Local Path Planning Algorithm for Autonomous Vehicle Based on Multi-objective Trajectory Optimization in State Lattice

Preprint · November 2020

CITATIONS
0

READS
963

3 authors, including:

Ivan Kornev
Moscow State Institute of Radio Engineering, Electronics and Automation
2 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Oleg Shipitko
The Institute for Information Transmission Problems
36 PUBLICATIONS **49** CITATIONS
SEE PROFILE

Local Path Planning Algorithm for Autonomous Vehicle Based on Multi-objective Trajectory Optimization in State Lattice

Ivan I. Kornev[1, 2], Vladislav I. Kibalov[1, 2], Oleg S. Shipitko[1, 2]

[1] Institute for Information Transmission Problems – IITP RAS, Bol'shoy Karetnyy Pereulok 19, Moscow, Russia, 127051.
[2] Evocargo LLC, Vyatskaya street 27, Moscow, Russia, 127015.

ABSTRACT

The paper presents an algorithm for constructing a local path for a vehicle with nonholonomic kinematics of an automobile type. A local path is a sequence of transitions in the graph of possible maneuvers that minimizes a given cost function. The graph is constructed by duplicating along the global path pre-calculated in a curvilinear coordinate system set of kinematically feasible motion primitives. The use of pre-computed motion primitives significantly reduces the time of graph construction. The weight of each maneuver – the edge of the transition graph – is calculated as a weighted sum of costs based on several criteria. The specified cost function minimizes maneuvering and maintains a safe distance to static obstacles. The information about obstacles is extracted from an occupancy grid map. Dijkstra's algorithm is used to search a path in the weighted directed graph. The algorithm was tested on a dataset containing real road scenes. Each scene represents a given global path and a static environment model where a safe local path must be found. Local path search is performed in real-time. Experiments have shown that safe local paths have been found in all scenes where it was physically possible. At the same time, the obtained local paths were on average only on 1.3% longer than the given global paths which demonstrate the high applicability of the proposed algorithm.

Keywords: path planning, local path planning, state lattice, multi-objective optimization, collision avoidance, autonomous vehicle, nonholonomic kinematics

1. INTRODUCTION

The problem of path planning is still relevant despite the large body of dedicated research. It gained particular importance due to the rapid development in the field of autonomous vehicle (AV). The operational design domain of such vehicles imposes strict requirements on the properties of algorithms, among which are computational speed and safety of the generated paths. Since the calculation of the path is preceded by object recognition and road situation analysis, the time constraints on the planning algorithms become even more severe [1]. Moreover, many additional requirements may be imposed on the planned path, for instance, it should be optimal in terms of travel distance and safe in terms of vehicle-to-obstacles distance; in the case of people transportation, the ride comfort will be appreciated, which is achieved by minimizing the first and second derivatives of the vehicle speed. The high number of requirements and limitations lead to the fact that the local path planning problem has to be considered as a multi-objective optimization problem.

The task of path planning is often divided into two subtasks – global and local planning. The global path is built according to a prior map, connecting the start point with the endpoint. The local path allows the vehicle to move along the global one, taking into account the information received from the sensors in real-time. The need to separate the problems of local and global planning is dictated by the high computational complexity of finding the global path and, as a consequence, the impossibility of the underlying search algorithms to provide real-time performance. So additional local path planning is used to quickly make decisions about maneuvering to avoid obstacles.

2. RELATED WORK

The Darpa Grand Challenge (held in 2004, 2005, 2007) provided a strong impulse to the development in the field of path planning for AV. During the second competition in 2005 [2] the participants were given a file containing a sequence of waypoints – the global path to follow. To win the competition, it was necessary to drive along the global path faster than the rest of the teams, without leaving the track. The third season of the competition was held in a recreated urban environment, so AVs had to avoid collision with other traffic participants. To solve this problem teams came up with various approaches. For example, the team from the Massachusetts Institute of Technology (MIT) used a modified fast-growing

random tree method [3]. The main disadvantage of this approach was the introduction of "lazy check". At each iteration of the algorithm, the tree growth did not start from scratch. Instead, it was based on the tree calculated at the previous iteration. The solution that was the best at the previous time step could be the worst at the current iteration. The "lazy check" only controlled the compliance of this solution with all the constraints set, but did not allow finding a better solution for the current iteration. The introduction of such a feature was necessary to release computing resources to expand the tree growth and therefore increase search coverage.

