Vision Based Obstacle Avoidance Techniques

Mehmet Serdar Guzel and Robert Bicker
Newcastle University,
United Kingdom

1. Introduction

Vision is one of the most powerful and popular sensing method used for autonomous navigation. Compared with other on-board sensing techniques, vision based approaches to navigation continue to demand a lot of attention from the mobile robot research community. This is largely due to its ability to provide detailed information about the environment, which may not be available using combinations of other types of sensors. One of the key research problems in mobile robot navigation is the focus on obstacle avoidance methods. In order to cope this problem, most autonomous navigation systems rely on range data for obstacle detection. Ultrasonic sensors, laser rangefinders and stereo vision techniques are widely used for estimating the range data. However all of these have drawbacks. Ultrasonic sensors suffer from poor angular resolution. Laser range finders and stereo vision systems are quite expensive, and computational complexity of the stereo vision systems is another key challenge (Saitoh et al., 2009). In addition to their individual shortcomings, Range sensors are also unable to distinguish between different types of ground surfaces, such as they are not capable of differentiating between the sidewalk pavement and adjacent flat grassy areas. The computational complexity of the avoidance algorithms and the cost of the sensors are the most critical aspects for real time applications. Monocular vision based systems avoid these problems and are able to provide appropriate solution to the obstacle avoidance problem. There are two fundamental groups of vision based obstacle avoidance techniques; those that compute the apparent motion, and those that rely on the appearance of individual pixels for monocular vision based obstacle avoidance systems. First group is called as Optical flow based techniques, and the main idea behind this technique is to control the robot using optical flow, from which heading of the observer and time-to-contact values are obtained (Guzel & Bicker, 2010). One way of the control using these values is by acting to achieve a certain type of flow. For instance, to maintain ambient orientation, the type of Optic flow required is no flow at all. If some flow is detected, then the robot should change the forces produced by its effectors so as to minimize this flow, based on Law of Control (Contreras, 2007).

A second group is called Appearance Based methods rely on basic image processing techniques, and consist of detecting pixels different in appearance than that of the ground and classifying them as obstacles. The algorithm performs in real-time, provides a high-resolution obstacle image, and operates in a variety of environments (DeSouza & Kak, 2002). The main advantages of these two conventional methods are their ease of implementation and high availability for real time applications. However optical flow *based* methods suffer from two major problems, which are the illumination problem that varies with time and the

problem of motion discontinuities induced by objects moving with respect to other objects or the background (Contreras, 2007). Various integrated methods for solving these problems have been proposed; nevertheless it is still a key challenge to employ optical flow for mobile robot navigation. Furthermore, appearance based methods also suffer from illumination problems and, are highly sensitive to floor stains, as well as to the physical structure of the terrain.

Consequently, while having significant performance advantages, there are certain drawbacks which restrict the applicability of these methods. In order to solve those challenges, a novel obstacle avoidance method is introduced in this chapter. The method is principally designed to fuse a Scale invariant features transform (SIFT) algorithm (Lowe, 1999), and template matching with a convolution mask technique, using a *Fuzzy Logic approach*. As opposed to the *Appearance based methods*, previously mentioned, an occupancy map of the environment is generated with respect to the local features and a template. The experimental results reveal that the proposed obstacle avoidance technique allows the robot to move efficiently within its environment and to successfully attain its local goals.

This chapter is organized as follows. In Section 2, the background knowledge to the conventional methods is briefly introduced. In Section 3, new technique is introduced. Section 4 provides the implementation of the behaviour-based robot and the experiment results from both the real and simulation experiments. Section 5 provides a summary of the work.

2. Background

In this section, optical flow based navigation techniques will first be outlined, followed by a brief introduction of the appearance based methods.

2.1 Optical flow

Optical flow, illustrated in Fig. 1, is an approximation to the motion field, summarizing the temporal change in an image sequence. The main idea behind the technique assumes that for a given scene point, the corresponding image point intensity I remain constant over time, which is referred as *conservation of image intensity* (Atcheson et al., 2009). Therefore, if two consecutive images have been obtained at the following time intervals, the basic idea is to detect the motion using image differencing. If any scene point projects onto image point (x, y) at time t and onto image point $(x + \delta x, y + \delta y)$ at time $(t + \delta t)$, the following equation is inferred based on the conservative of image intensity assumption.

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \tag{1}$$

Expanding the right-hand side of the Eq. 1 using a Taylor series about (x, y, t), and ignoring the higher order terms then by rearrangement gives the following expression.

$$\delta x \frac{\partial I}{\partial x} + \delta y \frac{\partial I}{\partial y} + \delta t \frac{\partial I}{\partial t} = 0 \tag{2}$$

A simpler expression, is obtained by dividing by δt throughout and movement along the horizontal $(\frac{\delta x}{\delta t})$, and vertical $(\frac{\delta y}{\delta t})$ directions are u and v respectively. Having these rearrangements and denoting partial derivatives of I by I_x, I_y and I_t gives the differential flow equation shown in following expressions:

$$I_x u + I_y v + I_t = 0 \tag{3}$$

where, I_x, I_y and I_t are the partial derivatives of image brightness with respect to x, y and t, respectively. Having one equation in two unknowns δx, δy for each pixel is an aperture problem of the optical flow algorithms. To find the optical flow another set of equations is needed, using some additional constraint. All optical flow methods introduce additional conditions for estimating the actual flow. There are several methods employed to determine optical flow, namely: Block-based methods, differential methods, Phase Correlation and General variational methods (Atcheson et al., 2009). Differential methods are widely used for navigation tasks, and are mainly based on partial derivatives of the image signal and/or the sought flow field and higher-order partial derivatives. One of those methods is used to estimate flow vectors to steer the robots.

2.1.1 The optical flow method proposed by Horn and Schunk

Horn and Schunk proposed one of the most important optical flow methods using an gradient based approach (Horn & Schunck, 1981). According to their methodology, a regularizing term associated with smoothness is added to the general flow equation, as illustrated in Equation 3, in which neighbouring pixels have the same velocity as moving objects, so the brightness pattern of an image changes regularly. This constraint is demonstrated by minimizing the squares of gradient magnitudes. Smoothness of an optical flow area can also be calculated by determining the Laplacian of optical flow vectors speed both horizontal and vertical directions denoted by u and w respectively, illustrated in following expressions:

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

$$\nabla^2 v = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \tag{4}$$

Where $E_i = \frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t}$ and $E_s = \nabla^2 u + \nabla^2 v$. The aim is to minimize the total error given by the following expressions that includes σ as the regularization parameter, controlling the association between the detail and the smoothness. High values of σ makes the smoothness constraint dominate and leads to a smoother flow

$$\iint \left(E_i{}^2 + \sigma^2 E_s{}^2 \right) dx dy \tag{5}$$

Horn and Schunk, can be used as the main reference to understand and solve the given error function, from which a pair of equations for each point can be obtained. Direct solution of these equations such as Gaus-Jordan Elimination (Bogacki, 2005) would be very costly. Instead, an iterative Gauss Seidel, approach is used to reduce the cost and obtain the flow vectors, as follows (Horn & Schunck, 1981):

$$u^{n+1} = \bar{u}^n - \left(\frac{I_x \bar{u}^n + I_y \bar{u}^n + I_t}{\sigma^2 + I_x^2 + I_y^2} \right); \quad v^{n+1} = \bar{v}^n - \left(\frac{I_x \bar{v}^n + I_y \bar{v}^n + I_t}{\sigma^2 + I_x^2 + I_y^2} \right) \tag{6}$$

where I_x I_y and I_t are the partial derivatives with respect to x,y and t respectively, and the superscript *n+1* denotes the next iteration, which is to be calculated and *n* is the last calculated result (Horn & Schunck, 1981).

Fig. 1. Optical Flow vectors (Guzel & Bicker, 2010).

2.2 Optical flow for mobile robot navigation

Flow vectors are utilized to navigate autonomous systems based on the Balance Strategy (Souhila & Karim, 2007), shown in the following equation, and the depth information which is extracted from the image sequence using Focus of Expansion(FOE) and Time To Contact values(TTC) (Souhila & Karim, 2007). The fundamental idea behind the Balance strategy is that of motion parallax, when the agent is translating, closer objects give rise to faster motion across the retina than farther objects. It also takes advantage of perspective in that closer objects also take up more of the field of view, biasing the average towards their associated flow (Contreras, 2007). The agent turns away from the side of greater flow. This control law is formulated by:

$$\Delta(F_l - F_r) = \left(\frac{\sum|w_L| - \sum|w_R|}{\sum|w_L| + \sum|w_R|}\right) \tag{7}$$

where $\sum|w_L|$ and $\sum|w_R|$ are the sum of the magnitudes of optical flow in the visual hemi fields on both sides of the robot's body. The following expression gives the new heading angle

$$\theta_{new} = (\Delta(F_l - F_r) \times k) \tag{8}$$

where k, is a constant, and used to convert the obtained result to an appropriate control parameter to steer the robot.

