Identification of Underwater Mines from Electro-Optical Imagery using an Operated-Assisted Reinforcement On-Line Learning

J. Salazar and M.R. Azimi-Sadjadi
Department of Electrical and Computer Engineering
Colorado State University
Fort Collins CO, 80523
Email: azimi@engr.colostate.edu

***Abstract*— This paper presents a new approach for using an operated-assisted reinforcement on-line learning for mine identification from electro-optical images. The images acquired from Streak Tube Imaging Lidar (STIL) that constitute contrast and range maps are used. A reduced set of features using the Zernike moments is extracted from each preprocessed and detected/segmented object. This set is fed to a flexible network which uses a new on-line reinforcement learning based on expert operator's votes. An important feature of this system is that it allows for the incorporation of new objects learning without deleting or modifying the previously learnt cases. The performance of this preliminary in-situ learning system will be demonstrated in this paper on several STIL images and the confusion matrix of the overall system will be presented.**

I. Introduction

The identification of underwater mines presents a major challenge for any automatic recognition system. Among the factors that impair the performance of these systems are: Target features change significantly with the aspect angle variation and target range. The surface characteristics and reflectivity are highly variable, thus impacting the signal-to-noise ratio (SNR). Targets may be obscured by biological sources such as vegetation or plankton in the water column or on the sea bottom. Natural and man-made clutter can increase the occurrence of false alarm. Environmental conditions and in particular the variations in the turbidity can substantially change the quality of the collected data and hence the conspicuousness of the targets. Presently, laser-based optical devices that offer not only good optical contrast images but also range maps with depth information are being used [1] [2]. One of these devices is the electro-optical identification (EOID) sensor embedded in the Streak Tube Imaging Lidar (STIL). The identification method proposed here basically involves five major steps: preprocessing, detection, segmentation, feature extraction and classification.

The first step of the identification method involves some low-level preprocessing of the contrast images in order to enhance the target signatures and to reduce the effects of noise and sensor artifacts. Typical methods are image transformations and linear or non-linear filtering [3]. Preliminary studies performed on the contrast images indicated that application of a windowed 3x3 2D Wiener filer followed by a 3x3 median filter applied in a raw-wise manner, helps to remove line structures caused by the scanning mechanism of the STIL sensor. In [4], an adaptive order statistic filter was used to suppress noise with long heavy-tailed density functions. It is well known that the median filter doesn't produce good results in the presence of Gaussian noise. However, our preliminary study using the Lilliefor's test [5] didn't give any evidence for normality assumption. Moreover, the application of the median filter suppressed some of the noise associated with the sensor, most importantly the line artifacts introduced by the scanning mechanism.

The second and third steps of the process involve object detection and a robust and reliable segmentation scheme. Variational methods [6], [7] are widely used for image segmentation purposes. The goal of these methods is to the find an energy function and minimize that subject to one or several meaningful constraints. The variational method used here for segmenting potential targets purposes is based upon the gradient vector flow (GVF) snake [8] - [9]. For the snake, the best function $\Gamma(s) = (x(s), y(s))$ represents the set of points over a closed boundary and s is the arc length of the contour.

Once the desired contour of the segmented is obtained using the GVF snake algorithm that will be described later, the next task is to find a representative and salient set of features to represent targets (mines) and non-target objects. The challenge in this task is to extract a low dimensional and robust shape-dependant feature vector that captures all the important attributes of the segmented objects. To this end, a reduced set of shape dependant features based on the Zernike moments [10], [11] is extracted from the range and contrast maps of each detected/segmented object. This set contains information about the binary silhouettes of each object. These features are invariant to object rotation, translation and size scaling. Additionally, Zernike moments are insensitive to noise and nominal distortion.

The last step in our underwater target/non-target identification system consists of a flexible network that is capable of learning based not only on well-established information such as the class membership of any particular objet (target/non-target) but also from the information provided by users of the system. The network's structure is formed of three layers: two static and one flexible. The input (first) layer receives the set of features extracted from each object and propagates these features to the units of the second layer. The second (flexible) layer then maps the input feature vector via a linear or non-linear transformation to another vector with the same

dimension. The third layer acts as a retrieval layer which is used to score on the detected objects. This network allows for the expert operator to evaluate the results of the classification and retrieval and then provide on-line reinforcement, if needed. The learning mechanism is derived on the basis of the existence of correlation between features for any object. The capability of leaning from the user and from the image identified by the user during the feedback process is a special property of our learning scheme.

The organization of this paper is as follows: Section II presents the segmentation method used in this paper. For this method, a 5-steps procedure is presented at the end of the section along with a mathematical review of the snake algorithm. Section III describes the feature extraction method for which a small set of Zernike moments is used as features. Section IV introduces the connectionist system with linear cells used for feature mapping and retrieval. The training algorithm for this network is presented in Section V. The last section gives the results of the classification scheme realized over three targets and two non-targets objects.

II. Segmentation via the GVF snake method

Snakes, or active contours, are used extensively in the computer vision and image processing applications. Snake are curves defined within an image domain that can move under the influence of internal forces coming from within the curve itself and external forces computed from the image data. The internal and external forces are defined so that the snake will conform to an object boundary or other desired features within an image. Snakes are typically used in applications such as edge detection [8] , shape modeling [12] and segmentation and motion tracking [13].

