Performance of the MACH Filter and DCCF algorithms on the 10-class public release MSTAR data set

Abhijit Mahalanobis, Luis A. Ortiz
Raytheon Missile System Company
Bldg. 840, MS 8
P.O. Box 11337
Tucson, AZ 85734

B.V.K. Vijaya Kumar
Dept. of Electrical and Computer Engr.
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

The *maximum average correlation height* (MACH) filter and *distance classifier correlation filter* (DCCF) correlation algorithms are evaluated using the 10 class publicly released MSTAR database. The successful performance of these algorithms on a 3-class problem has been previously reported. The algorithms are optimized by design to be robust to variations (distortions) in the target's signature as well as discriminate between classes. Unlike Matched Filtering (or other template based methods), the proposed approach requires relatively few filters. The paper reviews the theory of the algorithm, key practical advantages and details of test results on the 10-class public MSTAR database.

1. Motivation

The performance of correlation based techniques for recognizing targets in SAR imagery is very promising and has been reported in recent publications [1]. It was shown that advanced correlation filter design techniques known as the MACH filter [2] and the DCCF [3] can recognize targets in a 3-class subset of the public domain MSTAR data set. In this paper, we extend the results to the presently available 10-class data set. Correlation offers many benefits such as analytical performance optimization and easy algorithm implementation. In fact, the main processing engine is the fast Fourier transform (FFT) which is standard in many software packages and also available in special purpose hardware. The processing complexity depends on the scene size (or search area), and not on scene content, clutter or the number of targets. Another benefit of correlation not to be overlooked is *shift-invariance* and the ability to process multiple instances of an object in an image in parallel.

It has been pointed out in previous publications that the analytical formulation of the algorithms exploits linear systems theory to recognize patterns by optimizing statistical and signal processing metrics. Thus, the algorithms are known to be optimum within the confines of the definitions of distortion tolerance and discrimination as formulated. At Raytheon, we have shown that the resulting algorithms are easy to implement with relatively simple code in existing low-cost processors. We feel that based on the results reported in this paper and elsewhere, the proposed correlation approach to ATR offers a high *cost versus performance* advantage.

Part of the SPIE Conference on Algorithms for Synthetic Aperture Radar Imagery VI
Orlando, Florida • April 1999
SPIE Vol. 3721 • 0277-786X/99/$10.00

2. ATR Description

The application of correlation filters for SAR ATR was first examined by Mahalanobis et. al [4,5], and subsequently reported by other independent investigators [6]. A simplified architecture[1] for SAR ATR is shown in Figure 1. The wide area image is rapidly processed by a screener to find regions of interest. Every detection which exceeds a threshold is processed by the main ATR section. For speedier processing, a pose estimate is used to index the correct filters. In this case, the MACH and DCCF filters for the estimated angle as well as its opposite should be used to account for the 180° ambiguity in pose estimates. On the other hand, knowledge of pose is not essential since the test image can be processed by all filters and the best results over all aspects can be selected. The peak-to-sidelobe ratio (PSR) of the peaks generated by the MACH filter and the distances calculated using the DCCF are compared to thresholds. The image is considered to belong to the class with the highest PSR and smallest distance or rejected if the metrics do not pass the threshold tests.

