Vision-Guided Automatic Parking for Smart Car

Jin Xu, Guang Chen and Ming Xie

School of Mechanical & Production Engineering
Nanyang Technological University
Singapore, 639798
Tel +65 7906942 – Fax +65 7935921
mmxie@ntu.edu.sg

Abstract

This paper presents our work on automatic parking of a smart car that relies on vision to estimate free parking slots. All problems involved in implementing an automatic parking behavior are discussed. Solutions are given together with experimental results obtained from real data.

1 Introduction

How to develop an intelligent vehicle or robot is an interesting topic in robotics. There are many ways to define vehicle intelligence. One possible definition is the ability to handle information from sensors, plan and execute vehicular motion control in a similar way done by human driver. Parking is certainly one among the vehicular motion controls that present some challenge.

For automated control of vehicular motions, there are two typical scenarios: a) driving on road and b) automatic parking. For the first scenario, a lot of solutions has been developed for the case of lane following or lane changing. The existing systems work fine on highway with input data collected from Vision [1]. The second scenario is more difficult to implement because the space is less structured and more skills are needed to plan the motion.

It is not an easy task to develop automatic parking skill for a car. First of all, the car must be able to sense the environment, to find a free parking space and to detect obstacles in the carpark. Secondly, it should have the ability to plan the motion itself from the input of perception system. Lastly, it should be able to follow the planned motion sequences and to make sure that a goal position is reached, regardless of noise and errors.

1.1 Previous Works

There are many researchers working on AGV (automated guided vehicle) with focus on different aspects. Basically, all the existing solutions use vision or active sensors such as ultrasonic or laser range finder to build a local map for navigation, then followed by planning and execution of a designed motion series.

The group in INRIA has built up a complete sensor-based control architecture for an autonomous vehicle [2]. They make use of ultrasonic sensors. Their planning method is a model-based approach that decomposes the motion into a number of "*parallel parking*" series. At each step of motion, the orientation of the car-like vehicle is identical at the beginning and at the end. By use of the ultrasonic sensors, the size of the free parking space is determined.

Wolfgang A. has proposed a skill-based visual parking control using neural and fuzzy network [3]. Their approach is based on the acquisition and transfer of an experienced human driver's skills to an automatic parking controller. The controller processes visual information from a video sensor and generates the corresponding steering commands by use of neural networks. Two neural control architectures were considered: *Direct Neural Control Architecture* and *Fuzzy-hybrid Control Architecture*.

1.2 Proposed Solution

Here, a solution based on the use of vision system to detect free parking slots is proposed. Vision is chosen as the main perception system because of rich information in images and the immunity to interference and noise. By image processing and vision computing, we can detect an empty parking slot (where to go) from the images. Detection of obstacles is also achievable with the same vision system.

Once the detection of free parking slot has been completed, the next step is to plan a proper path that will bring the vehicle into the empty slot. Our solution is based on the analysis of parking strategy and local path planning. A new solution of local path planning based on quintic polynomials and concept of symmetric postures is discussed.

Based on the feasible path planned, we generate all the necessary executable commands to be sent to the vehicle. These commands will then be executed consequently to drive Smart Car into the parking slot.

2. Parking Slot Detection Using Vision

2.1 Problem statement

To park a car into an empty parking lot properly, the following data are required: the position and orientation of the empty parking slot and the position of the obstacles around it. By using one or two digital camera(s), we can acquire the images around the car at real time. The car should be able to detect and localize of the empty parking slot as well as possible obstacles above the ground plane.

In order to extract information from images, the image pixels need to be grouped to form clusters. Each cluster of pixels corresponds to an object of interest. (e.g, the land-mark of an empty slot or an obstacle). Subsequently, one may undertake some measurement to obtain the geometric information from the grouped pixels.

The geometric information can then be transformed from image space to a task space. In the case of detecting free parking slot, the position and orientation of the parking slot need to be computed in a coordinate system attached on the ground plane. As for obstacle detection, the 3D geometric of an obstacle in a 3D coordinate system can be computed from images taken at two different views.

2.2 Parking Slot Detection Using Color

In order to find out where to go, we need to first detect the empty slot in image and then calculate their coordinates with respect to a ground plane.

Since the colours of parking slot's marking are quite uniform and different from the backgrounds, it is valid to suggest the use of colour to identify the pixels of these markings from images. Our colour-segmentation method based on RCE neural network was first developed by Guo et al. [4]. Color (formulated in HSI color space) of an object of interest is learned by the training process of a RCE neural network. After the adaptive training, the segmentation can be done automatically. The structure of RCE neural network is shown in Figure 1. Figure 2 demonstrates the results of the segmented markings of parking slots from a real image.

