

Motion Planning and Decision-Making for Autonomous Systems: From Quadrotors To Autonomous Vehicles

by

Wenchao Ding

A Thesis Submitted to
The Hong Kong University of Science and Technology
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy
in the Department of Electronic and Computer Engineering

August 2020, Hong Kong

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.



Wenchao Ding


August 2020

Motion Planning and Decision-Making for Autonomous Systems: From Quadrotors To Autonomous Vehicles

by

Wenchao Ding

This is to certify that I have examined the above Ph.D. thesis
and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by
the thesis examination committee have been made.



Prof. Shaojie Shen, Thesis Supervisor
Department of Electronic and Computer Engineering



Prof. Bertram Emil Shi, Head of Department
Department of Electronic and Computer Engineering

Thesis Examination Committee:

- | | |
|--|---|
| 1. Prof. Shaojie SHEN (Supervisor) | Department of Electronic and Computer Engineering |
| 2. Prof. Bertram Emil SHI | Department of Electronic and Computer Engineering |
| 3. Prof. Wei ZHANG | Department of Electronic and Computer Engineering |
| 4. Prof. Xiaojuan MA | Department of Computer Science and Engineering |
| 5. Prof. Jianru XUE
(External Examiner) | Institute of Artificial Intelligence and Robotics,
Xi'an Jiaotong University |

Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology

August 2020

ACKNOWLEDGMENTS

This thesis would never be completed without the help of many people. First of all, I would like to express my sincere thanks to my supervisor, Professor Shaojie Shen. I did not major in automation/cv nor any domain related to robotics before my Ph.D. Instead, I majored in Wireless Communication in my Bachelor and the first year of my Ph.D. study. However, when I started my second year, I became confused, because I found out that although I love math, I do not like pure mathematical derivations that are far away from being applied to the real world. Keeping working on wireless communication problems seemed not to steer my Ph.D. study effectively. Therefore, I decided to switch to another area which makes me excited. I had a hard time checking out the research of different professors and finally, I found robotics. I was fascinated by real-world robots that have “math” really running on it. However, as you can imagine, I was a freshman in robotics with an almost mutually uncorrelated background. However, Prof. Shen accepted me and guided me through the hardship. Without him, there wouldn’t be this man with a communication background defending his robotics PhD.

I thank the other members of my thesis committee, Professor Bert Shi, Professor Wei Zhang, Professor Xiaojuan Ma, Professor Jianru Xue. Scheduling the examination should have been one of the challenges of the graduation, but with their help and kind consideration, scheduling is far simpler than I expected.

I also would like to thank my colleagues in the HKUST Aerial Robotics Group. I thank Lu Zhang who has been working closely with me. We encountered many challenges but we never gave up. I thank Peiliang Li and Kaixuan Wang. We work, play, and graduate together. I enjoyed doing experiments with Wenliang Gao. And I also enjoyed playing badminton with Siqi Liu, Shaozu Cao, Luqi Wang. My daily life would be plain without communicating with Fei Gao, Tong Qin, Kejie Qiu, William Wu, Jieqi Shi, Chuhao Liu, Pan Jie, Boyu Zhou, Yi Lin, and Ke Wang.

Last but not least, I thank my family and my girlfriend Jieru Zhao. They have been with me, through all the ups and downs.

TABLE OF CONTENTS

Title Page	i
Authorization Page	ii
Signature Page	iii
Acknowledgments	iv
Table of Contents	v
List of Figures	ix
List of Tables	xiv
Abstract	xv
Chapter 1 Introduction	1
1.1 Research Problems	2
1.1.1 Motion Planning for Quadrotors in Unknown 3-D Environments	4
1.1.2 Behavior Prediction for Autonomous Vehicles	4
1.1.3 Decision-making for Autonomous Vehicles	4
1.1.4 Motion Planning for Autonomous Vehicles	5
1.1.5 System Integration for Autonomous Vehicles	5
1.2 Thesis Overview	6
1.3 Thesis Contributions	7
Chapter 2 Kinodynamic Motion Planning for Quadrotors	9
2.1 Scientific Background and Literature Review	9
2.1.1 Background	9
2.1.2 Related Works	12
2.2 System Overview	14
2.3 B-spline Curve and Replanning	15
2.3.1 Local Control Property and Replanning	16
2.3.2 Convex Hull Property and Dynamical Feasibility	17
2.4 B-spline-Based Kinodynamic Search	18

