PROCEEDINGS

SPIE—The International Society for Optical Engineering

Wavelet Applications

Harold H. Szu
Chair/Editor

5–8 April 1994
Orlando, Florida

SPIE P PROCEEDINGS SERIES

Volume 2242

Wavelet based progressive classification of high range resolution radar returns

John S. Baras*
Department of Electrical Engineering
and Institute for Systems Research
University of Maryland
College Park, MD 20742

Sheldon I. Wolk
Code 5750
Tactical Electronic Warfare Division
Naval Research Laboratory
Washington, D.C. 20375

ABSTRACT

We investigate the problem of fast and accurate classification of high range resolution radar returns from ships. In addition we investigate the problem of efficient organization of large databases of pulsed high resolution radar returns from multiple targets in order to economize memory requirements and minimize search time. We use synthetic radar returns from ships as the experimental data. We develop a novel algorithm for hierarchically organizing the database, which utilizes a multiresolution wavelet representation working in synergy with a Tree Structured Vector Quantizer (TSVQ), utilized in its clustering mode. The tree structure is induced by the multiresolution decomposition of the radar returns. The TSVQ design algorithm is of the "greedy" type. We demonstrate that our algorithm automatically computes the aspect graph (i.e. the simultaneous representation of compressed pulses as functions of aspect and elevation) for a single target or for a group of targets. We also develop a novel optimization framework for the simultaneous design of the wavelet basis, the Tree-Structured Vector Quantizer and the Classification rule. We describe an efficient and promising implementation consisting of an adaptive Wavelet Transform - Tree Structured Vector Quantization with Learning. We show experimental results which indicate that the combined algorithm executes orders of magnitude faster data search time, with negligible performance degradation (as measured by rate-distortion curves).

1 INTRODUCTION

The problem of automatic classification of high range resolution (HRR) radar returns when a large number of targets is possible, presents formidable algorithmic and computational difficulties. A closely related problem is related to the organization and construction of efficient and economic target model databases which will result in significant search speed-up and memory reduction. These problems are of general interest for one and two dimensional signals in general. HRR radar returns are known to be very sensitive to even small viewpoint (i.e. aspect and elevation angles) variations (of the receiver), and this adds to the difficulty and the size of the problem.

In order to develop high performance algorithms for such problems which are at the same time practical to implement (i.e. they have reasonable computational requirements) it is essential to: (a) develop systematic ways to reduce the representation of the data needed for storage and classification; and (b) develop algorithms that are amenable to massively parallel implementation. In this paper we investigate such methods based on multiresolution representations of the data and "greedy" search algorithms based on Adaptive Vector Quantization (AVQ) [11]. Wavelet-based multiresolution representations [1]—[9] of the data appear to be particularly well suited for this problem because, as we showed in [16], they are equivalent to radar returns with pulsed radars of different pulsewidths (due to the basic convolutional nature of both operations). These representations lead naturally to classification schemes that are progressive. That is a small amount of information, in the form of a coarse approximation to the return, is used first to provide partial classification and progressively finer details are added until satisfactory performance is obtained. This results in a scheme where small amounts of computation are used initially and additional computations are performed as needed, resulting in extremely fast searches while preserving high fidelity in the search. Each target is reperesented by its multiresolution aspect graph [17], which is a quantization (produced by clustering) of the space of HRR returns and view points. Using an efficient Tree-Structured Vector Quantization (TSVQ) algorithm we cluster the returns from the various viewpoints into equivalence classes according to an appropriate discrimination measure. This approach automatically accounts for the discrimination capability of the sensor and in effect it performs a quantization of the sensory data which reduces the data input to the classification algorithm by orders of magnitude. In each equivalent class a "paradigm" is selected and the collection of these typical pulses arranged in a multi-scale tree constitute the target model that is guiding the on-line classification search.

To study such schemes and their alternatives we develop a mathematical framework that is based on a combined compression-classification approach. This is done by casting the overall problem as a multi-objective optimization problem and by investigating fast algorithms for its resolution. A convex combination of the three competing performance measures: distortion, rate and Bayes missclassification cost, is constructed and is used to organize and analyze the overall algorithmic approach. This formulation makes explicitly clear the trade offs between the details in the signal representation and the efficiency of the database search that need to be considered for overall algorithm performance. Within this framework we are able to indicate how our TSVQ wavelet algorithm provides a fast and

*Also with AIMS, Inc.. Research supported by AIMS, Inc., 6110 Executive Blvd., Suite 850, Rockville, MD 20852.

accurate suboptimal scheme for compressing the data representations in a way that is suited for classification. At the same time we indicate how Learning Vector Quantization (LVQ) [5] can be succesfully employed to implement the labeling of the VQ cells required for classification in a suboptimal but fast and accurate way.

2 HIGH RANGE RESOLUTION RADAR RETURNS

High range resolution radar returns contain in their structure substantial information about the target which can be used to better identify complex targets consisting of many scatterers. This applies to many forms of radar signatures, including the amplitude of pulsed radar (PR) returns, the phase of pulsed radar returns, Doppler radars (DR), synthetic aperture radar (SAR) returns, inverse synthetic aperture radar (ISAR) returns, millimeter–wave (MM–wave) radar returns. With the increasing resolution of modern radars it is *at least theoretically* possible to store many of the possible returns (i.e returns organized according to aspect, elevation, pulsewidth etc.) of a complex target and use them in the field for target identification. The advantage of the increasing radar resolution is the availability of more detailed information, and ultimately of *specific features*, characteristic of the radar return from a specific target. The disadvantage is that these very detailed characteristics require an ever increasing amount of computer memory to be stored. The latter not only results in unfeasible memory requirements but it also slows down the search time in real field operations. It is therefore important to develop extremely efficient ways to compress the representations of high resolution data returns from real targets, and to design efficient search schemes which operate in a progressive manner on the compressed representations to recover the target identity. Wavelet theory [1]—[9] offers an attractive means for the development of the required multi-resolution representations. This can be roughly explained by the fundamental property of wavelet representations of signals to uncover the superposition of these signals in terms of different structures occurring on different time scales at different times.