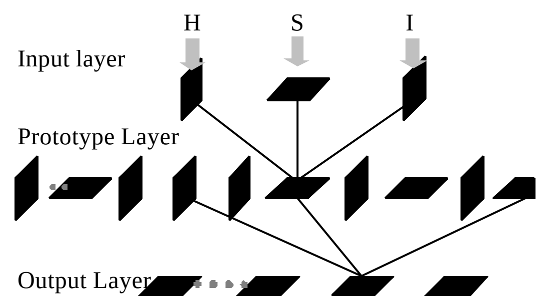

Figure 1: Structure of RCE neural network

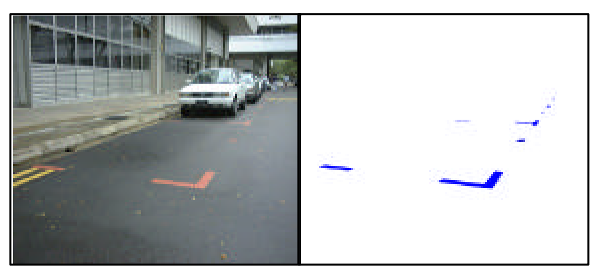

Figure 2: Segmented markings from real image

When the pixels corresponding to the markings have been segmented out, we use the outline of these pixels as the geometric feature. By scanning through the processed image column by column from bottom to top, we get a set of isolated points of the contour. Then by estimating the two lines of this contour (using least square method), we can get equations for the two lines (shown in Figure 3) in the image plane.

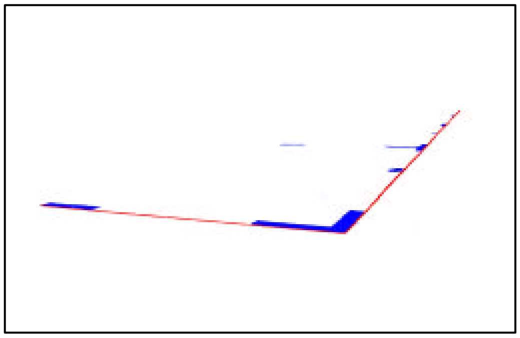

Figure 3: Two lines estimated as the contour

When these information have been extracted and estimated from the images, we need to transform the geometric parameters into the 2D coordinate system attached to the ground plane. This transformation depends on a constant mapping matrix that can be estimated by calibration. This mapping matrix relates image co-ordinates to 2D co-ordinates in the following way:

$$\begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \qquad (1)$$

In Eq.1, (X, Y) are the coordinates of a point on the ground plane in a coordinate system attached to that plane while (x, y) are the coordinates of the corresponding pixel in the image plane. The transformation matrix M_{3x3} can be obtained by calibration (using at least 4 calibration points.

From this equation, once we find the "empty slot" in the image plane, we can use an inverse projective transformation to get the coordinates of this slot in the ground plane. The inverse transformation of Eq.1 can be formulated as Eq.2 below.

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = M_{3x3}^{-1} \bullet \begin{bmatrix} sx \\ sy \\ s \end{bmatrix} \qquad (2)$$

Once the two lines of the contour of the parking slot are transformed into the 2D coordinate system attached to the ground plane, we know the position and orientation of the empty parking slot.

2.3 Obstacle Detection

Detection and localization of obstacles is another task for an in-vehicle vision system. We propose the idea of using 2D vision principle [11]. Since there is a projective transformation between a plane and its images captured by any camera, there exists a projective transformation between the pixels of the ground plane in left and right images captured by a stereo vision system. This transformation can be determined by at least 4 matches (more matches are used to get better result by least-square method). Then we can apply this transformation to the two images to identify pixels corresponding to the projection of points on a ground plane. The points on an obstacle above the ground plane will not verify this condition. Figure 4 shows an example of result.

Figure 4: Obstacle detection by 2D calibration

3. Motion planning

3.1 Problem statement

When a local map has been built based on the information of the empty slot and obstacles that are obtained from the in-vehicle vision system, the next step is to plan the motion that brings the car from the current posture (position plus orientation) to the goal posture. This motion-planning problem for car-like vehicle has been investigated intensively and is still an open problem [5]. There exist two different categories of approaches.

The first category is skill-based. The control commands are generated in real time according to the current state. Fuzzy logic and neural network are used to transfer skills of human beings to an intelligent vehicle [3]. The second category aims at first planning the whole motion series in advance, and then sending these commands to the controller consequently. The proposed solution here falls into the second category.

The plan-and-control methods include two standard steps [6]. The first step is *path planning* and the second step is called *trajectory planning*. The definition of these two terms is as follows:

Path planning: Design the route on which the vehicle can move from the initial configuration to the final configuration while avoiding stationary and moving obstacles.

