2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics
Suntec Convention and Exhibition Center
Singapore, July 14-17, 2009

Vision-Based Fuzzy Navigation of Mobile Robots in Grassland Environments

A. R. Norouzzadeh Ravari, H. D. Taghirad, *Member, IEEE,* A. H. Tamjidi

***Abstract*— In this paper a vision-based algorithm for mobile robot navigation in unknown outdoor environments is proposed. It is based on a simple phenomenon, that when the robot moves forward, projected images of the near obstacles grow in captured frames faster than that of the far objects. The proposed algorithm takes advantage of this property and extracts features from each grabbed frame of the camera and tracks the vertical position of the features and their speed along the Y axis of the image plane over multiple frames as the robot moves. The relative height of the features and their distance from the robot in 3D is inferred based on this data and they are fed into a fuzzy reasoning system which marks the features from safe to unsafe according to their suitability for navigation. Then a second fuzzy system summarizes these scores in different image regions and directs the robot toward the area containing more features marked as safe. Simulation and implementation results confirm the efficacy of the proposed simple algorithm for mobile robot navigation in outdoor environments.**

I. Introduction

Nowadays, robots are used in various environments for different applications, from industrial fields, manufacturing to autonomous exploration of remote planets. Among these, mobile robots have achieved great consideration from scientific, industrial and military communities. A mobile robot needs a navigation algorithm to find a suitable and safe path to reach its goal without colliding with obstacles.

Different types of sensors are used for mobile robots navigation, but in the last three decades, visual navigation has become a source of countless research contributions as vision strategies provide a perception of environment in a single shot [1]. Navigation strategies of mobile robots are grouped into indoor [2, 3] and outdoor [4, 5] categories. Navigation in outdoor environments is subsequently divided into navigation in structured [6, 7] and unstructured environments [5, 8] .

Some researchers have sought the remedy for the problem of complexity of vision based navigation in using fuzzy logic approaches. It turned out to be applicable due to its simplicity of implementation, satisfactory results in spite of parameter variation and uncertainty and easy interpretation of fuzzy rules by a human expert [9, 10] (the most related methods to the proposed algorithm).

Howard *et al.* [9] apply a horizon line extraction algorithm to identify peripheral boundary of the ground plane plus a region growing based on edge detection to determine the size and concentration of rocks to a pair of stereo images. In addition to terrain roughness, the terrain slope is extracted from stereo pair of images and used as inputs of a fuzzy inference system that determines the relative traversability of viewable area.

Shieh *et al.* [10] used a vision-based two-stage fuzzy navigation method for navigation of an intelligent shopping service robot. The robot is equipped with nine sonar sensors to measure the distance to obstacles and a CCD camera to detect obstacles using edge detection and image processing algorithms. One fuzzy logic system is used for steering angle control based on the output of visual sensory module which extracts vanishing lines of hallway and uses edge detection to find obstacles, another fuzzy logic system is used for speed control according to the distance to nearest obstacle. The simulation results show that robot navigates through a safe path to reach the goal.

Recently, navigation techniques based on feature tracking are vastly used as tracking algorithms have become more robust and are suitable for navigation as feature extraction often imposes no specific assumption on the type of the environment [1].

As an example homography based navigation algorithms [11, 12], which use feature tracking for ground plane detection, assume that the robot is moving on a planar ground or a small neighborhood of the robot is planar. The homography based navigation methods can no longer be used for navigation of a mobile robot in a rough terrain of an outdoor environment.

There are some research about navigation in rough terrains [9, 13] but few authors used just a single camera for navigation of a mobile robot in outdoor environments [14, 15].

In this paper, we propose a new algorithm for visual navigation of mobile robots in unstructured outdoor environments based on feature tracking and fuzzy logic. The structure of the paper is as follows: in section II the design of the proposed method is explained. In section III simulation and implementation results are shown and benefits of our method are presented. Finally, in section IV some concluding remarks are mentioned.

A. R. Norouzzadeh Ravari is with the Department of research and planning of K.R.E.C. phone: 0983412222104; fax: 0983412228014; e-mail: a.norouzzadeh@gmail.com).

