PERGAMON

Pattern Recognition 33 (2000) 295–308

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Active vision-based control schemes for autonomous navigation tasks☆

Sridhar R. Kundur[a,*], Daniel Raviv[a,b]

[a]*Robotics Center and Department of Electrical Engineering, Florida Atlantic University, Boca Raton, FL 33431, USA*
[b]*Intelligent Systems Division, National Institute of Standards and Technology (NIST), Bldg. 220, Room B124, Gaithersburg, MD 20899, USA*

Received 28 August 1998; accepted 2 March 1999

Abstract

This paper deals with active-vision-based practical control schemes for *collision avoidance* as well as *maintenance of clearance* in a-priori unknown textured environments. These control schemes employ a *visual motion cue*, called the visual threat cue (VTC) as a sensory feedback signal to accomplish the desired tasks. The VTC provides some measure for a *relative change in range* as well as *clearance* between a 3D surface and a moving observer. It is a *collective* measure obtained *directly from the raw data of gray level images*, is *independent* of the type of 3D surface texture. It is measured in [time^{-1}] units and needs no *3D reconstruction*. The control schemes are based on a set of If–Then fuzzy rules with almost no knowledge about the vehicle dynamics, speed, heading direction, etc. They were implemented in real-time using a 486-based Personal Computer and a camera capable of undergoing 6-DOF motion.

Keywords: Active vision; Visual navigation; Collision avoidance

1. Introduction

The problem of automating vision-based navigation is a challenging one and has drawn the attention of several researchers over the past few years (see for example Refs. [1–18]). When dealing with a camera-based autonomous navigation system, a huge amount of visual data is captured. For vision-based navigation tasks like obstacle avoidance, maintaining safe clearance, etc., relevant visual information needs to be extracted from these visual data and used in real-time closed-loop control system. Several questions need to be answered, including: (1) What is the *relevant* visual information to be extracted from a sequence of images? (2) How does one extract this information from a sequence of 2D images? (3) How to generate control commands to the vehicle based on the visual information extracted?

This paper provides answers to all three questions with emphasis on the third one, i.e., generation of control signals for collision avoidance and maintenance of clearance using visual information only.

Vision-based autonomous navigation systems consist of a vehicle (such as a car, golf cart), visual sensing devices (camera, frame grabber for digitizing images) and other mechanical actuators for braking/steering of the vehicle. Relevant visual information is extracted from an image sequence and serves as input(s) to the feedback controller. The feedback controller generates appropriate signals to the mechanical actuators to brake/steer the vehicle (as shown in Fig. 1). Design of conventional feedback controllers needs a mathematical model of the system (including the vehicle as well as the mechanical actuators). Mathematical models for such systems are usually complex and may be difficult to define in some cases. On the other hand fuzzy logic control, which is closer in spirit to human thinking, can implement linguistically expressed heuristic control policies directly without any knowledge

☆This work was supported in part by a grant from the National Science Foundation, Division of Information, Robotics and Intelligent Systems, Grant # IRI-9115939.

*Corresponding author.
E-mail address: kundurs@rocketmail.com (S.R. Kundur)

0031-3203/99/$20.00
PII: S0031-3203(99)00058-8

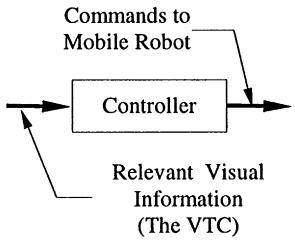


Fig. 1. Proposed controller.

about the mathematical model of a complex process. This paper presents two practical control schemes for vision-based autonomous navigation tasks such as collision avoidance and maintenance of clearance. The control schemes are based on a set of If–Then fuzzy rules and need almost no information about the vehicle kinematics, dynamics, speed, heading direction, etc. Also no a-priori information about the relative distances between the camera and the surfaces is necessary.

The main focus of this paper is to present details of the controllers to accomplish the above-mentioned tasks. The input to these controllers is the visual threat cue (VTC) that can be extracted from a sequence of images (see Refs. [24–26] for details on the VTC) and the output is appropriate braking/steering commands to the mobile robot (see Fig. 1). These control schemes were implemented in textured environments and are almost independent upon the type of texture in the scene. These approaches can be extended to texture-less environments as well [27].

1.1. Related work in vision-based autonomous navigation

Autonomous intelligent robots play an important role in many applications including industrial automation, space exploration, autonomous driving/flying, handling of hazardous materials, etc. Over the past few decades several researchers have been exploring approaches to build intelligent, goal driven robotic systems which can interact with the environment autonomously (see for example Refs. [1–18]). In the absence of a-priori information about the environment, such an intelligence in the robots may be imparted by using external sensors such as tactile, visual, audio, etc., to sense the environment and interact with it in an intelligent manner. Since our approach is based on visual sensing, we restrict our attention to vision-based intelligent robots only.

In the animate world visual information plays a key role in controlling animal behavior in the environment (see for example Refs. [22,23]). Several psychologists have suggested that vision is the primary source of information about the surroundings and is responsible for controlling visual behavior of humans in the environment (see for example Refs. [19–23]). These observations have motivated many researchers to employ visual information as the primary source of sensory feedback in building intelligent robotic systems. A brief review of related work in vision-based autonomous navigation is described in the following paragraphs.

1.1.1. Vision-based autonomous navigation using a-priori information

Papanikolopoulos and Khosla [1] presented algorithms for real-time visual tracking of arbitrary 3D objects moving at unknown velocity in a plane whose depth information is assumed to be known a-priori. They proposed an optical flow-based approach to compute the vector of discrete displacements each instant of time. In Ref. [2], they described a vision sensor in the feedback loop within the framework of controlled active vision. This approach *requires partial knowledge* of the relative distance of the target with respect to the camera which obviates the need for off-line calibration of the eye-in-hand robotic system.

Feddema and Lee [3] proposed an adaptive approach for visual tracking of an a-priori *known* moving object with a monocular mobile camera. They employed a *geometrical* model of the camera to determine the linear differential transformation from image features to the camera position and orientation. Computer simulations were provided to verify the proposed algorithms.

Reid et al. [4] described an active vision system to perform a surveillance task in a dynamic scene. These ideas were implemented in a special purpose high performance robotic head/eye platform. The controller was divided into two parts namely the high-level control which has some knowledge about vision and behaviors and the low-level control *had information* about head kinematics and dynamics, via joint angles and velocities from the encoders and the motor torques.

Han and Rhee [5] describe a navigation method for a mobile robot that employs a monocular camera and a *guide mark*. They instruct the robot by means of a path drawn on a monitor screen. The images of the guide mark obtained by the camera provides information regarding the robot's position and heading direction. It adjusts the heading direction if any deviation in the specified path is detected. This approach was implemented on a real system with average speeds of 2.5 feet/s. with deviations of less than one foot in an indoor environment.

Turk et al. [6] described an approach to distinguish road and non-road regions by employing color images. They generate a new trajectory by minimizing a cost function based on the current heading of the vehicle, curvature of the road scene model, attraction to a goal location, and changes in the road edges. It is then sent to the pilot module which controls vehicle motion using lateral position, heading, and velocity error signals. They successfully implemented this approach to drive an autonomous land vehicle (ALV) at speeds up to 20 km/h. The vehicle motion is assumed to be *known*.

1.1.2. Autonomous navigation using conventional feedback control approaches

Feng and Krogh [7] describe a general approach for local navigation problems for autonomous mobile robots and its applications to omnidirectional and conventionally steered wheel-bases. They formulate the problem of driving an autonomous mobile robot as a *dynamic feedback control problem* in which local feedback information is employed to make steering decisions. A class of satisficing feedback strategies is proposed to generate reasonable collision-free trajectories to the goal by employing robot dynamics and constraints. This approach was *tested in simulations*.