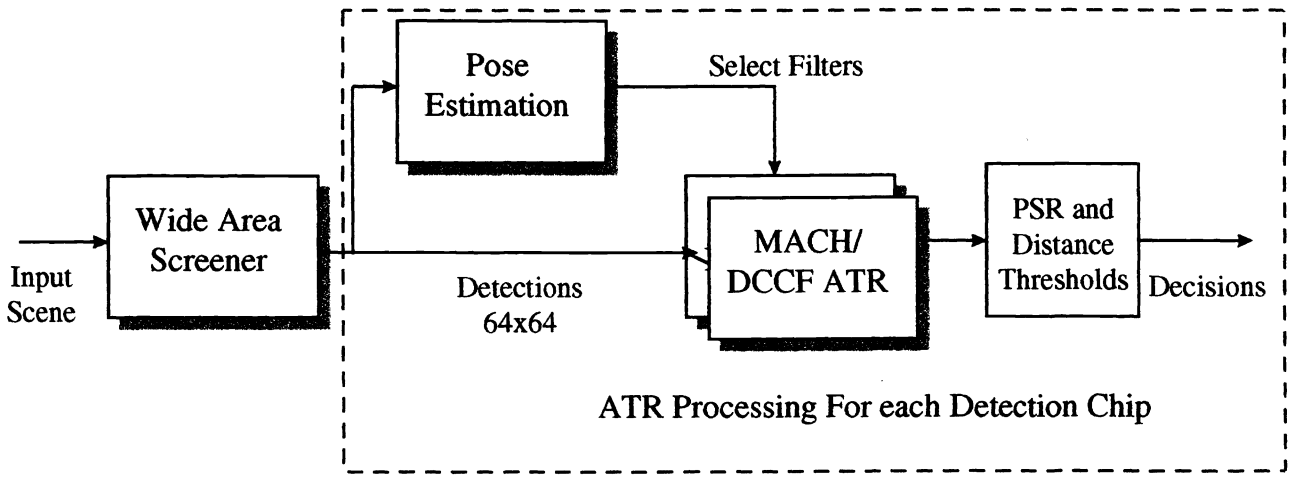

Figure 1: Simplified SAR ATR Architecture using MACH/DCCF correlation Filters

2.1 Algorithm Overview

The 10-class ATR is a direct extension of the earlier work reported in Reference 1. The details of the MACH filter and DCCFs are given in detail in references 2 and 3. However, the conceptual operation of the algorithm is reviewed here for completeness. The locations of the scene that are nominated by the screener as potential targets are "chipped" out and processed by the MACH/DCCF ATR. The appropriate MACH filter and DCCFs for the estimated pose are then used to classify the target. An example of MACH filter processing is shown in Figure 2 along with sample test images of the three classes (BTR70, T72 and BTR). Also shown is a MACH filter designed to recognize a T72 between 0° and 45° along with the correlation surfaces produced by the filter in response to the three target images. It is evident that the best peak with the highest PSR corresponds to the T72.

[1] Details of algorithm implementation which are Raytheon proprietary have been intentionally excluded from this paper

If the MACH PSR is above an acceptable threshold, the image is further processed by the DCCFs. The basic idea of the DCCF is shown in Figure 3. Essentially, the image to be classified is first processed by a *transformation filter* (correlation 1) followed by a *shift-invariant* minimum mean square error (MSE) calculation (calculation 2). The purpose of the transform filter is to enhance discrimination by maximizing the average spectral separation between the classes. The image is then classified by computing the MSE (or distance) between the transformed test image and the ideal references of each class (which are also similarly *transformed* by the DCCF). Since DCCFs minimize in-class distances by design, the input image is assigned to the class with the smallest distance.

