Texture-Independent Vision-Based Closed-Loop Fuzzy Controllers for Navigation Tasks*

Sridhar R. Kundur[1] and Daniel Raviv[1,2]

[1]Robotics Center and Department of Electrical Engineering
Florida Atlantic University, Boca Raton, FL 33431
[2]Intelligent Systems Division,
National Institute of Standards and Technology(NIST)
Bldg. 220, Room B124, Gaithersburg, MD 20899
email: kundur@acc.fau.edu and ravivd@acc.fau.edu

Abstract

This paper deals with *vision-based closed-loop* control schemes for *collision avoidance* as well as *maintenance of clearance* in *a-priori* unknown textured environments. These control schemes are based on *fuzzy logic* and employ a *visual motion cue*, we call the Visual Threat Cue (VTC) that provides some measure for a *relative change in range* as well as *clearance* between 3D surface and a *fixated* observer in *motion*. It is a collective measure obtained *directly from the raw data of gray level images,* is *independent* of the 3D surface texture and needs *no optical flow information, 3D reconstruction, segmentation, feature tracking or preprocessing*. This motion cue is *scale-independent, rotation independent* and is measured in [time^{-1}] units.

Design of a closed-loop conventional controller for vision based navigation tasks pose a problem as the system is complex and ill-defined. On the other hand fuzzy control which 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. The fuzzy controllers were tested in real time using a 486-based Personal Computer and a camera capable of undergoing 6-DOF motion. Results are highly encouraging.

Key Words: Active Vision, Collision Avoidance, Maintenance of Clearance, Visual Motion

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

1 Introduction

When dealing with a moving camera-based autonomous navigation system, a huge amount of visual data is captured. For vision-based navigation tasks (such as obstacle avoidance, maintaining safe clearance, etc.), relevant visual information needs to be extracted from the visual data and used in real-time closed-loop perception-action control system. In order to accomplish safe visual navigation 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.

Usually the process of driving or flying in a 3D environment involves a human operator. The operator acts in part as a sensory feedback in the perception-action closed-loop control to ensure safe navigation in real time. It becomes a difficult problem to replace the human operator by a vision-based system to achieve similar tasks for the following reasons: In outdoor navigation the environment is usually unknown and unstructured, and the same 3D scene may result in many different images due to changes in illumination conditions, relative distances, orientation of the camera, choice of fixation point, etc., as well as various camera parameters such as zoom, resolution, focus, etc. There is a need for an approach, to obtain relevant visual information about relative proximity in the presence of the above mentioned factors and employ it as a set of sensory feedback signals to accomplish the tasks of safe navigation.

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 [1-

14]). Usually identifying the surrounding object is not important for such tasks, i.e., is it a tree, mountain or another vehicle; what is more important is whether a particular object is an obstacle or not, i.e., is the camera on a collision course with it, is there enough clearance, etc.? For navigation tasks recovering 3D scene and its attributes may not be necessary as it may contain information which is not relevant for the task at hand. Visual cues such as time-to-contact [12] and looming [10, 20, 21] carry important information about the relative proximity. These cues do not need any reconstruction process which is usually computationally intense and in many cases are sufficient for safe navigation.

It is well established in the literature (computer vision as well as psychology) that optical flow plays an important role in the control of human and machine behavior in the environment [15-19]. The extraction of optical flow from a sequence of images is sensitive to noise in images and needs pre-processing like spatio-temporal smoothing [22-24] which may be computationally expensive. Alternatives to optical flow information as sensory feedback for obstacle avoidance include geometrical properties like size, shape, contour and area of image entities, imaged texture [25, 26], focus [28-30], etc.

A differential invariant of the image field based visual information about time-to-collision is presented in [14]. In [26] variations in image statistics are employed to extract the four components of an affine transformation. A qualitative view of the use of these components as sensory feedback information for collision avoidance is also presented [26]. In [27] it is shown that the relative changes in edgels of visible texture in a unit area to be equal to looming described in [10]. This approach of using edge density in an image is an alternative to the use of flow based approach to extract looming which is very sensitive to noise.

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. Research in the area of fuzzy control was initiated by Mamdani's

pioneering work [47], which had been motivated by Zadeh's seminal papers on fuzzy algorithms [48] and linguistic analysis [49.]. In the past few years several researchers have addressed the use of fuzzy control for ill-defined processes for which it is difficult to model the dynamics (see for example [46, 50, 51]).

This paper deals with *vision-based closed-loop* control schemes for *collision avoidance* as well as *maintenance of clearance* in *a-priori unknown* textured environments. These control schemes are based on fuzzy logic and employ a visual motion cue, we call the Visual Threat Cue (VTC) that provides some measure for a *relative change in range* as well as *clearance* between 3D surface and a *fixated* observer in *motion*. It is a collective measure obtained *directly from the raw data of gray level images*, is *independent* of the 3D surface texture and needs no *optical flow information, 3D reconstruction, segmentation, feature tracking or preprocessing*. This motion cue is *scale-independent, rotation independent* and is measured in [time^{-1}] units.

Design of a closed-loop conventional controller for vision based navigation tasks pose a problem as the system is complex and ill-defined. On the other hand fuzzy control which 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. The fuzzy controllers were tested in real time using a 486-based Personal Computer and a camera capable of undergoing 6-DOF motion.

This paper is organized as follows: In section 2 an overview of the VTC is presented, details of the control schemes are presented in section 3, in section 4 we present the experimental results, section 5 presents experimental results and analysis and finally section 6 presents conclusion and an overview of future work.

2 Overview of the Visual Threat Cue (VTC)

We review a visual motion cue called the Visual Threat Cue (VTC) [43] that provides some measure for a *relative change in range* as well as for *specific clearance* between a 3D surface and a *fixated* observer in *motion*. This cue is *independent* of the 3D environment and needs no *a-priori knowledge* about it. It is *time-based*, *rotation independent* and *does not need 3D reconstruction*. This cue can be extracted *directly from the raw gray level data* of images and does not need optical flow information, segmentation, feature tracking and pre-processing. Mathematically the VTC (for $R > R_0$) is defined as follows:

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

where R is the range between the observer and a point on the 3D surface, *d(R)/dt* is the differentiation of R w.r.t. time and R_0 is the desired minimum *clearance* and has the same units as R. Note that the units of the VTC are [time^{-1}]. The VTC has been shown to be *independent* of the rotational motion and can be measured without knowledge about R [43].

The VTC corresponds to a visual field surrounding the moving observer, i.e., there are imaginary 3D surfaces attached to the observer that are moving with it, each of which corresponds to a value of the VTC. 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 value on the minimum clearance sphere of R_0 centered at the location of the observer is the maximum which is infinity, indicating that the absolute distance between the observer and the camera is the minimum clearance. See Figure (1). Note that this field is *not* a sphere.

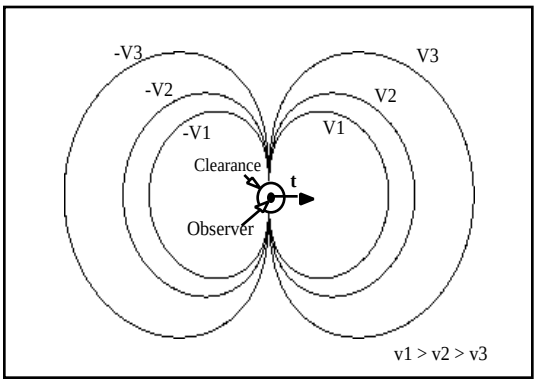

Figure 1: Cross section of the VTC, **t** is the instantaneous translational vector

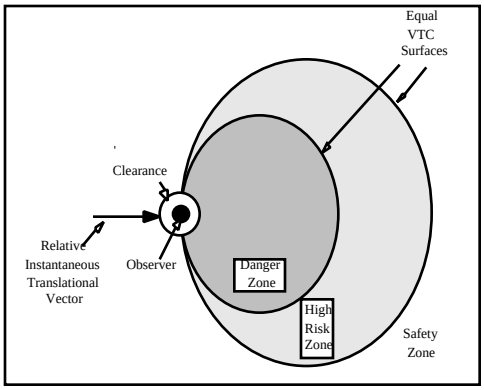

Figure 2: Using the VTC in Navigation. Note the clearance sphere

Based on this knowledge about the cue, one can demarcate the region around an observer into safe, high risk and danger zones (see Figure (2)). This can be used to directly generate control action tasks such as collision avoidance, maintenance of clearance, etc. The VTC is related to, but different from, the time-to-contact and the looming concepts [10, 12, 20, 21].

