ICIC Express Letters
Part B: Applications
Volume **13**, Number **6**, June **2022**

ICIC International ©2022 ISSN 2185-2766
pp. **597–605**

A BINARY DECISION DIAGRAM BASED APPROACH FOR REFINING ROAD SAFETY SCENARIOS IN THE LOCAL DYNAMIC MAP

Arvind Kumar[1], Keita Kawano[1], Huynh Le Phu Trung[1]
and Hiroaki Wagatsuma[1,2]

[1]Graduate School of Life Science and Systems Engineering
Kyushu Institute of Technology (Kyutech)
2-4 Hibikino, Wakamatsu-Ku, Kitakyushu 808-0196, Japan
{ kumar.arvind510; keita.kawano556; huynh.trung-le437 }@mail.kyutech.jp

[2]RIKEN Center for Brain Science (RIKEN CBS)
2-1 Hirosawa, Wako, Saitama 351-0198, Japan
waga@brain.kyutech.ac.jp

Received October 2021; accepted December 2021

Abstract. *In international automobile communities, cooperative, connected and automated mobility have been discussed. For actual implementations of those schemes in the road infrastructure, an effective separation of information types such as static geographical properties and dynamic vehicle movements and its integration is necessary to prevent potential accidents in an early stage. As a de facto international standard in progress, European Commission has supported the layered data structure depending on the spatio-temporal characteristics of things on the road to manage static and dynamic data with different updating time cycles, known as Local Dynamic Map, and recently technical problems to realize its actual implementation were discussed as issues need to be solved. In this paper, we proposed a compact representation method of geographical vehicle positions by using Binary Decision Diagrams (BDDs) integrated with the Geohash geographical encoding of target vehicles in the aim of a significant reduction of computational cost to prevent a serious communication delay. In the computer experiment, our proposed method was validated in the platform of the Robot Operating System (ROS). The comparative analysis with other traditional methods demonstrated that the proposed method clearly reduced the computational time three times smaller than others in cases of two vehicles and the same tendency was observed even if the number of vehicles increases. Therefore, the BDD-based formulation might be applicable not only for an effective geographical encoding but also for conflict management in traffic scenarios.*

Keywords: Local Dynamic Map, Binary decision diagram, Cooperative intelligent transport system, Lanelet, CoincarSIM

1. Introduction.

Recently, automated driving is a hot topic. According to SAE (Society of Automotive Engineers) International in the United States, the international standard SAE J3016 was published in 2014 in the first place, revised three times and released in the form of the latest version in 2021 [1], which defines ADAS (Advanced Driving Assistance System) and ADS (Automated Driving System) clearly. Simultaneously, European Commission has strongly promoted industry-government-academia research group projects on intelligent transport systems in the framework of Horizon 2020, which focused on domains of CCAM (Cooperative, Connected and Automated Mobility) and C-ITS (Cooperative Intelligent Transport Systems) [2, 3] with white papers and reports in 2017-2018. In the aim of the establishment of sustainable mobility extended from technological aspects, they

---

DOI: 10.24507/icicelb.13.06.597

have encouraged a developmental process from three evolutions as C-ITS, CV (Connected Vehicles) and AV (Automated Vehicles) to an effective CCAM realization in society. One of the interesting common framework designs is the concept of the Local Dynamic Map (LDM) which realizes a plausible C-ITS in consideration of Vehicle-to-Vehicle (V2V) communications, Vehicle-to-Infrastructure (V2I) communications and Vehicle-to-Person (V2P) communications to prevent potential accidents. Therefore, nearby vehicle detection in possible traffic scenarios is the most frequent communication in the actual LDM implementation [4-6]. As Yeo et al. [7] demonstrated, sensor-based vehicle detection techniques were developed as to be sufficient to an automatic traffic monitoring, which is a basement technology of the LDM. In the consideration of the LDM implementation in the system level, GIS (Geographical Information Systems) database is necessary for managing multiple vehicle locations and SQL (Structured Query Language) queries assist to detect potential risks in accordance with traffic scenarios. In the C-ITS station, road conditions and actual situations will be monitored based on the world model [8, 9]. Indeed, PostgreSQL [10] and PostGIS [11] are frequently introduced for the LDM database as examined in the work of Shimada et al. [4]. They implemented the LDM as a part of the SAFESPOT project [8] and evaluated its response time of SQL queries with respect to the number of vehicles in the collision detection task. Although the LDM requires the layered data structure for improvements in transparency and accountability, the complexity of data structure lies a potential problem on limitations of the computation time in the database. As Eggert et al. [5] noticed, a single layered structure is simple and effective to minimize necessary computation time, and then they proposed a fully interconnected graph-based representation to realize the Relational Local Dynamic Maps (R-LDM) based on a world model by interconnected structures on spatial, temporal, and semantic attributes for ADAS and ADS analyses. A breakthrough is expected to solve the trade-off problem between enrichment capabilities in layered data structure and reduction strategies in computational cost.