Krotkov and Hoffman [8] present a terrain mapping system that constructs quantitative models of surface geometry for the Ambler, an autonomous walking robot designed to traverse terrains as on Mars. It employs a *laser* range finder to construct elevation maps at arbitrary resolutions. A *PI control* scheme that employed the elevation error as an error signal to increase the accuracy of the elevation maps was implemented to adjust the elevation values.

1.1.3. Optical flow-based autonomous navigation

In Ref. [9], Olson and Coombs outlined a general approach to vergence control that consisted of a control loop driven by an algorithm that estimates the vergence error. Coombs and Roberts [10] demonstrated the centering behavior of a mobile robot by employing the peripheral optical flow. The system employs the maximum flow observed in left and right peripheral visual field to indicate obstacle proximity. A steering command to the robot is generated on the basis of left and right proximities extracted using optical flow information.

Santos-Victor et al. [11] presented a qualitative approach to vision-based autonomous navigation on the basis of *optical flow*. It is based on the use of *two cameras* mounted on a mobile robot and with the optical axis directed in opposite directions such that there is no overlap between the two visual fields. They implemented these schemes on a computer controlled mobile platform *TRC Labmate*.

1.1.4. Non-optical flow-based autonomous navigation

Romano and Ancona [12] present a visual sensor to obtain information about time-to-crash based on the expansions or contractions of the area without any explicit computation of the optic flow field. This information extracted from images was fed to an opto-motor reflex, operating at 12.5 Hz. This controller was able to keep a constant distance between a frontal obstacle and itself. The whole approach was implemented and tested on a mobile platform.

Joarder and Raviv [13] describe a looming-based algorithm [14] for autonomous obstacle avoidance. Visual looming is extracted from relative temporal variations of projected area in the image and employed it as a sensory feedback to accomplish obstacle avoidance. In Ref. [15], they have implemented a similar algorithm for obstacle avoidance by measuring looming from relative temporal variations of the edge density in a small window around the fixation point. Both the algorithms were implemented on a 6DOF flight simulator in indoor environment.

In Ref. [16], Broggi presented a vision-based road detection system that is implemented on a land vehicle called the MOB-LAB. It is assumed that the *road is flat* and the complete acquisition parameters (camera optics, position, etc.) are *known*. The system is capable of detecting road markings on structured roads even in extremely severe shadow conditions.

In Ref. [17], Leubbers describes a neural-network-based feature extraction system for an autonomous high-mobility multi-wheeled vehicle application (HMMWV). A video camera is employed to obtain the images of the road and a neural network is employed to extract visual features from image sequences. The road following task was posed as a regulatory control task and an expert system was used to improve the robustness of the control system.

Yakali [18] describe several 2D visual cues for autonomous landing and road following tasks. Using these visual cues road following tasks were successfully tested on a US army land vehicle HMMWV equipped with a video camera in outdoor environments and also on a Denning mobile robot in indoor environments. The autonomous landing task was implemented on a 6-DOF flight simulator in indoor environment.

The above-mentioned references indicate that some autonomous vision-based navigation systems need a-priori information about the environment [1–6]. Information about the environment may not be known a-priori in some situations. Some approaches employ conventional feedback controllers [7,8]. The design of conventional controller needs mathematical models of the navigation system. The navigation systems are usually complex and may be difficult to obtain their mathematical models. The reliability of optical flow-based approaches depends upon the reliability of measurement of optical flow from a sequence of images. Reliable extraction of optical flow may be difficult in some outdoor scenarios such as variations in lighting, vehicular vibrations, wind, etc. The non-optical flow-based approaches need information about image features such as areas, centroids, edges, texture, etc. These image features usually depend upon the type of texture in the environment, the camera used to capture the image and the camera parameters such as focus, zoom, etc.

This paper describes control schemes for *collision avoidance* as well as *maintenance of clearance* tasks in *a-priori unknown* textured environments (it is possible to extend these approaches to texture-less environments also). These control schemes employ a *visual motion cue*,

called the visual threat cue (VTC) as a sensory feedback signal to accomplish the desired tasks. The VTC is a *collective* measure that can be obtained *directly from the raw data of gray level images, is independent* of the type of 3D surface texture. It is measured in [time^{-1}] units and needs no *3D reconstruction*. The VTC is described in the following section.

2. Overview of the visual threat cue (VTC)

Mathematically the VTC is defined (for $R > R_0$) as follows [24–27]:

$$VTC = -R_0\frac{(d/dt)(R)}{R(R-R_0)},$$

where R is the range between the observer and a point on the 3D surface, $d(R)/dt$ is the differentiation of R with respect to time and R_0 is the desired minimum clearance. Note that the units of the VTC are [time^{-1}].

There are imaginary 3D iso-VTC surfaces attached to an observer in motion and are moving with it [24–27]. A qualitative shape of the iso-VTC surfaces is presented in Fig. 2a. A positive value of the VTC corresponds to the space in front of the observer and a negative value corresponds to the region back of the observer. The points that lie on a relatively smaller surface corresponds to a relatively larger value of VTC, indicating a relatively higher threat of collision. The VTC information can be used to demarcate the region around an observer into safe, high risk, and danger zones (Fig. 2b). Based on this knowledge one can take an appropriate control action to prevent collisions or maintain clearance [28].

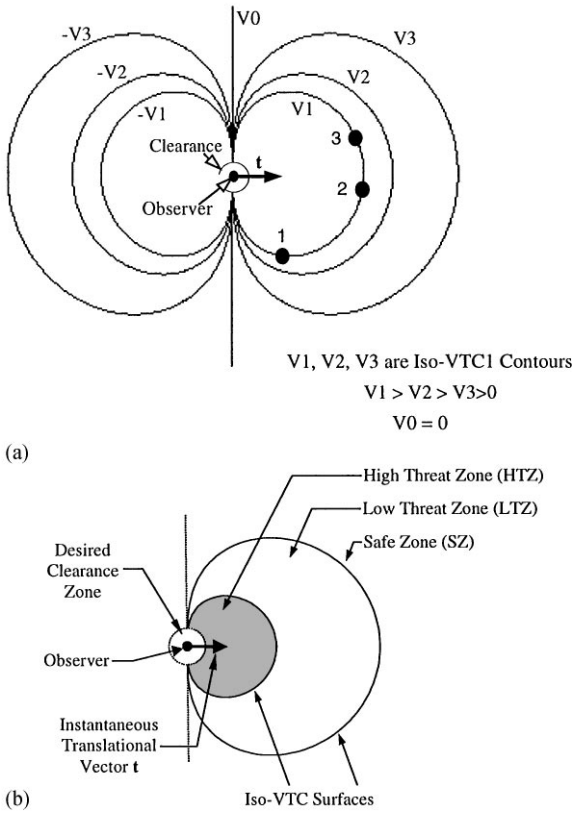


Fig. 2. (a) Visual field of VTC. (b) Qualitative demarcation of space into threat zones by the VTC.

A practical method to extract the VTC from a sequence of images of a 3D textured surface obtained by a *fixated, fixed-focus monocular* camera in motion has been presented in Refs. [24–27]. This approach is *independent* of the type of 3D surface texture and needs almost no camera calibration. For each image in such a 2D image sequence of a textured surface, a *global variable* (which is a measure for dissimilarity) called the image quality measure (IQM) is obtained *directly* from the raw data of the *gray-level images*. The VTC is obtained by calculating relative temporal changes in the IQM. This approach by which the VTC is extracted can be seen as a *sensory fusion* of focus, texture and motion at the *raw-data level*. The algorithm to extract this cue works *better* on natural images including fractal-like images, where more details of the 3D scene are visible in the images as the range shrinks and also can be implemented in parallel hardware. The VTC can be used to directly maintain clearance in unstructured environments.