Trajectory planning: Design a time sequence of configuration (orientation together with position) for the vehicle and corresponding reference inputs to the motion control system so that the vehicle will move along the specified path.

Here, we propose a new concept of *parking strategy* to guide the path planning, which is especially useful for parallel parking. The idea is to break the whole motion down into several steps (forward and backward) instead of planning the whole procedure as one step. In our work, we have developed and tested a two-step algorithm and a four-step algorithm. The result of the two-step algorithm is shown as below

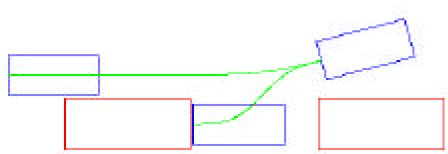

Figure 5: two-step parking algorithm

3.2 Local Path planning and control commands generation

The path planning for a car-like vehicle is different from path planning for mobile robot. A collision-free path for mobile robot may not be executable for a car-like vehicle. Due to the characteristic of car, a good path for a car-like vehicle should be smooth, continuous in first and second derivatives and has an upper bounded curvature all along the path. Due to these requirements, many conventional methods for path planning are not applicable for a car-like vehicle. These methods include the road map method and cell division method.

To plan a smoothed path for a car-like vehicle, many types of curves have been proposed, among which are circular arcs and straight lines [7], clothoid curves [8], cubic polynomial curves and quintic (fifth-order) polynomial curves [9]. Here, we adopt quintic polynomial curves.

Many researchers have noticed the property of symmetric postures [12] [13]. Quintic polynomials connecting two symmetric postures have an interesting advantage: the maximum curvature is always at the half distance. This property allows partial adjustment of the maximum curvatures along the path. In fact, the well-studied parallel parking can be treated as one special example of symmetric postures. D. Lyon used this property to build a one-step algorithm for parallel parking [9]. But his solution didn't consider the existence of obstacles and is impractical without considering backward motion. In our solution, by introducing an intermediate posture that is symmetric to both initial and final posture, we can connect each pair of postures (not symmetric for most cases) by two segments of symmetric quintic polynomials.

It is proved that between two arbitrary postures, there is always a set of postures that are symmetric to both of them [12]. Thus we can connect any two arbitrary postures with two segments of symmetric quintic polynomials. These two segments should satisfy:

1. The curvature all long the curve is within an upper limit.
2. The first-order and second-order derivatives of consecutive curves are continuous at the intermediate posture.

When the path is planned, the next step is to plan the trajectory of the car along that path. Since a car-like vehicle must move in its longitude direction, the only possible orientation on the path is the tangential direction at that point. Thus the orientation of the car along the planned path is fully determined by:

$$\theta = \arctan(f'(x)) \quad (4)$$

Once the path and trajectory are determined, we can generate motion commands (velocity and steering angle) that make the car to follow that designed path. Motion equations of a car-like vehicle is formulated in Eq.5 (these are well known as non-holonomic constraints).

$$\begin{cases} \dfrac{dx}{dt} = v_D \cos\theta \\ \dfrac{dy}{dt} = v_D \sin\theta \\ \dfrac{d\theta}{dt} = \dfrac{v_D}{L}\tan\alpha \end{cases} \quad (5)$$

It can be seen that a change of the time scale is equivalent to the scaling of the driving velocity V. In the practical sense, this property means that one can drive the car along the same path with an arbitrary varying driving velocity (under the dynamic constraints). Therefore, the velocity planning and steering angle planning can be solved separately. In velocity planning, we consider the dynamic constraints and in steering angle planning, without loss of generality, we can assume that V=const>0.

For the planning of the velocity, we aim to consume as little time as possible, thus we can choose the upper limit under the dynamic constraints from the dynamic model of the car. Since the speed is not a main concern in a parking task, we just choose V=2m in the experiment.

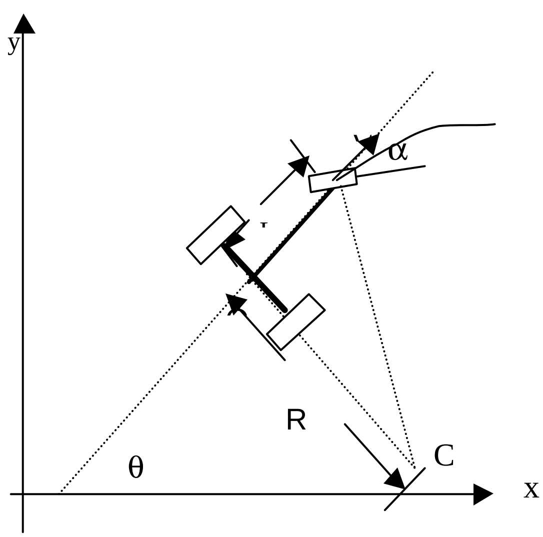

Figure 5: Kinematic model of Smart Car

For steering angle planning, what we have is a path in x-y plane as y=f(x) (where f is a quintic polynomial). From Figure 5, we can easily get the steering angle α as:

$$\alpha = \arctan(L/\rho) \quad (6)$$

where ρ is the instant turning radius of the vehicle (inverse of the curvature at the point on that curve).