2.4.1	Motivating Example	18
2.4.2	Problem Formulation	19
2.4.3	Optimal B-spline-Based Kinodynamic Search	21
2.4.4	Feasibility Condition	25
2.4.5	Efficient Low Dispersion Search	25
2.4.6	Analysis of the EBK Search	26
2.5	Elastic Optimization	28
2.5.1	Elastic Tube Expansion	28
2.5.2	Elastic Optimization Formulation	29
2.5.3	Enforcing Safety Guarantee	31
2.6	Validation	33
2.6.1	Implementation Details	33
2.6.2	Analysis	35
2.6.3	Experimental Results	45
2.7	Conclusion	48
Chapter 3	Behavior Prediction for Autonomous Vehicles	49
3.1	Scientific Background and Motivation	49
3.2	Related Works	51
3.3	Methodology	52
3.3.1	Problem Formulation	52
3.3.2	RNN Encoder Network	54
3.3.3	Vehicle Behavior Interaction Network (VBIN)	54
3.4	Experimental Results	57
3.4.1	Dataset	58
3.4.2	Baselines	58
3.4.3	Implementation Details	58
3.4.4	Evaluation Metrics	59
3.4.5	Analysis	60
3.5	Conclusion	63
Chapter 4	Efficient Uncertainty-aware Decision-making for Autonomous Vehicles	64
4.1	Scientific Background and Literature Review	64
4.1.1	Scientific Background	64
4.1.2	Related Works	66

4.2	System Overview	67
4.3	Decision-making via Guided Branching	68
4.3.1	Preliminaries on POMDP	68
4.3.2	Domain-specific Closed-loop Policy Tree	69
4.3.3	Conditional Focused Branching	71
4.4	Implementation Details	72
4.4.1	Semantic-level Actions	72
4.4.2	Forward Simulation	74
4.4.3	Belief Update	74
4.4.4	CFB Mechanism	75
4.4.5	Policy Evaluation	76
4.4.6	Trajectory Generation	78
4.5	Experimental Results	78
4.5.1	Simulation Platform and Environment	78
4.5.2	Qualitative Results	79
4.5.3	Quantitative Results	80
4.6	Conclusion	81
Chapter 5	Spatial-Temporal Motion Planning for Autonomous Vehicles	82
5.1	Scientific Background and Literature Review	82
5.1.1	Scientific Background	82
5.1.2	Spatio-temporal motion planning for AVs	83
5.1.3	Corridor generation for AVs	84
5.2	System Overview	85
5.3	Preliminaries on Multi-policy Decision Making	86
5.4	Spatio-temporal Semantic Corridor	87
5.4.1	Semantic Elements And Frenét Frame Representation	87
5.4.2	Semantic Corridor Generation	89
5.5	Trajectory Generation With Safety and Feasibility Guarantee	91
5.5.1	Bézier Basis and Its Properties	91
5.5.2	Piecewise Bézier Curve Representation	92
5.5.3	Enforcing Safety and Dynamical Constraints	93
5.6	Experimental Results	95
5.6.1	Implementation Details	95
5.6.2	Qualitative Results	96

5.6.3 Comparisons and Analysis	96
5.7 Conclusion	99
Chapter 6 System Integration and Field Tests for Autonomous Vehicles	100
6.1 Scientific Background and Motivation	100
6.2 Related Work	102
6.3 System Overview	104
6.4 Complete System Formulation	106
6.5 Implementation Details	108
6.6 Field Tests	109
6.6.1 Intelligent gap finding and merging	109
6.6.2 Interaction-aware overtaking	110
6.6.3 Low-speed automatic lane change in dense traffic	111
6.6.4 Cut-in handling and automatic lane change	112
Chapter 7 Conclusion and Future Work	113
7.1 Thesis Summary	113
7.2 Lessons Learned and Future Work	114
7.2.1 Thinking Beyond Path Search (MAV)	114
7.2.2 Uncertainty in Behavior Prediction (AV)	115
7.2.3 Incorporating Domain Knowledge in Decision-making (AV)	117
7.2.4 Coupling Prediction and Planning (AV)	117
Appendix A Supplementary Material for Kinodynamic Search	119
A.1 Proof of Prop. 1	119
A.2 Relationship between \mathcal{G} and \mathcal{G}_H	119
A.3 Characterization of the inflation for the B-spline-based kinodynamic search	120
A.4 Performance analysis of the EBK search	120
A.5 Proof of Theorem 1	122
Publication List	123
References	125

