Car-Like Mobile Robot Navigation: A Survey

Sotirios Spanogianopoulos and Konstantinos Sirlantzis

Abstract Car-like mobile robot navigation has been an active and challenging field both in academic research an in industry over the last few decades, and it has opened the way to build and test (recently) autonomously driven robotic cars which can negotiate the complexity and uncertainties introduced by real urban and suburban environments. In this chapter, we review the basic principles and discuss the corresponding categories in which current methods and associated algorithms for car-like vehicle autonomous navigation belong. They are used especially for outdoor activities and they have to be able to account for the constraints imposed by the non-holonomic type of movement allowable for car-like mobile robots. In addition, we present a number of projects from various application areas in the industry that are using these technologies. Our review starts with a description of a very popular and successful family of algorithms, namely the Rapidly-exploring Random Tree (RRT) planning method. After discussing the great variety and modifications proposed for the basic RRT algorithm, we turn our focus to versions which can address highly dynamic environments, especially those which become increasingly uncertain due to limited accuracy of the sensors used. We, subsequently, explore methods which use Fuzzy Logic to address the uncertainty and methods which consider navigation solutions within the holistic approach of a Simultaneous Localization and Mapping (SLAM) framework. Finally, we conclude with some remarks and thoughts about the current state of research and possible future developments.

Keywords Rapidly-exploring random trees (RRT) · Simultaneous localization and mapping (SLAM) · Sensor-based methods · Fuzzy logic · Path planning

---

S. Spanogianopoulos (✉) · K. Sirlantzis
School of Engineering and Digital Arts, University of Kent, Canterbury, UK
e-mail: ss976@kent.ac.uk

K. Sirlantzis
e-mail: K.Sirlantzis@kent.ac.uk

G.A. Tsihrintzis et al. (eds.), *Intelligent Computing Systems*,
Studies in Computational Intelligence 627, DOI 10.1007/978-3-662-49179-9_14

1 Introduction

During the past few years there has been significant progress in navigation applied to outdoor robots with several industrial applications in well defined environments. At the same time there is still a need of fundamental breakthroughs in autonomous systems to make them reliable in much less structured environments.

Once a reasonable representation of the environment is obtained, the vehicle needs to be controlled to perform a certain path. Path following has three main stages: navigation, path planning and guidance. The navigation module is usually responsible for the localization of the vehicle within a given map. The path planning module deals with defining global as well as local paths and the guidance module is responsible for keeping the car on the defined path within acceptable errors. The application of such a techniques has many applications in areas such as robotics, manufacturing, pharmaceutical drug design, computational biology and computer graphics.

2 RRT-Based Methods

2.1 Unsafe Path Planning

The real time operation of a car-like vehicle in a large unknown/uncertain environment is still a very challenging problem. In Pepy et al. (2006), in order to face unsafe path planning during the navigation process, authors are using a five degrees of freedom dynamic car model which leads to better placed estimated points, considering skidding and sliding. Using a vehicle model identical to the car that will have to follow the planned path, they use a Rapidly-exploring Random Tree (RRT) planner to quickly explore the whole configuration space. The algorithm focuses also to the computation of the unexplored parts of the space by breaking the large Voronoi areas (Fig. 1).

An interesting feature of this algorithm for path planning is that a path planned with RRT does not need local planner to find a way from a configuration to another. It also allows the RRT to rapidly explore in the beginning, and then converge to a uniform coverage of the space.

2.2 Safe Path Planning

In Pepy and Lambert (2006) authors address the problem of safe path planning in an uncertain-configuration space. Considering the case of a car-like robot moving in an indoor environment (three-dimensional space), they used the Extended Kalman Filter (EKF) to localize such a robot and to estimate its configuration uncertainty

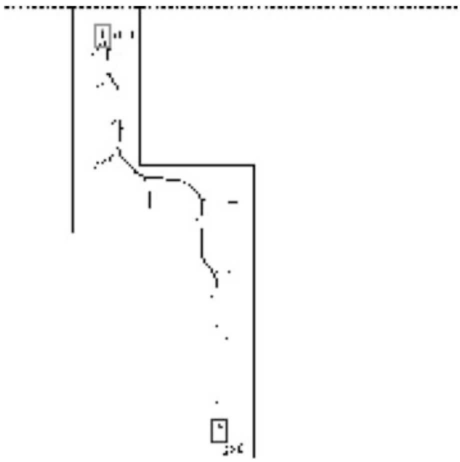

Fig. 1 Obstacle avoidance using kinematic model (Pepy et al. 2006)

during navigation. More specifically, their car-like robot involves non-holonomic constraints and their planner uses an ideal indoor 2D world map, where obstacles are represented by polygonal lines. Then, they used the Rapidly-exploring Random Trees (RRT) method, which is an incremental method to quickly explore the whole configuration space, in order to visit the unexplored parts of the space by breaking the large Voronoi areas.

But since RRT is often slow to plan a path as it randomly reaches a defined position, so in order to speed up the algorithm, they biased it towards the goal by modifying a specific function of the system (goalbias and goalzoom modifications). In the proposed method, authors argue that it is also possible to use two RRTs to plan paths faster, and using this bidirectional RRTs, the path is planned when these two trees meet each other. But since the current configuration of the robot is always uncertain and has limited accuracy, authors used also a probabilistic model to represent the uncertainty, while for the localization process, they divided it into two phases, the prediction and the estimation phase.

2.3 Rapidly Exploring Random Tree Algorithm on Rough Terrains (RRT-RT)

In Tahirovic and Magnani (2001) a different approach is presented, where a novel Rapidly exploring random tree algorithm on rough terrains (RRT-RT) has been developed for the purpose of outdoor mobile robot navigation. While other RRTs, which have been adopted for rough terrains, have been used to find a nearest neighbor from a new random state within the tree based on Euclidian distance, the presented algorithm in this work uses a roughness based metric. The metric is defined by the help of the roughness based navigation function, RbNF, which is a numerical function that provides the cost-to-go values (roughness-to-go) for each terrain location. The roughness-to-go value of a terrain location represents an approximate value of the remaining cost of the terrain traversability toward the goal position, or differently, to compute a numerical function RbNF which provides estimated roughness-to-go values for each patch of the given terrain.

This function is used either as a navigation function or as a cost-to-go term within the MPC optimization guiding the vehicle toward the more traversable areas while approaching the goal position. The RbNF has been also used for the purpose of navigation planning based on the model predictive (MPC) paradigm, which repeatedly performs the optimization within a region around the current vehicle position to generate an appropriate path toward the goal. In this approach, the fundamental difference comparing to the state of the art RRTs is comprised in a function, where, while a classical RRT algorithm uses a metric based on Euclidian distance to find the nearest vertex of the tree to a new random state, the algorithm proposed in Tahirovic and Magnani (2001) uses a measure based on the terrain roughness. As such, the simulation results show that the RRT-RT planner explores the terrain in an efficient manner, and even generates final paths that slightly deviate from paths obtained by the Dijkstra's algorithm.

2.4 RRT Motion Planning Subsystem

The work presented in Kuwata et al. (2008) provides a detailed analysis of a motion planning subsystem, which is based on the Rapidly-exploring Random Trees (RRT) algorithm and it is used to present the numerous extensions made to the standard RRT algorithm that enable the on-line use of RRT on robotic vehicles. It actually provides a path and a speed command to the controller that can track the path, such that the vehicle is able to avoid obstacles and stay in lane boundaries under the different conditions of urban driving. The proposed RRT algorithm samples the input to the controller instead of the input to the vehicle, and obtains the dynamically feasible trajectory by running forward a simulation of the closed-loop system, consisting of the vehicle model and the controller.

By using this stable closed-loop system, the method in Kuwata et al. (2008) has the advantage of enabling the efficient use of RRT algorithms on vehicles with unstable open-loop dynamics. Similar to the standard RRT, the proposed algorithm performs sampling, node selection, expansion and constraint check, while the planner keeps providing the command to the controller at a fixed rate. Finally, the best trajectory is selected and sent to the controller, and the tree expansion is resumed after updating the vehicle states and the situational awareness. To sum up the main extensions from the standard RRT, we can say that this approach firstly improves the computational efficiency and secondly the input to the closed-loop system is sampled, which also enables RRT to handle complex/unstable dynamics of the vehicle. Third, the lazy check enables RRT to focus on the tree expansion even with the constantly changing situational awareness, while forth, the uncertainty in the environment is captured in the form of a risk penalty in the tree. The

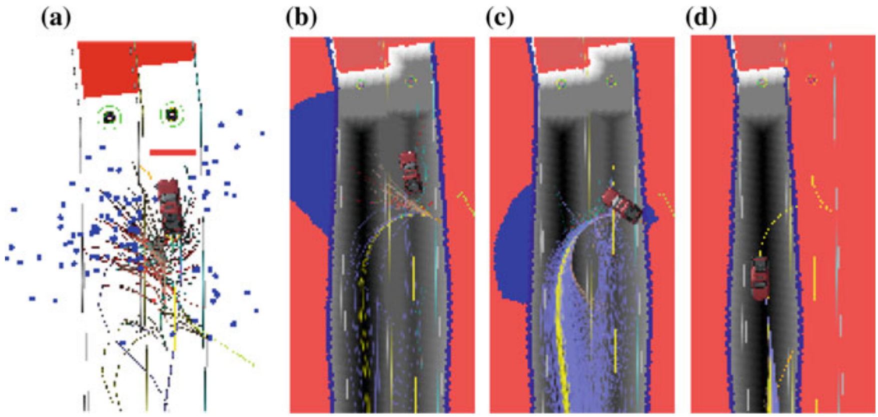

Fig. 2 Planning of a U-turn maneuver. Figures **b**, **c**, and **d** show different evolutions of the tree as the vehicle executes a U-turn. Notice in Figure **d** the trace of the path followed by the vehicle (shown in *yellow*) (Kuwata et al. 2008)

sampling using the environmental structure also significantly reduced the time to find trajectories for various maneuvers and the safety of the vehicle is guaranteed by requiring that the trajectory sent to the controller end in a stopping state (Fig. 2).