H. D. Taghirad, is with K. N. Toosi University of Technology, Tehran, Iran. He is currently an associate professor of Control Department of the Faculty of Electrical Engineering (e-mail: taghirad@kntu.ac.ir).

A. H. Tamjidi is an M.S. student of K. N. Toosi University of Technology, Tehran, Iran. (e-mail: ahtamjidi@kntu.ac.ir)

II. Theory And Design

Suppose a wheeled mobile robot needs to autonomously navigate in an unstructured outdoor environment using a non-calibrated regular camera as its input sensor.

For safe navigation of a mobile robot in an unknown outdoor environment, we need to do the following tasks:

- Ground plane detection
- Obstacle identification
- Traversable area specification
- Navigation

We consider that the robot is navigating in a rough terrain with static obstacles, perceives the required information from a single camera and makes navigation decisions in real-time.

While the robot traverses in the real world, the relative positions of the obstacles vary in the image plane and consequently the 2D projections of these points, in our case extracted features, move in some direction depending on the heading of the robot and the location of obstacle in real world. It can be seen in Fig. 1, that camera movement toward an object, increases the scale of the object in the image plane and causes apparent motion of features in the image plane. When the robot moves toward an obstacle, projected features from the obstacle move upward in the image plane if they are located above the camera's X-Z plane (refer to fig. 3 for definition of camera's coordinates). On the contrary, if the features are located below the camera's X-Z plane, they move downward as the robot draws near the obstacle. Taking into account this property and based on the movement of features in the image plane, the robot can decide whether the corresponding 3D point is an obstacle or not, and by this way it can avoid moving toward the obstacles in the environment. However, in this process, closer obstacles should have priority over the distant ones, and therefore, the relative distance of the obstacles should also be taken into account when choosing proper direction for robot movement. This can be achieved if the speed of the feature's movement in the image plane is also calculated. Features corresponding to the close obstacles move faster in the image plane than those corresponding to the far ones.

Also there are situations that the algorithm makes wrong decisions based on apparent motion of features. One example of such situations is when one side of the wheeled mobile robot passes over a bump, the camera tilts and features suddenly rotate in opposite direction in relation to each other. We discuss about this in section III in more detail.

Using these two properties of the apparent motion of features and a fuzzy inference system, features can be compared in relation to each other and represented by linguistic fuzzy sets, which is the base of our vision-based fuzzy navigation algorithm.

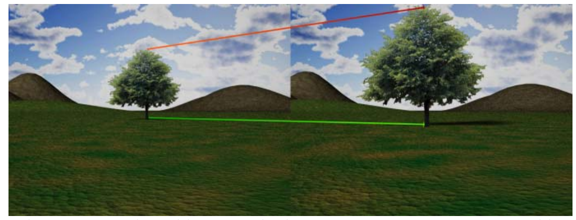

Fig. 1. Increasing object's scale due to camera movement toward it.

The diagram of the proposed method is shown in Fig. 2. This algorithm can be divided into the following stages:

- Initializing new features
- Tracking features over multiple frames
- Computing the apparent motion of features
- Fuzzy reasoning and scoring features
- Navigation toward features with higher score

In the following sections, the algorithm is explained in detail but for now let's take a look at the overall progress. At first some features are selected, while the robot is moving along Z-axis, the features are tracked and their displacement along Y-axis with the vertical element of their speed are computed. Using the first fuzzy inference system, we give a score to each feature according to its apparent motion in the image plane. The image plane is divided into 3 equal columns and the mean of the features' scores is computed for each column. The difference between mean score of the left column with the middle one is used as first and the difference between mean score of the right column with the middle one is used as the second input of another fuzzy inference system whose output is the speed difference between left and right wheels of the mobile robot.

The next section describes the pinhole camera model briefly, as it is used for computation of projected features motion on the image plane. Then, the feature selection and tracking algorithm is explained and finally the design of two fuzzy inference systems is introduced.