Interestingly, there is another trend in safety-critical software and hardware system evaluations by using formal verification techniques such as Linear Temporal Logic (LTL) and symbolic model checking [12]. The formal verification has been developed as a method for inspections of systems prior to release and shipment, and recently it is highlighted to be an effective method embedded in systems in operation and social infrastructures to detect safety-critical events according to potential scenarios [13-15] including cases focusing on road traffic rules and scenarios [16-18]. In the field of LTL and model checking, a Binary Decision Diagram (BDD) is frequently introduced to minimize computational cost [19, 20], which can provide a compressed representation of the sets as relations. If a BDD based minimization of the computational cost in critical event detections for moving vehicles in dynamic environments, it will be a breakthrough to solve the trade-off problem in LDM and contributes to enhancing an actual implementation of LDM accessing the Relational Database Management System (RDBMS) even it is a layered data structure.

According to the hypothesis that an effective BDD based minimization of the computational cost is possible under the condition that a suitable GIS representation can be proposed, this paper focused on its realization and the comparative analysis of computation time between cases of the traditional way for representing geographical vehicle positions for the nearby vehicle detection and the proposed way. The rest of the paper is structured as follows. Section 2 introduced the LDM concept and basement technologies including BDD. Section 3 presented our hypothesis and essential points of the proposed method. Section 4 described experimental conditions and computer experiments demonstrated in the ROS (Robot Operating System). Finally, Section 5 summarized the results and the future direction as conclusion.

2. **Local Dynamic Map (LDM) and Basement Technologies.**

2.1. **LDM concept.** As described above, LDM is a critical component for the realization of C-ITS that was standardized by ETSI (European Telecommunications Standards Institute) [21] and ISO (International Organization for Standardization) [22], because the system has to integrate various information spatial and temporal domains. Thus, V2V and V2I commonly require geographical positions of things in the road with respect to the environmental (road) conditions and circumstances. For example, the SAFESPOT project [8] introduced LDM scheme for the management of traffic scenarios. The LDM consists of the following four layers as follows:

Layer 1. (Static) Information which is basically sustainable such as road construction and its geographical information for the navigation (e.g., roads, lanes, buildings and other yearly updating information);
Layer 2. (Semi-static) Information which is expected to be updated depending on road conditions (e.g., traffic bans in plan, road repairing in plan, broad-area weather forecasting and other monthly updating information);
Layer 3. (Semi-dynamic) Information which is updated depending on moving object conditions (e.g., traffic accidents, road repairing, weather forecasting in a local area and other daily updating information);
Layer 4. (Dynamic) Information like ITS predictions on vehicles, pedestrians, traffic signals and other highly dynamic objects (e.g., vehicles, motorcycles, bicycles, pedestrians and other sec/min/hour-order updating information).

2.2. **Geohash.** Geohash is a public domain geocode system presented by Niemeyer [23] in 2008 and a similar proposal was done by Morton [24] in 1966. In the code, a geographical position is represented by strings composed of letters and digits and the string length controls the size of the target geographical area as shown in Table 1.