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 [43]. This approach is *independent* of the 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 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)

Local spatial gray tone variations in an image give rise to a visual pattern in the image known as texture. These spatial gray level variations are due to the visual characteristics of the 3D scene being imaged, the illumination, the range between the scene and the observer, as well as due to camera parameters like zoom, aperture, resolution, focus, etc. When there is a relative motion between a textured surface and a fixated, fixed-focus moving observer, the perceived texture in the 2D image varies. For instance, consider the case of a camera that is initially focused to a 3D surface at a very short distance and gradually moves away from this surface. As a result, the perceived 2D image texture varies from one image to another, mainly due to focus, i.e., the image of the scene in perfect focus is very sharp and has many details, then as the camera moves away from the scene, fine details gradually get smeared and eventually disappear (see Figure (3)). When the image is in perfect focus, the dissimilarity, i.e., spatial gray level variations is very high, and as the details get smeared the dissimilarity gets smaller and smaller. We describe an IQM to measure the dissimilarity of the image. Using the relative temporal variations in this IQM we extract the VTC. Next we present a brief overview of several possible approaches to extract the dissimilarity of images, and describe a practical way to extract the VTC from variations in the IQM.

The area of texture classification has drawn the attention of researchers in the area of computer vision over the past two decades (see for example [31-39]). One of the earliest areas of interest in the texture analysis was texture segmentation and scene analysis. These approaches may be broadly classified as statistical approaches and approaches based on structural properties. The statistical approaches usually employ features that measure the coarseness and the directionality of textures in terms of the averages over a windowed portion of the image. While structural methods on the other hand describe the geometrical properties like size, shape, area, etc. of the objects in the scene. Structural analysis is suitable for structured environments and also needs some a-priori knowledge

about the scene. The statistical methods that are in use are mainly classified into the following categories:

1. Spatial gray level dependence methods [31, 33, 34]

2. Spatial frequency based methods [33, 35]

3. Stochastic model based features [36, 37]

4. Heuristic approaches [38, 39]

The most popular being the spatial gray-level dependence and the stochastic model-based approaches. Usually spatial gray level based approaches are probabilistic, dependent on the number of gray levels in the image and computationally expensive. Stochastic model approaches are dependent on the model of the texture. Among several possible approaches to describe the quality of texture in an image, we extended an approach presented in [40] to describe the dissimilarity of images using IQM. The advantages of using this approach over the other approaches 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 spatial or temporal smoothing.

3. It does not need a model of the texture and is suitable for many textures.

4. It is simple and can be implemented in real time on parallel hardware.

5 It is non-probabilistic and is independent of the number of gray levels used in the image

Mathematically, the IQM is defined as follows [43]:

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

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 Lc and Lr are positive integer constants; and D is a number defined as $D = (2L_c + 1) \times (2L_r + 1) \times (x_f - x_i) \times (y_f - y_i)$.

2.2 Extraction of the VTC from relative variations of IQM

Based on our experimental results [43], we observed that relative temporal changes in the IQM behave in a very similar fashion to the VTC , i.e., $\frac{d(IQM)/dt}{IQM} \quad R_0 \frac{d(R)/dt}{R(R - R_0)}$. The VTC is *independent* of the magnitude of the IQM. The following is a sample set of five images (out of 71) that corresponds to a texture from Brodatz album [42] (see Figure (3)) as seen by a fixating, moving, fixed-focus camera. The graphs of the *measured* IQM and VTC are obtained directly from the images without additional processing. Note that the *measured* IQM (Figure (4)) is a "smooth" function, resulting in good VTC values. Very similar results were reported in [43] for twelve different textures of the same album [42].

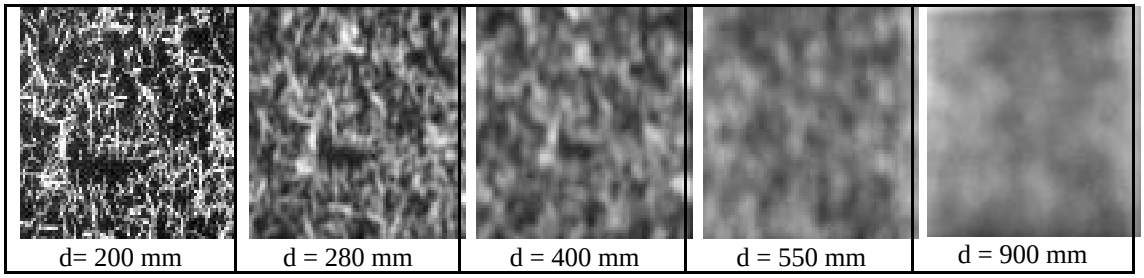

Figure 3: A sample of five images of a textured surface (D110) , d is the relative distance

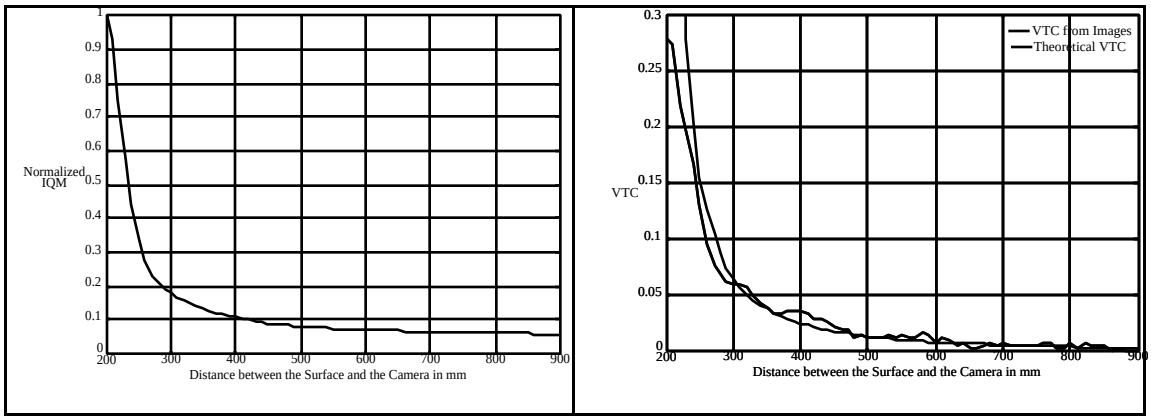

Figure 4: The measured IQM and the VTC for a texture (D110) using 71 images

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 the VTC cannot be employed when the robot is in this region. However the IQM and relative temporal variations in IQM ({d(IQM)/dt}/{IQM}) can be used since it is an image measure and is defined irrespective of the distance between the camera and the surface. In this section we present an overall qualitative behavior of the IQM as well as {d(IQM)/dt}/{IQM}.

Consider the case of a fixed focus camera that is initially focused to a 3D surface at a very short distance say R_0. As the camera moves away from the surface ($R > R_0$) or moves towards the surface ($R<R_0$) the perceived 2D image texture varies from one image to another mainly due to focus. The details in the image get smeared when R is not equal to R_0. As the details get smeared the dissimilarity of the image becomes smaller and smaller. A plot depicting the qualitative behavior of the IQM versus the distance between the camera and the surface is shown in Figure (5). Experimental results to support this qualitative behavior of the IQM which is basically a focus measure can be found in [44, 45]. A qualitative plot of {(d(IQM)/dt)}/{IQM} for a given speed is also shown in Figure (5).

