POLITECNICO
DI TORINO

Dipartimento di
Automatica e Informatica

PhD in Computer and Control Engineering

XXXI cycle

Supervisor

Prof. Gianpiero Cabodi

Data fusion methods and algorithms for automotive applications

PhD Candidate: *Francesco SAVARESE*

1. Context

Path Planning is a crucial task for autonomous systems. It exploits data retrieved from different sensors fused together.

2. Goal

Two different path planning problems are presented:

- A GPGPU based trajectory planner for autonomous car
- A path planner for solving the “door opening” problem in the context of indoor robotics

3. GPGPU Based Trajectory Planner – Baseline

A common structure for a path planner is presented in Figure 1. It highlights the interaction between the path planner itself and the external environment. A trajectory planner works close to the vehicle controller. It provides low level trajectories to follow. Because of its role it is subject to hard real time constraints. Aim of this work is the time reduction for trajectory generation and the maximization of the environment exploration. As baseline algorithm we choose a sample based trajectory planner [1]. These types of algorithms generate feasible trajectories in the control space of the vehicle and then check for the collision avoidance. The algorithm generates and collects a set of trajectories in a tree structure where each node represent a future possible state of the vehicle. A cost function selects the best one. We used the CPU based algorithm as a baseline to tune the algorithm parameters according to some metrics we defined:

- **RMSE** between the curve drawn by the vehicle and the trajectory to follow
- **Reaction Time** for starting a maneuver after an obstacle discovery
- **Safety Distance** measuring the minimum distance from the obstacle

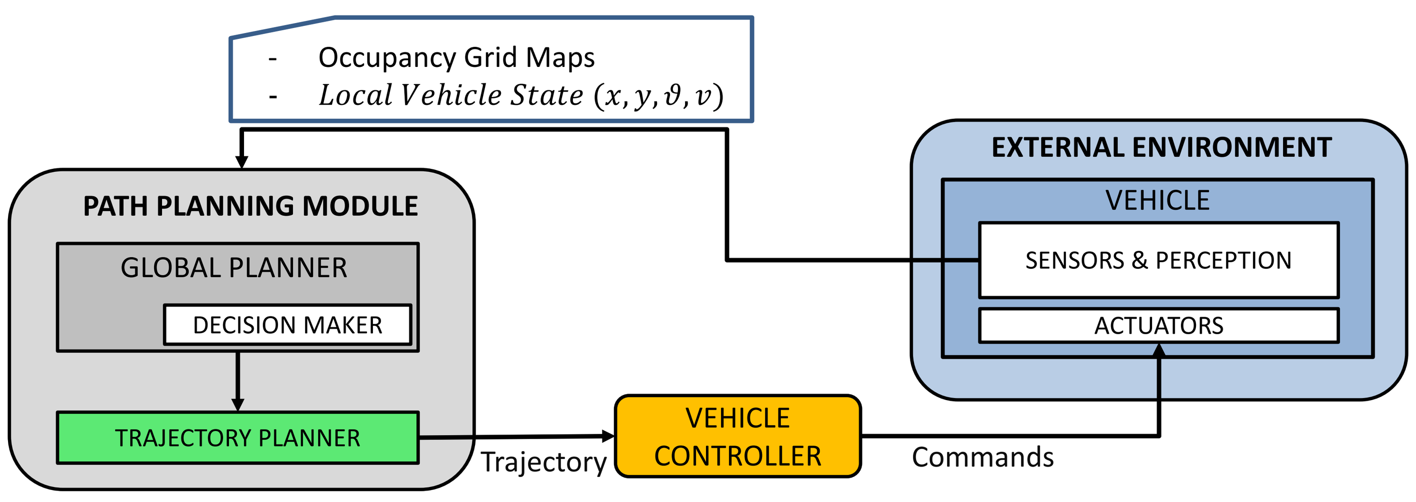

Figure 1: Path Planner architecture

4. GPGPU Based Trajectory Planner

The GPGPU version parallelly generates each node for each layer by applying a breadth first strategy as shown in Figure 2. The node generation is split in two phases: (1) Future state prediction and (2) Command generation to reach the predicted position. Two different CUDA kernels take care of these different phases. We accomplished a wider space exploration around the vehicle and a time reduction for trajectories generation. This result is quantified in the Table 1.