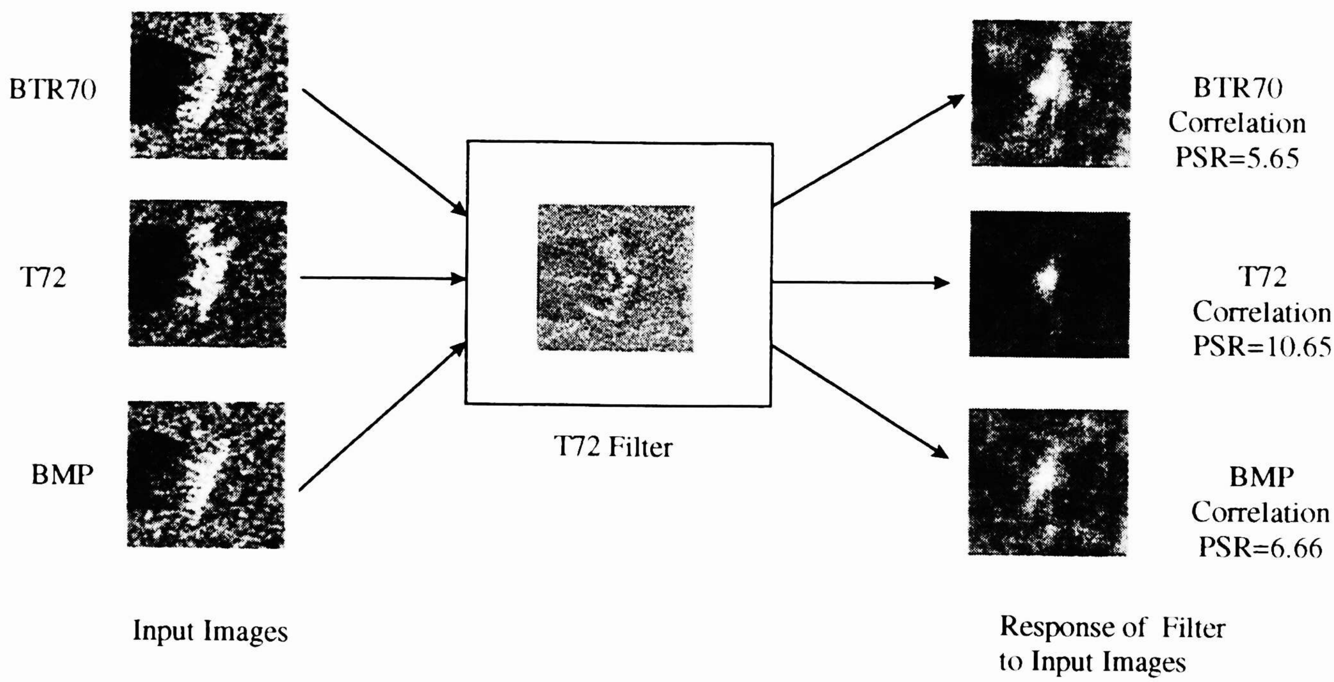

Figure 2: When correlated with test images of a BTR70, T72 and BMP, a T72 MACH filter produces the strongest peak (PSR = 10.65) in response to the T72 image.

The ATR uses several filters each of which is designed to recognize the targets over a range of angles (referred to as *aspect bins*). In general, the number of filters to be stored is given by the formula $N = \frac{A*(2C+1)}{2}$, where A is the number of aspect bins and C is the number of classes. Not all of the references need to be run at the same time. Assuming that pose estimate is used, only C correlation must be performed. The runtime ATR code requires about 300 lines in MATLAB® (equivalent to about 5000 lines of C code). Low-cost processor designs based on four C60 processors can process a 1024 x 1024 image in less than 0.2 seconds (or at a rate of over 5 million pixels per second).

® MATLAB is a registered trademark of Mathworks, Inc.

