Engineering Letters, 14:1, EL_14_1_27 (Advance online publication: 12 February 2007)

Vision Based Target-Tracking Realized with Mobile Robots using Extended Kalman Filter

Sheli Sinha Chaudhuri and Amit Konar

Abstract— **The paper presents a novel scheme for target-tracking realized with two mobile robots, where one robot is configured as tracker and the other as moving target. Fuzzy C-means clustering algorithm has been employed here to segment the target robot in images grabbed by the tracker. A new localization algorithm has also been proposed to determine the location of the target in the segmented images. An extended Kalman filter has been employed here for predicting the direction of motion of the moving target from its current and last few positions. The robot is pre-trained with back-propagation learning algorithm to plan its trajectory amidst obstacles. The pre-trained neural net is used in target-tracking application to control the motion of the tracker in the predicted direction of the moving target. Performance of the proposed neuro-Kalman synergism in target-tracking has experimentally been found to be superior to a tracking scheme without prediction by Kalman filter.**

Index Terms— **Fuzzy c-means, Kalman Filter, Robotic Vision, Target Tracking.**

I. INTRODUCTION

The paper provides a novel approach to target-tracking realized with two mobile robots, where one robot acts as a moving target and the other plays the role of a tracker. The target robot is controlled manually, while the tracker has to predict the motion of the target and plan its trajectory to meet the target. The target-tracking problem has many interesting applications in defense, such as missile and aircraft tracking .The problem addressed in this paper, however, is restricted to 2-D only for simplicity of realization with mobile robots. The tracking problem in the present context is concerned with: i) prediction of the next location of the tracker from its previous and current positions and ii) controlling the motion of the tracker towards the predicted location of the target. Nomad Super Scout II mobile robots have been used to realize the proposed target-tracking scheme. These robots have a fixed camera mounted on top of the robot and 16 sonar transducers mounted around the cylindrical structure of the robot. To determine the current location of the moving target, the tracker rolls around its z-axis to grab an image of its target on the camera and then determine the radial distance of the target by activating the sonar sensors below the camera. The well-known extended Kalman filter algorithm has been employed in this paper to determine the next position of the moving target from it current and last few positions. After the next position of the target is predicted, a pre-trained back propagation neural net based algorithm is invoked to control the motion of the tracker towards the predicted location of the target.

To determine the next position of the target from its current and previous few positions, we need to recognize the target from its visual image, which is grabbed by the tracker. The recognition process includes an image segmentation algorithm realized with the well-known fuzzy C-means clustering technique, and followed by a novel localization algorithm, to be addressed in Section 2. After localization is over, an image matching algorithm, proposed by our research team elsewhere [2] is used to recognize the target robot in the localized position of the possible target region.

Significant work has already been undertaken by our research team to identify the most appropriate neural algorithm for motion planning by a mobile robot [9]. The benchmark analysis performed by our research team suggests that the Back-propagation neural net algorithm is most efficient for motion planning with reference to both the shortest path and the shortest time. The same algorithm has therefore been employed in the present context to move the tracker towards the predicted position of the target.

Some traces of progresses in target-tracking by mobile robots have been reported in the current literature on mobile robotics. For example, Ollero and Garcia-Cerezo in a recent book paper [16], provide a novel approach to target tracking by mobile robots using a fuzzy tracking controller. The merit of their tracking scheme lies in automatic updating of the look-ahead distance of the target with the help of a fuzzy predictor. Hitchings et a*l*. in a recent paper presented a motion-tracking scheme of two co-operative robotic agents, where the leader is the target and the follower is the tracker. Experiments undertaken by these researchers reveal that the follower can track both the linear and the curved motion of the moving leader [6]. Besides robotic applications, there exists a vast literature on target-tracking in applied engineering literature. Well-known stochastic algorithms, such as Bayesian tracking [19,20] and Kalman Filtering [4] have successfully been used for multi-agent tracking in mobile robotics. A complete listing of the relevant references will be too large to be presented in this paper. Some of the popular works that need special mention include tracking using a non-linear filter based on *portraying min principle* [1]. Optimum tracking of a

Manuscript received April, 2006.

F. A. Author is with the E.T.C.E. Department of Jadavpur University, Kolkata-700032, India (phone:-033-24146002, e-mail: shelism@rediffmail.com).

S. B. Author is with the E.T.C.E Department of Jadavpur University, Kolkata-700032, India (e-mail: konaramit@yahoo.co.in)

maneuvering target in clutter [8], neural net based tracking [11], integer programming based target tracking [13] and tracking and classifying without *a priori* identification [14]. Shirai et a*l*. introduced a novel method for visual tracking using the well-known principles of optical flow techniques [18]. This scheme is useful for its inherent robustness, and has many applications in real-time human tracking system in cluttered background. The whole work presented in this paper, however, is novel from the point of view of the technological merits of synergism of Kalman filter and neural networks for tracking applications in mobile robotics.

The paper has been divided into six sections. Section 2 provides the experimental details of image segmentation and object localization. In section 3, we provide the prediction scheme of the extended Kalman filter and demonstrate its role in determining the next position of the target from its current and previous few positions. In section 4, the back-Propagation neural net algorithm has been briefly outlined to illustrate its application in motion planning of the tracker. Experimental results and their interpretations are included in section 5, and conclusions are listed in section 6.

II. IMAGE SEGMENTATION USING FUZZY C-MEANS CLUSTERING ALGORITHM

A monochrome digital image usually is a two-dimensional array of gray pixels. On occasions, the components of the image are needed to be isolated. The process of isolating important regions of an image into components/ modules is generally referred to as image segmentation. A number of well-known algorithms of image segmentation is available in any textbook of image processing [12] [17] [7]. In this section, we following Raghukrishnapuram [10] present a study of image segmentation [12] using fuzzy c-means algorithm.