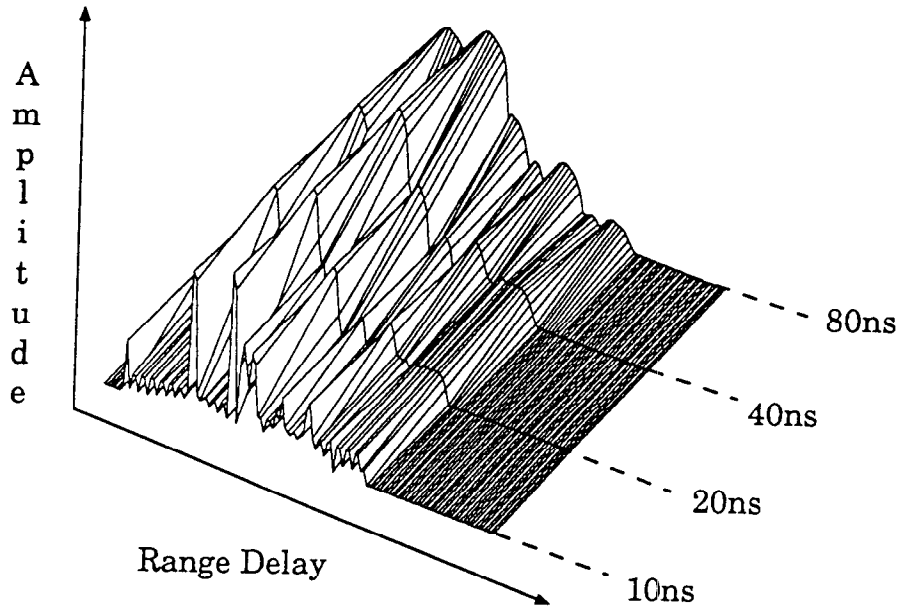

Figure 1: Variation of complex target radar return vs pulsewidth.

High range-resolution radar returns can be described as complex valued signals of finite duration. For a complete characterization of a complex target one can store the whole set of these two dimensional functions (pulses) for all possible values of radar pulsewidth δ (i.e. different resolution), aspect α and elevation ϵ. Even if one quantizes the three–dimensional space of δ, α, ϵ the required storage is enormous and impractical for real applications. Our efforts to date have concentrated on amplitude representations only. Given the amplitude of a high–range resolution radar return, several characteristics of the scatterer distribution of the target can be revealed. Varying the radar pulsewidth δ changes the resolution of the returned pulse, in the sense that more (narrow pulse) or less (wide pulse) details can be distinguished. Varying the view-point (i.e. the aspect, elevation (α, ϵ) pair) changes the shape of the returned pulse, because dominant scatterers have typically highly directive returns (in space), and because small variations in aspect produce large variations in the phase of the signal returned from each scatterer. Successful methods to provide effective compression of radar returns must address the substantial variability of the returns. As a consequence, some sort of averaging (or clustering) is necessary in representing the more meaningful, slower variation of the radar return (or the RCS) as aspect and elevation are changing. It is therefore physically meaningful to cluster the radar returns from various viewpoints into equivalence classes using a measure of similarity. The resulting quantization of

the signal space (i.e. of the radar returns) characterizes the limits of discriminating between returns from different targets using information about the viewpoint; in essence if we insist on extremely fine quantization cells we are modeling the radar sensor noise and not the underlying complex target.

Experiments with variable pulsewidths and real targets, in order to obtain a multiresolution representation of the ship are not a very practical solution. The NRL Code 5750 digital simulation model is a flexible tool for experimentation, and it has been used as the basic data generation source for the studies reported here. This model has been validated against field returns and provides high accuracy simulations. The digitally simulated ship model consists of over 800 scatterers (for each viewpoint) of a variety of types, including flat plates, point scatterers and dihedrals. These scatterers are distributed in both range and space in accordance with their actual locations on a ship. To capture safely all ship pulses we used a range gate of 128 bins corresponding to a returned signal time duration of 1280 *ns*. At the finer resolution of 10 *ns*, and sampling at the corresponding rate produces 2^7 samples.

In Figure 1 we show a typical result of a ship return with transmit pulse widths of 10 *ns*, 20 *ns*, 40 *ns*, 80 *ns*. In the three–dimensional representation shown, we see clearly the coalescence of the ship scatterers as we move from fine to coarse pulsewidths. This is also demonstrated in Figure 2, showing the pulse returns corresponding to the four different pulsewidths. In Figure 2 we also have traced, by connecting the major peaks of each return, the coalescence of the ship scatterers as we move from fine to coarse pulsewidths.

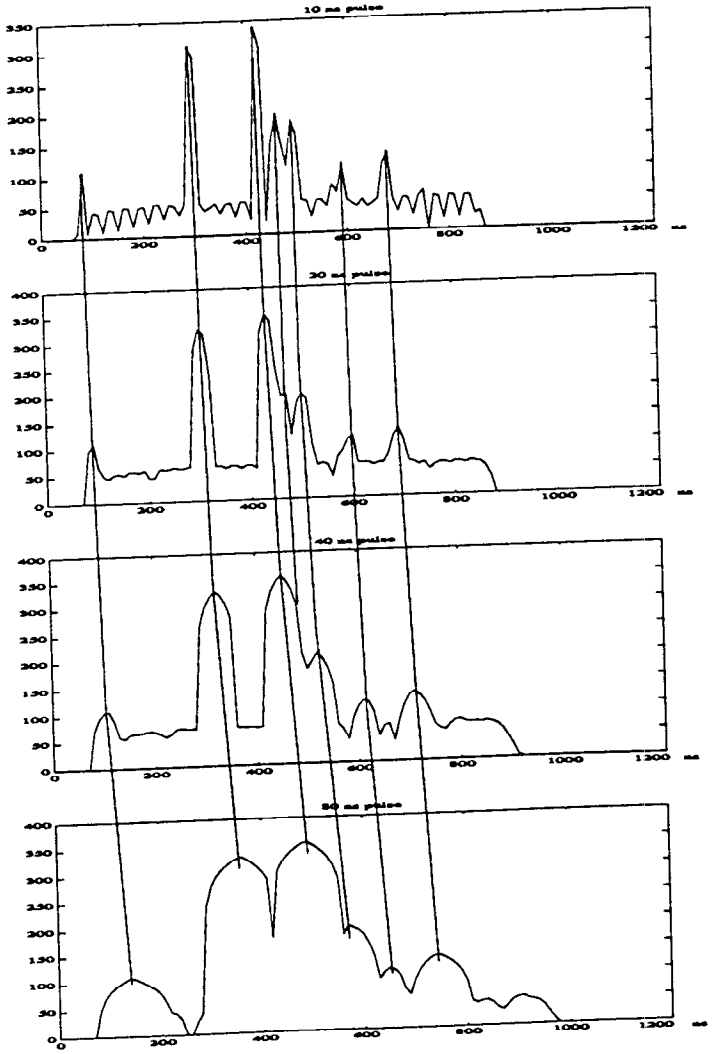

Figure 2: Individual target pulse returns from the same target model for different pulsewidths.

