Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2007)

Autonomous Visual Self-localization in Completely Unknown Environment using Evolving Fuzzy Rule-based Classifier

Xiaowei Zhou
Dept of Communication Systems
InfoLab21
Lancaster University
Lancaster LA1 4WA, UK

Plamen Angelov
Dept of Communication Systems
InfoLab21
Lancaster University
Lancaster LA1 4WA, UK

***Abstract*—A novel approach to visual self-localization in completely unknown environment with a fully unsupervised and computationally efficient algorithm is proposed in this paper. It is based on the recently developed evolving fuzzy classifier (eClass). The problem of self localization and landmark recognition is of extreme importance for designing efficient and flexible land-based autonomous uninhabited vehicles (AUV). The availability of global coordinates, a GPS link, and unrestricted communication is often compromised by a number of factors, such as interference, weather, and mission objectives. The ability to self-localize and recognize landmarks is vital in such cases for an AUV to survive and function effectively. The self-organizing classifier (*eClass*) is designed by automatic labeling and grouping the landmarks that are detected in real-time based on the image data (video stream grabbed by the camera mounted on the mobile robot, AUV). The proposed approach makes possible autonomous joint landmark detection and recognition without the use of absolute coordinates, any communication link or any pre-training. The proposed algorithm is recursive, non-iterative, one pass and thus computationally inexpensive and suitable for real-time applications. A set of new formulae for on-line data normalization of the data are introduced in the paper. Real-life tests has been carried out in outdoor environment at the Lancaster University campus using Pioneer3 DX mobile robots equipped with a pan-tilt zoom camera and an on-board PC. The results illustrate the viability and flexibility of the proposed approach. Further investigations will be directed towards teams of mobile robots (AUV) performing a task in completely unknown environment.**

I. INTRODUCTION

TERRAIN navigation requires the use of a map and absolute coordinates, or landmarks. In case when absolute coordinates are unavailable or unreliable the obvious choice is to use the surrounding environment to extract knowledge and self-localize by selecting trees, rocks, terrain contours or other objects as landmarks [1]. Navigation in an outdoor environment is substantially more complex than navigation in an indoor environment due to the lack of well defined features such as corners, walls, corridors etc. Even more, one can not afford to build a map at such a detailed level as required by a successful navigation in an outdoor terrain describing *a priori* the location and orientation of natural features, such as trees, rocks, cliffs etc. [1]. One obvious obstacle is the huge dimensionality of such a hypothetical task. Another, less obvious, reason is the fact that natural landmarks (such as trees, rocks etc.) are subject to weather exposure, seasonal changes etc. [1]. Therefore, successful terrain navigation, and thus operation, in an outdoor environment strongly requires an *on-line* ***autonomous*** self-localization and map-building [1]. In terms of learning method, the *autonomy* requires the use of *unsupervised* learning approaches. This capability should also have an ***evolving*** aspect in the sense that the number of landmarks can not be pre-specified and fixed *a priori*. This requires capability to design *on-line* ***evolving*** maps (sets of landmarks) in an understandable for the AUV form. Additionally, this should be computationally inexpensive and fast in order to be suitable for ***real-time*** implementations.

A newly introduced approach for joint landmark recognition and classifier design [2] is taken further in this paper and a set of realistic outdoor experiments were performed to validate it. It is based on the recently introduced evolving fuzzy rule-based classifier that will be presented in a parallel paper [3].

Unsupervised landmark recognition (and thus self-localization) schemes are possible using, for example, self-organizing maps (SOM) [4]. They are computationally less expensive and has been developed further into eSOM (evolving SOM) for the case when the cluster centers 'evolve' [5]. However, they, as well as a number of other evolving and self-organizing neural networks such as growing cell structures [6], adaptive resonance theory mapping [7], dynamic evolving neuro-fuzzy inference systems [8], resource allocation networks [9] and their applications [12-17], do not take into account data density and are prone to generating too many classes (pseudo landmarks) and thus they require pruning. All these approaches are not prototype-based in the sense that the centre of the clusters is not necessarily and is often not located at a feasible point in the data space. It is usually located at the mean or its location is a result of an adaptation. Thus, its location in the data space is an abstraction and may not be a feasible data point. Additional disadvantage of these approaches is that new data point is compared to the cluster centers only, not to all previous data because the real-time nature precludes memorizing the data history. In this way

1-4244-0700-1/07/$20.00 ©2007 IEEE

important information is usually lost.

Recently, a novel approach to real-time fuzzy rule-based clustering [18] was proposed that can be seen as an on-line and evolving extension of the well-known Mountain and subtractive clustering approaches [19] and the recently introduced concept of evolving fuzzy rule-based systems [20,21]. It has also been used for objects [22] and mobile robot tracking [23]. It is fully unsupervised in the sense that not only the cluster labels/outputs but also the number of clusters is not pre-defined but is determined based on the data density. This approach has been extended to classification in a parallel paper [3] and called *eClass* when the class labels are assigned automatically and again the class number is not pre-defined. In the present paper, the prototypes are extracted from images and are selected as landmarks using *eClass*. The data (color intensity of pre-defined bins of the image) are normalized using a newly introduced procedure for on-line normalization suitable for the *eClass* algorithm. Landmarks are automatically labeled and further data are classified into classes associated with the most similar landmark detected so far in real-time. The number of classes (respectively landmarks) is not pre-specified. Instead it evolves starting '*from scratch*' with the very first landmark detected while exploring previously unseen environment. By associating location data a simple map of the environment based on the automatically detected landmarks can be build.