2.5 *Partial Motion Planning*

The work presented in Petti and Fraichard (2005) tries to solve the problem of computing a complete motion to the goal within a limited time. More specifically, planning in a changing environment implies to plan under real time constraints and a robotic system cannot safely remain passive, since it might be collided by a moving obstacle (decision constraint). For this reason, authors used a Partial Motion Planning (PMP) approach in order the algorithm to operate until the last state of the planned trajectory reaches a neighbourhood of the goal state. But since PMP has no control over the duration of the partial trajectory that is computed, they considered a selected milestone of a point mass robot with non zero velocity moving to the right, and depending upon its state there is a region of states for which, even though it is not in collision, it will not have the time to brake and avoid the collision with the obstacle [Inevitable Collision State (ICS)]. In order to avoid this state, they used a property which firstly proves that a trajectory is continuously safe while the states safety is verified discretely only, and secondly, it permits a practical computation of safe trajectories by integrating a dynamic collision detection module within the Rapidly-Exploring Random Tree (RRT) (Fig. 3).

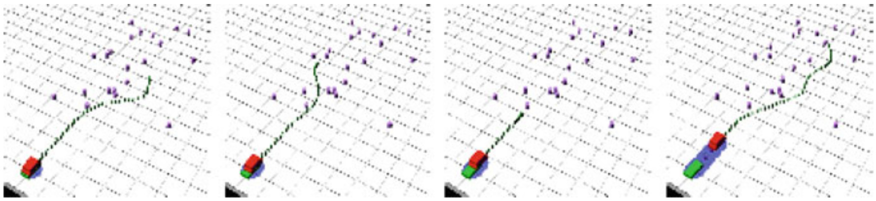

Fig. 3 Navigation within an environment cluttered with moving pedestrians (Petti and Fraichard 2005)

2.6 Sensor-Based Random Tree (SRT)

In Espinoza et al. (2006) a new method is presented for sensor-based exploration of unknown environments, which proceeds by building a data structure called Sensor-based Random Tree (SRT). The SRT structure represents a roadmap of the explored area with an associated safe region, and estimates the free space as perceived by the robot during the exploration. The technique which is used for this case is called SRT-Radial and deals with non-holonomic constraints using two alternative planners. More specifically, the method builds a data structure through random generation of configurations. Then, the SRT represents a roadmap of the explored area with an associated Safe Region, an estimate of the free space as perceived by the robot during the exploration. Depending on the shape of the Local Safe Region, the general method results in different exploration strategies and the idea is to increase the exploration efficiency by biasing the randomized generation of configurations towards unexplored areas. In order to do this, authors used the SRT-Radial strategy, which takes advantage of the information reported by the sensors in all directions, to generate and validate configurations candidates through reduced spaces.

The paper in Fulgenzi et al. (2008) describes a navigation algorithm for dynamic and uncertain environment, based on the assumption that moving obstacles are supposed to move on typical patterns which can be pre-learned and represented by Gaussian processes (GP). More specifically, they use a robot which is equipped with a distance sensor and models the static environment in an occupancy grid. The moving obstacles they used follow typical patterns with some amount of uncertainty, and these patterns are a priori known and represented with GP. Then, the moving obstacles are detected and tracked on-line and the prediction of their future position is computed on the base of the known typical paths. To update previously explored states with the on-line estimation, authors integrated the likelihood of obstacle paths and the probability of collision. The planning algorithm is based on an extension of the Rapidly-exploring Random Tree algorithm (RRT), where the likelihood of the obstacles trajectory and the probability of collision is explicitly taken into account. The algorithm is used in a partial motion planner, and the probability of collision is updated in real-time according to the most recent estimation.

2.7 RRT* Algorithm

In Karaman and Frazzoli (2013) a different approach is presented, where authors extend the RRT algorithm to handle a large class of non-holonomic dynamical systems. As it is known, sampling-based motion planning algorithms, such as the Probabilistic RoadMap (PRM) and the Rapidly-exploring Random Tree (RRT), guarantee asymptotic optimality, providing almost-sure convergence towards optimal solutions. In order to face the drawbacks of these methods (such as computational complexity, quality of the motion, etc.) and improve their results, authors modified them in the following way. First, they seek connections within boxes that are substantially larger in some dimensions than others, in order to ensure both asymptotic optimality and computational efficiency. Then, they computed the shape and orientation of these boxes for a large class of dynamical systems, based on differential geometry, which is dictated by the ball-box theorem of sub-Riemannian geometry. They then compute the optimal shape and orientation of these probabilistic trajectory boxes using the ball-box theorem of differential geometry, which is a class of sub-Riemannian problem where shapes can be constructed and joined in a "fuzzy" manner, i.e. without definitive boundary constraints.

2.8 Voronoi Fast Marching (VFM) and Fast Marching (FM2)

In Garrido et al. (2009) authors present a new method in order to improve the trajectories based in the Voronoi Fast Marching Method (VFM). The proposed method is suitable for improving the smoothness and the length of the trajectories calculated with probabilistic methods with bad quality trajectories, such as RRT or PRM. More specifically, in order to calculate the trajectories with the best properties, authors used a hybrid Path Planning method split into two parts: in the first one, the RRT method is used to obtain a first trajectory and in the second one, this trajectory is improved using the Voronoi Fast Marching Method. In order to apply VFM method, they calculated a tube around the previous trajectory. This tube is intersected with the walls and obstacles map and then the VFM method is applied. This way, authors take advantage of the best properties of the two methods, i.e. the possibility of working in many dimensions of RRT and the smoothness and the quality of the trajectories of VFM. In particular, the smoothing is calculated on a vectorial field that has the same goal point and it is repelled by obstacles and walls and admits the non-holonomic constraints. Finally, the VFM method uses the propagation of a wave (Fast Marching) operating on simple grid-based world model, to determine a motion plan over a slowness map (similar to the refraction index in Optics) extracted from the updated grid-map model.

In Garrido et al. (2011) authors present the application of Voronoi Fast Marching (VFM) and FM2 methods to non-holonomic mobile robot path planning. More

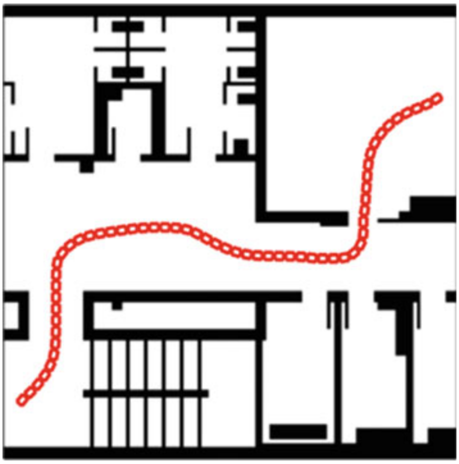

Fig. 4 Non holonomic version of the proposed method (Garrido et al. 2011)

specifically, the VFM and FM2 methods use the propagation of a wave (Fast Marching) operating on the world model (i.e. from the current position of the robot to the goal), to determine a motion plan over a slowness map (or refractive indexes, or the inverse of velocities) similar to the repulsive electrical potential of walls and obstacles, and the calculation of the path by using the gradient method from the goal to the current position point. This algorithm starts with the calculation of the Logarithm of the inverse of the Extended Voronoi Transform of the 2D updated map, in order to obtain a potential proportional to the distance to the nearest obstacles to each cell. Then they apply the Fermat's least time principle (i.e. the Fundamental Equation of the Geometrical Optics) for light propagation in order to get the refractive index and then they eliminate the poses that are not feasible. In the next step, they apply the Voronoi Extended Transform to the configuration space, along with the Fast Marching Method, and finally they calculate the trajectory from the initial pose to the goal by using the vectorial field of the expansion wave (Fig. 4).