2.1. Image quality measure (IQM)

Mathematically, the IQM is defined as follows [24–26]:

$$IQM = \frac{1}{|D|}\sum_{x=x_i}^{x_f}\sum_{y=y_i}^{y_f}\left(\sum_{p=-L_c}^{L_c}\sum_{q=-L_r}^{L_r}|I(x,y)-I(x+p,y+q)|\right),$$

where $I(x, y)$ is the intensity at pixel (x, y) and x_i and x_f are the initial and final x-coordinates of the window respectively; y_i and y_f are the initial and final y-coordinates of the window in the image respectively and L_c and L_r are positive integer constants; and D is a normalization factor defined as $D = (2L_t + 1)\times(2L_r + 1)\times(x_f - x_i)\times(y_f - y_i)$. The IQM is a measure for the dissimilarity of gray level intensity in the image.

The advantages of using this measure are: (1) It gives a *global* measure of quality of the image, i.e., *one* number which characterizes the image dissimilarity is obtained, (2) It *does not need any preprocessing*, i.e., it works directly on the raw gray level data without any spatio-temporal smoothing or segmentation, (3) It *does not need a model of the texture* and is suitable for many textures and (4) It is *simple* and can be *implemented in real time on parallel hardware*.

2.2. Extraction of the VTC from IQM

Based on experimental results (indoor as well as outdoor) [26], we observed that the relative temporal changes in the IQM behave in a very similar fashion to the VTC, i.e.,

$$\frac{\mathrm{d}(IQM)/\mathrm{d}t}{IQM} \cong -R_0 \frac{\mathrm{d}(R)/\mathrm{d}t}{R(R-R_0)}.$$

This means that the VTC can be measured using the IQM. The VTC is *independent* of the magnitude of the IQM. A sample set of four images (out of 71) that corresponds to a texture from Brodatz's album [30] as seen by a visually fixating, moving, fixed-focus camera is shown in Fig. 3a. A comparison of the measured as well as theoretical values of the VTC is shown in Fig. 3b. Very similar results were reported in Ref. [25] for 12 different textures of the same album [30].

2.3. Qualitative view of $\{d(IQM)/dt\}/\{IQM\}$

As shown in the previous sections the VTC is defined only in a region beyond a certain desired minimum clearance R_0 and is not defined when the distance between the camera is less than R_0. Though we restrict ourselves to regions beyond the desired minimum clearance there might be situations when one is in the region for which the distance between the camera and the surface is less than R_0. Since the VTC is undefined in this region it cannot be employed when the robot is in this region. However the IQM and relative temporal variations in IQM ($\{\mathrm{d(IQM)}/\mathrm{d}t\}/\{\mathrm{IQM}\}$) can be used since it is an image measure and is defined irrespective of the distance between the camera and the surface. Note that the VTC is very similar to the relative temporal variations of the IQM (see Fig. 4a–c).

3. Control objectives

Two vision-based control schemes have been implemented on a six DOF flight simulator using the VTC as a sensory feedback signal. This section describes the desired control tasks, the constraints and the motivation for the choice of the control schemes employed.

3.1. Control task I: Collision avoidance

The objective of this control task is to *stop* a moving robot in front of an a-priori *unknown textured obstacle* when the distance between the camera and the obstacle is equal to a certain desired clearance R_0 (see Fig. 5a), employing visual information only.

3.2. Control task II: maintenance of clearance

The objective of this control task is to maintain a constant clearance between an a-priori unknown textured surface and a mobile robot using visual information only (see Fig. 5b).

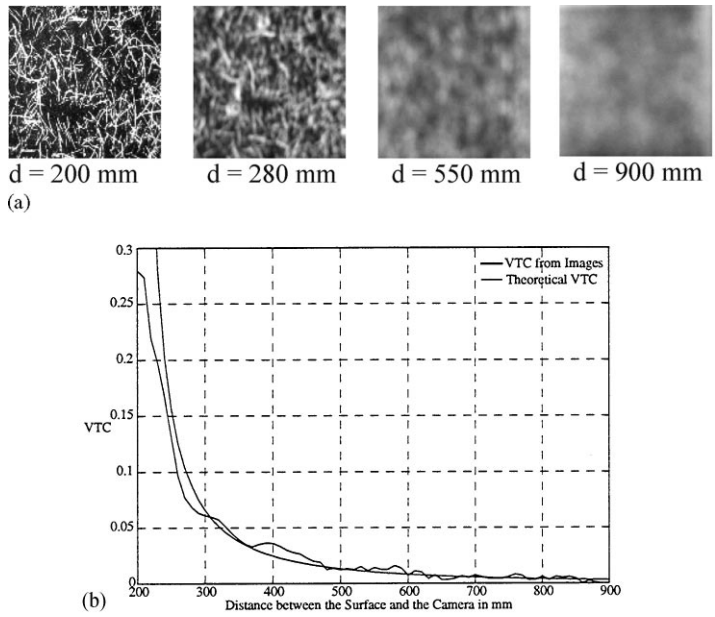


Fig. 3. (a) Images of D110, d is the relative distance. (b) Comparison of theoretical and measured values of the VTC.

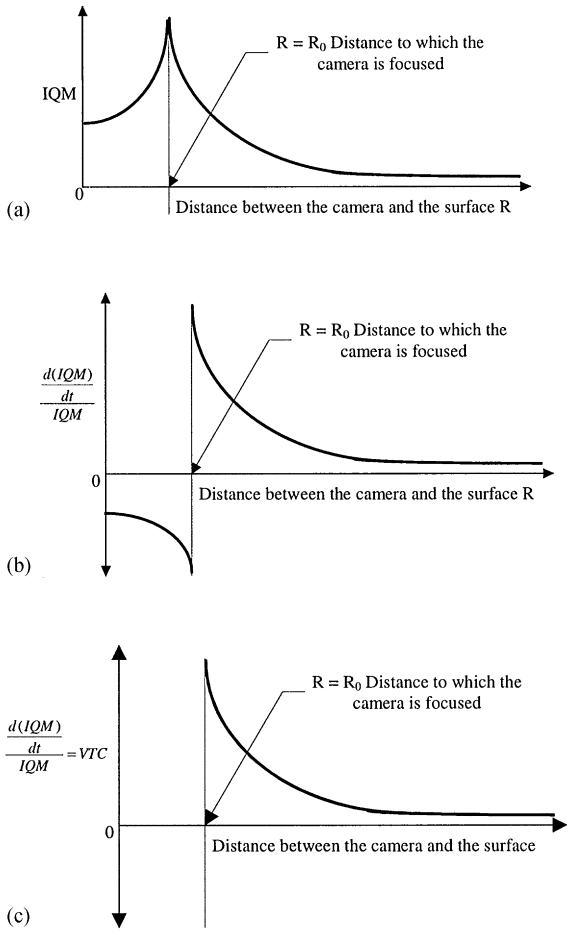


Fig. 4. (a) Qualitative behavior of IQM vs. relative range. (b) Qualitative behavior of the relative temporal variations of IQM vs. relative range. (c) Qualitative behavior of the VTC vs. relative range. *Note*: for $R > R_0$ VTC is very similar to relative temporal variations of IQM.