Since the Darpa Grand Challenge, the research focus in the field of path planning for AV was concentrated in developing algorithms that allow finding optimal trajectories in real-time and satisfying a set of criteria for maneuvering in a dynamic environment [1]. The developed approaches based on numerical optimization [4], [5] are capable of finding the best solution for multi-criteria optimization problem converging to the global extremum. However, they are computationally complex, and the solution found in a limited time may turn out to be far from optimal. Approaches based on numerical optimization compete with approaches graph-based approaches [6, 7, 8]. Their main advantage is computation speed. These methods can find an optimal solution but in state-space limited to precomputed transitions graph (i.e. state lattice [9]). The transitions are precomputed at the time of algorithm initialization. Then transitions are stored in memory and used during the planning phase in real-time to reduce the cost of graph generation.

In this work, an algorithm for finding the optimal local path based on a space lattice is presented. The proposed algorithm has an advantage of state space-based approaches since the search space is discretized and the set of kinematically possible maneuvers is precomputed at the time of algorithm initialization. The set of motion primitives is initially constructed in a curvilinear coordinate system and is used to create multiple possible local paths along the global path. The introduction of a multi-criteria cost function allows to take into account multiple criteria and to find a path that satisfies them the most. Therefore the suboptimality of found paths is traded on the computational efficiency obtained tanks to the state-space discretization and the use of precomputed motion primitives.

3. LOCAL PATH PLANNING ALGORITHM BASED ON MULTI-CRITERIA TRAJECTORY OPTIMIZATION IN STATE LATTICE SPACE

In this section, a detailed description proposed path planning algorithm is given. The algorithm steps of state lattice generation and multi-criteria path optimization are discussed in dedicated sections.

3.1 State Lattice Generation

The proposed path planning algorithm interacts with the local environment model, which contains a static representation of an environment for each time step of the planning horizon. One of the inputs of the proposed algorithm is the part of a global path closest to the vehicle. All potential local paths are constructed along the global path. At the moment of algorithm initialization, a set of motion primitives in the form of a directed unweighted graph is constructed. The graph vertices are represented by points in the curvilinear coordinate system (i.e. Frenet coordinate system [10]) where each vertex is described by longitudinal and lateral (l, r) offset. The values of longitudinal and lateral coordinates are discrete with a given step along each of the axes $(\Delta l, \Delta r)$. Vertices with the same value of a longitudinal offset form a so-called layer (Fig. 1). Graph edges connect vertices of the first layer to subsequent layers with specified indices. Edges represent the trajectory along which the maneuver is performed by a third-degree polynomial. By that, it is possible to calculate a curvature of an edge at any of its points analytically and exclude edges that do not satisfy the kinematic constraints of the vehicle. Fig. 2a shows an example of a set of maneuvers in which the first layer vertices are connected only to the next ones, in Fig. 2b the vertices of the first layer are connected to the vertices of 2, 3, 5 layers. The sequence of layer indices is set manually based on the parameters of the vehicle, computing capacity, and particular problem statement. An increase in the number of connected layers leads to an increase in the number of edges and, subsequently, increases the computation time. On the other hand, such a configuration produces a path closer to the global optimum. Also, longer edges provide smoother maneuvers, allowing the vehicle to maintain high speed, while short ones allow the vehicle to perform sharp maneuvers. The local path is shown in Fig. 3a is built based on a set of motion primitives in which the vertices of the first layer are connected to the vertices of only the second layer. Obstacles are avoided by sequential execution of short maneuvers, which leads to excessive steering (Fig. 3c). The local path is shown in Fig. 3b, built based on a set in which the vertices of the first layer are connected to the vertices of the 2, 3, 5, 7 layers. The path is smoother, and the sum of the absolute AV orientation changes decreased by 2 times (Fig. 3d) compared with the previous example.