2.3 Appearance-based methods

Appearance based methods that identify locations on the basis of sensory similarities are a promising possible solution to mobile robot navigation. The main idea behind the strategy is to head the robot towards the obstacle-free position using similarities between the template and the active images (F. Vassallo et al., 2000). The similarity between the image patterns can be obtained by using feature detectors, involving corner based detectors, region based detectors and distribution based descriptors (Alper et al., 2006). However, most of these techniques consume a lot of process on time which is not appropriate for real time systems. In order to handle this problem in mobile robot applications, algorithms are designed based on the appearance of individual pixels. The classification of the obstacles is carried out by using the pixel difference between the template and active image patterns.In principally; any pixel that differs in appearance from the ground is classified as an obstacle. However, the method requires three assumptions that are reasonable for a variety of indoor and outdoor environments which are (Saitoh et al., 2009):

a. Obstacles must be different in appearance from the ground.
b. The ground must be flat.
c. There must be no overhanging obstacles.

The first assumption is to distinguish obstacles from the ground, while the second and third assumptions are required to estimate the distances between detected obstacles and the robot. There are several models for representing colour. The main model is the RGB(Red, Green, Blue) model which is used in monitor screens and most image file formats however, colour information for RGB model is very noisy at low Intensity. The RGB format is mostly converted to a HSV (Hue, Saturation, and Value). In HSV, Hue is what humans perceive as colour, S is saturation and Value is related to brightness, (or HIS(Hue, Intensity, Saturation) model) and in HIS, H and S represents the same as parameters in HSV colour models but I is an intensity value with a range between [0,1] where 0 is black and white is 1. These colour spaces are assumed to be less sensitive to noise and lighting conditions. The flow chart of the appearance based obstacle detection systems is illustrated in Figure 2. The input image is first convolved with a smoothing filter to reduce the noise effects, and then smoothed image is converted to *HIS, HSV* or any related *colour space* with respect to the developed algorithm (Fazl-Ersi & Tsotsos, 2009). A reference area is obtained from this image which might be any shape of geometry such as trapezoidal, triangle or square, and histogram values of this reference area are generated (Saitoh et al., 2009). Finally, a comparison between the reference image and the current image is made using some predefined threshold values. For instance, assume that the bin value, Hist($H(x, y)$), of the generated histogram and the threshold value,T_H, are compared, where $H(x, y)$ is the H value at pixel (x, y). If Hist($H(x, y)$) > T_H then the pixel (x, y) is classified into the safe region, or else it is classified into the obstacle region. In order to simply the problem, the results are represented in a binary image in which the safe path is represented with white but the obstacles are represented with black, as illustrated in Figure 3.However, identifying places purely on the basis of sensory similarity is too simplistic; different places may look very similar, even with a rich sensing methodology due to lighting conditions, shadows on illumination Furthermore, for dynamic environments there might be unexpected stains on the ground which may be the detected as an obstacle and leads the robot to an unsafe path. An example with respect to this case is illustrated in Figure 4.
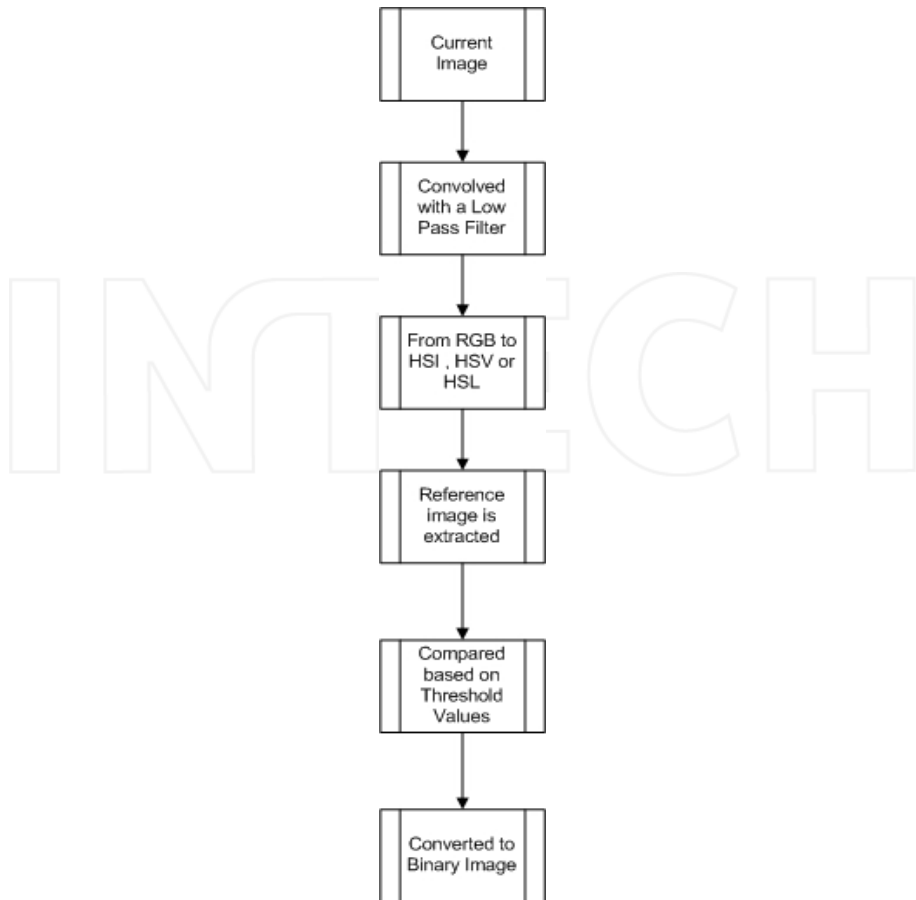
Fig. 2. Flow chart of the Appearance Based obstacle detection algorithm.

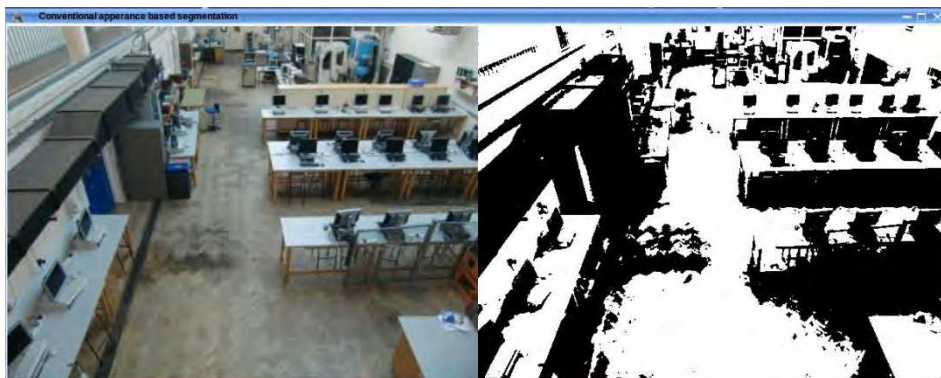Fig. 3. Appearance based obstacle detection method.

Fig. 4. Effects of lighting conditions and unexpected stains on the floor.

3. SIFT and template matching based obstacle avoidance strategy

In order to cope with the drawbacks of the conventional appearance based methods, a novel feature matching based technique, comprising a Scale Invariant Feature Transform and Template matching with a convolution mask, will be discussed in this section. The detail of the control algorithms with respect to these techniques is illustrated in Figure 5. Before introducing the proposed control algorithm and fusion technique, essential background knowledge regarding the SIFT and Template matching will be presented.
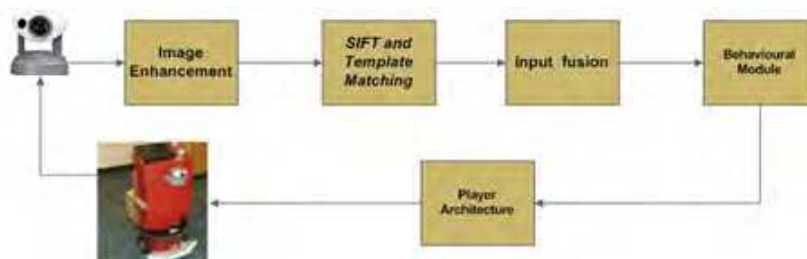
Fig. 5. Control architecture of the obstacle avoidance system.