We call diagrams such as the one depicted in Figure 2 *scale space diagrams*, extending a notion originally introduced by Witkin in computer vision [13]. It is clear that such scale space diagrams provide a "fingerprint" of the target, since they indicate how scatterers combine; a property of the target geometry when all other scattering parameters

are kept constant. Our results todate [16, 18, 19] show that by combining wavelet representations with clustering algorithms we obtain similar multi resolution representations of the target radar return; recapturing in essence the effects of variable radar pulsewidth data.

3 SCALE SPACE ASPECT GRAPHS FOR RADAR RETURNS

A key construct in our approach to economical target model hierarchies is that of an aspect graph [17], which is a hierarchical data structure, indexed by sections of viewpoint, that stores compressed target formats. A general definition of the aspect graph [17] is that it is a graph structure in which there is a node for each *general view* of the object as seen from some maximal connected cell of *viewpoint space*, and there is an arc for each possible transition across the boundary between the cells of two neighboring general views, which is called a *visual event*. A general viewpoint is defined as one from which an infinitesimal movement in any possible direction in viewpoint space results in a view that is equivalent to the original. In contrast, a visual event is one for which there is at least one direction in which an infinitesimal movement results in a view that is different from the original. Under this definition, the aspect graph is complete in that it provides an enumeration of the fundamentally different views of an object, yet is minimal in the sense that the cells of general viewpoint are disjoint. Thus the aspect graph is equivalent to a parcellation (tesselation) of viewpoint space into general views. Considerable research has been performed in recent years on algorithms that compute the aspect graph and its related representations [17]. However, todate these conventional methods have addressed only the ideal case of perfect resolution in object shape, in the viewpoint, and the projected image, leading to the following set of practical difficulties [17]:

- A very small change in the detail of the 3-D shape of an object may drastically affect the number of visual events and nodes in the aspect graph. This is unrealistic and creates very brittle parcellations of the viewpoint space.
- A node in the aspect graph may represent a view of the object that is seen from such a small cell of viewpoint space that it is extremely unlikely to ever be witnessed. This is unrealistic also in the sense that all nodes of the aspect graph are not of equal significance. The underlying shape and size of the parcellation cell should have bearing on its importance.
- The views represented by the two neighboring nodes in the aspect graph may differ only in some small detail that is indistinguishable in a real image. In practical terms such nodes must be the same. In addition the detection of differences in views must depend on distance and noise level.

Almost exclusively, previous work on aspect graphs has focused on computer vision and object geometry [12, 17]. Our notion of the aspect graph as developed here is an extension of these concepts to sensors other than cameras such as radar. Earlier work by one of the authors has addressed these very points successfully for FLIR sensors [20].

The various algorithms that have been developed may be classified using three properties: the object domain, the view representation, and the model of viewpoint space. We use object domain and the 2-D viewing sphere in our algorithms. According to a panel discussion [17] on "Why aspect graphs are not (yet) practical for computer vision" held at the *1991 IEEE Workshop on Directions in Automated CAD-Based Vision* concluded that an important problem has been the fact that aspect graph research has not included the notion of *scale*. The results reported here and in [20] incorporate scale in the construction of the aspect graph in a manner consistent with the sensor considered. Indeed we have developed an algorithmic construction of *scale space aspect graphs*. Scale space aspect graphs are equivalent to families of viewpoint space tesselations parameterized by scale.

The approach described here is based on the physical principle that the view of the target signature remains invariant over regions of viewpoint and resolution, depending on sensor physics, environmental conditions, etc. The fact that the sensor data are noisy further reinforces the argument. In our algorithmic construction we first determine viewpoint equivalence classes according to a distortion measure and then identify the significant radar return features that remain invariant in the viewpoint tessels (see Figure 3) determined by the equivalence classes. In this *iterative process*, refinements can result, causing further subdivision of a class. Our algorithms work across multiple resolutions. Furthermore, we can compare characteristic radar returns from each equivalence class of each target to test if the resulting cluster has acceptable classification performance; this brings the characteristics of the sensor directly in the target model hierarchy. Motivated by existing terminology in computer vision [12, 17] we call the resulting viewpoint equivalence classes *aspects*. If we arrange the aspects as nodes in a graph we can connect them with links to obtain what we call the *aspect graph* of the target. The links denote the appearances of new features, or the disappearances of features relating the transitions between various aspects. This precompiled object can be used to guide the target classification process *on-line*.

We now turn to the explicit description of our algorithm. We employ wavelets and Tree Structured Vector Quantization. We refer to [2], [3], [9] for wavelet fundamentals. In such a multiresolution analysis [2] one has *two functions: the mother wavelet* ψ *and a scaling function* ϕ. We denote by f the generic radar pulse, by $\psi_{m,n} = 2^{-m/2}\psi(2^{-m}t - n)$, $\phi_{m,n}(x) = 2^{-m/2}\phi(2^{-m}x - n)$ the functions obtained by dilation and translation from ψ and ϕ. The coefficients of expanding f in terms of the $\psi_{m,n}$ are $c_{m,n}(f)$, while $a_{m,n}(f)$ are the coefficients of expanding f in terms of $\phi_{m,n}$. Usually one denotes by V_m the space spanned by the $\phi_{m,n}$. The spaces V_m describe successive approximation spaces, $\ldots V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \ldots$, each with resolution 2^m. This sequence of successive approximation spaces V_m constitutes a *multiresolution analysis* [2, 9]. W_m denotes the space which is exactly the

orthogonal complement in V_{m-1} of V_m. These concepts result in a fast algorithm for the computation of the $c_{m,n}(f)$ (or $\mathbf{c^m}$) and $a_{m,n}(f)$ (or $\mathbf{a^m}$) [2]. The whole process can also be viewed as the computation of successively coarser approximations of f, together with the "difference in information" between every two successive levels.

We generated radar return databases for various different ships, utilizing the NRL Code 5750 ship radar return simulator. In generating the synthetic data we kept the radar fixed and turned the ship, including the motion induced by sea waves. We varied the aspect angle from 0° to 360° in increments of 0.05°. This allowed for large variation in the number and appearance of dominant scatterers. Each database contained 7,200 pulses at fine resolution.