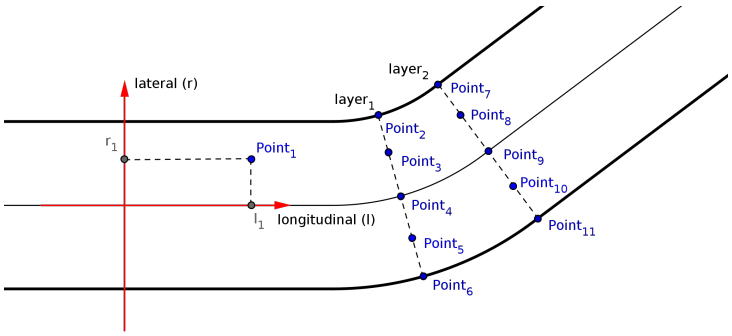
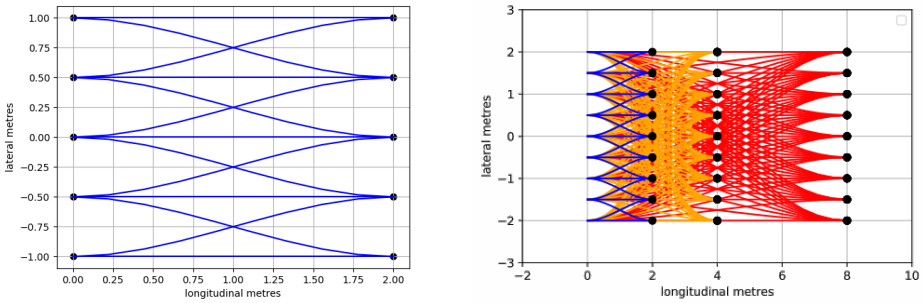


Figure 1: Curvilinear coordinate system. The points $Point_2$ - $Point_{11}$ form single graph layer.



(a) (b)

Figure 2: The set of motion primitives. (a) A simple set with longitudinal step Δl between layers 2 m, lateral step Δr between vertices in one layer 0.5 m, vertices of the first layer are connected to vertices of the second layer only, the minimum turning radius of the vehicle is 1 m, the maximum lateral vertices displacement is 1 m. (b) Set with longitudinal step Δl between layers 2 m, lateral step Δr between vertices in one layer 0.5 m, vertices of the first layer are connected to vertices from 2, 3, 5 layers, the minimum turning radius of the vehicle is 1 m, the maximum lateral vertices displacement is 2 m.

The constructed set of motion primitives in a curvilinear coordinate system is used to obtain a set of potential paths. The set is "duplicated" along a global path in a curvilinear coordinate system with a given longitudinal step. All vertices and edges then converted from a curvilinear coordinate system to a Cartesian one. The first vertex of the graph is the current position of the vehicle. The edges connecting the first vertex with the first layer of the graph are built based on fifth-degree polynomials [11] and are connected only with those vertices, the transition to which satisfies kinematic constraints and will not lead to a collision. The last layer of the graph contains only one vertex, which lies strictly on a global path at a planning horizon distance from a vehicle's current position (Fig. 4).

3.2 Optimal solution search

The graph constructed in Cartesian space is projected onto a local obstacle map. If movement along an edge leads to a collision, such an edge is removed from the graph. Each edge E is assigned with a weight $W(E)$. The weight is defined as a weighted sum of several criteria, each can be interpreted as a cost function:

$$W(E_i) = k_{safe} * w_{safe}(E_i) + k_{dist} * w_{dist}(E_i) + k_{man} * w_{man}(E_i), \tag{1}$$

where $w_{safe}(E_i) = \sum_{n=1}^{N} f(e_n)$ – criterion for the distance to a nearest obstacle. To evaluate w_{safe}, the edge is divided into segments e_n. The number of segments is defined as the longitudinal length of the edge divided by the longitudinal step of the set $N = \frac{L(E_i)}{\Delta l}$. For each segment, a distance to the nearest obstacle is calculated. Fig. 5 shows the safety function $f(e)$ that defines the dependence of the weight on the minimum distance between the vehicle when passing along the current segment of an edge and obstacles. The function is quadratic, since the closer the vehicle passes to an obstacle,