3.1 Scale invariant feature transform (SIFT)

The Scale Invariant Feature Transform formerly abbreviated as SIFT is an algorithm in computer vision to detect and describe local features in images. The algorithm was published by Lowe (Lowe, 1999a, 2004b), and since then has been accepted as one of the most powerful local feature detection technique. The most notable improvements provided by SIFT are invariance to scale and rotation, and accuracy in feature point localization and matching. The evaluations carried out proposes that SIFT-based descriptors, which are region-based, are the most robust and distinctive, and are therefore best suited for feature matching. A summary of the SIFT methodology is illustrated in Fig. 6. (Lowe, 1999a, 2004b). The initial state of this algorithm is Scale space extreme detection where the interest points, which are called key-points in the SIFT framework, are detected. For this, the image is convolved using Gaussian filters, proved the only possible scale-space kernel, at different

Problem	Technique	Advantage
key localization / scale / rotation	DoG / scale - space pyramid / orientation assignment	accuracy, stability, scale & rotational invariance
geometric distortion	blurring / resampling of local image orientation planes	affine invariance
indexing and matching	nearest neighbour / Best Bin First search	Efficiency / speed
Cluster identification	Hough Transform voting	reliable pose models
Model verification / outlier detection	Linear least squares	better error tolerance with fewer matches
Hypothesis acceptance	Bayesian Probability analysis	reliability

Fig. 6. SIFT Methodology (Lowe, 1999a, 2004b).

scales, and then the difference of successive Gaussian-blurred images are obtained, illustrated in Figure 7. The convolved images are grouped by octave which corresponds to doubling the value of standard deviation of the Gaussian distribution (σ).The Convolution of the image at scale $k\sigma$ with a Gaussian filter is expressed as follows:

$$L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y) \tag{9}$$

where,

$$G(x, y, \sigma) = \frac{1}{(2\pi\sigma^2)} exp^{-(x^2+y^2)/2\sigma^2} \tag{10}$$
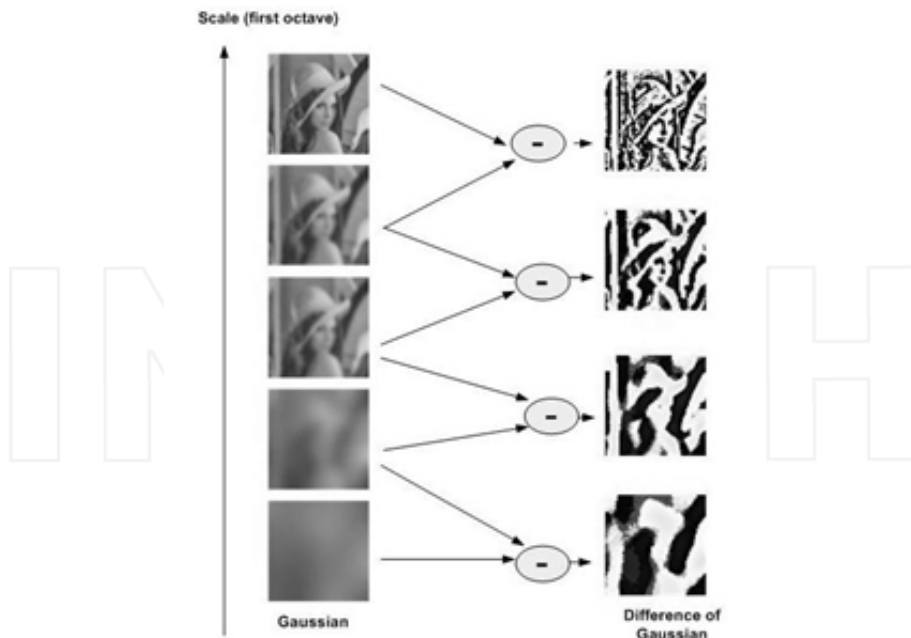
Fig. 7. Gaussian images are subtracted to produce the DoG.

When DoG images have been obtained, key- points are identified as local minima/maxima of the DoG images across scales. This is done by comparing each pixel in the DoG images to its eight neighbors at the same scale and nine corresponding neighboring pixels in each of the neighboring scales. If the pixel value is the maximum or minimum among all compared pixels, it is selected as a candidate keypoint', as shown in Figure 8.

Following steps are Key-point localization and Orientation assignment (Lowe, 1999a, 2004b). After key-point orientation has been completed, each key specifies stable 2D coordinates, comprising x, y, scale and orientation. Finally, a signature, local descriptor, is computed as a set of orientation histograms on 4x4 pixel neighbourhoods. Histograms have 8 bins each, and each descriptor contains an array of 4 histograms around the key-point. This leads to a SIFT feature vector with 8x4x4 = 128 elements, illustrated in Figure 9. This vector is normalized to enhance invariance to changes in illumination.
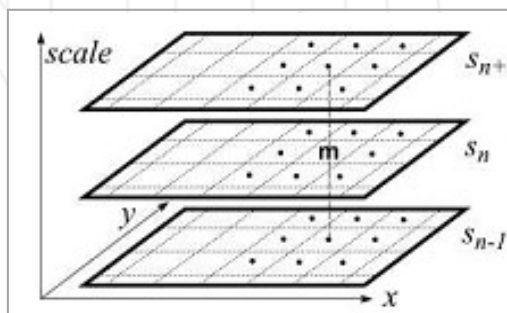
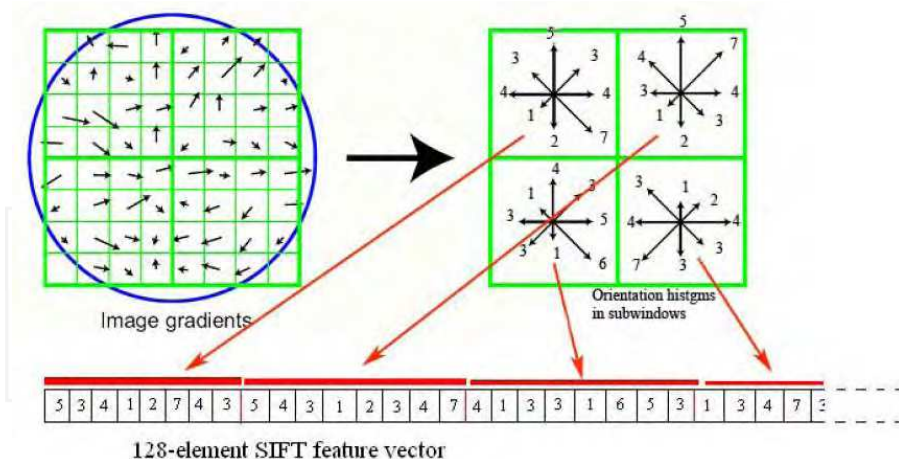Fig. 8. A key-point is defined as any value in the DoG.

Fig. 9. SIFT Feature Descriptor.

3.1.1 SIFT matching

Feature vectors which extracted from the SIFT algorithm to solve common computer vision problems, comprising object detection, 3D scene modeling, recognition and tracking, robot

localization and mapping. This procedure requires an appropriate and fast matching algorithm. An example with respect to the SIFT matching is illustrated in Figure 10. The main matching algorithm is able to find each key-point by identifying its nearest neighbor in the database of key-points from training images. The nearest neighbor is defined as the key-point with minimum Euclidean distance for the invariant descriptor vector, as previously discussed.However, many features may not be matched correctly due to background clutter in natural images or may not have correct match in the training database, and hence, mismatches should be discarded in order to obtain accurate results. Global thresholding on distance to the closest feature does not perform well, as some descriptors are much more discriminative than others. Alternatively, a more effective measure is obtained by comparing the distance of the closest neighbor to that of the second-closest neighbor, which performs well. An appropriate threshold value regarding this comparison, called distance ratio, is employed to reject false matches while increasing the correct matches. This value varies from 0.1 to 0.9, depending on the application type. In this case, 0.7 is employed with respect to the experimental results, which eliminates %90 of the false matches while discarding almost %10 correct matches. In addition, to reject rest of all false matches, an essential statistical method is applied to the matching space, fundamentally using the rate between scale and orientation parameters of the feature vectors. According to this method, even though the matching is validated between any two feature vectors, if the scale or the orientation parameter rate between them is more than a threshold value, matching is discarded; this procedure performs robustly to decrease false matches over all data sets (Lowe, 1999a,2004b).