Let $\mathcal{S}$ denote the set of discretized radar pulses. The fine resolution data will be denoted by $S^0 f(n)$, $n \in I^0$, where $I^0 = \{1, 2, \cdots, 2^7\}$, is the index set of the fine resolution data. We shall let $N = 2^J$ denote the number of samples in the fine resolution data, where J is the maximum possible number of scales that we can consider. In practice one considers scales up to J^* where $J^* < J$. Respectively for each resolution m we denote by I^m the subset of I^0 where sampled values of the m^{th} resolution pulse representation $S^m f$ are computed. I^m is obtained from I^{m-1} by decimation. We used $N = 128$, and $J^* = 3$. This gives us four scales (including the given fine scale) $m = 0, 1, 2, 3$, with vector lengths $128, 64, 32, 16$ and resolutions 10 ns, 20 ns, 40 ns, 80 ns, respectively. We identify $\mathbf{a^0}$ with the vector of sampled data $S^0 f$. Then we use the *pyramid* scheme [2] to recursively compute the successive approximations $S^m f$ to the pulse f at various scales m and the residual pulses $W^m f$. As we proceed with this analysis step from scale m to the coarser scale $m+1$, the space of signals becomes smaller, and the length of vectors is halved. Thus the algorithm recursively splits the initial vector $\mathbf{a^0}$ representing the sampled pulse $S^0 f$ to its components $\mathbf{c^m}$ at different scales indexed by m representing the wavelet residuals $W^m f$; the multiresolution scheme replaces the information in each pulse $f = S^0 f$ with the set $\{W^m f,\ m = 1, 2, \cdots, J^*,\ S^{J^*} f\}$.

We construct the scale space aspect graph by developing a hierarchical, tree–structured organization of radar returns, which utilizes the multiresolution representations provided by wavelets. Vector Quantization (VQ) is primarily used as a data compression method. By properly defining a rate-distortion measure between the respective sample distributions one can reinterpret the process of vector quantization in the context of optimal decision theory. In fact, this flexibility of the definition and interpretation of rate and distortion in Shannon's theory has recently led to very beneficial cross-fertilization between these two areas, in particular between tree-structured vector quantizers [11] and classification (decision) trees [14]. VQ in addition is a clustering algorithm. Indeed the codewords, represented by the centroids, can be thought of as representatives of the equivalence class represented by each cell of the VQ (each Voronoi cell). It is in this sense that we use VQ in our approach to the problem of hierarchical representations for HRR radar returns.

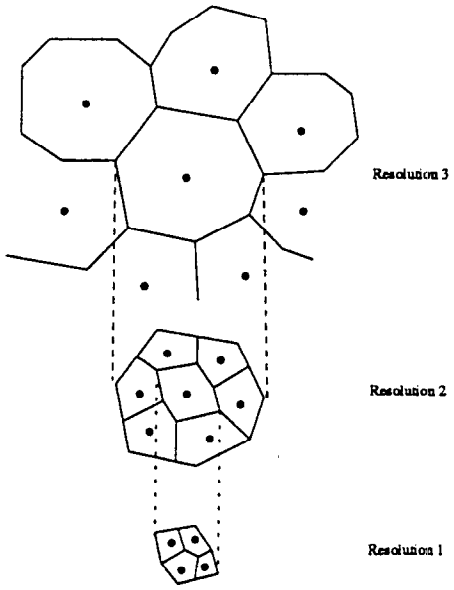

Figure 3: Illustrating a multiresolution TSVQ by splitting Voronoi cells based on different resolution data.

Implementations of the basic VQ algorithm in which the search is exhaustive are called *full-search*. In many VQ systems the time overhead associated with this search is too costly. Of particular interest is tree structured vector quantization (TSVQ) [11], which provides logarithmic (in the number of data) search time *vs* linear (in the number of data) search time provided by full search VQ. TSVQ is a special case of hierarchical VQ [11]. TSVQ is one of the most effective and widely used techniques for reducing the search complexity in VQ. A useful method for designing

the tree structure is based on the application the Linde–Buzo–Gray (LBG) algorithm [11] to successive stages using a training set. We have used a variant of this method which is of the "greedy" [15] variation. More precisely our algorithm splits the cell which contributes the largest portion of the current overall distortion. We first perform a multiresolution wavelet representation of the radar pulses, based on the selection of a mother wavelet. This allows us to consider each pulse reconstructed at different resolutions $S^0 f,\ S^1 f, \cdots,\ S^{J^*} f$. We then proceed by splitting the signal space at various resolutions in cells as indicated pictorially in Figure 3.

The data vector space (signal space) is partitioned into cells, or collections of data vectors which are determined by the repeated application of the Linde-Buzo-Gray (LBG) algorithm. LBG is first applied to the coarsest resolution representation of the data vectors $\{S^{J^*} f,\ f \in \mathcal{S}\}$. Since it is the coarsest representation, the corresponding length of the data vectors is the shortest; in our experiments that length was 16. As a result this clustering is faster than a clustering performed on the much longer fine resolution representations of the data vectors. The resultant distortion is determined based on a mean squared distance metric, and is computed using the finest resolution representation of the data vectors. The cell (equivalence class of coarse resolution representations) which is the greatest contributor to the total average distortion for the entire partition is the cell which is split in the next application of LBG. A new Voronoi vector is found near the Voronoi vector for the cell to be split and is added to the Voronoi vectors previously used for LBG. LBG is then applied to the entire population of data vectors, again using the coarsest representation of each vector. These steps are repeated until the percentage reduction in distortion for the entire population falls below a predetermined threshold. The partition in the coarsest resolution is then fixed, and further partitioning continues by splitting the cells already obtained based on finer resolution representations of the data vectors in the cell. The algorithm then iterates through these steps until the allotted number of cells have been allocated, or until total average distortion has been reduced to a requisite level. Each new layer in the tree corresponds exactly to partitions based on the next finer resolution representation of the data.

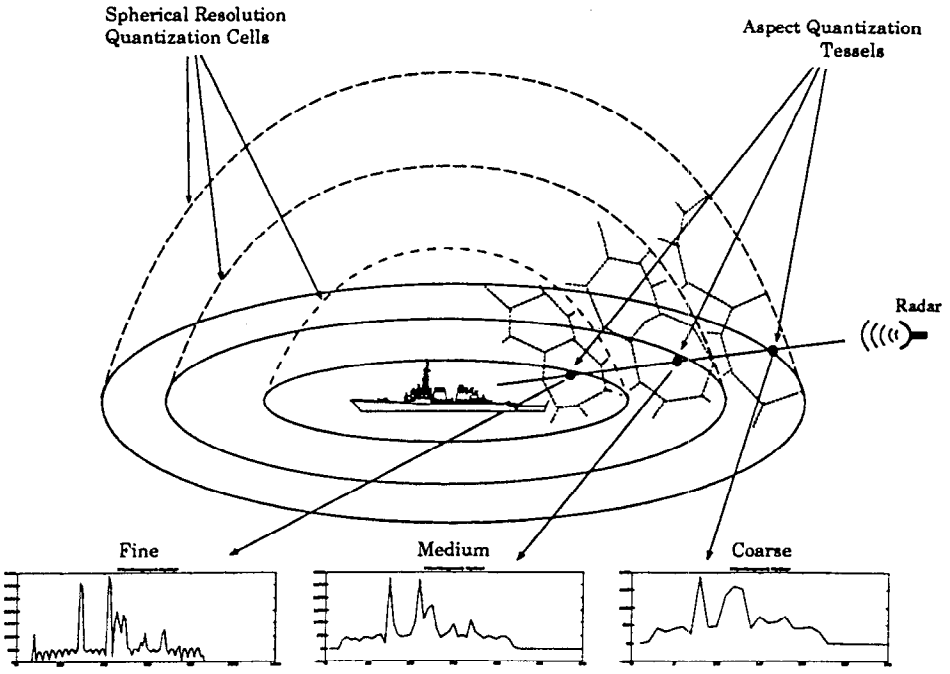

Figure 4: Illustrating a multiresolution aspect graph for radar ship data.