A traditional snake is a curve $X(s) = [x(s), y(s)]$,$s \in [0, 1]$ that moves through the spatial domain of an image to minimize the energy function

$$E = \int_0^1 \{1/2[\alpha \left|X'(s)\right|^2 + \beta \left|X''(s)\right|^2] + E_{ext}(X(s))\}ds \tag{1}$$

where α, β are weighing parameters that control the snake's tension and rigidity respectively, and $X'(s)$ and $X''(s)$ denote the first and second derivatives of $X(s)$ with respect to s. The external energy function E_{ext} is derived from the image so that the energy function E takes on small values at the features of interest, such as boundaries. Given a gray level image, $f(x, y)$, typical external energies designed to lead an active contour toward the edges are:

$$E_{ext}(x, y) = -|\nabla f(x, y)|^2 \quad \text{or}$$
$$E_{ext}(x, y) = -|\nabla (G_\sigma(x, y) * f(x, y))|^2$$

where $G_\sigma(x, y)$ represents a 2-D Gaussian function with standard deviation σ and ∇ is the gradient operator. Here, we use Canny edge operator [3], [14] to generate the binary edge map, $I(x, y)$ and then use

$$E_{ext}(x, y) = 1 - I(x, y) \tag{2}$$

A snake that minimizes E must satisfy the Euler equation

$$\alpha X''(s) - \beta X''''(s) - \nabla E_{ext} = 0 \tag{3}$$

Figure 1 shows an example of how GVF snake is working. Figure 1(a) demonstrates the progress of deformation and convergence of the snake to the edges. The outer circle is the initial snake. As can be seen, the GVF snake has a large capture range, i.e. the snake can converge to the boundaries of the object even when the initial snake is far from the actual boundaries. This characteristic shows the robustness of this method to the initial choice of snake. Figures 1(b)-(d) show the edge map captured, the gradient of the edge map and the gradient vector flow, respectively.

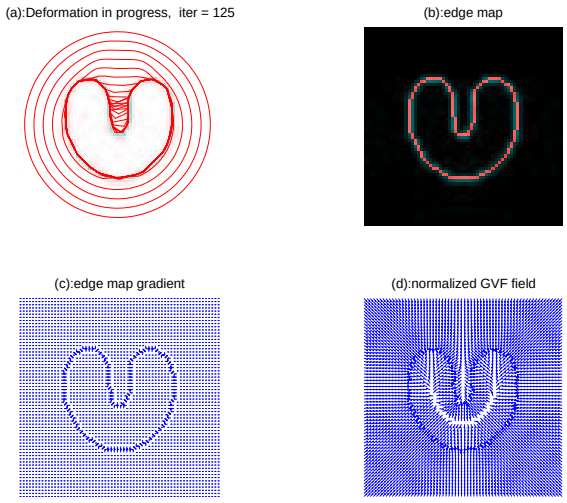


Fig. 1. An Example of GVF Snake Process

In this paper, we form the GVF snake using the following procedure:

1: To reduce the sensitivity of the snake to missing edges, sub-sample the original images and generate low-resolution versions. This could also be done using 2-D discrete wavelet transform [3].

2: Use Canny edge detection operator [3], [14] to get the binary edge map.

3: Based on the edge map as the external force, derive the gradient vector flow.

4: Set the initial snake. Since the target recognition should be done automatically, the initial snake should also be set automatically. The image center is computed using the image geometric moments described by $m_{k,l} = \sum_{\forall x,y} x^k y^l I(x, y)$, where $I(x, y)$ is the image intensity function at pixel (x, y). Therefore $xcenter = m_{10}/m_{00}$, $ycenter = m_{01}/m_{00}$. This is used as the center of the initial snake. Then, for all the possible edge points on the binary edge map, the distances between the edge points to the snake center are calculated,

and the median value of the distance is used at the radius of the initial snake.

5:Update the snake until the snake reaches the stable state.

Once the binary silhouettes of the detected objects are formed using GVF snake, shape dependent features using Zernike moments are extracted. The method is presented next.

III. Feature extraction using Zernike moments

Zernike moments form a complete set of complex polynomials defined in the interior of the unit circle. For a discrete image (e.g a window containing the segmented object, $f(x, y)$, the Zernike moments of order n with repetition m where $|m| \leq n$ and $n - |m|$ constrained to an even number, are give by

$$A_{nm} = \frac{n+1}{\pi} \sum_x \sum_y f(x, y) V^*_{nm}(\rho, \theta) \tag{4}$$

where $x^2 + y^2 \leq 1$ i.e. confined to the interior of the unit circle, ρ is the length of a vector from the origin of the unit circle to (x, y) point, θ is the corresponding phase angle, and $V(\rho, \theta)$form a complete set of orthogonal complex polynomials over the unit circle. These are defined by

$$V_{nm}(\rho, \theta) = R_{nm}(\rho) e^{jm\theta} \tag{5}$$

where the radial polynomial $R_{nm}(\rho)$ is given by

$$R_{nm}(\rho) = \sum_{s=0}^{n-|m|/2} \frac{(-1)^s [(n-s)!] \rho^{n-2s}}{s!(n + |m|/2 - s)!(n - |m|/2 - s)!} \tag{6}$$

Note that $R_{n,-m}(\rho) = R_{nm}(\rho)$ and also $A^*_{nm} = A_{n,-m}$. Clearly, the magnitude of the Zernike moments gives rotation invariant features. To make the moments translation invariant, the image is transformed to a new coordinate system by moving the origin to the centroid prior to the moment calculation. Then the first-order moments of the new image become zero. To achieve scale invariance, the image is resized by making its zeroth-order moment to a predetermined value. Thus a general mapping of type $f^{new}(x, y) = f(\overline{x} + \frac{x}{a}, \overline{y} + \frac{y}{a})$ where $(\overline{x}, \overline{y}$ are coordinates of the centroid of the original image $f(x, y)$and a is a scaling parameter. The scale and translation normalization process affects two of the Zernike features namely $|A_{00}|$ and $|A_{11}|$. Thus, $|A_{00}|$will be the same for all the images and $|A_{11}|$will be zero. As a result, these moments will not be included in the extracted feature set and the selected features start from the second-order moments. The great benefits of these moments are the orthogonal property of the expansion and their immunity to noise and distortion [15], [16] comparing to the regular moments.