2.9 SBL Algorithm

Authors in Balakirsky and Dimitrov (2010) presented several enhancements that improve the quality of the generated path in comparison with the simple adaptation of the Single-query, Bi-directional, Lazy roadmap (SBL) algorithm, which successfully builds upon the traditional Probabilistic Roadmaps (PRM), solving also the planning problem in the context of non-holonomic constraints of car-like robots. More specifically, the adaptations they made can be summarized as such: first, they limit the connections between milestones to a single constant curvature arc, in order to restrict the milestone's neighborhood to a relatively small subset of the geometric region to the front and rear of the robot. Secondly, they restricted the early creation

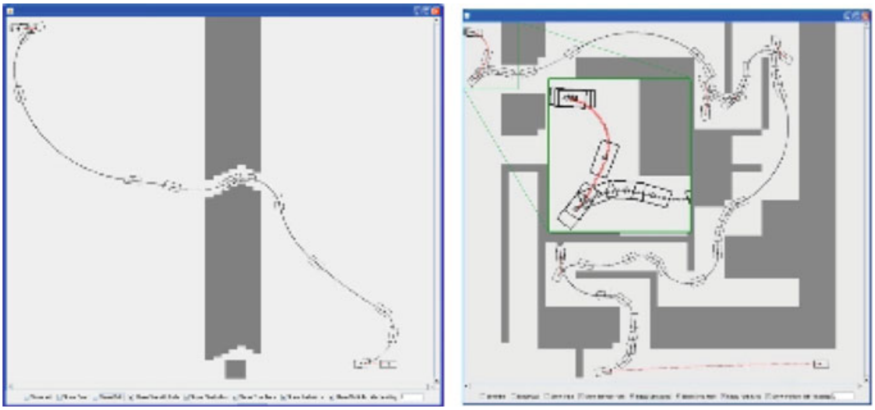

Fig. 5 PRM-AS path solution in the "corridor (maze)" environment. The magnified area (*green box*) shows the maneuver (*red curved line*) undertaken to exit the tight space (*bottom right of green box*) while positioning the vehicle for forward movement (Balakirsky and Dimitrov 2010). The maneuver (*red curved line*) also indicates the vehicle moving in reverse

of speed profiles by adding a "speed" dof to each milestone and allowing connections only between milestones with close speed values. Thirdly, they used the arc length as a metric to utilize during tree growing, tree connection, and collision checking. Forth, in order to prevent frequent switching between forward and backward motion, they used the PRM-AS with the higher probability, which generates either a forward or a backward arc according to the direction of the milestone. In addition, they defined that the modified version favors forward movement when expanding the init tree and backward movement when expanding the goal tree. They also introduced three configurable parameters, which provide a significant degree of control over the generated path by the robot, while they implemented a path smoother, in order to test each milestone for a potential connection with each of the subsequent milestones on the path (Fig. 5).

2.10 Single-Query Motion Planning

The paper in Burns and Brock (2007) presents a utility-guided algorithm for the online adaptation of the random tree expansion strategy. It is evident that the randomly expanding trees are very effective in exploring high-dimensional spaces, but as the dimensionality of the configuration space increases, the performance of the tree-based planners that use uniform expansion degrades. The proposed algorithm is based mainly on RRT and guides the expansion towards regions of maximum utility based on local characteristics of state space. More specifically, the planner guides the ongoing tree-based exploration of state space using information about state space obtained from previous tree expansions. The planner incrementally learns how to adjust the parameters of tree-based exploration based on the structure of state space that is revealed during the planning process. In order to manage it, the planner identifies expansion steps with maximal expected utility given its current knowledge of the state space. Thus, the planner uses available information to maximize expected progress towards a successful path.

2.11 Dynamic-Domain RRT

In Yershova et al. (2005) authors analyze the weaknesses of RRT when the obstacles in the configuration space are not taken into account and/or the sampling region is inappropriately chosen, and then they explain the reasons why adaptations and extensions of RRTs are generally proposed in bibliography. As a further step, they propose a general framework for minimizing the effect of some of these weaknesses by considering a new sampling strategy based on the visibility region of the nodes in the tree. More specifically, they have developed and implemented a simple new planner, which defines a boundary domain for a boundary point as the intersection of the Voronoi region of that point and an n-dimensional sphere centered at that point. In addition, authors defined also the dynamic domain of radius R for a set of points, which is the boundary domains of the boundary points combined with the Voronoi regions of all other points. They call this uniform distribution over this domain the dynamic domain distribution, and they argue that using this method, the performance of the system can become at some cases orders of magnitude better (Fig. 6).

Authors in Jaillet et al. (2005) analyze the influence of a parameter introduced in Yershova et al. (2005), which relies on a new sampling scheme that improves the performance of the RRT approach, and propose a new variant of the dynamic-domain RRT, which iteratively adapts the sampling domain for the

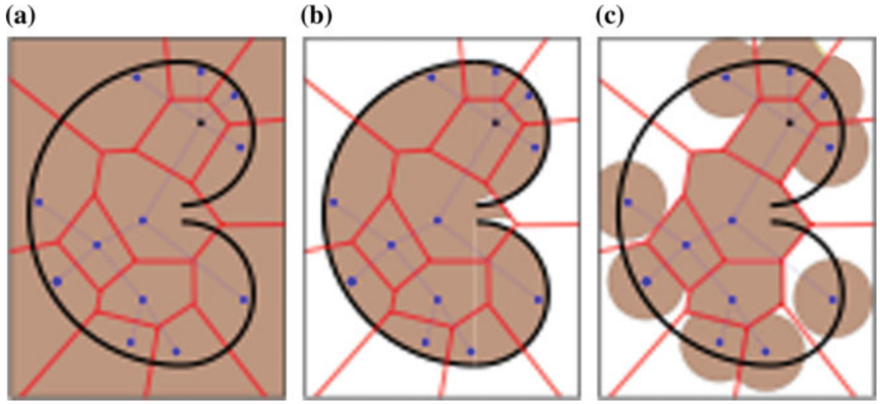

Fig. 6 For a set of points inside a bug trap different sampling domains are shown: **a** regular RRTs sampling domain, **b** visibility Voronoi region, **c** dynamic domain (Yershova et al. 2005)

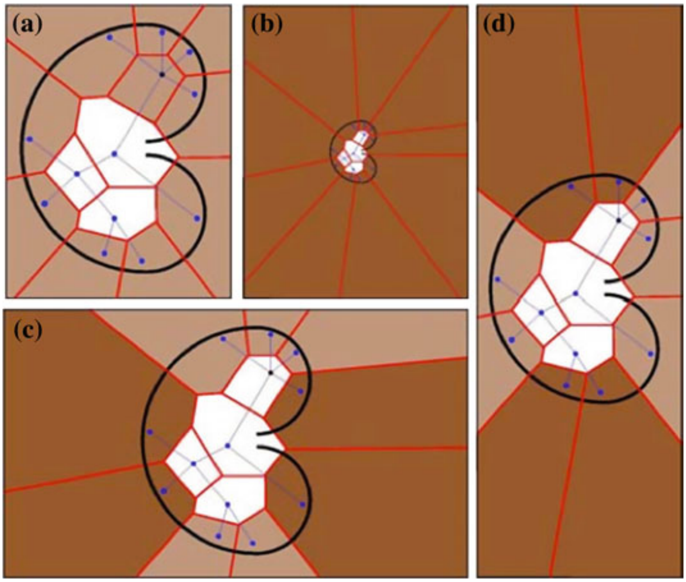

Fig. 7 For the RRT algorithm, the Voronoi region of the frontier nodes is growing together with the size of the configuration space. Therefore, depending on the boundaries of the space, the bias toward unexplored regions can be small (**a**), strong (**b**) or biased toward only some of the parts of the space (**c** and **d**) (Jaillet et al. 2005)

Voronoi region of each node during the search process. In particular, authors propose to adapt the boundary domain of a given node (i.e. its associated radius) as a function of the number of expansion attempts and failures from this node. In addition, in order to keep the probabilistic completeness of the algorithm, they ensure that the possibility for a node to be extended by putting a lower bound on the possible radius values of the nodes (Fig. 7).

2.12 Transition-Based RRT

In Jaillet et al. (2012) a new method called Transition-based RRT (T-RRT) for path planning in continuous cost spaces is presented, which combines the exploration strength of the RRT algorithm that rapidly grow random trees toward unexplored regions of the space, with the efficiency of stochastic optimization methods that use transition tests to accept or to reject a new potential state. More specifically, the proposed planner relies mainly on the notion of minimal work path that gives a quantitative way to compare path costs. The minimal work path is based on the notion of mechanical work, and the resulting loss of "energy" due to this mechanical work is the criterion that authors try to minimize. The algorithm also uses the exploration strategy of RRT resulting from the exploration bias toward

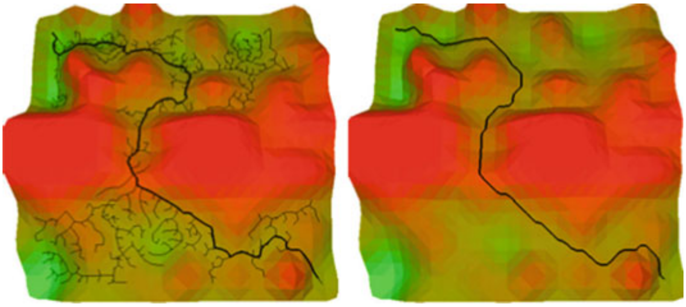

Fig. 8 Optimal paths based on the minimal work criterion (Jaillet et al. 2012)

large Voronoi regions of the space and combines it with a stochastic optimization method developed for computing global minima in complex spaces (such as Monte Carlo optimization or simulated annealing), in order to introduce it to a transition test and classify the new potential state as accepted or rejected (Fig. 8).