The algorithm constructs a hierarchical organization of the radar return data as a tree, which is conformant with the wavelet multiresolution data representations. This representation can be constructed for a single target or for a collection of targets. In the former case this construction produces a "model" for the target as viewed by the radar sensor. In the later this construction organizes an entire database of target radar returns. The resulting tree is our *aspect graph of the target(s)*. As illustrated in Figure 4 a multiresolution (or scale space) aspect graph for radar ship data results naturally from our construction. Here the concentric spheres designate different resolutions (scales). The cells on these spheres illustrate aspect equivalence classes for the radar signals (returned pulses). These equivalence classes mean that the pulses in these clusters are difficult to discriminate due to their similarity. As we move inwards in this graph, the outside cells split as we can now get further characteristics of the target based on finer resolution information on the pulse. These characteristics are related to dominant scattering centers. To construct the aspect graph we select as the representative from each equivalence class, the radar pulse corresponding to the centroid of

the corresponding Voronoi cell. The resulting graph has geometrically the appearance of a tree; a typical one is depicted in Figure 5. The nodes (cells in these figures) correspond to aspect–elevation neighborhoods for which the corresponding returned pulses are too similar to be separated. The nodes are given for various resolutions as well as the percentages of pulses that were clustered in each cell. It is clear from this discussion that the aspect graph is a

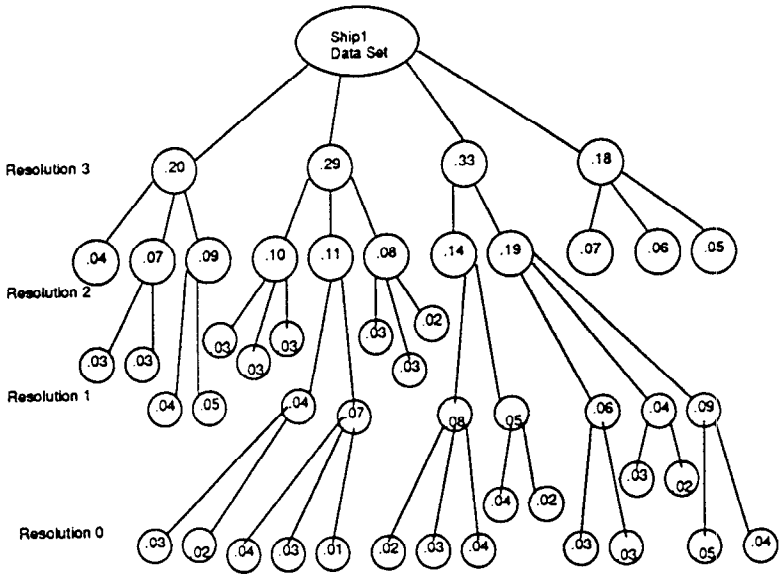

Figure 5: Apect graph (as a tree) for a typical ship radar data set

reduced but accurate model of the target and can be used to guide the ATR process in model–based ATR. In such an application the received pulse is compared with the "canonical" pulse at each node sequentially as the ATR process evolves. The aspect graph directs the search in an efficient and speedy manner; it is well known that tree based search is logarithmic in the number of terminal nodes, which is a substantial reduction from conventional methods.

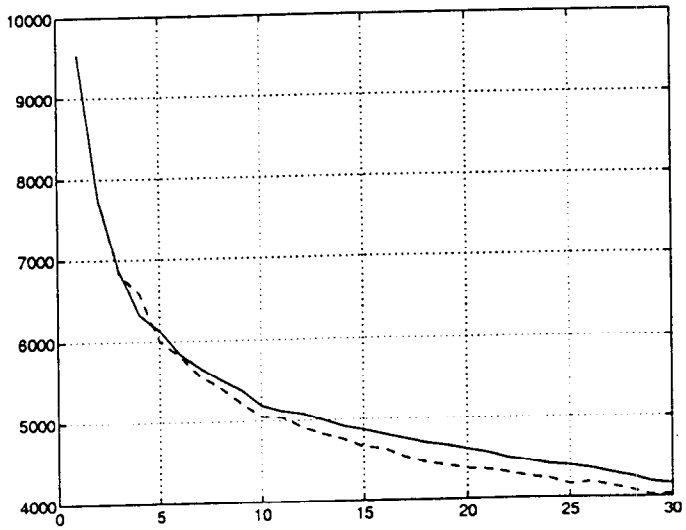

Figure 6: Comparison between full-search VQ on fine scale pulses vs wavelet-TSVQ on multiresolution pulses

It is clear that the search of the hierarchically organized database (as a tree) will be much faster than the case by case search of the overall database. The question is: how much performance did we sacrifice? To answer this

question we compared the results of the wavelet-TSVQ algorithm with those of full-search VQ applied to the finest resolution data, by means of the total (operational) distortion *vs* number of cells (i.e. the number of terminal nodes) performance curve. In all our experiments the two curves were very close showing that the performance of the wavelet-TSVQ algorithm is indeed excellent. Additional results can be found in [16]. This performance of our algorithm shows that we are getting high performance in an efficient way. A typical result is depicted in Figure 6. The solid curve corresponds to the wavelet TSVQ algorithm, while the dashed curve corresponds to full-search VQ. As shown in [19] using "water fall" diagrams it is actually a quite good approximation.

As explained in detail in [18, 20] the multiresolution aspect aspect graph constructed by our wavelet-TSVQ method provides an extremely efficient hypothesis generation and indexing mechanism to guide the on-line classification process. Our method constructs these efficient indexing schemes algorithmically, without resorting to *ad hoc* heuristics typical of previous work [23, 24, 25, 26]. Our method can be further extended to provide extremely efficient indexing methods of local features or groups of features, which can be thought of as *fingerprints* of the targets. The methodology described here has distinct advantages over other indexing schemes such as the *Geometric Hashing* technique of Lamdan *et al* [22], or the methods of Knoll and Jain [21]. Our techniques hold great promise for ATR algorithms involving large libraries of target models, without commensurate increase in computational complexity. The key idea we are exploring with our aspect graphs is the use of clusters of features, represented in terms of geometric properties that are invariant under projection, as keys for indexing into a hashed library.