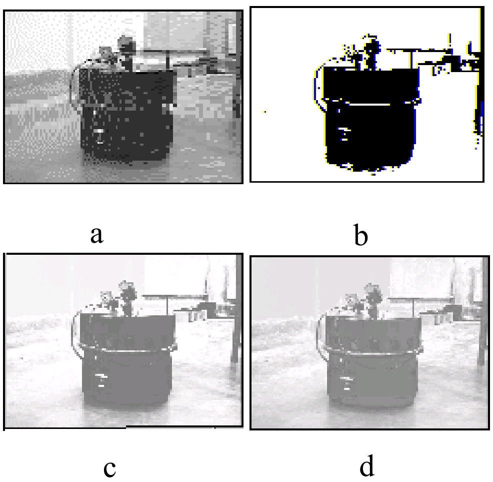


Fig. 1: Fuzzy C -means clustering algorithm applied to segment the given image (a) into 2 clusters with m=1.01 in (b), m=1.2 in (c), and m=2.5 in (d).

The fuzzy c-means clustering algorithm for image segmentation has 2 parameters, namely, the number of clusters, denoted by ‘c’, and the exponential weighting factor ‘m’ over the membership functions. The experiments are performed by gradually varying these two parameters and their effects on clustering are noted. The number of clusters needed is usually determined by the problem in hand. For segregating a dark object from a light background or vice versa, we should select c=2 and thus obtain 2 clusters, one corresponding to the dark region and the other to the lighter region. Fig 1 shows the results of clustering with c=2. Further, the value of exponential weighting factor m has been increased in steps from m slightly greater than 1, followed by m=1.2 and m=2.5. The variation of m clearly indicates the difference between a hard cluster and a soft cluster.

For the purpose of illustration, we have constructed the figures using the membership value of each pixel mapped to gray value levels. The contrasting shades in Fig. 2(b) indicate that each pixel belongs to either of the classes with large membership value and with a very small membership value for the other class. This is typical of a hard cluster where the pixels are assigned to either of the two classes. In the subsequent figures 2(c) and 2(d), the 2 shades become less contrasting. As the shades are representing the membership values, it means that each pixel now have intermediate membership values of belonging to the 2 classes. This is fuzzy clustering where each pixel has finite memberships of belonging to the 2 classes. Thus, we have greater latitude of deciding which pixels to select based on their membership values.

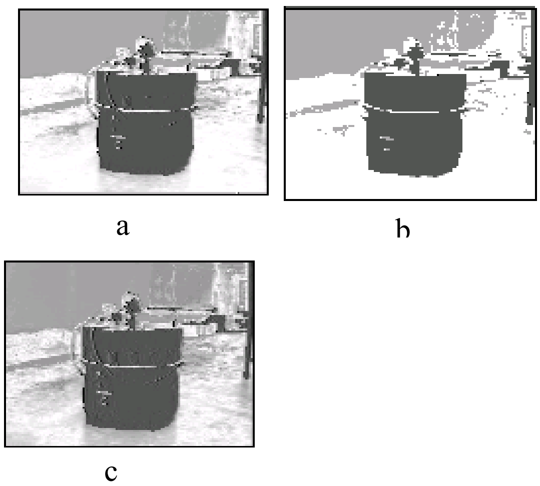
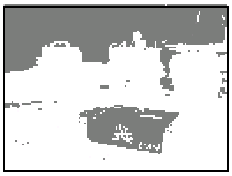


Fig. 2: Fuzzy C-means clustering algorithm applied to segment the image in 2(a) into 3 Clusters with m=1.01 in (a), m=1.2 in (b) and m=2.5 in (c).

In Fig. 3, the number of clusters is 3 and they are represented by 3 shades: dark, intermediate gray and light. Here, too, we observe that how the value of m alters the final membership values of the pixels. As explained in Fig. 2, a value of m close to 1 (m=1.01) generates crisp clusters while m=1.2 or larger makes the clusters fuzzy.

In all the above experiments, we used gray level of the pixels as the feature of interest and accordingly constructed the feature vector. In Fig. 3(b), we observe that although the clustering is done, we do not get the cluster as expected intuitively. This may be attributed to the fact that the general illumination of the image is so low that the black objects cannot be distinguished from the surrounding floor based on gray level alone. We propose that in such circumstances, some transformation of the gray level values be taken to construct the feature vector. The choice of this transformation is based on some prior knowledge of the object of interest.


a


b

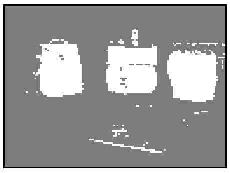

c

Fig. 3: Fuzzy C-means clustering algorithm applied to isolate very black objects from the image in (a) with grayscale as the feature vector in (b), logarithm of the grayscale as the feature vector in (c).

To illustrate the above point, let us consider the case where we are interested in segregating some dark object from a poorly illuminated workspace. The suitable transformation in such a circumstance is to take the logarithm of the gray level values of the pixels. As we know, logarithmic transform expands and emphasizes the lower values and cramps the larger value in a smaller range, thus de-emphasizing them. Fig. 3(c) illustrates the process of clustering with gray levels as the elements of the feature vector. It is now observed that this transformation has successfully clustered the 'very' dark objects from the dark environment. Thus, we have a third controlling factor, namely, the transformation on the gray level, whereby we may classify the image according to the requirements of the problem in hand.