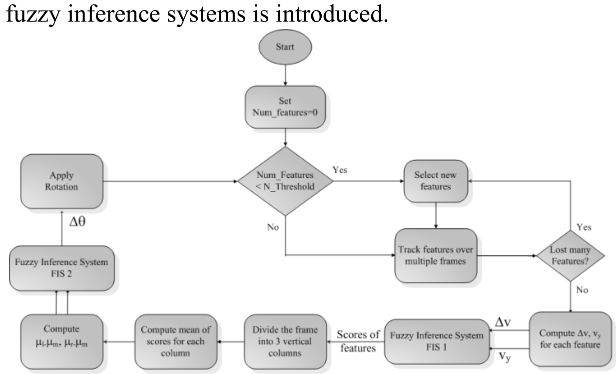

Fig. 2. Vision-Based fuzzy navigation diagram.

A. Camera Model and Motion Equations

Using a simple pinhole camera model, as shown in fig. 1, consider $P = [X, Y, Z]^T$ a 3D point in the camera reference frame, Z the optical axis and f the focal length, is projected to $p = [u, v]^T$ a 2D point in the image plane through the following equation:

$$p = f\frac{P}{Z} \quad (1)$$

Also the velocity vector $V = [v_x, v_y]^T$ can be expressed as follows:

$$
\begin{aligned}
v_x &= \frac{T_z X - T_x f}{Z} - \omega_y f + \omega_z Y + \frac{\omega_x XY}{f} - \frac{\omega_x X^2}{f} \\
v_y &= \frac{T_z X - T_y f}{z} - \omega_x f + \omega_z Y + \frac{\omega_y XY}{f} - \frac{\omega_x Y^2}{f}
\end{aligned}
\tag{2}
$$

Where: $T = [T_x, T_y, T_z]^T$ and $\omega = [\omega_x, \omega_y, \omega_z]^T$ are translational and rotational velocities of the camera respectively.

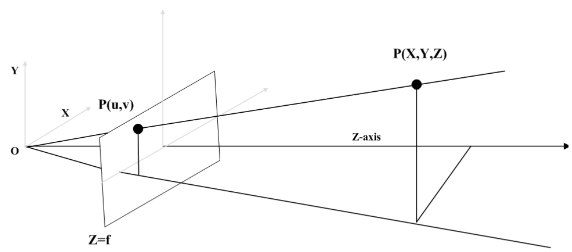

Fig. 3. Camera pinhole model.

From (2) we find that the depth information can't be recovered directly and in apparent motion of features in the image plane, we have no information about depth of projected points. But we can estimate depth information from relative distance and movement speed of projected points.

B. Feature Selection and Tracking

In order to extract qualitative characteristics of the 3D environment and safely navigate among obstacles, we select features and track them as long as possible. The motion parameters of the features, such as their position and the speed of movement along the vertical axis are used to find traversable area.

The Shi and Tomasi [16] algorithm is used for feature selection which is implemented in OpenCV library [17]. The number of features, tracked in consecutive frames, is `Num_Fearures` which are spread all over the input frame with uniform distance of $dist_{pxl}$ pixels in our case.

The selected features are tracked using Lukas and Kanade [18] method. As long as a feature is in field of view, its initial and current positions are stored and used for obstacle detection and navigation. As the robot moves, some features get out of the camera's field of view. The feature selection and initialization is done, when the number of visible features is less than a threshold `N_THRESHOLD` which is typically considered to be half the maximum number of features.

C. Vision-Based Fuzzy Logic Control

Special properties of fuzzy logic system such as its robustness to noisy inputs, has made it a widely used method in different control applications and hence in mobile robot navigation [9, 10]. Therefore, we have used fuzzy logic for interpretation of features motion and consequently for making navigation decisions. As discussed before, while the robot moves, selected features can be compared in relation to each other and represented by linguistic fuzzy set {flat, sloped, hill} according to the estimated height and {near, distant, far} according to the estimated distant. The inputs of the first fuzzy system (FIS 1) are the projected displacement of features into the image plane along Y-axis represented by Δv and the vertical element of their speed shown by v_y where the output is a score for each feature representing the traversability index.