The evolving fuzzy rule-based classifier, *eClass* is formed by real-time detection of landmarks and labeling them. It is then used to classify in real-time the data produced by the camera mounted on the AUV (a mobile robotic platform Pioneer3-DX). In the experiment carried out in an outdoor environment at the campus of Lancaster University, Lancaster, UK an AUV (mobile robot) collects images while traveling in the campus. The algorithm identifies automatically landmarks and if the robot passes again near the same landmark it already knows that it has been there without any pre-training. Note that by using fuzzy classifier the second time the AUV gets a similar image it does not need to be exactly the same. Instead a degree of similarity is required (measured by Euclidean or cosine distance). This contributes to the flexibility and robustness in recognition of previously seen scenes.

Future investigations will be directed towards development of team-oriented methods for self-localization, novelty detection, and landmark and target recognition and improved feature selection.

II. Vision-based Landmark Recognition

A. *Visual-based novelty detection and landmark recognition*

As the robot travels in previously unseen environment it generates a video stream using its on-board camera. In a scenario when the AUV can not rely on absolute localization such as GPS, on pre-specified maps and when the communication is band-limited an ability to extract knowledge and use this to improve navigation can be vital for survivability and efficiency of the AUV [14-17,25,26]. The ability to differentiate between common background and patterns never met before which is called 'novelty detection' [17] is a very useful competence for a mobile robot (AUV) operating in a real dynamic unexplored environment. Using such ability the robot can select which aspects of the environment are unusual, differ from the contextual background and use them as 'landmarks'. By differ from 'dead reckoning', which is prone to drifting errors [14-17], landmark-based navigation does not suffer from this disadvantage [1]. Thus it can effectively be used for adaptive navigation and route planning.

At the same time the limited computational resources available to an autonomous mobile robot often present challenge for applications that demand real-time processing of large amounts of sensory data, therefore, a recursive algorithm is highly desirable to cope with the memory and time limitations. This is especially important for designing agile compact autonomous devices [22] where the computational and energy requirements are usually very restrictive. Another important requirement to such an algorithm is to be open or flexible that means the number of landmarks not to be pre-specified in advance. This leads to the necessity of evolving structures to realize such an algorithm.

B. *Feature selection and image processing*

The inputs to the *eClass* are formed by the pre-processed image in real-time. Preprocessing is simple and includes grabbing the bitmap image and segmenting it into bins (column-wise and row-wise) similarly to the approach adopted in [27]. The visual information that is contained in a frame of size *M x N* pixels can be decomposed into a grid of smaller (*m x n*) local images. In our experiments we have segmented images of size *640 x 480 pixels* into twelve *160 x 160 pixel* areas (bins) as illustrated in Figure 1.

Fig. 1 An example of image segmentation into twelve bins

In each bin, the mean value of the color intensity of all pixels that form that area/bin is calculated for each of the 3 color channels, namely R (red), G (green), and B (blue):

$$\mu_{pq}^{R} = \frac{1}{(160)^2}\sum_{i=1}^{160}\sum_{j=1}^{160} I_{ij}^{R}$$

where I_{ij}^{R} denotes the intensity of the Red of the i^{th} column; j^{th} row of the bin;

μ_{pq}^{R} denotes the mean value of the Red in the bin formed by the p^{th} vertical and q^{th} horizontal of the image (*p=[1,3]; q=[1,4]*).

Having 12 bins in the image and for each channel (Red, Green or Blue) we form 36 features for each frame. These 36 features form a frame that represents the area/bin of the image that is further processed by the evolving fuzzy classifier.

Note that better results can be expected if use HUV (hue, saturation, and value of brightness) as described in [28].

III. Evolving Fuzzy Rule-based Classifier (eClass)

A. Structure of the evolving classifier eClass

eClass can be described as a set of fuzzy rules of the following form:

$$\begin{aligned} &R_l^i : IF(x_1 \text{ is close to } x_1^{i*}) \text{ AND...} \\ &\text{AND } (x_n \text{ is close to } x_n^{i*}) \text{ THEN}(x \text{ is } LM_l) \end{aligned} \quad (1)$$

where R_l^i denotes the i^{th} fuzzy rule; *i=[1,N]; N* is the number of fuzzy rules; the consequent LM_l is the l^{th} class/landmark label; *l=[1,L]; L* is the number of classes (note that $L \le N$ which means there is at least one fuzzy rule per landmark); $x=[x_1,x_2,...,x_n]^T$ is the features vector that is $x=[\mu_{11}^R,...,\mu_{34}^R;\mu_{11}^G,...,\mu_{34}^G;\mu_{11}^B,...,\mu_{34}^B]^T$; $\left(x_j \text{ is close to } x_j^{i*}\right)$ denotes the j^{th} fuzzy set of the i^{th} prototype (fuzzy rule), *j=[1,n];* x^{i*} is the prototype of the i^{th} rule antecedent.

The membership function that describes the closeness to the prototype is assumed to be of Gaussian type. The overall firing level, τ of the i^{th} fuzzy rule of the l^{th} class (landmark) is given as a product of the membership functions of the fuzzy sets for that rule [19] and results again in a Gaussian:

$$\tau_l^i = \prod_{j=1}^{n} e^{-\frac{1}{2}\left(\frac{d_{ij}}{\sigma_j^i}\right)^2} = e^{-\frac{1}{2}\sum_{j=1}^{n}\left(\frac{d_{ij}}{\sigma_j^i}\right)^2} \quad i=[1,N] \quad (2)$$

where d_{ij} is the dissimilarity between a frame and the prototype of the i^{th} landmark (the focal point of the fuzzy rule); σ_j^i is the spread of the membership function.