LIST OF FIGURES

- 1.1 Illustration of the typical hierarchical structure of an autonomous system. For quadrotors, the perception layer includes depth sensing (i.e., static obstacles) and 3-D map reconstruction, while the recognition layer may include trajectory prediction for tracked objects. For autonomous vehicles, the perception layer may further include free-space detection, traffic signal detection, etc, while the recognition layer includes behavior and trajectory prediction for traffic participants. The control layer for both quadrotor and autonomous vehicle has the same meaning, namely, tracking the output trajectory of the planning layer as accurate as possible. 2
- 2.1 Illustration of the motivating example. The initial state has non-zero velocity (*red arrow*). The traditional geometric planner finds the shortest path (red squares) and then parameterizes it using a piecewise polynomial (*blue*). However, the local path parameterization is restricted to a homotopy class (*blue area*), and the resultant trajectory is jerky (even infeasible) with respect to the given time allocation. In contrast, our framework produces a dynamically feasible trajectory (*purple line*) using kinodynamic planning. 10
- 2.2 A diagram of our kinodynamic replanning framework together with state estimation and mapping modules. 14
- 2.3 Illustration of the B-spline local control property and its application to the replanning system. The control points are shown by squares. The control points corresponding to the executing trajectory are shown in *green*. The original locations of the control points ahead of the executing control point are in *orange*, and their subsequently modified positions are marked in *red*. Due to the locality of the B-spline, the replan causes no perturbation to the executing trajectory. 17
- 2.4 Illustration of the convex hull property. The dashed *red* box shows the feasible velocity hull (1.2 m/s for each axis). Applying Prop. 1 under the objective of minimum change to control point positions (bottom left figure), the resulting velocity profile is shown at the bottom right, where the velocity profile is strictly bounded. 18
- 2.5 Illustration of the construction process of 3-degree vertex tuples from an admissible path. Each vertex tuple is formed by combining four consecutive control points. $[\tilde{\pi}]_0^3$ and $[\tilde{\pi}]_1^3$ are two neighboring vertex tuples since they overlap for three vertices. 21
- 2.6 Illustration of the mapping process to \mathcal{G}_H and the graph aggregation process used by the EBK search. 24
- 2.7 Illustration of the EO approach: (a) shows the elastic tube expansion process (Sect. 2.5.1); (b) shows the optimization process (Sect. 2.5.2). In (a), the original tube is marked in *green*, while the inflated tube is marked in *yellow*. 30
- 2.8 Illustration of (a) the geometric incompleteness of the ball constraint, (b) the two-level inflation scheme, and (c) the iterative convex hull shrinking process for enforcing the safety guarantee. It can be observed in (a) that even the straight-line segments between the balls may collide with the obstacle. 31

2.9	Illustration of our (a) monocular vision-based quadrotor testbed and (b) dual-fisheye vision-based quadrotor testbed.	33
2.10	Comparisons of different kinodynamic planning approaches.	37
2.11	Comparisons of different trajectory optimization methods on two different maps. The EO method is shown in <i>purple</i> , the CT method is shown in <i>red</i> and the GS method is shown in <i>blue</i> .	41
2.12	Illustration of our replanning system on different maps.	43
2.13	Illustration of the simulated environment for benchmarking.	43
2.14	Illustration of different replanning methods in the same simulated environment. The trajectory is shown in <i>purple</i> , and the acceleration profile is marked in <i>green</i> . For the replanning cases where the shortest path direction is inconsistent with the initial state, we mark the initial velocity with a <i>red arrow</i> and the shortest path direction in <i>brown</i> .	44
2.15	Illustration of the snapshot of the indoor replanning with the monocular vision. In (a), the control points found by EBK (<i>blue</i>), executed control points (<i>pink</i>), committed control points (<i>green</i>), and control points under optimization (<i>yellow</i>) are marked. The corresponding indoor environment is shown in (b).	46
2.16	Illustration of the whole trajectory and final accumulated map for the indoor replanning using the monocular perception system.	46
2.17	Illustration of the snapshot of the indoor replanning with the omnidirectional vision. With a fixed yaw angle, the obstacles around the quadrotor can be mapped.	47
2.18	Illustration of the whole trajectory for the replanning using the dual-fisheye omnidirectional perception system. Unlike the experiments using the monocular testbed, we can achieve round-trip flight in an unknown complex indoor environment.	47
2.19	Illustration of the outdoor experiments using the monocular vision: A flight through a pavilion is shown in (a) (part of the map on the top is cut for visualization purposes); a flight avoiding trees is shown in (b).	48
3.1	Illustration of the benefit of extending the prediction horizon. Assume that the green vehicle is the ego vehicle with the planning module, while the transparency of the vehicles represents the time elapsed. For a detection-based method, as shown on the top, the LC prediction is given when the blue vehicle has a clear LC pattern, which may result in a sudden braking of the ego vehicle due to the late discovery. However, from the interaction point of view, the blue vehicle is moving at a high speed and is blocked by the slowly moving red car. The blue vehicle has two interaction choices: brake to avoid collision or merge into the other lane. By learning from a large number of interaction patterns, the likelihood of LC can be estimated, even before the blue vehicle has a clear LC maneuver. More examples can be found in the video https://www.youtube.com/watch?v=SuQzxAusU_0 .	50