Before the fuzzification process, the displacement of features in the image plane along Y-axis is normalized in the range of [0,1]. The output of the fuzzy inference system is a score in the range of [0,1] using the max criterion for de-fuzzification.

The membership functions of two input variables (Δv and v_y) and output of first fuzzy logic system (FIS 1) can be seen in Fig. 4.

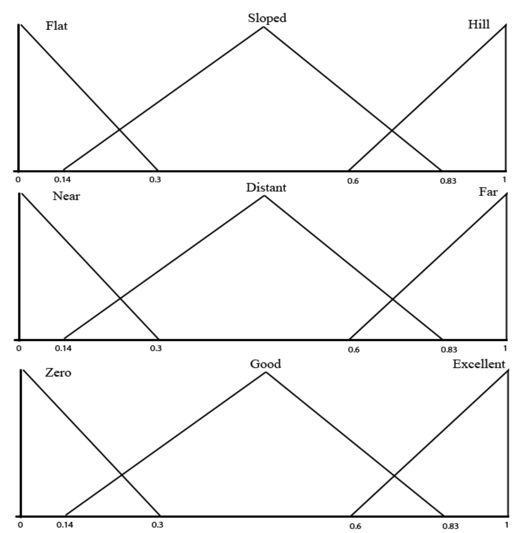

Fig. 4. Membership functions of Δv , v_y and score.

The rule base of this fuzzy logic system which gives a score to each feature according to its relative traversability is shown in Table I.

The rule base is designed such that features on the ground get maximum score while giving fewer score to the features moving toward the up side of the image. Features on an obstacle get fewer score as their height increases in relation to features on the ground when the robot gets close to them.

In order to avoid near obstacles and traverse toward free spaces or far obstacles, we used another fuzzy logic system deciding to go straight, rotate left or right. The first fuzzy logic system just assigns a traversability index to each feature, but we need to make navigation decisions, in our case, going forward, rotating to the left or right. So we used another fuzzy logic system to find a safe direction for navigation. Therefore the image plane is divided into 3 equal columns virtually. The mean score of features for each column is computed and the difference of mean of scores with mean score of the middle column is used as inputs of the second fuzzy inference system. If we represent the mean score of features in left, middle and right columns as μ_l, μ_m and μ_r respectively, the inputs of the second fuzzy inference system (FIS 2) are μ_r and $\mu_l - \mu_m$ and $\mu_r - \mu_m$.

TABLE I
FEATURE SCORING FUZZY RULE BASE

Δv	v_y	Score
Flat	Near	Excellent
Flat	Distant	Excellent
Flat	Far	Excellent
Hill	Near	Zero
Hill	Distant	Zero
Slope	Near	Zero
Slope	Distant	Good
Slope	Distant	Good

The membership functions of this fuzzy logic system, as shown in Fig. 5, are manually configured such that desired rotation in relation to score difference is generated and the robot navigates toward features with more scores.

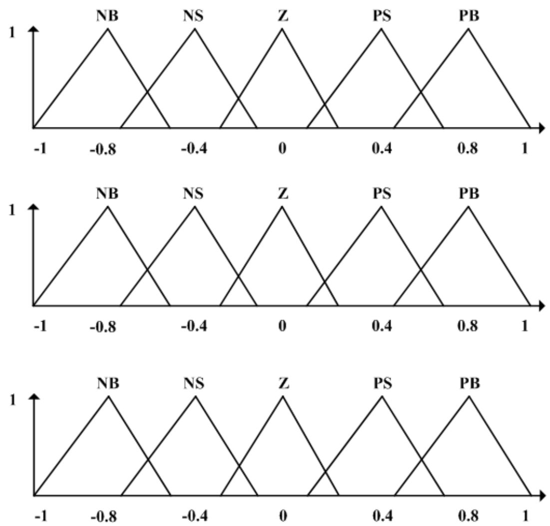

Fig. 5. Membership functions of FIS 2. From top to bottom, membership function of $\mu_l - \mu_m$, $\mu_r - \mu_m$ and rotation.