Note that this representation resembles the normal distribution used widely in statistics and the spread of the membership function can also be represented by the standard deviation. The spread of the membership function σ can be determined recursively as [3]:

$$\left(\sigma_k^l\right)^2 = \left(\sigma_{k-1}^l\right)^2 + \frac{1}{S_k^l}\left(d(x^{i*},x_k)^2 - \left(\sigma_{k-1}^l\right)^2\right); \; \sigma_0^i = 1 \quad (3)$$

where *i=[1,n]* is the number of classes/landmarks; $d(x^{i*}, x_j)$ denotes the dissimilarity between the prototype and the new frame assigned into this class.

Since we assume an evolving classifier, new landmarks (classes/rules) will be formed on-line. When a new rule is formed, *N←N+1*, its spread is initialized by [3]:

$$\sigma_k^{N+1} = \frac{1}{N}\sum_{i=1}^{N}\sigma_k^i \quad (4)$$

The overall classification is produced by an inference from the fuzzy rule base using so called 'winner-takes-all' defuzzification approach [3,19] (the label of the winning rule forms the output of the fuzzy rule base):

$$l^* = \arg\max_{i=1;l=1}^{N;L}\left\{\tau_i^l\right\} \quad (5)$$

The *eClass* method (as the name suggests) assumes an unspecified number of classes that gradually evolve. Their labels are automatically assigned. The approach also assumes joint classification and classifier generation similarly to the joint adaptation and prediction used in conventional (linear) adaptive systems [29]. *eClass* starts with an empty rule-base until a landmark is detected. The images are processed in real-time and a vector of features is formed based on each frame. Each feature vector can be represented as a data point in the data space, $x_k = \left[x_k^1, x_k^2, ..., x_k^n\right]^T$ where k is the current time instant (in a real-time application the time is open-ended and stops when a stop condition that is external to this algorithm is reached, thus *k=1,2,…*).

The structure of the proposed classifier is thus formed by sets of fuzzy rules of type (1) in such a way that there is at least one fuzzy rule per class associated to a landmark. The prototypes around which the fuzzy rules are formed are frames selected by unsupervised learning (*eClustering*).

eClass, similarly to *eClustering*, is based on the concept of data spatial density measured by so called 'potential' [18-21]:

$$P_k(x_k) = \frac{1}{1+\left(\sum_{i=1}^{k-1} d^2(x_k,x_i)\right)/(k-1)}; k=1,2,... \quad (6)$$

As a measure of dissimilarity, d one can use Euclidean [18-21]:

$$d_E = \left\|x_k - x_i\right\| = \sqrt{\sum_{j=1}^{n}\left(x_k^j - x_i^j\right)^2} \quad (7a)$$

or *cosine* distance [3]:

$$d_{\cos} = 1 - \frac{\sum_{j=1}^{n} x_k^j x_i^j}{\sqrt{\sum_{j=1}^{n}\left(x_k^j\right)^2 \sum_{j=1}^{n}\left(x_i^j\right)^2}} \quad (7b)$$

Note that the expression for the potential (6) is suitable for off-line (batch) calculation only since the summation over **all** previous data points is needed to determine the data density. To use potential in a real-time algorithm where memorizing the previous history is prohibitive, a *recursive* version is needed. For Euclidean distance it was given as [18]:

$$P_k(z_k)=\frac{k-1}{(k-1)(a_k+1)+b_k-2c_k} \tag{8a}$$

Where the following notations has been used:

$$a_k=\sum_{j=1}^{n}\left(x_k^j\right)^2;\quad b_k=\sum_{i=1}^{k-1}\sum_{j=1}^{n}\left(x_i^j\right)^2;\quad c_k=\sum_{j=1}^{n}x_k^j f_k^j;\quad f_k^j=\sum_{i=1}^{k-1}x_i^j \tag{8b}$$

Values a_k and c_k can be calculated based on the availability of the current frame, x_k only. The values b_k and f_k^j require accumulation of past information. This can be stored in two auxiliary variables only (the scalar, b_{k-1} and the n-dimensional vector-column $f_k=\left(f_k^1,f_k^2,...,f_k^n\right)^T$). One can recursively update these n values by:

$$b_k=b_{k-1}+a_{k-1};\ b_1=0 \tag{9a}$$

$$f_k^j=f_{k-1}^j+x_{k-1}^j;\ f_1^j=0 \tag{9b}$$

Alternatively, if we use cosine distance/dissimilarity we arrive at the following formula for recursive calculation of the potential [3]:

$$P_k(x_k)=\frac{1}{2-\frac{1}{\sqrt{\sum_{i=1}^{n}\left(x_k^i\right)^2}}\sum_{i=1}^{n}x_k^i a_{k-1}^i};\quad P_1(x_1)=1 \tag{10a}$$

where $a_k^i=a_{k-1}^i+\sqrt{\frac{\left(x_k^i\right)^2}{\sum_{l=1}^{n}\left(x_k^l\right)^2}};\ a_1^i=\sqrt{\frac{\left(x_1^i\right)^2}{\sum_{l=1}^{n}\left(x_1^l\right)^2}};\ i=[1,n]$ (10b)

In this way, the spatial density at each new frame, P_k in respect to **all** previous frames can be *recursively* calculated using n accumulated values in the two auxiliary variables only. This makes possible real-time applications of the algorithm while keeping the information of spatial data density regarding the whole previous history which is the distinctive feature of the proposed algorithm.