3.2	Illustration of the popular social pooling strategy [1]. The RNN hidden states in the same spatial cell are pooled and passed to a fully connected layer to generate the total social effect on the target vehicle (yellow). Vehicles which have totally different dynamics in the same cell will share the same weight. However, as shown in the Illustration, if the vehicle that is highlighted with a circle is moving slowly, it is supposed to have a large weight since it blocks the LC route of the target vehicle, but if it is moving much faster than the target vehicle, it should have little impact on the target vehicle's LC.	52
3.3	VBIN building block: pairwise interaction unit (PIU). RNNs are implemented using gated recurrent units (GRUs) [2], and the hidden states are used as the input of the PIU.	54
3.4	Illustration of the PIU connections for each selected neighboring vehicle.	56
3.5	VBIN building block: neighborhood interaction unit (NIU). All the PIUs are identical.	56
3.6	Illustration of the VBIN structure. All the NIUs are identical.	57
3.7	Illustration of the dataflow of the system.	57
3.8	Illustration of the comparison on the NGSIM dataset. The vehicle marked in yellow is the prediction target. The vehicles filled with cyan and red are the selected neighboring vehicles. The pair of each red vehicle and target vehicle may have potential interaction according to the RSS safety model [3]. Note that we do not use any handcrafted models and the RSS is used to aid understanding of the scenario. The behavior predictions are marked in arrows, and the transparency represents the magnitude of the likelihood.	61
3.9	Illustration of the prediction accuracy w.r.t. the TTLC.	63
4.1	Illustration of the proposed decision-making framework. The initial belief for different intentions (i.e., LC, LK) is marked by the <i>red</i> arrows on the agent vehicles with length representing the probability. The ego vehicle generates a set of sequential semantic-level policies according to the DCP-Tree. Each behavior sequence is simulated in closed-loop (marked by <i>black</i> dots) considering nearby agents. The risky scenario (in <i>orange</i>) in which the leading vehicle inserts to the ego target lane is identified by our CFB mechanism.	65
4.2	Illustration of the proposed behavior planning framework (in the <i>blue</i> box) and its relationship with other system components.	67
4.3	Comparison of MPDM (left) and EUDM (right). The ego vehicle (<i>gray</i>) needs to conduct an overtaking maneuver. The simulated behaviors of MPDM are fixed in the planning horizon. The ego vehicle cannot make a lane-change-left (LCL) decision until it passes the blocking vehicle, so the generated plan is local and reactive. EUDM considers the change of behavior in different future stages, which results in a consistent and farsighted plan.	69

4.4	Illustration of the proposed DCP-Tree and the rebuilding process after its <i>ongoing</i> action changes. Suppose there are three discrete semantic-level actions $\{a_1, a_2, a_3\}$ and the height of the tree is three. The <i>ongoing</i> action for the left and middle tree is a_1 , and the best policy is the dashed <i>purple</i> trace. After executing a_1 , the <i>ongoing</i> action switches to a_3 , and the DCP-Tree is updated to the right tree. Each trace only contains one change of action.	70
4.5	An illustrative example from MOBIL model [4] .	75
4.6	Open-loop test using onboard sensing data. Up: The ego vehicle is approaching another vehicle and making a LC decision (1 and 2). After the ego vehicle finishing LCR, EUDM detects several risky scenarios due to the uncertain intention of the front vehicle. Bottom: Another vehicle overtakes the ego vehicle and merges into our lane aggressively (1 and 2). EUDM captures the risky situation and takes a proper deceleration policy (3).	76
4.7	Illustration of different decision-making results in a conflict zone.	79
5.1	Illustration of our trajectory generation framework. Complex semantic elements of the environment are projected to the spatio-temporal domain w.r.t. the reference lane. The SSC encodes the requirements given by the semantic elements and a safe trajectory is generated accordingly. Note that the visualization of the static obstacles is clipped to show the details of other components. More examples can be found in the video https://www.youtube.com/watch?v=LrGmKaM3Iqc .	83
5.2	Illustration of the proposed trajectory generation framework and its relationship with other system components.	85
5.3	Illustration of merging into congested traffic under a speed limit. For the two potential behaviors, i.e., lane change and lane keeping, the optimal trajectories are generated inside the SSC for each behavior.	85
5.4	Illustration of a toy example of the SSC generation algorithm in the <i>st</i> domain. There is a speed limit which takes effect between the two <i>orange</i> boundaries, as shown in (a). To begin, the first initial cube is inflated until the two inflation directions touch the semantic boundary and the obstacle, as shown in (b). Next, the last seed in the first cube and the first seed outside the first cube are picked out to construct the second initial cube, as shown in (c). The inflation for the second initial cube terminates at the semantic boundary. Then for the third initial cube, the inflation direction opposite to the entry direction is disabled. After the cube inflation, a cube relaxation process is applied depending on the constraints associated and the free-space, as shown in (d).	87
5.5	Illustration of using the convex hull property to constrain a velocity profile inside a feasible region (dashed red lines).	93
5.6	Illustration of an unprotected left turn in a busy urban intersection. When the ego vehicle is approaching the intersection, it finds the left turn is not feasible and it reduces speed to wait. Once feasible, the vehicle quickly accelerates to complete the left turn.	95
5.7	Illustration of overtaking on an urban expressway.	97
5.8	Illustration of the comparison on a benchmark track.	98