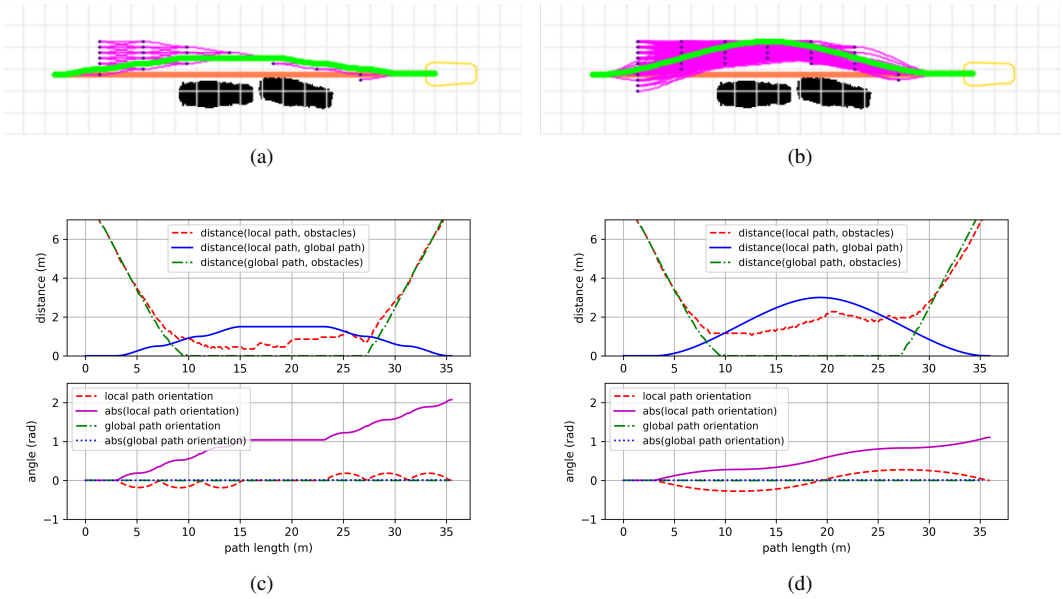


Figure 3: Influence of edges length on a local path: (a) local path (green) searched in the graph with edges connecting only first and second layers (purple); (b) path that is searched in the graph with short and long edges. Characteristics of both local paths are shown in (c) and (d). Upper plots: green and red colors correspond to the distance between global/local path and nearest obstacles. Blue color corresponds to the distance between the global path and local path. Lower plots: green and red colors correspond to the relative orientation of AV while moving along global and local paths respectively. Blue and purple colors demonstrate integral absolute orientation change while AV moving along global and local paths

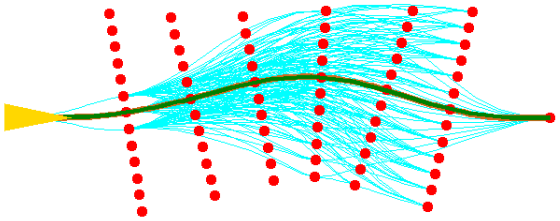


Figure 4: A graph of possible transitions in a Cartesian coordinate system, based on a set of motion primitives in a curvilinear coordinate system.

the more undesirable it is to move along this segment if the distance is greater than the inflation radius, the safety weight is zero, which means that edge of the graph is not penalized.

$w_{dist}(E_i) = L(E_i) * dist(E_i)$ – criterion for the average lateral distance of an edge to a global path, where $dist(E) = (|r_{from}| + |r_{to}|) * 0.5$. r_{from} and r_{to} - lateral coordinates of the start and end edge vertices, respectively. Therefore, the further from a global path the edge is, the higher it is penalized.

$w_{man}(E_i) = |r_{from} - r_{to}|$ – the criterion for changing the lateral position, imposes a penalty on the edge that change the lateral position of the vehicle, allowing to avoid unnecessary maneuvering. Thus, the problem of finding an optimal local path is reduced to finding a path with minimum weight in a weighted directed graph. Values of the k_{man}, k_{dist}, k_{safe} coefficients should be set based on the particular requirements to the path planning algorithm. The greater the coefficient value, the more impact the corresponding criterion has.

