ATR complexity and template set size

Eric Freudenthal[a], Eugene Lavely[b], William Pierson[c], Mariam "Ali" Argyle[a] and Joshua Fishman[a]

[a] Courant Institute, New York University, 715 Broadway 10th Floor, New York, NY 10003
[b] ALPHATECH, Inc., 50 Mall Road, Burlington, MA 01803
[c]AFRL/SNAT, 2241 Avionics Circle, Wright-Patterson AFB, OH 45433-7321

ABSTRACT

We investigate the complexity of template-based ATR algorithms using SAR imagery as an example. Performance measures (such as P_{id}) of such algorithms typically improve with increasing number of stored reference templates. This presumes, of course, that the *training* templates contain adequate statistical sampling of the range of observed or *test* templates. The tradeoff of improved performance is that computational complexity and the expense of algorithm development training template generation (synthetic and/or experimental) increases as well. Therefore, for practical implementations it is useful to characterize ATR problem complexity and to identify strategies to mitigate it. We adopt for this problem a complexity metric defined simply as the size of the minimal subset of stored templates drawn from an available training population that yields a specified P_{id}. Straightforward enumeration and testing of all possible template sets leads to a combinatorial explosion. Here we consider template selection strategies that are far more practical and apply these to a template-based SAR target identification problem. Our database of training templates consists of targets viewed at 3-degree increments in pose (azimuth). The template selection methods we investigate include uniform sampling, sequential forward search (also known as greedy selection), and adaptive floating search. The numerical results demonstrate that the complexity metric increases with intrinsic problem difficulty, and that template sets selected usingour greedy algorithmsignificantly outperform uniformly sampled template sets of the same size. The adaptive method, which is far more computationally expensive, selects template sets that outperform those selected by the greedy technique, but the small reduction in template set size was not significant for the specific examples considered here.

Keywords: SAR, ATR, template selection, adaptive floating search, complexity

INTRODUCTION

ATR algorithm complexity is a difficult theoretical problem and has significant practical importance as well. Characterization of complexity using either theoretical or empirical means is useful for a number of purposes including *i*) assessing the operational utility of a given data class and algorithm as a function of the *a priori* data and computational resources available, *ii*) predicting performance of ATR algorithms for a given mission description and specified statistical distributions of the targets and the data properties, *iii*) providing insight into approaches for complexity mitigation via new algorithm and/or sensor design, and improved sensor deployment strategy, and *iv*) providing improved cost-reduction strategies for experimental collection and/or synthetic generation of *training samples* (e.g., SAR chips) while maintaining acceptable performance levels. These application areas clearly have important tactical, operational, and economic ramifications. Despite the obvious benefits of having one, a first-principles ATR complexity theory is not yet available. In fact, there is not even an agreed upon definition in the ATR community of ATR complexity. This reflects not only the relative infancy of this subject but the fact that complexity is a diverse and multifaceted subject as indicated, for example, by the contributions in the journal *Complexity*[1]

For the purposes of this paper, we identify three definitions of complexity that we believe are of direct practical relevance to the ATR problem, and that capture commonly discussed notions of complexity. These are *representation complexity*, *search complexity*, and *data distribution complexity*. For the ATR context we define **representation complexity** as the size of a *target representation library* that must be exploited to solve the problem to a pre-specified level of performance. In this way representation complexity is closely related to Kolmogorov complexity (e.g., Cover and Thomas, 1991). This library is a set of target descriptors or models of the signal utilized in the recognition process. For a template-based algorithm of the library members would include image and/or feature templates. For a model-based algorithm the library would include model

[1] See http://journals.wiley.com/complexity/

representers e.g. a *Computer Aided Design* model of the targets of a given fidelity (resolution). For a feature-based or trained non-parametric algorithm, the library would include the number of trained parameter vectors and/or their dimensions. Empirical insight into this class of complexity may be gained by evaluating the performance of classification algorithms as applied to example problems using a variety of match metrics and a variety of reference template sets as their input. Meaningful complexity characterization must rely on the optimality or near-optimality of these input template sets. Implementation of this empirical approach is the major thrust of this paper. The second major class of complexity that we may identify is **search complexity** which we define as the number of operations (or the computational cost) required to correctly identify the target type with a desired probability of correct classification. This type of complexity has strong analogs with optimization theory and inverse theory in which the objective is to discover the set of parameters that opitmally maximize (or minimize) an objective criterion (see *e.g.*, Mosegaard and Tarantola, 1995; Wissinger *et al*., 1996; Sambridge, 1998). For example, we may wish to maximize a match metric that depends on a large number of parameters including target identity, target pose, target configuration, target context. This is an optimization problem in a multi-dimensional parameter space. In this case, the convergence points of local search methods may depend on the initial guess, and the number of operations required for global search methods cannot generally be computed *a priori*. Also, elaborate multiple hypothesis testing strategies are strongly problem dependent and behavioral models are not generally available. Despite these obstacles, one may posit reasonable measures or indicators of search complexity. For example, Ryan *et al*., (2001) discussed the relation of distance measures and search complexity. Another approach is to quantify the topographic complexity of the multi-dimensional match surface (e.g. the number and distribution of basins, hills, and saddle points in the surface). The third definition of complexity that we may identify is **data distribution complexity.** We define this as the effect of noise on both representation and search complexity. Therefore, in some sense, it could be regarded as a corollary to representation and search complexity. However, the effects of noise on ATR problems and on inference problems in general can be profound, and statistical approaches to address the effect of noise on these problems represent an extremely important literature (e.g. Tarantola, 1987). For this reason, we separately categorize this type of complexity as above.