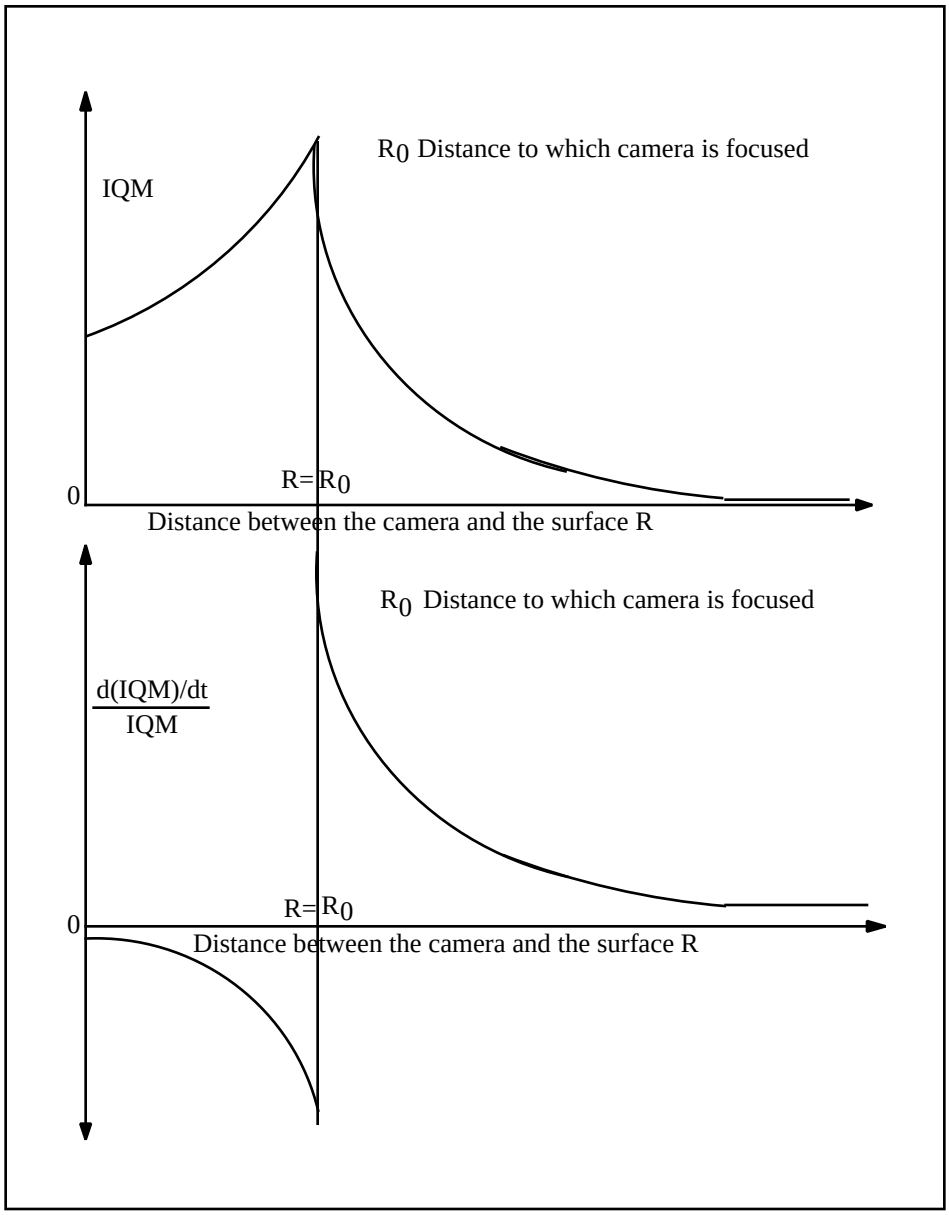

Figure 5: Qualitative IQM and {(d(IQM)/dt)}/{IQM}for a given speed

3 Control Schemes

In this section we describe the vision-based fuzzy control schemes employed to achieve the tasks of collision avoidance and maintenance of clearance using the {(d(IQM)/dt)}/{IQM} as sensory feedback signal.

In this section we present two vision-based fuzzy control schemes to accomplish the following tasks:

Task I: Collision Avoidance: The task is to ***stop*** a robot in motion in front of a *textured surface* when the relative distance between the robot and the surface is a desired one using visual information only.

Task II: Maintenance of Clearance: The task is to ***follow*** a textured surface using visual information only.

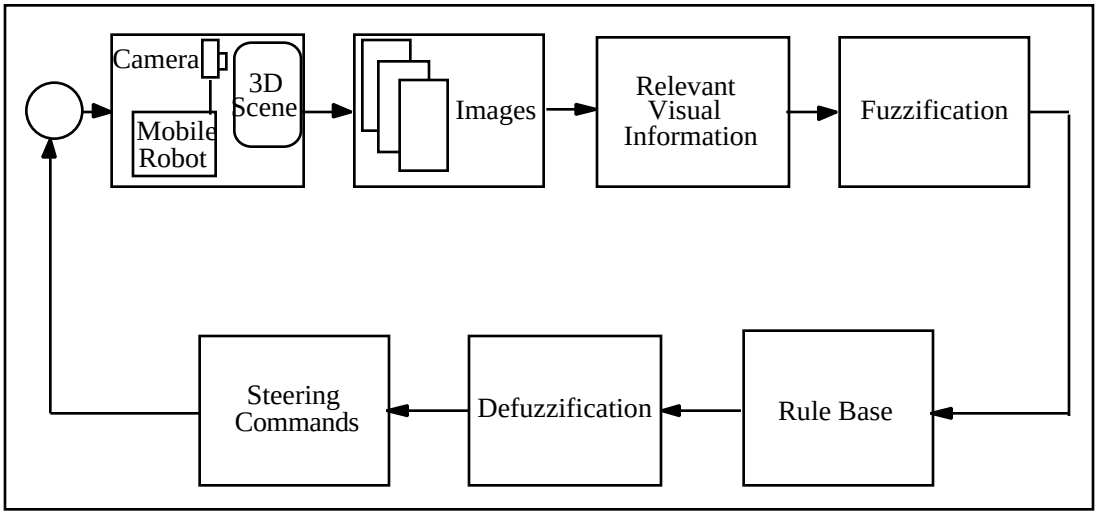

Figure 6: Block diagram of the control scheme

3.1 Control Scheme I: This control scheme has been employed to achieve task I. Initially the camera is focused to distance which is equal to R_0. For ranges R greater than R_0, as the range increases the IQM (denoted as C from now on) value decreases and vice-versa. Based on the relative temporal variations in the IQM value, we divide the region in front of the mobile robot into three different regions as shown in Figure (2). Region I can

be seen as a safe region (refer to Figure(7)) and regions II and III can be seen as danger zones.

The control actions are based on the fuzzy relative temporal changes in IQM and can be described by the following rules, (d(.)/dt indicates differentiation of (.) with respect to time, {d(IQM)/dt}/{IQM} = {d(C)/dt}/{C}), (assume t2 >t1):

Rule I: This rule corresponds to the case when the robot is in the safe zone (region I in Figure (7)). 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 $\left[\frac{d}{dt}\left(\frac{1}{C}\right)\right]_{t=t2} > 0$ **and** $\left[\frac{d}{dt}\left(\frac{1}{C}\right)\right]_{t=t1} > 0$ **and** $\left[\frac{d}{dt}\left(\frac{1}{C}\right)\right]_{t=t2} > \left[\frac{d}{dt}\left(\frac{1}{C}\right)\right]_{t=t1}$ ***Then*** **No Action**

Note: $\left[\frac{d}{dt}\left(\frac{1}{C}\right)\right]_{t=t2} > 0$ **and** $\left[\frac{d}{dt}\left(\frac{1}{C}\right)\right]_{t=t1} > 0$ indicates the motion of the robot towards the surface when the robot is in the region beyond the desired minimum clearance.

Rule II: This rule corresponds to region II in Figure (7). In this region it is required to stop the moving robot if the robot is moving towards the surface and if it crosses the desired clearance. The motion of the robot towards the surface for this region can be expressed in an IF-THEN format as follows.

IF $\left[\frac{d}{dt}\left(\frac{1}{C}\right)\right]_{t=t1} >$ Threshold **and** $\left[\frac{d}{dt}\left(\frac{1}{C}\right)\right]_{t=t2} < 0$ ***Then*** **Stop**

Rule III: This rule corresponds to region III in Figure (7). In this region the robot is required to stop if it is moving towards the surface. Note that in this region $\dot{r} < 0$. The motion of the robot towards the surface for this region can be expressed in the IF-THEN format as follows:

If $\dot{r}_{t=t2} <$ 0 **and** $\dot{r}_{t=t1} <$ 0 **and** $\dot{r}_{t=t1} > \dot{r}_{t=t2}$ ***Then*** **Stop.**

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

Usually this situation arises when the robot is either stationary or moving away from the surface.

All four rules for the control scheme I can be summarized as follows:

Rule		Sensing		Action
1	IF	$\dot{r}_{t=t2} >$ 0 **and** $\dot{r}_{t=t1} >$ 0 **and** $\dot{r}_{t=t2} > \dot{r}_{t=t1}$ **(t2 > t1)**	THEN	No Action
2	IF	$\dot{r}_{t=t1} >$ Threshold **and** $\dot{r}_{t=t2} <$ 0 **(t2 > t1)**	THEN	Stop
3	IF	$\dot{r}_{t=t2} <$ 0 **and** $\dot{r}_{t=t1} <$ 0 **and** $\dot{r}_{t=t1} > \dot{r}_{t=t2}$ **(t2 > t1)**	THEN	Stop
4	IF	ELSE	THEN	No Action