3.3 Results

We have implemented our algorithm in a simulation program and a real vehicle. In the experiment, a feasible path is generated first and then control commands are generated by the method described above. A simulated car with non-holonomic constraints and the maximum steering angle constraint is driven by these commands to follow the planned path (see Fig. 6).

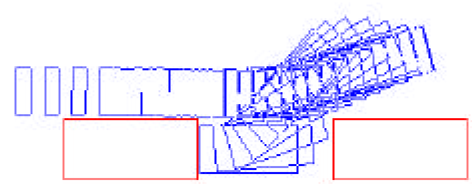

Figure 6: Simulation results of a two-step parking

The control commands generated by this method are also sent to the controller of our vehicle-Smart Car (CyCab electric car from Robosoft) in a real carpark (see Fig. 7). A program combing vision, path planning and motion generation discussed above has been developed. The commands were sent to the on-board controller and the car tracks the path exactly as we planned (error less than 10 cms).

Figure 7: Smart Car at a campus carpark

4. Conclusion

In this paper, a complete solution for autonomous parking of an intelligent vehicle under the guidance of in-vehicle vision is proposed. The algorithms for vision and motion planning are discussed in detail respectively with results obtained from real data and simulation. Based on the results obtained so far, we conclude:

1. Vision provides a good guidance for autonomous parking and color information is a good help in parking slot detection and obstacle detection.
2. The proposed parking strategy for motion planning is effective for automatic parking task.
3. The combination of quintic polynomials and symmetric postures leads to good results in local path planning for automatic parking behaviors.

References

[1] T.M. Nguyen, J.W. Sinko and R.C. Galijan. Using differential carrier phase GPS to control automated vehicles. *Proceedings of the 40th Midwest Symposium on Circuits and Systems,* Volume 1, pages 493 –496, 1998.

[2] C. Laugier and F. Thierry. Sensor-based control architecture for a car-like vehicle. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems,* Volume 1, pages 216 –222, 1998.

[3] W.A Daxwanger and G.K. Schmidt. Skill-based parking control using Neural and Fuzzy Networks. *IEEE International Conference on Systems, Man and Cybernetics, 1995*. Volume 2, pages 1659 –1664, 1995.

[4] D. Guo and M. Xie. A neural network architecture for adaptive color image segmentation. *Fifth International Conference on Control, Automation, Robotics and Vision (ICARCV' 98)*, Volume 1, pages 146-150, 1998.

[5] Jean-Claude Lamtobe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[6] J-P. Laumond. Finding collision-free smooth trajectories for a non-holonomic mobile robot. In *Proc. of the Int. Joint Conf. on Artificial Intelligence*, pages 1120-1123, Milan (Italy), Aug. 1987.

[7] J. Vandorpe. *Navigation techniques for the mobile robot LiAS*. PhD thesis, Katholieke Universiteit Leuven-Faculteit Teogepaste Wetenschappen, 1997.

[8] S. Fleury et al. Primitives for smoothing mobile robot trajectories. *IEEE transactions on robotics and automation,* Volume 11, No. 3, June 1995.

[9] D. Lyon. Ad-hoc and derived curves for parallel parking a car. *Symposium on Advances in Intelligent Systems*, Proceedings Volume 1388-04, SPIE, Bellingham, WA.

[10] Y. Kanayama, Y. Kimura, F. Miyazaki and Y. Noguchi. A stable tracking control method for a non-holonomic mobile robot. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and systems*, Osaka (JP), November 1991.

[11] Swee Meng Wong and Ming Xie. Vision Functions for the Guidance of Smart Vehicle. *Proceedings of the 9th International Conference on Advanced Robotics ('99 ICAR),* Tokyo, Japan, October 25-27 1999.

[12] Yutaka Kanayama and Bruce I. Hartman. Smooth Local Path Planning for Autonomous Vehicles. *IEEE International Conference on Robotics and Automation*, Volume 3, pages 1265-1270, 14-19 May 1989.

[13] A. Scheuer and Th. Fraichard. Planning Continuous-Curvature Paths for Car-Like Robots. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Volume 3, Pages 1304 –1311, 1996.