III. OBJECT LOCALIZATION

Once the robot and other objects, which have sufficiently low gray scale values, are segmented from the image by FCM clustering algorithm, the task that remains at hand is to localize the robot solely from the other black objects. This is however difficult because it involves separating the geographically isolated objects in the image. The most widely used technique for object localization is region growing. In the present scenario, the conventional region-growing technique fails because the seed point, which needs to be initialized in the algorithm, cannot always be located within our object of interest. Moreover, high time complexity of the algorithm renders it automatically vulnerable for real-time applications like navigation or target tracking.

After considering all these constraints, a novel algorithm for object localization has been devised [5]. One useful feature that is used at this point is the size information of the object under consideration. The essence of this algorithm lies in splitting the FCM clustered image into some fixed size windows/blocks and determining whether each window is dark black, indicating the robot region. The next task is to identify the connected windows by testing the 4-connectivity between them. Then the 4-connected windows are clubbed together to form the separate geographic regions.

The selection of the experimental window size is a pertinent parameter to be determined judiciously. A large sized window may include more than one object. For example, a large window may include the target object with others. A small window size is good for accuracy, but a too small window demands a significant computational time. The size of the window chosen in the present application is of 12×16 pixels. In fact, the choice of 4-connectivity testing over that of the 8-connectivity testing for identifying geographically connected regions is also guided by the same consideration that the whole computation process should be amenable to a real-time application. A procedure is presented below, describing the various steps of the object localization process.

PROCEDURE LOCALIZE-OBJECT

Input: FCM clustered image (containing the robot and other black objects as one cluster and every other object as another cluster)

Output: An image containing localized target object, well separated from noise and other objects having similar type of coloration as that of the target

Begin

Step1: partition the input image into (m x n) number of equal sized blocks;

Step2: For each block (i, j) **Do**
    Begin
        For each pixel within a block **Do**
        Begin
            If intensity of pixel <=20
            Then declare it dark and increase dark_pixel_count_block (i, j) by 1;
    End For;

Step3: If dark_pixel_count_block (i, j) >50
    Then mark block (i, j) as dark;
    End For;

Step4: k: =1;
    Repeat
    Region[k]= any dark block(i, j);
    Region_count[k]: =0;
    For each dark block (i, j) in Region[k]
    If its neighbors are dark,
    Then do
    Begin
    Region[k]: =Region[k] U (block(i, j));
    Region_count[k]: = Region_count[k] +1;

End For;
Mark Region[k] dark;
k: =k+1;
Until no dark blocks remain;
K: $=k_{max}$;

Step 5: For k: =1 to k_{max}
Identify the Region with the largest
Region_count[k] and
Call it target;
End For;

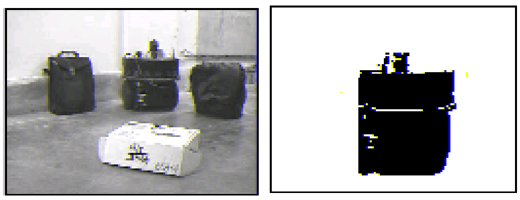
End.

The procedure localize-object aims at identifying the locations of the target object in a segmented image. in the present context, the target object has been assumed to occupy the largest area. thus identifying the largest dark region in the image suffices our purpose.

The procedure comprises of 5 main steps. In step 1, we partition the given segmented image into m.n number of equal sized blocks. Step 2 of the algorithm identifies dark pixel based on their intensity values. For this implementation, we select a threshold of 20. Thus if pixel intensity is less than 20, it is declared dark. The dark pixel count in each block is also determined in this step. Step 3 marks a block dark if its dark pixel count exceeds 50. Step 4 of the procedure assembles neighborhood dark blocks in the segmented image into regions such that each two regions are disjoint. In step 5 of the algorithm we declare the largest region as the target object. The localization of the robot from the segmented image of Fig. 4(a) is presented in Fig. 4(b) for convenience.

In case the largest region does not correspond to the target object, the regions need to be sorted in descending order based on their block counts. Now, a shape-matching algorithm may be invoked for comparing the boundary of the target object with each region in the list in sequence. The region having the closest resemblance with the reference object shape is declared as the target.

The process of image localization presented in this section has been utilized to build a real-time system for vision based target tracking and co-operation schemes in mobile robotics. The novelty of the approach lies in a successful merging of the FCM algorithm for image segmentation with the "windowing-and-connectivity-checking" method for image localization.



a b

Fig. 4: The target robot (b) has been isolated from the image (a) by "windowing and connectivity checking".

One interesting and noteworthy issue about the application of the FCM algorithm is the selection of the logarithm of the pixel intensities as the feature. Such selection is important for segmenting a dark target from a background of comparable (but unequal) darkness. If gray scale intensities, instead of their logarithms, are directly used as the feature, then segmentation of comparable dark regions in an image is not feasible.

IV. TARGET TRACKING AND INTERCEPTION BY MOBILE ROBOTS USING KALMAN FILTERING

Target tracking is a problem of common interest for researchers of numerous domains. The concept of target tracking classically emerged from the disciplines of control engineering. The concerned problem in the present context is to predict the online trajectory of a moving target by a given tracker. Usually, the speed of response of the tracker is higher or at least comparable to that of the target. Since the intelligent target and the tracker both use the same level of technology, it is expected that their speed of response is more or less comparable. Designing an intelligent tracker under such a circumstance is really a complex problem. In this section, we present the design and implementation aspects of an intelligent tracker. The tracker employs a video camera to capture the images of the moving target for segmentation and subsequent localization of the target in its image. It then identifies the location of the target by a range finder and consequently plans a path towards the target by using the knowledge of the obstacle map in its workspace. An overview of the proposed scheme of target tracking is outlined below.