Table 1: Summary of rules for vision-based collision avoidance

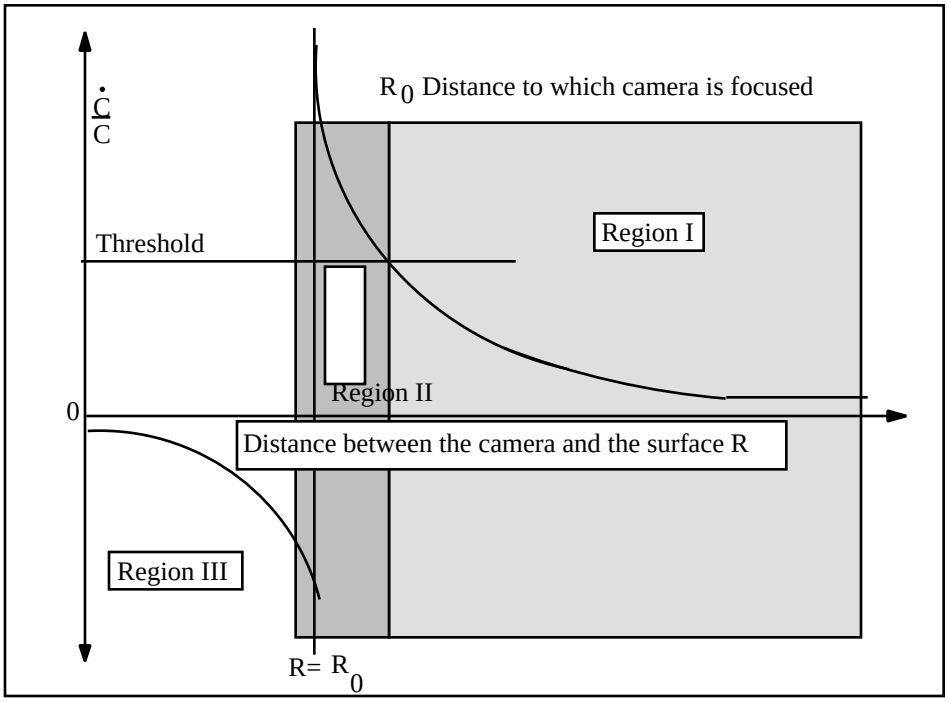

Figure 7: Qualitative plot of the relative variations of IQM {d(C)/dt}/{C}

3.2 Control Scheme II: This control scheme has been employed to accomplish task II, i.e., maintenance of clearance. Refer to Figures (8a) and (8b).

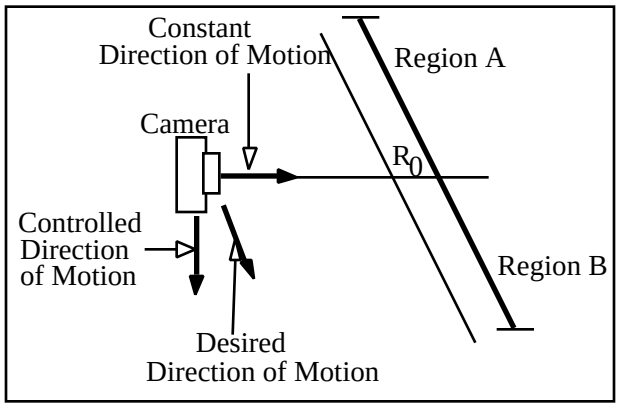

Figure 8a: Region A corresponds to the left of the camera and region B corresponds to the right of the camera

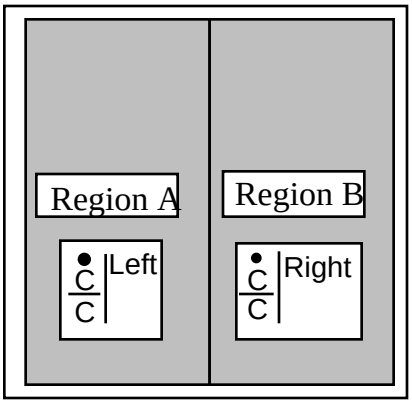

Figure 8b: The image corresponding to Figure (8a).

In Figure (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. As mentioned earlier, when the distance between the camera and the surface is greater than the desired minimum clearance, the points located at a greater distance have a relatively smaller relative temporal variations in the IQM, i.e., {d(C)/dt}/{C} than those located at a relatively smaller distance. The computer compares the {d(C)/dt}/{C} of the left window as shown in Figure (8b) with the {d(C)/dt}/{C} of the right window This is the key to generate steering commands to the robot, i.e., the difference between $(\frac{\dot{C}}{C})$ corresponding to the left window (denoted as $\left.\frac{\dot{C}}{C}\right|^{Left}$) and $(\frac{\dot{C}}{C})$ corresponding to the right window (denoted as $\left.\frac{\dot{C}}{C}\right|^{Right}$) is the control signal for control scheme II. Note that a change in the direction of motion of the robot can be accomplished by changing forward, lateral or both components of velocity (as shown in Figure (8a)). When the robot is in a region beyond the desired minimum clearance the change in heading direction is generated by controlling only the lateral component. When the robot is in the clearance zone both

lateral as well as forward velocity components are controlled. The control scheme can be explained by IF-THEN rules as follows (assume t2 > t1):

Rule I: ***IF*** $\left.\frac{\dot{C}}{C}\right|_{t=t2}^{Left} > \left.\frac{\dot{C}}{C}\right|_{t=t1}^{Left}$ **and** $\left.\frac{\dot{C}}{C}\right|_{t=t2}^{Right} > \left.\frac{\dot{C}}{C}\right|_{t=t1}^{Right}$ **and** $\left(\left.\frac{\dot{C}}{C}\right|_{t=t2}^{Left} - \left.\frac{\dot{C}}{C}\right|_{t=t2}^{Right}\right)$ is **approximately** zero ***THEN*** no change in the velocity.

According to this rule when the robot approaches the surface and if both the left and right regions are very far from the robot then there is no change in the current velocity. Region A and region B are beyond the desired minimum clearance.

Rule II: ***IF*** $\left.\frac{\dot{C}}{C}\right|_{t=t2}^{Left} > \left.\frac{\dot{C}}{C}\right|_{t=t1}^{Left}$ and $\left.\frac{\dot{C}}{C}\right|_{t=t2}^{Right} > \left.\frac{\dot{C}}{C}\right|_{t=t1}^{Right}$ and $\left(\left.\frac{\dot{C}}{C}\right|_{t=t2}^{Left} - \left.\frac{\dot{C}}{C}\right|_{t=t2}^{Right}\right)$ is **small** ***THEN*** motion to the right is **small.**

Rule III: ***IF*** $\left.\frac{\dot{C}}{C}\right|_{t=t2}^{Left} > \left.\frac{\dot{C}}{C}\right|_{t=t1}^{Left}$ and $\left.\frac{\dot{C}}{C}\right|_{t=t2}^{Right} > \left.\frac{\dot{C}}{C}\right|_{t=t1}^{Right}$ and $\left(\left.\frac{\dot{C}}{C}\right|_{t=t2}^{Left} - \left.\frac{\dot{C}}{C}\right|_{t=t2}^{Right}\right)$ is **medium** ***THEN*** motion to the right is **medium.**

Rule IV: ***IF*** $\left.\frac{\dot{C}}{C}\right|_{t=t2}^{Left} > \left.\frac{\dot{C}}{C}\right|_{t=t1}^{Left}$ and $\left.\frac{\dot{C}}{C}\right|_{t=t2}^{Right} > \left.\frac{\dot{C}}{C}\right|_{t=t1}^{Right}$ and $\left(\left.\frac{\dot{C}}{C}\right|_{t=t2}^{Left} - \left.\frac{\dot{C}}{C}\right|_{t=t2}^{Right}\right)$ is **big** ***THEN*** motion to the right is **big.**

Rule V: ***IF*** $\left.\frac{\dot{C}}{C}\right|_{t=t2}^{Left} < \left.\frac{\dot{C}}{C}\right|_{t=t1}^{Left}$ and $\left.\frac{\dot{C}}{C}\right|_{t=t2}^{Right} > \left.\frac{\dot{C}}{C}\right|_{t=t1}^{Right}$ ***THEN*** motion to the right is **big** and **reverse** the current direction of motion.

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.

Rule VI: ***IF*** $\left.\frac{\dot{C}}{C}\right|_{t=t2}^{Left} < \left.\frac{\dot{C}}{C}\right|_{t=t1}^{Left}$ and $\left.\frac{\dot{C}}{C}\right|_{t=t2}^{Right} < \left.\frac{\dot{C}}{C}\right|_{t=t1}^{Right}$ ***THEN*** motion to the right is **big** and **reverse** the current direction of motion.

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.

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

Usually this situation arises when the robot is either stationary or moving away from the surface.

3.3 Membership Functions

In this section we present a qualitative view of the membership functions employed for fuzzification.

3.3a Control Scheme I: Since only two modes of operation are necessary in this control scheme, no membership functions are employed in this control scheme.

3.3b Control Scheme II: In this the section the membership functions used in the Control Scheme II are described. We employ linear membership functions as shown in Figure (9).

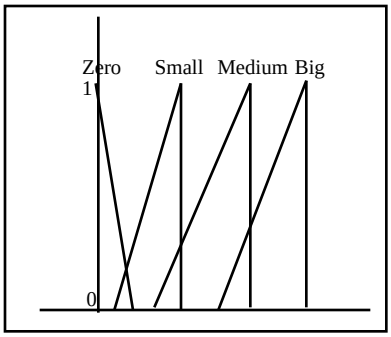

Figure 9: Qualitative Membership functions used

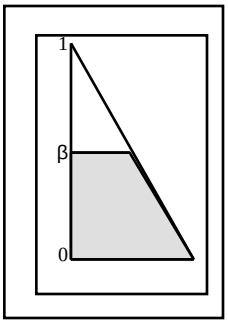

Figure 10: Evaluation of weight of a particular rule

3.4 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, which is stated 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.