2.13 Parallelizing Rapidly-Exploring Random Tree (RRT) Algorithm on Large-Scale Distributed-Memory Architectures

In Devaurs et al. (2013) authors address the problem of parallelizing the Rapidly-exploring Random Tree (RRT) algorithm on large-scale distributed-memory architectures, using the message passing interface. The parallelization schemes they compared are the OR parallel RRT, the distributed RRT, and the manager–worker RRT. In the first case, they used the OR parallel paradigm for its implementation, where each process computes its own RRT and the first to reach a stopping condition broadcasts a termination message. The second and the third one belong to the category of collaborative RRTs, where all processes collaborate to build a single RRT. Parallelization is then achieved by partitioning the building task into subtasks assigned to the various processes. So, in order to achieve this, they rely on classical decomposition techniques and more specifically in the following two ways. First, since the construction of an RRT consists of exploring a search space, they used an exploratory decomposition, where each process performs its own sampling of the search space, without any space partitioning involved, and maintains its own copy of the tree, exchanging with the others the newly constructed nodes. This leads to a distributed (or decentralized) scheme where no task scheduling is required, aside from a termination detection mechanism. Secondly, they perform a functional decomposition of the task, leading to the choice of a manager–worker (or master–slave) scheme as the dynamic and centralized task-scheduling strategy, where the manager maintains the tree, and the workers have no access to it. The proposed methods can be used in mainly two cases. First, problems whose variability in sequential runtime is high can benefit from the OR parallel RRT, while problems for which the computational cost of an RRT expansion is high can benefit from the distributed RRT and manager–worker RRT.

In Rodriguez et al. (2006), a variant of the Rapidly-Exploring Random Tree (RRT) path planning algorithm is presented, which is able to explore narrow passages or difficult areas more effectively. More specifically, authors in Rodriguez et al. (2006) used some obstacle hints for directions to grow the tree for path planning in order to find difficult areas of configuration space (C-space). The planner they used uses obstacle vectors obtained from the obstacle and it focuses on how the tree decides to grow. They presented nine possible ways to expand a tree, in which the orientations to grow are either the same as the source configuration or random orientations. They also proposed a modification to a greedy algorithm for calculating the planner's path, such that it would take as big a step length as possible, as long as it is less than some maximum step length specified. Based on these modifications, authors argue that they could result in a path planning technique that solves motion planning problems more quickly and efficiently than other techniques (Fig. 9).

The paper in Phillips and C.S. Draper Laboratories (2004) presents a path planning algorithm for handling systems with constraints on controls or the need for relatively straight paths for real-time actions. The initial phase of the algorithm finds an efficient path using guided Expansive Spaces Trees (guided ESTs) and focuses on a randomized search on the low cost region while expanding a tree. It generates also new waypoints by probabilistically branching off of existing waypoints and weighting each waypoint based on, not only the number of close waypoints, but also on the estimated total cost of going through that waypoint on a path to the goal. The second phase of the algorithm refines the existing path according to a cost function by following the gradient of the path. This technique does not enforce elastic properties of the path and it is able to take more robust precautions in repelling from obstacles.

In Kim et al. (2005) authors try to address the problem of testing complex reactive control systems and validating the effectiveness of multi-agent controllers. More

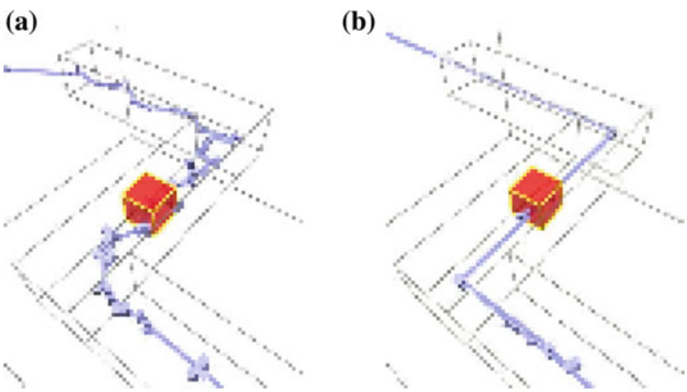

Fig. 9 Differences when growing a tree: **a** basic expansion and **b** using obstacle information (Rodriguez et al. 2006)

specifically, they consider the application of the Rapidly-exploring Random Tree (RRT) algorithm to the testing and validation problem and they propose three modifications in order to improve its results. First, they introduce a new distance function which encodes local information about the system's dynamic constraints with a first order approximation (i.e. dynamics-based selection of proximal node). Secondly, because the reachable state space is generally a small fraction of the total state space, they developed a weighting factor to penalize nodes which are repeatedly selected but fail to extend (i.e. history-based selection of proximal node), and finally, they proposed a scheme for adaptively modifying the sampling probability distribution between the traditional uniform distribution and heavily biased toward the specification set based on tree growth (i.e. adaptively biased sample generation).

2.14 Obstacle Sensitive Cost Function for Navigating Car-Like Robots

In Ziegler and Werling (2008) authors propose a new method for navigating a car-like vehicle within an unstructured environment, using a path planning technique which is posed as a graph search problem. Actually they define an implicit graph that is expanded on the fly by an A* search algorithm. In this algorithm, the search graph is set up in a way that implies derivation of a feed forward term for a downstream closed loop controller. More specifically, authors have added a feed forward term, which makes the controller react more quickly and accurate, since reaction of the vehicle to steering input is modelled separately from controller offset introduced by noise. Then, an informed search algorithm is used, which is guided by a heuristic cost function that accounts for both kinematic constraints of the vehicle and the topology of the vehicle's free space. This cost function gives expected cost-to-go for each node of the search graph, so if the cost function underestimates the actual distance to the goal, A* is guaranteed to find the least-cost path. If the error of the cost function is big, A* quickly degenerates to an exponential time algorithm. The configuration space obstacles are then computed from an obstacle map acquired from a high definition laser range scanner and search is restricted to the collision free subset of the configuration space. Authors suggest that this algorithm is suitable for solving all of the following problems: precise parking maneuvers, narrow turns and long distance navigation.

3 Methods Based on Fuzzy Logic

In Baturone and Gersnoviez (2007) a novel work is proposed, which combines some neuro-fuzzy techniques with geometric analysis in order to get a good trade-off between purely heuristics and purely physical approaches. Specifically, the

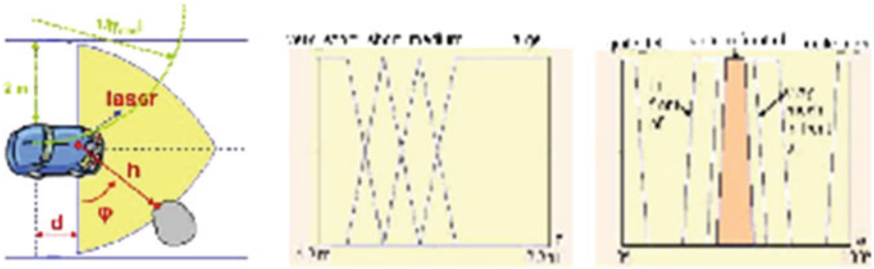

Fig. 10 Area of unavoidable and very close obstacles (Baturone and Gersnoviez 2007)

controller follows a reactive technique and generates trajectories of near-minimal lengths when no obstacles are detected, while in presence of obstacles, it generates minimum deviations from them. The fine structure of the controller constituent modules has been obtained by applying supervised learning with numerical data, satisfying the non-holonomic constraints. In particular, it consists of several neuro-fuzzy modules, whose fuzzy classifiers have two inputs and one output. For close objects, authors used two approaches to define a fuzzy classifier, a monolithic system and a hierarchical one, while for obstacle avoidance, the controller evaluates an angular aperture and determines the sign of the curvature to avoid the obstacle. Without obstacles, the robot can navigate towards the goal by the shortest path, using again a monolithic or a hierarchical fuzzy system. Experimental results indicate that if no obstacles are detected, the robot goes to the goal by a near-minimal length path, while if obstacles are close, they are avoided by minimum deviations from the quasi-optimal path (Fig. 10).

3.1 Distributed Active-Vision Network-Space System

In Hwang and Shih (2009) a navigation scheme that contains complex pattern, non-uniform illumination, and strong reflection based on a distributed active-vision network-space system (DAVNSS), is presented. This system is subject to three fuzzy variable-structure decentralized controls (FVSDCs), which includes trajectory tracking and obstacle avoidance. Two distributed wireless charge-coupled-device (CCD) cameras individually driven by two stepping motors are constructed to capture the dynamic pose of the car-like wheeled robot (CLWR) and the obstacle. The control system includes quad processors with multiple sampling rates, while a personal computer (PC) is employed to receive the image of the CLWR or obstacle by a wireless transmitter and then to plan three reference commands for the CLWR and the cameras. Next, a six-step image-processing routine and the calibration between the world coordinate and the image plane coordinate using multilayer perceptrons (MLPs) are established, while in the final step the radial distortion of ACCD is reduced for better localization and tracking (Fig. 11).