Each time a new frame is grabbed and its features extracted it affects the data density of the global data space, therefore the potentials of all existing prototype frames needs to be updated. This update is also done in a recursive way for both Euclidean and cosine distance cases respectively, and no extra variable needs to be memorized, apart from the current potential of the existing prototypes (focal points):

$$P_k(x^*)=\frac{(k-1)P_{k-1}(x^*)}{k-2+P_{k-1}(x^*)+P_{k-1}(x^*)d(x^*,x_k)} \tag{11}$$

B. *Online normalization*

For off-line problems the normalization or standardization is a straightforward task [10]:

a) normalization:

$$x_{norm}=\frac{x-\min(X)}{\max(X)-\min(X)} \tag{12a}$$

b) standardization:

$$x_{std}=\frac{x-mean(X)}{std(X)} \tag{12b}$$

Where X denote the dataset to be processed, and x denotes each data sample in X.

For online, real-time implementations it requires special attention. In online mode, when the number of elements in X grows, the results of *min(X)* and *max(X)* changes, but this takes place only when the new element x exceed one of the existing *lower* or *upper* boundary of X. At the same time, in online mode *mean(X)* and *std(X)* change all the time when new element comes in, unless the new element equals exactly to existing *mean(X)*. Therefore, the online standardization requires updating all the parameters for every new data sample, while the normalization requires update only when new data exceeds one of the boundaries (note that a single data sample can not exceed both boundaries at the same time). This is an argument in favor of the normalization for on-line cases.

The proposed online normalization starts with the third data point, x_k. We first check if x is within current boundaries:

$$x_{\min}^c\le x_k\le x_{\max}^c \tag{13}$$

where $x_{\max}^c=\max_{i=1}^{k-1}\left(x_i\right)$ denotes the current upper boundary of x_k;

$x_{\min}^c=\min_{i=1}^{k-1}\left(x_i\right)$ denotes the lower boundary of x_k.

If (13) is satisfied then x_k is normalized by the current $x_{\min}^c$ and $x_{\max}^c$ by applying:

$$x_k^{norm}=\frac{x_k-x_{\min}^c}{x_{\max}^c-x_{\min}^c} \tag{14}$$

If (15) is not satisfied then we first update the boundaries, $x_{\min}^c$ and $x_{\max}^c$ used for normalization and afterwards we normalize x_k.

When an update of the boundaries is required, the new lower and upper boundaries are determined by:

$$x_{\min}^n=\min(x_{\min}^n,x_k) \tag{15a}$$

$$x_{\max}^n=\max(x_{\max}^n,x_k) \tag{15b}$$

Let us denote the current and the new range respectively by $r^c=x_{\max}^c-x_{\min}^c$ and $r^n=x_{\max}^n-x_{\min}^n$. Let us further introduce the ratio $\rho=\frac{r^c}{r^n}$ and the increment $\delta=\frac{x_{\min}^c-x_{\min}^n}{r^n}$. Note, that when $x_k > x^c_{\max}$ we have $x^c_{\min} = x^n_{\min}$ thus $\delta=0$.

Each time the normalization boundaries change we need to update all the parameters used by the algorithm as follows:

$$x^* = \rho x^{*c} + \delta \tag{16a}$$

$$x_{k-1} = \rho x^c_{k-1} + \delta \tag{16b}$$

$$f_{k-1} = \rho f^c_{k-1} + (k-1)\delta \tag{16c}$$

$$b^j_{k-1} = b^{jc}_{k-1} + 2\delta\rho f^{jc}_{k-1} + (k-1)\delta^2;\;\; j=[1,n] \tag{16d}$$

$$b_{k-1} = \sum_{j=1}^{n} b^j_{k-1} \tag{16d}$$

$$P_{k-1}(x^*) = \frac{1}{1+\rho^2\left(\frac{1}{P^c_{k-1}(x^*)}-1\right)^2} \tag{16e}$$

$$\sigma^i_j = \rho\sigma^{ic}_j \;\; j=[1,n] \tag{16f}$$

Note that the superscript 'c' denotes the 'current' value (before the update). The derivation of the transformations (16) is given in the Appendix.

Once all these parameters are updated the old boundaries are replaced with the new ones:

$$x^c_{\min} = x^n_{\min};\quad x^c_{\max} = x^n_{\max} \tag{17}$$

And only after that, x_k is updated by (14).

C. Learning eClass (generating fuzzy rules)

The proposed evolving fuzzy rule-based classifier starts '*from scratch*' (with an empty rule-base). Each new frame can be used to upgrade the rule base. From the second frame onwards its *potential*, $P_k(x_k)$ is updated recursively by (8)-(9) or by (10). Then the potential of each of the previously existing prototypes, $P_k(x^*)$ is also updated using (11). Comparing the potential of the new data sample with the potential of each of the existing prototypes the following outcomes are possible:

a) $\Delta P_i < 0;\; \forall i$; (18a)

b) otherwise (18b)

where $\Delta P_i = P_k(x_k) - P_k(x^{i*})$ is called '*potential difference*'.

If condition (18a) occurs that means that we have a distinctive frame that can be used as a prototype for a landmark. Thus we evolve the rule base by adding a new fuzzy rule formed around the features of this frame. Otherwise, (18b), we do not change the overall structure of the classifier and proceed further.