Table 1. Geographical size of Geohash encoding

#Label in Geohash	Distance in north and south [m]	Distance in east and west [m]	An Geohash example
1	4989600	4050000	w
2	623700	1012500	wy
3	155925	126562.5	wyh
4	19490.625	31640.625	wyhb
5	4872.65625	3955.07813	wyhby
6	609.082031	988.769531	wyhby3
7	152.270508	123.596191	wyhby3k
8	19.0338135	30.8990479	wyhby3kf
9	4.75845337	3.86238098	wyhby3kf5
10	0.59480667	0.96559525	wyhby3kf5f
11	0.14870167	0.12069941	wyhby3kf5fs
12	0.01858771 ($\approx$ 1.86 [cm])	0.03017485 ($\approx$ 3.02 [cm])	wyhby3kf5fst

Therefore, Geohash provides a discrete representation of geographical positions and the resolution of the spatial mesh can be flexibly decided depending on requirements of the target problem.

2.3. **BDD and related methods.** BDD is a data structure to represent a Boolean function in the form of a graph [19, 20], which can provide a compressed representation of the sets as relations. As a Directed Acyclic Graph (DAG), the graph-based representation

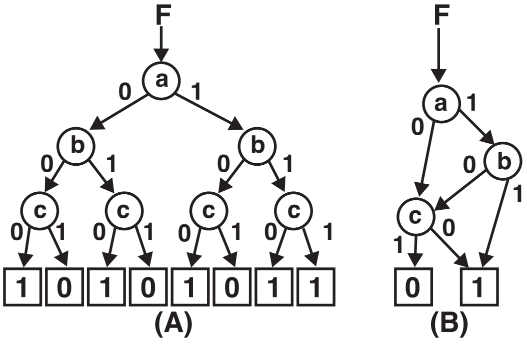


Figure 1. Boolean functions to represent $F = (a \wedge b) \vee \neg c$ in cases of the binary tree (A) and a compact BDD (B) [28]

consists of two terminal nodes, as 'false (0)' or 'true (1)', and branch nodes. Each branch node represents a Boolean variable such as x_i which has two children branch nodes as 'low' and 'high' separately (Figure 1). In a well-ordered BDD, all variables from a root node to children nodes can be represented systematically and the reduced BDD can be obtained by minimizing of redundancy in the representation according to the following rules:

1) Merge any isomorphic subgraphs;
2) Eliminate any nodes to have two children nodes as isomorphic.

As a more sophisticated way, ROBDD (Reduced Ordered Binary Decision Diagram) [19, 20] can be introduced. ROBDD is always canonical, which provides a unique representation for the given variable order. Further discussion for the compaction can be found [25], which was highlighted in a significant reduction of computational cost in quantum computing by using a specific method called DD package [26].

In the consideration of minimization of computation time, a Shared Binary Decision Diagram (SBDD) can be used. It is a representation of the target multiple-output function, the nodes are shared among BDDs representing the various outputs and a partitioned SBDD consists of two or more SBDDs that share nodes, which can be obtained systematically in an optimized way [27]. The reduction of nodes to represent the same function indicates the reduction of computation time.

3. Hypothesis.

3.1. Summary of the hypothesis.

In consideration of a minimization of the geographical position representation to reduce computational cost including operations in the risk detection, we focused on two points as:

(i) Area-based detection of nearby objects, instead of point-based positions;
(ii) Logical operations by Boolean functions, instead of operations with the floating-point arithmetic processing.

In the point (i), it enables the minimization of the data size if the accuracy level is clearly defined and guaranteed. In the point (ii), there is a possibility to reduce a computational cost if the shared binary decision diagram can be introduced for the detection of nearby objects. In this sense, it can be hypothesized that the encoding of geographical positions into Boolean values and a calculation in the form of Boolean functions are beneficial to reduce the computational time [29, 30].

In the work of Shimada et al. [4], they implemented the LDM and evaluated its response time of SQL queries with respect to the number of vehicles in the collision detection task. The collision detection of vehicles was relying on a classical warning algorithm for rear-end collisions proposed by Burgett et al. [31], which was initially designed for providing an alert to human drivers. The warning algorithm calculates the stopping distance derived from vehicle speeds, acceleration/deceleration and vehicle-to-vehicle local distance measured by on-vehicle sensors, which presumably require a floating-point arithmetic processing of sensor real values if it is implemented in a V2V system. Thus, the computational cost needs to be evaluated in comparison with cases of the floating-point real value representation of geographical positions in the world model.