The range of input variables is between [-1, 1] and using the max criterion as defuzzification method, the output rotation is a signed value in the range of `[-MAX_ROTATION, MAX_ROTATION]` where negative values indicates rotation to the left and positive values correspond to the rotation to the right. The `MAX_ROTATION` parameter should be relatively small in relation to the camera's field of view to keep track of features and navigation through a smooth path.

Finally, the output of the second fuzzy inference system is applied to the mobile robot to navigate toward the direction with more traversability index which is done by changing the speed of the wheels. Table II shows the rule base of the second fuzzy inference system (FIS 2). As this fuzzy logic system determines the navigation direction, the rules are designed such that robot traverses on a relatively smooth path.

TABLE II
NAVIGATION FUZZY RULE BASE

$\mu_l - \mu_m$	$\mu_r - \mu_m$	Rotation
PB	Any	NB
Any	PB	PB
NS	PS	PS
PS	NS	NS
PS	PS	Z
NS	NS	Z
Z	Z	Z

The next section shows the results of our simulations in a virtual outdoor environment and the implementation of the algorithm in an outdoor environment.

III. SIMULATION AND IMPLEMENTATION RESULTS

A. Simulation Results

Simulation results show that the proposed vision-based fuzzy navigation can direct a mobile robot toward safe area in a virtual outdoor environment. The simulation has been done using a C++ program and OpenCV [17] in a 3D grassland simulated environment.

The feature selection strategy is such that we have uniformly placed features over the input frame in order to retrieve required information from different parts of the input frame. Also there's a tradeoff between maximum number of features, the resolution of the input video and real-time implementation of this algorithm.

The simulation program selects 250 features placed minimum 30 pixels away from each other for a 640×480 input video and tracks them for 10 frames in initialization phase. Then the score of each feature is calculated using the first fuzzy inference system. After computing the mean score of features in each of left, middle and right columns, the second fuzzy logic system determines the more safe direction and the robot rotates toward it. The algorithm keeps track of features as much as possible and makes navigation decisions in every two frames. If the number of features is less than 125, the initialization process is executed and new features will be used. In our simulations, the `MAX_ROTATION` parameter is equal to one degree, which is small in comparison to the camera's field of view, which is 45 degrees.

Fig. 6 shows the structure of the simulated environment from top view and screen shots of the output of the simulation program. The tracked features can be seen in Fig. 6 as green, orange and red arrows. The arrow itself shows the direction of each feature's movement and the color shows its traversability index, where green color stands for safe to navigate, orange indicates sloped area and red color corresponds to unsafe areas.

As Fig. 6 shows, the robot avoids near obstacles and moves toward far ones. Also, it can be seen that features are tracked and marked with traversability index.

In our simulations, the robot had a zigzag movement in a sparse environment, because of using a non-calibrated camera. In a sparse environment, when an obstacle goes out of camera's field of view, the robot thinks that there's no obstacle there as it can't be seen anymore. But when the robot rotates toward that area, the obstacle comes in the view again so the robot tries to get away from it. This problem can be solved by using a calibrated camera or adding memory to the system so it remembers the features went out of field of view.

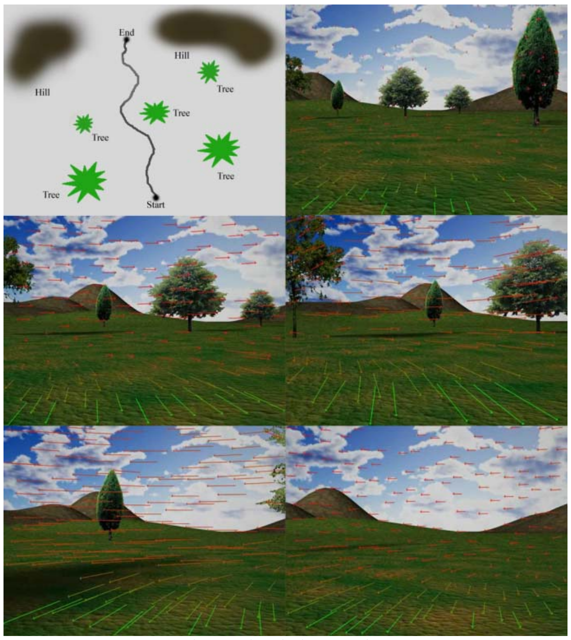

Fig. 6. Simulation results.