IV. Flexible network classifier

In this section an adaptable information search and retrieval system (see Fig. 2) for identification of underwater mines is introduced where the end results of the search are evaluated by an expert operator. This on-line interaction between human and machine could help, through time, to reduce the number of mis-classification errors and to incorporate, in a supervised fashion, new targets or non-targets into the machine's 'knowledge'. Similar applications are encountered in a wide variety of problems including but not limited to facial pattern identification using security cameras and computer-aided detection of tumors.

A typical image retrieval system, as shown in Figure 2(a), comprises four units: storage, image indexing, user interface and search/retrieval subsystems. The image indexing unit processes each document or image in the database and generates an indexed file based upon certain attributes in the document or sets of features associated with different objects in the image. The information of each document/image is stored in the database with possibly a label that identifies a class for each pattern.

The flexible part of the system consists of a three-layer network (Fig. 2(b)). The first layer receives the input feature image and distributes each component to the units of the second layer which then maps, via a linear transformation, this feature vector into another one with more discriminative capability. The third layer or "retrieval layer" performs search and information retrieval by mapping the input image to the image indices (labels) and generating a rank or score for each stored image based upon a similarity measure that will be defined later.

The second layer in this network is a flexible layer referred to as "feature mapping layer", which maps the original feature vector by incorporating relevance feedback and high-level knowledge from the expert users. The weights of the third layer are obtained from the set of training vectors that represent the importance of each the image. Figure (3) shows this three-layer network structure, which can easily be trained to provide a flexible learning engine for inferring feature (query) concept association and clustering of queries on features. The mapping layer enhances or assists the task of specifying the required information in the image by augmenting the user's original feature vector with information extracted from the relevant images (mines/non-mines) and identified by previous expert users (high-level) to be most relevant. This is accomplished through an on-line iterative learning that allows for relevance feedback from the user and accounts for changes in the data and/or environmental conditions. The learning dynamically builds statistical associations between the images and known solutions identified by the expert users. When an image is marked as relevant or correctly tagged by the expert user/operator, the proposed architecture goes through an iterative on-line learning to extract additional low-level attributes or concepts from the selected image(s) that are statistically significant to the image, and maps the original image feature vector to an "optimal feature vector" within which these additional concepts or attributes are integrated with those high-level semantics from the expert users. This is an efficient way of specifying the required information because it frees the user from having to think about all the possible relevant attributes in the image. Instead, the user deals with

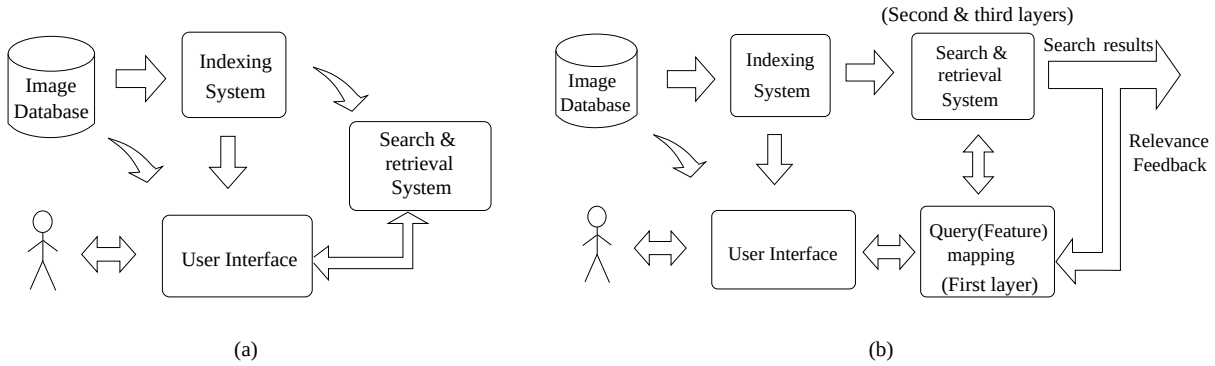


Fig. 2. (a) Typical and (b) Adaptable Image Retrieval Systems.

the actual images. The user can also define certain constraints on the retrieved results while imposing relevance feedback. For instance, the user may select one image as the most relevant to be elevated to the first place while maintaining the other images identified by the previous users and/or the search engine as relevant.

The breakthrough advantage of the approach is the ability for the expert users to contribute to the decision-making capability of the system and to enhance the relevancy of the suggested solutions without the slow and expensive process of authoring or modifying the information content within the retrieval system directly. In this model, the efficiency of the system improves over time as users actively provide relevancy information in the context of their needs. It is very important to note that the layers in this network are simple linear, easily trainable, scalable, and amenable for real-life implementation.