For realization of the target-tracking scheme, 2 mobile robots identical in all respects have been configured as the target and the tracker. Each robot has its own desktop server and is equipped with a movie type video camera, a video frame grabber, a radio communication system and 16 ultrasonic transducers mounted around the periphery of the robot at uniform spacing. The target robot is controlled to move on a fixed trajectory by a control program running at its desktop server. The tracker robot on the other hand receives sensory information by using the video camera and the ultrasonic sensors. The received real time video frames collected by the tracker robot are first transferred to its server. The server preprocesses the image and segments it into objects of interest (here the target robot). A fuzzy clustering technique is then invoked for segmentation and subsequent localization of the target. After the target is identified in the image, its distance and orientation with respect to some reference axis of the tracker needs to be determined. The following scheme has been undertaken to evaluate the polar co-ordinate of the target with reference to two mutually perpendicular axes of the tracker.

First, the tracker robot should activate its sonar transducers. The particular sonar transducer radiating its beam towards the target gives an approximate measure of the range of the target with respect to the tracker. The polar co-ordinate (r, θ) of the target with respect to a reference x-axis, say $Sonar_{12}$-axis, of the tracker is then determined.

The sonar readings in all other directions of the tracker describe the range of the obstacles in the tracker's world map. These readings along with the estimated range of the target are supplied as input to a pre-trained neural net for controlling the speed and direction of motion of the tracker. Since the control decision about the motion generated by the neural net is based on the sampled (sonar/ vision) data of the last cycle, the tracker may occasionally be misled by wrong control commands. To overcome the limitation, a 'tracker motion predictor' may be employed in the system. The predictor should provide the possible current direction of motion of the target from its preceding positions. This has been realized by embedding an extended Kalman filter in the proposed scheme. A complete schematic diagram of the proposed tracking system is presented in Fig. 5.

The proposed work has a number of merits with respect to traditional target tracking systems. First, most traditional systems are designed with the pre-assumption that the speed of the target is less than that of the tracker. Fortunately, the proposed design is free from such constraints. Secondly, conventional trackers usually do not estimate the range of the target, and therefore do not pay much attention to velocity modulation of the tracker. The present work, however, takes care of the range measurement of the dynamic target, and consequently adjusts the velocity of the tracker on-line. Thirdly, accuracy in range estimation along near-linear trajectories has shown significant improvement in this presentation by employing extended Kalman filtering. Lastly, the proposed tracker requires insignificantly small time of the order of 400 milliseconds only to respond to a change in the target position. Finally, the back-propagation algorithm being trained with quite a large number of sensory-response instances selects the right step of discrete motion of the tracker. The min-square error in position of the tracker with respect to the dynamic target position thus diminishes in most cases.

IV.1 Measurements of the Input to Kalman Filter

The Kalman filter employed in the tracking system can predict the current position of the target from its preceding positions. The accuracy in prediction by a Kalman filter greatly depends on the time gap between successive data samples. The time needed for on-line image registration, segmentation, localization and neural net activation being of the order of 400 milliseconds, the sampling interval cannot be fixed less than 400 milliseconds. The prediction of the current position of the target from its last, say 3 positions indirectly means exciting the filter with the sonar data collected (400 × 3) milliseconds = 1.2 seconds earlier. Since the speed of the robots is considerably high (around 20 inches/ second), the predicted current position of the target may suffer from inaccuracy. To overcome the above problem the tracker is designed to work in 2 phases.

In the first phase, the tracker is given a controlled rotation around its z-axis so that it can direct its vision system to grab 3

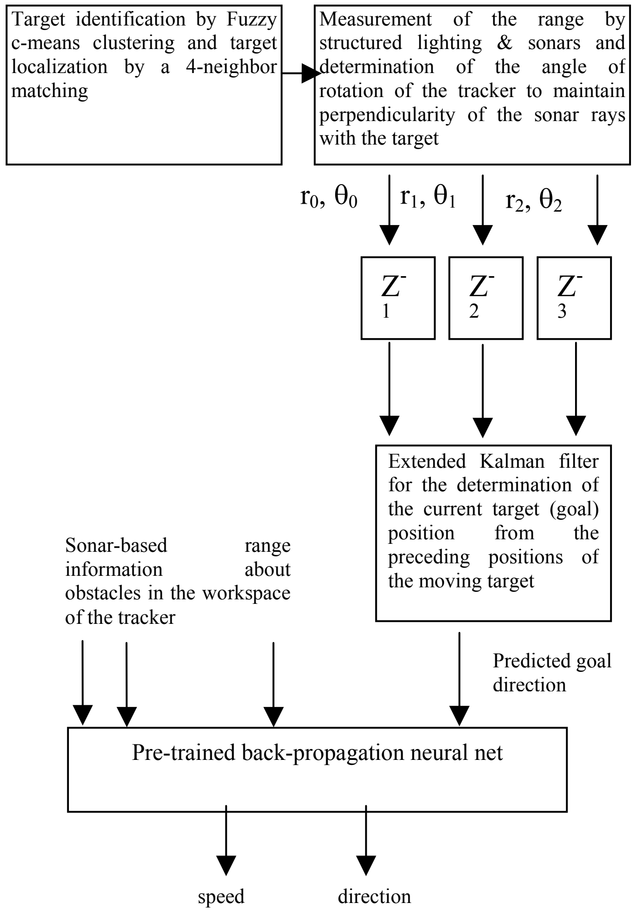


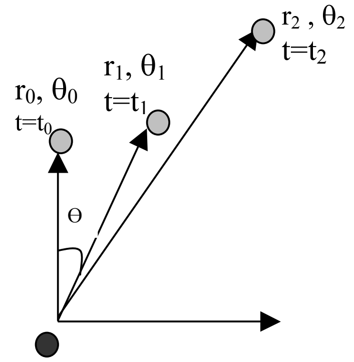
Fig. 5: A schematic diagram of the proposed target – tracking scheme.

successive frames of the target. Let the polar co-ordinates of these 3 positions be (r_0,θ_0), (r_1, θ_1) and (r_2, θ_2) respectively. All these measurements are performed by considering the S_{12} -axis of the Super Scout II robot as the reference x-axis and S_0 as the reference y-axis.