In order to illustrate key notions in representation and data distribution complexity, we consider the scenario shown in Figures 1a-b. Figure 1a represents a simple (but canonical) ATR problem in which it is assumed the template database contains only two target classes (Target 1 and Target 2). The data or data features represented in a template for any given ATR is at the discretion of the designer. For example, the template depicted in this figure could be interpreted as the spatial location (e.g. range and cross-range) of the brightest peak in a SAR image as the pose angle θ (*e.g*., target azimuth) varies. The solid line (Target 1) and dashed line (Target 2) depict the theoretical trajectories $\boldsymbol{f_1}(\theta)$ and $\boldsymbol{f_2}(\theta)$ of the feature location for the two targets. In general, feature representations within an ATR data library may be derived from theory (numerical predictions) or from actual observations collected during the training phase. The peak locations derived from observed test scenes for either of the target types will generally not agree precisely with any given representation in the library, and there are a number of reasons for this. First, the extracted feature may depend on the details of the feature extraction algorithm and its detailed sensitivity to noise. Second, for theoretically derived library templates, the predictions typically utilize an idealized model representation for the target that inevitably will differ from the actual target. In addition, the physics-based modeling may not capture all of the relevant physics of the scattering process and SAR image formation. Third, there may be errors or uncertainty in the experimental conditions (e.g. depression, aspect, etc.). Fourth, the pose of the observed target is not likely to match precisely the pose angles sampled in the library. Fifth, real images always display a finite signal to noise ratio. Finally, variable clutter and texture backgrounds in which the target is embedded may, in turn, lead to feature variability. The feature variability induced by these types of effects is depicted in Figure 1a by the distribution of experimentally derived feature values. Figure 1b emphasizes this effect by depicting a band of possible observed feature values that arise due to these *noise-like* effects. Library templates are indicated by the open squares (Target 1) and open circles (Target 2) in the figure. In this case the template values are randomly scattered about the theoretical prediction. These features constitute the stored template database to be used as the reference values with known pose for the ATR problem. The filled circles in Figure 1a depict Target 1 data observations and the filled triangles indicate Target 2 data observations. Of course, in practice, the actual identity of the target generating these observations will not be known *a priori* (*i.e*., they will not be labeled as shown here). The star-shaped markers indicate features that may be observed in a test image but generated by a target (or clutter) not included in the reference database.

There are several additional challenges to the ATR problem that are indicated by Figure 1. These include: (1) determination of the minimum number of templates required for a specified performance level (measured, for example, with P_{id}), (2) determination of the optimal distribution of these templates with pose, (3) sensitivity of the template selection to variance in the representation database as well as to variance in the new data observations, (4) development of a robust criterion to reject observed template features as out of class rather than association to nearest neighbor in reference database, and (5) quantification of the difficulty of the identification problem as a function of the pose and the targets in the mission description. Regarding the latter point, and for the present example, problem difficulty apparently maximizes near the intersection of the two curves in Figure 1a.

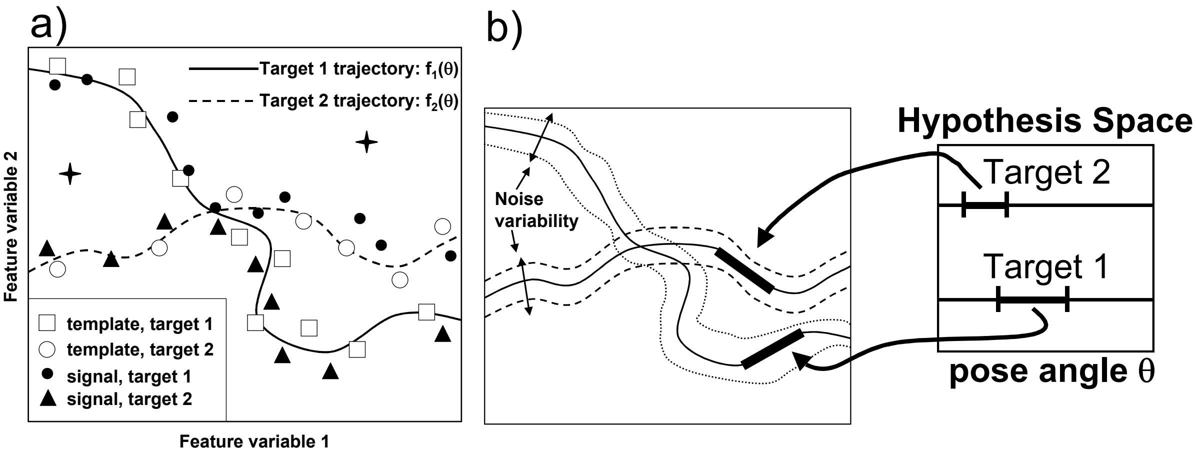

Figure 1: *a) Depiction of feature values that compose a hypothetical template (e.g. range and cross-range of the brightest peak in a SAR image). Trajectory of feature values shown for total range in a single pose variable (e.g. azimuth). b) There are numerous sources of signal variability so that in general there will not be a precise correspondence between observed templates and templates stored in a reference library; a hypothetical signal variability band is shown. Also shown is mapping of hypothesis space (in pose and target identity) to feature variable space.*

Because these problems are rather abstract, our approach is to attempt to gain insight by studying various example applications. Thus, in this paper, we consider a sequence of template-based, three-class target identification problems. Here, our data library consists of a set of templates, and our research focus is determination of the number and distribution of templates required to achieve a specified performance level using various template selection algorithms. We specifically consider template-based ATRs that search an entire template library for closest matches. In this process we apply a fixed scheme for comparing test inputs to templates within the library using various match metrics that compare extracted features to stored features within the database. Our feature-based matcher represents the salient information of a target's SAR signature in terms of the spatial topography of the image's bright pixels. The peaks are local amplitude maximums in both the range and cross-range directions. A detailed description of the peaks matching metric can be found in Ettinger *et al*., (1996). We also utilize a grayscale match metric (see Irving and Ettinger, 1999)

Our goal is to compute minimal template sets that yield a specified P_{id}. Before proceeding it is useful to consider the computational challenge of testing the various possible template combinations. Assume that there are m templates available in the full template set, and that we seek the best subset of n templates for $1 < n < m$. For each combination we compute a separability criterion or performance measure such as probability of identification (P_{id}) and rank-order the performance of each set as the results accumulate. The total number of possible subsets increases with increasing m for fixed n and is given by the well-known combinatoric result for exhaustive search in Table 1. Table 1 includes two well-known greedy techniques, sequential forward selection (SFS) and sequential backward selection (SBS). In the SBS technique one starts with m templates and at each step removes the single template that results in the smallest performance degradation. This process is continued until n templates remain. This method is sub-optimal since there is no guarantee that the template set of dimension i must originate from the previous set in this sequence (with dimension i+1). Nonetheless, the number of combinations that must be tested are far fewer than in the exhaustive search case. The SFS technique is nearly the reverse in procedure. The first template set is dimension unity and is given by the template that returns the highest-value criterion. The optimal two-member template set contains the first selection plus a second template that maximizes the criterion, and so on.