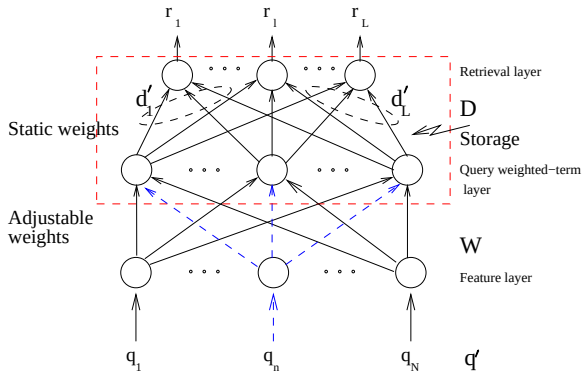


Fig. 3. Proposed Three-Layer Query Mapping an Retrieval System.

V. Training Algorithm

The training process of the system starts by splitting the training data into two sets: one set is used for setting up the connections D of the retrieval layer, and the other, called the validation set, serves to adjust the parameters $\mathbf{W}$ of the first layer (Feature mapping layer). Even though the retrieval layer is static layer, the selection of the 'best' subset of cells from the training data is fundamental to good generalization of the system.

Consider again the retrieval system shown in Fig. 2(a), when an unknown query $\mathbf{q}$ (feature vector or an image) is submitted to the system, the search and retrieval unit uses the previously stored images (documents) to assign a similarity measure or rank $r(.)$ for each of these documents. The quantity, $r(\mathbf{d},\mathbf{q})$, indicates the importance of document $\mathbf{d}$ to query $\mathbf{q}$. One measure frequently used in information retrieval systems is the 'dot' product between document $\mathbf{d}$ and query $\mathbf{q}$, i.e, $r(\mathbf{d},\mathbf{q}) = \mathbf{d}^t\mathbf{q}$. If all the stored documents and queries are normalized, this measure takes its maximum value for a document that is identical to the submitted query. However, in most practical cases, the queries and the documents stored in the system are different, hence the document with the maximum rank for a particular query is not necessarily the most relevant one . To overcome this problem, it is convenient to modify previous equation by incorporating parameters which can be fine-tuned. For instance, the rank function $r_{\mathbf{S}}(.)$ with parameter set $\mathbf{S}$ can be defined as follows

$$r_{\mathbf{S}}(\mathbf{d},\mathbf{q}) = \frac{(\mathbf{Sd})^t\mathbf{Sq}}{\|\mathbf{Sd}\|\|\mathbf{Sq}\|} \tag{7}$$

This implies that both the documents and applied query are mapped using matrix S to yield desired ranking. Additionally, we can write

$$r_{\mathbf{S}}(\mathbf{d},\mathbf{q}) = \frac{\mathbf{d}^t}{\|\mathbf{Sd}\|}.\mathbf{S}^t\mathbf{S}.\frac{\mathbf{q}}{\|\mathbf{Sq}\|} \tag{8}$$

$$= \frac{\mathbf{d}^t}{\sqrt{\mathbf{d}^t\mathbf{W}\mathbf{d}}}.\mathbf{W}.\frac{\mathbf{q}}{\sqrt{\mathbf{q}^t\mathbf{W}\mathbf{q}}} \tag{9}$$

The latter implementation of this ranking scheme is incorporated in the network in Fig. 3. The normalized query image $\mathbf{q}' = \mathbf{q}/\sqrt{\mathbf{q}^t\mathbf{W}\mathbf{q}}$ is first presented to the input layer and then mapped in another layer via a linear transformation $\mathbf{W} = \mathbf{S}^t\mathbf{S}$ that is symmetric and positive definite. This mapped query is then applied to the retrieval layer with normalized weights that are the stored documents $\mathbf{d}' = \mathbf{d}/\sqrt{\mathbf{d}^t\mathbf{W}\mathbf{d}}$ embedded in matrix $\mathbf{D}$. This layer then provides the scores.

A. In Time Forward-backward Learning Algorithm

In the relevance feedback used in many retrieval systems, it is customary to modify the original query $\mathbf{q}$ when the list of documents and their ranks are not the desired ones. This method is called query expansion. Rocchio's formula [17] is commonly used in many text-oriented search engines for query expansion. This is given by

$$\hat{\mathbf{q}}(t+1) = \mathbf{q}(t) + \alpha * \frac{\sum_{\mathbf{d}_i \in \mathbf{R}} \mathbf{d}_i}{\|\mathbf{R}\|} - \beta * \frac{\sum_{\mathbf{d}_i \in \mathbf{NR}} \mathbf{d}_i}{\|\mathbf{NR}\|} \quad (10)$$

where $\mathbf{q}(t)$ represents the query applied at time t, $\mathbf{R}$ is the set of relevant documents for that query, $\|\mathbf{R}\|$ represents the cardinality of set $\mathbf{R}$ and $\mathbf{NR}$ is the set of non-relevant documents. The "optimal" query $\hat{\mathbf{q}}(t+1)$ is the query that, when applied to the system at time $t+1$, delivers better results than those delivered by the system at the time when the original query $\mathbf{q}(t)$ was applied.

There could be several drawbacks when this formula is used: First, the lists of relevant and non-relevant documents is generally unknown because the user has neither the patience nor the time to identify these set for every submitted query. Second, it has been demonstrated that the formula works well with some retrieval models such as the binary model [18]. However, its effectiveness in the image retrieval area for any arbitrary model hasn't been demonstrated yet. Furthermore, its capability to transfer what has already been learnt with one user to other users is limited since the formula is one-user oriented. To alleviate these problems, the forward-backward method developed here uses a predicted query to adjust the parameters of the retrieval system.

