Iranian Journal of Science & Technology, Transaction B, Engineering, Vol. 29, No. B1
Printed in The Islamic Republic of Iran, 2005
© Shiraz University

A NEW METHOD FOR EDGE FEATURE DETECTION AND MATCHING*

A. BEHRAD** AND S. A. MOTAMEDI

Dept. of Electrical Engineering, Amir Kabir University of Technology, Tehran, I. R. of Iran
Email: a_behrad@aut.ac.ir, motamedi@aut.ac.ir

Abstract– Image feature detection and matching is a fundamental task in image processing and machine vision. In this paper we present novel methods for feature detection and matching. We have used edge points and their accumulated curvature information for feature detection. The detected features are then matched using a new method called "fuzzy-edge based feature matching". To increase the speed of matching algorithms, a new multi-resolution based method and edge pyramids are proposed. Experimental results have shown that the algorithms are fast and reliable and can be used in the environments with noise and illumination change. The proposed algorithms can be used in various machine vision applications such as target tracking, image registration and stereovision.

Keywords– Feature detection, edge matching, multi-resolution based edge matching, edge pyramid

1. INTRODUCTION

One of the most important tasks in machine vision applications is the generation of image descriptors, which are more useful than the description of image with image pixels. The aim of these descriptors is to reduce the computation overhead of the machine vision algorithms by changing the large set of pixels to a list of features which are more manageable and applicable for higher-level processes. Applications of the feature detection algorithms are numerous, spanning a wide range of usages including image registration, stereovision, pattern recognition and moving target tracking, to name a few. The main purpose of feature detection algorithms in applications like target tracking, stereovision and image registration is to find corresponding points for features in different images. For example, by finding the matching points in tracking algorithms, the motion vectors or motion models between two consecutive frames are calculated. In the image registration applications, the transformation between two different views of the scene is also calculated using feature points in the first view and their matches in the second view. The use of features in these applications has two main advantages. First, since the matching process is carried out in the number of feature points, the computation overhead of the algorithm largely decreases. Second, because a feature is an image property in which image correspondence can be performed more accurately, therefore, the overall accuracy of the algorithm is improved.

There are many difficulties with feature detection and matching algorithms including the noise of image, change of scene illumination, change in object geometry and real-time constraint, to mention a few. In this paper, we propose new methods for image features detection and matching. To make the algorithms robust against noise and illumination variations, we have used edge points and their accumulated curvatures. The detected features are then matched using a new method called "fuzzy-edge based feature matching." To increase the speed of matching algorithms, a new multi-resolution based method and edge pyramids are proposed. Experimental results have shown that the algorithms are fast and reliable and can be used in the environments with noise and illumination change.

2. REVIEW OF PREVIOUS WORKS

Various methods have been proposed for the task of feature extraction such as edges [1], lines [2], corners and also regions [3]. Among these methods, corners are more popular and robust and a variety of algorithms have been proposed to extract corners.

*Received by the editors June 9, 2003 and in final revised form March 7, 2004
**Corresponding author

The Moravec operator [4] is one of the well-known interest point extractors. This operator extracts points which have higher intensity variations, however the variations are only measured in four directions. Therefore it is sensitive to strong edges only under certain directions and is very sensitive to noise. Another class of corner detection methods is based on a ‘cornerness’ approach. These methods involve using non-linear second derivative operators to detect high curvature regions in the image. The determinant of the Hessian matrix is a simple operator, ($C = I_{xx}I_{yy} - I_{xy}^2$), and gives a high response in the regions of image curvature. The method of Nobel [5] defines ‘cornerness’ measure as the change of gradient direction along an edge contour multiplied by local gradient magnitude. The expression for corner strength in this method is given as:

$$C = \frac{I_{xx}I_y^2 - 2I_{xy}I_xI_y + I_{yy}I_x^2}{I_x^2 + I_y^2} \tag{1}$$

where $I_x, I_y, I_{xx}, I_{yy}, I_{xy}$ are image first and second order derivatives.

The use of second order derivatives has, however, the property of amplifying image noise. This class of corner detection algorithms is also not very suitable for handling other types of junctions such as the T-junction. The Plessey algorithm (or Harris algorithm)[6] is another method of corner detection, which defines ‘the cornerness’ measure function to detect corners. To overcome the problem of the second order derivation, it uses first order derivatives to approximate the second order derivatives. However the algorithm is computationally expensive and the Gaussian smoothing is applied three times. The algorithm also doesn’t have good T-junction localization.

SUSAN [7] is another method for corner detection which doesn't use the derivatives of the image or edge pixels for corner detection. In this method a small disk-shaped mask is moved over the image pixels. The central point of the mask is called the Nucleus. The intensity value of the Nucleus is compared with other pixels in the mask. If the difference is less than a threshold, the pixel is categorized in a group called USAN. According to USAN values for different pixels, the locations of the edge pixels or corners are detected. Because this method doesn't use image derivatives, it has less sensitivity to image noises, especially snow noise.

The use of edge pixels and their curvature is another method of corner detection [8-12]. The CSS [8] method, a well-known corner detector of this class, is based on the Canny edge detector. This detector first extracts the edge contours from the input image using the Canny edge extractor and detects T-junctions, then using contours curvatures which are calculated in different scales, the remaining corners are detected.

Another class of feature detectors is based on the detection of features of interest using filters or spatial operators [13, 14]. A detection filter followed by post-processing and peak picking forms a sample feature detector of this class. The detection filter is usually a band-pass, which is formed from the difference of two low-pass Gaussian or Gabor filters.

The use of a covariance matrix of the gradient vector and applying canonical correlation is another method of corner detection [15]. In this method, two operators are obtained from a covariance matrix called PEG and QEG. Based on the values of PEG and QEG, the image is categorized to different regions, and then corners are detected. The method uses first order derivates of the image for corner extraction, however a three smoothing filter is required.

A large variety of methods have also been proposed for the task of feature matching. Among these methods, the similarity measure is one of the most powerful tools for feature matching [18, 19]. In order to find the corresponding point for a feature point using the similarity measure, a template window is considered around the feature point and this window is shifted pixel by pixel across a larger search window around an estimated corresponding point, and in each position, the similarity between the two regions is measured. The maximum or minimum value of the resultant measurements defines the position of the best match.

Normalized cross correlation and SSD [20] are well-known methods for measuring similarity between two regions. In addition to a normalized similarity value, normalized cross correlation has the advantage of being invariant to the linear change between the data sets, which makes the algorithm robust against low varying illumination change the scene.