Both the SFS and SBS methods suffer from the nesting effect. In SFS, once a template is selected, there is no way to remove it. In SBS, once a template is rejected, there is no way to reinsert it. This type of problem motivates the development of adaptive techniques that have the flexibility to delete previously selected templates or to reinsert deleted ones. For example, Pudil *et al*., (1994) developed two variants of a floating search method that directly address the nesting problem, a floating forward selection (FFS) method and a floating backward search (FBS) method. These methods generally yield superior performing

TABLE 1. *Total number of template combinations that must be searched where m is the total number of available templates and n is the size of the subset.*

Exhaustive search (ES)	$N(\text{ES}) = m!/[n!(m-n)!]$
Sequential Backward Selection (SBS)	$N(\text{SBS}) = 1 + [(m+1)m - n(n+1)]/2$
Sequential Forward Selection (SFS)	$N(\text{SFS}) = nm - n(n-1)/2$

template sets. However, the discovery process is more expensive computationally than the SBS and SFS methods (although less so than Exhaustive Search). Figure 2 is a plot of the total number of template combinations that must be considered using the ES, FFS, and FBS algorithms. In this specific example we chose m =21 and $0 < n < 21$. Even for this relatively small value of m, the total possible number of combinations in the ES method is enormous for most values of n (particularly mid-range values $n \sim m/2$) and motivates the development of "smarter" methods. For example, with these parameters the SFS and SBS methods, the number of templates that must be tested is down by approximately three orders of magnitude relative to the exhaustive search method.

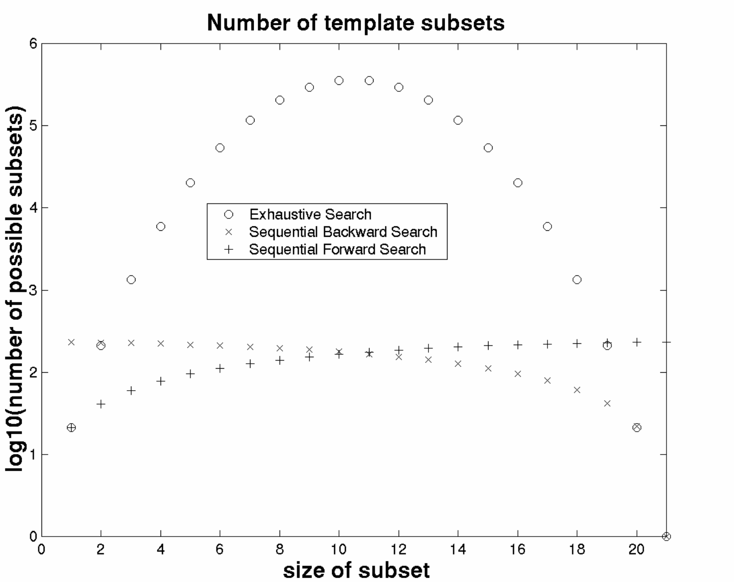

Figure 2. *A plot of the total number of template combinations that must be considered using the ES, FFS, and FBS algorithms.*

This dramatic difference becomes even more pronounced when one considers template selection from a statistical perspective. Referring to Figure 1, it seems reasonable to speculate that the optimal template set for fixed (m,n) should not be chosen relative to just one "snapshot" of test sample and reference templates, rather an ensemble of "snapshots" or realizations drawn from a distribution consistent with the known (or hypothesized) variance statistics of the features that define the templates. In this case, the same experiment would need to be repeated for each realization. This approach would quickly become impractical or even impossible using the ES method or other methods that do not demonstrate radically superior efficiency. Of course, there are many other selection algorithms that one may consider (*e.g.*, Jain *et al*., 2000), and their use for this or any other problem should be considered relative to the specific physics or phenomenology at hand. For example, the dynamic programming and Branch-and-Bound methods will likely yield template sets with performance superior to the sets chosen by SFS and SBS but the computational complexity will be much higher and cannot be precisely quantified in advance. Further, it is difficult to predict in advance the margin in improvement. In this paper, the template selection methods that we have numerically tested include a new joint class SFS technique and a new joint forward adaptive technique as well that we label FFS (but which is distinct from the FFS method of Pudil *et al*., 1994).

EXPERIMENTAL APPROACH

As described in the introduction, our primary interest in this study is representation complexity. For the template-based ATR problem, an obvious choice of metric for this type of complexity is simply the number of templates required to achieve a desired performance level. This metric has a clear meaning, is quantitative rather than qualitative, and at least for the experiments considered here it yields logical orderings. That is, the metric value increases monotonically with increasing problem difficulty.

Our approach is to assign a complexity metric based upon the number of templates required by an ATR to achieve a desired performance level. As mentioned earlier, this approach to complexity characterization is closely linked to the well-developed notions of Kolmogorov (or algorithmic) complexity of finite sequences of binary digits, *i.e*., binary strings. The Kolmogorov complexity of a binary string is equal to the length of the shortest computer program that generates the string. A somewhat surprising fact is that Kolmogorov complexity is essentially computer independent. The reason is that given any two computers *A* and *B*, an emulation program can be written for computer *B* that allows it to run programs produced for computer *A*. Suppose the emulation program has length L_E. Then, given any computer program of length L_A for computer *A*, an equivalent program for computer *B* can be written having length $L_A + L_E$ by concatenating the emulation program followed by the program written for computer *A*. The crucial point is that the length of the emulation program is fixed and independent of the length of any other program that we might want transition from computer *A* to computer *B*; thus, *the difference in the Kolmogorov complexity of any binary string x with respect to two different computers has a magnitude bounded by a constant that is independent of x.* In exactly the same way and for exactly the same reason, our measure here of ATR problem complexity is essentially computer independent. However, in the ATR context, rather than building minimal descriptions of binary strings, we are building minimal descriptions of target-signature spaces, *i.e*., we are encoding just enough information into our classification computer program to ensure that competing target hypotheses can be disambiguated accurately. As with the previous two metrics, the numerical value of this metric is an intrinsic attribute of the classification problem.

In order to develop empirical insight into representation complexity for template-based ATR, we performed a sequence of template selection experiments for the three-class target problems illustrated below in Figure 3. Previously conducted comparison experiments for SAR-based ATRs have shown that classification performance for multi-target problems degrades if the targets are similar in size. The targets in Problem 1 (see Fig. 3) differ greatly in size whereas the targets in Problem 2 are similarly sized. As we shall see, our numerical experiments yield larger-valued complexity measures for the second problem, consistent with our expectations.

For the problem considered here, the value of the complexity metric is simply the dimension of the sub-population that yields the desired performance level in the evaluation population. Meaningful rank ordering of problem complexity using this approach can only be achieved if due measures have been taken to ensure that the optimal or near-optimal combination of templates has been discovered for each dimension of the sub-populations. For example, it would not be meaningful to compare performance levels of a given algorithm for Problems 1 and 2 as a function of the size of the template subset if the templates in these subsets are arbitrarily chosen. In our experiments uniform sampling of the pose space provides a baseline metric value. We compare the performance of these template sets to that of two distinct selection techniques that attempt to provide better-reasoned selection guidelines. These latter techniques require a set of scenes for training, and a disjoint evaluation scene set is utilized to evaluate the performance of the template sets from all three methods. For these problems, the synthetic aperture radar (SAR) data is synthetically generated at *standard operating conditions* (*i.e*., nominally configured targets in the clear).