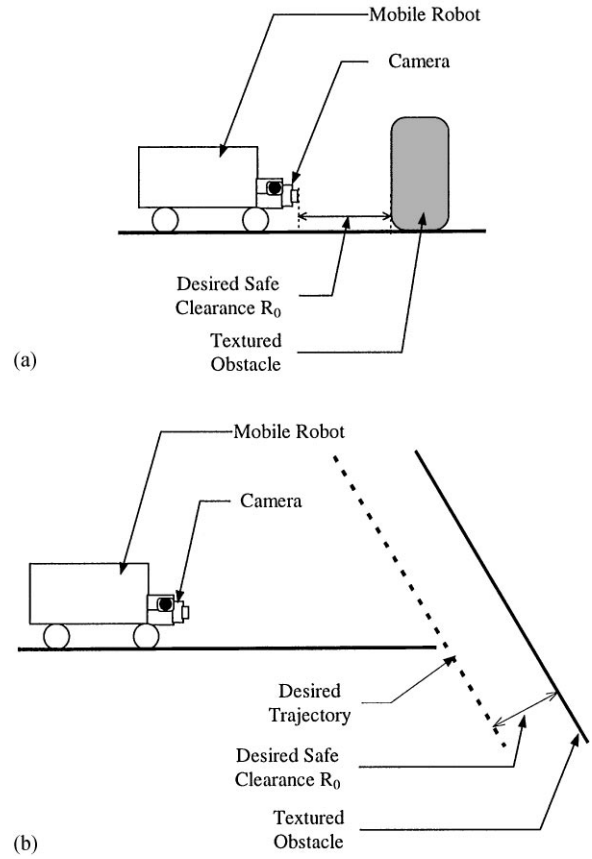


Fig. 5. (a) Control objective I. (b) Control objective II.

3.3. The constraints

The above-mentioned control tasks have to be accomplished with the following constraints:

- The input to the controllers is *visual information* only.
- *No information* about the vehicle speed or dynamics is available.
- The controllers have no a-priori information about the distance between the camera and the obstacles or the type of the texture on the obstacles.
- Obstacles must have texture on them.

3.4. Choice of control schemes

In the absence of vehicle dynamics conventional control schemes such as PID schemes are difficult to implement. On the other hand, Fuzzy Logic Control which consists of a set of collection of rules seems to be more appropriate for the control tasks with the above mentioned constraints.

Research in the area of fuzzy control was initiated by Mamdani's pioneering work [31], which had been motivated by Zadeh's seminal papers on fuzzy algorithms [32] and linguistic analysis [33]. In the past few years several researchers have addressed the use of fuzzy control for various ill-defined processes for which it is difficult to model the dynamics (see for example Refs. [34–36]). Fuzzy control is closer in spirit to human thinking and can implement linguistically expressed heuristic control policies directly without any knowledge about the dynamics of the complex process.

Several collision avoidance schemes based on fuzzy approaches have been suggested for autonomous navigation tasks [37–41]. These approaches required many parameters such as the range between the camera and the surface, slant of the surface, heading angle of the robot, width of the road, shape of the road, etc. Usually these control schemes are simulated on a computer without real implementations.

The VTC mentioned in Section 2 provides an indication for relative variations in ranges as well as clearances.

In other words, if the VTC values increase one can say that the vehicle is moving forward and vice-versa. In order to judge whether the vehicle is moving far away or close to the desired clearance it may not be necessary to know the mathematical model of the vehicle. Autonomous navigation tasks such as collision avoidance and maintenance of clearance can be accomplished by a set of heuristic rules based on the VTC, without vehicle models. This is the main motivation for choosing fuzzy control schemes among several possible conventional control schemes.

4. Fuzzy approach to task I: collision avoidance

This section describes the fuzzy logic control scheme employed to accomplish task I, namely to *stop* a moving robot in front of a textured surface when the distance between the surface and the robot equals a desired range R_0. No a-priori information such as the relative distance, type of texture, etc., about the obstacle is available to the robot. The design of this controller assumes no mathematical model of the robot and is based on a set of simple IF–THEN rules. A block diagram of the proposed control scheme is shown in Fig. 6.

The camera is initially focused to the distance which is equal to R_0 which is the desired stopping gap between the robot and the textured surface. For ranges R greater than R_0, as the range increases the VTC value decreases and vice-versa. Based on the VTC values, we divide the space in front of the mobile robot into three different regions as shown in Fig. 7a and b. Region I can be seen as a safe region and regions II and III can be seen as danger zones. If the VTC value is greater than a certain positive threshold say VTC^{Th} then the textured surface is in the danger zone of the robot. When the measured VTC is smaller compared to the threshold VTC^{Th} then the textured surface is in the safe zone (region I in Fig. 7a). If the measured value of the VTC is greater than the threshold VTC^{Th} then the textured surface is in the danger zone of the robot. Finally when the VTC values change from positive to negative it provides an indication that the textured surface has entered the desired clearance zone and the robot has to stop moving forward to avoid a collision with the textured surface.

Based on the heuristic information about the behavior of the VTC as a function of the range between the robot and the textured surface, we formulate the following rules suitable for achieving the desired control task. It should be noted that we try to demonstrate the use of the VTC as sensory feedback information for collision avoidance and these set of the rules are not the only possible set of rules to accomplish the desired task. Alternative set of rules could be formulated for better control.

Rule I: This rule corresponds to the case when the robot is in the safe zone (region I in Fig. 7a). In this zone, no control action should be taken, i.e., no change in speed is necessary. The sensing and action corresponding to this region can be expressed in the IF–THEN format as follows:

If the measured VTC value *is less* than the threshold VTC^{Th} *then* take no action.

Rule II: This rule corresponds to region II in Fig. 7a. If the textured surface is in this region then the value of the value of the measured VTC is greater than the threshold and the robot has to be prepared to stop any time the measured VTC values changes from positive to negative. The condition can be expressed in an IF–THEN format as follows:

If the measured VTC is *greater* than the threshold VTC^{Th} *then* be prepared to stop anytime the measured VTC value becomes negative.

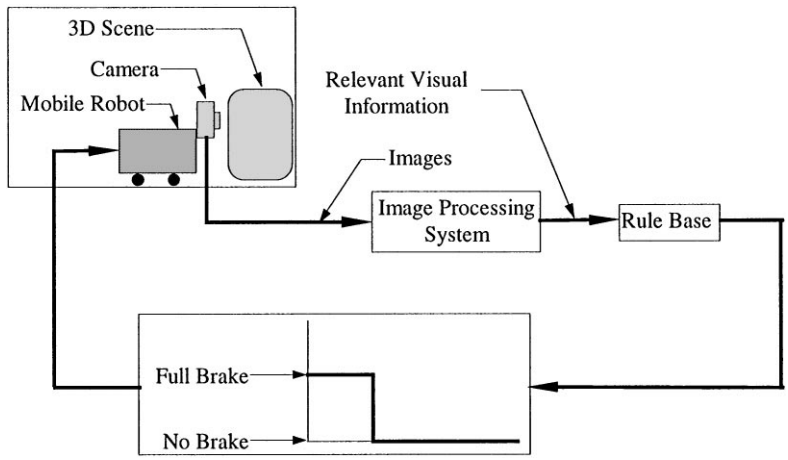


Fig. 6. Block diagram of control scheme I.

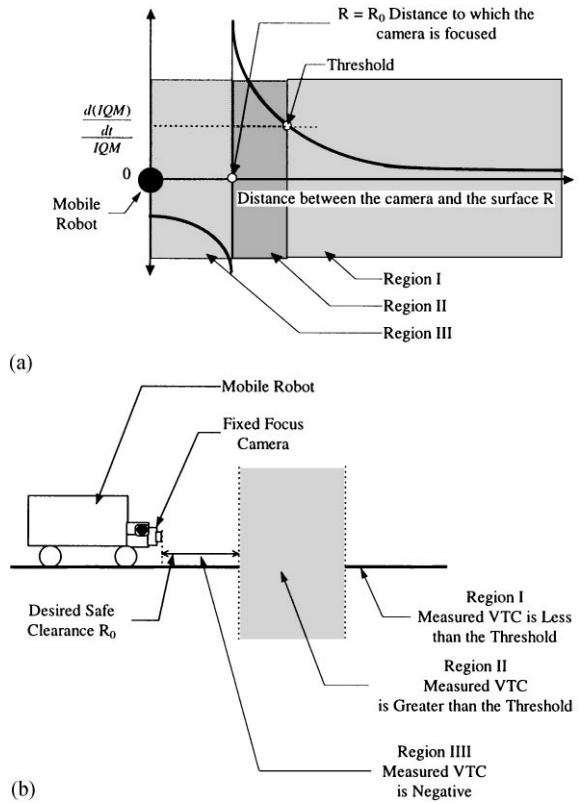


Fig. 7. (a) Qualitative plot of relative temporal variations of IQM. (b) Fuzzy demarcation of space around the mobile robot for control task I.