Another strategy to find match points is the use of corners attributes. For example, in [22] the corner brightness and the x and y components of the position of USAN center of gravity are used for matching purposes. In these methods it is necessary for the corners to be detected for both images used for feature matching. The method has a low computation overhead, however the matching algorithm is sensitive to noise and illumination changes.

To increase the robustness of matching algorithms, other application dependent constraints such as target motion information in tracking applications [23] or epipolar constraints in stereovision [24, 25] may be used along with matching algorithms. One of the methods to increase the robustness of the matching process, especially in the tracking algorithms, is the use of statistical data association [26, 27]. In statistical data association the mach point and search area are first estimated using motion information. Then the real match point is found using a method like normalized cross correlation. However in non-uniform motions this may make the matching algorithm more erroneous.

3. EDGE FEATURE DETECTION

As mentioned before, noise and change in illumination of the images are some of the factors which may lessen the performance of feature matching algorithms. In order to deal with these difficulties, it is necessary that both feature detection and feature matching algorithms have less sensitivity to the presence of noise and illumination change. Edge points are proper features of image, which have less sensitivity to the image noise and illumination variance. However, when the edge points are in the form of straight lines or lines with lower curvature, the corresponding points can not be determined precisely, which is called an aperture problem [28]. To solve an aperture problem and determine the precise position of corresponding points, image locations should be selected as edge features, which have enough information for the matching process. The CSS corner detector is a suitable corner detector, which extracts the corners of the image from the contours of the edge-detected image. The CSS corner detector works as follows:

- Extract the edge contours from the input image using any good edge detector such as Canny.
- Fill small gaps in edge contours. When the gap forms a T-junction, mark it as a T-corner.
- Compute curvature on the edge contours at a high scale.
- The corner points are defined as the maxima of absolute curvature that are above a threshold value.
- Track the corners through multiple lower scales to improve localization.
- Compare T-corners to the corners found using the CSS procedure and remove very close corners.

Although the CSS corner detector considers the edge junctions and edge curvature, which are good features for edge matching, the curvature of only one contour is considered. In other words, each contour is handled separately for the purpose of corner detection. This enables the CSS algorithm to have good corner localization properties, however some of the features which are proper for matching are not detected. To detect more appropriate edge features, we have developed an edge feature detector algorithm which considers the accumulated curvature of edge pixels in the match window. The algorithm also considers the number of edge pixels in the match window, which is another useful factor for correct edge matching. Our algorithm consists of the following steps:

- Extract the edge contours from the input image using any good edge detector such as Canny.
- Fill small gaps in edge contours. When the gap forms a T-junction, mark it as a T-corner. To fill small gaps in the edge contour, we check the small windows (typically 5*5) centered on the end points of the contours. In the case of at least two other edge contours in the neighborhood of an end point, it is considered as a T-junction. If only one contour is found, two contours are merged by considering the pixels in the shortest distance between their end points as edge pixels.
- Calculate the curvature of Gaussian smoothed edge pixels. Because of the averaging property of accumulated curvature, in our algorithm it is not necessary to calculate the curvature at different

scales to reduce the noise effect. To calculate the curvature for edge pixels, each edge contour is represented as parametric vector $r(u) = (x(u), y(u))$. The vector is formed by assigning the value of $u = 0$ to the first pixel of the edge contour and the values $u = 1,2,..., N-1$ to other pixels of the edge contour, where N is the number of edge pixels in the contour.

Then the curvature is calculated using the following equations [8]:

$$\kappa(u,\sigma) = \frac{X_u(u,\sigma)Y_{uu}(u,\sigma) - X_{uu}(u,\sigma)Y_u(u,\sigma)}{(X_u(u,\sigma)^2 + Y_u(u,\sigma)^2)^{3/2}} \tag{2}$$

where

$$X_u(u,\sigma) = \frac{\partial}{\partial u}(x(u) \otimes g(u,\sigma)) = x(u) \otimes g_u(u,\sigma) \tag{3}$$

$$X_{uu}(u,\sigma) = \frac{\partial^2}{\partial u^2}(x(u) \otimes g(u,\sigma)) = x(u) \otimes g_{uu}(u,\sigma) \tag{4}$$

$$Y_u(u,\sigma) = y(u) \otimes g_u(u,\sigma), \qquad Y_{uu}(u,\sigma) = y(u) \otimes g_{uu}(u,\sigma) \tag{5}$$

$$g_u(u,\sigma) = \frac{\partial}{\partial u} g(u,\sigma), \qquad g_{uu}(u,\sigma) = \frac{\partial^2}{\partial u^2} g(u,\sigma) \tag{6}$$

where $g(u,\sigma)$ denotes a Gaussian function of width σ, and $\otimes$ represents one-dimensional convolution.

- Find the minimum and average of the absolute curvature values in the image and denote them as $\kappa_{\min}, \kappa_{ave}$.
- Assign the value of 0 to non-edge pixels and the value of $|\kappa(u,\sigma)| + \alpha\kappa_{ave} + (1-\alpha)\kappa_{\min}$ to edge pixels. Where α is a small positive number (typically $\alpha = 0.1$) and the constant term $\alpha\kappa_{ave} + (1-\alpha)\kappa_{\min}$, which is added to all edge pixels, aims to consider the density of edge pixels in the detection of proper edge features. Here $|\kappa(u,\sigma)|$ denotes the absolute value of $\kappa(u,\sigma)$.
- Scan the image and calculate the accumulated curvature for each pixel. The accumulated curvature is the sum of assigned values in the previous step and the summation is carried out in the small windows (typically 7*7) centered on pixels.
- Find local maximums of accumulated curvature values and apply a threshold to suppress weak features.