When adding a new fuzzy rule around the prototype, we check whether any of the already existing prototypes for the landmark C^l are described *well* by the newly added fuzzy rule. By *well* we mean [3] that the value of the membership function satisfies:

$$\exists i, i=[1,N];\quad \tau^j_i > 1/3 \quad \forall j, j=[1,n] \tag{19}$$

If (19) holds then the newly formed fuzzy rule is assigned to the sub rule-base for the latest landmark; otherwise, the new fuzzy rule describes a new landmark and, thus, forms a new sub rule-base (Figure 2). The procedure for automatic landmark detection and recognition using *eClass* can be represented by the following pseudo-code:

```
BEGIN eClass
  Initialize (grab first frame, F1 and
  extract its features);
    DO for any next frame, Fk, k=2,3,…
      Extract features xk;
      /*---Classify xk---*/
      Classify xk to one of existing
      Landmarks (LM^l)using the rule-base
      Check (13)and re-normalize if needed
      /*---Evolve Rule-base---*/
      Calculate Pk(xk)using (8)(9)or(10);
      Update P(x*) using (11)
      Calculate the potential difference
      IF (18a)AND(19) THEN add a new fuzzy
      rule around xk used as a prototype;
      IF (18a)BUT NOT(19)THEN form a new
      new landmark (LM^N+1) and a new
      sub-rule using xk as a prototype;
    WHILE video stream exists
  END DO
```

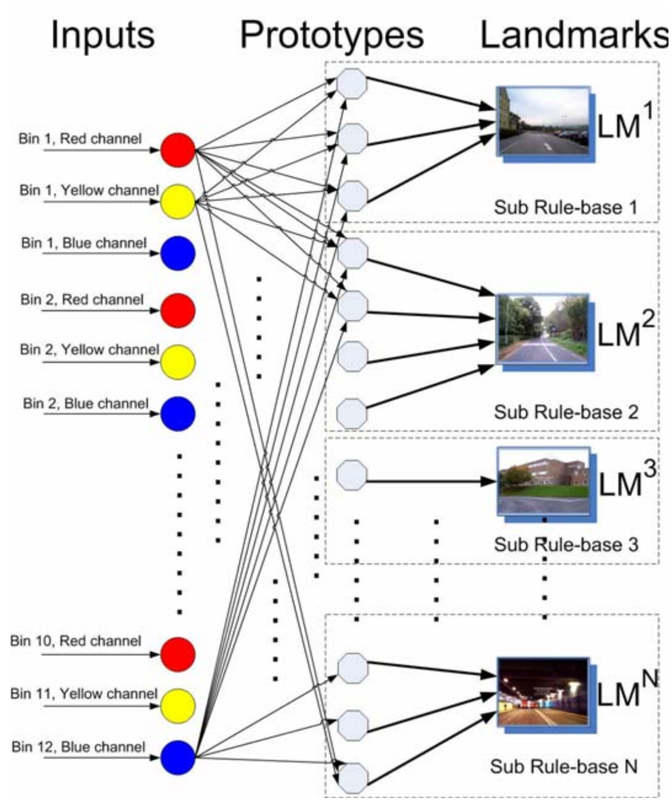

Fig.2 Network diagram of *eClass* for landmark recognition

Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2007)

IV. EXPERIMENT

A. Experiment outline

The experiment is carried out with a Pioneer3 DX mobile robot [11] equipped with an onboard PC and pan-tilt-zoom camera. There is no external links such as GPS and the wireless data connection is disabled. Thus, the task is performed fully unsupervised by the AUV (mobile robot).

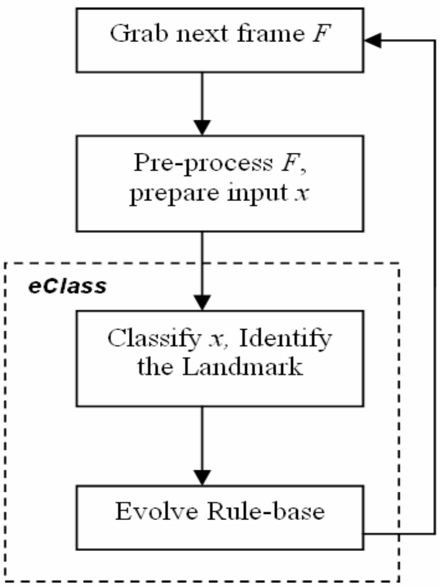

Fig.3 Flow chart of the of landmarks recognition

eClass runs on the onboard computer in real-time with no pre-defined rules for the landmark recognition task when exploring an unknown outdoor environment. The camera shots a video and grabs and sends images back to the onboard computer at the rate of 11.6 frames per second. The images are pre-processed as described in session II. As a result an input 36-element vector containing the colour information for 12 sub-areas in the frame is prepared. Each time a frame is received and pre-processed, the input vector is then fed to *eClass* for joint landmark detection and recognition.

When the mobile robot explores the unknown environment, meaningful novel images are detected and they are used as prototypes that form the rule base which represents the landmarks. If the input is similar to an existing prototype (and a fuzzy rule is fired, which means the received frame is similar to the previously seen scene, *eClass* classifies this frame to the nearest class/landmark applying so called by "winner-take-all" strategy.

The inputs are discarded after they are used by *eClass* to evolve the rule-base in each step.