3.2. **Proposed method as an integration of Geohash and BDD.** According to reports to propose ADS international standards [1, 2], it is known that a high-precision 3D map is formulated as a cm-resolution of the geographical information [32]. Referring to Geohash (Table 1), it indicates that 12 digits/letters are enough for the representation of moving entities in the sense of ADS and C-ITS, which corresponds to the bottom of Table 1. By using Geohash encoding, a nearby position detection among different objects is easily verified by the comparison of the Geohash code as a string, the accuracy level can flexibly change depending on the number of digits/letters in the code.

In the proposed method, a single SBDD (Figure 1) was used for the integration of the Geohash-based geographical representation, which is able to verify multiple vehicle positions and then the graph representation is optimized by using the SBDD procedure.

For the integration in the proposed method, a Geohash code was converted from characters (a-z) and digits (0-9) to the binary representation by using a conversion table as '0':(00 000), …, '9':(01 001), 'b':(01 010), …, 'z':(11 111), which eliminates specific characters as {'a', 'i', 'l', 'o'} according to the Geohash definition. Thus, five Boolean variables were applied to representing each character of the target Geohash code, and then the converted Boolean representation represents a geographical position in the world model consistently. Depending on the given length of the Geohash code, the position accuracy is determined. For the utilization to the LDM vehicle management in the proposed method, procedures were designed as follows.

1) Geographical position conversion: (latitude, longitude) to Geohash code, e.g., $(33.88919521551, 130.71065559849)$ is represented as Geohash of 'wyhby3kdbeyd'.
2) Boolean expression conversion of Geohash code for the BDD procedure as hashing, e.g., Boolean expression for Geohash 'wy' as a part of the given code is '$(x1 \wedge x2 \wedge x3 \wedge \sim x4 \wedge \sim x5) \vee (x1 \wedge x2 \wedge x3 \wedge x4 \wedge \sim x5)$'.
3) Construction of the shared BDD for a given Boolean expression. To combine two (or more) positions of target vehicles, a disjunction operation is used in the BDD, which is equivalent to a union operation in the set theory.

4. Computer Experiment and Results.

4.1. **ROS based implementation of the LDM framework.** Lanelet road network framework [33-35] was modified for the current purpose to integrate the CoInCar-Sim simulator [36] to manage road scenarios, and a geographical map for vehicles was derived from the open-source map project known as Open Street Map (OSM). The OSM data format was transformed to our customized Lanelet system working on the Robot Operating System (ROS) as shown in Figure 2.

4.2. **Numerical comparisons.** For the validation, we compared computational costs in multiple conditions. The experiments were done in the computer with Intel(R) Core(TM) i9-9900K CPU (3.60GHz) having 64 GB RAM. Different calculation methods were applied to the validation of computational costs.

For the validated comparison, at least, three conditions are necessary, such as a condition equivalent to the traditional implementation as demonstrated by Shimada et al. [4] (floating-point real value representation: 'F'), discrete spatial representation as Geohash (Geohash only: 'G') and the proposed method as the integration of BDD and Geohash