4. FUZZY-EDGE BASED FEATURE MATCHING

When the feature points are detected, the second step is to find the corresponding points of features in the second image. In addition to noise and illumination change of the scene, a change in geometry and the size of objects in the two images being matched are other factors which might deteriorate the efficiency of the matching process. For example, in tracking algorithms, a three-dimensional movement of the target or zoom operation of the camera may result in a geometry change of the target. Therefore the direct matching of edge pixels, such as edge correlation, is not a proper method to find corresponding points in these applications. To alleviate this problem we have utilized a new edge matching method called fuzzy-edge based feature matching. In this method edge pixels are described as fuzzy values, which are called fuzzy-edges. To describe edge points as fuzzy values we assign the value of 1 to edge points and a value between 0 and 1 to the points that are in the neighborhood of the edge points. This value is determined based on the distance from the edge points and the fuzzy membership function, which is used for this purpose. Different membership functions [29] such as triangular and trapezoid can be used for the construction of fuzzy-edges. Since a point may take different fuzzy values from its neighbors, the maximum of assigned fuzzy values is considered as a final value. Figure 1 shows a typical edge-detected image and related fuzzy-edges using triangular and trapezoid membership functions. The algorithm for generating fuzzy edges using a triangular membership function is given as follows:

- Scan the input image and find edge points. It is assumed that input image is a binary image and edge points have the value of 1.
- Consider square windows of size *b*b* centered on edge points, where *b* is the base for fuzzy membership function.
- Assign the fuzzy values for pixels in the windows according to the following equation

$$I(x+d_x, y+d_y) = \max\left[I(x+d_x, y+d_y), 1 - \frac{\max(|d_x|, |d_y|)}{b+1} \right] \tag{7}$$

where (x, y) are the coordinates of the center of windows, $-b/2 \le d_x \le b/2$, $-b/2 \le d_y \le b/2$, and I is the output image after fuzzification. The $|a|$ denotes the absolute value of a and $\max(a,b)$ represents the maximum values of a and b.

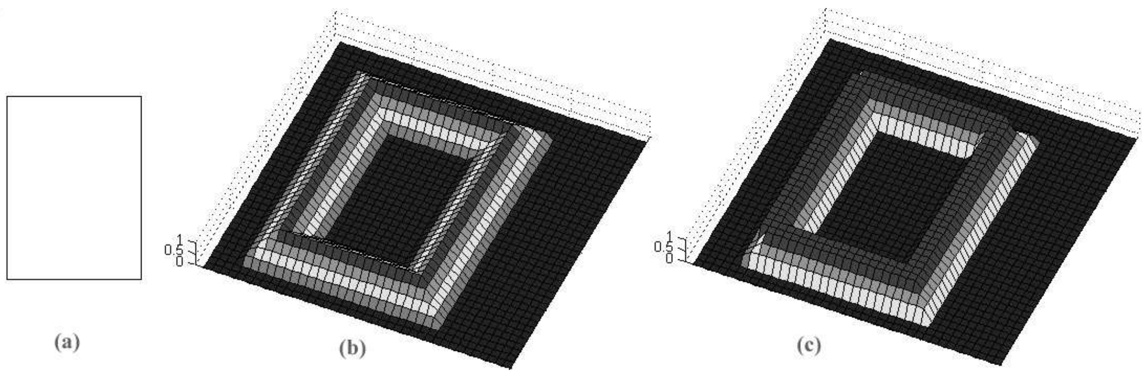

Fig. 1. a) An edge-detected image, b) fuzzy-edges using triangular membership function with the base of 7, c) fuzzy-edges using trapezoid membership function with the base of 7

When the fuzzy edges are obtained for both images being matched, an SSD or normalized correlation method is used to locate the corresponding points of features in the second image. Normalized correlation equation is given by [20]

$$r = \frac{\sum_{x,y\in S}[f_1(x,y)-\bar{f}_1][f_2(x,y)-\bar{f}_2]}{\left\{ \sum_{x,y\in S}[f_1(x,y)-\bar{f}_1]^2 \sum_{x,y\in S}[f_2(x,y)-\bar{f}_2]^2 \right\}^{1/2}} \tag{8}$$

Here $\bar{f}_1$ and $\bar{f}_2$ are the average of the fuzzy values in the two regions being compared, and the summations are carried out over all fuzzy values within small windows centered on the features.

a) Sub-pixel feature matching

The corresponding points, which are obtained using a similarity measure of fuzzy values in Eq. (10), have pixel resolution. In some of the applications it is also required to calculate the corresponding points with sub-pixel accuracy. Since our method is a kind of area-based matching, it is possible to calculate the sub-pixel corresponding points using parabola fitting. In our implementation we have used the least square parabola fitting on a similarity profile in the neighborhood of the optimum value to identify the sub-pixel peak location and calculate the sub-pixel corresponding point. To reduce the computational overhead of the fitting process, two one-dimensional parabola fittings for x and y directions are used.

5. INCREASING THE SPEED OF MATCHING ALGORITHM USING EDGE PYRAMID

In the case of large disparities, the search space for finding match points is large and this makes the computation overhead of the algorithm increase significantly. Multi-resolution feature matching is one of the most commonly used methods to overcome the problem of large disparities. In multi-resolution matching, different levels of resolution of the image are created, and the match is usually applied in the highest level and then propagated to the lowest levels. Since the disparities lessen at high levels, this method considerably reduces the computational overhead of the matching algorithm.

Our fuzzy-edge based matching algorithm can also benefit from using a multi-resolution method. Our algorithm for multi-resolution feature matching has the following steps:

- Build edge pyramids of images. The algorithm to construct the edge pyramid is explained below.
- Perform matching at lowest resolution.
- Propagate matches down to next higher resolution.
- Correct match estimates and continue to next level.

Suppose the edge detected image g_0 is represented initially by the array of edge contours

$$r_{0n}(u) = (x_{0n}(u), y_{0n}(u)), \qquad n = 1:N \tag{9}$$

where N is the number of edge contours in the image. Also, suppose that the image contains C columns and R rows of pixels. This image becomes the bottom or zero level of the edge pyramid. Pyramid level 1 image g_0, which contains $C/2$ columns and $R/2$ rows of pixels, are represented by

$$r_{1n}(u) = (x_{1n}(u), y_{1n}(u)) \qquad n = 1:N \tag{10}$$

where $r_{1n}(u)$ are reduced and low-pass filtered versions of $r_{0n}(u)$ are calculated as follows:

$$X_{0n}(u,\sigma) = x_{0n}(u) \otimes g(u,\sigma)\,, \qquad Y_{0n}(u,\sigma) = y_{0n}(u) \otimes g(u,\sigma) \tag{11}$$

$$R_{0n}(u,\sigma) = (X_{0n}(u,\sigma), Y_{0n}(u,\sigma))\,, \qquad r_{1n}(u) = R_{0n}(2u,\sigma)/2 = ((X_{0n}(2u,\sigma)/2, Y_{0n}(2u,\sigma)/2) \tag{12}$$

The pyramid level 2 image is then obtained from image level 1 by applying the same algorithm described above. The algorithm is repeated to build all levels of the edge pyramid. Figure 2 illustrates the contents of an edge pyramid generated with $\sigma = 2$.