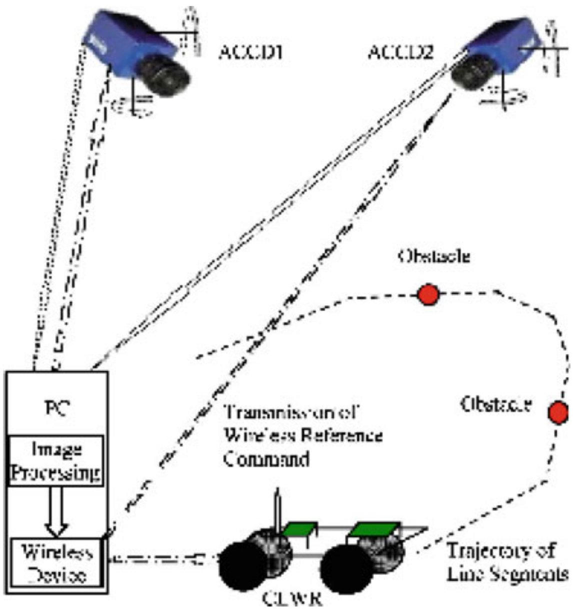

Fig. 11 Diagram of the overall control system (Hwang and Shih 2009)

In El-Khatib and Hamilton (2006), an approach which consists of two layers for real-time navigation of a non-holonomic car-like robot in a dynamic environment is described. More specifically, the first layer is a Sugeno-type fuzzy motion planner of four inputs and one output, which is used to give a clear direction to the robot controller. The second stage is a modified proportional navigation-based fuzzy controller, which is based on the proportional navigation guidance law and it is able to optimize the robot's behavior in real time. This means that it is able to avoid stationary and moving obstacles (such as pedestrians and vehicles) in its local environment obeying specific kinematics constraints. The proposed system consists of two subsystems, a fuzzy motion planner (FMP) and a modified proportional navigation (PN) based fuzzy controller, inspired by human routing. The fuzzy motion planner uses Takagi-Sugeno fuzzy inference for the rule evaluation, resulting in the output of a control function for the system depending on the values of the inputs. The proportional navigation method is a guidance law, which seeks to null the line of sight changing rate (LOS) by making the controlled system (robot) turn rate be directly proportional to the rate of turn of sight (i.e. it seeks to nullify the angular velocity of the line of sight (LOS) angle). For the intelligent behavior of the fuzzy controller, the direction control behavior of the robot in response to an obstacle was also incorporated. As was indicated, the main advantage of this system is the simplicity of its design which makes it suitable for hardware implementation and extensibility as it does not rely on any specific robotic platform, while it is able to use also linguistic representation, which allows the capture of human experiences and intuitive reasoning (Fig. 12).

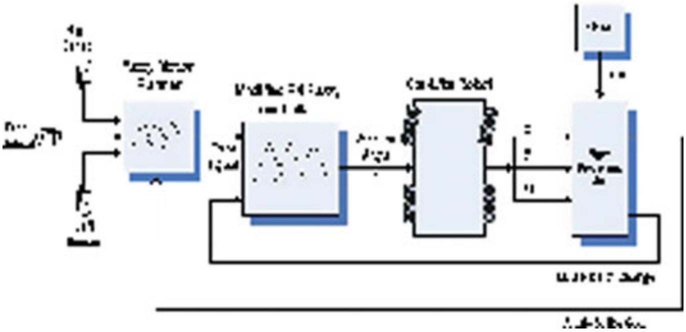

Fig. 12 The complete system of fuzzy motion planner and behavioral fuzzy controller (El-Khatib and Hamilton 2006)

3.2 *Internet-Based Smart Space Navigation Using Fuzzy-Neural Adaptive Control*

In Hwang and Chang (2008) a different approach is presented for a navigation system, which includes a path tracking and an obstacle avoidance apparatus for a car-like wheeled robot (CLWR) within an Internet-based smart-space (IBSS) using fuzzy-neural adaptive control (FNAC). This method relies on two distributed charge-coupled device (CCD) cameras, which capture both the dynamic pose of the CLWR and the obstacle. Based on the control authority of these two CCD cameras, a suitable reference command has been planed, which contains the desired steering angle and angular velocity for the FNAC built into the client computer. The FNAC method that the authors presented in Hwang and Chang (2008) contains also a neural network consisting of a radial basis function (RBFNN) to learn the time-related uncertainties due to the fuzzy-model error, which stem from wireless network delays and CLW slippage (Fig. 13).

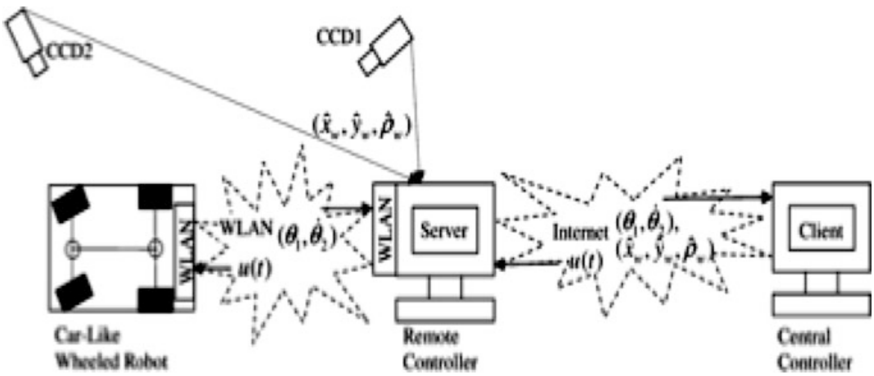

Fig. 13 Block diagram of the overall system (Hwang and Chang 2008)

4 Sensor-Based Methods

4.1 Dynamic Window Approach (DWA)

Another approach presented in Rebai et al. (2007) investigates the use of Dynamic Window Approach (DWA) to solve the high speed autonomous navigation problem for mobile robots in unknown and unstructured environments. Since the DWA algorithm considers periodically a short time interval when computing the next motion command and based on the fact that the obstacles in the closer environment of the robot impose restriction on the translational and rotational velocities, authors define a Dynamic Window (DW) in order to limit the accelerations executable by the motors. In addition, in order to reduce the time of the motion command selection, they use the sensory data from the environment directly in the obstacle avoidance process without the grid cells building in the velocity space, while in order to determine the Distance To Collision (DTC), they have adopted an analytic solution for polygonal robot. Regarding the experimental results of the algorithm, the obstacle avoidance tests using the extended DWA for different environments (simple and cluttered) at high speeds indicated a good performance and efficiency.

4.2 Generalized Voronoi Graph (GVG) Theory

Another approach which relies on a sensor based algorithm for car-like robot based on GVG theory is presented in Quan et al. (2011). For generating the completed GVG, the car-like robot goes through each edge and vertex of GVG in two tangent directions. In addition, the authors proposed backward motion for direction changes at boundary points, also with favorable results (no collision) in unknown environments.

In Quan et al. (2011) a new algorithm is presented that enables a car-like robot to explore an unknown planar workspace, based on Generalized Voronoi Graph (GVG) theory. More specifically, since GVG is a set of points in the plane equidistant to two obstacles, the robot of the proposed system has three degrees of freedom and hence the authors defined a rod-GVG edge as the set of the points equidistant to three obstacles.

In Gall et al. (2010), an intelligent scaled car-like mobile robot that possesses the capability of autonomous driving in an extra-road environment and fully autonomous parking on standard parking lots is presented. In particular, authors describe a low weight and low cost complex mobile robot that is able to navigate across a previously unknown terrain combining some mechanical, sensorial, computing and communication modules (rather than implementing a new sophisticated algorithm). An algorithm associated with the autopilot of the system was also implemented, in order to make the mobile robot completely autonomous; many of these functions

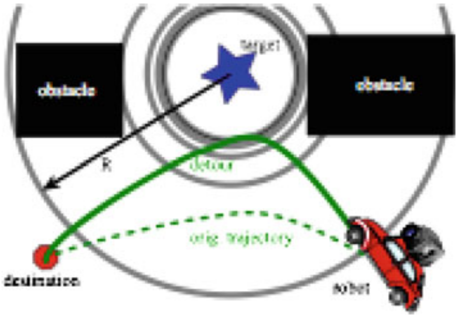

Fig. 14 A schematic representation of a car-like robot making a detour from a path towards its primary destination to opportunistically gather additional information about a secondary target (indicated by a *blue star*) once the presence of the latter has been detected at distance R (Grady et al. 2012)

are written in MATLAB, and therefore available for analysis and modification using open-source modules (xPC toolbox).

The robot described in Grady et al. (2012) is equipped with a sensor that can alert it if an anomaly appears within some range while the robot is moving. In that case, the robot tries to deviate from its computed path and gather more information about the target without incurring considerable delays in fulfilling its primary mission, which is to move to its final destination. The originality of this approach is to take a "semi-corrective" action, i.e. deviating while attempting to further define the problem, akin to a car stepping out of its lane when flashing lights appear ahead—not changing lanes yet, just gaining a view of the obstacle. This model relies on a sampling-based planner called SYCLOP, which works by automatically defining a decomposition of the workspace, creating an adjacency and abstraction graph, and searching that graph for a high-level guide. Then, a low-level planning layer computes the actual dynamically feasible paths and informs the upper layer for how to assign informative weights to the edges of the abstraction graph (Fig. 14).