Fig. 6 describes 3 successive positions of the target in the measurement update phase of the tracker. The time t=0, $t=t_1$, $t=t_2$ correspond to the image sampling times associated with these measurements. After the measurement update phase is over, the tracker switches to prediction phase. In the prediction phase, the tracker determines the current position of the target from its preceding positions (r_i, θ_i) for i= 0 to 2. Once the prediction phase is over, the tracker starts its next measurement update phase, and the process continues until the tracker intercepts the target. A schematic view of a tracking cycle comprising of the measurement update phase and prediction phase is presented in Fig. 7.

Fig. 6 describes 3 successive positions of the target in the measurement update phase of the tracker. The time t=0, $t=t_1$, $t=t_2$ correspond to the image sampling times associated with these measurements. After the measurement update phase is over, the tracker switches to prediction phase. In the prediction phase, the tracker determines the current position of the target

from its preceding positions (r_i, θ_i) for i= 0 to 2. Once the prediction phase is over, the tracker starts its next measurement update phase, and the process continues until the tracker intercepts the target. A schematic view of a tracking cycle comprising of the measurement update phase and prediction phase is presented in Fig. 7.



● The tracker robot in its observation phase locates the target

○ The target robot in three different positions.

Fig. 6: The tracker observing the motion of the target.

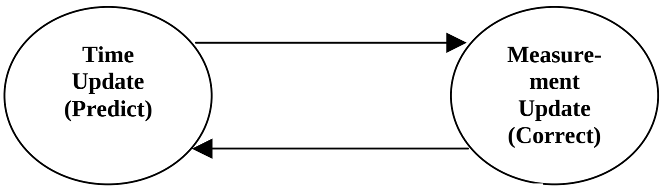


Fig.7: The recursive Kalman filter cycle.

IV.II Extended Kalman Filter- an Overview

Kalman Filtering [3], which basically stems from statistical estimation theory, has tremendous applications in robotics. A Kalman filter is a digital filter that provides a recursive solution to an estimation problem. It is impossible to explore such a vast and important mathematical tool in a few paragraphs. In the present section, the principle of extended Kalman filtering is just outlined to demonstrate its application in target tracking.

An extended Kalman filter [15] is a digital filter that attempts to minimize the measurement noise for estimating a set of unknown parameters, linearly related with a set of measurement variables. The most important significance of this filter is that it allows recursive formulation and thus improves accuracy of estimation up to the users' desired level at the cost of new measurement inputs.

Let

f_i ($\mathbf{x}_i$, $\mathbf{a}$) = 0 be a set of measurement equations describing relationships among an estimator vector $\mathbf{a}$ and measurement variable vector $\mathbf{x}_i$,

$\mathbf{x}_i{}^* = \mathbf{x}_i + \mathbf{l}_i$, where $\mathbf{l}_i$ is a white Gaussian type measurement noise such that

$E[\mathbf{l}_i] = 0$, $E[\mathbf{l}_i \mathbf{l}_i{}^{\mathbf{T}}]$ = positive symmetric matrix Λ_i, and $E[\mathbf{l}_i \mathbf{l}_j{}^{\mathbf{T}}] = 0$,

$\mathbf{a}_{i-1}{}^* = \mathbf{a} + \mathbf{s}_{i-1}$, where $\mathbf{s}_{i-1}$ is a white Gaussian type estimation noise such that $E[\mathbf{s}_{i-1}] = 0$,

$E[\mathbf{s}_{i-1} \mathbf{s}_{j-1}{}^{T}]$ = positive symmetric matrix $\mathbf{S}_{i-1}$ and $E[\mathbf{S}_{i-1} \mathbf{S}_{i-1}{}^{T}] = 0$.

Expanding f_i ($\mathbf{x}_i$, $\mathbf{a}$) by Taylor's series around ($\mathbf{x}_i{}^*$, $\mathbf{a}_{i-1}{}^*$), we find

$$f_i(\mathbf{x}_i, \mathbf{a}) = f_i(\mathbf{x}_i{}^*, \mathbf{a}_{i-1}{}^*) + (\partial f_i / \partial \mathbf{x})(\mathbf{x}_i - \mathbf{x}_i{}^*) + (\partial f_i / \partial \mathbf{a})(\mathbf{a} - \mathbf{a}_{i-1}{}^*) = 0. \quad (1)$$

After some elementary algebra, we find

$$\mathbf{y}_i = \mathbf{M}_i\ \mathbf{a} + \mathbf{w}_i \quad (2)$$

where $\mathbf{y}_i = -\ f_i(\mathbf{x}_i{}^*, \mathbf{a}_{i-1}) + (\partial f_i / \partial \mathbf{a})(-\mathbf{a}_{i-1}{}^*)$ (3)

is a new measurement vector of dimension (p_i x 1).