B. Experimental settings

The experiment was conducted outdoor, on the campus of Lancaster University, UK. The AUV travels along a pre-defined route for two rounds. The route consists of 4 significant locations (denoted by A, B, C, D in Figure 4). They are identified subjectively and are only used to verify

Fig. 4 The route of the mobile robot during the experiment.

and analyze the result produced by the proposed scheme.

The whole experiment lasts about 6 minutes. Real-time video was produced by the camera mounted on the AUV with the frame rate at 25fps. During the experiment, 11.6 frames per second were grabbed and sent to the on board computer for landmark detection and recognition. The objective is to first correctly classify each frame to the nearest landmark (class) and relate to the location using compass or odometer data. The second objective is to detect the significant landmarks for the locations, generate one class for each location it detected.

TABLE I
RESULTS FOR LANDMARK RECOGNITION FOR A CAMPUS ROUTE

Location	Landmarks per location	Prototypes per Location	Lap when Generated	Frame when a new Landmark is Identified
A	4	62	1	1, 92, 155, 174
B	2	17	1	267, 375
C	1	11	1	1539
D	3	16	1	1594,1534,1736
Noise	1	1	2	2828

C. Results and analysis

In the experiment *107* prototype frames were selected and fuzzy rules of the type (1) were identified 'on the fly' by *eClass* running autonomously during the first lap along the *1*-mile route we chose for the experiment in the campus:

$$R_1^1 : IF(\mu_{11}{}^R \text{ is close to } 0.5407) AND (\mu_{11}{}^G \text{ is close to } 0.5548)$$

$$\cdots\ AND\ (\mu_{34}{}^B \text{ is close to } 0.3374)\ THEN\ (F_k \text{ is } LM_l)$$

$$R_2^1 : IF(\mu_{11}{}^R \text{ is close to } 0.30906) AND (\mu_{12}{}^G \text{ is close to } 0.17433)$$

$$\cdots AND\ (\mu_{34}{}^B \text{ is close to } 0.1053)\ THEN\ (F_k \text{ is } LM_l)$$

Note that the values of the mean RGB intensity per area/bin are normalized. The *107* fuzzy rules were grouped in 10 different classes representing *10* different landmarks It is interesting to note that the identified landmarks extracted autonomously represent very well the four locations defined subjectively (location *A, B, C, D*). The noise frames caused by novel objects in front of the camera are successfully identified by *eClass* and classified into a separate class (see the bottom row in Table I).

During the second lap, no more landmarks are identified, as the environment does not change comparing to the first round. In real-time, the frames received from the camera are classified into the classes representing the most matching (similar) landmarks. One new class was generated; due to several frames that are not related to a previous landmark (we treat them as a noise). Comparing to the classification which would be done by a human in a batch mode, the result is around 82% (3287 frames) out of 4050 frames correct classification.

a) Entrance b) Inside c) Exit

Fig. 5 Snapshot of Underpass: Entrance, Inside and Exit.

It is interesting to notice that the three landmarks generated at location *C*, are well describing the entrance, interior and the exit of the underpass on the campus (Figure 5).

This illustrates the ability of *eClass* to detect meaningful prototypes and the important, significant changes in the frames and thus to extract meaningful representation of the environment that if linked to the location information can be used for mapping and navigation, can be transmitted to another AUV (if performing a cooperative task) or to a human.

When a landmark was visited for second time the classifier was able to recognize this fact and information was displayed on the screen of the on-board computer of the mobile robot (AUV) which can optionally be send wirelessly to another robot or to a monitoring desktop workstation.

The proposed approach demonstrates its advantages: high recognition rate, high degree of autonomy, high flexibility (*eClass* structure is not fixed and can accommodate more classes if the environment changes) and high computational efficiency. Additional important advantage of the proposed approach is that the information extracted from the video stream in real-time and summarized in the rule base is fully linguistically transparent and interpretable.

V. CONCLUSION

A novel approach to self-localization in completely unknown environment with a fully unsupervised and computationally efficient algorithm is proposed in this paper. By using *eClass*, a transparent, compact and accurate fuzzy rule-based classifier can be evolved in real-time based on experimental data only. It is interesting to note that the rate of generating new classes and fuzzy rules representing a distinct landmark does not lead to an excessively large rule base. The reason for this is that the condition (12a) is practically very strong and gets stronger the more the data because it concerns **all** previously seen data. Additionally, the possible proximity of a candidate prototype to the already existing landmarks (condition (13)) leads to just a replacement of the existing landmark, and thus to the rule-base size.

The self-organizing classifier (eClass) is designed by automatic labeling and grouping the landmarks that are detected in real-time based on the image data from a camera mounted on the mobile robot (AUV). The proposed approach makes possible fully autonomous and unsupervised joint landmark detection and recognition without the use of absolute coordinates, any communication link or any pre-training. The proposed algorithm is recursive, non-iterative, one pass and thus computationally inexpensive and suitable for real-time applications.

Real-life tests has been carried out in outdoor environment at the Lancaster University campus using Pioneer3 DX mobile robot (AUV) equipped with a pan-tilt zoom camera and an on-board PC. The results illustrate the viability and flexibility of the proposed approach. Further investigations will be directed towards development of a cooperative scheme when a team of mobile robots (AUV) cooperate in the same environment.

REFERENCES

[1] H. Durrant-Whyte, "A Critical Review of the State-of-the-Art in Autonomous Land Vehicle Systems and Technology", Sandia Report, SAND2001-3685, Nov. 2001.