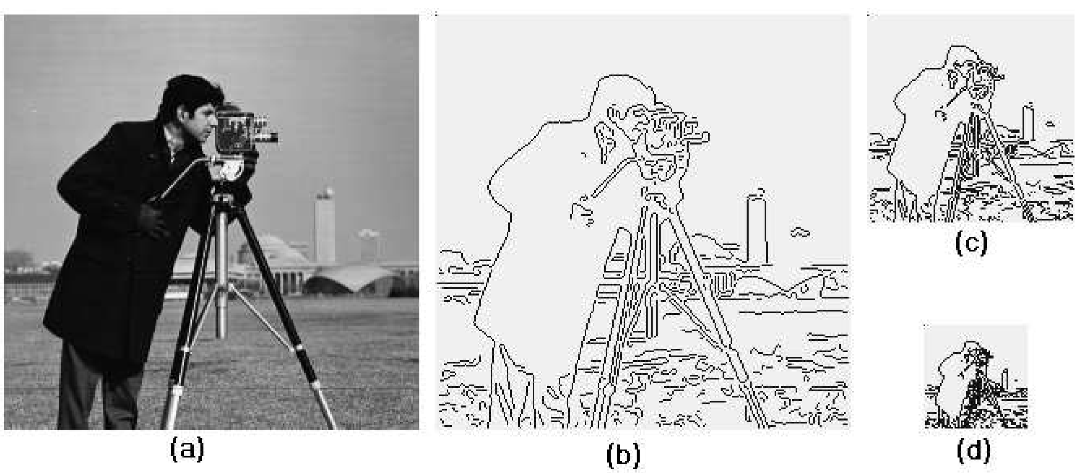

Fig. 2. Different levels of edge pyramid for a typical image. a) original image, b) edge image level 0, c) edge image level 1, d) edge image level 2

6. EXPERIMENTAL RESULTS

The proposed algorithms have been implemented on a Pentium III 500MHZ under a Windows 98 operating system using a Visual C++ program. We have tested the algorithms with different images including both simulated and actual sequences of images. The Canny edge detector is used for the extraction of edge points and gaps of 1 pixel wide are filled for the detection of edge features. Figure 3 shows the result of convolving Iran's map with a Gaussian function of different σ. Increasing σ results in smooth contours, hence more noise removing effects, however the contour fine details are lost. Our experimental results have shown that value of between 2 and 3 gives good results. Therefore, we have used the value of $\sigma = 2.5$ for our experiments. The two-level edge pyramid and triangular membership function with a base of 7 are used for the purpose of edge features correspondence in these experiments. We also tried to set other parameters and thresholds of the different algorithms for the best performance. To compare the efficiency of different matching algorithms, matching percentage is calculated where this percentage is given by [22]

$$matching\ percentage = \frac{(number\ of\ correct\ matches - number\ of\ incorrect\ matches)}{maximum\ possible\ number\ of\ correct\ matches} \tag{13}$$

To calculate the number of correct and incorrect matches, the affine motion model between two images is calculated using least median square (LMedS) algorithm [30,31], which gives robust motion

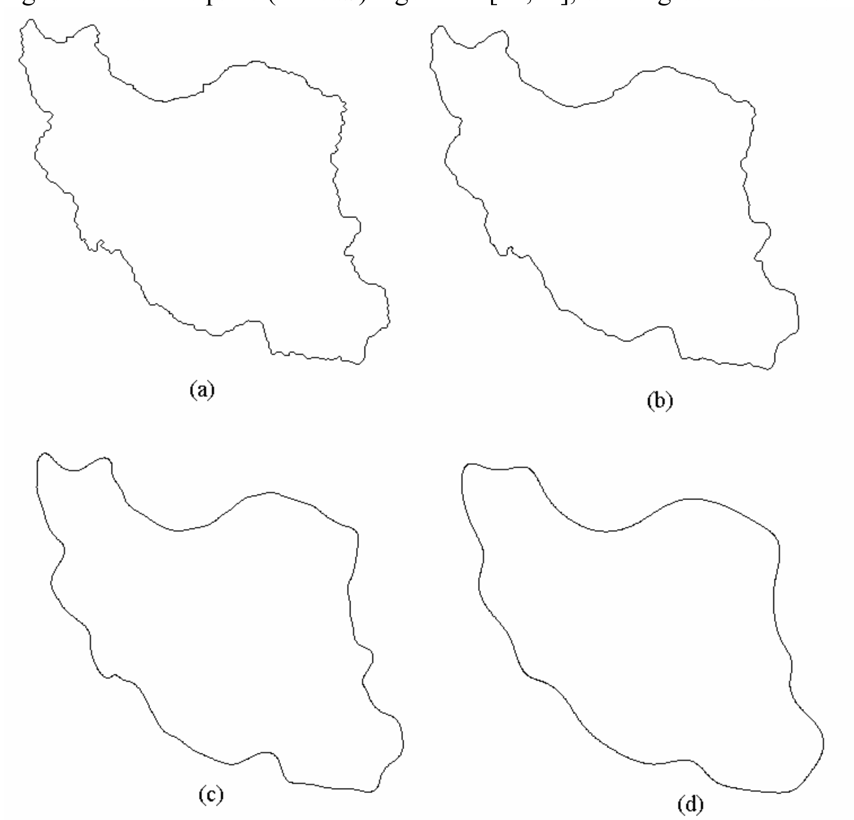

Fig. 3. The result of convolving Iran's map with Gaussian function of different σ,
a) original contour, b) $\sigma = 2$, c) $\sigma = 6$, d) $\sigma = 15$

model in the presence of outliers and incorrect matches. The robust standard deviation estimate for LMedS algorithm is given by [31]

$$\hat{\sigma} = 1.4826[1 + 5/(N - p)]\sqrt{M_J} \tag{14}$$

where M_J is the minimal median, p=3 for the affine model and N is the total number of features used for the calculation of the motion model. By using $\hat{\sigma}$, feature points, of which the residual values r_i satisfies the equation, $r_i^2 \le (2.5\hat{\sigma})^2$ are categorized as correct matched features. The results of the matching process are also shown graphically as vectors by the software, and a human operator checks these vectors for further robustness.