Fig. 10. SIFT matching.

3.2 Template matching

Template matching is a simple and popular technique in computer vision and image processing to find small parts of an image which match a template image. It can be used in mobile robot navigation or as a way to detect edges or objects in images; an example with respect to this technique is illustrated in Figure 11. A basic method of template matching uses a convolution mask which can be easily performed on grey images. The convolution output will be the highest at places where the image structure matches the mask structure, i.e. where large image values get multiplied by large mask values. This method is normally implemented by first picking out a part of the search image to use as a template.
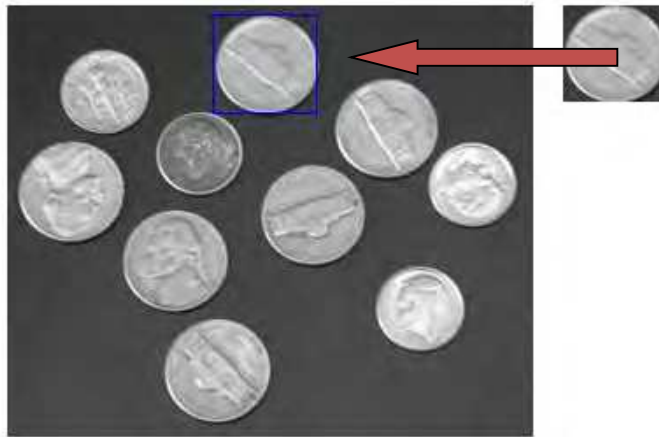
Fig. 11. Template Matching.

For instance, the input and output images are called I(x, y) and O(x, y) respectively, where (x, y) represent the coordinates of each pixel in the images and the template is called $T(x_t, y_t)$, where (x_t, y_t) represent the coordinates of each pixel in the template. The technique simply moves the centre of the template $T(x_t, y_t)$ over each (x, y) point in the search image and calculates the sum of products between the coefficients in I(x, y) and $T(x_t, y_t)$ over the whole area spanned by the template. As all possible positions of the template with respect to the Input image are considered, the position with the highest score is the best position, and which is represented in the output image. There are several techniques to handle translation problem; these include using SSD (Sum of squared differences), CC (Cross Correlation) and SAD (Sum of absolute differences) (Wen-Chia & Chin-Hsing, 2009). One of the most powerful and accurate of those is CC, which basically measures the similarity of two variables and defined as follows (9):

$$Cor = \frac{\sum_{i=0}^{N-1}(x_i - \dot{x}) \times (y_i - \dot{y})}{\sqrt{\sum_{i=0}^{N-1}(x_i - \dot{x})^2 \times \sum_{i=0}^{N-1}(y_i - \dot{y})^2}} \qquad (11)$$

Where N is the template image size; $\dot{x}$ and $\dot{y}$ represents average gray level in the template and source image respectively. The goal is to find the corresponding (correlated) pixel within a certain disparity range that minimizes the associated error and maximizes the similarity. This matching process involves computation of the similarity measure for each

disparity value, followed by an aggregation and optimization step (ZitovÃ¡ & Flusser, 2003). An example related to correlation based technique is illustrated in Figure 12.
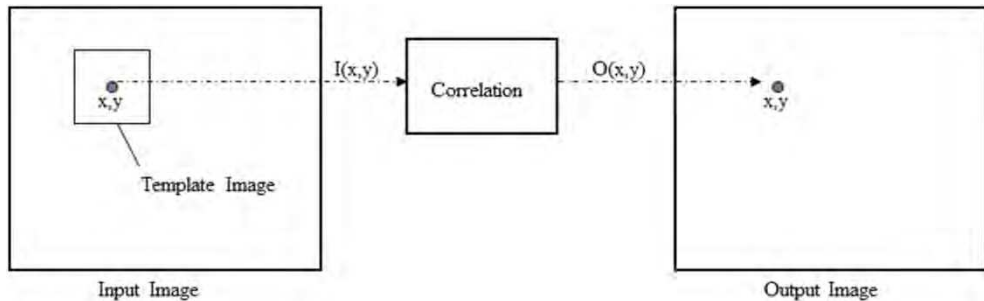

Fig. 12. Correlation technique based Template Matching.

3.3 Obstacle avoidance using SIFT and appearance based

Appearance based methods have significant processing and performance advantages which make them a good alternative for vision based obstacle avoidance problems. However as mentioned previously, there are certain drawbacks, which *restricts the applicability of these methods*. In order to handle these drawbacks, the results of the conventional method is improved by using the results of *SIFT* based feature matching approach. The flowchart diagram of the proposed algorithm is illustrated in Figure 13. First the acquired image is smoothed using a Gaussian filter to eliminate the noise in the image. Then a copy of the original image is converted to PGM image format which is required to carry out the *SIFT* matching process. Both images are divided into 16 sub-images composed of 44×36 pixels. For each sub images, template matching using cross correlation and *SIFT* matching are performed against reference images, illustrating the safe route, simultaneously. The results for each segment are fused using fuzzy logic to build up a sufficiently accurate occupation map of the environment.

The robot, which employs a Subsumption Architecture (Brooks, 1986), is successfully directed along a collision-free path using this map. To evaluate the performance of the proposed algorithm, it is applied to a test case, as illustrated in Figure 14. The matching results, shown in Figure 15, indicate that both techniques generate similar results under ideal conditions, involving low illumination changes and flat ground. However, template matching may fail to provide accurate matching results against illumination problem, and non-flat surfaces. In order to handle these cases, *SIFT* matching provides a reliable matching strategy, which is able to match the extracted features, invariant to scale, orientation, affine distortion, and partially invariant to illumination changes, with high accuracy. An example illustrating this condition can be seen in Figure 16, comprising 9th and 10th sub-image sequences. Despite the illumination problem and non-flat ground, the SIFT matching performs well for these cases illustrated in Figure 17. Consequently; the results indicate that instead of using each method separately, fusion of them generates more reliable results. A fuzzy logic based approach with respect to this fusion procedure will be discussed in the following section.
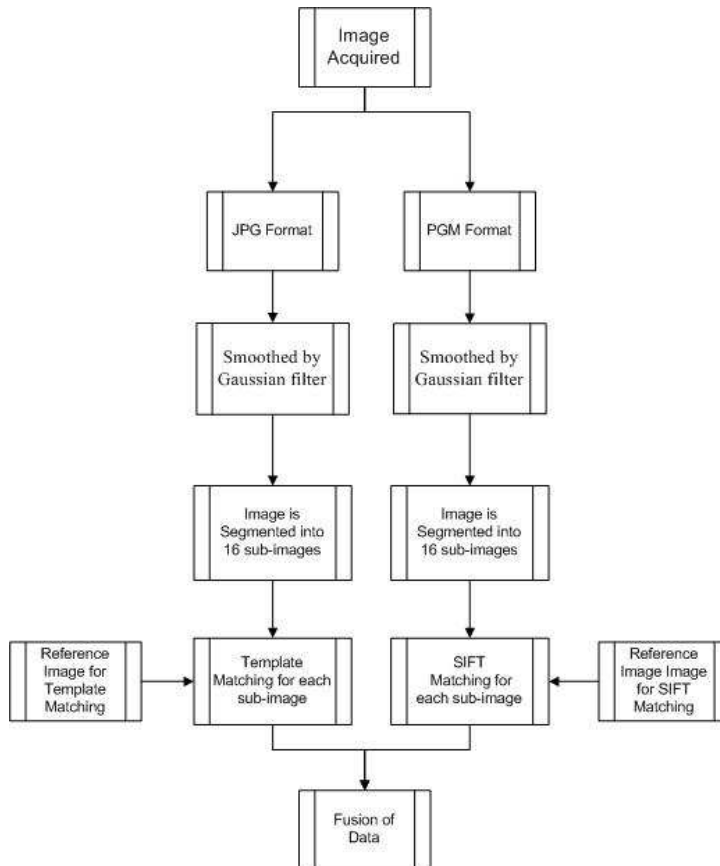
Fig. 13. An example from the testing environment.
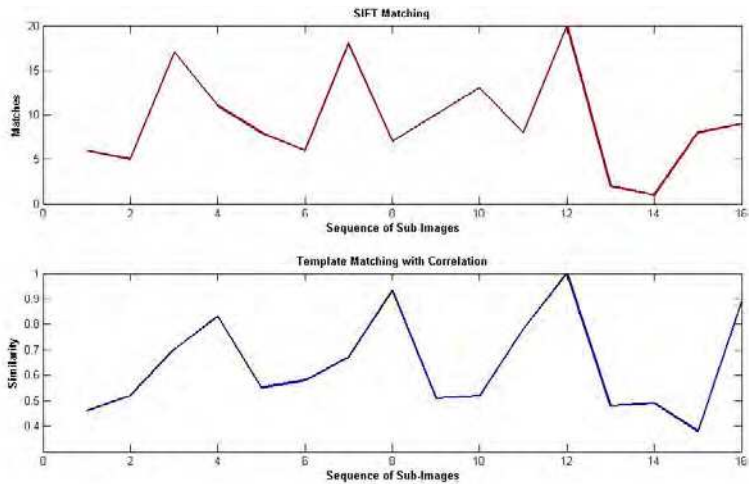

Fig. 14. An example from the testing environment.