4. EXPERIMENTAL RESULTS

A series of experiments was conducted to evaluate the efficiency of the algorithm. All input data were created manually. The experimental data are the series of occupancy grid maps with initial global paths, and the vehicle position at the initial time moment. The result of the experiment is a generated local path. The generated path was compared to the global path

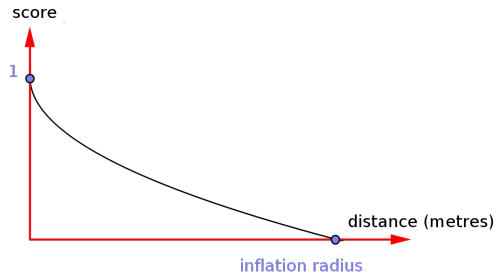


Figure 5: Safety function $f(e)$.

Figure 6: Obstacles samples: walking man (a), a car (b), a cyclist (c), a concrete mixer truck (d).

in terms of collisions and length. Section 4.1 describes the methodology of conducted experiments and creating test scenes. Section 4.2 presents the experimental results.

4.1 Experiment Setup

The experiments to validate the proposed algorithm include calculating a local path for a vehicle to avoid static obstacles for each scene of a given set. A scene consists of a priori local occupancy grid map, a global path, a vehicle pose at the initial moment and algorithm parameters. A local occupancy map represents the environment in the form of a binary image (with 10 pixels per meter scale). Black pixels on the map represent obstacles. White pixels encode free space through which the vehicle can pass. When creating scenes, real sections of city roads were used in compliance with proportions and sizes accurate to the scale of the map. The vehicle is approximated by a polygon approximately equal in size to the average car (the width of an average car is 1.5 - 1.8 m, length 3 - 4.8 m). The test scenes contain frequently encountered road objects such as cars, trucks, pedestrians, cyclists (Fig. 6). The dimensions of the obstacles correspond to the real ones. The poses of obstacles and the global path were set manually. The test scenarios included both complex scenes with one or several obstacles, and simple scenes without obstacles (such scenes are intended to test the property of the algorithm to generate a path that, in accordance with the introduced optimality criteria (1), would not deviate from the global path in the absence of obstacles. In total, 11 scenes were generated, close to the situations potentially possible on the roads. Among the created scenes: movement along the cars parked in the right lane Fig. 7a, roundabout Fig. 7b, driving through difficult intersections Fig. 7c, 7e, traffic island section Fig. 7d and driving into a parking space Fig. 7f.

4.2 Results and Discussion

The purpose of the experiments was to show that the proposed algorithm finds a local path that will avoid collisions with obstacles with the smallest possible increase in the distance traveled compared to moving along a global path. In 10 test cases out of 11, driving along the global path led to collisions while driving along the local path let avoid them entirely. At the same time, the local path became longer on average by only 1.3%. The obtained paths satisfy the kinematic constraints, and due to the use of long edges, the transitions between the vertices is smoother, which allows the vehicle to maintain high speed when moving along them. For 1 scene, it was not possible to construct a collision-free path because of the high obstacle density (Fig. 7d). A local path can not be found if a path in the directed graph between start (vehicle position) and end (last point of the global path) vertices does not exist. The influence of the weighting coefficients 1 on the generated path also was analyzed. A test scene 9 (a) was chosen for the experiment. In the experiment, each of the weighting coefficients of the cost function was in turn equated to zero. The results are shown in Fig. 9. A local path with all nonzero coefficient values is shown in blue. This path is built as a reference. Based on the results obtained, the following conclusions can be made:

- Setting k_{man} equal to zero led to sharper maneuvering (results are marked in green). This is especially noticeable on sections of the path from 7.5 to 15 meters (on the reference image, maneuvers occur from 5 to 15 meters. The