4.3 Navigation in Dynamic Environments Using Trajectory Deformation

A different approach is presented in Delsart and Fraichard (2008), where authors present a new trajectory deformation scheme in order to improve path deformation. During the course of execution, the still-to- be-executed part of the motion is continuously deformed in response to sensor information (internal and external) acquired on-line, thus accounting for the incompleteness and inaccuracies of the a priori world model (Fig. 15).

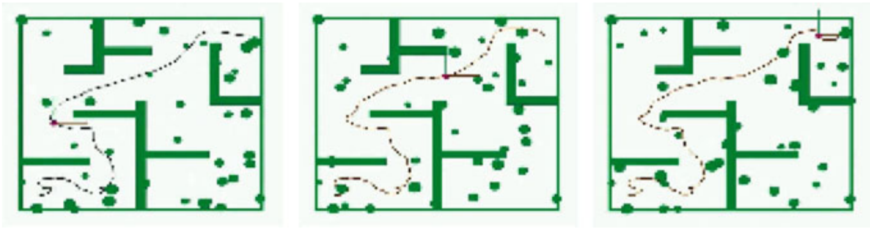

Fig. 15 Double integrator system: the snapshots depict the path at different time instant (Delsart and Fraichard 2008)

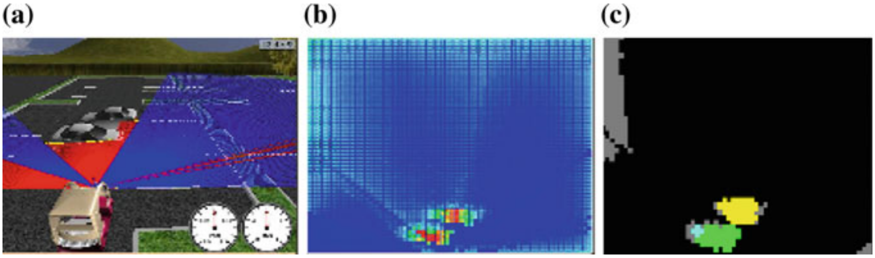

Fig. 16 Simulated detection of two cars crossing each others. **a** Simulated environment: the robot equipped with a laser range finder detects a car moving from left to right and a second car moving from right to left. **b** Dynamic occupancy grid: *red* is high, *blue* is low probability of occupation. The space behind the cars has low probability of occupation. **c** Clustering: different colours characterise objects and occluded or free space (Fulgernzi et al. 2007)

4.4 Probabilistic Velocity Obstacle (PVO)

In Fulgernzi et al. (2007), the Probabilistic Velocity Obstacle (PVO) provides a probabilistic estimation of the occupied free space around the robot and of the velocity with which the objects are moving. The observations of the mobile robot update a 4D probabilistic occupancy grid (incl. space and velocity), and the probability of collision in time is estimated for each reachable velocity of the robot. The proposed system shows that is able to take directly into account limited range and occlusions, uncertain estimations of velocity and position of the obstacles, allowing the robot to navigate safely toward the goal (Fig. 16).

5 SLAM-Based Methods

5.1 On-line Path Following

In Rezaei et al. (2004) the authors address the problem of on-line path following for a car working in unstructured outdoor environments. More specifically, the partially

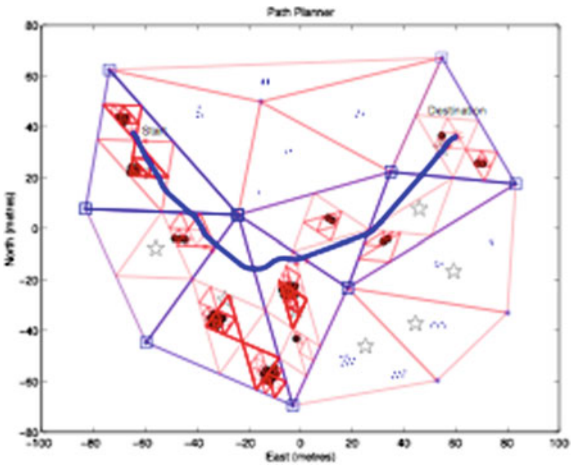

Fig. 17 An example of a generated path (Pepy and Lambert 2006)

known map of the environment is updated and expanded in real time by a Simultaneous Localization and Mapping (SLAM) algorithm. This information is used to implement global path planning based on a new method which constructs a cost graph using the D* search algorithm. In this stage, uncertainty is incorporated in the cost function, and since the continuity of the path is crucial for car type robots, the algorithm chooses only the continuous-curvature local paths. Finally, an improved feedback linearization control algorithm is used to guide the car along this computed reference path (Fig. 17).

5.2 *The CyCab: A Car-Like Robot Navigating Autonomously and Safely Among Pedestrians*

In Pradalier et al. (2005) authors present a bi-steerable car, which allow steering by turning either the back or the front wheels. Using this car, they address the integration of the four essential autonomy abilities (i.e. simultaneous localisation and environment modelling, motion planning and motion execution) into a single application. Then they build a kind of simplified occupancy grid on the environment and they apply the motion planner adopted for the CyCab, expressed as a Bayesian inference problem. Bayesian methods are also used for trajectory tracking (Fig. 18).

In Pradalier et al. (2004) a new kind of public transportation system is presented, which relies on a particular double-steering kinematic structure (as described above). The authors in this work address the integration of these four essential autonomy abilities into one application, applying a reactive execution of planned motion. In addition, they address the fusion of controls, issued from the control law and the obstacle avoidance module, using probabilistic techniques. The planner first

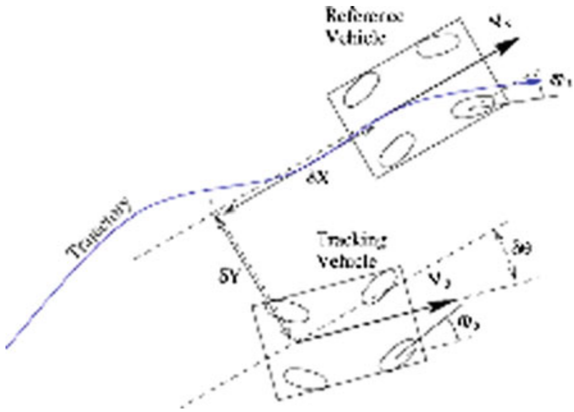

Fig. 18 Variables involved in trajectory tracking behaviour, using Bayesian inference (Pradalier et al. 2005)

builds a collision-free path without taking into account the non-holonomic constraints of the system. Then, this path is approximated by a sequence of collision-free feasible sub-paths computed by a suitable steering method and then is smoothed properly.

5.3 V-Slam

In Lategahn et al. (2011) authors propose a dense stereo V-SLAM algorithm that estimates a dense 3D map representation which is more accurate than raw stereo measurements. The proposed system is composed of two main parts. First a sparse V- SLAM system based on an EKF is calculated, which takes the resulting pose estimates in order to compute a locally dense representation from dense stereo correspondences. The state vector of the EKF contains all landmark positions, the current camera pose and a subset of past camera poses. To tackle the computational complexity problem inherent to EKF SLAM, authors utilize a sub mapping method called conditionally independent sub maps. After incorporating new observations and updating the EKF state vector a new camera pose is obtained. This allows the dense part to be continuously updated.

5.4 SLAM-Based Turning Strategy in Restricted Environments

In Cheein et al. (2010) a strategy to turn a car-like mobile robot in a restricted environment using a Simultaneous Localization and Map Building (SLAM) algorithm is presented. More specifically, in the first step of the proposed method, the environment's information and the vehicle's pose (position and orientation) estimation is provided to the vehicle by a SLAM algorithm, which is implemented on an Extended Kalman Filter (EKF), extracting the lines and corners (convex and concave) from the environment. In the next phase, a turning algorithm, which is based on a semi-circle trajectory, following with direction switching, plans from the vehicle's initial pose the first semi-circle trajectory with respect to the environment until it reaches a neighborhood of the closest geometric map feature provided by the SLAM system state. Then, a next semi-circle trajectory is planned in the opposite direction to the previous trajectory. The proposed algorithm continues until the vehicles reaches the desired orientation, while a kinematic trajectory controller drives the vehicle through the generated paths.

5.5 L-Slam

Authors in Petridis and Zikos (2010) present a new SLAM method, called L-SLAM. It is a low dimension version of the FastSLAM family algorithms, which reduces the dimensionality of the particle filter that FastSLAM algorithms use, while achieving better accuracy with less or the same number of particles. The key idea they used is to sample only the robot's orientation on each particle, in contrast to the FastSLAM algorithms that sample the orientation along with the position of the robot.