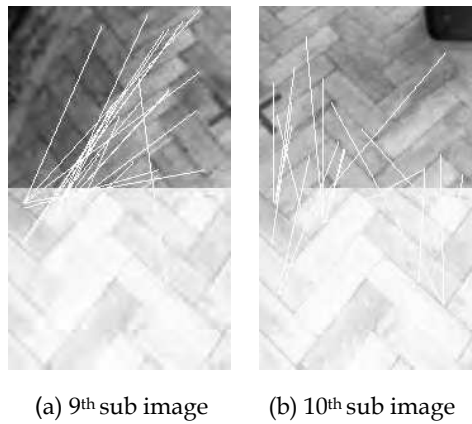Fig. 15. Comparison of two algorithms.

(a) 9th sub image   (b) 10th sub image

Fig. 16. Matching results of two algorithms.

Fig. 17. False matching generated by SIFT matching.

3.4 Fuzzy logic

Fuzzy logic (FL) is a form of many-valued logic derived from fuzzy set theory to deal with reasoning that is robust and approximate rather than brittle and exact. In contrast with two-valued Boolean logic, FL deals with degrees of membership and degrees of truth. FL uses the continuum of logical values between 0 (completely false) and 1 (completely true). FL has been utilized as a problem-solving control system by several researchers for different problems. Since, FL lends itself to implementation in systems ranging from simple, small, embedded micro-controllers to large, networked, multi-channel PC or workstation-based data acquisition and control systems.

FL is fundamentally easy to implement and provide faster and more consistent results than conventional control methods. In this study, a FL based control system based on the Mamdani method is designed to fuse given algorithms. The basic configuration of a fuzzy-logic system is composed of three parts: *Fuzzification*, *Inference Mechanism* and *Deffuzification* (*Driankov, 1987*). These will be presented and associated with the fusion problem in the following parts.

3.4.1 Fuzzification

Fuzzfication comprises a scale of transformation of input data of a current process into a normalised domain. This process requires the identification of two parts: the first part defines the fuzzy variables that correspond to the system input variables. The second part is to define the fuzzy sets of the input variables and their representative membership functions including the ranges of the data. Membership function, may cross the boundary of another fuzzy membership function. Each membership function may be triangular, a trapezoidal or bell shaped, as illustrated in Figure 18. The choice of the fuzzy sets is based on expert opinion using natural language terms that describe the fuzzy values. In this study triangle and trapezoid models are utilized to design membership functions of input and output values.

Fuzzy logic uses intersection, union, and complement operations to represent the standard common operators of AND, OR, and NOT, respectively. The most common method used to calculate intersection and union operations are the Minimum and Maximum functions. For the fuzzy sets M and N which are subsets of the universe X, the following definitions are proposed to represent the AND, OR, and NOT operators, respectively (Ross & Hoboken, 2004) (see Figure 19).
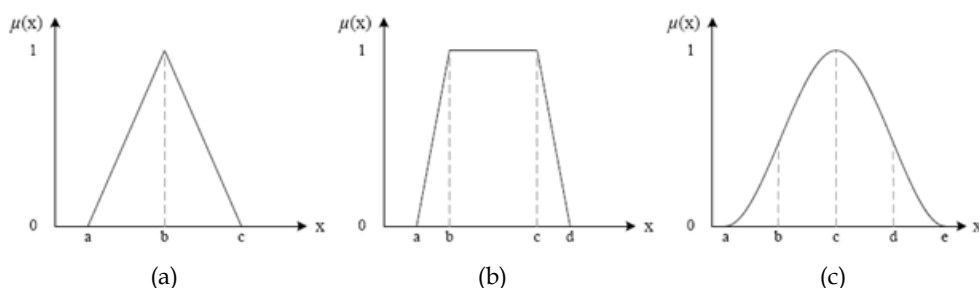

Fig. 18. Membership function shapes, (a) Triangular, (b) Trapezoidal, (c) Gaussian.

The triangular function has three parameters which can be defined as follows:

$$\mu(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x < b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & x > c \end{cases} \tag{12}$$

The trapezoidal function incorporates four parameters can be represented as:

$$\mu(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x < b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c < x \leq d \\ 0, & x > d \end{cases} \tag{13}$$

3.4.2 Inference mechanism

The generation of the fuzzy rules is a first step which depends on the knowledge and experience of the human operators, the fuzzy model of the plant concerned, and an analysis of the system. The rule-base is composed of two parts namely, the IF-part and the THEN-part. The IF-part is the antecedent part where rules are defined to describe the system state in terms of a combination of fuzzy propositions while the THEN-part is the consequent part which forms the desired conclusion of the output variable. Afterwards, fuzzy inference provides the conclusion of the rule-base and forms the intermediate stage between the fuzzification and defuzzification of the fuzzy system. There are two methods used to find the rules conclusion namely Max-Min inference and Max-Product inference. Max-Min utilizes the Minimum operator to combine the antecedent of the IF-THEN rules which produces modified fuzzy sets for the outputs. These modified sets are then combined using the Maximum operator and Max-Product inference and utilizes the standard Product operator to combine the antecedent of the IF-THEN rules. Then the Maximum operator is used to combine these modified sets (Ross & Hoboken, 2004).
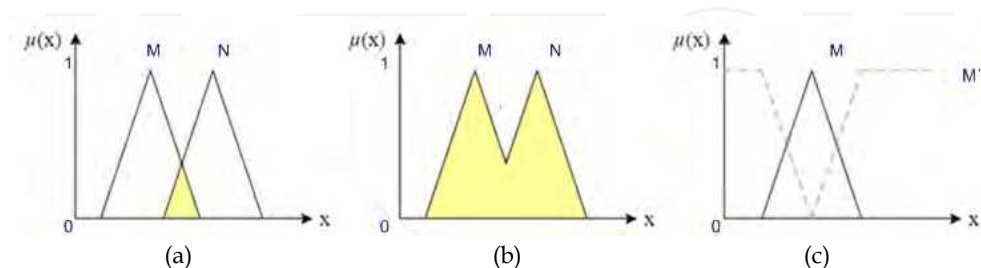

Fig. 19. Fuzzy Operators, (a) AND, (b) OR, (c) NOT

3.4.3 Defuzzification

Defuzzification is the process of mapping from a space of inferred fuzzy control action to a space of non-fuzzy control actions where the calculated crisp value is that which best

represents the inferred control action. Several methods can be used to calculate this crisp value such as the Centre-of-Area, Centre-of-Largest-Area, Centre-of-Sums, and Mean-of-Maximum. These methods are based on two basic mechanisms: Centroid and Maximum. The centroid methods are based on finding a balance point while the Maximum methods search for the highest peak of weight (area) of each fuzzy set (Ross & Hoboken, 2004). Centre-of-Sum is used in this study which is faster than many defuzzification methods, and is not restricted to symmetric membership functions. This process performs the algebraic sum of individual output fuzzy sets instead of their union, illustrated in the following equation:

$$\mu^* = \frac{\int y \sum_{n=1}^{k} \mu_n(y) d_y}{\int \sum_{n=1}^{k} \mu_n(y) d_y} \tag{14}$$

3.4.4 Fusion of algorithms with fuzzy logic

The fundamental architecture of the proposed system is illustrated in Figure 20. The FL controller has two inputs, namely SIFT and Correlation which involves matching strength and similarity rate respectively. The output of the controller generates an appropriate turning rate (w) to avoid obstacle. In order to adapt the results of vision based algorithms efficiently as well as to operate the robot smoothly, the image is divided into n clusters (sub-images), based on the resolution of the image, its parts, and for each cluster a fuzzy fusion algorithm is applied. SIFT matching value of each cluster is rescaled by multiplying with n which normalizes the input for Fuzzy Inference System (FIS). Final turning rate is calculated to sum up all clusters, considering the sign of each part which is defined as follows:

$$w = \sum_{i=1}^{n} w_i , \ n \ is \ even, left \ clusters \ are \ positive \ and \ right \ ones \ are \ negative \tag{15}$$
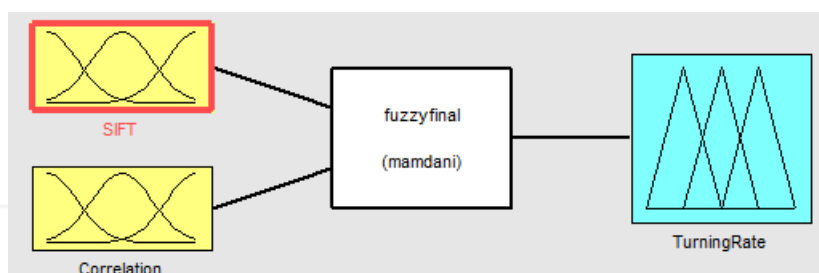

Fig. 20. Architecture of Fuzzy control system for fusing operation.