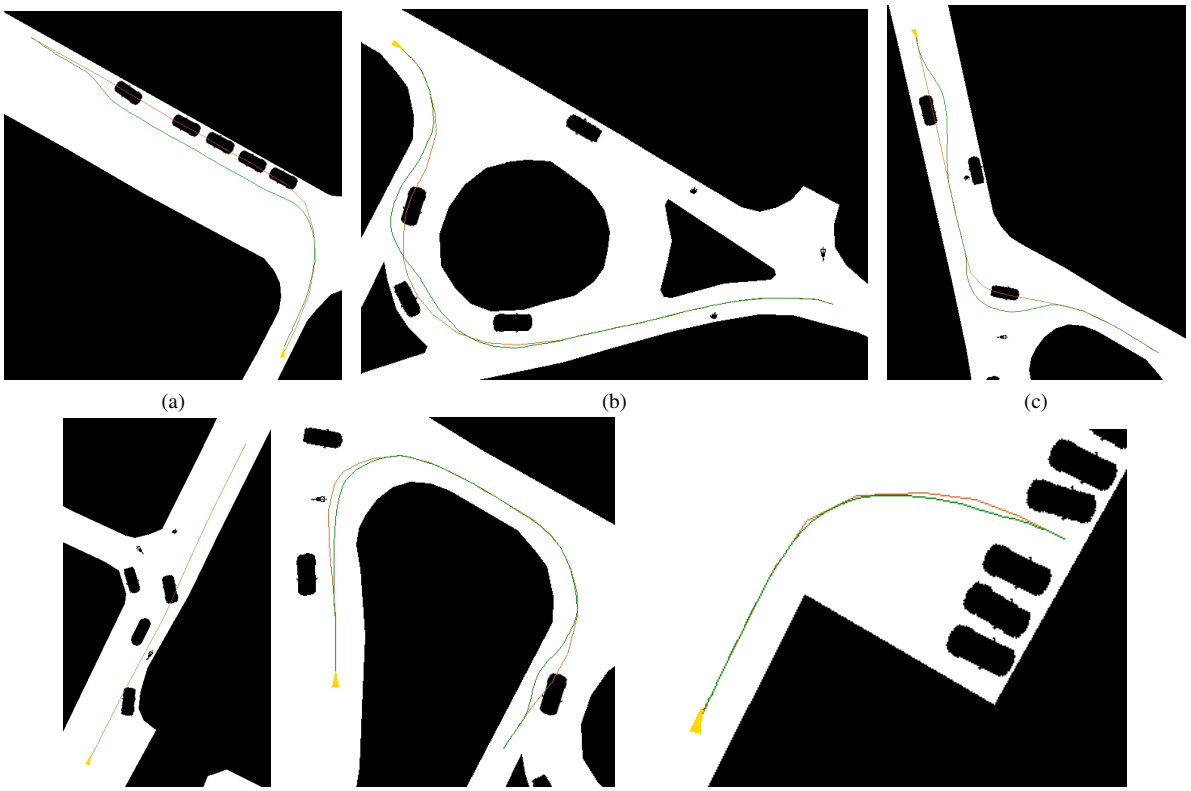


Figure 7: In all figures, the brown line is the global path, the green one is the obtained local path, the yellow triangle indicates the vehicle initial pose. The algorithm parameters $k_{man} = 2$, $k_{dist} = 0.14$, $k_{safe} = 7$.

indicated maneuvers are highlighted in orange in both images), while the deviation from the global path did not change compared to the reference.

- Setting k_{safe} equal to zero (graphs in yellow) resulted in strict moving along the global path, since it satisfies the safety criterion. In general, a zero or small value of the safety factor can lead to building a local path too close to obstacles.
- Setting k_{dist} equal to zero (graphs in blue) increased the deviation of the local path from the global one. The graph shows an increase in the total change in the AV orientation. For 30 meters of travel, it was almost 3 radians (for the reference path, this value is slightly more than 2 radians).

A significant increase in any of the coefficients leads to a small influence of the remaining criteria, which can cause excessive maneuvering or too dangerous maneuvers performed close to obstacles. The experiments with different values of the weighting coefficients and their influence on the local path are presented in Fig. 8. Increasing the value of the coefficient k_{dist} causes the local path to "stick" to the global one, which can lead to excessive steering when avoiding several obstacles or passing too close from obstacles and maybe unsafe.