$$\mathbf{M}_i = (\partial f_i / \partial \mathbf{a}) \quad (4)$$

and $\mathbf{w}_i = (\partial f_i / \partial \mathbf{x})(\mathbf{x}_i - \mathbf{x}_i{}^*)$ (5)

is a measurement noise vector of dimension ($p_i \times 1$). We also want that $E[\mathbf{w}_i] = 0$ and define

$$\mathbf{W}_i = E[\mathbf{w}_i \mathbf{w}_i{}^{T}] = (\partial f_i / \partial \mathbf{x})\, \Lambda_i\, (\partial f_i / \partial \mathbf{x})^{T}. \quad (6)$$

Let $\mathbf{S}_i = E[(\mathbf{a}_i - \mathbf{a}_I{}^*)(\mathbf{a}_i - \mathbf{a}_i{}^*)^{T}]$ (7)

An attempt to minimize $\mathbf{S_i}$ yields the filter equations [10], given by:

$$\mathbf{a}_i{}^* = \mathbf{a}_{i-1}{}^* + \mathbf{K}_i(\mathbf{y}_i - \mathbf{M}_i\, \mathbf{a}_{i-1}{}^*) \quad (8)$$

$$\mathbf{K}_i = \mathbf{S}_{i-1}\, \mathbf{M}_i{}^{T}(\mathbf{W}_i + \mathbf{M}_i\, \mathbf{S}_{i-1} \mathbf{M}_i{}^{T})^{-1} \quad \text{and} \quad (9)$$

$$\mathbf{S}_i = (\mathbf{I} - \mathbf{K}_i\, \mathbf{M}_i)\, \mathbf{S}_{i-1}. \quad (10)$$

Given $\mathbf{S}_0$ and $\mathbf{a}_0$, the Kalman filter recursively updates $\mathbf{a}_i$, $\mathbf{K}_i$, $\mathbf{S}_i$ until the error covariance matrix $\mathbf{S}_i$ becomes insignificantly small, or all the number of data points have been submitted. The $\mathbf{a_i}$ thus obtained after termination of the algorithm is the estimated value of the parameters.

IV.III Predicting Target Position Using Extended Kalman Filter

Let the input measurement vector $\mathbf{x}$ be given by

$$\mathbf{x} = \begin{bmatrix} r \\ \theta \\ t \end{bmatrix} \quad (11)$$

where

r is the perpendicular distance of the target from the tracker at time t,

θ is the angular displacement of the target measured with respect to the axis S_{12} of the tracker, i.e. θ is the angular shift of tracker with respect to the target from time $t=0$ to the current time of observation,
t is the time elapsed measured from the beginning of the observation phase

Let the measurement equations in the present context be

$$\mathbf{f_i} = \begin{pmatrix} at^2+bt+c-r\cos\theta \\ pt^2+qt+s-r\sin\theta \end{pmatrix} = 0 \qquad (12)$$

where
c and s are the initial displacements of the target with respect to S_{12} axis and its perpendicular direction,
b and q are the velocity in the corresponding directions, and
a and p are the time rate of change of velocity in the corresponding directions.

For determination of the current position of the target we need to evaluate a, b, c, p, q, s from the measured (r, θ)s at time $t=0$, $t=t_1$ and $t=t_2$ respectively. The estimated values of a, b, c, p, q, s then may be substituted in the measurement equations to evaluate $r\cos\theta$ and $r\sin\theta$ at time $t \geq t_2$.
The estimator in the present context thus is given by

$$\mathbf{a} = \begin{pmatrix} a \\ b \\ c \\ p \\ q \\ s \end{pmatrix} \qquad (13)$$

For the evaluation of the estimator we, however, need to determine the following derivatives:

$$\partial \mathbf{f_i} / \partial \mathbf{x} = \begin{pmatrix} -\cos\theta & r\sin\theta & 2at+b \\ -\sin\theta & -r\cos\theta & 2pt+q \end{pmatrix} \qquad (14)$$

$$\partial \mathbf{f_i} / \partial \mathbf{a} = \begin{pmatrix} t^2 & t & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & t^2 & t & 1 \end{pmatrix}. \qquad (15)$$

Let

$$\Lambda_i = \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & \gamma \end{pmatrix} \qquad (16)$$

where α = the variance of the noise in measurement of range r,
β = the variance of the noise in measurement of angle θ,
and γ = the variance of the noise in measurement of time t.

The algorithm for the evaluation of the current position (r,θ) of the tracker at time $t = t_3$ from its preceding positions is presented below.

Procedure Evaluate (r_0, θ_0, r_1, θ_1, r_2, θ_2)
Begin
Initialize
a) $\mathbf{W}_0 := (\partial f_0 / \partial \mathbf{x})\, \Lambda_i\, (\partial f_0 / \partial \mathbf{x})^T$ where Λ_0 and $(\partial f_0 / \partial \mathbf{x})$ are available in expressions (16) and (14) respectively;
b) $\mathbf{S}_0$ as a diagonal matrix with large positive diagonal values;
c) $\mathbf{a}_0$ to be zero;
d) $\mathbf{M}_0 := \partial f_0 / \partial \mathbf{a}$ vide expression (15);
e) loop iteration index i:= 1;
Repeat
a) input new measurement $\mathbf{x}_i$ and evaluate $\mathbf{y}_i$ by expression (2);
b) update $\mathbf{K}_i$, $\mathbf{a}_i$, $\mathbf{S}i$ in order using expressions (9), (8) and (10);
Until abs($\mathbf{a}_i$ - $\mathbf{a}_{i-1}$) <a pre-defined threshold;
Determine
a) at^2+bt+c at time $t= t_3 > t_2$;
b) pt^2+qt+r at time $t=t_3> t_2$
for known a, b, c, p, q and r;
End.