To generate a value of turning rate (w), a Max-Min type FIS and a Centre-of-Sums defuzification method were used, as previously discussed. The first step is to design membership functions for fuzzification and defuzzification processes. While, several researchers utilize different membership function shapes regarding to the problem in various applications. The trapezoidal and triangular shapes have been selected in this work to simplify the computation. However, it should be noted that there are no precise methods to adjust the membership functions. Table 1 provides an example of the selection of the fuzzy terms that describe linguistic variables used in this study. For this study three membership functions are defined *namely: SIFT (s_m), Correlation (c_s) and Turning Rate (w). SIFT* function, representing the matching strength between the reference image and the

current image, is illustrated in Figure 21(a). Whereas the correlation function represents the similarity between the reference and current images, as illustrated in Figure 21(b). Turning Rate is the output function which represents the angular velocity value to steer the robot whilst avoiding obstacles (see Figure 22). The next step is to design appropriate fuzzy rules depending on the detail of each Fuzzy Inference System (FIS). A set of experiments were carried out until the outputs are judged to satisfy each different situations. Table 2 displays the fuzzy rules for the given problem.

Linguistic Variables	Linguistic Terms
SIFT	Weak, Medium, Strong
Correlation	Weak, Medium, Strong
Turning Rate	Straight, Less, Medium, Sharp

Table 1. Linguistics variables and their linguistics terms.

Inputs	Correlation (c_s)		
SIFT (s_m)	Weak	Medium	Strong
Weak	Sharp	Medium	Less
Medium	Sharp	Medium	Straight
Strong	Medium	Less	Straight

Table 2. Fuzzy rule-base for Turning rate (w).

Test results lead to tune the system by changing rules, adjusting the membership functions shapes of both input and outputs. Once the procedure has been run several times, a consistent system is attained. The following section will integrate the proposed algorithm to a behavioral based architecture.
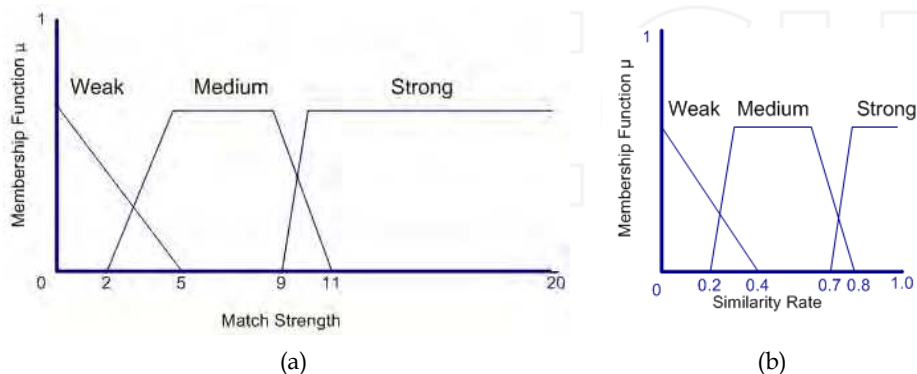

(a)   (b)

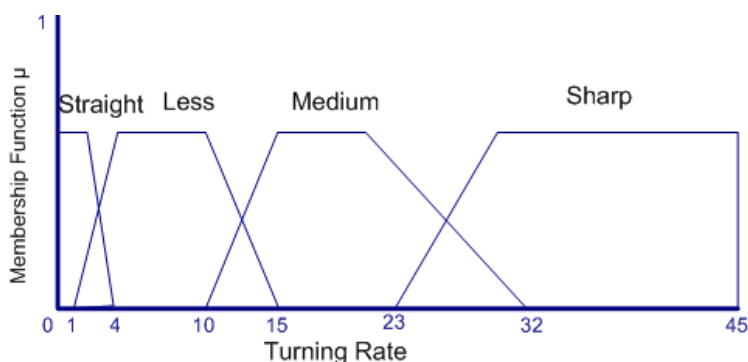Fig. 21. Membership function, (a) 'input' SIFT matching, (b) 'input' Correlation.

Fig. 22. Membership function, 'output' Turning rate.

4. Evaluation and implementation of the proposed system

In order to evaluate the performance of the algorithm, it is integrated using a behavioral based architecture. There are several approaches to designing a behavioural-based architecture depending on the required task. In this study, the architecture has been designed based on the subsumption architecture in which each layer or behaviour implements a particular goal of the robot and higher layers are increasingly abstract. Each layer's goal subsumes that of the underlying layer, and their interaction with each other will be illustrated by using finite state machines (FSM) which defines several states (behaviours) that represents a current situation for the robot. Certain events from the outside of the world can change the state. For instance, the robot could have a *Goto* state whereby it is moving about the environment trying to get closer to its goal. When any obstacle is detected nearby, the state may change from *Goto* to *Obstacle Avoidance*, and the avoidance algorithm will move the robot away from the obstacle. When the obstacle has been avoided, the robot state will change back to the *Goto*. The architecture, designed for this study, comprises three behaviors, namely: *Goto, Avoid_ Obstacle* and *Finish*. *Goto* behavior steers the robot to a specific goal position, *Avoid_Obstacle behaviour* utilizes the proposed vision based intelligent algorithm to avoid obstacles, and *Finish* behavior is merely enabled after the goal is found, and the robot is stopped. FSM diagram of the system is illustrated in Figure 23.

The system was developed using a Pioneer 3-DX mobile robot, with an on-board IntelPentium 1.8 GHz (Mobile) processor, and includes 256 Mbytes of RAM memory, as shown in Figure 24 (a). The mobile robot used in this study has been developed as a part of the Intelligent Robot Swarm for Attendance, Recognition, Cleaning and Delivery (IWARD) project. An Axis-213 camera, 25 frame rate, was integrated into this system. The software architecture of the proposed system is supported by CIMG Library and Player Architecture, which are open-source software projects. All experiments were conducted in an area of the Robotics and Automation Research Laboratory of Newcastle University, which has physical dimensions of 15.60m x 17.55m, as illustrated in Figure 24 (b). The camera tilted down 30 degrees to detect the floor precisely, and a reference image was taken from this environment. To evaluate the performance of the system, several different scenarios were performed and four of them will be discussed in this section.
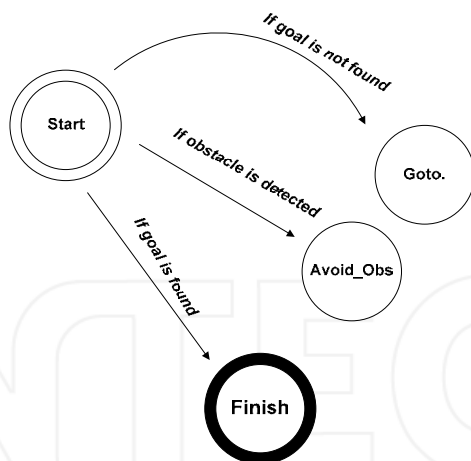
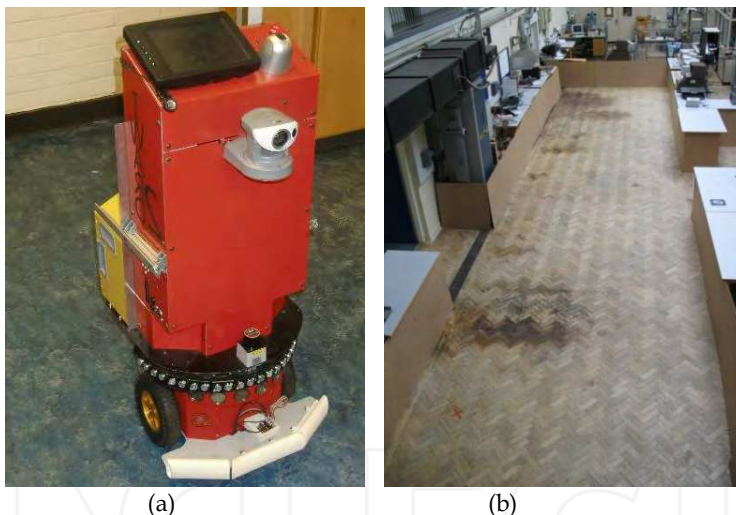Fig. 23. FSM diagram of the behavioral architecture.