Figure 4 shows the first image sequence used for testing the proposed algorithms and comparing the results with those of other methods. Figure 5 depicts the matching percentage for different frames of the image sequence of Fig. 4. The results of four matching algorithms are shown in this figure, which are: 1- the proposed edge based feature detection and matching algorithms, 2- the CSS corner detector and proposed edge based feature matching algorithm, 3- SUSAN corner detector and correlation based feature matching [18], 4- matching using SUSAN corners properties [22]. As it is shown in this figure, the proposed algorithm has the best results.

To show the efficiency of the proposed algorithms in the presence of noise, the above experiment is repeated with additive Gaussian noises of 10dB. Figure 6 shows the results of different matching algorithms for the image sequence of Fig. 4 with additive Gaussian noises of 10dB. In Fig. 7, the results of different matching algorithms are shown for the image sequence of Fig. 4 with the addition of an illumination change. To generate images with an illumination change, the following equation is used [32]

$$I_{ill}(x,y) = (ax + by + c)I(x,y) + (dx + ey + c) \tag{15}$$

where $I(x,y)$ are input images, $a,b,c,d,e,$ and f are random numbers which are selected differently for each frame, and $I_{ill}(x,y)$ are the images with illumination changes.

As is shown in Figs. 5 to 7, in all cases, the proposed methods have good results. Figure 8 shows the matching percentage for different frames of the image sequence of Fig. 4 using edge correlation. In this experiment, the proposed edge feature detection algorithm is used to detect image features. Comparing the results of this figure with Fig. 5 shows that the fuzzy-edge based method has considerably enhanced the matching percentage.

Figure 9 shows another image sequence, which is used for comparison of different matching algorithms. This image sequence has lower motion speed compared to the image sequence of Figure 4. The matching percentages of different algorithms for the image sequence of Fig. 9 are shown in Figure 10.

One of the most important properties of the feature detection algorithms is the stability of the feature detection algorithm against noise. It means that the features should be detected consistently and should not move when multiple image of the same scene are captured (insensitive to variation of noise). Most published corner or point feature detectors have not used properly defined criteria for measuring stability. They have only shown the results corner detector in test images by adding noise and marking the corners. Trajkovic and Hedley [33] have used the measure of $S = \frac{C}{N}$ for stability, where N is the total number of features in the original image and C is the number features containing matches found in the noisy image. However this measure of stability doesn't consider features which may be detected in the noisy image because of noise. Tables 1 and 2 show the stability of different feature detection algorithms for Gaussian noise of 10dB and 7dB, respectively. To calculate the stability of different algorithms we have used the following equation:

$$Stability = \frac{2(C - NC)}{N_1 + N_2} \tag{16}$$

where N_1 and N_2 are the total number of detected features in noiseless and noisy images, respectively, C is the number of features in the noiseless image with matches found in noisy images, and NC is the number of features in the noiseless image, with matches not found in noisy images. As is shown in Table 1 and Table 2, our feature detection method has a good stability compared to other methods.

Figure 11 shows the results of applying various feature detection algorithms to a typical image. Please note that our feature detection aims to find feature points which are proper for feature matching and may detect points which are not corner.

Table 3 compares different feature detection algorithms with respect to their speed. The execution time of canny edge detector is shown in a different column and not considered in the calculation of the execution time of the CSS and our feature detection method. Table 4 shows the execution time for matching a feature point using different matching algorithms.

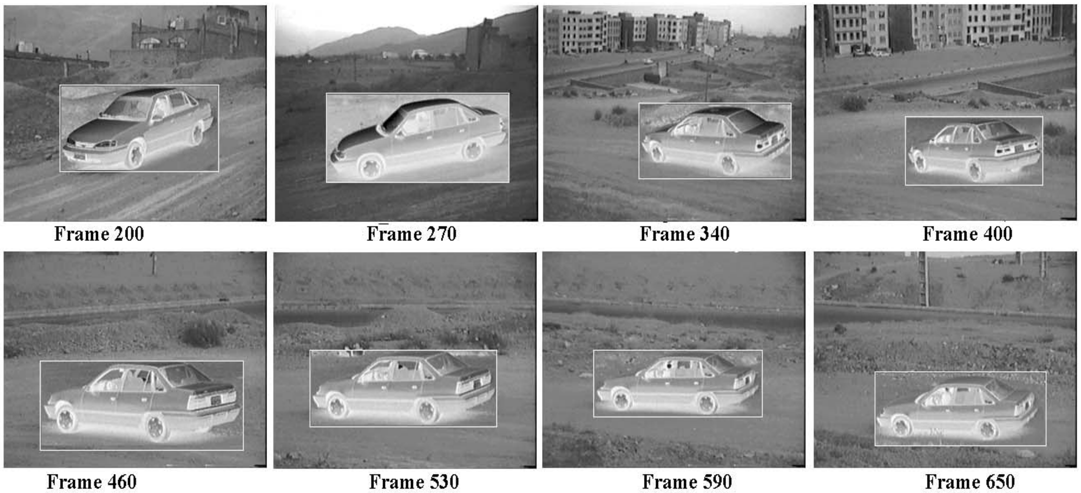

Fig. 4. Different frames of Car image sequence. Rectangular regions with inverted color show regions used for the calculation of matching percentage

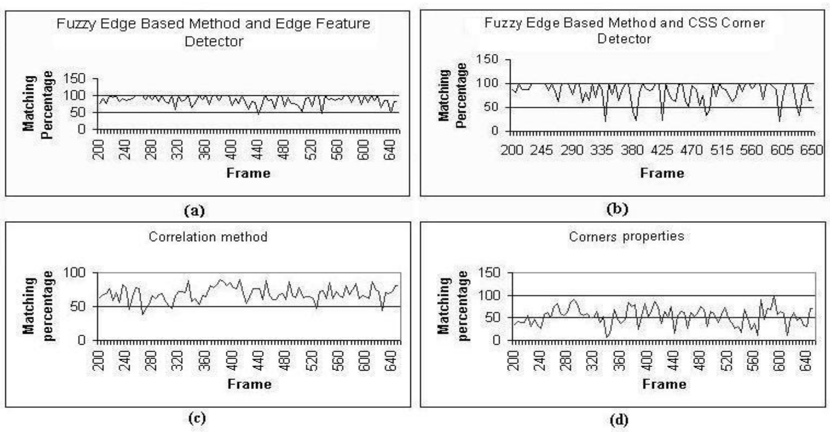

Fig. 5. Matching percentage for different frames of Car image sequence of Figure 4, a) fuzzy-edge based method and edge feature detector: average matching percentage=86.21, b) fuzzy-edge based method and CSS corner detector: average matching percentage=82.06, c) correlation method [18]: average matching percentage=68.72, d) matching using corners properties [22]: average matching percentage=53.48