IV. IV Use of the Back-Propagation Neural Net

After the current target position and the obstacle locations in the tracker's workspace are determined, we use a pre-trained back-propagation neural net for controlling the step-wise motion of the tracker. A three layered feed-forward neural net has been employed in the present context for generating the control commands for motion of the tracker. The inputs of the neural net are the sonar readings obtained by the tracker robot at time $t=t_3$ and the predicted target position at time $t= t_3$. In our realization, we considered the readings from 7 sonar transducers; consequently, the neural net has 8 inputs, the last one being the predicted position of the target.
The output of the neural net could be the amplitude and direction of motion of the tracker robot. But since speed of response is a prime consideration in the tracker design, we need to modulate the speed of the target depending on the estimated range of the tracker and obstacle locations. Thus speed and heading direction have been considered as the output fields of the proposed neural net. It may be added here that unlike in section 2 of the previous paper, where the heading direction was represented by 2 fields (amplitude and clockwise/ counter-clockwise orientation), the heading direction in the present context has been represented by a single field to limit the output fields to 2 only.
The proposed neural net is trained using 600 training instances with a root mean square sum error at the output layered nodes below 0.003 units. Approximately 10,000 learning epochs are

needed to train the neural net with the said error margin. In the application phase, the neural net just requires one forward pass, thus the speed of response of the neural net is very fast of the order of 0.5 milliseconds on a IBM 300 M-Hz Pentium machine.

V. Experimental Results

The experiments were carried out on Nomad simulator and also on 2 practical super scout II mobile robots. Figures 8 and 9 describe 2 simulated runs of the tracking program. Fig. 8 has been run by directly feeding the measured range to the input of the neural net. In Fig. 9 extended Kalman filter has been employed to predict the current position of the moving target. It is clear from these 2 figures that the tracker can generate more accurate control commands in presence of the extended Kalman filter.

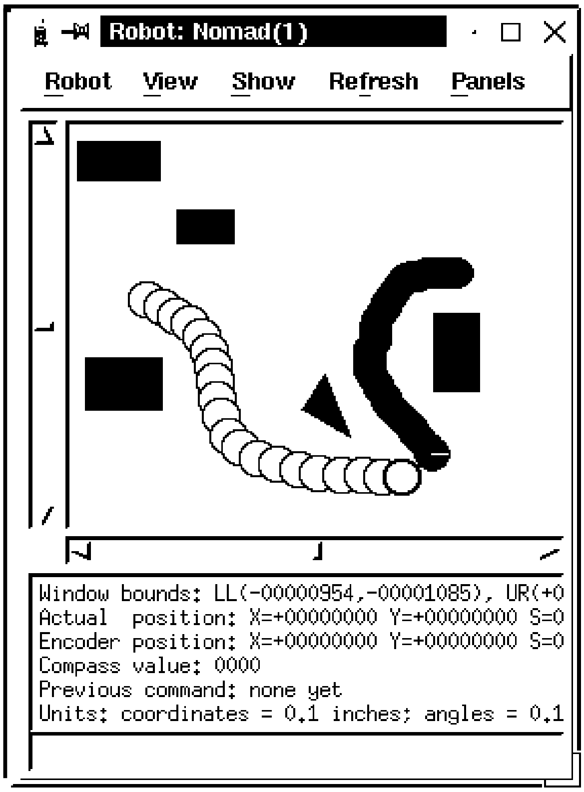


Fig 8: Target tracking and interception without Kalman filter. The rectangular white window represents the tracker's world map. The dark objects denote the obstacles in the map. The path with circular traces denotes the trajectory of the target, and the solid path describes the trajectory of the tracker. The scheme was tested on a Nomadic platform.

VI Conclusions

The paper examined the scope of fuzzy c-means clustering algorithm in segmentation of a moving object from the real-time video stream and its localization by the proposed 4-neighbor match algorithm. The experimental results on FCM clustering reveal that the logarithm of the gray scale pixel intensity is a good feature for clustering the dark pixels from relatively less dark ones. The localization algorithm is very efficient, as it needs minimum search to localize the object in the scene.

The paper also attempted to develop a scheme for target tracking and interception of mobile robots. An extended Kalman filter has been employed here to predict the next position of the target robot from its current and few preceding positions. A pre-trained back-propagation neural net is used to generate the motion information of the tracker from the predicted target position and the obstacle map around the tracker. The speed of response of the tracker, being very fast of the order of 400 milliseconds, significantly reduces the probability of missing the target.

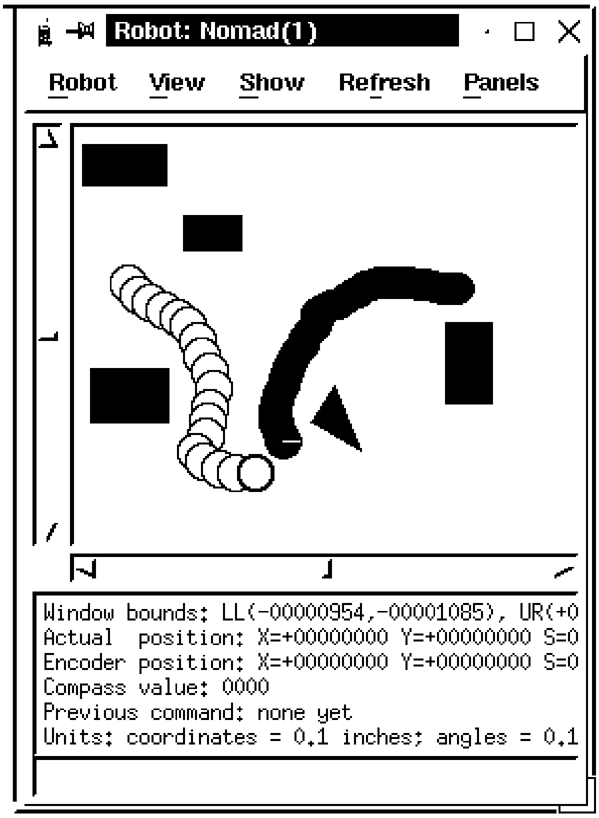


Fig. 9: Target tracking and interception with Kalman filtering. The tracker here intercepts the target much earlier than what it had done in fig. 8.