(a)   (b)

Fig. 24. Test environment, (a) Pioneer 3-DX robot with Axis-213, (b) Robotics and Automation Laboratory

Scenario 1: The mobile robot is required to navigate from 'Start Position' (-6, 0) along a forward direction in a partially cluttered enviorement and avoids one obstacle located along its path as shown in Figure 25 (a). Performance of the image processing algorithms during the task is given Figure 25 (b). The robot's GoTo behaviour steers the robot forward direction.

Scenario 2: The mobile robot navigates from 'Start Position' (-6, 0) along a forward direction in a partially cluttered enviorement and has to avoid two obstacles located along its path as shown in Figure 26 (a) and evaluation of the image processing algorithms illustrated in Figure 26 (b). The robot's GoTo behaviour steers the robot forward direction.

Scenario 3: The mobile robot navigates from 'Start Position' (-7, 0) along a forward direction, two obstacles located on both sides of its path as shown in Figure 27 (a). Performance of the image processing algorithms are illustrated in Figure 27 (b). The robot's GoTo behaviour steers the robot forward direction.

Scenario 4: The mobile robot is required to navigate from 'Start Position' (-6, 0) to the 'Goal Position' (3, 0.5) in a partially cluttered enviorement as shown in Figure 28(a). Performance of the image processing algorithms during the task is given Figure 28 (b). The robot's GoTo behaviour steers the robot to a specific position which evaluates the performance of the obstacle avoidance algortihm with a Wavefront path planning algorithm (Barraquand & Latombe, 1991).

Figure 25 (a) presents the navigation results for scenario 1, in which the robot steers forward until it percives the obstacle. The robot avoids the obstacle succesfully and then keeps going forward until it detects the wall, after which it avoids the wall and continues heading forward until it encounters the door. The robot succesfully avoids the door and continues moving.
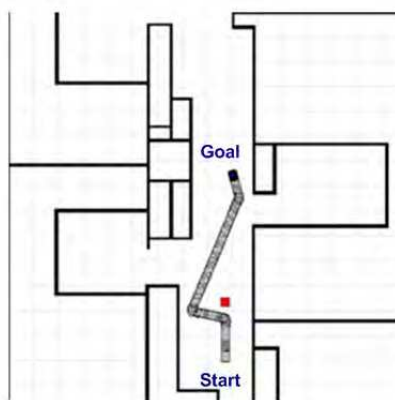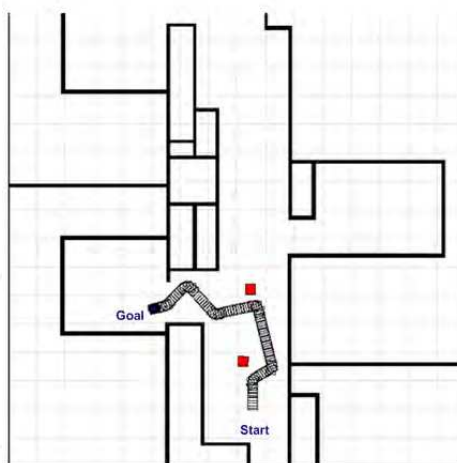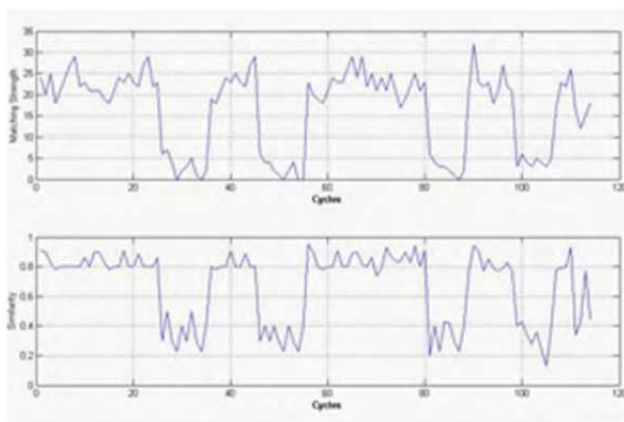

(a)
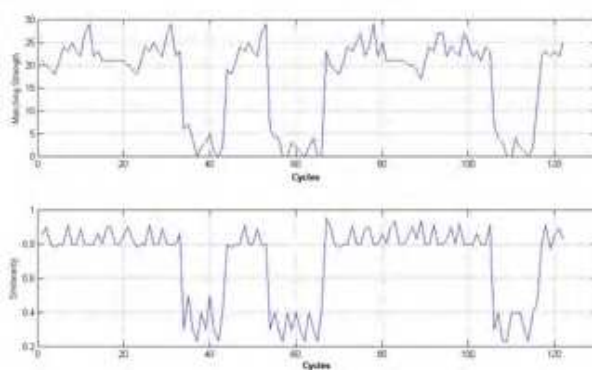
(b)

Fig. 25. Scenario 1, (a) Estimated Trajectory, (b) Performance of the image processing algorithms.

(a)

(b)

Fig. 26. Scenario 2, (a) Estimated trajectory, (b) Performance of the image processing algorithms.

Figure 26(a) displays the test results for scenario 2. Two obstacles are located along the robot's path in the forward direction. The robot navigates until it detects the first obstacle, then it avoids the obstacle and steers forward. Having detected the wall, the robot ceases forward motion and avoids the wall, after which, it moves forward until it detects the second obstacle, which it avoids succesfully. Subsequently, the robot avoids the door and the wall respectively.It finally moves forward until it is ceased. The third simulation is given Figure 27(a) where the robot attempts to move forward while passing the gap between two obstacles. It keeps moving forward until the obstacles are detected, initially turns left, followed by a right maneuver. After the robot succesfully avoids the objects it then resumes its path and continues moving forward.
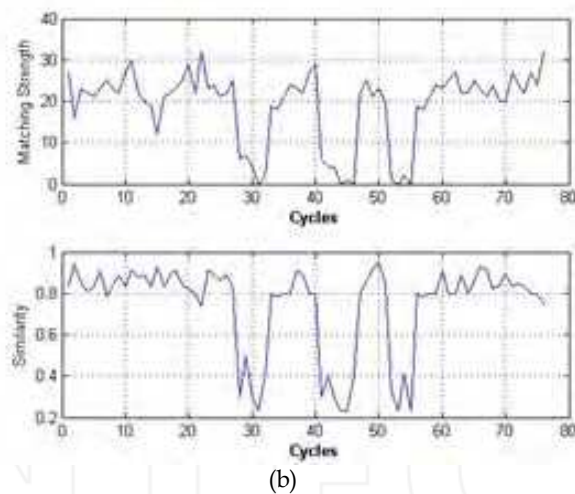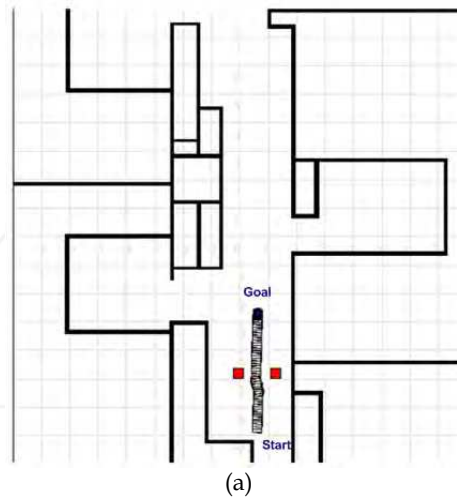
(a)

(b)

Fig. 27. Scenario 3, (a) Estimated trajectory, (b) Performance of the image processing algorithms.