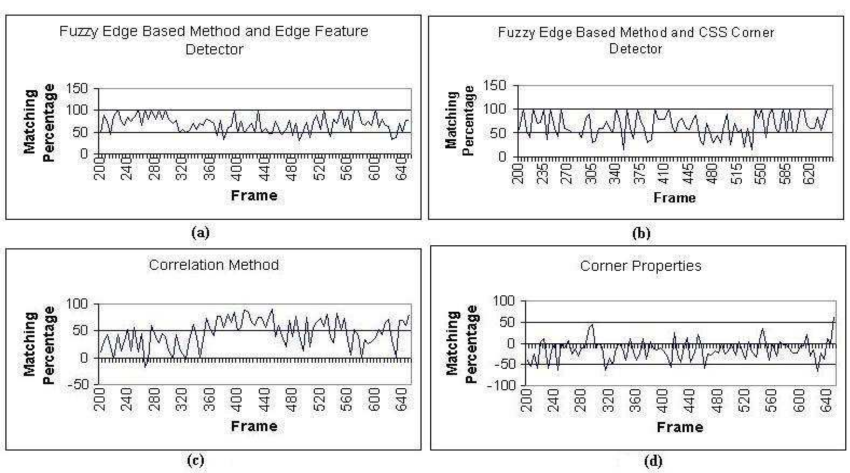

Fig. 6. Matching percentage for different frames of Car image sequence of Fig. 4 with additive Gaussian noise of 10dB, a) fuzzy-edge based method: average matching percentage=70.02, b) fuzzy-edge based method and CSS corner detector: average matching percentage=65.42, c) correlation method [18]: average matching percentage=45.13, d) matching using corners properties [22]: average matching percentage=-15.38

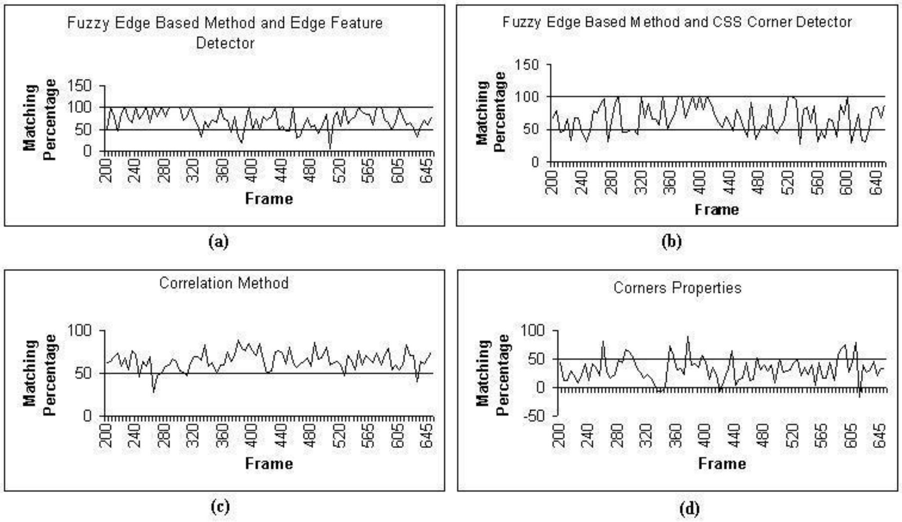

Fig. 7. Matching percentage for different frames of Car image sequence of Fig. 4 with illumination change addition, a) fuzzy-edge based method: average matching percentage=71.49, b) fuzzy-edge based method and CSS corner detector: average matching percentage=66.19, c) correlation method [18]: average matching percentage=64.97, d) matching using corners properties [22]: average matching percentage=32.37

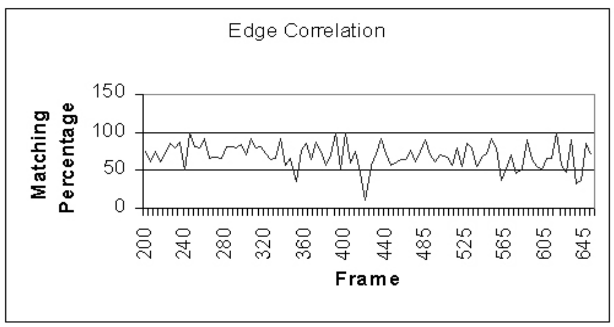

Fig. 8. Matching percentage for different frames of Car image sequence of Fig. 4 using proposed edge features detection algorithm and edge correlation, average matching percentage=70.05

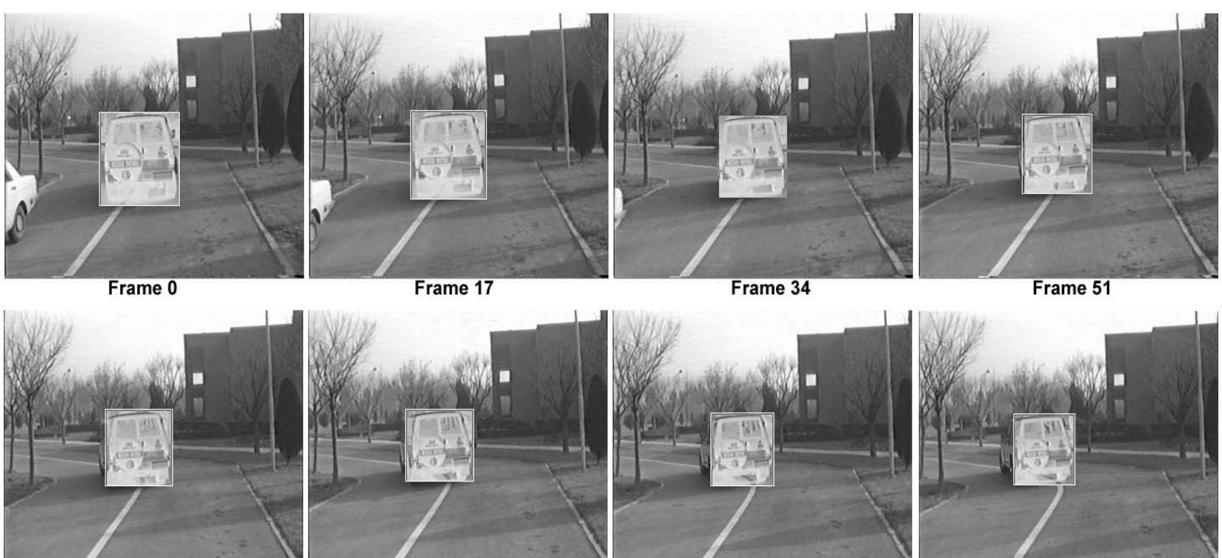

Fig. 9. Different frames of Patrol image sequence. Rectangular regions with inverted color show regions used for the calculation of matching percentage