We constructed a simple ATR that identifies targets within scene ROIs, (regions of interest), each containing the signature of a single target. Our classifier compares each such scene ROI (henceforth referred to as a *scene*) against a library of stored templates, and identifies it as the target type of the template that most closely matches. We utilize the quantized MSE and peak match metrics from the MSTAR ATR system to compute similarity between chips and templates.

The imagery utilized for this research was generated synthetically using the MSTAR *Predict* module (*e.g*., Keydel and Lee, 1996) and the electromagnetic simulation code XPATCH (Andersh *et al*., 1994) as turntables of target signatures at one degree spacing. Each of these turntables was generated at one foot resolution and partitioned into three disjoint subsets, each containing 120 images at 3-degree intervals:

- **Templates:** A set *U* containing the universe of potential templates.
- **Training:** A set *R* of scenes available for training the system.
- **Evaluation:** A set *E* of scenes used for evaluating system performance

Challenges with our empirical approach include a lack of useful universal distance metrics between templates, sensitivity to template selection techniques, and cost of data collection. Our experiments investigate if our proposed ATR complexity metrics classify problem 1 as easier than problem 2 for the match metrics described above and for three template selection algorithms.

The MSTAR *Predict* module, which approximates XPATCH output, provides a fast tool that can efficiently generate target signatures at low cost. Our experiments were duplicated on XPATCH-T and MSTAR *Predict* generated signatures in order to detect differences in the characteristics of signatures generated using either system.

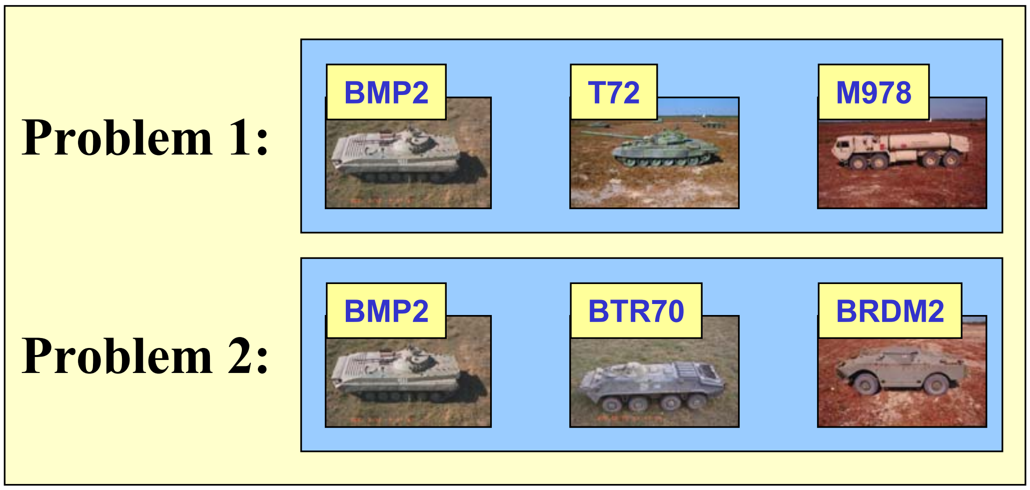

Figure 3. *The two, three-class problems considered for this paper. Note that the three targets in problem 1 are significantly different in size, whereas the three targets of problem 2 are all similar in size. In consequence, problem 1 is less difficult than problem 2.*

TEMPLATE SET SELECTION ALGORITHMS CONSIDERED

Common approaches to template set selection consider only one target class at a time. From a practical perspective, this approach seems reasonable, since it does not require re-selection of templates for each new classification problem. However, this simplification is almost certainly sub-optimal, *i.e.,* a performance gain can almost certainly be realized by determining jointly the set of templates to use from for each target class. We consider two algorithms that perform such joint building, and compare their performance to template sets selected from uniform distributions.

We define an optimally efficient template set as one that yields the required performance level with the minimal number of templates. We use training set *R* to evaluate potential template sets. Clearly an exhaustive search of all potential template sets is intractable. Our work investigates three sub-optimal methods for generating templates sets. These include:

- Uniform sampling over the available pose space,
- Greedy insertion, also known as sequential forward selection (SFS),
- Adaptive floating method, or floating forward selection (FFS).

In all of these experiments, template sets were selected from the universe *U* of potential templates sampled at 3-degree intervals as described above. We define $P(S,T,m)$ as an ATR performance metric that equal the fraction of scenes from set *S* that are identified correctly using template library *T* and match metric *m*.

Our uniformly sampled template libraries are sampled at equal intervals throughout the pose space, without reference to the training data. Our multi-class joint SFS algorithm[2] sequentially generates a vector of template sets ranging from size *0* to maximum size *x*. We use the notation T_i to denote a template set containing *i* templates. Setting T_0 to the null set initializes the process. T_{i+1} is generated from T_i by inserting a template *t* from the set $U\text{-}T_i$ selected such that $P(T_{i+1}, S, m)$ is maximized. The insertion process is terminated at set size *x* when the performance metric ceases to improve. That is, *x* satisfies the condition $P(T_{x+1}, \text{S}, m) = P(T_x, S, m)$. Our implementation of SFS avoids re-computation of scores by successively refining sets of templates that can improve scene identification as templates are added. Note that this algorithm only makes irreversible template insertions and may therefore generate sub-optimal sets. For example, for set size *j*, there may be some template *t* such that $P(T_{j+1} - t, S, m) > P(T_j, S, m)$ for T_{j+1} and T_j both generated by the greedy technique. We also developed an adaptive floating technique motivated by the developments in Pudil *et al*., (1994) and Somol *et al*., (1999) that, like the greedy SFS algorithm, begins with a minimal (size zero) template set and iteratively adds templates to generate

[2] Alternative greedy algorithms that remove (rather than inserting) templates were not examined. This approach also suffers from the irreversibility of decisions *i.e*., a template, once removed, cannot be reinserted.

larger template sets that maximize the performance level. However, at each step, our FFS algorithm considers single template deletions that result in better performing template sets at a fixed size.

Our FFS algorithm utilizes a cursor k (initially $k=0$) that selects the template set size being considered for editing. This algorithm first considers the highest performing template set of size k-1 that can be generated by a single deletion. If this edit results in a significant score improvement, the edit is performed and the cursor k is decremented. Otherwise, an insertion edit is similarly considered, and k is incremented. As with the SFS algorithm, this algorithm terminates when advancing the cursor beyond some $k=x$ yields no performance improvement beyond the best template set yet found. There exists a finite set of template sets of any size, and the FFS algorithm selects only the template sets that improve performance. This property bounds the number of operations performed by the algorithm and it ultimately terminates.