- 6.1 Illustration of using semantic-level action to guide the exploration of the action space. Even with coarsely discretized control actions (e.g., three discretized longitudinal velocities and three lateral accelerations), the action space to be explored (*gray*) of a POMDP-based planner is huge during a multi-step look-ahead search, while much of the space explored is of low-likelihood. The situation is much worse when considering the multi-agent setting. After incorporating semantic-level action (e.g., pursuing center lines with different aggressiveness) using predefined controllers, the exploration (*purple*) can be guided using domain knowledge. Although much of the exploration is tailed, the resulting exploration still faithfully captures the potential high-level decisions. 103
- 6.2 A diagram of EPSILON together with environment understanding and execution modules. HD-Map is optional for EPSILON. 105
- 6.3 Illustration of the relationship between behavior planning and motion planning defined in this chapter. The controlled vehicle (marked in *purple*) can yield and insert behind the nearest vehicle (maneuver ①), or accelerate and insert between two nearby vehicles (maneuver ②). Visualized in the spatio-temporal domain, the two maneuvers belong to two different homotopy classes. The behavior planning targets at discovering diverse tactical maneuvers, while the motion planning is for generating a local smooth and safe trajectory. 106
- 6.4 Illustration of two representative scenarios collected from road test. The tracklets are represented by *blue* bounding boxes, the static obstacles are marked in *red* and the controlled vehicle is represented using the model vehicle. In the visualization, we illustrate the forward simulation results both for the controlled vehicle and surrounding vehicles using *rainbow* dots where the color reflects the elapsed time. For a clear visualization, we only draw the forward simulation results for the most likely scenario, and the corresponding intention estimations are marked by *yellow* arrows. In these two cases, the user input a stick signal which requires the planning system to conduct a lane change for the user. EPSILON automatically finds the best time and the best gap to smoothly and safely complete the merging requirement. On the right we plot the dynamic profile (incl. velocity, longitudinal and lateral acceleration) during the process. 110
- 6.5 Illustration of two automatic lane change scenarios dense traffic. The visualization scheme is the same as Fig. 6.4 and the difference is that the lane changes in Fig. 6.4 are upon user requests while here the lane changes are proposed by the planning system. 111

LIST OF TABLES

2.1	Illustration of the on-demand graph construction.	24
2.2	Comparison of different kinodynamic planning approaches.	38
2.3	Comparison of different trajectory optimization approaches on random map.	38
2.4	Run-time analysis on different maps	42
2.5	Performance of different replanning systems	45
3.1	Feature selection for local maneuver encoding.	59
3.2	Network specifications of the VBIN.	59
3.3	A comparison of different prediction methods. The methods marked with † are interaction aware.	60
3.4	Critical misclassifications over 13, 338 predictions.	62
4.1	Comparison of different decision-making approaches.	80

Motion Planning and Decision-Making for Autonomous Systems: From Quadrotors To Autonomous Vehicles

by

Wenchao Ding

Department of Electronic and Computer Engineering

The Hong Kong University of Science and Technology

Abstract

Autonomous systems have been widely deployed in various field applications, such as inspection, exploration, and aerial videography using micro aerial vehicles (MAVs), and highway pilot (HWP), rush-hour pilot (RHP), robo-taxi in field of intelligent vehicles. For accomplishing such missions in challenging environments, navigation functionality is essential.