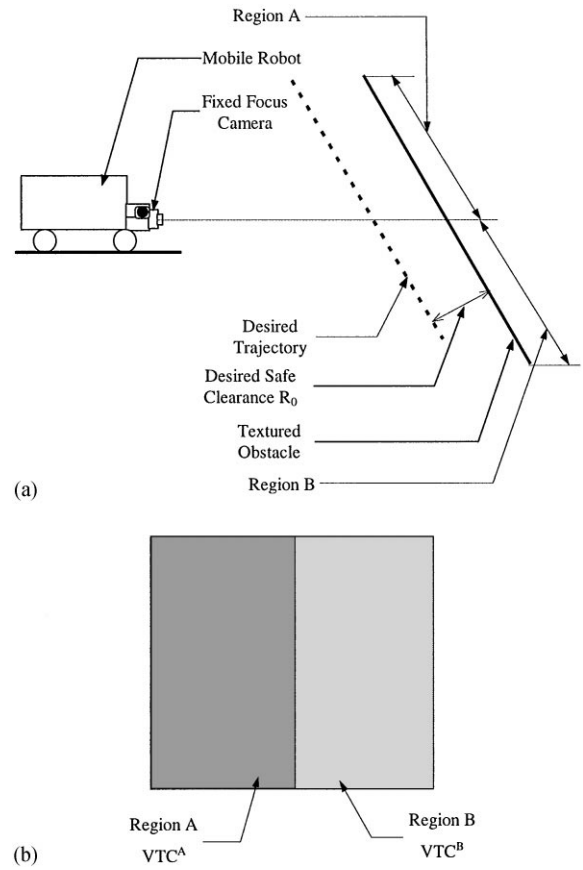


Fig. 8. (a) Control task II. Note: Region A corresponds to the left of the camera and region B corresponds to the right of the camera. (b) Regions of interest in images for control task II.

Rule III: This rule corresponds to region III in Fig. 7a. In this region the robot is required to stop if it is moving towards the surface. Note that in this region VTC is negative. This condition can be expressed in the IF–THEN format as follows:

If the measured VTC is negative *then* Stop.

Rule IV: When the robot is either stationary or moving away from the surface, none of the above conditions are satisfied and hence no control action is taken. This condition can be expressed in IF–THEN format as follows:

IF none of the above situations occur *THEN take no action*.

5. Fuzzy approach to task II: maintenance of clearance

This section describes the fuzzy logic control scheme employed to accomplish task II, i.e., maintenance of clearance (refer to Fig. 8a and b). A block diagram of the proposed control scheme is shown in Fig. 9.

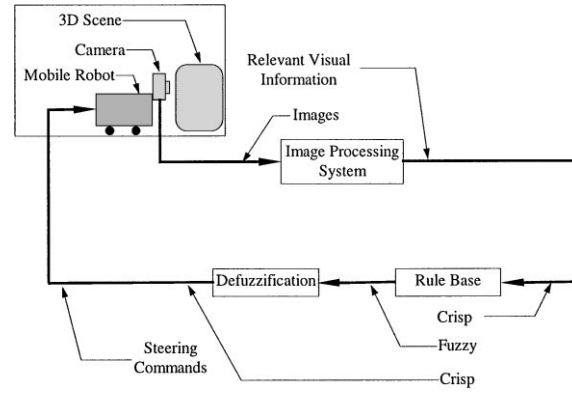


Fig. 9. Block diagram of the control scheme II.

In Fig. 8a the left region (region A) is closer to the camera than the right region (region B). The camera is initially focused at a desired minimum clearance R_0. When the distance between the camera and the surface is greater than the desired minimum clearance, the points located at a greater distance have relatively smaller values of the VTC than those located at a relatively

smaller distance. In other words, the difference between the VTC values of the left window (denoted as region A) and the VTC value of the right window (denoted as region B) can be used to generate appropriate control action to maintain a safe clearance between the textured surface and the mobile robot. The difference in the measured VTC values of the left and the right windows is the only information that is fed to the controller and the controller generates appropriate control action to the mobile robot to accomplish the desired task.

Based on the heuristic information about the behavior of the VTC as a function of the range between the robot and the textured surface, we formulate the following rules suitable for achieving the desired control task. It should be noted that we try to demonstrate the use of the VTC as sensory feedback information for collision avoidance and these set of the rules are not the only possible set of rules to accomplish the desired task. Alternative set of rules could be formulated for better control.

For the sake of simplicity let the difference between VTC^A and VTC^B be denoted as Err_AB. In other words, $Err\_AB = VTC^A - VTC^B$.

Rule I: This rule is formulated to take care of the control when the Err_AB is almost equal to zero. In such a situation the motion to the right (see Fig. 8a) is almost zero. This can be expressed in an IF–THEN format as follows:

IF Err_AB is approximately zero *THEN* motion to the right is approximately zero.

Rule II: This rule is formulated to take care of the control when the Err_AB small. In such a situation the motion to the right is small (see Fig. 8a). This can be expressed in an IF–THEN format as follows:

IF Err_AB is positive small *THEN* motion to the right is small.

Rule III: This rule is formulated to take care of the control when the Err_AB is medium. In such a situation the motion to the right medium (see Fig. 8a). This can be expressed in an IF–THEN format as follows:

IF Err_AB is positive medium *THEN* motion to the right is medium.

Rule IV: This rule is formulated to take care of the control when the Err_AB is big. In such a situation the motion to the right is big (see Fig. 8a). This can be expressed in an IF–THEN format as follows:

IF Err_AB is positive big *THEN* motion to the right is big.

Rule V: According to this rule when the region A is within the desired clearance and region B is in the region beyond the desired clearance, the desired control is to move to the right and move backwards (see Fig. 8a). This can be expressed in an IF–THEN format as follows:

IF $VTC^A < 0$ and $VTC^B > 0$ *THEN* motion to the right is *big* and *reverse* the current direction of motion.

Rule VI: According to this region both region A and region B are within the desired clearance region. The desired control action is to move the robot backwards.

IF $VTC^A < 0$ and $VTC^B < 0$ *THEN* motion to the right is *big* and *reverse* the current direction of motion.

Rule VII: When the textured surface is perpendicular to the direction of motion of the mobile robot, Err_AB is going to be zero irrespective of the distance between the robot and the mobile robot. In such a case Rules I – VI will fail and the robot might collide with the textured surface instead of maintaining a safe clearance. This situation may be overcome by the following IF–THEN condition:

IF Err_AB is almost zero and VTCA < 0 and VTCB < 0 *THEN* move sidewards (either right or left).

Rule VIII: When the robot is either stationary or moving away from the surface none of the above mentioned conditions are satisfied. This situation can be expressed in an IF–THEN format as follows:

IF none of the above situations occur *THEN take no* change in the velocity.

5.1. Membership functions employed

Simple linear membership functions are employed in the fuzzy rule base (as shown in Fig. 10).