B. Implementation Results

After testing the vision based fuzzy navigation algorithm in a simulated environment, the navigation program is used to direct a mobile robot toward safe area in a cluttered outdoor environment.

Fig. 7 shows the KNTU Pars mobile robot which is used for our implementation test. The implementation results show that the algorithm can guide a mobile robot in an outdoor environment as can be seen in fig. 8. The algorithm works the same as the simulation environment but with some limitations.

Due to the tracking limitations, the robot should move small enough such that the features move slowly and can be tracked which in our case was 5 cm/sec. Therefore the distance between the obstacles should be large enough so the robot can traverse without collision. In our test, the required distance between obstacles was 3 meters which depends on the robot's dimensions, its speed and number of initialization frames. Also the environment can include traversable cliffs as the algorithm cannot detect the cliffs and avoid them. If one side of the wheeled robot passes over a bump making camera to tilt, causes wrong decisions from the navigation algorithm. This problem can be solved using a gyroscope.


Fig. 7. KNTU Pars mobile robot


Fig. 8. Mobile robot's view

Finally, the proposed method is compared to three similar visual navigation techniques where only a single camera is used for navigation of a mobile robot.

In [15], the authors proposed a method for outdoor navigation of a mobile robot even on a non planar ground. The obstacle detection algorithm in [15], needs some feature points with distance of greater than 10 meters in relation to the nearest obstacle for computation of time to contact. For this purpose, it is assumed that features in two rectangles on the upper left and right side of the camera's frame are placed at distant in the real world and only a region of interest in the middle of the image plane is used for detection of obstacles. Therefore, this technique can't detect large obstacles when the two upper corners of the image contain features from that obstacle such the environments used in our simulations. Also in their experiment, 2500 features are tracked and used for detection of obstacles, which makes the algorithm unsuitable for real-time applications. Besides, no specific technique is presented for detection of sloped area of the ground.

The homography based visual navigation method presented in [11] assumes that the robot is moving on a planar ground or at least a small neighborhood of the robot is planar. The obstacle detection method used in [11] is limited to ground plane detection and considering objects outside of the ground plane as obstacles. In homography based visual navigation algorithms, there are situations that some points on the ground, share the same homography with points on obstacles [11]. Although, this ground plane detection method uses color classification and an algorithm for homography refinement, but the problem of similar homography values for some points on the ground and obstacles still exists. Also the complexity of this algorithm makes it inappropriate for real-time applications.

The algorithm presented in [12] also uses homography for dominant plane ground detection and needs a downward-looking camera. Besides the problem mentioned for [11], this algorithm reacts to obstacles only when the robot is too close to it as the camera is looking downward and its field of view is limited to a small neighborhood of the robot.

From simulation and implementation results, presented here, it can be seen that the proposed vision-based fuzzy navigation algorithm can guide a mobile robot in outdoor environments without the common problem of conventional algorithms like tracking of large number of features, working just on planar grounds or complex calculations for obstacle detection.

IV. Conclusions

In this paper we have presented a vision-based fuzzy navigation algorithm and showed in simulation that this algorithm can guide a mobile robot with a non-calibrated camera, in an unknown outdoor environment toward more traversable area without colliding to obstacles. The proposed method selects a number of features with a fixed distance between them and tracks them as long as the number of visible features in greater than a threshold.

The main idea of this method is based this property of apparent motion of features that, when a camera moves in a static environment along its optical axis, the points on top of the camera's X-Z plane in the real world, move toward upside of the image plane and vice versa. Also the points on the near obstacles move faster than points on the far ones. Based on these two properties, we have constructed a vision-based fuzzy navigation algorithm and showed in simulation that using the proposed algorithm a mobile robot can traverse in an unknown outdoor environment without colliding to obstacles.