Let us divide the process of finding the parameters in two consecutive stages. At stage k, when query $\mathbf{q}$ is presented, a retrieval system delivers a set of documents with their respective scores according to the function $L_{\mathbf{W}(k)}(\mathbf{q}) = f(\mathbf{W}(k), \mathbf{q})$, where $\mathbf{W}(k)$ is the set of parameter at stage k. Here, each query vector $\mathbf{q}$ is in the set of training queries $\mathbf{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_i, \ldots, \mathbf{q}_N\}$. This set is known and remains unchanged regardless of the stage.

Now, assume that at stage k when query $\mathbf{q}_i$ is submitted, a user compares the list he/she already expects $L_{user}(.)$ with the list of documents delivered by the system $L_{\mathbf{W}(k)}(.)$ and assigns a satisfaction measure to the latter list according to some positive stochastic function $g(.)$

$$\xi_{\mathbf{W}(k)}(\mathbf{q}_i) = g(L_{user}(\mathbf{q}_i), L_{\mathbf{W}(k)}(\mathbf{q}_i)), \qquad 1 \leq i \leq N \quad (11)$$

where $\xi_{\mathbf{W}(k)}(\mathbf{q}_i)$ gives the user satisfaction measure for a system with parameters $\mathbf{W}(k)$ when query $\mathbf{q}_i$ is applied. In this formula a value equals zero means that the user is totaly satisfied with the list of documents delivered by the system. Now, we know that for cases in which the user is not satisfied the application of a predicted query $\hat{\mathbf{q}}$ in (10) at stage $(k+1)$ leads to a better level of user's satisfaction. Thus,

$$\xi^1_{\mathbf{W}(k)}(\hat{\mathbf{q}}_i(k+1)) = g(L_{user}(\mathbf{q}_i), L_{\mathbf{W}(k)}(\hat{\mathbf{q}}_i(k+1))), 1 \leq i \leq N \quad (12)$$

represents the satisfaction level of the user at stage $k+1$ using the old parameter matrix $\mathbf{W}(k)$ (those parameters that the system had at stage k) and the predicted query $\hat{\mathbf{q}}_i(k+1)$ using for instance Rocchio's formula.

Assuming that the predicted query delivers a better list of documents/ranks, we can write

$$\xi^1_{\mathbf{W}(k)}(\hat{\mathbf{q}}_i(k+1)) \leq \xi_{\mathbf{W}(k)}(\mathbf{q}_i)$$

Now, the remaining problem is to find the matrix $\mathbf{W}(k+1)$ that yields the same desired results with the original query $\mathbf{q}_i$, i.e.

$$\begin{aligned} &g(L_{user}(\mathbf{q}_i), L_{\mathbf{W}(k+1)}(\mathbf{q}_i)) = \xi^1_{\mathbf{W}(k)}(\hat{\mathbf{q}}_i(k+1)) \\ &= g(L_{user}(\mathbf{q}_i), L_{\mathbf{W}(k)}(\hat{\mathbf{q}}_i(k+1))) \end{aligned} ; 1 \leq i \leq N \quad (13)$$

Therefore, the in-time forward-backward learning rule is aimed at solving for $\mathbf{W}(k+1)$ to satisfy (13). This rule is stated next and illustrated through an example in the next section.

If it is known that a change of an input variable will produce a better output of a system at time $t+1$, then instead of changing the variable at time $t+1$ adjust the parameters of the system itself in such a way that the output will be the same as that of the modified input variable with the old system parameters.

Using (13), the search for $\mathbf{W}(k+1)$ can also be formulated as a minimization problem

$$\begin{aligned} \mathbf{W}_{opt}(K+1) = min_{\,\mathbf{W}(k+1)} \textstyle\sum_i |&g(L_{user}(\mathbf{q}_i), L_{W(k+1)}(\mathbf{q}_i)) \\ &- g(L_{user}(\mathbf{q}_i), L_{W(k)}(\hat{\mathbf{q}}_i(k+1)))| \end{aligned} \quad (14)$$

The process of finding the parameters at stage $k+1$ is best illustrated in Fig. 4. At stage k, based on the list of documents delivered by the system when query $\mathbf{q}_i$ is submitted, a user forms a new predicted query $\hat{\mathbf{q}}_i$ with the hope that this query will deliver a better list of documents at stage $k+1$. At stage $k+1$ the goal is to change the parameters in a fashion that the output for the initial query $\mathbf{q}_i$ with these modified set of parameters will be the same as that of the predicted query $\hat{\mathbf{q}}_i$ using the old parameters $\mathbf{W}(k)$.

There are basically three steps for the formulation of the algorithm. The first step involves the definition of an appropriate error function. The second step entails finding the predicted query that reduces the error function. And the last step uses this predicted query to find the modified parameters of the retrieval system. The first task is difficult since, except for the user who submits a query, no one else knows exactly what the user wants. Two assumptions are made to define an appropriate error function. These are:

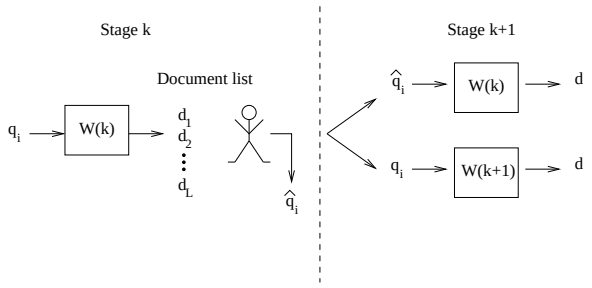


Fig. 4. Adaptable model for the in time forward-backward algorithm

1) From the list of documents delivered by the retrieval system for a particular query $\mathbf{q}$ a user singles out only one document, e.g. $\mathbf{d}_R$.
2) All the documents with scores greater or equal to the score of $\mathbf{d}_R$ are considered non-relevant, therefore their scores should be less than that of $\mathbf{d}_R$.