The template selection algorithm that we implemented for the SFS method scales as $n^2 \log n$. However, we did not attempt to implement an efficient algorithm for the FFS template selection method. We used the interpretive language Python[3] to implement these algorithms since it greatly facilitates rapid algorithmic prototyping. Python's syntax is exceptionally expressive. However, its execution speed is abysmal. Despite this, our SFS implementation runs quite quickly. For example, execution of this algorithm to compute template sets from a database of match distances for the three-class problems examined required less than two minutes computation time on a 350 MHz COTS system. By comparison, our unoptimized prototype implementation of the FFS method required many hours of CPU time to compute a full range of template set sizes. For both algorithms, implementations coded in a compiled language will execute far faster. All of our experiments were conducted using a pre-computed database of template-to-scene distances.

EXPERIMENTAL RESULTS

Figure 4 shows the azimuthal distribution of templates selected using the SFS and FFS selection algorithms on imagery generated using the MSTAR *Predict* module. (Table 2 provides a list of abbreviations used in Figures 4-6). Scenes were identified using the MSTAR quantized grayscale match metric, for the easy and hard classification problems. In these polar plots the radial coordinate indicates the P_{id} level whereas template azimuths are given by the angular coordinate. The irreversibility of insertion decisions in the SFS algorithm implies that template populations in successive P_{id} levels (the circles in Fig. 4) are supersets of populations in the lower-valued P_{id} levels, a property not necessarily shared by template populations selected by the FFS algorithm. The left and right columns of the figures denote the distribution of templates for the easy and hard problems and evaluation scene sets E, respectively. The plots in the upper and lower rows are for template sets selected by the FFS and SFS algorithms respectively.

Table 2. *Abbreviations used in Figures 4-6.*

Category	Value	Abbreviation
Data Source	MSTAR Predictor	pred
	XPatch-T	xpatch
Selection Algorithm	SFS	sfs
	FFS	ffs
	Uniform	uni
Problem	hard (2)	hard
	easy (1)	easy
Scene set	evaluation scenes *(E)*	eval
	training scenes *(R)*	train
Match metric	Quantile MSEgrayscale	gray
	Peak	peaks

[3] See http://www.python.org/.

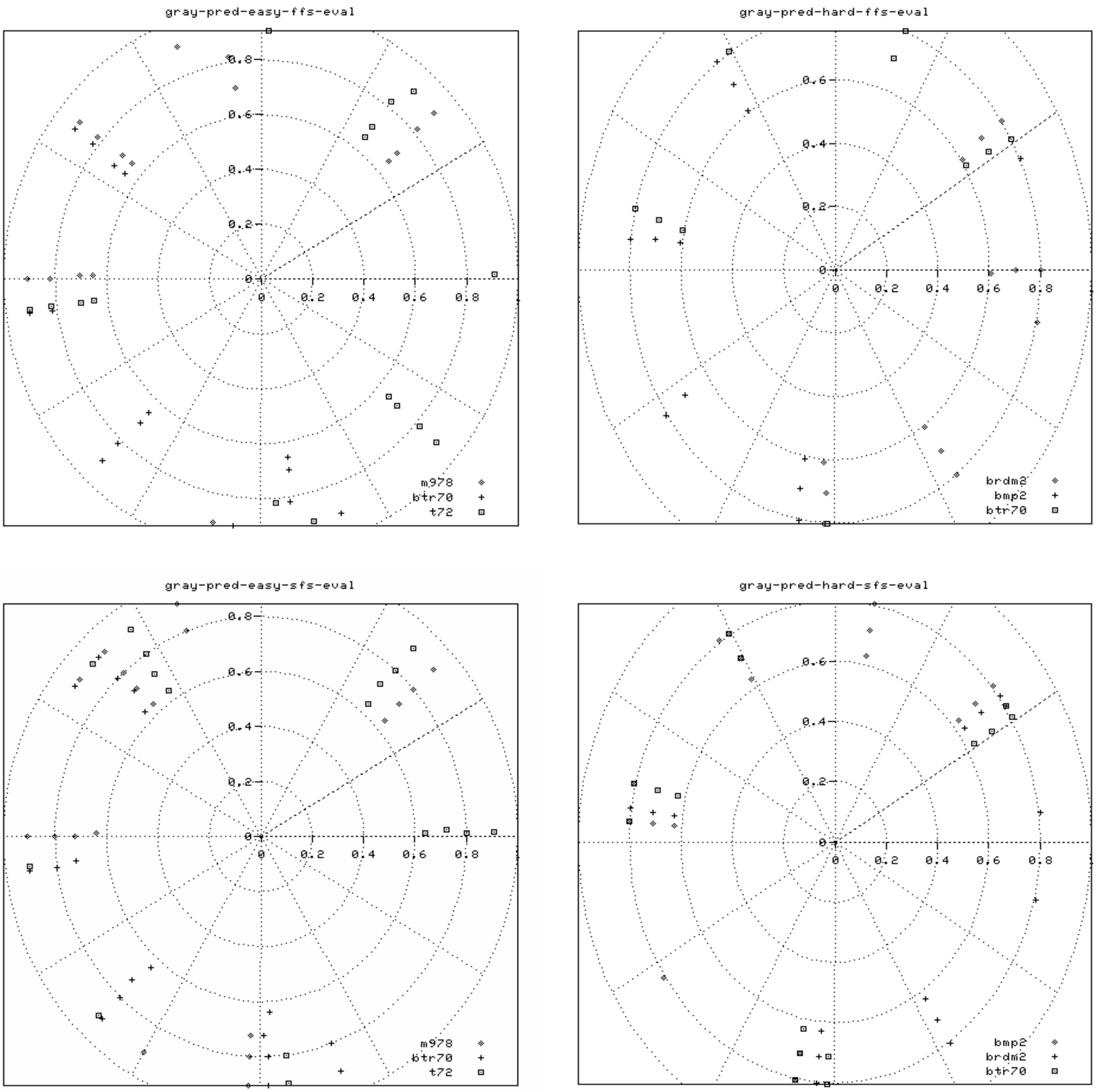

Figure 4. *Azimuth distribution of templates using the grayscale MSE metric on PREDICT data at P_{id} 0.6, 0.7, 0.8, and 0.9 on the evaluation scene set E. Plots on the left (right) column are template distributions for the easy (hard) problem. The plots in the upper and lower rows are for template sets selected by the FFS and SFS algorithms, respectively. Note that for a fixed P_{id} level, the number of required templates is significantly greater for the hard problem set than the easy problem, and that the SFS and FFS algorithms do not generate template sets with P_{id} =0.9 for the hard problem.*