VII References

[1] Abutaleb, A.S., "Target Image Tracking using a Nonlinear Filter Based on Portraying Min. Principle", in *Data Fusion in Robotics and Machine Intelligence*, Eds. Abidi, Mongi, A. and Gonzalez, R.C.. Academic Press Inc., San Diego, pg 307, Ref. 1, 1992. .

[2] Biswas, B., Konar, A. and Mukherjee, A. K., "Image matching with fuzzy logic," *Proc. of Int. Conf. on Control, Automation, Robotics and Computer Vision (ICARCV'94*), Singapore, 1994.

[3] Brown, R. G. and Hwang, Patrick Y. C., *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley & Sons, New York, 1997.

[4] Brown, C., Durrant-Whyte, H., Leonard, J., Rao, B. and Steer, B., "Distributed Data Fusion Using Kalman Filtering: A robotics Application", in *Data Fusion in Robotics and Machine Intelligence*, Eds. Abidi, Mongi, A. and Gonzalez, R.C.. Academic Press Inc., San Diego, 1992. .

[5] Fok, Koon-Yu and Kabuka, M. R., "An automatic navigation system for vision guided vehicles using a double heuristic and a finite state machine," *IEEE Trans. on Robotics and Automation,* vol.7, no.1, Feb. 1991.

[6] Hitchings, M.R., Vlacic, L.B. and O'Sullivan, Z., "Fuzzy Control of Distance and Motion Tracking Applied to Co-operative Autonomous Robots", in *Distributed Autonomous Robotic Systems*, Eds. Parker, L.E., Bekey, G. and Barhen, J., Springer-Verlag, Tokyo, 2000.

[7] Jain, A.K., *Fundamentals of Digital Image Processing*, Prentice Hall, N.J., Oct.2000

[8] Kenefic, R.J., "Optimum Tracking of a Maneuvering Target in Clutter", in *Data Fusion in Robotics and Machine Intelligence*, Eds. Abidi, Mongi, A. and Gonzalez, R.C.. Academic Press Inc., San Diego, pp. 308, Ref. 23, 1992.

[9] Konar, A. and Chakraborty, U.K., "Neuro-GA Synergism in Motion Planning of a Mobile Robot," Information Science, Elsevier, 2006(to appear).

[10] Krishnapuram, R., Segmentation, in *Handbook of Fuzzy Computation*, Ed. Ruspini, E. H., Bonissone, P. P. and Pedrycz, W., IOP Press, pp. F. 7. 4:1-F. 7.4:5, 1998.

[11] Kuezewski, R.M., "Neural Network Approaches to Multi-Target Tracking", in *Data Fusion in Robotics and Machine Intelligence*, Eds. Abidi, Mongi, A. and Gonzalez, R.C.. Academic Press Inc., San Diego, pp. 309, Ref. 24, 1992.

[12] Luo F. and Unbehauen, R., *Applied Neural Networks For Signal Processing,* Cambridge University Press, pp. 22-26. 1997.

[13] Morefield, C.L., "Application of 0-1 Integer Programming to Multi-Target Tracking Problems", in *Data Fusion in Robotics and Machine Intelligence*, Eds. Abidi, Mongi, A. and Gonzalez, R.C.. Academic Press Inc., San Diego, pp. 309, Ref. 30, 1992

[14] Mori, S., Chong, C., Tse, E. and Wishner, R., "Tracking and Classifying Multiple Targets Without a Prior Identification", in *Data Fusion in Robotics and Machine Intelligence*, Eds. Abidi, Mongi, A. and Gonzalez, R.C.. Academic Press Inc., San Diego, pp. 309, Ref. 31, 1992.

[15] Nicholas, A., *Artificial Vision for Mobile Robots*, The MIT Press, Cambridge, MA, 1991.

[16] Ollero, A., García-Cerezo A., Martínez, J.L. and Mandow A., "Fuzzy Tracking methods for Mobile Robots", in *Applications of Fuzzy Logic, Towards High Machine Intelligence Quotient Systems*, Eds. Jamshidi M., Titli A., Zadeh L. and Boverie S., Prentice Hall, N.J., 1997.

[17] Pratt, W.K.,*Digital Image Processing*, John Wiley & Sons Inc., NY, 2004.

[18] Shirai, Y., Odaka, R. and Yamane. T., "Robust Vision for Vision-based control of Motion", in *Robust Visual Tracking by Integrating Various Cues*, Eds. Vincez, M. and Hager, G.D., IEEE Press, Piscataway, N.J., 2000.

[19] Stone, L.D., Barlow, C.C. and Corwin, T.L., *Bayesian Multiple Target Tracking*, Artech House, Boston, Massachusets, 1999.

[20] Warren, R.C., "A Bayesian Track-before-Detect Algorithm for IR Point Target Detection," in *Advances in Soft Computing*, Eds. Abraham, A. and Köppen, M., Pysica-Verlag, Warsaw, 2002.