[2] Zhou X., P. Angelov, "Real-Time Joint Landmark Recognition and Classifier Generation by an Evolving Fuzzy System," *2006 IEEE World Congress on Computational Intelligence*, Vancouver, BC, Canada, July 16-21, 2006, pp.6314-6321, ISBN 0-7803-9489-5.

[3] P. Angelov, X. Zhou, F. Klawonn, Evolving Fuzzy Rule-based Classifiers, *2007 IEEE Intern. Conf. on Comp. Intell. Applic. for Signal and Image Proces.*, April 1-5,2007, Honolulu, Hawaii, USA.

[4] T. Kohonen, *Self-Organization and Associative memory.* New York: Springer Verlag, 1984.

[5] Kasabov N "Evolving fuzzy neural networks for on-line supervised/unsupervised, knowledge-based learning," *IEEE Trans. SMC - part B,* Cybernetics Vol.31, pp.902-918, 2001.

[6] Fritzke B. "Growing cell structures – a self-organizing network for unsupervised and supervised learning," *Neural Networks*, vol.7 (9) pp. 1441-1460,1994.

[7] Carpenter G. A., S. Grossberg, J. H. Reynolds, "ARTMAP: Supervised real-time learning and classification of non-stationary data by a self-organizing neural network", *Neural Networks*, vol.4, pp.565-588, 1991.

[8] Kasabov N., Q. Song "DENFIS: Dynamic Evolving Neural-Fuzzy Inference System and Its Application for Time-Series Prediction," *IEEE Trans. on Fuzzy Systems, Vol.10* (2), pp. 144-154, 2002.

[9] J. Plat, "A resource allocation network for function interpolation", *Neural Computation*, vol. 3 (2) 1991, 213-225.

[10] Duda, R.O., P.E. Hart, D.G. Stork (2000) *Pattern Classification – 2nd Edition*. Wiley-Interscience, Chichester, UK.

[11] *Pioneer 3^{tm} Operations Manual,* ActivMedia Robotics, LLC Amherst, NH, 2004, chap 5.

[12] L. Jun, T. Duckett, "Robot Behavior Learning with a Dynamically Adaptive RBF Network: Experiments in Offline and Online Learning," *Proc. 2nd Intern. Conf. on Comput. Intelligence, Robotics and Autonomous System, CIRAS,* Singapore, Dec. 15-18, 2003.

[13] L. Jun, T. Duckett, "Growing RBF networks for learning reactive behaviours in mobile robotics", *Intern. Journal of Autonomous Vehicle Systems*, *special issue on Comp. Intell. and its Applic. to Mobile Robots and Autonomous Systems,* 2005

[14] S. Marsland, U. Nehmzow, J. Shapiro, "On-line Novelty Detection for Autonomous Mobile Robots," *J. Robotics and Autonomous Systems,* Vol. 51, pp. 191-206, 2005.

[15] S. Marsland, S., U. Nehmzow, T. Duckett "Learning to select distinctive landmarks for mobile robot navigation," *Robotics and Autonomous Systems,* Vol. 37, pp.241-260, 2001

[16] U. Nehmzow, T. Smithers, J. Hallam, *Location Recognition in a Mobile Robot Using Self-Organising Feature Maps, In G. Schmidt (ed.), Information Processing in Autonomous Mobile Robots,* Springer, 1991.

[17] H. V. Neto, U. Nehmzow "Visual Novelty Detection for Inspection Tasks using Mobile Robots," *in Proc. of 8th Brazilian Symposium on Neural Networks , IEEE Press,* São Luís, Brazil, 2004.

[18] P. Angelov, "An Approach for Fuzzy Rule-base Adaptation using On-line Clustering," *Intern. Journal of Approx. Reasoning*,v.35, 275-289, 2004.

[19] Yager R.R., D.P. "Filev Learning of Fuzzy Rules by Mountain Clustering," *Proc. of SPIE Conf. on Application of Fuzzy Logic Technology,* Boston, MA, USA, pp.246-254,1993.

[20] P. Angelov, *Evolving Rule-based Models: A Tool for Design of Flexible Adaptive Systems*. Heidelberg, Germany: Springer-Verlag, 2002.

[21] P. Angelov, D. Filev, "An approach to on-line identification of evolving Takagi-Sugeno models", *IEEE Trans. on Systems, Man and Cybernetics, part B,* vol.34, No1, pp. 484-498, 2004.

[22] M. A. Memon, P. Angelov, H. Ahmed, An Approach to Real-Time Color-based Object Tracking, *In Proc. 2006 Intern. Symp. on Evolving Fuzzy Systems*, 7-9 Sept., 2006, Ambelside, UK, IEEE Press, pp.81-87, ISBN 0-7803-9719-3.

[23] P. X. Liu, and M. Q.-H. Meng, "Online Data-Driven Fuzzy Clustering with Applications to Real-Time Robotic Tracking", *IEEE Trans. on Fuzzy Syst.,* special issue on Robotics, **vol. 12**, 2004, pp.516-523.

[24] Carline D., P. Angelov, R. *Clifford, Agile Collaborative Autonomous Agents for Robust Underwater Classification Scenarios,* International Conference on Underwater Defense Technology, Amsterdam, June 2005.

[25] P. Carinena, C.V. Regueiro, A. Otero, A.,J., Bugarin, S. Baro, "Landmark Detection in Mobile Robotics using Fuzzy Temporal Rules", *IEEE Trans. on Fuzzy Syst., special issue on Robotics*, **vol. 12**, 2004, pp.423-435.