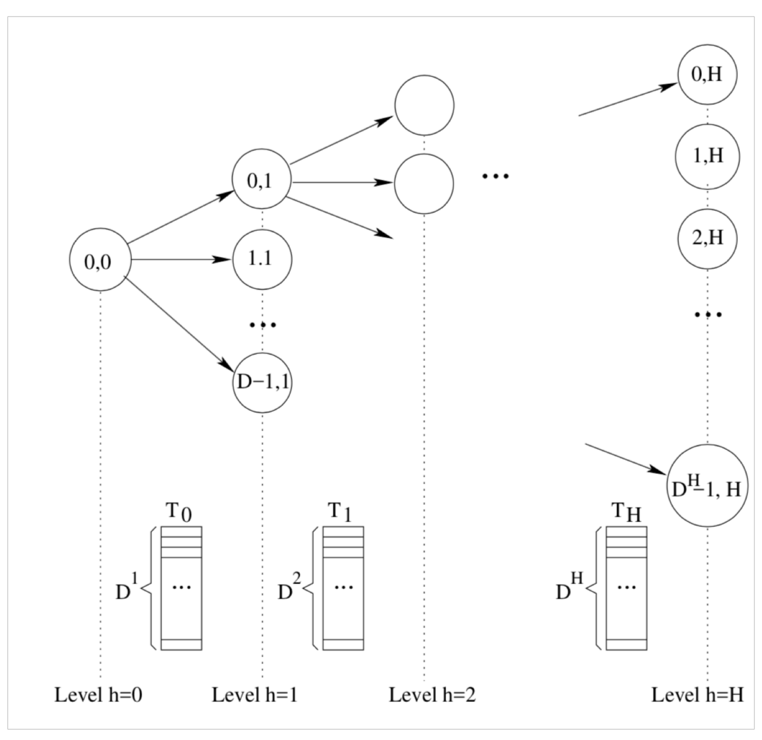

Figure 2: Tree, reserved memory to communicate information between layers, and mathematical dependency among tree nodes

D	H	# TRAJECTORIES	CPU M [KB]	CPU T_1 [ms]	CPU T_2 [ms]	CPU Tot [ms]	GPU M [KB]	GPU T_1 [ms]	GPU T_2 [ms]	GPU Tot [ms]
6	3	216	10	15	4	19	16	2	2	4
5	4	625	32	26	7	33	51	3	2	5
6	4	1296	62	45	10	55	99	9	3	12
5	5	3125	156	77	14	86	250	11	4	15
4	6	4096	218	83	16	99	349	12	6	18
5	6	15625	781	218	44	262	1250	38	18	46

Table 1: Computation time comparison between CPU and GPGPU Algorithm. D is the tree degree, H the tree height. T_1 is the tree generation time, T_2 the cost function computation.

The controller needs a new trajectory every 20 ms. To meet this constraint the highlighted row represents the best parameters combination because memory transfer time have to be added for the GPGPU version.

5. The Door Opening Problem*

One of the goal of the indoor robotics is to create robots that can help people in their daily environment, improving the quality of life. In this context the robot should be able to handle complex situation without human help. As demonstrated by several published papers [2][3] to open a closed door is an attractive problem and still far to be solved. It strictly depends on the robot type. We used the Toyota HSR. Main contribute of this work is the design of a state machine, in a ROS environment, able to manage pushing and pulling doors even in an unknown environment. To understand the door type, after grasping, the torque on the wrist sensor is monitored. For the handle and door recognition we applied a deep learning approach using the state of the art SSD [4] deep network. In order to do so we first created a handles and doors dataset. The approach is described in Figure 3. The QR code below shows a brief video of the robot at work.