6 Conclusions and Future Work

As we have seen, for any mobile device, the ability to navigate in its environment is important. Avoiding dangerous situations such as collisions and unsafe conditions (temperature, radiation, exposure to weather, etc.) comes first, but if the robot has a purpose that relates to specific places in the robot environment, it must find those places. As a result, mobile robots capable of moving in a dynamical and uncertain environment is an important issue in real-world applications. The problem that how to find an optimal real-time collision-free path with a limited sensing range in the presence of dynamically moving objects is arising naturally. The optimal solution should take motion constraints into consideration (including boundary conditions and kinematic constraint), explicitly handle dynamically moving objects, and be analytical.

Based on the previous mentioned methods, it is evident that this technology is well promising for the future. While the human-machine interface is not yet at a transparent level, the degree of autonomy available after a machine has been program is now approaching that once considered purely science fiction. Things that could be done in the future, related to the previous algorithms, are for example to optimize the current techniques using more state-of-the-art methods, testing the navigation algorithms to have a measure of its performance in more complex and realistic scenarios, or even considering for instance that the knowledge about the future behaviour of a robot is less reliable in the distant future, so it could be interesting to monotonically decrease the influence of the obstacles with respect to time.

6.1 Future Directions in Autonomous Robot Navigation and Obstacle Perception

One of the ways in which autonomous robot perception could be improved, especially for safe navigation of urban environments, is by object perception. For the majority of this review, all obstacles have been treated as essentially equal (i.e. rough patches to be avoided). However, as the reader may have suspected at some point, not all obstacles are equal. In fact, some obstacles present quite opposite problems to the optimization routine. A patch of ground for example is a "rough patch" that, although preferably avoided, could in theory be traversed if the cost-to-go function (i.e. "roughness-to-go") were to deem a trajectory through that path to be necessary. However, in other situations, especially in urban environments, "traversing" an obstacles is absolutely not an option. One obvious case of an obstacle that may not under any circumstances be traversed is a pedestrian. Pedestrians must be avoided at all costs, including the cost of potentially never reaching the end destination or (from a programmatic point of view) never being able to calculate possible trajectories leading to the destination. This would be the case in a hypothetical situation where a never-ending stream of pedestrians is crossing a street.

One of the recent entries into the US Department of Defense—sponsored annual competitions for autonomous robot navigation was the Stanford car dubbed "Junior" (2013). As reported (Levinson et al. 2011), "Junior" was able to very accurately tell the difference between people, cars, animals, signs, and roads. This was largely thanks to a novel laser and sensor calibration scheme that involved a great deal of machine learning in the original situation in the navigation environment. This was an example of a case for which existing information about the visual environment was used to estimate or interpolate parameters for that environment at later times points, by looking primarily at the aspects of the environment that change. This calibration scheme is illustrated in Fig. 19.

Another means by which machine vision is becoming more sophisticated in the perception of objects, is in a sense by moving in the opposite direction to how Stanford's "Junior" progressed from earlier autonomous vehicles. Whereas "Junior" was able to use more specific, fine-grained features of the environment, (Maddern and Vidas 2012) is able to detect whether it is nighttime or daytime outside, and base interpretation of features, obstacles density paths, and corresponding trajectories on this information. For example, an accurate assessment of the position of the sun (as well as other light sources, during the night) allows for the detection of

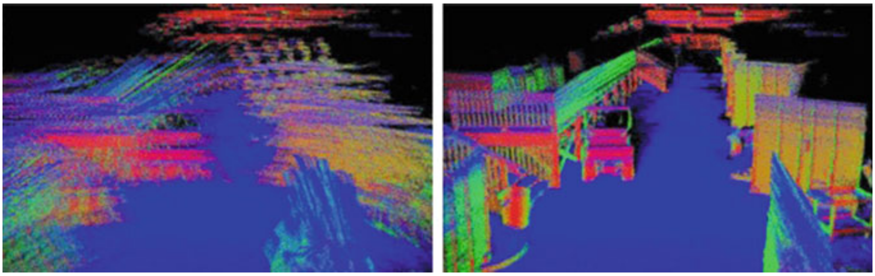

Fig. 19 The evolution of obstacle perception—obstacle discrimination. The Stanford "Junior" autonomous vehicle (Levinson et al. 2011) is able to turn a density "cloud" of obstacle perception (*left*) into a refined, crisp image (*right*) with sufficient details to make out the identity of different obstacles, provided that an initial sensor calibration is performed. Once this calibration has been performed at the beginning of the vehicle's trip or trajectory, it does not need to be repeated until the vehicle is transported by carrier and placed in a new environment. The refined image allows for certain obstacles (such as pedestrians in a crowded urban environment) to be identified and avoided at all costs, in favor of traversing less important obstacles if need be (such as curbs, stairs, etc.)

shadows with greater accuracy. Evidently, shadows can be traversed provided there are not hidden obstacles. To this author's knowledge, no methods that delve into predicting obstacles hidden in shadows have been developed to date.

As one can see, the richness of the visual information is increasingly being taken advantage of by autonomous vehicles, thanks in large part to these vehicles' increasingly powerful artificial intelligences. As a result, the rate of acquisition of this data is becoming increasingly "thirsty", and autonomous vehicles are traveling faster and faster. "Junior" Niclass et al. (2013), for example, can travel up to 35 miles per hours in a crowded urban environment (slowing or stopping where necessary, of course, to avoid pedestrians and other key obstacles). However, the faster that autonomous vehicles go, the more error is introduced to their sensors thanks simply to some basic principles of optics. For example, bending of light occurs even at moderate (highway) speed. As a results, in Niclass et al. (2013) the authors present a sensor capable of recording single-photon time of flight information based on correlation with other photons.

6.2 Future Directions in Applications of Autonomously-Navigating Robots

With all the above literature review and discussed devoted to sensors, techniques, problem types, and optimization algorithms for autonomous vehicle perception of obstacles and navigation by optimization of trajectories around obstacles, little has been said thus far about the actual applications of autonomously-navigating robots. What is the interest in, and what are therefore some possible applications of, these increasingly intelligent and self-aware road and off-road travelers? Below is a discussion describing a variety of different existing and emerging applications of such robots.

An obvious, although far from universally accepted or even much considered, application for autonomous vehicles is for the transportation of people. Some autonomous vehicles already transport people. However, there is potential for autonomous personal and public vehicles to largely replace the manually-operated equivalents of today and yesterday. Because there is not yet an "internet of things" (IoT), a term used below that refers to the potential future in which all objects are connected to the internet via tiny wireless sensors, directions and features of roads and in particular traffic conditions (for example detours due to construction) are often not updated into mobile road navigation apps on many drivers' cell phones or GPS units. Therefore, at least for the foreseeable future (until there is a veritable IoT or at least higher-integrity, more reliable set of traffic/road conditions comprehensively and instantaneously updated in real time—no 15-min delay allowable) autonomous vehicles would have to be able to read road signs just as any human driver would. A recent study Mathias et al. (2013) provides a method for achieving rapid, in-transit machine-vision sign reading. However, the authors performed their training as well as validation under conditions of fair lighting, unlike what may often be the case even with headlights illuminated.

One possible applications autonomously-navigating robots is as traffic-monitoring "drone" vehicles. These vehicles would patrol highways, and collect information about traffic density, and other environmental factors such as temperature, humidity, and surface conditions (e.g. precipitation accumulation). These drone vehicles would be networked to an information hub, either a higher-level computer or a human operator and traffic surveyor. Relaying information to a central information hub would enable high-resolution, real-time information about traffic to be distributed to passengers. This would be accomplished in multiple possible ways, for example by allowing the information to be accessible to mobile phone apps. One recently-proposed means of distributing the information gathered by robotic drone vehicles is by using cloud computing Whaiduzzaman et al. (2014). In this model, cloud computing would also be used to allow communication (and thereby formation of consensus data interpretation and analysis) between different robots. Cloud computing is fast, easily accessible, and cheap.

In order to address issues of security related to autonomous vehicles, especially those with a multitude of sensors containing possible sensitive information (but generally without the size, complexity, or infrastructure to effectively protect against virus or rogue cyber intrusions), some sophisticated theoretical as well as practical design steps have already been taken. As explained in Ho et al. (2012), while denial of service (DoS) style attacks (or sophisticated cyber-attacks originating from multiple points simultaneously) on static networks remains a problem, the technological development of wireless sensors has in many cases led to the adoption of a mobile, robotic platform. In parallel, the possibility of DoS by a mobile, malignant node arises. This article is the first to describe this problem explicitly, describing the unique advantages to DoS agents that mobility brings, and to propose a solution for overcoming these new advantages.

The article Ho et al. (2012) mentions several means by which malicious, mobile nodes could disrupt wireless sensor networks (WSNs) that would be impossible without mobility. A mobile node, if equipped with robotic arms, could move up to a node in the WSN, pick it up, and move it. This disruption in position would throw flags in the WSN security routine, and may even automatically cut out the node from the WSN, leaving it easy pretty for the malicious node. Malignant, mobile nodes could also move to as many different positions as possible, searching for weaknesses (i.e. spots where their positions would be more likely to be accepted as characteristic of a "safe" node), jamming communications, and moving nodes. In addition, the diversification of attack paths would make traceback impossible, without a prior assumption that the nodes were both mobile and hostile.

