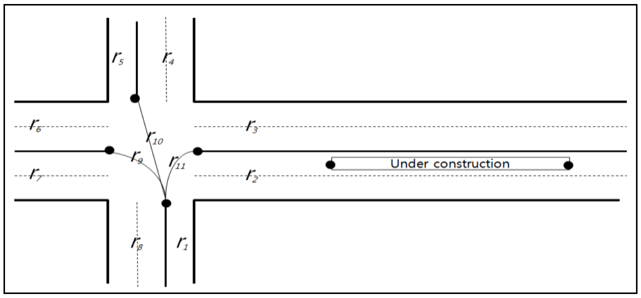

Figure 1. Roads in intersection.

Information regarding intersections must be stored in the node, and information regarding roads must be stored in the link. The road database for self-driving vehicles must be modeled by the graph.

In order for conventional road databases to be utilized as self-driving vehicle road databases, additional information must be added as following. Information about curved roads that are used to connect two disconnected roads at an intersection must be stored. Information about time intervals for each connection of two disconnected roads must be stored. The location of the starting point and ending point must be stored. This is used to calculate the distance from an intersection. Information about the waypoints for curved roads must be stored. In order for one-way lanes and two-way lanes to be differentiated, additional information must be stored. The area of the lane must be stored in order to exactly compute the location of the lane.

3. Design of Road Database

3.1 Entity Relationship Model in Road Database

This paper uses the Entity Relationship Model (ERM) in order to design a road database for the self-driving vehicle. The ERM has been enhanced by many scholars since it was proposed first by[12] to help the designing of various databases[13]. Currently the Enhanced Entity Relationship Model (EERM) is widely used in the database designing process. The model, as a popular model for conceptual designing, abstracts in a high level and is easy to understand and has exceptional expressiveness of statements and thinks in the perspective of a human, and is compatible with various CASE tools.

The ERM represents the real world in terms of relationships amongst entities, attribute, and entities. The ER diagram graphically expresses the entity type, relationship type, and the attributes of entities and relationships. The ERM is easily learned and is easy to understand even without profession of the area. It is more independent in implementations and more formal than natural languages. All these factors make the ERM suitable for the database designers in communicating with the final users.

Based on the requirements mentioned above, an entity-relationship model is used to the method of design in this paper. As entities that must be manipulated, we concern intersections, roads, routing at intersections, waypoints at roads. For intersection entities, information about the center of the intersection is needed. For routing at intersections, information about the routing order, allowed time interval must be stored. One intersection connects multiple roads. A road should be represented as the starting point of the intersection and the ending point of the intersection. And also it should have the points for multiple waypoints, number of lanes, width of lane, and multiple maximum speeds. On the other hand, one road must be connected with two intersections. In a case where a road is not connected with any other intersection, the ending point is also considered as an intersection. One road includes multiple waypoints. A waypoint must always be included in a road. The waypoint must store the related order of the road. One routing must be included in one intersection. One routing must store connection order information of the related intersection. One intersection must include multiple routings. One routing includes multiple waypoints. One waypoint must be included in one routing or one road. One waypoint must include order information about the related routing.

3.2 Entities and Attributes in Road Database

An entity refers to an object that exists independently and can be distinguished: A person, location, incident, etc. An entity is a shapeless or shaped object that has information about a particular target. An entity may be substantial but may also be an abstract thing such as a thought or concept.

There are numerous entities in a road database. However, not much of it is required for a database used for self-driving vehicles. The most fundamental information that is needed for self-driving is the current location of the vehicle. Data of the current location is also the most fundamental data when constructing a map. Thus, the location data is the entity that needs the most regular maintenance. The main attribute of such location entity is the latitude and longitude. Latitude and longitude must be the key for this entity, but it is better to give a code attribute as the primary key when considering the reference from other entities[14].

A road is basically considered as a graph. Thus, there are nodes that are like intersections, but there are also roads that connect one intersection to another. When an intersection corresponds to a node, the road corresponds to a link. Thus, node and link information are also entities that need to be stored in the road database. The most important factor that the self-driving vehicle requires is being able to find the shortest path from the location to the destination. One advantage about modeling roads into graphs is that A* algorithm can be applied for searching the routing. As mentioned above, there are various nodes related to roads. The point where property of the road changes can also become a node. These properties include change in number of lanes, change in the limit speed of the vehicle and reversible lanes. These properties are reflected on the driving of self-driving vehicles and thus must be stored as entities. Also, nodes can be created by various facilities existing on the road. These facilities include tunnels, bridges and etc. Attributes of node include location data, information on other properties. Location information may be used as a key for nodes, but just like location entities, a separate code is better to be made as the primary key for the nodes.