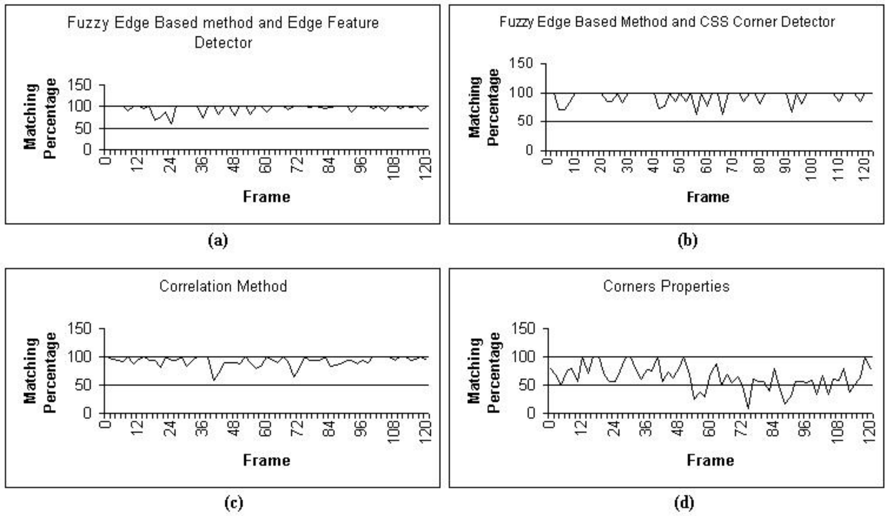

Fig. 10. Matching percentage for different frames of Patrol image sequence of Fig. 9, a) fuzzy-edge based method: average matching percentage=95.51, b) fuzzy-edge based method and CSS corner detector: average matching percentage=93.44, c) correlation method [18]: average matching percentage=92.205, d) matching using corners properties [22]: average matching percentage=63.46

Table 1. Stability of feature detection algorithms for Gaussian noise of 10dB

Method / Image	SUSAN [7]	Moravec [4]	Plessey [6]	CSS [8]	Our method
Cameraman	0.11	0.05	0.2	0.29	0.44
Lena	0.12	0.04	0.19	0.25	0.32
Car	0.12	0.06	0.25	0.04	0.21
Patrol	0.14	0.02	0.22	0.08	0.27

Table 2. Stability of feature detection algorithms for Gaussian noise of 7dB

Method / Image	SUSAN [7]	Moravec [4]	Plessey [6]	CSS [8]	Our method
Cameraman	0.05	-0.04	0.12	0.09	0.35
Lena	0.07	-0.09	0.1	0.1	0.2
Car	0.08	-0.06	0.18	-0.08	0.1
Patrol	0.09	-0.075	0.13	-0.05	0.14

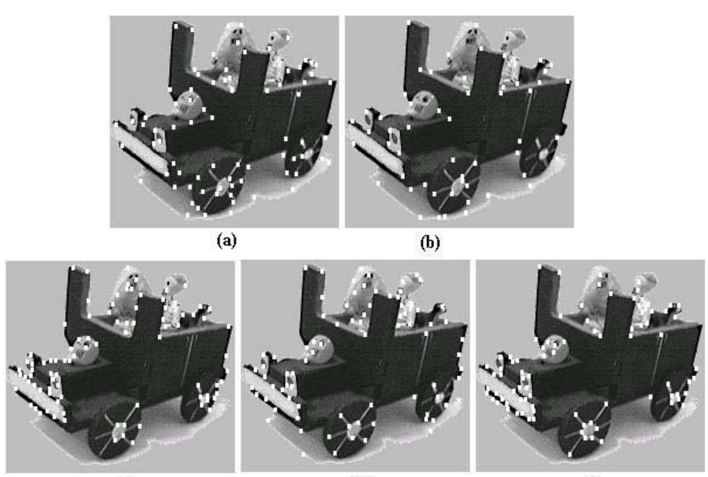

Fig. 11. Results of applying various feature detection algorithms, a) proposed edge feature detector, b) CSS corner detector, c) SUSAN corner detector, d) Moravec algorithm, e) Plessey algorithm

Table 3. Execution time for different corner detection algorithm

Method / Image	SUSAN [7]	Moravec [4]	Plessey [6]	CSS [8]	Our method	Canny edge detector
Cameraman	50ms	120ms	331ms	200ms	250ms	250ms
Lena	50ms	120ms	331ms	200ms	250ms	250ms
Car	71ms	170ms	490ms	1035ms	1051ms	351ms
Patrol	60ms	191ms	515ms	580ms	791ms	381ms

Table 4. Execution time for matching of a feature point using different matching algorithm

Search area / Method	20*20	15*15	10*10
Correlation	3.46ms	1.86ms	1.02ms
Corners properties	0.14ms	0.12ms	0.11ms
Fuzzy-edge based	2.97ms	1.83ms	1.27ms
Fuzzy-edge based with 2 level edge pyramid	1.56ms	1.34ms	1.12ms

7. CONCLUSION

In this paper, new methods for detection and matching edge features were presented. To detect features, we considered edge points and their accumulated curvatures. When the edge features are detected, they are matched with points in the second image using fuzzy-edge based feature matching. To increase the speed of matching algorithms we proposed a new multi-resolution based algorithm which utilizes edge pyramids. The comparison of the results with those of other methods has shown that more reliable results can be obtained with the aid of the proposed methods. The proposed methods also have good results in the case of noisy images or images with illumination change. The proposed algorithms can be used in various machine vision applications such as target tracking, stereovision and image registration.