In this thesis, we study practical and efficient planning methods for autonomous systems, which can achieve full autonomy in real-world complex environments. Our study covers both low-level motion planning problems and high-level decision-making (i.e., behavior planning) problems, targeting for safe and smooth autonomous navigation with minimum user intervention. We investigate the navigation problem for two different platforms, namely, quadrotors for MAVs, and autonomous vehicles (AVs), which have significant practical impacts while illustrating different focuses and functionalities.

We start with a novel kinodynamic motion planning method for quadrotors, which achieves safe and efficient replanning in unknown cluttered 3-D environments. Leveraging the knowledge of motion planning for quadrotors, we further study the planning problem for autonomous vehicles in highly dynamic environments. Different from the motion planning for quadrotors, planning for AVs requires reasoning about other dynamic traffic participants and social compliance. To this end, a novel behavior prediction method is proposed to understand the behaviors of

other traffic participants in the real world. The behavior prediction is then tightly incorporated into our proposed decision-making framework which generates robust decisions efficiently. The output decision is subsequently utilized by our novel motion planning module to generate a smooth and safe trajectory for closed-loop execution. We assemble all the individual modules into a complete and robust planning system which is validated in real-world dense city traffic.

PREVIEW

CHAPTER 1

INTRODUCTION

The development of mechanization and autonomy has always been the source of the industrial revolution. New machine tools have boosted production efficiency in a wide range of field applications, such as industrial machines, automobiles, etc. Unlike robots that work in simple and dedicated environments (e.g., manipulators in the industry), mobile robots need to work in fast-changing environments. There are numerous kinds of mobile robots emerging nowadays, such as micro aerial vehicles (MAVs), micro-ground vehicles (e.g., sweeping robots), autonomous vehicles (AVs), etc. Onboard perception and planning capabilities are essential for mobile robots to ensure their general applicability.

In this thesis, we focus on planning for *fully* autonomous navigation in complex environments. Specifically, we focus on two typical autonomous systems, namely, MAV and AV. The problem which planning needs to solve can be concluded in one sentence: how to accomplish the user-input task *safely*, *smoothly* and *optimally* with *minimum intervention*? For example, we may require MAVs to inspect some narrow, confined environments for us. Or we input our destination and require AVs to take us to the destination without any human intervention. The first term “safely” is straightforward: the robot should not collide with any obstacle even faced with unknown, cluttered, or changing environments. The second term “smoothly” can be interpreted differently depending on the application. For example, in autonomous driving, this term poses comfort constraints, while for micro aerial robots this term represents that the planned trajectory should be smooth enough so that the controller can follow. The third term “optimally” requires an elegant plan. For instance, the trajectory for MAVs should be energy-efficient, while the decisions of AVs should illustrate enough intelligence (e.g., automatically overtakes when appropriate). The fourth term “minimum intervention” means that the planning system should handle all the circumstances which the robot will potentially encounter in the operational design domain (ODD). This is the ultimate goal of using an autonomous system, namely, “click, go and done”, and this is exactly where the working efficiency and user-experience come from.

To this end, we target at building complete planning systems for MAV and AV to approach the goal of fully autonomous navigation. To build such systems, several research problems are raised and solved in this thesis. Specifically, navigating quadrotors smoothly in unknown

whole planning system into several components, namely, behavior prediction, decision-making, and motion planning. We specify the research problems for these aspects one by one and finally we integrate these modules into a complete planning system for autonomous vehicles.

At a high level, although quadrotors and autonomous vehicles may seem quite different, they share a similar hierarchical structure as shown in Fig. 1.1. For quadrotors, the perception layer mainly focuses on sensing static obstacles and reconstructs accurate 3-D map for the environments, while the planning layer basically focuses on the low-level motion planning. On the other hand, for autonomous vehicles, the perception layer may include broader categories including perceiving the traffic signals, road structures, dynamic objects, etc. At the same time, since autonomous vehicles need to drive in social environments, they require a recognition layer to identify and understand complex social behaviors. For example, the recognition layer may include behavior and trajectory prediction for other traffic participants. Moreover, since autonomous vehicles need to accomplish complex user-input tasks (e.g., highway pilot, rush-hour pilot, shared control), their planning system requires high-level behavior planning to facilitate accomplishing the high-level goals specified by users.