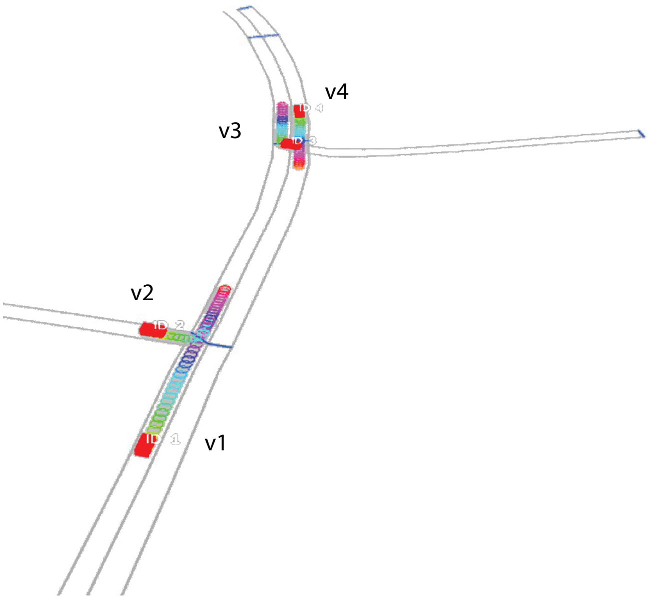
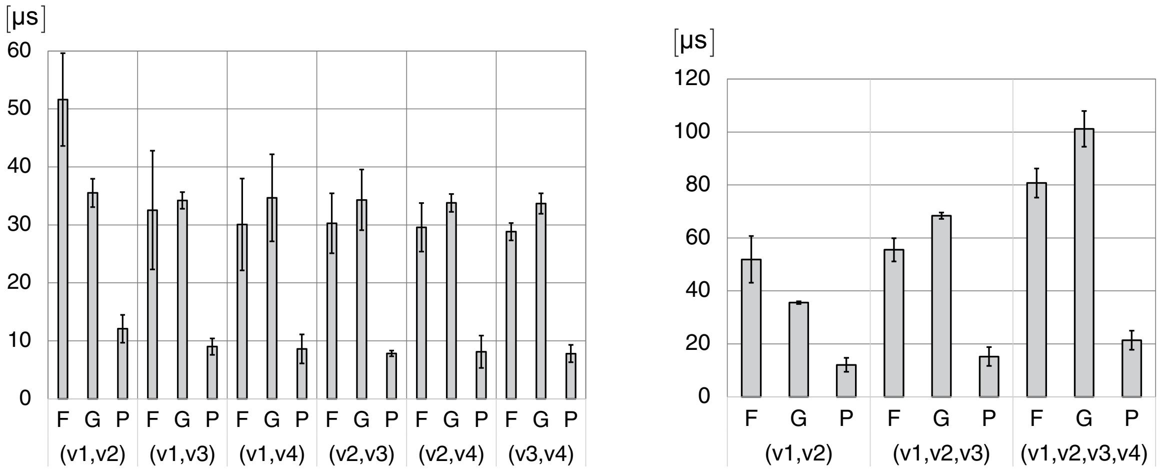


Figure 2. An ROS-based demonstration of the LDM system with four vehicles moving on the road, which was implemented in the modified Lanelet associated with the CoincarSIM. It was used for the validation framework of the proposed method. v1, v2, v3 and v4 represent respectively vehicle1, vehicle2, vehicle3 and vehicle4.



(a) Two-vehicle interactions

(b) N vehicle interactions $(N = 2, 3, 4)$

Figure 3. Comparison among different calculation methods in computational costs. Position verifications of multiple vehicles by using the floating-point number (F), Geohash code string without BDDs (G) and the proposed method (P) were shown in each panel. Each average elapsed time was obtained from 100 trials in each condition and the error denotes the standard deviation.

(proposed method: ‘P’) as discussed in Section 3.1. The comparison between ‘F’ and ‘G’ indicates the effectiveness of discrete representations of spatial locations with respect to the continuous representations. The comparison between ‘G’ and ‘P’ indicates the effectiveness of the BDD implementation with respect to the calculation without BDD.

Figure 3 showed the average time to compute from 100 trials in each condition. In the condition of the floating-point number comparison (‘F’ in the figure panel), geographical positions of vehicles were represented by the floating-point number, and the ordinary arithmetic calculation was used for the detection of nearby vehicles. In the Geohash condition without any BBD scheme (‘G’ in the figure panel), geographical positions of vehicles were represented by the Geohash encoding as a string and the string operation was used for the detection of nearby vehicles. The proposed method (‘P’ in the figure panel) was implemented as described in Section 3.1.

Interestingly, computation time in conditions of F and G was almost three times larger than the proposed method in the case of two-vehicle interactions (Figure 3). The tendency was consistent in any combinations of vehicles (Figure 3(a)). In the comparison with various numbers of vehicles, the proposed method (‘P’) took the computation time as 12.1 [μs] (two vehicles), 15.23 [μs] (three vehicles), and 21.43 [μs] (four vehicles). This increase rate was significantly low in comparison with other methods as shown in Figure 3(b). This result clearly proved that our hypothesis was valid and the proposed method effectively reduces the necessary computation time even with an increase of the number of vehicles.