Figures 5.1 through 5.4 contain performance results for both the easy and hard problems for template sets selected by the SFS algorithm and uniform azimuth distributions as a function of template set size. The sets with uniformly distributed templates have similar performance for both the training and evaluation scene sets. Note that in evaluation against both the training and testing sets, more templates are required for the hard problem than for the easy problem to achieve a fixed P_{id} level. This behavior is consistent with expectations, and persists regardless of whether the templates are uniformly or non-uniformly spaced. Note that for a fixed P_{id} level, the number of required templates is significantly greater for set *E* than set *R*. This behavior is not surprising, since the template-building algorithm is specifically tailored to work well on the training set *R*; thus, the distribution of templates in shown in the evaluation plots provides a more realistic prediction of what size template set will be required in practice to achieve a desired level of classification performance. In all of our numerical experiments, template sets selected by the greedy SFS technique yielded substantially superior performance to those with uniform sampling over the full range of template set sizes. In fact, the performance levels of uniformly sampled template sets of size *n* are matched by template sets half that size selected using the SFS or FFS algorithms for a wide range of *n*.

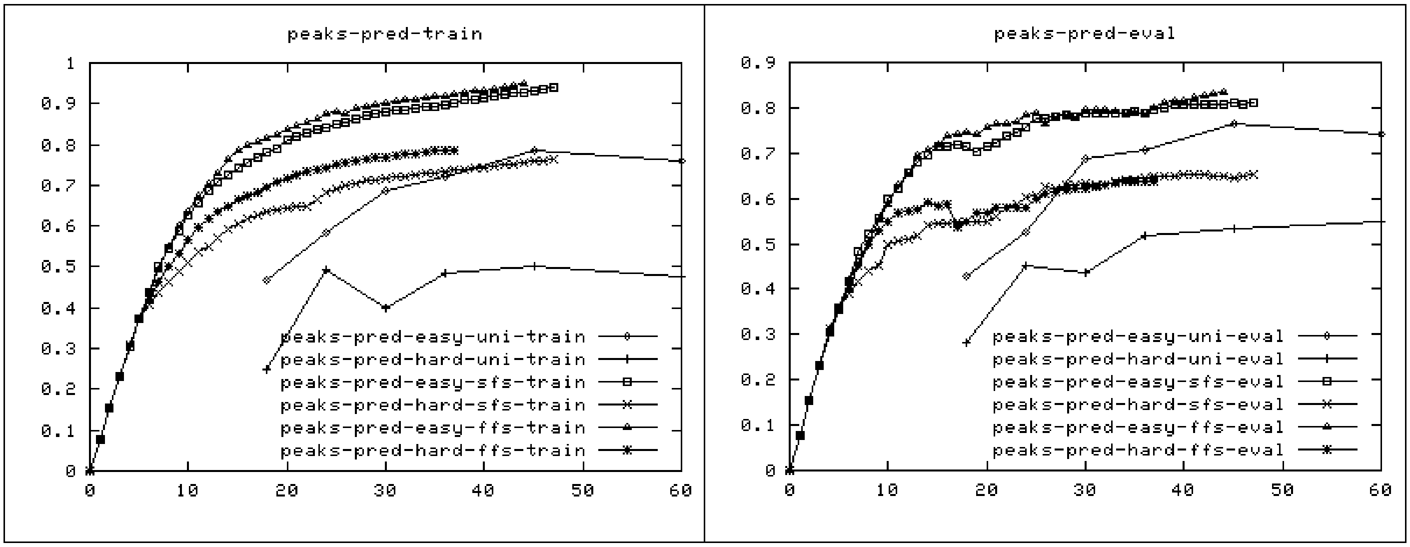

Figure 5.1. *P_{id} versus number of templates for both easy and hard problems as evaluated on training and evaluation data, using the peak match metric and imagery generated using the MSTAR Predict module.*

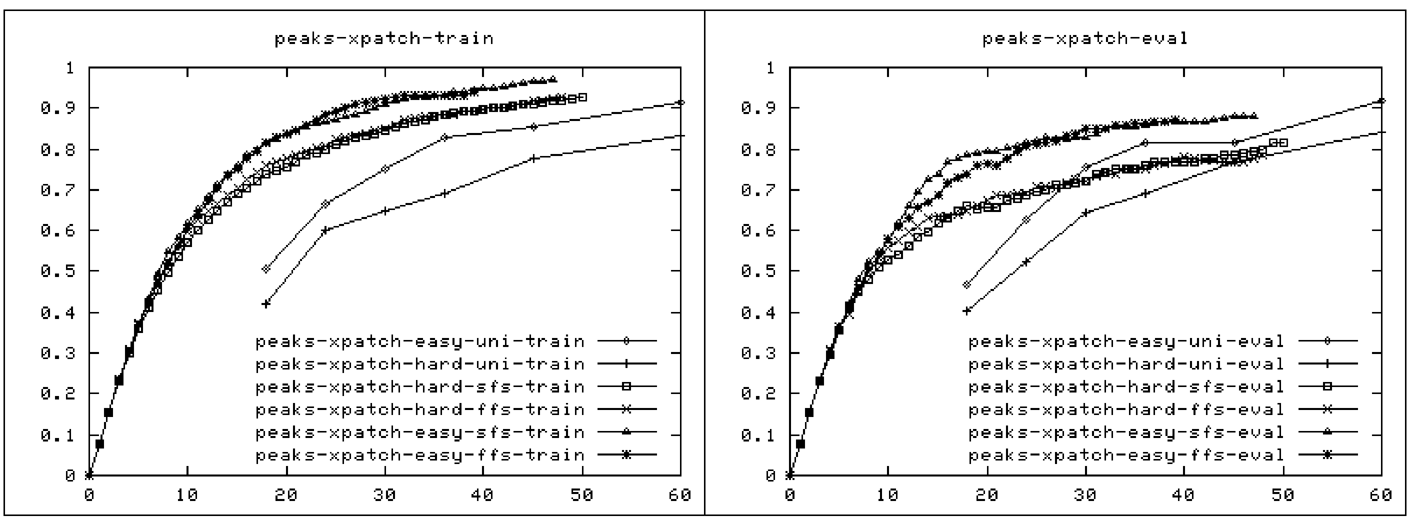

Figure 5.2. *P_{id} versus number of templates for both easy and hard problems as evaluated on training and evaluation data, using the peak match metric and imagery generated using Xpatch-T.*