Links - roads that are represented as entities - can also be largely divided into two entities. One is the actual road that represents the road itself. The other is the additional road that is like tunnels, bridges, and changing section of properties. One actual road may have the relation with several additional roads. Especially, because a link of an actual road must express the current situation of a road, it must include several locations information to represent curved roads. As an entity that includes these various location data, the waypoint entity must be stored. A code attribute of link entity must be made and used as the primary key to store links. Attributes of a link entity include the starting node, ending node, number of lanes, speed limits and etc.

Important data regarding the self-driving vehicle include traffic lights and crosswalks. Traffic lights and crosswalks also have location data as attributes. For efficient references, A code attribute of traffic light and crosswalk entity should be made to be used as primary keys. Traffic lights need sequential attributes because it requires sequences of other traffic lights. Traffic lights also need information regarding the time interval of traffic lights. Crosswalks need attributes that deal with the width of each crosswalk. Table 1 summarizes the entities needed for a road database and the attributes of each entity.

3.3 Relationships in Road Database

Relationships represent the relation between one entity and another. It is a method that maps the correspondence of various elements of the group. Relationships include One-to-One, One-to-Many, Many-to-Many mapping.

To simplify relationships, a method called as abstraction is used. As shown in Table 1, there are three different nodes such as intersection node, changing node, facility node. These nodes are connected to links. These nodes also have a common attribute of location information and code of node. Thus these nodes can be abstracted into one entity called a node. It simplifies the model, making it easier to understand. Also, there are various type links including actual speed link, attribute change link, facility link which all must be connected to two nodes. Thus, these links are also abstracted into one entity called a link.

One node is connected to several links. An intersection is a representative example. An intersection is connected by many roads. Not all roads are two-way, thus each way of a road must be represented as a single link. Thus every link has a relationship: Connection between the starting node and ending node. The relationship of a node, on the other hand, is that it is connected to several links. A node

Table 1. Summary of entities and attributes for road database

Name of entities	Attributes
Location	Code , Latitude, Longitude
Intersection node	Code, Location, Other
Changing node	Code, Location, Changing properties
Facility node	Code, Location, Facilities
Actual road link	Code, Starting node, Ending node, Number of lane, Maximum Speed
Changing link	Code, Starting node, Ending node, Property type, Property
Facility link	Code, Starting node, Ending node, Facility type, Facility
Waypoint	Location
Traffic Light	Code, Location, Sequence, Interval
Crosswalk	Code, Location, Width

must have location information. One node must have a relationship with one location.

A link must have several waypoints in order to express the curved lines of a road. Thus one link has a relationship of One-to-Many with waypoints. Also, actual roads may include several changing links, facility links. For example, suppose that one road has several bridges and tunnels. It also has various speed limits. All facility links and changing links related to the actual road must be able to be searched. It implies that one link can be related as several children links.

One link has relationships with several traffic lights, while one traffic light must have a relationship with one link. This relationship applies as the same for crosswalk. Traffic lights and cross walks also need location information so they must have One-to-One relationships with location. A waypoint must have location information. Thus, one waypoint must have a relationship with one location information. One location information has relationship with multiple waypoints. Most waypoints and location information have a One-to-One relationship. However, points where high-level roads cross may be used as two different waypoints. This relationship also occurs at intersections.

3.4 Entity Relationship Diagram of Road Database

An Entity Relationship Diagram (ERD) is a graphical representation of an information system that shows the relationship between people, objects, places, concepts or events within that system. An ERD is a data modeling technique that can help define business processes and can be used as the foundation for a relational database[13].

Figure 2 is an ERD for road database. The abstraction of nodes and links were not represented in Figure 2. In Figure 2, the squares refer to entities. The lines that connect the squares refer to relationships. The most fundamental attributes of entities were only represented in Figure 2. Additional attributes may be easily added later on the ERD. The solid lines indicated that there must be a relationship. The dotted lines indicate that a relationship may or may not exist. The bird foot shape refers to many. All others mean One. PK stands for primary key. FK stands for foreign key. PF refers to one that is foreign key and primary key at the same time.