5. CONCLUSION

The paper proposes a local planning algorithm based on finding the optimal path in a directed weighted graph. The graph is obtained by constructing a state lattice – a repeating set of kinematically achievable precalculated motion primitives at the algorithm initialization. The proposed method of connecting not only a neighbor but also separated from each other layers of the graph makes it possible to find smoother, kinematically achievable paths while leaving the opportunity to make sharp maneuvers to avoid dynamic obstacles. After the graph construction, the best path, the edges of which minimize the

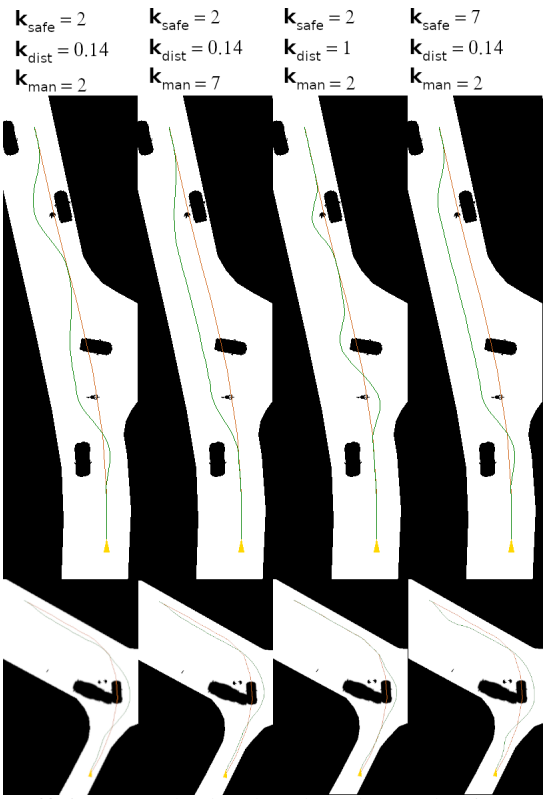


Figure 8: Influence of weighting coefficients on the local path. The predominant weighting coefficient in the second column is the change in the lateral position, in the third – the deviation from the global path, in the fourth – the distance from obstacles.

given cost function, is searched. The multicriteria of the cost function on the one hand guarantees the distance of the path from obstacles and allows to minimize unnecessary maneuvering on the other. The proposed algorithm has been tested on scenes that simulate real road scenarios. The local path planning algorithm avoided collisions in all the scenarios presented. The obtained local paths turned out to be on average only 1.3 % longer than the a priori specified global path, which does not take into account the presence of other road users. These results confirm the efficiency and applicability of the proposed algorithm. Future work can be aimed at comparing the algorithm with other existing approaches to local planning to identify potential weaknesses of the algorithm and find ways to further improve it.