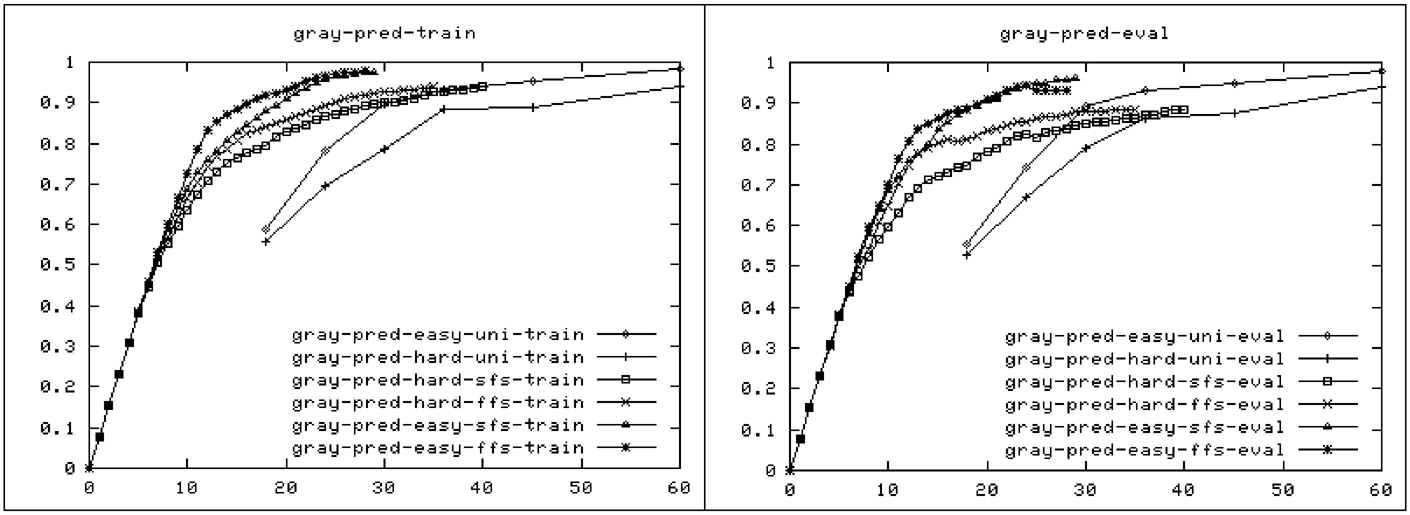

Figure 5.3. *P_{id} versus number of templates for both easy and hard problems as evaluated on training and evaluation data, using the quantized MSE match metric and imagery generated using the MSTAR Predict module.*

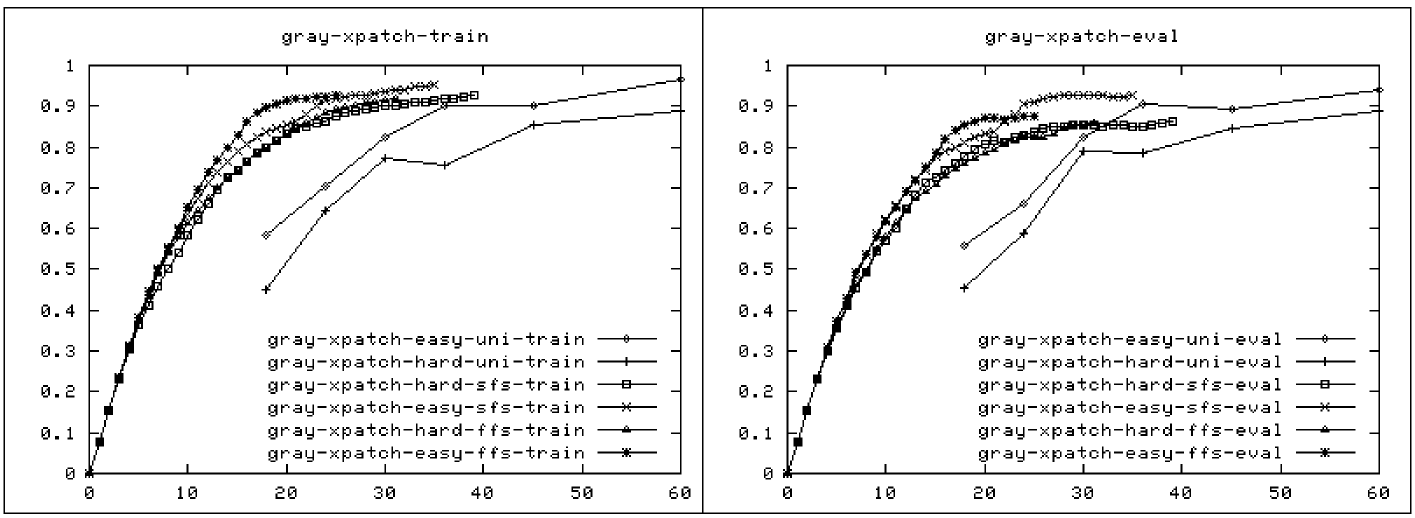

Figure 5.4. *P_{id} versus number of templates for both easy and hard problems as evaluated on training and evaluation data, using the quantized MSE match metric and imagery generated using Xpatch-T*

Figure 6 presents the performance of ATRs constructed using differing match metrics and image sources, and identical template selection algorithms. These plots indicate that ATRs containing the same number of templates that utilize the quantized MSE grayscale metric have superior performance than ATRs that utilize the peak metric. The template sets with uniform sampling in the pose angle yield similar performance for both the training and evaluation scene sets, indicating that both scene sets have similar characteristics. Interestingly, the uniformly sampled template sets' performance often does not increase monotonically with the number of templates; this effect is not present for the template sets selected by the SFS and FFS algorithms, whose performance increases monotonically with template set size. This instability is more pronounced for ATRs that utilize the peak metric on imagery generated by the MSTAR predictor, indicating a qualitative difference between these matchers and signature synthesis systems