This algorithm is suitable for real-time visual navigation of mobile robots in unstructured outdoor environments as it's not based on special assumptions such as planarity of the ground, specific type of obstacles, and requirement of using large number of features or heavy computations.

References

[1] Francisco Bonin-Font, Alberto Ortiz, and G. Oliver, "Visual Navigation for Mobile Robots: A Survey," Intell Robot Syst, pp. 263-296, 2008.

[2] Jones S.D., Andresen C., and C. J.L., "Appearance based processes for visual navigation.," in IEEE Int'l Conf. on Intelligent Robots and Systems (IROS), 1997, pp. 551–557.

[3] Matsumoto Y., Inaba M., and I. H., "Visual navigation using view sequenced route representation," in Proc. of IEEE Int'l Conf. on Robotics and Automation (ICRA). vol. 1, 1996, pp. 83-88.

[4] Jochem T.M., Pomerleau D.A., and T. C.E., "Vision based neural network road and intersection detection and traversal," in Proc. of IEEE Int'l Conf. on Intelligent Robots and Systems (IROS). vol. 3, 1995, pp. 344-349.

[5] Krotkov E. and H. M., "Mapping and positioning for a prototype lunar rover," in IEEE Int'l Conf. on Robotics and Automation (ICRA), 1995, pp. 2913–2919.

[6] Graefe V, "Vision for intelligent road vehicles," in IEEE Symposium of Intelligent Vehicles, 1993, pp. 135–140.

[7] Jochem T.M., Pomerleau D.A., and T. C.E., "Vision based neural network road and intersection detection and traversal," in IEEE Int'l Conf. on Intelligent Robots and Systems (IROS). vol. 3, 1995, pp. 344–349.

[8] Wilcox B., Matties L., Gennery D., Copper B., Nguyen T., Litwin T., Mishkin A., and S. H., "Robotic vehicles for planetary exploration," in IEEE Int'l Conf. on Robotics and Automation (ICRA), 1992, pp. 175–180.

[9] Howard A., Tunstel E., Edwards D., and C. A., "Enhancing fuzzy robot navigation systems by mimicking human visual perception of natural terrain traversability," in IFSA World Congress and 20th NAFIPS International Conference. vol. 1, 2001, pp. 7–12.

[10] M. Y. Shieh, Y. H. Chan, C. P. Cheng, and J. C. Hsieh, "Two-Stage Fuzzy Navigation Control of a Vision-Based Intelligent Shopping Service Robot," in The 30th Annual Conference of the EEE Industrial Electronics Society Busan, Korea, 2004.

[11] Jin Zhou and B. Li, "Homography-based Ground Detection for A Mobile Robot Platform Using a Single Camera," in Jin Zhou and Baoxin Li, 2006.

[12] Naoya Ohnishi and A. Imiya, "Visual Navigation of Mobile Robot Using Optical Flow and Visual Potential Field," RobVis, pp. 412-426, 2008.

[13] Motilal Agrawal, Kurt Konolige, and R. C. Bolles, "Localization and Mapping for Autonomous Navigation in Outdoor Terrains : A Stereo Vision Approach," in IEEE Workshop on Applications of Computer Vision (WACV'07), 2007.

[14] Stephen Williams and A. M. Howard, "A Single Camera Terrain Slope Estimation Technique for Natural Arctic Environments," in IEEE International Conference on Robotics and Automation Pasadena, CA, USA, 2008.

[15] Yehu Shen, Xin Du, and J. Liu, "Monocular Vision Based Obstacle Detection for Robot Navigation in Unstructured Environment," Springer ISNN, pp. 714–722, 2007.

[16] Shi J. and T. C., "Good Features to Track," in IEEE Conf. Computer Vision and Pattern Recognition, 1994, pp. 593-600.

[17] Intel, "Open Source Computer Vision Library (OpenCV)," Intel.

[18] Lucas B. and K. T., "An iterative image registration technique with an application to stereo vision," in Int. Joint Conf. on Artificial Intelligence, 1981, pp. 674–679.