REFERENCES

1. Canny, J. (1986). A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, *8*(6), 679-698.
2. Van der Heijden, F. (1995). Edge and line feature extraction based on covariance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *17*, 16-33.
3. Nerriec, C., Diez, A., Alvarez, I., Cancelas, J. A. & Gonzalez, R. (1994). A syntactic and contextual edge detector and application of a rewriting rule set to improve detection. *20th International Conference on Industrial Electronics, Control and Instrumentation (IECON '94)*, 994-997 Bologna, Italy.
4. Moravec, H. P. (1977). Towards automatic visual obstacle avoidance. *In Proc. of the International Joint Conference on Artificial Intelligence*, p. 584, Cambridge, MA USA.
5. Nobel, J. A. (1988). Finding corners. *Image and Vision Computing*, *6*, 121-127.
6. Harris, C. G. & Stephens, M. (1988). A combined corner and edge detector. *In 4th Alvey Vision Conference,* 147-151, Manchester, England.
7. Smith, S. M. & Brady, J. M. (1997). SUSAN - a new approach to low level image processing. *Int. Journal of Computer Vision*, *23*(1), 45-78.
8. Mokhtarian, F. & Suomela, R. (1998). Robust image corner detection through curvature scale space.' *IEEE Trans. Pattern Analysis and Machine Intelligence*, *20*(12), 1376-1381.
9. Nanzhi, J., Wu, R. & Li, J. (2001). Super resolution feature extraction of moving targets. *IEEE Transactions on Aerospace and Electronic Systems*, *37*(3), 781-793.
10. Bandera, A., Urdiales, C., Arrebola, F. & Sandoval, F. (2000). Corner detection by means of adaptively estimated curvature function. *Electronics Letters*, *36*(2), 124-126.

11. Park, K. H., Cho, H. J., Kee, C. D. & Jeong, S. H. (2001). Vision based position control of robot manipulator using by wavelet based feature extraction. *Proceeding of the IEEE International Symposium on Industrial Electronics (ISIE 2001), 1*, 237-242.
12. Quddus, A. & Gabbouj, M. (2000). Wavelet based corner detection using singular value decomposition method. *Proceeding of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2000), 4*, 2227-2230.
13. Irene, Y. H. G. & Tardi T. (2002). Multiresolution feature detection using a family of isotropic bandpass filters. *IEEE Transaction on System, Man and Cybernetics*—Part B: Cybernetics, *32*(4), 443-454.
14. Quddus A., & Fahmy, M. M. (1998). Corner detection using Gabor-type filtering,. *Proceeding of the IEEE International Symposium on Circuits and Systems (ISCAS'98),* 150–153.
15. Shigeru, A. (2000). Image field categorization and edge/corner detection from gradient covariance. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *22*(2), 179-190.
16. Jensfelt, P. & Kristensen, S. (2001). Active global localization for a mobile robot using multiple hypothesis tracking. *IEEE Transactions on Robotics and Automation*, *17*(5), 748-760.
17. Schulz, D., Burgard, W., Fox, D. & Cremers, A. B. (2001). Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. *Proceedings of International Conference on Robotics and Automation (ICRA 2001), 2*, 1665-1670.
18. Araki, S., Matsuoaka, T., Yokoya, N. & Takemura, H. (2000). Real-time tracking of multiple moving object contours in a moving camera image sequence. *IEICE Trans. Inf. & Syst.*, *E83-D*(7).
19. Hager, G. D. & Belhumeur, P. N. (1998). Efficient region tracking with parametric models of geometry and illumination. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *20*(10).
20. Jain, R., Kasturi, R. & Schunck, B. G. (1995). *Machine vision*. McGraw-Hill Inc.
21. Weng, J., Ahuja, N. & Huang, T. S. (1992). Matching two perspective views. *IEEE Trans. Patt. Anal. Machine Intell., 14*, 806-825.
22. Smith, S. M. & Brady, J. M. (1995). ASSET-2: real-time motion segmentation and shape tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *17*, 814-820.
23. Yao, Y. S. & Chellappa, R. (1995). Tracking a dynamic set of feature points. *IEEE Transaction. On Image Processing*, *4*(10), 1382-1395.
24. Candocia, F. & Adjouadi, M. (1997). A similarity measure for stereo feature matching. *IEEE Transaction On Image Processing*, *6*(10), 1460-1464.
25. Braunegg, D. J. (1990). Stereo feature matching in disparity space. *Proceedings of IEEE International Conference on Robotics and Automation, 2*, 796- 803.
26. Arras, K. O., Castellanos, J. A. & Siegwart, R. (2002). Feature-based multi-hypothesis localization and tracking for mobile robots using geometric constraints. *IEEE International Conference on Robotics and Automation (ICRA 2002)*, *2*, 1372-1377.
27. Cox, I. J. & Hingorani, S. L. (1996). An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 18*(2), 138 –150.
28. Barron, J. L., Fleet, D. J. & Beauchemin, S. S. (1994). Performance of optical flow techniques. *Int. Journal of Computer Vision, 12*(1), 43-77.
29. Wang, L. X. (1997). *A course in fuzzy systems and control*. Prentice-Hall international Inc.
30. Behrad, A., Shahrokni, A., Motamedi, S. A. & Madani, K. (2001). A robust vision-based moving target detection and tracking system. *Proceeding of Image and Vision Computing conference (IVCNZ2001)*, University of Otago, Dunedin, New Zealand 26 th –28 th November.
31. Rousseeuw, P. J. & Leroy, A. M. (1987). *Robust regression and outlier detection*. John Wiley & Sons, New York.
32. Lai, S. H. & Fang, M. (1999). Robust and efficient image alignment with spatially varying illumination models. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2*, 167-172.
33. Trajkovic, M. & Hedley, M. (1998). Fast corner detection. *Image and Vision Computing, 16*(2), 75-87.

34. Behrad, A., Motamedi, S. A., Madani, K. & Esnaashari, M. (2002). A new algorithm for target tracking using fuzzy-edge-based feature matching and robust statistic. *IEEE International Conference on Image Processing ICIP2002*, Sept. 22-25, Rochester, New York USA.
35. Kass, M., Witkin, A. & Terzopoulos, D. (1987). Snakes: Active contour models. *International Journal of Computer Vision., 1*(4), 321-331.
36. Behrad, A. & Motamedi, S. A. (2003). Multiple moving targets detection in a mobile video camera image sequence using fuzzy-edge based feature matching. *International Conference on Fuzzy Information Processing Theories and Applications,* March 1-4, Beijing, China.