We used the ratio of the shaded area in Figure (10) to the area of the triangle as the firing strength and is derived to be:

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

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

4 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. 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 Figure (11a). Photo copies of texture pattern (D5, D110 from [42], see Figure (12)) pasted on a flat surface is presented as the obstacle along the path of the robot (see Figure (11b)). For both control schemes the camera is initially focused to the desired minimum clearance (R_0 = 200 mm.). We use qualitative measures for fuzzy sensing and action (small, medium, big, etc.) rather than the exact speeds.

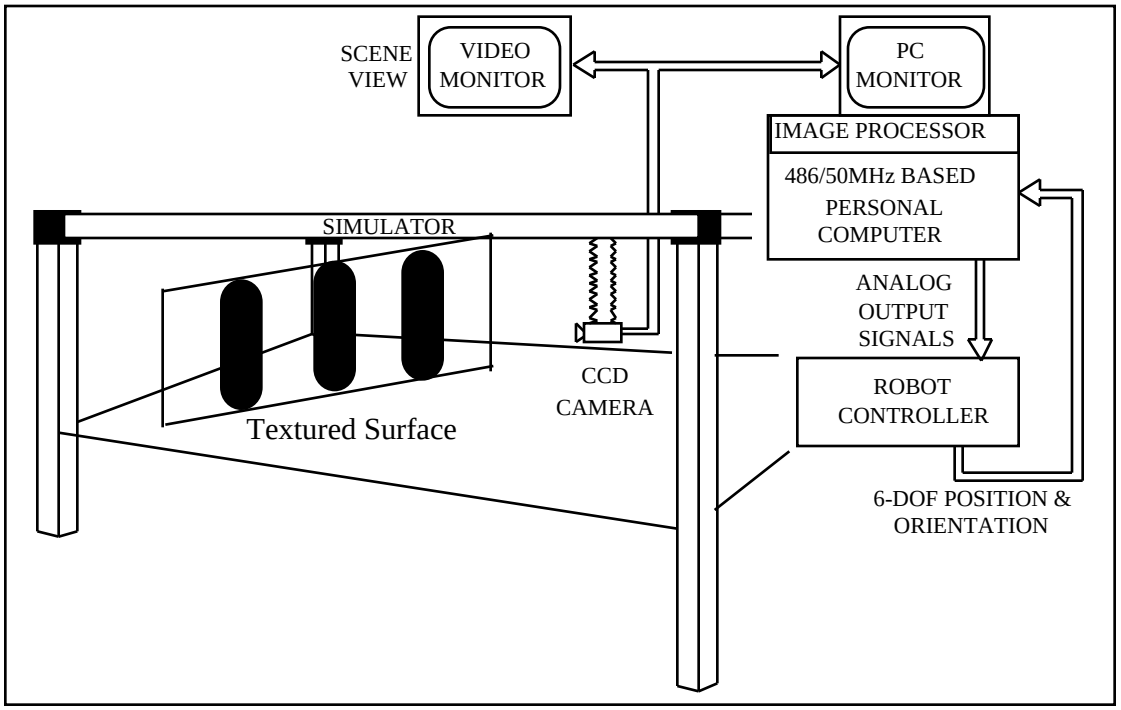

Figure 11a: Block diagram of implementation

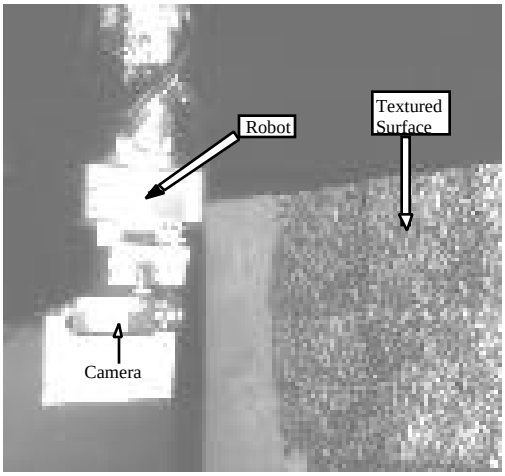

Figure 11b: Camera mounted on the Robot

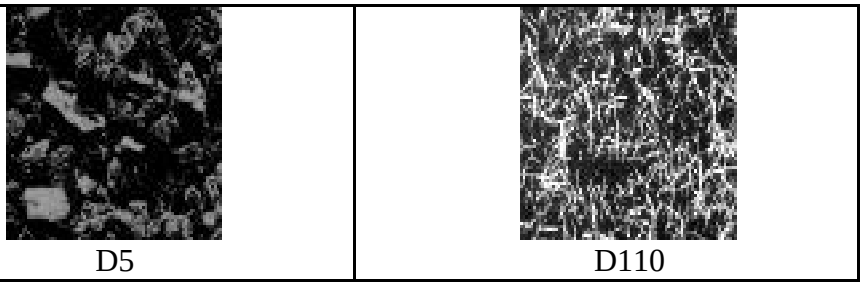

Figure 12: Textures used in the experiments

4.1 Control Scheme I

A window of size 50 X 50 pixels is chosen in the center of the image to evaluate the visual feedback signal {d(C)/dt}/{C}. According to the rules presented in the previous section the crisp control action is (either move or stop) is generated. Two different speeds were employed in this control scheme (speed2 > speed1).

4.2 Control Scheme II

Two windows (left and right) each 50 X 50 pixels are opened in the image. In each of these windows, the visual parameter {d(C)/dt}/{C} 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.

5 Results and Analysis

In this section we present the results and analysis of the control schemes are presented.

5.1 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 (see Figure (13)). The results can be summarized as follows:

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.

Table 2: Summary of vision-based collision avoidance results

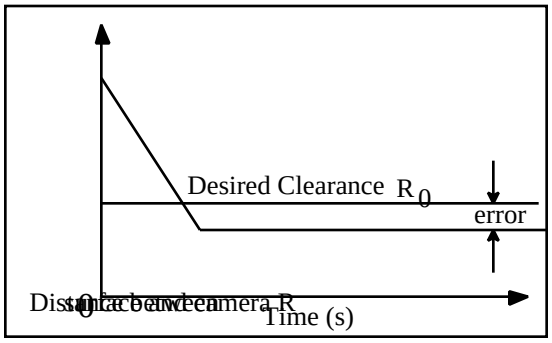

Figure 13: Control scheme I

5.2 Control Scheme II

The lateral and longitudinal components of the heading vector were recorded. The resultant was plotted manually (see Figures (14-21)). 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.