Generally speaking, the perception layer is “getting there”. The reason is that from algorithmic perspective, the perception algorithms are becoming mature in terms of accuracy, latency and robustness. However, how to guarantee the perception quality with low-cost sensor suites remains a challenging problem. For example, with high-end lidars, depth sensing and object tracking can be extremely accurate. However, high-end lidars are too expensive to be deployed widely in industrial products. Perception with low-cost sensors such as cameras and noisy inertial measurement units (IMUs) is the current research direction. Honestly speaking, the recognition and planning layer are quite preliminary in the current stage. The algorithms are far from becoming mature and no consensus is reached. As a result, autonomous vehicles nowadays still seem “stupid”. The control layer is mature since the control algorithms have been mature for decades. The remaining challenges mainly involve controlling complex systems such as a truck with payload.

To summarize, in this thesis we mainly focus on the two layers we mark as “need more work”, as in Fig. 1.1, namely, the recognition layer and planning layer. For the recognition layer, we study the behavior prediction problem for autonomous vehicles in Chap. 3. For the planning layer, we study the high-level decision-making problem in Chap. 4 while the low-level motion planning problems in Chap. 2 and Chap. 5.

1.1.1 Motion Planning for Quadrotors in Unknown 3-D Environments

Consider a typical application that we need the quadrotors to complete an inspection task with full autonomy. We can derive two basic requirements for this problem. First, the environment is probably unknown beforehand. Second, the quadrotor may need to deal with narrow and confined 3-D environments such as inspection during disaster response. Considering these two problems together, the quadrotor is required to *replan* constantly when encountering unforeseen obstacles. To this end, we focus on trajectory replanning for quadrotors. However, replanning is a challenging task. First of all, the replanning should be highly efficient especially during fast autonomy. As such, heavy-weight motion planning methods can not be adopted. Second, replanning typically starts from nonstatic initial states, i.e., the quadrotor is not static for replanning. This poses additional feasibility constraints for planning. Many previous works [5–9] may be problematic when faced with nonstatic initial states. To this end, kinodynamic replanning is proposed in this thesis to accomplish efficient dynamically feasible replanning.

1.1.2 Behavior Prediction for Autonomous Vehicles

When coming to the problem of autonomous driving, we have to consider other traffic participants. However, traffic participants are known to be stochastic, noisily rational, and hard to predict. Therefore, it is essential to build behavior models for traffic participants. During on-line execution, through the inference process based on past observations, we can interpret the future motions of other traffic participants. However, building such behavior models is complex due to the multi-modal nature of the problem (e.g., thinking about predicting a vehicle in a uncontrolled intersection). Moreover, the uncertainty of the model increases dramatically as the prediction horizon increases. The problem is that, for planning purpose we need long-term prediction up to 5 to 10 s, while prediction is only accurate for short term (less than 3 s) in many existing works [10–13]. To this end, we require accurate long-term prediction while accounting for the multi-modal nature of the behaviors. And in this thesis, we work towards extending the prediction horizon by modeling additional multi-agent interaction information and build a learning-based behavior prediction pipeline.

1.1.3 Decision-making for Autonomous Vehicles

After building up the understanding of other traffic participants, the remaining problem is how to make intelligent decisions to interact, cooperate, or even compete with them. This problem

can be characterized as decision-making in a multi-agent setting. When considering onboard autonomous driving, the problem becomes more complex. Since for onboard driving, all the observations come from onboard sensors, while sensing is hardly perfect. Various restrictions (e.g., occlusion, shading) impose errors in perception, which will be then propagated to prediction. As a result, from the decision-making perspective, the prediction received may be uncertain. The core problem is how to efficiently conduct safe decision-making in uncertain and multi-agent environments. In this thesis, we propose an efficient uncertainty-aware decision-making framework to tackle the above challenges.

1.1.4 Motion Planning for Autonomous Vehicles

The difference between decision-making and motion planning is that decision-making cares about a rough course of behavior plan which can be typically represented by a preliminary sequence of states, while the role of motion planning is to generate a smooth and safe trajectory in the local feasible solution space surrounding the initial guess provided by the decision-making. Unlike quadrotors which typically operate in static or semi-static environments, the challenge for motion planning of autonomous vehicles is to deal with dynamical obstacles. To this end, it is essential to take temporal information into account. Moreover, constraints or cost engineering has been a headache for motion planning, since there are various constraints imposed by the semantics of the environments such as lane geometry and traffic rules. We require a unified trajectory generation process which is easy to tune and seamlessly adapt to diverse environments. In this thesis, we propose a spatio-temporal semantic corridor structure to achieve this goal.