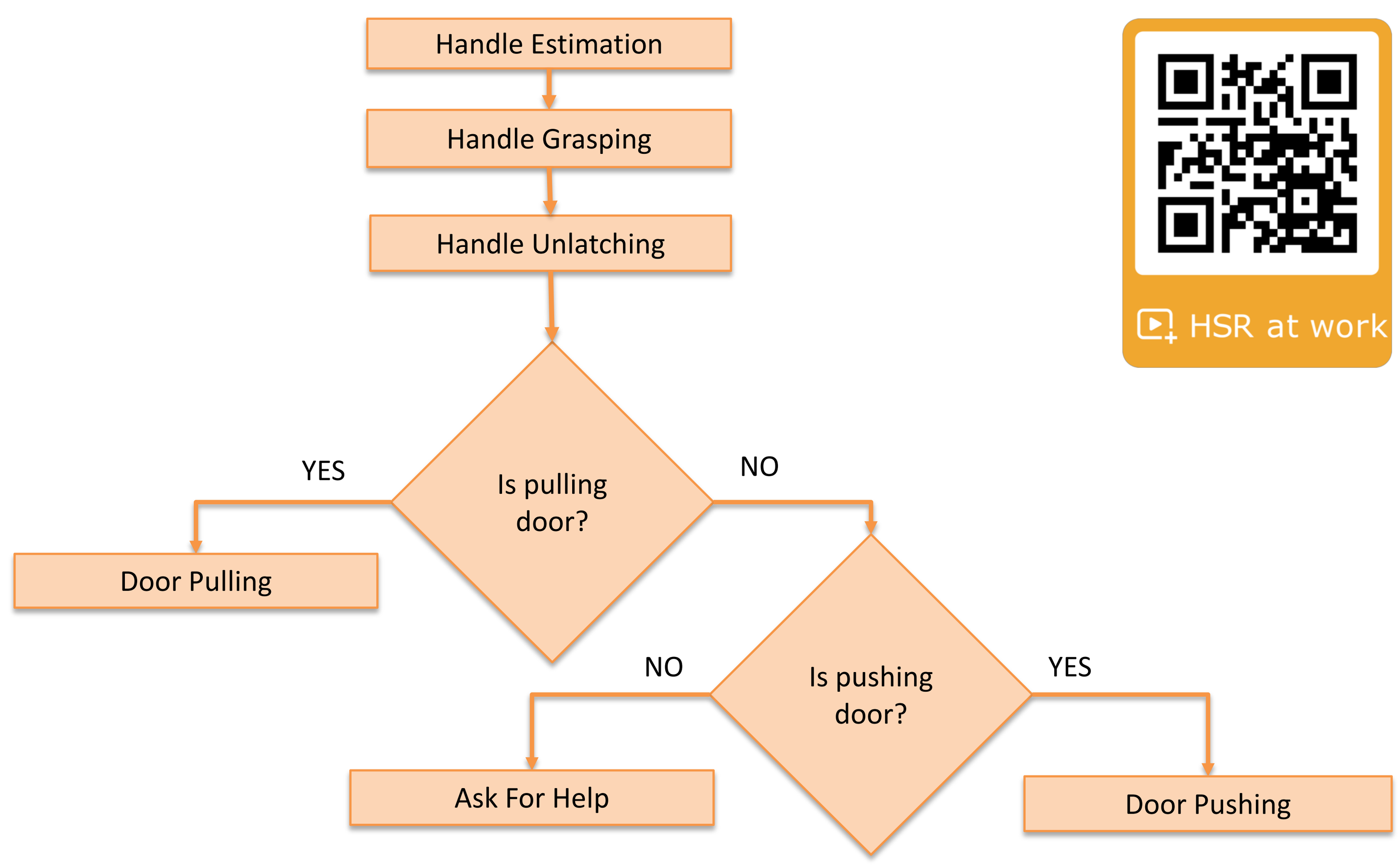

Figure 3: Door Opening High Level Approach

6. References

1. U. Schwesinger, M. Rufli, P. Furgale, and R. Siegwart. A sampling-based partial motion planning framework for system-compliant navigation along a reference path. In 2013 IEEE Intelligent Vehicles Symposium (IV), pages 391–396, jun 2013
2. W. Meeussen et al., "Autonomous door opening and plugging in with a personal robot," 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK,2010,pp.729-736. doi: 10.1109/ROBOT.2010.5509556
3. S. Chitta, B. Cohen and M. Likhachev, "Planning for autonomous door opening with a mobile manipulator," 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, 2010, pp. 1799-1806. doi: 10.1109/ROBOT.2010.5509475
4. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C., & Berg, A.C. (2016). SSD: Single Shot MultiBox Detector. ECCV.

*This research have been made in collaboration with The Tokyo University and in the context of the Robocup2018.