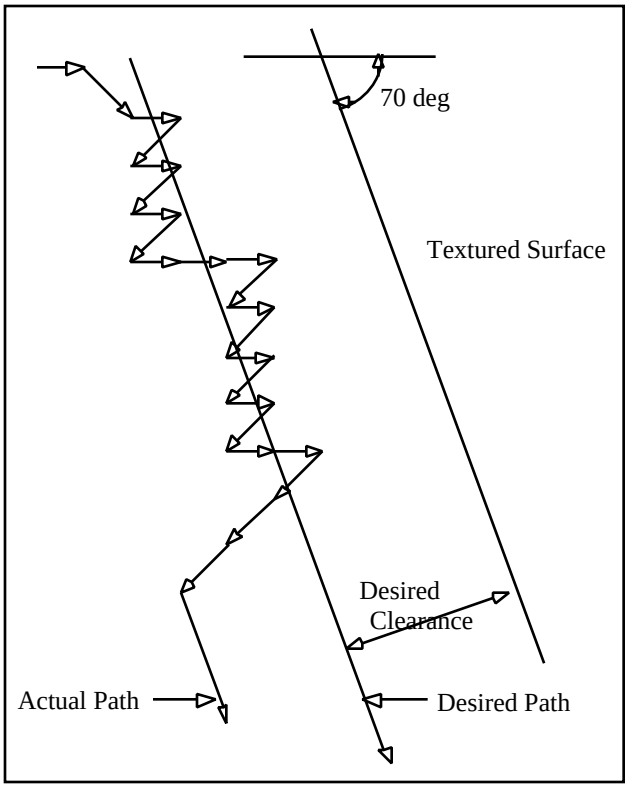

Figure 14: Maintenance of Clearance: Actual Data **for** Texture D110

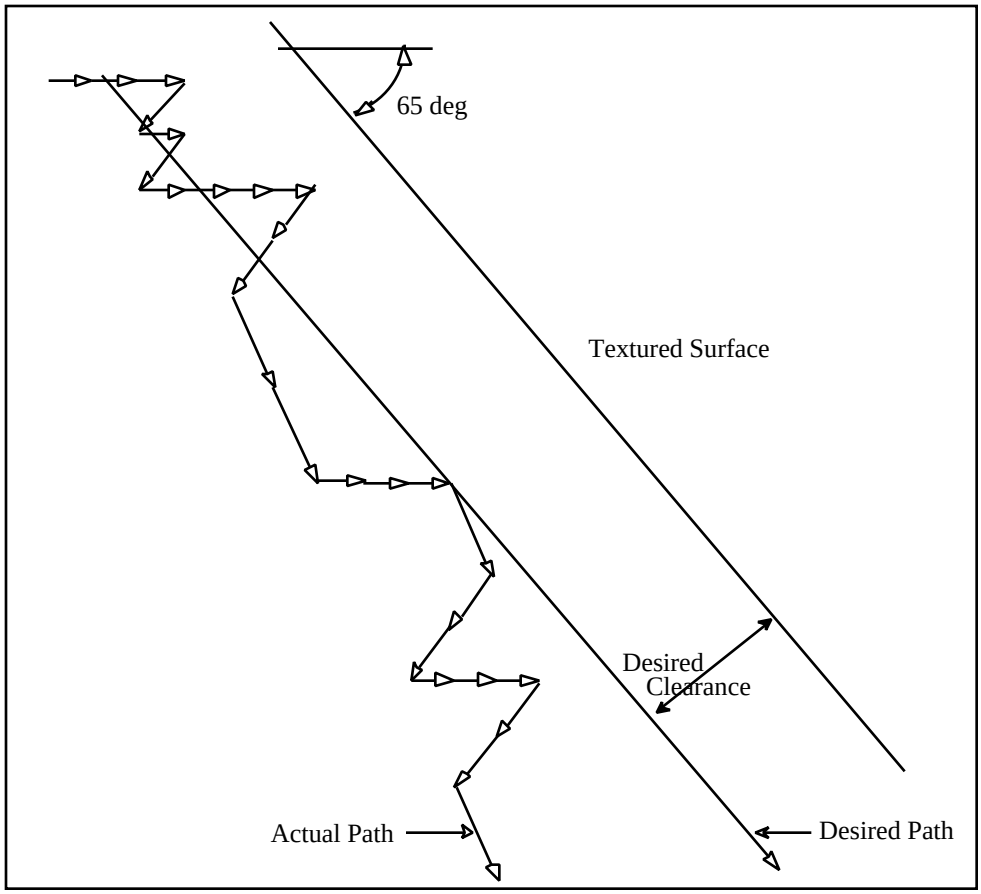

Figure 15: Maintenance of Clearance: Actual Data for Texture D110

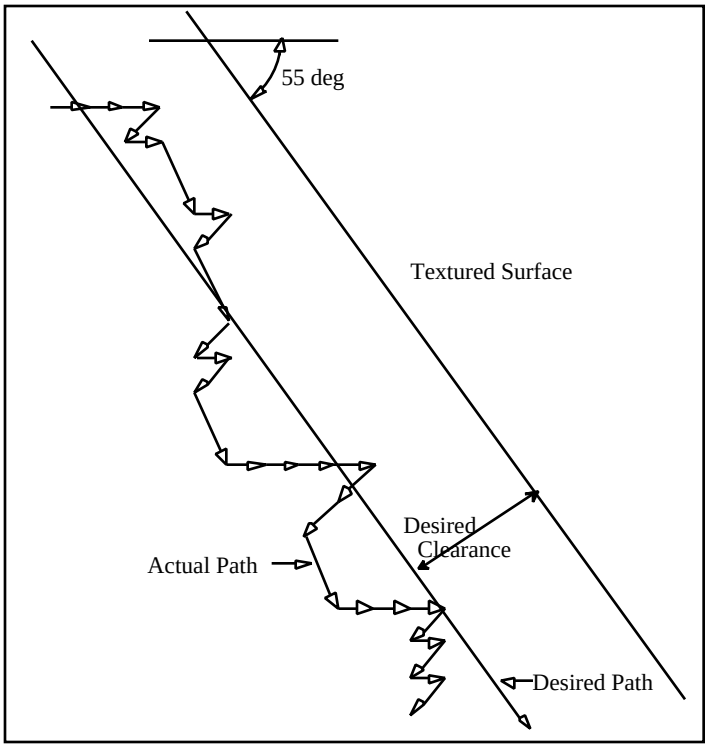

Figure 16: Maintenance of Clearance: Actual Data for Texture D110

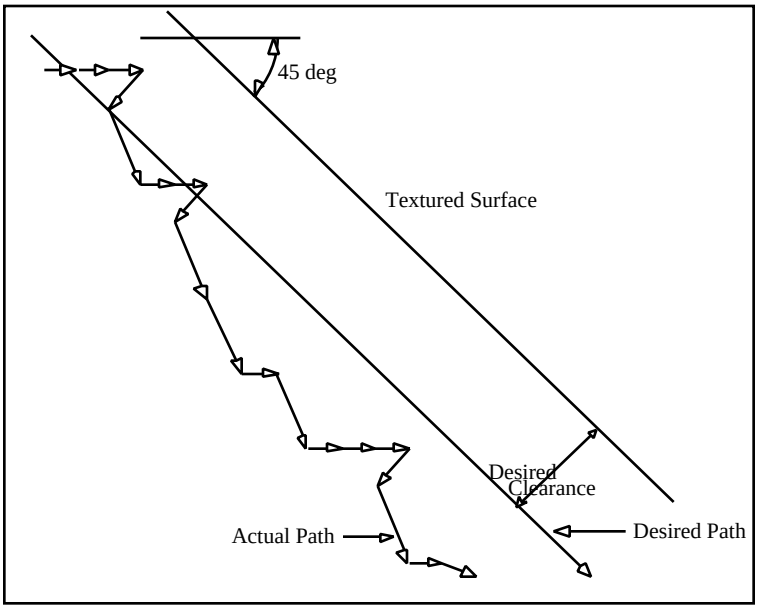

Figure 17: Maintenance of Clearance: Actual Data for Texture D110

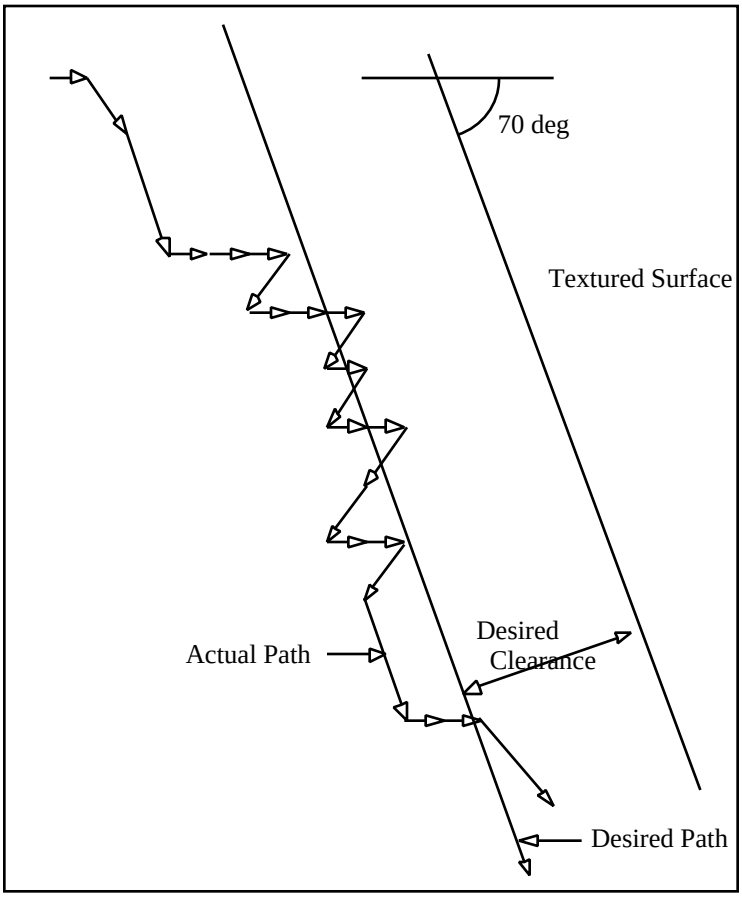

Figure 18: Maintenance of Clearance: Actual Data for Texture D5

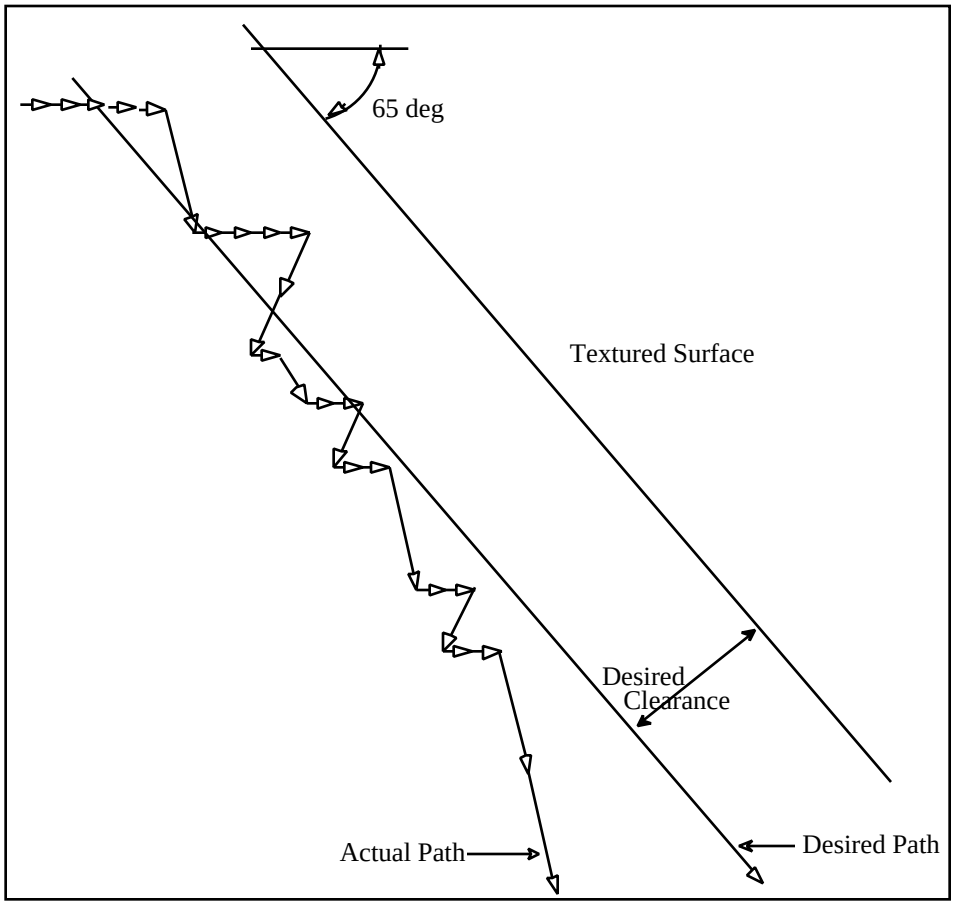

Figure 19: Maintenance of Clearance: Actual Data for Texture D5

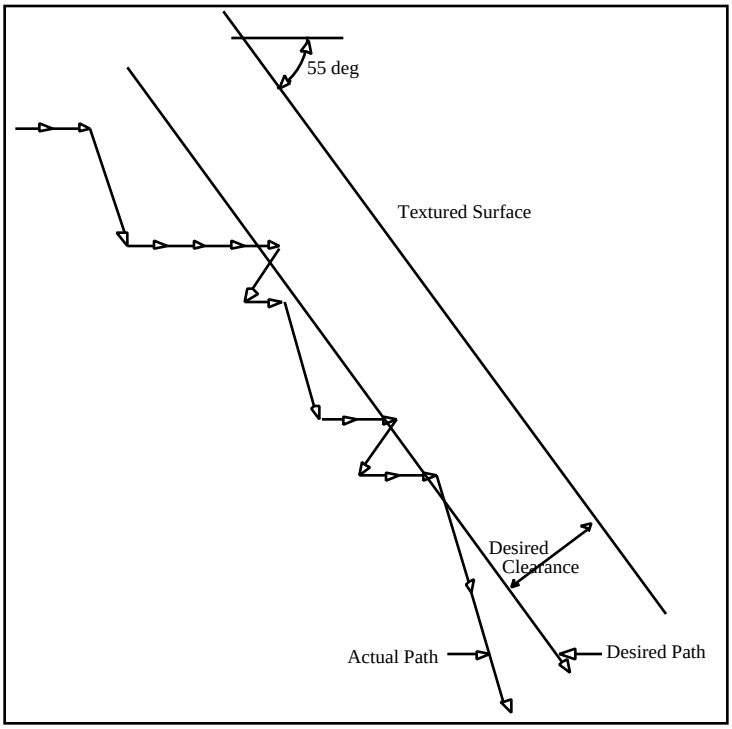

Figure 20: Maintenance of Clearance: Actual Data for Texture D5

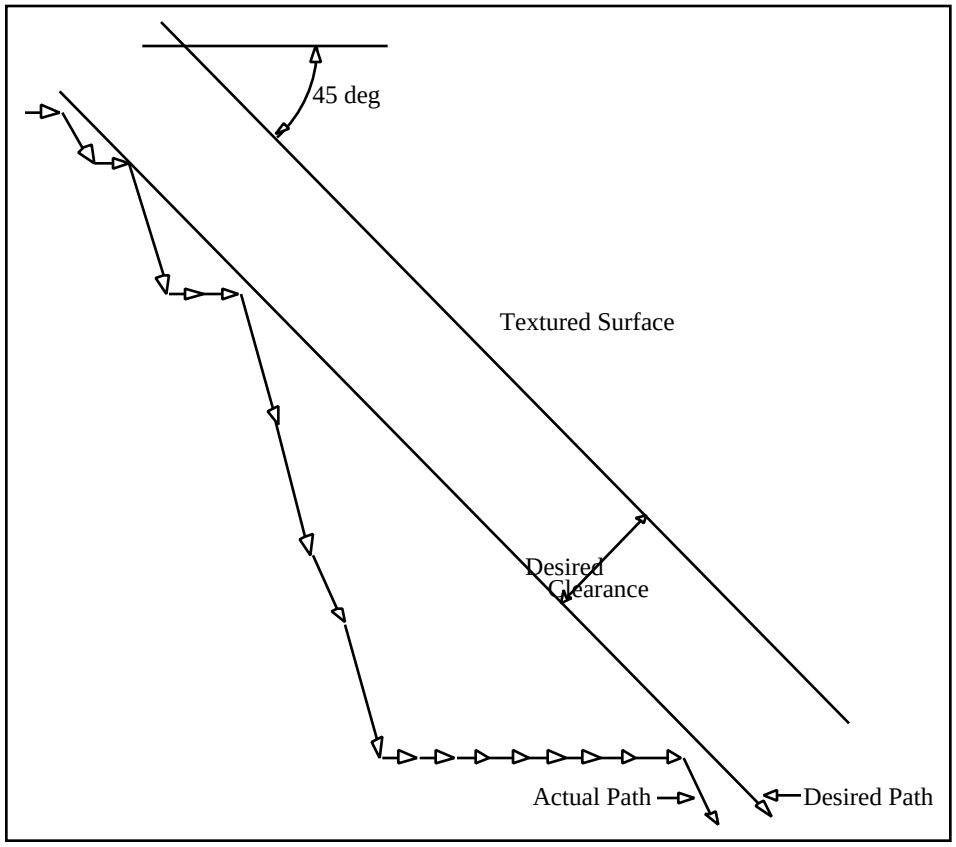

Figure 21: Maintenance of Clearance: Actual Data for Texture D5

6 Conclusions and Future Work

This paper presented two vision-based fuzzy logic control schemes using relative temporal variations in the image dissimilarity for the tasks of collision avoidance and maintenance of clearance. The use of relative temporal variations of image dissimilarity needs no optical flow information, segmentation or feature tracking. These control scheme demonstrate the