1.1.5 System Integration for Autonomous Vehicles

Based on our studies on individual components, it is time to assemble these modules into a complete planning system. In academia, a lot of planning systems have been proposed claiming various kinds of novelties. However, few of them are validated on a real autonomous vehicle. In industry, various demos are presented showing promising autonomy, but few technical details are published. For autonomous driving, the applicability of a planning system cannot be verified without onboard experiments. The reason is that real-world driving is constantly different from simulation and dataset, no matter how multi-modal and accurate the simulation or dataset is. To this end, we require verifying the whole planning system on a real autonomous vehicle with purely onboard sensors, and achieve closed-loop driving in dense city traffic. In this thesis, we achieve this goal leveraging all the advances made in this thesis.

1.2 Thesis Overview

The whole thesis is devoted to answer the following question:

How to accomplish the user-input task *safely*, *smoothly* and *optimally* with *minimum intervention*?

This question is repeatedly raised throughout this thesis and will be answered in detail in the following chapters.

We start this thesis by introducing our motion planning framework for quadrotors in unknown complex 3-D environments (Chap. 2). The planning system is implemented on a quadrotor with only one monocular camera and one IMU. The system is extensively validated in both simulation and real-world experiments.

Based on the understanding of motion planning for quadrotors, we turn to the planning system for autonomous vehicles. Unlike planning for quadrotors, motion planning module is only one particular component for the planning system of autonomous vehicles. Before stepping into the motion planning for autonomous vehicles, we first introduce behavior prediction for autonomous vehicles (Chap. 3) which is employed to model traffic participants for driving. This behavior prediction module is further incorporated into the decision-making module (Chap. 4) which provides high-level decisions for the ego (controlled) vehicle in interactive driving environments. The interesting part is that our decision-making module (Chap. 4) can account for uncertain predictions provided by (Chap. 3), which is critical in real-world driving. Since in real-world, perfect prediction never exists. A robust decision-making system has to take the responsibility of tolerating uncertainties of upstreaming modules while still being able to ensure safety.

Given the decision, the remaining problem is to generate a smooth and safe trajectory which faithfully follows the decision. A motion planning module for autonomous vehicles is then introduced in Chap. 5. Having all the individual components, in Chap. 6 we assemble them into a complete planning system, and conduct field tests in real world to verify its effectiveness. The story is then finished and conclusions are drawn in Chap. 7, where we also point out promising future directions for the planning community.

1.3 Thesis Contributions

This thesis contributes to several aspects of motion planning and decision-making for two autonomous systems, i.e., MAV and AV. The contributions are not only algorithms but also complete system designs and extensive real-world validations.

In Chap. 2, we propose a complete and efficient kinodynamic replanning for quadrotors. This framework is designed to overcome the problem that traditional methods often generate non-smooth or even infeasible trajectory under non-static initial states. The kinodynamic search we propose provides superior performance in terms of the optimality and efficiency tradeoff compared to several state-of-the-art methods. Moreover, we implement the whole framework on two different vision-based platforms: one quadrotor with one camera and one IMU (monocular) and the other with dual-fisheye vision. Although the perception quality and field of view are quite different for two platforms, our planning framework consistently provides safe and smooth performance in real-world unknown 3-D environments.

In Chap. 3, we illustrate our first step into the field of autonomous driving. Before considering planning for autonomous vehicles, we have to understand the driving environments at the first hand. In this chapter, we propose a learning-based behavior prediction method for autonomous vehicles. This method is dedicated to solving the problem that the prediction horizon of existing methods is insufficient. However, planning requires long-term prediction. To this end, we propose utilizing interaction information among agents to introduce an additional clue for prediction and achieve extending the prediction horizon. Our method has superior accuracy and a longer prediction horizon compared to several state-of-the-art methods. We validate the method using a real-world highway driving dataset.

In Chap. 4 and Chap. 5, we finally approach the problem of planning for autonomous vehicles after obtaining the understanding of other traffic participants. In Chap. 4, we propose an efficient uncertainty-aware decision making (EUDM) framework. EUDM is designed to be robust to uncertain predictions while being highly efficient, which addresses two key concerns of decision making (i.e., efficiency and uncertainty). In Chap. 5, we introduce how to realize a particular decision to an executable trajectory. We propose a unified abstraction which wraps various constraints and obstacles in the driving scenario, i.e., the spatio-temporal semantic corridor (SSC). With the help of SSC, the trajectory generation problem boils down to a straightforward optimization formulation. Moreover, the SSC makes the formulation free of tuning, which is the key challenge for motion planning for autonomous vehicles.

In Chap. 6, we assemble a complete planning system for autonomous vehicles, based on our accumulated contributions. The key feature of our system that we not only have a systematic and consistent design but also validate our system in real-world city traffic. Although many existing works have been published on the planning for autonomous vehicles, few of them are validated in the real world. In this Chap. 6 we validate the real-world impact of our planning methods.

PREVIEW