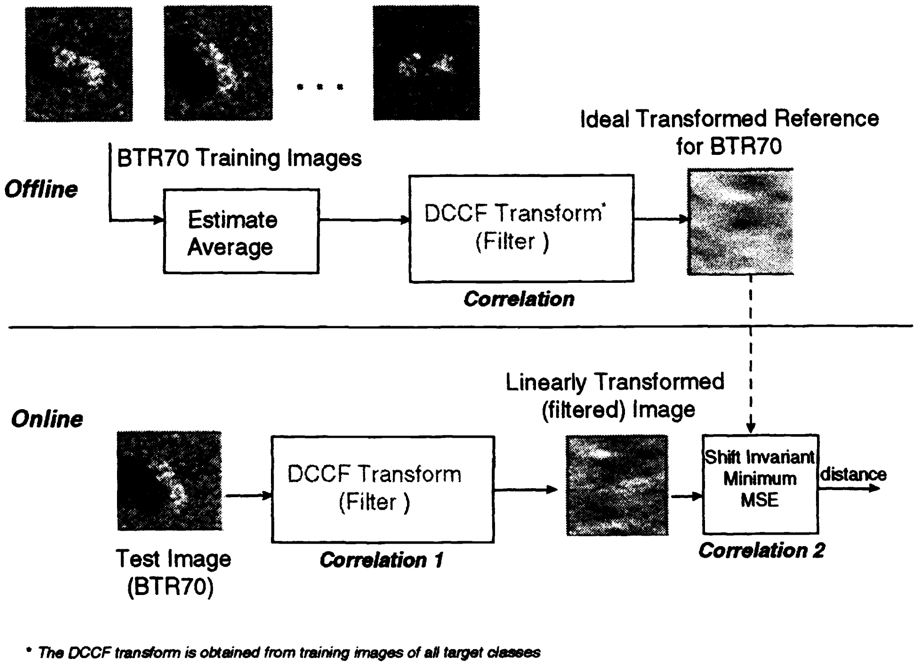

Figure 3: The DCCF classification process

3. SAR ATR Results

The results described here are obtained using the public domain MSTAR database which contains 10 targets at 1-foot resolution. The number of aspect bins was chosen to be 16. For each aspect bin, the filters were trained using targets at 17° depression and tested at the 15° depression angle. Simple image processing techniques were used to remove the background from the training images. For the 10-class problem the number of MACH filters is 160. As mentioned earlier, the optimum transform which maximizes the DCCF performance criterion is obtained by solving an eigen-value/eigen-vector problem. Typically only the dominant eigen-vector is used as the transform which maximizes the performance criterion. However, the remaining eigenvectors can be also used as DCCF transforms, albeit they yield sub-optimum values for the performance criterion. It has been observed that unless the eigen-values decay sharply, these sub-optimum solutions do carry useful information. Using any one eigenvector, the number of DCCF filters to be stored for the 10-class problem divided into sixteen aspects is 16 aspects $\times$ eleven filters = 176. Experiments were conducted using DCCFs based on all the eigen-vectors. Very simply, the distances were calculated separately using each of the nine available eigen-vectors as DCCF transforms (i.e., not just the dominant one) and added to obtain a composite distance measure for each class. In this approach, the total number of DCCF filters is 176 filters $\times$ 9 eigenvectors (or DCCF transform) = 1584. It should be noted that if necessary, various data compression techniques can be used for storing and efficient processing of the MACH and DCCF filters.

The results of the 10-Class ATR by combining all 9 eigenvector solutions for the DCCFs is shown in Table 1. The total number of filters required is 1744. The total number of test images is 2360. Over this set of images, the overall probability of being correct is (P_c) 94%, the probability of rejection (P_r) is 5%, and the average error rate (P_e) is 1%. It was observed that if the PSR and distance ratio thresholds are set such that P_r=0 , then P_c increases to 98% with an error rate of 2%. These thresholds can be adjusted depending on the desired false alarm rate which will reported in a later publication. We also studied the effect of using different number of eigenvectors. The change in the P_c as the number of eigenvectors used for DCCF computations is varied is shown in Figure 4. The case shown is for P_r =0. It is interesting to note that when only one DCCF solution is used (i.e., just the dominant eigenvector), the P_c is still 96%. However, the total number of filters required is only 336.