use of vision as a sensory feedback signal in unknown environments for two navigation tasks. The control schemes are approximate and do not take into account the robot dynamics. Currently we are developing a scheme that takes into account the temporal sampling and robot dynamics. This paper deals with linear membership functions only. Several other monotonic functions are currently being employed. It is also possible to implement classical control techniques like PD, PID control and use a fuzzy logic controller as a supervisor to monitor the PD, PID control schemes.

References

[1] C. Thorpe, M. H. Hebert, T. Kanade and S. A. Shafer (1988), "Vision and Navigation for the Carnegie Mellon Navlab", *IEEE Transactions of Pattern Analysis and Machine Intelligence*, PAMI-10, No. 3, pp. 362-373.

[2] E. D. Dickmanns, B. Mysliwetz and T. Christians (1990), "An Integrated Spatio-Temporal Approach to Automatic Visual Guidance of Autonomous Vehicles", *IEEE Transactions on Systems, Man, and Cybernetics*", Vol. 20, No. 6, pp. 1273-1284.

[3] D. Coombs and K. Roberts (1992), ""Bee-Bot":Using Optical Flow to Avoid Obstacles", *Proc. SPIE Conf. on Intelligent Robots and Computer Vision XI: Algorithms, Techniques and Active Vision*, Boston, MA., pp. 714-721.