5. **Conclusion.** We hypothesized that the integration of Geohash encoding of geographical positions of vehicles and the shared BDD minimizes the computation time in the verification of multiple vehicle positions, and successfully established the testable framework based on ROS with modified Lanelet system [33-35] and CoInCar-SIM-based scenario manager [36]. Results of computer experiments clearly demonstrated the effectiveness to reduce the necessary computational time in comparison with other conventional conditions.

This fact implies that Geohash or discrete representations in the encoding of vehicle conditions is effective if the shared BDD can be applied. In the analysis of the present study, we simply applied the proposed method to the position information, while it may have a new capability for inclusion of other types of information of vehicles in the further analysis. For example, the string code concept can be upgraded to extended representations with various vehicle conditions on the road. Integrative representations with action information (or other semantic information) can be considered such as going-straight, turning-left, or turning-right. The detection of various interactions can be verified more specifically with minimum computational cost if semantic and contextual information is represented in a discrete way, for example, ‘in-the-same-lane’, ‘in-different-lane’, and ‘approaching-to-intersection’. Therefore, this approach can contribute to the actual implementation of the LDM framework in multiple aspects.

Acknowledgment. This work was supported in part by JSPS KAKENHI (16H01616, 17H06383) and the New Energy and Industrial Technology Development Organization (NEDO).

REFERENCES

[1] SAE International, *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, https://www.sae.org/standards/content/j3016_202104/, Accessed on 2021-4-30.

[2] Administrator of the CAR 2 CAR Communication Consortium, *CAR 2 CAR Communication Consortium*, https://www.car-2-car.org/about-c-its/, Accessed on 2021-11-20.

[3] H. Makano, *C-ITS and Connected Automated Driving in Japan*, Routes/Roads 2017, http://www.nilim.go.jp/lab/qcg/japanese/3paper/pdf/2017_13.pdf, Accessed on 2021-11-20.

[4] H. Shimada, A. Yamaguchi, H. Takada and K. Sato, Implementation and evaluation of local dynamic map in safety driving systems, *J. Transp. Technol.*, vol.5, no.2, pp.102-112, 2015.

[5] J. Eggert, D. Salazar, T. Puphal and B. Flade, Relational local dynamic maps for driving situation analysis, *Proc. of FAST-Zero Symposium*, 2017.

[6] T. Eiter, H. Füreder, F. Kasslatter, J. Parreira and P. Schneider, Towards a semantically enriched local dynamic map, *International Journal of Intelligent Transportation Systems Research*, vol.17, no.2, pp.32-48, 2019.

[7] B.-C. Yeo, W.-S. Lim and H.-S. Lim, Vehicle detection for thermal vision-based traffic monitoring system using principal component analysis, *International Journal of Innovative Computing, Information and Control*, vol.12, no.5, pp.1467-1480, 2016.

[8] L. Andreone, R. Brignolo, S. Damiani, F. Sommariva, G. Vivo and S. Marco, *SAFESPOT Final Report*, Technical Report, D8.1.1, 2010.

[9] B. Netten, L. Kester, H. Wedemeijer, I. Passchier and B. Driessen, DynaMap: A dynamic map for road side ITS stations, *Proc. of ITS World Congress*, 2013.

[10] *PostgreSQL: The World's Most Advanced Open Source Relational Database*, https://www.postgresql.org/, Accessed on 2021-11-20.

[11] *PostGIS: Spatial and Geographic Objects for PostgreSQL*, https://postgis.net/, Accessed on 2021-11-20.

[12] K. Rozier, Linear temporal logic symbolic model checking, *Computer Science Review*, vol.5, pp.163-203, 2011.

[13] H. Kugler, D. Harel, A. Pnueli, Y. Lu and Y. Bontemps, Temporal logic for scenario-based specifications, *Proc. of the 11th International Conference of Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, pp.445-460, 2005.

[14] S. Coogan, E. A. Gol, M. Arcak and C. Belta, Traffic network control from temporal logic specifications, *IEEE Trans. Control of Network Systems*, vol.3, no.2, pp.162-172, 2016.

[15] A. Souri, A. Rahmani, N. Navimipour and R. Rezaei, A symbolic model checking approach in formal verification of distributed systems, *Human-Centric Computing and Information Sciences*, vol.9, 2019.