5.2. Defuzzification

Defuzzification of the inferred fuzzy control action is necessary in order to produce a crisp control action. Since monotonic membership functions are used, we use Tsukamoto's defuzzification method [37], which is stated

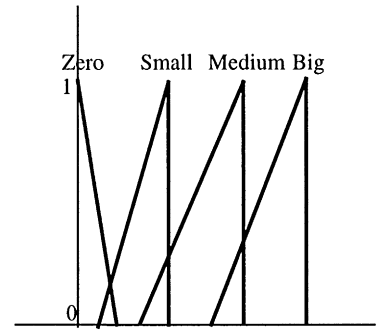


Fig. 10. Qualitative membership functions for control scheme II.

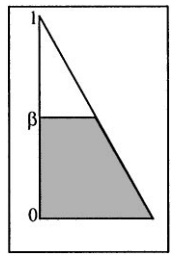


Fig. 11. Evaluation of weight of a particular rule.

as follows:

$$Z^* = \frac{\sum_{i=1}^{n} \alpha_i y_i}{\sum_{i=1}^{n} \alpha_i},$$

where Z^* is the defuzzified crisp control command and α_i is the weight corresponding to the rule i; y_i is the amount of control action recommended by rule i and n is the number of rules.

The ratio of the shaded area to the area of the triangle is used as the firing strength (see Fig. 11) and is employed as the weight corresponding to that particular rule:

$$\alpha_i = \beta_i(2 - \beta_i),$$

when β_i equals 1, the shaded area equals the area of the triangle, hence α_i is 1.

6. Implementation details

The control algorithms presented in the previous sections are implemented on a 6-DOF vision-based flight simulator controlled by a 486-based Personal Computer. This section presents implementation details of the control schemes.

6.1. Experimental setup

The system used in the experiments include:

1. Six DOF miniature flight simulator
2. CCD video camera
3. Imaging Technology^{tm} Frame Grabber
4. 486-based personal computer
5. Photocopies of texture plates D9, D110 from Brodatz's Album [30] pasted on a flat board employed as textured surface.

6.2. Six-DOF miniature flight simulator

An IBM gantry robot has been modified such that all the six motor controllers can accept velocity inputs. A monocular camera is attached to the end effector of the robot. This camera is capable of undergoing six-DOF motion within the workspace of the robot (a block diagram is shown in Fig. 12). Various types of miniature environments (structured as well as unstructured) can be simulated in the workspace by physical placing of objects such as toy mountains, trees, etc.

A sequence of images is obtained by the camera and the relevant image processing is done by the image processing hardware/software housed in the 486-based Personal Computer. A single 486-based, 50 MHz, Personal Computer is employed to control the robot as well as perform the relevant image processing. Control

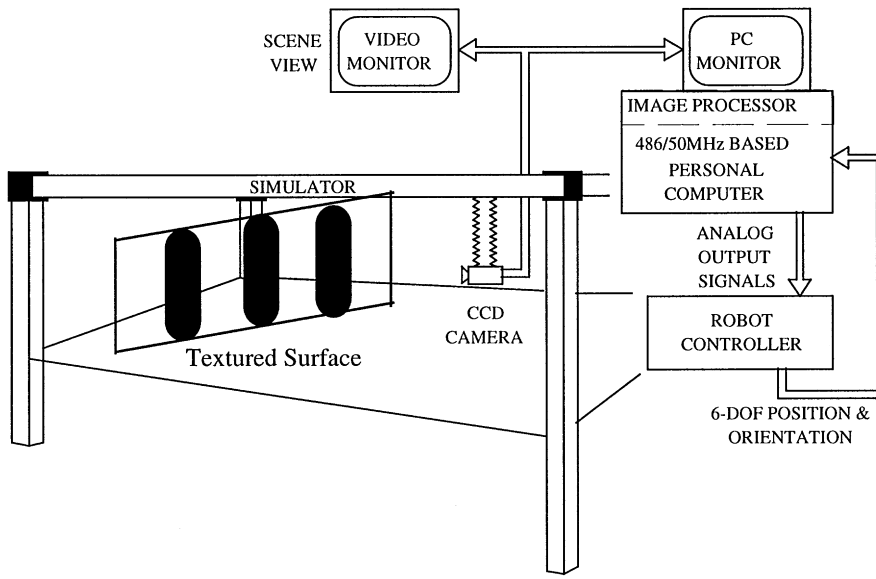
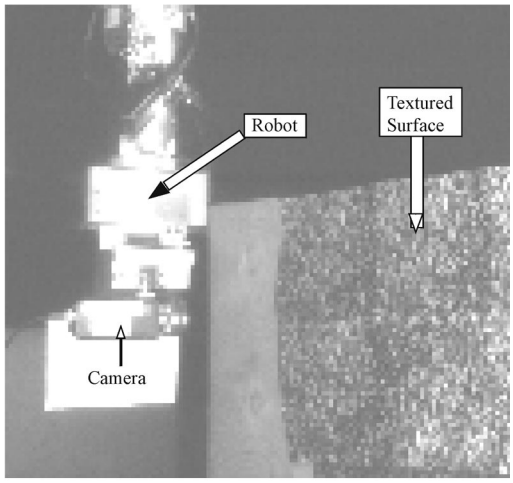
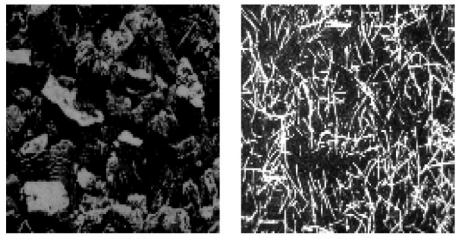


Fig. 12. Block diagram of the simulator and setup.



(a)



(b)

Fig. 13. (a) Camera mounted on the robot. (b) Textures used in the control experiments.

commands to the robot are generated on the basis of relevant visual information extracted from the image sequence.

6.3. Procedure

A CCD video camera is used to obtain the images of the 3D textured environments. These images are digitized by an image processing PC-board ITEX PC-VISION PLUS. A block diagram of the experimental setup is shown in Fig. 12. Textured pattern (D5, D110 from Ref. [30], see Fig. 13b) pasted on a flat surface is presented as the obstacle along the path of the robot (see Fig. 13a). For both control schemes the camera is initially focused to the desired minimum clearance ($R_0 = 200$ mm.). Qualitative measures for fuzzy sensing and action (small, medium, big, etc.) are employed, rather than the exact speeds.

6.4. Control scheme I

A window of size 51×51 pixels is chosen in the center of the image to evaluate the visual feedback signal VTC. According to the rules presented in the previous section the crisp control action is (either move or stop) is generated (see Fig. 6). Two different speeds were employed in this control scheme (speed2 > speed1).

6.5. Control scheme II

Two windows (left and right) each 50×50 pixels are opened in the image. In each the visual parameter VTC is evaluated and based on the difference between left and right values an appropriate control signal is generated. This control scheme was tested for four different orientations of the texture surfaces used.

7. Results and analysis

7.1. Control scheme I

Two different speeds were used to test the braking capability of the control algorithm. We observed that the greater the speed of the robot, the greater is the error between the desired and actual values of the clearance between the robot and the surface. The results are summarized in Table 1.

From Table 1 it can be seen that there is an error between the desired stopping point and the actual stopping distance. This error is due to the inertia of the robot and mainly depends upon the speed at which the robot is traversing, in otherwords, at higher speeds the error is high and at lower speeds the error is lower. This error can be minimized by applying the brakes to the robot even before it reaches the desired clearance point. The point where it should start applying braking before reaching the desired clearance may be determined by employing additional visual motion cues (see Ref. [29] for additional visual motion cues).