4 ANALYTICAL FRAMEWORK: COMBINING COMPRESSION AND CLASSIFICATION

Vector quantization (VQ) has been traditionally used, by the majority of the practitioners as a compression algorithm [11]. VQ is a common method of lossy compression that applies statistical techniques to optimize distortion/bit rate tradeoffs. However, as we have already pointed out, VQ can be also used as a classifier quite successfully [20, 16]. Therefore VQ can be used in a combined mode to perform classification efficiently utilizing compressed data. This key idea provides an efficient analytical framework for the design and analysis of fast progressive classification algorithms. There are various methods to approach this promising idea. The one described here, combines some additional advantageous requirements. First, careful design of the overall scheme can emphasize local classifications involving only small regions of the signal. Second, these local classifications can be combined in a hierarchical fashion that reflects the signal model, the sensor performance and sensitivities, and progressive increase in the confidence of the classification.

In such an approach, which fundamentally combines data compression and classification, one needs to develop efficient adjustment methods to weight the relative importance of the two aspects of the algoritm. Indeed, while efficient compression can reduce significantly the complexity of classification by bringing forward essential local characteristics of the signal, excessive compression may throw away valuable information and thus reduce the accuracy of classification. The design of this tradeoff is the most difficult and less unexplored, albeit most promising, part of our approach.

Our approach applies equally well to one-dimensional (radar pulses, acoustic signals, etc.) or multi-dimensional signals (FLIR, ISAR, SAR, LADAR, etc.). By appropriate indexing we can always consider a given signal (one-dimensional or multi-dimensional) as a vector. VQ operates on subblocks or subvectors of the signal or, in more sophisticated schemes, on subblocks or vectors of a feature vector computed from the given signal by some transformation. These subblocks or subvectors of the vector, on which VQ is to operate, correspond to local information from the signal or local features of the signal. For each subvector, the VQ encoder determines the nearest codeword (which is also a vector of the same dimension) and outputs the chosen codeword's index. When VQ operates in the data compression mode the sequence of indices so generated can be stored and then transmitted. The VQ decoder reverses this operation: it receives as inputs the indices and outputs the appropriate codewords by simple table lookup. One can easily realize that the computational complexity of the encoding and decoding part of the VQ is asymmetrical. The decoder is very simple. The attempt of combining compression and classification stems from the idea that the centroids of the VQ cells (which were called codewords in the compression scheme) are prime candidates for most typical representatives of the subvectors belonging in the same cell. It is likely that subvectors that are assigned by the VQ algorithm to the same cell belong to the same class. Therefore one can assign cells to classes and obtain classification. An efficient way of doing this is represented in the Learning Vector Quantization (LVQ) algorithm of Kohonen [5, 27], which we have used and analyzed extensively before [27, 20]. This assignment of cells to classes can be best understood and analyzed as a method for partitioning the feature space into decision regions corresponding to each class. Indeed the boundaries of the cells approximate the Bayes decision surfaces for the classification problem at hand, if this assignment of cells to classes has been performed efficiently.

Our objective is to design algorithms that combine compression and classification and show that they result in high performance ATR algorithms. Towards this end we discuss next the various performance measures of such an algorithm based on VQ. Rate and Distortion characterize the performance of VQ from the point of view of compression. The performance of the classifier is measured by Bayes risk, which may include in general different costs for different types of errors. These performance measures are competing and there are various ways of approaching this *multi-objective* design problem. One is to combine the objectives (performance measures) by incorporating the Bayes risk in the distortion measure minimized by the design algorithm. Another is to treat the problem as a multi-objective optimization problem, where the design algorithm optimizes one performance measure while satisfying constraints on the other objectives. We plan to investigate both approaches and compare the results on

ATR problems with real data.

Here we consider the first formulation of the problem where we incorporate a Bayes risk into the average distortion measure minimized by the design algorithm. Such an approach introduces additional complexity which however occurs only in the design phase of the overall algorithm. The resulting algorithm has complexity equivalent to that of an ordinary VQ algorithm. In addition the combined classification and compression scheme requires no more bits to describe than the bits required for compression alone, which implies that there is no apparent memory overload.

Let $\mathcal{F}$ denote the K-dimensional vector space of features. We are given N feature vectors $\{\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_N\}$. Each feature vector $\mathbf{f}= (f_1, f_2, \cdots, f_K)^T$ is mapped by a full search VQ onto a codeword (or centroid) $Q(\mathbf{f}) = \theta_i$, from a set of centroids $\{\theta_1, \theta_2, \cdots, \theta_M\}$, where M is the number of cells in the tesselation of $\mathcal{F}$ constructed by the VQ. This tesselation, or partition, induced by Q is denoted by $\mathcal{P}^{\mathcal{F}} = \{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_M\}$, where $\mathcal{C}_m$ denotes the generic cell in the tesselation. Let γ, δ denote the encoder, resp. the decoder of the VQ implemented in the overall algorithm. That is for each feature vector $\mathbf{f}_i \in \mathcal{F}$, if $Q(\mathbf{f}_n) = \theta_m$, then $\gamma(\mathbf{f}_n) = m$, and $\delta(m) = \theta_m$. Then $Q(\mathbf{f}_n) = \delta(\gamma(\mathbf{f}_n))$. Since $M << N$, the VQ Q compresses the data. Next suppose that we have L classes (or hypotheses) $\{H_1, H_2, \cdots, H_L\}$. As explained above, to perform classification we assign a class H_l to each cell $\mathcal{C}_m$. This is the same as assigning a class label $l = 1, 2, \cdots, L$, to each cell label $m = 1, 2, \cdots, M$ in the partition of $\mathcal{F}$ induced by the VQ. This last assignment is the decision or classification rule d.

In such a scheme the signal vector $\mathbf{x}$ is first transformed by a preprocessor (which in our case is the wavelet transform (WT)) into a transformed vector $\mathbf{w}$, so that $\mathbf{w}=\mathbf{T}\mathbf{x}$, where $\mathbf{T}$ in our case denotes the WT. Then a feature selection map $\mathbf{F}$ is applied to bring the transformed vector $\mathbf{w}$ into $\mathbf{f}\in \mathcal{F}$, so that $\mathbf{f}=\mathbf{F}\mathbf{w}$. The design of the combined compression and classification algorithm is precisely the design of the encoder, decoder and decision rules γ, δ, d. Equivalently this design is the construction of a tesselation $\{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_M\}$ of $\mathcal{F}$ with centroids (or codewords) $\{\theta_1, \theta_2, \cdots, \theta_M\}$, and a decision rule d. As in LVQ this design can be accomplished by constructing the tesselation and the decision rule by an iterative process working with a training (or learning) set of features $\mathcal{L} = \{\mathbf{f}_1^L, \mathbf{f}_2^L, \cdots, \mathbf{f}_{N'}^L\}$, where N' is the number of training vectors available overall.