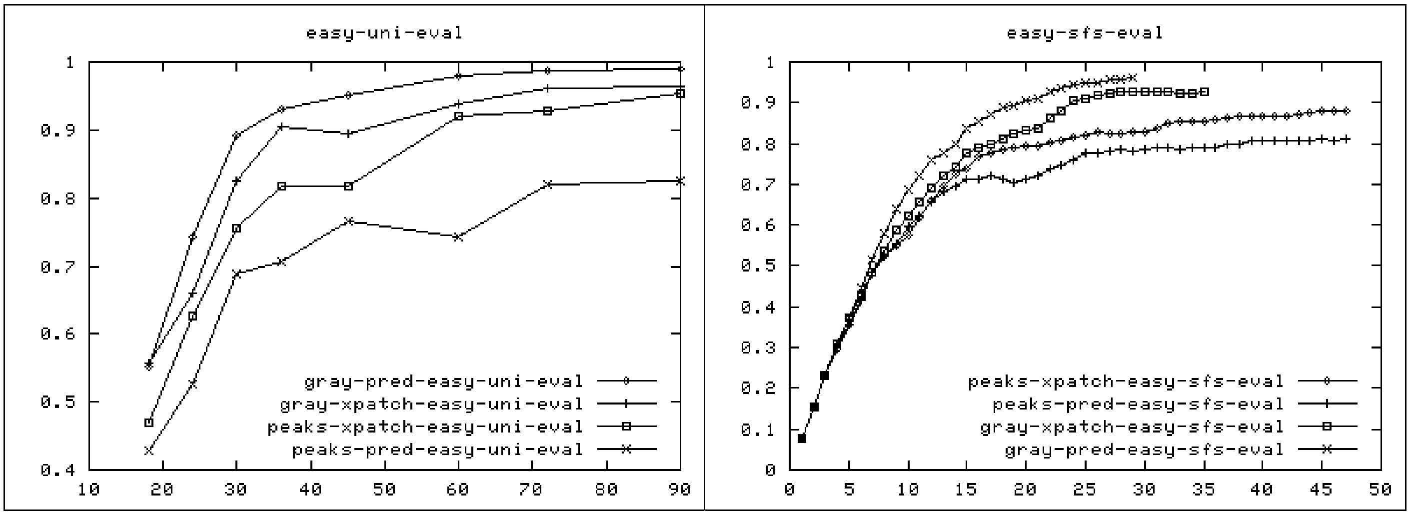

Figure 6. *P_{id} for multiple image sources & match metrics versus number of templates. The peak metric for both the uniformly sampled and SFS-generated template sets yields performance levels that are far inferior to those of the grayscale metric. The performance differences between data computed using XPATCH versus data computed using the MSTAR Predict Module are particularly large for the peak match metric.*

CONCLUSIONS

We have proposed a complexity metric for template-based SAR ATR given simply by the template set size required to achieve a prescribed performance level. Our results show that this metric has the desired properties of a complexity metric in that it is numerically valued, gives logical orderings, and is readily computable. We also investigated new joint template selection algorithms that yielded template sets whose performance levels were superior to those using simpler selection techniques such as uniform sampling.

Our complexity metric reliably identifies the more difficult of the two problems that we considered. All of the performance plots indicate that for equal-sized template sets there is a significant performance gap between the hard and easy problems. This result is independent of the selection algorithm and the match metrics that we considered. Since our template selection techniques are sub-optimal, we do not know the distance between the minimum complexity values of a particular problem and the highest performing template sets selected by our adaptive algorithm. However, as indicated in the introduction, a high-confidence determination could be made using a more sophisticated (but much more computationally intensive) technique such as Branch-and-Bound. An additional shortcoming of our complexity metric is its dependency on the algorithms used to compute image similarity and to select templates. Discriminators utilizing the quantized MSE grayscale match metric had higher performance for both Xpatch and MSTAR *Predict* data than discriminators utilizing the same number of templates and the peak metric.

In all cases, similarly sized template sets selected by the SFS algorithm outperformed uniformly sampled template sets of the same size. Template sets selected by FFS algorithms generally outperformed uniformly sampled template sets of the same size. Furthermore, the plots for uniform data are non-monotonic with the number of templates, indicating instability in this approach and suggesting that optimized template selection may lead to more robust ATR systems that utilize fewer templates.

The FFS selection algorithm may have utility for dynamically selecting template sets tailored for particular applications. For example, an ATR with a large library of available templates can be used for a variety of missions with differing performance requirements and potential targets. Optimized implementations of our FFS or SFS algorithms could be tasked to dynamically generate pruned template sets that are optimized for each mission, provided that a pre-computed template-to-scene distance database is also available.

ACKNOWLEDGEMENTS

This research was supported by AFRL under Contract Number F33615-99-C-1509. The authors would like to thank the contract sponsors at AFRL for their support. We would like to thank John Wissinger, Bill Irving, Tom Ryan, Carol Ouellette, and Robert Washburn for contributing to the early stages of this work. We are also grateful for the assistance of Alden Penn who composed scripts to generate the azimuth distribution plots.

REFERENCES

Andersh, D.J., Hazlett, M., Lee, S., W., Reeves, D.D., Sullivan, D.P., and Chu, Y., *Xpatch*: A high frequency electromagnetic scattering prediction code and environment for complex three-dimensional objects*, IEEE Antennas and Propagation Magazine*, **36**, 65-69, 1994.

Cover, T.M., and Thomas, J.A., *Elements of Information Theory*, John Wiley & Sons, Inc., New York, 1991.

Jain, A.K., Duin, R.P.W., and Mao, J., Statistical pattern recognition: a review, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **22**, 4-37, 2000.

Keydel, E.R., and Lee, S.W., Signature prediction for model-based automatic target recognition, in *Algorithms for Synthetic Aperture Radar Imagery III*, SPIE vol. 2757, 306-317, 1996.

Ettinger, G.J., Klanderman, G.A., Wells, W.W. and Grimson, E.L., 1996, A probabilistic optimization approach to SAR feature matching, in *Algorithms for Synthetic Aperture Radar Imagery III*, SPIE Vol. 2757, pp. 318-329.

Irving, W., and Ettinger, G.J., Classification of Targets in Synthetic Aperature Radar Imagery via Quantized Grayscale Matching, in *SPIE Conference #3721, Algorithms for Synthetic Aperture Radar Imagery VI,* Orlando, FL, April 1999.

Mosegaard, K., and Tarantola, A., Monte Carlo sampling of solutions to inverse problems, *J. Geophys. Res*., **100**, 12431-12447, 1995.

Pudil, P., Novovicova, J., and Kittler, J., Floating search methods in feature selection, *Pattern Recognition Letters*, **15**, 1119-1125, 1994.

Ryan, T.W., Pothier, S., Pierson, W., Search algorithms for vector quantization and nearest neighbor classification, these proceedings, 2001.

Sambridge, M., Exploring multidimensional landscapes without a map, *Inverse Problems*, **14**, 427-440, 1998.

Somol, P., Pudil, P., Novovicova, J., and Paclik, P., Adaptive floating search methods in feature selection, *Pattern Recognition Letters*, **20**, 1157-1163, 1999.

Tarantola, A., *Inverse Problem Theory*, Elsevier, Amsterdam, 1987.

Wissinger, J., Washburn, R.B., Friedland, N.S., Nowicki, A., Morgan, D.R., Chong, C., and Fung, R., Search algorithms for model-based SAR ATR, in *Algorithms for Synthetic Aperture Radar Imagery III*, SPIE vol. 2757, 279-293, 1996.