Figure 28(a) displays the test results for scenario 4. The aim of this scenario is to test the avoidance strategy with a Wavefront path planning algortihm (Barraquand & Latombe, 1991) provided by the Player Architecutre. The robot navigates towards the 'first waypoint' until it perceves the obstacle. The robot avoids the obstacle and resumes its desired path. As there are no obstacles located along the rest of its path to the goal, the robot, therefore navigates directly to its goal.
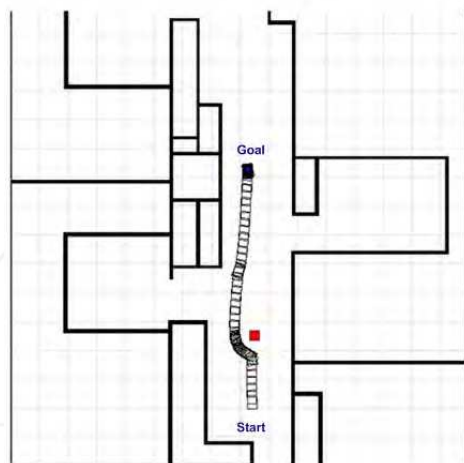
(a)
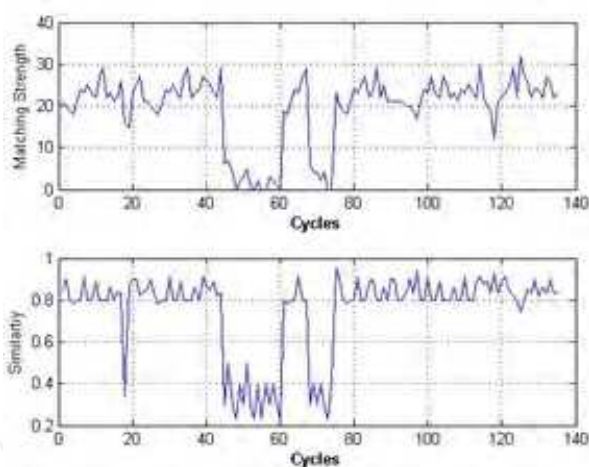
(b)

Fig. 28. Scenario 4, (a) Estimated trajectory, (b) Performance of the image processing algorithms.

5. Conclusion

The aim of this research was that it should be possible to develop a robust intelligent vision based obstacle avoidance system that can be adapted to realistic navigation scenarios. Most of the previously proposed techniques suffer from many problems. One of the most popular of those is *Apperance-based* methods which basically consist of detecting pixels different in appearance than the ground and classifying them as obstacles. Conventional *Appearance-*

based methods utilize simple template matching techniques which are fast but highly sensitive to lighting conditions and structure of the terrains. In addition, *Optical flow-based* methodologies,, the pattern of apparent motion of objects in a visual scene caused by the relative motion relying on apparent motion, are also highly sensitive to lighting conditions and suffer from nose.

To overcome those problems, a new obstacle avoidance method has been proposed which is inspired from feature matching principal. *SIFT-based* descriptors outperform other local descriptors on both textured and structured scenes, with the difference in performance larger on the textured scene. Accordingly, conventional *SIFT* algorithm, which is able to perform with high accuracy and reasonable processing time, is employed to match images, and it is finally adapted to an obstacle avoidance technique. According to the techniques, reference image, presenting the free path, is matched with the current image during the navigation to estimate the steering direction. However, preliminary experimental results reveals that despite the *SIFT* algorithm performs better than conventional methods with stained environment and compensates lighting failures, they may fail or produce several mismatched features due to low resolution or vibration caused by navigation over rough terrains. Therefore instead of using each method individually, conventional template matching method and *SIFT-based* descriptor are fused by using an appropriate *Fuzzy Fusion* algorithm which increases the accuracy and compensates the errors caused by lighting conditions and stains.

In order to verify the performance of the proposed obstacle avoidance algorithm, Real experiments to guide a *Pioneer 3-DX mobile robot* in a partially cluttered environment are presented, Results validate that proposed method provides an alternative and robust solution for mobile robots using a single low-cost camera as the only sensor to avoid obstacles.

6. References

Alper, Y., Omar, J. & Mubarak, S. (2006) Object tracking: A survey. *ACM Comput. Surv.,* 38, 13.

Atcheson, B., Heidrich, W. & Ihrke, I. (2009) An evaluation of optical flow algorithms for background oriented schlieren imaging. *Experiments in Fluids,* 46, 467-476.

BArraquand, J. & Latombe, J.-C. (1991) Robot motion planning. A distributed representation approach. *International Journal of Robotics Research,* 10, 628-649.

Bogacki, P. (2005) HINGES - An illustration of Gauss-Jordan reduction. *Journal of Online Mathematics and its Applications.*

Brooks, R. A. (1986) ROBUST LAYERED CONTROL SYSTEM FOR A MOBILE ROBOT. *IEEE journal of robotics and automation,* RA-2, 14-23.

Contreras, E. B. (2007) A biologically inspired solution for an evolved simulated agent. *Proceedings of GECCO 2007: Genetic and Evolutionary Computation Conference.* London.

Desouza, G. N. & Kak, A. C. (2002) Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 24, 237-267.

Driankov, D. (1987) Inference with a single fuzzy conditional proposition. *Fuzzy Sets and Systems,* 24, 51-63.

F. Vassallo, R., Schneebeli, H. J. & Santos-Victor, J. (2000) Visual servoing and appearance for navigation. *Robotics and Autonomous Systems,* 31, 87-97.

Fazl-Ersi, E. & Tsotsos, J. K. (2009) Region classification for robust floor detection in indoor environments. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).* Halifax, NS.

Guzel, M. S. & Bicker, R. (2010) Optical flow based system design for mobile robots. *2010 IEEE Conference on Robotics, Automation and Mechatronics, RAM 2010.* Singapore.

Horn, B. K. P. & Schunck, B. G. (1981) Determining optical flow. *Artificial Intelligence,* 17, 185-203.

Lowe, D. G. (1999) Object recognition from local scale-invariant features. *Proceedings of the IEEE International Conference on Computer Vision.* Kerkyra, Greece, IEEE.

Ross, T. J. & Hoboken, N. J. (2004) *Fuzzy Logic with engineering applications* Hoboken, NJ:Wiley.

Saitoh, T., Tada, N. & Konishi, R. (2009) Indoor mobile robot navigation by central following based on monocular vision. *IEEJ Transactions on Electronics, Information and Systems,* 129, 1576-1584+18.

Souhila, K. & Karim, A. (2007) Optical flow based robot obstacle avoidance. *International Journal of Advanced Robotic Systems,* 4, 13-16.

Wen-Chia, L. & Chin-Hsing, C. (2009) A Fast Template Matching Method for Rotation Invariance Using Two-Stage Process. *Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP '09. Fifth International Conference on.*

Zitovã¡, B. & Flusser, J. (2003) Image registration methods: A survey. *Image and Vision Computing,* 21, 977-1000.
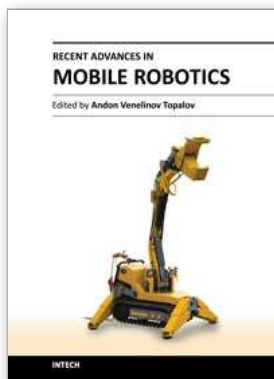
Recent Advances in Mobile Robotics

Edited by Dr. Andon Topalov

Mobile robots are the focus of a great deal of current research in robotics. Mobile robotics is a young, multidisciplinary field involving knowledge from many areas, including electrical, electronic and mechanical engineering, computer, cognitive and social sciences. Being engaged in the design of automated systems, it lies at the intersection of artificial intelligence, computational vision, and robotics. Thanks to the numerous researchers sharing their goals, visions and results within the community, mobile robotics is becoming a very rich and stimulating area. The book Recent Advances in Mobile Robotics addresses the topic by integrating contributions from many researchers around the globe. It emphasizes the computational methods of programming mobile robots, rather than the methods of constructing the hardware. Its content reflects different complementary aspects of theory and practice, which have recently taken place. We believe that it will serve as a valuable handbook to those who work in research and development of mobile robots.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Mehmet Serdar Guzel and Robert Bicker (2011). Vision Based Obstacle Avoidance Techniques, Recent Advances in Mobile Robotics, Dr. Andon Topalov (Ed.), ISBN: 978-953-307-909-7, InTech, Available from: http://www.intechopen.com/books/recent-advances-in-mobile-robotics/vision-based-obstacle-avoidance-techniques

INTECH
open science | open minds