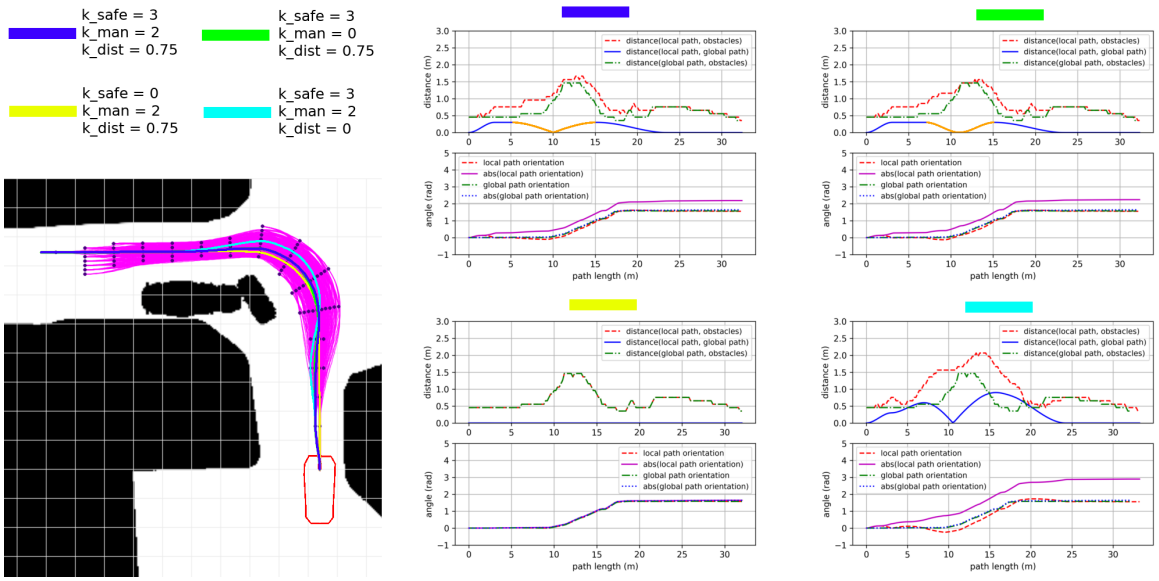
References

[1] David González, Joshué Pérez, Vicente Milanés, and Fawzi Nashashibi. A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1135–1145, 2015.

[2] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, et al. Stanley: The robot that won the darpa grand challenge. *Journal of field Robotics*, 23(9):661–692, 2006.

[3] Yoshiaki Kuwata, Gaston A Fiore, Justin Teo, Emilio Frazzoli, and Jonathan P How. Motion planning for urban driving using rrt. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1681–1686. IEEE, 2008.

[4] Julius Ziegler, Philipp Bender, Thao Dang, and Christoph Stiller. Trajectory planning for bertha—a local, continuous method. In *2014 IEEE intelligent vehicles symposium proceedings*, pages 450–457. IEEE, 2014.



(a) (b)

Figure 9: Influence of weighting factors on the shape of a local path. (a) Dependence of the shape of the local path on the values of the weight coefficients, (b) characteristics of the local and global paths. For the top rows: green and red are the distance from the global and local paths respectively to the nearest obstacle, the blue color is the distance between the global and local paths. For the bottom rows: green and red colors are relative change in the angle of rotation of the AV when moving along the global and local path respectively, blue and purple colors are the sum of the modules for changing the angle of rotation of the AV when moving along the global and local paths respectively.

[5] Christoph Rösmann, Wendelin Feiten, Thomas Wösch, Frank Hoffmann, and Torsten Bertram. Trajectory modification considering dynamic constraints of autonomous robots. In *ROBOTIK 2012; 7th German Conference on Robotics*, pages 1–6. VDE, 2012.

[6] Mihail Pivtoraiko and Alonzo Kelly. Efficient constrained path planning via search in state lattices. In *International Symposium on Artificial Intelligence, Robotics, and Automation in Space*, pages 1–7. Munich Germany, 2005.

[7] Julius Ziegler and Christoph Stiller. Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1879–1884. IEEE, 2009.

[8] Sivakumar Pothan, JL Nandagopal, and Gopinath Selvaraj. Path planning using state lattice for autonomous vehicle. In *2017 International Conference on Technological Advancements in Power and Energy (TAP Energy)*, pages 1–5. IEEE, 2017.

[9] Kristoffer Bergman, Oskar Ljungqvist, and Daniel Axehill. Improved path planning by tightly combining lattice-based path planning and optimal control. *IEEE Transactions on Intelligent Vehicles*, 2020.

[10] Moritz Werling, Julius Ziegler, Sören Kammel, and Sebastian Thrun. Optimal trajectory generation for dynamic street scenarios in a frenet frame. In *2010 IEEE International Conference on Robotics and Automation*, pages 987–993. IEEE, 2010.

[11] Arata Takahashi, Takero Hongo, Yoshiki Ninomiya, and Gunji Sugimoto. Local path planning and motion control for agv in positioning. In *Proceedings. IEEE/RSJ International Workshop on Intelligent Robots and Systems'.(IROS'89)'The Autonomous Mobile Robots and Its Applications*, pages 392–397. IEEE, 1989.

AUTHORS' BACKGROUND

Name	Title	Research Field	Personal website
Ivan Kornev	Master's student, engineer in EvoCargo LLC	Mechatronics and robotics	https://www.researchgate.net/profile/Ivan_Kornev2
Vladislav Kibalov	Engineer in EvoCargo LLC, junior researcher in IITP RAS	Robotics	https://www.researchgate.net/sciescientific-contributions/2161904141_Vladislav_Kibalov
Oleg Shipitko	PhD Student	Robotics, computer vision	https://www.researchgate.net/profile/Oleg_Shipitko