Based on the these assumptions, there exist many different ways to define an error function $J(.)$[1] suitable for the retrieval system shown in Fig. 3. Using (8), one possible choice is as follows

$$J(\mathbf{S},\mathbf{q}) = \frac{\sum_{\mathbf{d}_j\in\chi} r_{\mathbf{S}}(\mathbf{d}_j,\mathbf{q})}{K} - r_{\mathbf{S}}(\mathbf{d}_R,\mathbf{q}) \tag{15}$$

$$= \left[\frac{\sum_{\mathbf{d}_j\in\chi}\frac{(\mathbf{S}\mathbf{d}_j)^t}{\|\mathbf{S}\mathbf{d}_j\|}}{K} - \frac{(\mathbf{S}\mathbf{d}_R)^t}{\|\mathbf{S}\mathbf{d}_R\|}\right] * \frac{(\mathbf{S}\mathbf{q})}{\|\mathbf{S}\mathbf{q}\|} \tag{16}$$

where χ is the set of non-relevant documents defined by $\chi = \{\mathbf{d}_j | r_{\mathbf{S}}(\mathbf{d}_j,\mathbf{q}) \geq r_{\mathbf{S}}(\mathbf{d}_R,\mathbf{q})\}$ and K is its cardinality.

Now, using the normalized query vector $\mathbf{q}' = \mathbf{q}/\|\mathbf{S}\mathbf{q}\|$ the error function can be rewritten as follows

$$J(\mathbf{S},\mathbf{q}') = \left[\frac{\sum_{\mathbf{d}_j\in\chi}\frac{(\mathbf{S}\mathbf{d}_j)^t}{\|\mathbf{S}\mathbf{d}_j\|}}{K} - \frac{(\mathbf{S}\mathbf{d}_R)^t}{\|\mathbf{S}\mathbf{d}_R\|}\right] * (\mathbf{S}\mathbf{q}') \tag{17}$$

The gradient of this function with respect to $\mathbf{q}'$ is found to be

$$\nabla J_{\mathbf{q}'} = \mathbf{S}^t * \left[\frac{\sum_{\mathbf{d}_j\in\chi}\frac{\mathbf{S}\mathbf{d}_j}{\|\mathbf{S}\mathbf{d}_j\|}}{K} - \frac{\mathbf{S}\mathbf{d}_R}{\|\mathbf{S}\mathbf{d}_R\|}\right] \tag{18}$$

Using this gradient result, the second step, as mentioned before, involves finding the predicted query $\hat{\mathbf{q}}$ that reduces the error function $J(.)$ as expressed in (17). This is done using

$$\hat{\mathbf{q}} = \mathbf{q}' - \lambda\mathbf{H} * \nabla J_{\mathbf{q}'} \tag{19}$$

where H is a positive definite matrix and λ is the step size that is a small but positive scalar.

This learning rule can be simplified by choosing $\mathbf{H} = (\mathbf{S}^t\mathbf{S})^{-1}$. Now, using (18) and $\mathbf{H} = (\mathbf{S}^t\mathbf{S})^{-1}$, (19) can be rewritten as

[1]Here, $J(.)$ plays the same role as $\xi(.)$ in (11)

$$\hat{\mathbf{q}} = \mathbf{q}' - \lambda * \left[\frac{\sum_{\mathbf{d}_j\in\chi}\frac{\mathbf{d}_j}{\|\mathbf{S}\mathbf{d}_j\|}}{K} - \frac{\mathbf{d}_R}{\|\mathbf{S}\mathbf{d}_R\|}\right] \tag{20}$$

Rewritten (20) in terms of the normalized document vectors yields

$$\hat{\mathbf{q}} = \mathbf{q}' - \lambda * \left[\frac{\sum_{\mathbf{d}_j\in\chi}\mathbf{d'}_j}{K} - \mathbf{d'}_R\right] \tag{21}$$

Note the resemblance of this equation with Rocchio's formula (10) using the normalized query and document vectors, also note that the normalized vectors depend on the parameter matrix $\mathbf{W}(k)$.

Finally, the last step of our formulation involves the use of the predicted query and the error function. To this end, it is better to reformulate the error function (17) as

$$J(\mathbf{W},\mathbf{q}') = \left[\frac{\sum_{\mathbf{d}_j\in\chi}\frac{(\mathbf{d}_j)^t}{\sqrt{\mathbf{d}_j{}^t\mathbf{W}\mathbf{d}_j}}}{K} - \frac{(\mathbf{d}_R)^t}{\sqrt{\mathbf{d}_R{}^t\mathbf{W}\mathbf{d}_R}}\right] * (\mathbf{W}\mathbf{q}') \tag{22}$$