Given a decoder-encoder pair γ, δ, we associate the average distortion

$$D(\gamma, \delta) = E[\rho(\mathbf{f}, \delta(\gamma(\mathbf{f})))] \geq E[\min_{1 \geq n \geq N} \rho(\mathbf{f}, \delta(n))]. \tag{1}$$

Here ρ is the error (distortion) or distance function used in the VQ. Most of the work todate has used a quadratic function $\rho(\mathbf{f}, \delta(\gamma(\mathbf{f}))) = \| \mathbf{f}-\delta(\gamma(\mathbf{f}))\|^2$, as the distortion measure, basically for its mathematical tractability. Weighted quadratic functions have also been used [11], notably in speech coding. The selection and design of appropriate distance functions is very much in need of further research.

Given a classification rule d, the classification performance of the overall scheme can be measured by the Bayes risk

$$J_B(\gamma, d) = \sum_{i=1}^{L} \sum_{j=1}^{L} P(d(\gamma(\mathbf{f})) = H_j | \mathbf{f} \in H_i) P(H_i) C_{ij}, \tag{2}$$

where C_{ij} is the relative cost assigned to the decision that $d(\gamma(\mathbf{f})) = H_j$, while the feature vector $\mathbf{f}$ comes from class H_i. The most commonly encountered case is that where $C_{ij} = 0$. An important observation is that the encoder δ does not affect the Bayes risk J_B. Let $I_{H_i}(H_j)$ be the indicator function taking the value 1 if $H_i = H_j$ and the value 0 otherwise. We can then rewrite the Bayes risk as follows:

$$J_B(\gamma, d) = \sum_{k=1}^{M} \sum_{j=1}^{L} I_{H_j(d(k))} \sum_{i=1}^{L} P(\gamma(f) = k | f \in H_i) P(H_i) C_{ij}, \tag{3}$$

To implement the key idea of the aproach (i.e. to simultaneously consider compression and classification of the data) we use a Lagrangian formalism, which combines ordinary distortion, compression rate R (which is typically used as a constraint), and classification error:

$$J_\lambda(\gamma, \delta, d) = D(\gamma, \delta) + \lambda_R R(\gamma, \delta) + \lambda_B J_B(\gamma, d) \tag{4}$$

The resulting modified distortion measure is used to determine the partitioning γ of the training vectors by mapping each vector to the codeword/class label producing the minimum distortion J_λ. This combination of the performance measures provides for flexible trade-off between the compression and classification requirements: indeed $\lambda_B \to 0$ corresponds to regular VQ, while $\lambda_B \to \infty$ corresponds to Bayes classification.

The design of the algorithm in this formulation proceeds by employing a descent algorithm to minimize $J_\lambda(\gamma, \delta, d)$ by alternatively improving γ, δ and d. We start with an initial $(\gamma^{(0)}, \delta^{(0)}, d^{(0)})$ and iteratively apply an improvement transformation $(\gamma^{(t+1)}, \delta^{(t+1)}, d^{(t+1)}) = T(\gamma^{(t)}, \delta^{(t)}, d^{(t)})$ so that $J_\lambda(\gamma^{(t)}, \delta^{(t)}, d^{(t)})$ is nonincreasing in t. Since J_λ is bounded below by 0, it follows that such a scheme will converge as $t \to \infty$. This observation allows to implement simple stoping rules for the iterative algorithm. The most common such rule will be to select a value of λ and iterate until the value of J_λ falls below a certain level.

Thus the iterative transformation T is implemented by the following three successive steps:

Step 1 Choose $d^{(t+1)}$ to minimize $J_\lambda(\gamma^{(t)}, \delta^{(t)}, d^{(t+1)})$.

Step 2 Choose $\delta^{(t+1)}$ to minimize $J_\lambda(\gamma^{(t)}, \delta^{(t+1)}, d^{(t+1)})$.

Step 3 Choose $\gamma^{(t+1)}$ to minimize $J_\lambda(\gamma^{(t+1)}, \delta^{(t+1)}, d^{(t+1)})$.

The iterations continue until the desired stoping level for J_λ is met.

The interpretation of the steps is straightforward. Given a partition of $\mathcal{F}$ and a set of centroids, Step 1 minimizes the Bayes risk associated with the centroid labels. As can be seen from (3), this minimization depends only on the partition represented by γ; the codeword values given by δ do not affect the minimization.

Step 2 minimizes J_λ over the codeword values δ given the partitioning γ and the labeling d. Since the codeword values do not affect the Bayes risk, J_λ is minimized when the codewords are chosen as centroids based on the distortion measure given in (2).

Finally Step 3 determines the partitioning γ that minimizes J_λ given the centroids δ and labels d.

The modified distortion measure of the training sequence is given by

$$J_\lambda = \sum_{\mathbf{f}\in L} P(\mathbf{f})\{\rho(\mathbf{f}, \delta(\gamma(\mathbf{f})) + +\lambda_R R(\gamma(\mathbf{f}), \delta(\mathbf{f})) + \lambda_B \sum_{i=1}^{L}\sum_{j=1}^{L} I_{H_j}(d(\gamma(\mathbf{f})))\frac{P(\mathbf{f}|H_i)}{P(\mathbf{f})}P(H_i)C_{ij}\}, \tag{5}$$

To implement this scheme we need values for the probabilities appearing in (5); and this is a disadvantage if they are estimated using the vectors from the training set. We describe below how we resolve this difficulty.

The above procedure can be extended to a Tree-Structured VQ (TSVQ) [11]. The basic idea for this extension is quite straightforward. VQ employing full search is computationally costly in both codebook generation and encoding. On the other hand tree-structured VQ (TSVQ) can signicantly speed up the design with often neglizible decrease in performance. An additional advantage of TSVQ is that it uses variable rate, so that different numbers of bits are required depending on the path that the encoder takes through the tree. The extension then is achieved by combining the above algorithm with well known tree growing and pruning techniques [15]. A TSVQ tree is grown by successively splitting nodes, until the desired rate is reached. By modifying the splitting criterion and codeword design methods, the TSVQ can be grown to provide both classification and compression. Our wavelet–TSVQ algorithm for constructing aspect graphs (as described in section 3 above) implements Steps 2 and 3 in a fast but accurate suboptimal way.