[4] A. M. Waxman, LeMoigne and B. Srinivasan (1987), "A Visual Navigation system for Autonomous Land Vehicles", *IEEE Journal of Robotics Automation*, RA-3, No. 2, pp. 124-141.

[5] M. Turk, D. K. Morgenthaler, D. K. Gremban and M. Marra (1988), "A Vision system for Autonomous Land Vehicle Navigation", *IEEE Transactions of Pattern Analysis and Machine Intelligence*, PAMI-10, No. 3, pp. 342-360.

[6] Young Gin-Shu, Hong Tsai-Hong, M. Herman and J. C. S. Yang (1992), "Obstacle Detection and Terrain Charecterization using Optical Flow without 3D Reconstruction", Proc. *SPIE*. Vol. 1825, Intelligent Robots and Computer Vision XI, pp. 561-568.

[7] R. C. Nelson and J. Aloimonos (1989), "Obstacle Avoidance using Flow Field Divergence", IEEE *Transactions of Pattern Analysis and Machine Intelligence*, PAMI-11, No. 10., pp. 1102-1106.

[8] J. Santos-Victor, G. Sandini, F. Curotto, and Gabribaldi (1993), "Divergent Stereo for Robot Navigation: Learning from Bees", Proc. of *CVPR-93*, pp. 434-439.

[9] M. Tistarelli and G. Sandini (1990), "On Advantages of Polar and Log-Polar Mapping for Direct Estimation of Time-To-Contact from Optical Flow", Technical report *LIRA-TR3/90*, University of Genoa, Italy.

[10] D. Raviv (1992), "A Quantitative Approach to Looming", Technical Report *NISTIR 4808*, National Institute of Standards and Techology, Gaithersburg, Maryland.

[11] M. Brady and H. Wang (1992), "Vision for Mobile Robots", *Philosophical Transactions of Royal Society*, London, B337, pp. 341-350.

[12] D. N. Lee (1976), "A Theory of Visual Control of Braking Based on Information About time-to-collision", *Perception*, Vol. 5, 437-459.

[13] D. Raviv (1992), "Visual Looming", Proc. of *SPIE* Conference on Intelligent and Computer Vision XI: Algorithms, Techniques and Active Vision, Boston, MA.

[14] R. Cipolla and A. Blake (1992), "Surface Orientation and Time-to-Contact from Image Divergence and Deformation", In Proc. of *European Conference on Computer Vision*, pp. 187-202.

[15] J. R. Lishman (1981), "Vision and the Optic Flow Field", *Nature*, Vol. 293, pp. 263-264.

[16] J. J. Gibson (1979), "*The Ecological Approach to Visual Perception*", Houghton Mifflin, Boston.

[17] J. J. Gibson (1950a), "*The Perception of Visual World*", Houghton Mifflin, Boston.

[18] J. J. Gibson (1950b), "The Perception of Visual Surfaces", *American Journal of Psychology*, Vol. 63, pp. 367-384.

[19] D. N. Lee (1980), "The Optic Flow Field: The Foundation of Vision", *Philosophical Transactions of Royal Society of London*, B290, pp. 169-179.

[20] T. G. R.Bower, J. M. Broughton and M. K. Moore (1970), "The Coordination of visual and tactual input in infants", *Perception and Psychophysics*, Vol. 8, pp. 51-53.

[21] W. Schiff, J. A. Caviness and J. J. Gibson (1962), "Persistent Fear Responses in Rhesus Monkeys to the Optical Stimulus of "Looming", *Science*, Vol. 136, pp. 982-983.

[22] H. H. Nagel (1987), "On the Estimation of Optical Flow: Relations between Different Approaches and Some New results", *Artificial Intelligence*, Vol. 33, pp. 299-324.

[23] A. Singh (1991), "*Optic Flow Computation: A Unified Perspective*", IEEE Computer Society Press.

[24] J. L Barron, D. J. Fleet and S. S. Beauchemin (1992), "Performance of Optical Flow Techniques", *RPL-TR-9107*, Dept. of Computing and Information Science, Queens University, Canada.

[25] R. Bajcy and L. Liberman (1976), "Texture Gradients as Depth Cue", *Computer Graphics and Image Processing*, Vol. 5, pp. 52-67.

[26] J. Sato and R. Cipolla (1994), "Extracting the Affine Transform from Texture Moments", *TR-167*, Dept. of Engineering, Univ. of Cambridge, England.

[27] K. Joarder and D. Raviv (1994), "A New Method to Calculate Looming from Surface Texture", In Proc. of *CVPR-94*, Seattle, WA., pp. 777-780.

[28] M. Subbarao and N. Gurumoorthy (1988), "Depth Recovery from Blurred Edges", In Proc. of *CVPR-88*, pp. 498-503.

[29] S. K. Nayar (1992), "Shape from Focus System", In Proc. of *CVPR-92*, pp. 302-308.

[30] A. Pentland (1987), "A New Sense for Depth of Field", IEEE *Transactions of Pattern Analysis and Machine Intelligence*, PAMI-9, pp. 523-531.

[31] R. M. Haralick, K. Shanmugam, and I. Dinstein (1973), "Textural Features for Image Classification", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-3, No. 6, pp. 610-621.

[32] R. W. Conners and C. A. Harlow (1980), "A Theoretical Comparison of Texture Algorithms", *IEEE Transactions on Pattern Recognition and Machine Analysis*, Vol. PAMI-2, No. 3, pp. 204-222.

[33] J. S. Weszka, Dyer C. R. and A. Rosenfeld (1976), "A Comparative Study of Texture Measures for Terrain Classification", *IEEE Transactions on Systems, Man and Cybernetics*, SMC-6, pp. 269-285.

[34] M. Unser (1986), "Sum and Difference Histograms for Texture Classification", *IEEE Transactions on Pattern Recognition and Machine Analysis*, Vol. PAMI-8, pp. 118-125.

[35] W. D. Stromberg and T. G. Farr (1986), "A Fourier Based Textural Feature Extraction procedure", *IEEE Transactions on Geoscience Remote Sensing*, GE-24, pp. 722-731.

[36] M. Hassner and J. Sklansky (1980), "The use of Markov Random Fields as models of Texture", *Computer Graphics Image Processing*, Vol. 12, pp. 357-370.

[37] R. L. Kashyap (1984), "Charecterization and Estimation of two Dimensional ARMA models", *IEEE Transactions on Infromation Theory*, Vol. 30, pp. 736-745.

[38] M. M. Galloway (1975), "Texture Analysis using Grey level run lengths", *Computer Graphics Image Processing*, Vol. 4, 1975, pp. 172-179.

[39] O. R. Mitchell, C. R. Myers and W. Boyne (1977), "A Min-Max Measure for Image Texture Analysis", *IEEE Transactions on Computers*, C-26, pp. 400-414.

[40] C. M. Wu and Y. C. Chen (1992), "Statistical Feature Matrix," *Computer Vision Graphics and Image Processing: Graphical Models and Image Processing*, Vol. 54, No. 5,pp. 407-419.

[41] L. Kaufman and P. J. Rousseeuw (1990), "*Finding Groups in Data: An Introduction to Cluster Analysis*," John Wiley & Sons Inc., New York.

[42] P. Brodatz (1966), "*Textures: A Photographic Album for Artists and Designers*", Dover Publications, New York.

[43] S. R. Kundur and D. Raviv (1994), "An Image-Based Texture-Independent Visual Motion Cue for Autonomous Navigation", Technical Report *NISTIR 5567*, National Institute of Standards and Techology, Gaithersburg, Maryland.

[44] R. A. Jarvis (1976), "Focus Optimization Criteria for Computer Image Processing", *Microscope*, Vol. 24, #2, pp. 163-180.

[45] E. Krotkov (1987), "Focusing", *International Journal of Computer Vision*, Vol. 1, pp. 223-237.

[46] C. C. Lee (1990), "Fuzzy Logic in Control Systems: Fuzzy Logic Controller - Part I", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 2, No. 2, pp. 404-418.

[47] E. H. Mamdani (1974), "Applications of Fuzzy algorithms for Control of Simple Dynamic Plant", Proc. *IEE*. Vol. 121, No. 2, pp. 1585-88.

[48] L. A. Zadeh (1968), "Fuzzy Algorithms", *Information and Control*, Vol. 12, pp. 94-102.

[49] L. A. Zadeh (1973), "Outline of a new approach to the analysis of complex systems and decision processes", *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-3, pp. 28-44.

[50] C. P. Pappis and E. H. Mamdani (1977), "A Fuzzy Logic Controller for a Traffic Junction", *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-7, No. 10, pp. 707-717.

[51] M. Sugeno and M. Nishida (1985), "Fuzzy Control of Model Car", *Fuzzy Sets Sytems*, Vol. 16, pp. 103-113.