Now, the use of $J(.)$ as the error function in (13) leads to

$$J(\mathbf{W}(k+1),\mathbf{q}') = J(\mathbf{W}(k),\hat{\mathbf{q}}) \tag{23}$$

When the step size λ is close to zero, it is logical to assume that the change in the quantity in brackets in (22) is insignificant. Therefore from (23) the learning rule can be stated as follows

$$\mathbf{W}(k+1)\mathbf{q}'(k) = \mathbf{W}(k)\hat{\mathbf{q}}(k) \tag{24}$$

When this equation holds for all queries in the set of training queries $\mathbf{Q} = [\mathbf{q}'_1\mathbf{q}'_2...\mathbf{q}'_N]$ with their corresponding predicted queries $\hat{\mathbf{Q}} = [\hat{\mathbf{q}}_1\hat{\mathbf{q}}_2...\hat{\mathbf{q}}_N]$, it is easy to arrive at the updating equation

$$\mathbf{W}(k+1) = \mathbf{W}(k)\hat{\mathbf{Q}}\mathbf{Q}^t(\mathbf{Q}\mathbf{Q}^t)^{-1} \tag{25}$$

Remarks:

(1) The updated matrix $\mathbf{W}(k+1)$ in (25) could be non-positive definite, therefore a method must be applied to find its 'closest' positive definite matrix and then assign this matrix to $\mathbf{W}(k+1)$. Such method is the modified Cholesky algorithm [19].
(2) To avoid the vanishing of the error function (17) as $\mathbf{S} \rightarrowtail \mathbf{0}$ it is convenient to normalize the matrix $\mathbf{W}(k+1)$ by setting $\|\mathbf{W}(k+1)\| = 1$.

VI. Results and Discussions:

In this paper, all the STIL images acquired from CSS are used for target identification. Five different classes of objects are identified in this database. These are: man-made circular target (class 1), circular target (class 2) , man-made-rectangular-shape target (class 3) , rectangular-shape target (class 4), and bullet-shape target (class 5) . There were plenty

(48) of class 1 objects in the database, and hence all those images are used for training and testing of the classifiers. For the other four types of objects, since the number of images was not enough both the original and synthetically distorted silhouettes are used. The numbers of actual and synthetically distorted silhouettes for each object type are given in Table 1 below. For each type of objects, 24 silhouettes are selected for training while the rest are used for testing. Note that the binary silhouettes are obtained using the procedure given in Section II. (Also see [[20]])

TABLE I

NUMBER OF ACTUAL AND SYNTHETICALLY GENERATED CASES FOR EACH TYPE OBJECT.

	# of Actual	# of Distorted
Type 1	48	0
Type 2	11	37
Type 3	10	38
Type 4	17	31
Type 5	11	37

In this study the seventh order Zernike moments are used to generate the feature vector of dimension 18x1 for each object. However, owing to the fact that different type objects may have similar shape-dependent features, additional features based on the gray level of the contrast were also used. To this end, the mean and the standard deviation of each contrast image was included into the feature vector, leaving the total number of features at 20.

The classifier was a three-layer network. Different number of cells in the retrieval layer were tried. The network with the best performance had 20 inputs, 21 cells in the feature mapping layer (The last cell was used to provide a bias and had a value equal one), and 120 five output cells corresponding to the five classes (24 cells assigned to each class). A learning step size of $\lambda = .01$ was used during the training phase. Two schemes for selecting the best set of cells for the retrieval layer were tested and compared. For the first scheme, the training and testing sets were selected randomly from the set of 48 images, 24 different cases in each set. Meanwhile for the second scheme the document feature vector close to the centroid of each class was selected as the representaive of the class. It was noticed that the first scheme gave better results than the second one.

Not all the trained networks gave 100% correct classification on the training data, hence only those that achieved 100% were considered. Since the validation set was also used for training, the network that was trained in fewer epochs was chosen. The confusion matrix for the best three-layer network is given in Table 2. The numbers in the bracket are the number of misclassified cases for each object. As can be seen, all (3) misclassifications ocurred for class 3 cases.

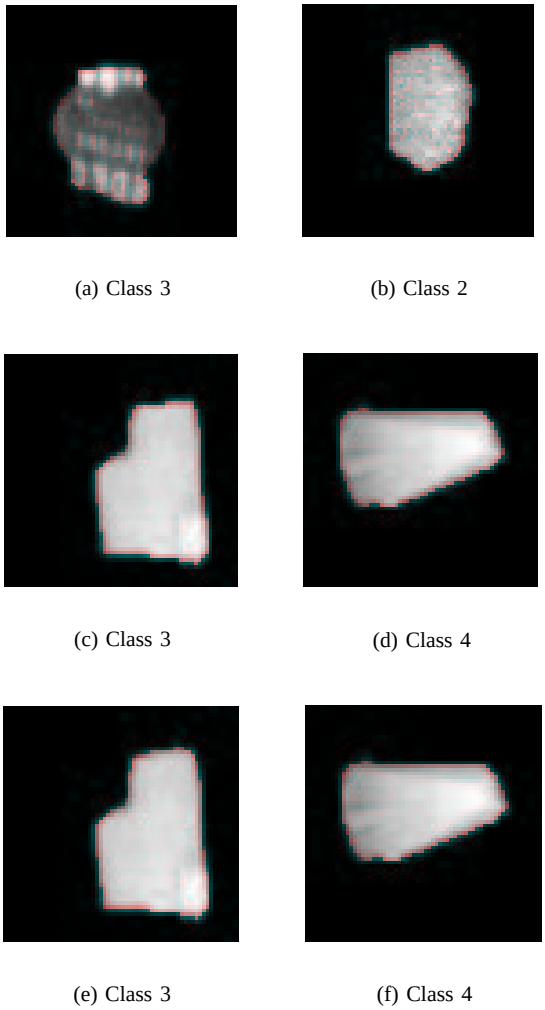
Figure (5) shows the cases from class 3 (left column) that were incorrectly classified as samples of classes 2 or 4 (right column). It seems possible that a class 3 (man made trapezoidal shape non-target) can be mistaken by a class 4

TABLE II

CONFUSION MATRIX OF THE FLEXIBLE THREE-LAYER NETWORK.