7.2. Control scheme II

The lateral and longitudinal components of the heading vector were recorded. The resultant was plotted manually (see Figs. 14–17). Two sets of results using two texture patterns (shown in Fig. 13b) are presented. Each

Table 1
Summary of vision-based collision avoidance results

No.	Texture	Speed	Desired	Actual	Error
1	D5	Speed1	200 mm	180 mm	20 mm
2	D5	Speed2	200 mm	165 mm	35 mm
3	D110	Speed1	200 mm	180 mm	20 mm
4	D110	Speed2	200 mm	165 mm	35 mm

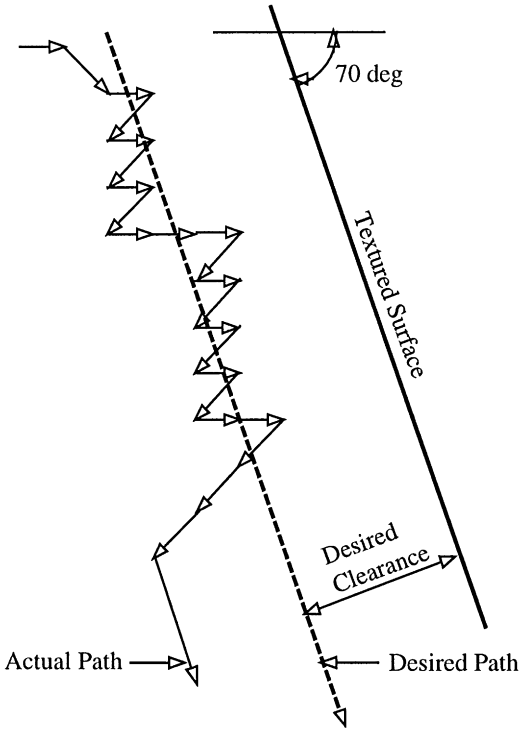


Fig. 14. Results of control scheme II for D110: Case 1.

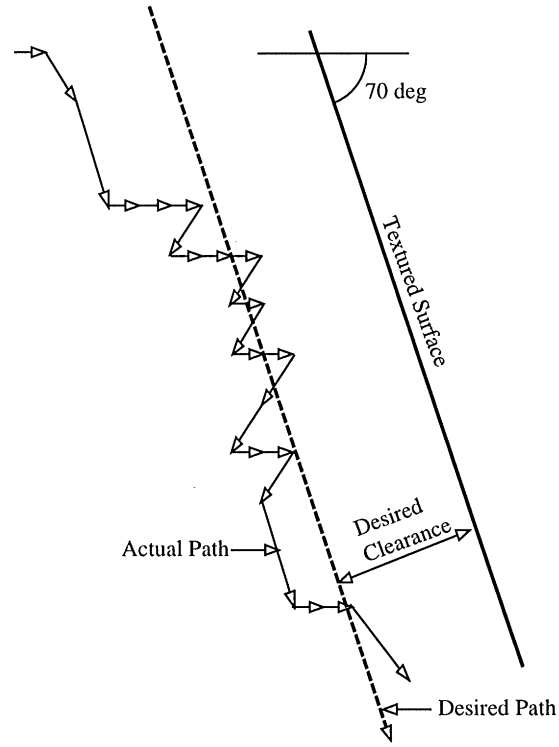


Fig. 16. Results of control scheme II for D5: Case 1.

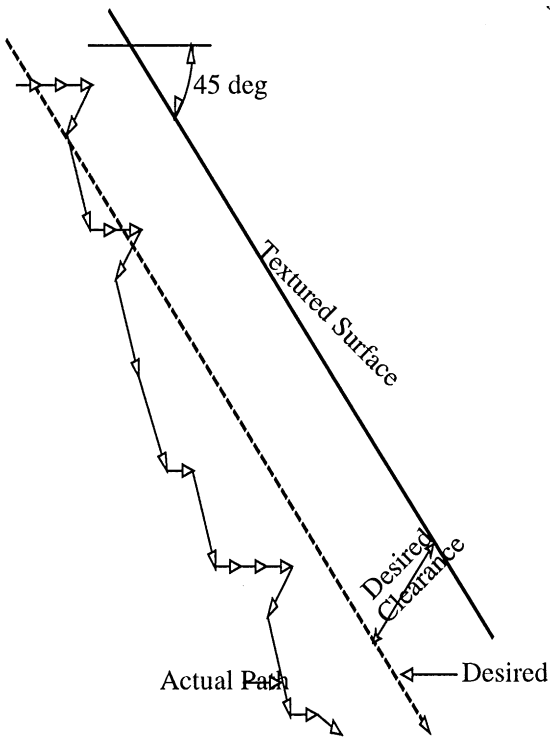


Fig. 15. Results of control scheme II for D110: Case 2.

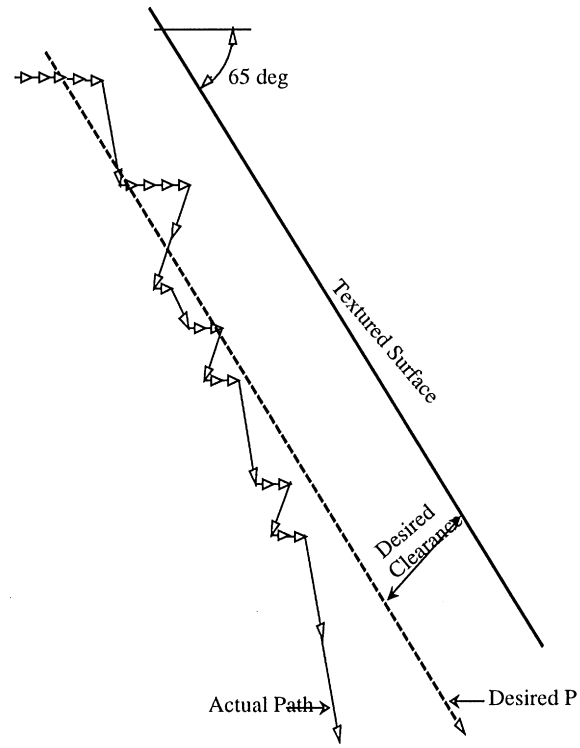


Fig. 17. Results of control scheme II for D5: Case 2.

texture pattern was tested under two different orientations. All the four experiments in this control scheme employed the same rule base. The error between the desired path and the actual path is highly dependent upon the choice of fuzzy membership functions, rule-base and defuzzification schemes used. Addition of more rules to the existing ones may improve the error between the desired and actual paths. Also by employing temporal smoothing to the measured VTC values may improve the error.

8. Conclusions and future work

This paper presented implementable active-vision-based real-time closed-loop control schemes for collision avoidance and maintenance of clearance tasks. The control schemes are based on a new measure called the VTC that can be extracted directly from raw gray-level data of monocular images. In other words, the VTC is the relevant visual information used in the control tasks. The VTC needs no optical flow information, segmentation or feature tracking. The control schemes are based on a set of If–Then fuzzy rules and needs no information about the robot dynamics, speed, heading direction, etc.

From the experimental results, it can be seen there is some error between the desired trajectory and the actual trajectory. Some possible sources of this error include: slower computation of the VTC, no temporal smoothing of the measured values of the VTC, choice of rules in the rule base, etc. It is possible to obtain better performances by using some temporal smoothing for the measured values of the IQM as well as using high-speed computers (may be parallel hardware implementations).

Currently we are working on the extensions of the control schemes mentioned in this paper to *unstructured outdoor* environments using a golf cart (known as LOOMY) designed and built at Florida Atlantic University. Preliminary results in real outdoor environments are highly encouraging.

References

[1] N.P. Papanikolopoulos, P.K. Khosla, Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision, IEEE Trans. Robot. Automat. 9 (1) (1993) 14–18.

[2] N.P. Papanikolopoulos, P.K. Khosla, Adaptive robotic visual tracking: theory and experiments, IEEE Trans. Automat. Control 38 (3) (1993) 429–444.