[26] D. Schoerter, I. Posner, K. Ho, P. Newman, "Using Vision for Outdoor-SLAM", *SEAS-DTC Conf.*, Edinburgh, UK, July 2006, on CD-ROM.

[27] D. Anguita, G. Parodi, and R. Zunino. "*Neural structures for visual motion tracking". Machine Vision and Applications,* 8:275-288, 1995.

[28] R. C. Gonzalez and R. E. Woods, "*Digital Image Processing",* 2nd edition, Prentice Hall, 2002. ISBN: 0201180758.

[29] Astrom K., B. *Wittenmark Computer Controlled Systems: Theory and Design,* NJ USA: Prentice Hall, 1984.

VI. APPENDIX

All the data, *x* processed in real-time are being normalized by current $x^c_{\min}$ and r^c :

$$x^c = \frac{x - x^c_{\min}}{r^c} \qquad \text{(A1)}$$

From (A1) we get:

$$x = x^c r^c + x^c_{\min} \qquad \text{(A2)}$$

When the boundary of the data space is updated, *x* is than normalized by the new (updated) boundary, $x^n_{\min}$ and the new range, and r^n :

$$x^n = \frac{x - x^n_{\min}}{r^n} \qquad \text{(A3)}$$

Expressing *x* in (A3) by (A2), we get:

$$x^n = \frac{x^c r^c + x^c_{\min} - x^n_{\min}}{r^n} = \rho x^c + \delta \qquad \text{(A4)}$$

where ρ and δ are defined in section *III B*.

Therefore, the existing prototypes and x_{k-1} are updated by (16a) and (16b).

By definition, (9b) we have:

$$f_{k-1} = x^c_1 + x^c_1 + \cdots + x^c_{k-1} \qquad \text{(A5a)}$$

Normalizing it using the new range we get:

$$f_{k-1} = (\rho x^c_1 + \delta) + (\rho x^c_2 + \delta) + \cdots + (\rho x^c_{k-1} + \delta) \qquad \text{(A5b)}$$

By reorganizing we arrive at (16c).

From (A4), we can have:

$$(x^n)^2 = (\rho x^c + \delta)^2 = \rho^2 x^c + 2\rho\delta x^c + \delta^2 \qquad \text{(A6)}$$

One can express (9a) as:

$$b_{k-1} = \sum_{j=1}^{n} b^j_{k-1}\,;\, b^j_{k-1} = (x^j_1)^2 + (x^j_1)^2 + \cdots (x^j_{k-1})^2 \qquad \text{(A7)}$$

Normalizing by the new range and taking (A6) into account we get:

$$\begin{aligned} b^j_{k-1} &= [\rho^2 (x^{jc}_1)^2 + 2\delta\rho x^{jc}_1 + \delta^2] + \cdots + [\rho^2 (x^{jc}_{k-1})^2 + 2\delta\rho x^{jc}_{k-1} + \delta^2] = \\ &= \rho^2 [(x^{jc}_1)^2 + \cdots (x^{jc}_{k-1})^2] + 2\delta\rho (x^{jc}_1 + \cdots + x^{jc}_{k-1}) + (k-1)\delta^2 \end{aligned} \qquad \text{(A8)}$$

Combining (A5a), (A7) and (A8), (16d) is obvious.

Let us denote:

$$D^c = \sum_{l=1}^{k} \sum_{j=1}^{n} (x^{*c}_j - x^{jc}_l)^2 \qquad \text{(A9)}$$

From definition of potential for prototype, we have:

$$P_{k-1}(x^*)^c = \frac{1}{1 + \frac{1}{k-1}(D^c)^2} \qquad \text{(A10)}$$

By rearranging (A10), we have:

$$D^c = (k-1)\left(\frac{1}{P_{k-1}(x^*)^c} - 1\right) \qquad \text{(A11)}$$

Updating the normalization of D^c using (A2) we get:

$$D = \sum_{l=1}^{k} \sum_{j=1}^{n} [(\rho x^{*c}_j + \delta) - (\rho x^{jc}_l + \delta)]^2 = \rho^2 D^c \qquad \text{(A12)}$$

Therefore, the update of the potential of the previously existing prototypes can be expressed as:

$$P_{k-1}(x^*) = \frac{1}{1 + \frac{1}{k-1}\rho^2 (D^c)^2} \qquad \text{(A13)}$$

Substitute (A11) into (A13), we finally reach (16e).

By definition [3], $\sigma^i_{jk-1} = \sqrt{\frac{1}{S^l_{k-1}} \sum_{l=1}^{S^l_{k-1}} \left\| x^{i*} - x_l \right\|^2_j}$. Using (A2), we easily express σ^i_{jk-1} as:

$$\sigma^i_{jk-1} = \sqrt{\frac{1}{S^l_{k-1}} \sum_{l=1}^{S^l_{k-1}} \left\| (\rho x^* + \delta) - (\rho x_l + \delta) \right\|^2_j} \qquad \text{(A11)}$$

Reorganizing (A11) we get $\sigma^i_{jk-1} = \sqrt{\frac{\rho^2}{S^l_{k-1}} \sum_{l=1}^{S^l_{k-1}} \left\| x^{i*} - x_l \right\|^2_j}$

Therefore we reach (16f): $\sigma^i_j = \rho \sigma^{ic}_j \;\; ; j = [1, n]$.