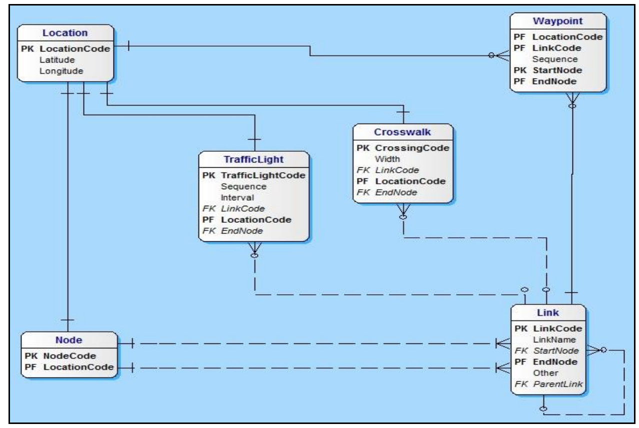

Figure 2. The entity-relationship diagram.

4. Conclusions

Many technologies on the self-driving vehicle have been studied in the past 10 years. It is because self-driving vehicles has advantages in that they significantly reduce car accidents, save energy, and help save time during commuting. Especially in the aging society, an self-driving vehicle will be inevitable. In this paper, we study the road database that can be considered as the most fundamental elements in the self-driving vehicle. It is very important because the self-driving vehicle uses the road database not only to predict and forecast its movement, but also to check traveling the correct path. Road database in many navigating systems is currently used universally as auxiliary equipment. The road database only serves to help safe driving under the assumption that a human is driving the vehicle. It implies that the conventional road database is not proper for self-driving vehicles. In order to design the road database, we introduces the requirements of the road database for a self-driving vehicle. From the requirements, we designed the database by using ERM. We extract 10 entities and 10 relationships from the requirements as shown in Table 1. We use the abstracting method in order to simplify the database. Finally we proposed 6 entities and 10 relationships. However, the essential attributes of each entity are proposed, but they are easily added or removed to the database. As further researches, we will implement the sample database for the specified region based on the proposed design.

5. Acknowledgment

This research was financially supported by Hansung University during sabbatical year (2015).

6. References

1. Guizzo E. How Google's self-driving car works. IEEE Spectrum Online; 2011 Oct.
2. Wei L, Soheilian B, Gouet-Brunet V. Augmenting vehicle localization accuracy with cameras and 3D road infrastructure database. In Computer Vision-ECCV Workshops; 2014. p. 194–208.
3. Singh AP, Agarwal PK, Sharma A. Road safety improvement: A challenging issue on Indian. International Journal of Advanced Engineering Technology. 2011 Apr-Jun; 2(2):363–9.
4. Budhathoki NR, Haythornthwaite C. Motivation for open collaboration crowd and community models and the case of OpenStreetMap. American Behavioral Scientist. 2013 May; 57(5):548–75.
5. The eno center for transportation. Available from: https://www.enotrans.org
6. Hu Y, Gong J, Jiang Y, Liu L, Xiong G, Chen H. Hybrid map-based navigation method for unmanned ground vehicle in urban Scenario. Remote Sensing. 2013 Jul; 5(8):3662–80.
7. Jo K, Sunwoo M. Generation of a precise roadway map for autonomous cars. IEEE Transactions on Intelligent Transportation Systems. 2014 Jun; 15(3):925–37.
8. Gruyer D, Rachid B, Marc R. Accurate lateral positioning from map data and road marking detection. Expert Systems with Applications. 2016 Jan; 43:1–8.
9. Kamboj D, Kumar V, Vaid R. Secure and authenticated vehicle navigation system. Indian Journal of Science and Technology. 2015 Oct; 8(28):1–7.
10. Rizwan JM, Krishnan PN, Karthikeyan R, Kumar SR. Multi layer perception type artificial neural network based traffic control. Indian Journal of Science and Technology. 2016 Feb; 9(5):1–6.
11. Fendi KG, Adam SM, Kokkas N, Smith M. An approach to produce a GIS database for road surface monitoring. APCBEE Procedia. 2014 Dec; 9:235–40.
12. Chen P. The entity-relationship model - Toward a unified view of data. ACM Transactions on Database Systems. 1976 Mar; 1(1):9–36.
13. Rajakumari SB, Nalini C. An efficient data mining dataset preparation using aggregation in relational database. Indian Journal of Science and Technology. 2014 Jun; 7(S5):44–6.
14. Tseng FH, Liang TT, Lee CH, Der Chou L, Chao HC. A star search algorithm for civil UAV path planning with 3G communication. 2014 10th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP); 2014 Aug. p. 942–5.