[3] J.T. Feddema, C.S.G. Lee, Adaptive image feature prediction and control for visual tracking with a hand-eye coordinated camera, IEEE Trans. Systems, Man, Cybernet. 20 (5) (1990) 1172–1183.

[4] I.D. Reid, K.J. Bradshaw, P.F. McLauchlan, P.M. Sharkey, D.W. Murray, From saccades to smooth purs uit: real-time gaze control using motion feedback, Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Yokohama, Japan, 1993, pp. 1013–1020.

[5] M-H. Han, S-Y. Rhee, Navigation control for a mobile robot, J. Robot. Systems 11 (3) (1994) 169–179.

[6] M.A. Turk, D.G. Morgenthaler, K.D. Gremban, M. Marra, VITS-A vision system for autonomous land vehicle navigation, IEEE Trans. Pattern Anal. Mach. Intell. 10 (3) (1988).

[7] D. Feng, B.H. Krogh, Satisficing feedback strategies for local navigation of autonomous mobile robots, IEEE Trans. Systems, Man Cybernet. 20 (6) (1989).

[8] E. Krotkov, R. Hoffman, Terrain mapping for a walking planetary rover, IEEE Trans. Robot. Automat. 10 (6) (1994) 728–739.

[9] T.J. Olson, D.J. Coombs, Real-time vergence control for binocular robots, Int. J. Comput. Vision 7 (1) (1991) 67–89.

[10] D. Coombs, K. Roberts, Centering behavior using peripheral vision, Proc. Computer Vision and Pattern Recognition, 1993, pp. 440–445.

[11] J. Santos-Victor, G. Sandini, F. Curotto, S. Garibaldi Divergent stereo for robot navigation: learning from bees, Proc. Computer Vision and Pattern Recognition, 1993, pp. 434–439.

[12] M. Romano, N. Ancona, A real-time reflex for autonomous navigation, Proc. Intelligent Vehicles Symposium, 1993, pp. 50–55.

[13] K. Joarder, D. Raviv, Autonomous obstacle avoidance using visual fixation and looming, Proc. SPIE Vol. 1825 Intelligent Robots and Computer Vision XI, 1992, pp. 733–744.

[14] D. Raviv (1992), A quantitative approach to looming, Technical Report, NISTIR 4808, National Institute for Standards and technology, Gaithersburg, MD 20899.

[15] K. Joarder, D. Raviv, A new method to calculate looming for autonomous obstacle avoidance, in Proc. Computer Vision and Pattern Recognition, 1994, pp. 777–779.

[16] A. Broggi, A massively parallel approach to real-time vision-based road markings detection, Proc. Intelligent Vehicles Symposium, Detroit, 1995, pp. 84–89.

[17] P.G. Luebbers, An artificial neural network architecture for interpolation, function approximation, time series modeling and control applications, Ph.D. Dissertation, Electrical Eng. Dept., Florida Atlantic University, Boca Raton, FL, 1993.

[18] H.H. Yakali, Autonomous landing and road following using 2D visual cues, Ph.D. Dissertation, Electrical Eng. Dept., Florida Atlantic University, Boca Raton, FL, 1994.

[19] J.J. Gibson, The Perception of the Visual World, Houghton Mufflin, Boston, 1950.

[20] J.J. Gibson, The Senses Considered as Perceptual Systems, Houghton Mufflin, Boston, 1954.

[21] J.E. Cutting, Perception with an Eye for Motion, The MIT Press, Cambridge, MA, 1986.

[22] I. Rock, Perception, Scientific American Library, New York, 1984.

[23] D.J. Ingle, M.A. Goodale, R.J.W. Mansfield (Eds.), Analysis of Visual Behavior, The MIT Press, Cambridge, MA, 1982.

[24] S.R. Kundur, D. Raviv, An image-based visual threat cue for autonomous navigation, Proc. IEEE International Symposium on Computer Vision, Coral Gables, FL, 1995, pp. 527–532.
[25] S.R. Kundur, D. Raviv, Novel active vision based visual threat cue for autonomous navigation tasks, Comput. Vision Image Understanding 73 (2) (1995) 169–182.
[26] S.R. Kundur, D. Raviv, Novel active-vision-based visual threat cue for autonomous navigation tasks, Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition, San Francisco, 1996, pp. 606–611.
[27] S. Kundur, D. Raviv, E. Kent, An image-based visual-motion-cue for autonomous navigation, Proc. IEEE Intl. Conference on Computer Vision and Pattern Recognition, 17–19 June, San Juan, Puerto Rico, 1997.
[28] S. Kundur, D. Raviv, vision-based fuzzy controllers for autonomous navigation tasks, Proc. IEEE Int. Symp. on Intelligent Vehicles '95, Detroit, 1995.
[29] S.R. Kundur, D. Raviv, A vision-based pragmatic strategy for autonomous navigation, Pattern Recog. 31 (31) (1998) 1221–1239.
[30] P. Brodatz, Textures: a photographic album for artists and designers, Dover Publications, New York, 1966.
[31] E.H. Mamdani, Applications of fuzzy algorithms for control of simple dynamic plant, Proc. IEE. 121 (2) (1974) 1585–1588.
[32] L.A. Zadeh, Fuzzy algorithms, Inform. Control 12 (1968) 94–102.
[33] L.A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, IEEE Trans. Systems, Man, and Cybernet. SMC-3 (1973) 28–44.
[34] C.C. Lee, Fuzzy logic in control systems: fuzzy logic controller - Part I, IEEE Trans. Systems, Man, Cybernet. 2 (2) (1990) 404–418.
[35] C.P. Pappis, E.H. Mamdani, A fuzzy logic controller for a traffic junction, IEEE Trans. Systems, Man, Cybernet. SMC-7 (10) (1977) 707–717.
[36] M. Sugeno, M. Nishida, Fuzzy control of model car, Fuzzy Sets Systems 16 (1985) pp. 103–113.
[37] H.J. Zimmermann, Fuzzy Set Theory and its Applications, Kluwer Academic Publishers, Boston, 1991.
[38] M. Sugeno, M. Nishida, Fuzzy control of model car, Fuzzy Sets Systems 16 (1985) 103–113.
[39] M. Maeda, Y. Maeda, S. Murakami, Fuzzy drive control of an autonomous mobile robot, Fuzzy Sets and Systems 39 (1991) 195–204.
[40] Y. Nagai, N. Enomoto, Fuzzy control of a mobile robot for obstacle avoidance, J. Informat. Sci. 45 (2) (1988) 231–248.
[41] P-S. Lee, L-L. Wang, Collision Avoidance by fuzzy logic control for automated guided vehicle navigation, J. Robot. Systems 11 (8) (1994) 743–760.

About the Author—SRIDHAR R. KUNDUR received the B. Tech. degree in Electrical and Electronics Engineering from Jawahar Lal Nehru Technological University, Hyderabad, India, and the M.S. degree in Electrical Engineering from Southern Illinois University, Edwardsville, Illinois, in 1990 and 1992 respectively, and the Ph.D. degree in Electrical Engineering from Florida Atlantic University, Boca Raton, FL, in 1996. He is currently an applications engineer at Sharp Digital Information Products, Inc., Huntington Beach, CA. His research interests are in image processing, computer vision, and vision-based autonomous navigation.

About the Author—DANIEL RAVIV received the B.Sc. and M.Sc. degrees in electrical engineering from the Technion in Haifa, Israel, in 1980 and 1982, respectively, and the Ph.D. degree in Electrical Engineering and Applied Physics from Case Western Reserve University, Cleveland, Ohio in 1987. He is currently an associate professor of Electrical Engineering at Florida Atlantic University, Boca Raton Florida and a visiting researcher at the National Institute of Standards and Technology (NIST), Gaithersburg, Maryland. His research interests are in vision-based autonomous navigation, computer vision, and inventive thinking.