[16] Y. E. Sahin, R. Quirynen and S. D. Cairano, Autonomous vehicle decision-making and monitoring based on signal temporal logic and mixed-integer programming, *Proc. of 2020 American Control Conference (ACC)*, pp.454-459, 2020.

[17] S. Maierhofer, A.-K. Rettinger, E. Mayer and M. Althoff, Formalization of interstate traffic rules in temporal logic, *Proc. of 2020 IEEE Intelligent Vehicles Symposium (IV)*, pp.752-759, 2020.

[18] A. Kumar and H. Wagatsuma, An implementation of linear temporal logic for driving safety suitable for the concept of local dynamic map, *Proc. of 2021 IEEE/SICE International Symposium on System Integration (SII)*, pp.708-709, 2021.

[19] R. E. Bryant, Graph-based algorithms for Boolean function manipulation, *IEEE Trans. Computers*, vol.C-35, no.8, pp.677-691, 1986.

[20] R. E. Bryant, Binary decision diagrams: An algorithmic basis for symbolic model checking, in *Handbook of Model Checking*, E. M. Clarke, T. A. Henzinger, H. Veith and R. Bloem (eds.), Cham, Springer, 2018.

[21] *Intelligent Transport Systems – Extension of Map Database Specifications for Local Dynamic Map for Applications of Cooperative ITS*, Technical Report, ETSI EN 302 895 (V1.1.0), 2014.

[22] *Intelligent Transport Systems – Cooperative Systems – Definition of a Global Concept for Local Dynamic Maps*, Technical Report, ISO/TS 18750, 2015.

[23] G. Niemeyer, *Enhancements on geohash.org*, https://blog.labix.org/2008/03/01/enhancements-on-geohashorg, Accessed on 2022-04-24.

[24] G. M. Morton, *A Computer Oriented Geodetic Data Base and a New Technique in File Sequencing*, Report in IBM Canada, 1966.

[25] R. Ebendt, G. Fey and R. Drechsler, *Advanced BDD Optimization*, Springer, 2005.

[26] A. Zulehner, S. Hillmich and R. Wille, How to efficiently handle complex values? Implementing decision diagrams for quantum computing, *International Conference on Computer-Aided Design*, 2019.

[27] M. Matsuura, T. Sasao, J. Butler and Y. Iguchi, Bi-partition of shared binary decision diagrams, *IEICE Trans. Fundamentals of Electronics*, vol.E85-A, no.12, pp.2693-2700, 2003.

[28] S. Minato, BDD and its applications, *Bulletin of the Japan Society for Industrial and Applied Mathematics*, vol.9, no.3, pp.194-206, 1999 (in Japanese).

[29] K. Havelund, D. Peled and D. Ulus, First-order temporal logic monitoring with BDDs, *Formal Methods in System Design*, vol.56, pp.1-21, 2020.

[30] K. Havelund and D. Peled, BDDs for representing data in runtime verification, *Proc. of the 20th International Conference of Runtime Verification (RV2020)*, 2020.

[31] A. L. Burgett, A. Carter, R. Miller, W. G. Najm and D. L. Smith, A collision warning algorithm for rear-end collisions, *Proc. of the 16th International Technical Conference on Enhanced Safety of Vehicles (ESV)*, pp.566-587, 1998.

[32] H. G. Seif and X. Hu, Autonomous driving in the iCity-HD maps as a key challenge of the automotive industry, *Engineering*, vol.2, no.2, pp.159-162, 2016.

[33] P. Bender, J. Ziegler and C. Stiller, Lanelet: Efficient map representation for autonomous driving, *Proc. of the 2014 IEEE Intelligent Vehicles Symposium*, pp.420-425, 2014.

[34] F. Poggenhans, J.-H. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt and M. Mayr, Lanelet2: A high-definition map framework for the future of automated driving, *Proc. of 2018 IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp.1672-1679, 2018.

[35] M. Althoff, S. Urban and M. Koschi, Automatic conversion of road networks from OpenDRIVE to Lanelet, *Proc. of 2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, pp.157-162, 2018.

[36] M. Naumann, F. Poggenhans, M. Lauer and C. Stiller, CoInCar-Sim: An open-source simulation framework for cooperatively interacting automobiles, *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp.1-6, 2018.