	Type 1	Type 2	Type 3	Type 4	Type 5
Type1	100%				
Type2		100%			
Type3		4%[1]	88%	8%[2]	
Type4				100%	
Type5					100%

(rectangular shape target) since they have very similar shape. The misclassification of the class 3 object, as shown in Fig. (5_a), as class 2 (see Fig. (5_b)) can also be explained as we have only used the contrast map. Clearly, the inclusion of high quality rendered range maps can significantly improve the overall performance.



(a) Class 3 (b) Class 2

(c) Class 3 (d) Class 4

(e) Class 3 (f) Class 4

Fig. 5. Class 3 objects (Left column) misclassified as objects shown in the right column.

VII. CONCLUSIONS AND FUTURE WORK

The network classifier gave good results even though all the cells used in the model were linear. For the best network

only three samples were misclassified. However, it could be possible to reduce the number of misclassified objects even further, if a non-linear feature mapping is used. In this study only the features extracted from the contrast maps are employed. It is expected that the performance of the system substantially improves when the range information is also included.

ACKNOWLEDGMENTS

This research was supported by NSWC-Coastal Systems Station (CSS) in Panama City, Florida under contract # N61331-01-M-1369. The authors would like to thank Mr. Andrew Nevis at CSS for providing the data as well as his technical support during the course of the project.

REFERENCES

[1] A. Nevis, "Adaptive background equalization and image processing applications for laser line scan data," *Proc. of The SPIE Inter. Symp. On Mine Detection and Remediation Technologies, Orlando, FL*, vol. 3710, pp. 1260–1271, 1999.

[2] W. Szymczak, W. Guo, and J. Rogers, "Mine detection using variational methods for image enhancement and feature extraction," *Proc. of the SPIE Inter. Symp. On Mine Detection and Remediation Technologies, Orlando, FO*, vol. 3392, pp. 286–296, 1998.

[3] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. New Jersey: Prentice Hall, 2 ed., 2002.

[4] M. F. Fernandez and T. Aridgides, "Adaptive order-satistic filters for sea mine classification," *Proc. of the SPIE on Detection and Remediation Technologies for Mines and Minelike Targets III, Orlando, FL.*, vol. 3392, pp. 255–263, Apr. 1998.

[5] W. J. Conover, *Practical Nonparametric Statistics*. New York: Wiley, 1980.

[6] G. Hewer, C. Kenney, and B. Manjunath, "Variational image segmentation using boundary functions," *IEEE Transactions on Image Processing*, vol. 7, pp. 1269–1282, Sep. 1998.

[7] J. Goldschneider and A. Li, "Variational segmentation by piecewise facet models with application to range imagery," *Proc. 2000 International Conference on Image Processing*, vol. 1, pp. 812–815, 2000.

[8] M. K. A. Witkin and D. Terzopoulos, "Snakes: Active models," *Int. J. Comput. Vis.*, vol. 1, pp. 321–331, 1988.

[9] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Transactions on Image Processing*, vol. 7, no. 3, 1998.

[10] S. Ghosal and R. Mehrotra, "Zernike moment-based feature detectors," *Proc. ICIP-94., IEEE International Conference on Image Processing*, vol. 1, pp. 934–938, Nov. 1994.

[11] R. Bailey and M. Srinath, "Orthogonal moment features for use with parametric and non-parametric classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 389–399, Apr. 1996.

[12] T. McInerney and D. Terzopoulos, "A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4d image analysis," *Comput. Med. Image Graph*, vol. 19, pp. 69–83, 1995.

[13] F. Leymarie and M. D.Levine, "Tracking deformable objects in the plane using an active contour model," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 617–634, 1993.

[14] V. H. Sonka and R. Boyle, *Image Processing, Analysis and Machine Vision*. PWS Publishing, 1999.

[15] A. Khotanzad and J.H.Lu, "Classification of invariant image representation using a neural network," *IEEE Trans. On Acoustics, Speech and Signal Processing*, vol. 38, pp. 1028–1038, Jun. 1990.

[16] A. Khotanzad and J.H.Liou, "Recognition and pose estimation of unoccluded three-dimensional objects from a two-dimension perspective view by banks of neural networks," *IEEE Trans. On Neural Networks*, vol. 7, pp. 897–906, Jul. 1996.

[17] R. Baeza and B. Ribeiro, *Modern Information Retrieval*. New York: Addison-Wesley, 1999.

[18] Z. Chen and B. Zhu, "Some formal analysis of roccio's similarity-based relvance feedback algorithm," in *International Symposium on Algorithms and Computation*, pp. 108–119, 2000.

[19] Gill, Murray, and Wright, *Practical Optimization*. Academic Press, 1981.

[20] G. Tao, M. Azimi-Sadjadi, and A. Nevis, "Underwater target identification using gvf snake and zernike moments," *Oceans '02 MTS/IEEE*, vol. 3, pp. 1535–1541, Oct. 2002.