Although purely theoretical, the article does propose several strategies for combatting the threat of a mobile, malignant node or swarm of nodes. The article focuses on the case wherein the WSN is static, and only the malignant nodes are mobile. A straightforward means of detecting a mobile malignant node would be to keep a list of neighbors, leveraging the fact that the WSN is static. However, this would place severe constraints on the topology of the WSN, as a pre-defined set of neighbors would have to be supplied to the base node as a unique key, for each node. The authors suggest using an adjustable threshold maximum time limit between signals from a neighboring node, with the assumption that, beyond this threshold, the neighbor would be considered as potentially mobile (and therefore malignant). The more nodes flag the same outside node as malignant according to this criterion, the more likely is the base node to pass a judgment of "malignant".

References

Balakirsky, S., Dimitrov, D.: Single-query, bi-directional, lazy roadmap planner applied to car-like robots. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 5015–5020 (2010)

Baturone, I., Gersnoviez, A.: A simple neuro-fuzzy controller for car-like robot navigation avoiding obstacles. In: IEEE FUZZ, pp. 1–6 (2007)

Burns, B., Brock, O.: Single-query motion planning with utility-guided random trees. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 3307–3312 (2007)

Cheein, F.A.A., Carelli, R., De la Cruz, C., Bastos-Filho, T.F.: SLAM-based turning strategy in restricted environments for car-like mobile robots. In: International Conference on Industrial Technology (ICIT), pp. 602–607 (2010)

Delsart, V., Fraichard, T.: Navigating dynamic environments using trajectory deformation. In: IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 226–233 (2008)

Devaurs, D., Simeon, T., Cortes, J.: Parallelizing RRT on large-scale distributed-memory architectures. IEEE Trans. Robot. Autom. **29**, 571–579 (2013)

El-Khatib, M.M., Hamilton, D.J.: A layered fuzzy controller for nonholonomic car-like robot motion planning. In: IEEE International Conference on Mechatronics, pp. 194–198 (2006)

Espinoza, J.L., Sánchez, A., Osorio, M.A.: Exploring unknown environments with RRT-based strategies. In: International Conference on Artificial Intelligence, pp. 1150–1159 (2006)

Fulgernzi, C., Spalanzani, A., Laugier, C.: Dynamic obstacle avoidance in uncertain environment combining PVOs and occupancy grid. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 1610–1616 (2007)
Fulgenzi, C., Tay, C., Spalanzani, A., Laugier, C.: Probabilistic navigation in dynamic environment using rapidly-exploring random trees and Gaussian processes. In: Intelligent Robots and Systems (IROS), pp. 1056–1062 (2008)
Gall, R., Troster, F., Mogan, G.: On the development of an experimental car-like mobile robot. In: International Conference on Optimization of Electrical and Electronic Equipment (OPTIM), pp. 734–739 (2010)
Garrido, S., Blanco, D., Moreno, L., Martın, F.: Improving RRT motion trajectories using VFM. In: IEEE International Conference on Mechatronics (ICM), pp. 1–6 (2009)
Garrido, S., Moreno, L., Blanco, D., Martın, F.: Smooth path planning for non-holonomic robots using fast marching. In: Int. J. Robot. Autom. 154–176 (2011)
Grady, D.K., Moll, M., Hegde, C., Sankaranarayanan, A.C., Baraniuk, R.G., Kavraki, L.E.: Multi-objective sensor-based replanning for a car-like robot. In: International Symposium on Safety, Security, and Rescue Robotics (SSRR), pp. 1–6 (2012)
Ho, J.W., Wright, M., Das, S.K.: Distributed detection of mobile malicious node attacks in wireless sensor networks. Ad Hoc Netw. **10**(3), 512–523 (2012)
Hwang, C.-L., Chang, L.-J.: Internet-based smart-space navigation of a car-like wheeled robot using fuzzy-neural adaptive control. IEEE Trans. Fuzzy Syst. **16**(5), 1271–1284 (2008)
Hwang, C.-L., Shih, C.Y.: A distributed active-vision network-space approach for the navigation of a car-like wheeled robot. IEEE Trans. Industr. Electron. **56**(3), 846–855 (2009)
Jaillet, L., Yershova, A., LaValle, S.M., Simeon, T.: Adaptive tuning of the sampling domain for dynamic-domain RRTs. In: IEEE International Conference on Intelligent Robots and Systems, pp. 2851–2856 (2005)
Jaillet, L., Cortés, J., Siméon, T.: Transition-based RRT for path planning in continuous cost spaces. In: IEEE International Conference on Robots and Systems, pp. 2646–2652 (2012)
Karaman, S., Frazzoli, E.: Sampling-based optimal motion planning for non-holonomic dynamical systems. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 5041–5047 (2013)
Kim, J., Esposito, J.M., Kumar, V.: An RRT-based algorithm for testing and validating multi-robot controllers. In: Conference on Robotics Science and Systems, pp. 249–256 (2005)
Kuwata, Y., Fiore, G.A., Teo, J., Frazzoli, E., How, J.P.: Motion planning for urban driving using RRT. In: International Conference on Intelligent Robots and Systems (IROS), pp. 1681–1686 (2008)
Lategahn, H., Geiger, A., Kitt, B.: Visual SLAM for autonomous ground vehicles. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 1732–1737 (2011)
Levinson, J., Askeland, J., Becker, J., Dolson, J., Held, D., Kammel, S., Thrun, S., et al.: Towards fully autonomous driving: systems and algorithms. In: IEEE Intelligent Vehicles Symposium (IV), pp. 163–168 (2011)
Maddern, W., Vidas, S.: Towards robust night and day place recognition using visible and thermal imaging. RSS 2012: beyond laser and vision: alternative sensing techniques for robotic perception (2012)
Mathias, M., Timofte, R., Benenson, R., Van Gool, L.: Traffic sign recognition—how far are we from the solution? In: IEEE International Joint Conference on Neural Networks (IJCNN), August 2013, pp. 1–8
Niclass, C., Soga, M., Matsubara, H., Kato, S., Kagami, M.: A 100-m range 10-frame/s 340 96-pixel time-of-flight depth sensor in 0.18-CMOS. IEEE J. Solid-State Circ. **48**(2), 559–572 (2013)
Pepy, R., Lambert, A.: Safe path planning in an uncertain-configuration space using RRT. In: Intelligent Robots and Systems (IROS), pp. 5376–5381 (2006)
Pepy, R., Lambert, A., Mounier, H.: Path planning using a dynamic vehicle model. In: ICTTA, pp. 781-786 (2006)

Petridis, V., Zikos, N.: L-SLAM: reduced dimensionality FastSLAM algorithms. In: IEEE International Joint Conference on Neural Networks (IJCNN), pp. 1–7 (2010)
Petti, S., Fraichard, T.: Safe navigation of a car-like robot in a dynamic environment. In: European Conference on Mobile Robots (2005)
Phillips, J.M., C.S. Draper Laboratories: Guided expansive spaces trees: a search strategy for motion- and cost-constrained state spaces. In: International Conference on Robotics and Automation (ICRA), vol. 4, pp. 3968–3973 (2004)
Pradalier, C., Hermosillo, J., Koike, C., Braillon, C., Bessihre, P., Laugier, C.: An autonomous car-like robot navigating safely among pedestrians. In: IEEE International Conference on Robotics and Automation (ICRA), vol. 2, pp. 1945–1950 (2004)
Pradalier, C., Hermosillo, J., Koike, C., Braillon, C., Bessière, P., Laugier, C.: The CyCab: a car-like robot navigating autonomously and safely among pedestrians. In: Robotics and Autonomous Systems, pp. 51–67 (2005)
Quan, Y., Lee, J.Y., Changsoo, H.: Sensor-based navigation algorithm for car-like robot to generate completed GVG. In: International Conference on Control, Automation and Systems (ICCAS), pp. 1442–1447 (2011)
Rebai, K., Azouaoui, O., Benmami, M., Larabi, A.: Car-like robot navigation at high speed. In: Robotics and Biomimetics, pp. 2053–2057 (2007)
Rezaei, S., Guivant, J.E., Nebot, E.M.: Car-like robot path following in large unstructured environments. In: IEEE IROS, pp. 2468–2473 (2004)
Rodriguez, S., Tang, X., Lien, J.-M., Amato, N.M.: An obstacle- based rapidly-exploring random tree. In: International Conference on Robotics and Automation, pp. 895–900 (2006)
Tahirovic, A., Magnani, G.: A roughness-based RRT for mobile robot navigation planning. In: IFAC World Congress, vol. 18, pp. 5944–5949 (2001)
Whaiduzzaman, M., Sookhak, M., Gani, A., Buyya, R.: A survey on vehicular cloud computing. J. Netw. Comput. Appl. **40**, 325–344 (2014)
Yershova, A., Jaillet, L., Simeon, T., LaValle, S.M.: Dynamic-domain RRTs: efficient exploration by controlling the sampling domain. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 3856–3861 (2005)
Ziegler, J., Werling, M.: Navigating car-like robots in unstructured environments using an obstacle sensitive cost function. In: IEEE Intelligent Vehicles Symposium, pp. 787–791 (2008)