Regarding Step 1, we use LVQ. Kohonen's stated goal in LVQ [5] was to imitate a Bayes classifier with less complexity than other neural network approaches, but there is no explicit minimization of Bayes risk in the code design. However, in [27, 20] we showed that indeed the way LVQ moves around the centroids during learning, asymptotically approximates the effect of optimizing Bayes risk. The argument is therefore that implementing LVQ in the above algorithm, essentially replaces the explicit optimization of the average Bayes risk as is formulated in (3). This can be established analytically by employing a combination of the methods used in [16] and [27]. This method has the additional advantage of not needing estimates of empirical distributions from the training set. The result is a much more efficient algorithm. It is important to emphasize that the overall approach is non-parametric, in the sense that probability distributions for the signal, the transformed signal, and the feature vector are not needed. Instead the approach can be interpreted as using the training set to learn the empirical distributions of the various vectors and use them as if they were true, very much like the interpretation we have given to the LVQ algorithm [27, 20]. Finally, we would like to mention that the mother wavelet is an additional parameter on which the modified distortion measure J_λ depends. An additional optimization iteration can be added to address this optimization. Progress in this direction which employes parametrizations of wavelets will be reported elsewhere.

References

[1] I. Daubechies, "Orthonormal Bases of Compactly Supported Wavelets", *Comm. Pure Appl. Math. 41*, pp. 909—996, 1988.

[2] S.G. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", *IEEE Trans. on Pattern Analysis and Mach. Intel.*, Vol. 11, No. 7, pp. 674—693, 1989.

[3] Y. Meyer, *Ondelettes et Operateurs I, II*, Hermann, Paris, 1990.

[4] J.G. Daugman, "Complete Discrete 2-D Gabor Transforms by Neural Networks for Image Analysis and Compression", *IEEE Trans. on Ac. Sp. and Sign. Process.*, Vol 36, No. 7, pp. 1169—1179, July 1988.

[5] Teuvo Kohonen, *Self-organization and Associative Memory*, Springer-Verlag, New York, 1989.

[6] S.G. Mallat and S. Zhong, " Characterization of Signals From Multiscale Edges", *NYU, Computer Science Technical Report, Nov. 1991*; also in *IEEE Trans. Pattern Anal. Machine Intell*, Vol. 14, No. 7, pp. 710-732, July 1992.

[7] S. Zhong and S.G. Mallat, "Compact Image Representation From Multiscale Edges", *Proc. of 3rd International Conf. on Comp. Vision*, Dec. 1990.

[8] O. Rioul and M. Vetterli, "Wavelets and Signal Processing", *IEEE Signal Processing Magazine*, Vol. 8, No.4, pp.14—38, October 1991.

[9] I. Daubechies, *Ten Lectures on Wavelets*, SIAM, 1992.

[10] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, "Image Coding Using Vector Quantization in the Wavelet Transform Domain", *Proc. 1990 IEEE ICASSP*, Albuquerque USA, pp. 2297—2300, April 1990.

[11] A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Press, 1991.

[12] K. Bowyer, D. Eggert, J. Stewman, and L. Stark, "Developing the Aspect Graph Representation for Use in Image Understanding", *Proc. DARPA Image Understanding Workshop 1989*, pp.831—849.

[13] A. Witkin, "Scale Space Filtering", *Proc. Int. Joint Conf. Artificial Intell."*, 1983.

[14] L. Breiman, J.H. Friedman, R.A. Olshen and C.J. Stone, *Classification and Regression Trees*, Wadsworth and Brooks, 1984.

[15] E. Riskin, "A Greedy Tree Growing Algorithm for the Design of Variable rate Vector Quantizers", *IEEE Trans. on Signal Process.*, Vol. 39, No. 11, pp. 2500–2507, Nov. 1991.

[16] J.S. Baras and S.I. Wolk, "Hierarchical Wavelet Representations of Ship Radar Returns", submitted for publication to *IEEE Trans. on Signal Process.*, 1993; also *NRL Technical Report* NRL/FR/5755–93-9593.

[17] D.W. Eggert, K.V. Bowyer, C.R. Dyer, H.I. Christensen and D.B. Goldgof, "The Scale Space Aspect Graph", *IEEE Trans. on Pattern Anal. and Mach. Intel.*, Vol. 15, No. 11, pp. 1114-1130, Nov. 1993.

[18] J.S. Baras and J.L. Preston, "Wavelet and Learning Clustering Algorithms for Automatic Target Recognition," AIMS Technical report AIMS-TR-93-04, Final progress report on DARPA SBIR Phase I Contract 92-61, September 1993.

[19] J.S. Baras and S.I. Wolk, "Efficient Organization of Large Ship Radar Databases Using Wavelets and Structured Vector Qantization", *Proc. of the 27th Annual Asilomar Conference on Signals, Systems, and Computers*, Vol 1, pp 491-498, November 1-3, 1993, Pacific Grove, California.

[20] J.S. Baras and D.C. MacEnany, "Model-Based ATR: Algorithms Based on Reduced Target Models, Learning and Probing," *Proceedings of the Second ATR Systems and Technology Conference*, Feb. 1992, Vol. 1, pp. 277-300.

[21] T.F. Knoll and R.C. Jain, " Recognizing Partially Visible Objects Using Feature Indexed Hypotheses", *IEEE J. Robotics and Automation"*, 2(1), 1986, pp. 3-13.

[22] Y. Lamdan and H.J. Wolfson, "Geometric Hashing: A General and Efficient Model-Based Recognition Scheme," *Proc. Second Int. Conf. Comp. Vision*, 1988, pp. 238-249.

[23] K. Ikeuchi and T. Kanade, "Automatic Generation of Object Recognition Programs", *Proc. IEEE*, Vol. 76, No. 8, Aug. 1988, pp. 1016- 1035.

[24] D.E. Dudgeon, J.G. Verly, and R.L. Delanoy, "An Experimental Target Recognition System for Laser Radar Imagery", *Proc. DARPA Image Understanding Workshop 1989*, pp. 479-506.

[25] J.G. Verly and R.L. Delanoy, "Appearance-Model-Based Representation and Matching of 3-D Objects", Proc. of IEEE Computer Society Workshop on Computer Vision, 1990, pp. 248-256.

[26] M. Seibert and A. Waxman, "Adaptive 3-D Object Recognition from Multiple Views", *IEEE Trans. on Pattern Match. and Mach. Intell.*, Vol. 14, 2, pp. 107-124, Feb. 1992.

[27] J.S. Baras and A. LaVigna, "Convergence of a Neural Network Classifier," *Proc of 29th IEEE Conf on Dec. and Control*, December 1990, pp. 1735–1740.