	2S1	BRDM	D7	T62	ZIL	ZSU	BMP2	BTR70	T72	BTR60	Reject
2S1	89.78						0.36	0.73		0.36	8.76
BRDM	0.3	92.7			0.36	0.36	0.36	0.36			5.47
D7			97.81								2.19
T62				96.70					0.73		2.56
ZIL		0.94			88.26			0.47			10.33
ZSU						97.81					2.19
BMP2	0.51				0.51		91.84				7.14
BTR70	1.02	1.53						91.84			5.61
T72						1.05			95.81		3.14
BTR60		0.51		1.54		0.51				92.82	4.62

Table 1: Summary of 10-Class ATR results

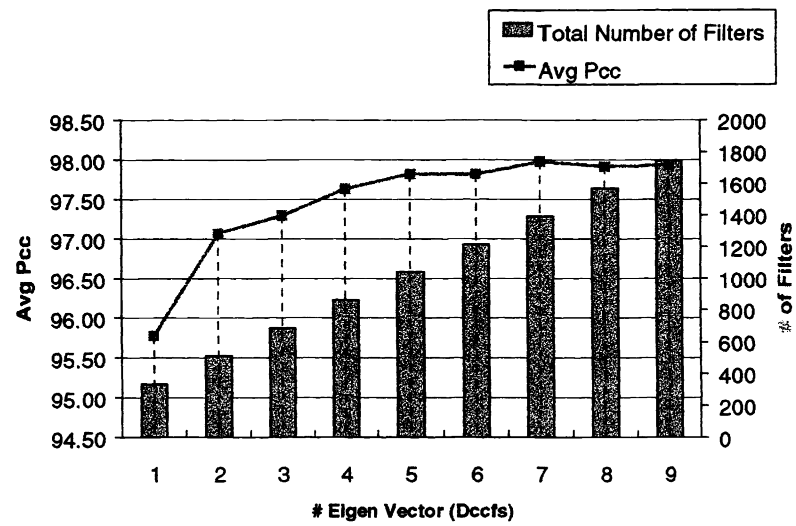

Figure 4: The total number of filters required and the behavior of P_c as a function of the number of eigen-vectors used for DCCF computations

Another approach which can significantly reduce the number of filters is to use a tree architecture where similar targets are grouped together at the higher nodes as shown in Figure 5. For instance, a 9-class problem can be implemented as a 5-class problem at the highest node, followed by further differentiation at the lower levels. This approach requires at most 800 filters instead of 1424 filters required for the direct 9-class implementation. Specifically, the T62 and T72 are combined into Class 1, the 2S1 and BMP are combined into Class 2, and the BRDM, BTR60 and BTR70 are combined into Class 3. The D7 and ZSU are distinct objects and treated as separate classes even at the highest level. Of course, the complexity of the tree can be arbitrarily increased with more nodes added at lower levels depending on the total number of classes to be recognized. Although fewer filters are required, there is a possibility that errors at higher nodes will propagate along an incorrect branch of the tree, leading to an erroneous decision.

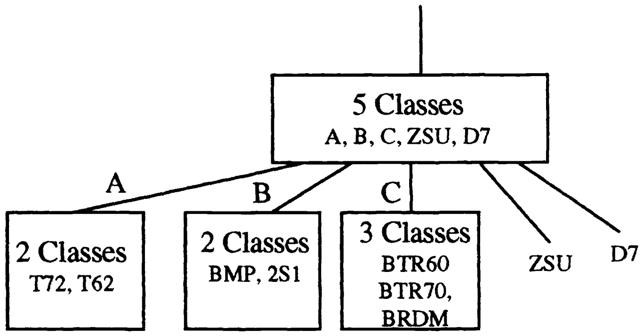

Figure 5: Multi-Level Tree Architecture for 9-class ATR using 5-, 3-, and 2- class stages.

	2S1	BRDM	D7	T62	ZSU	BMP2	BTR70	T72	BTR60	Reject
2S1	89.05	0.36			0.36		4.74	0.73	3.65	1.09
BRDM		94.89			0.36	2.55	2.19			
D7			100							
T62		0.37		97.80	0.37		0.73	0.37		0.37
ZSU			0.36		99.64					-
BMP2	0.51	0.51	1.02	1.53		92.35		3.57		0.51
BTR70	1.53	3.06		-			95.41			
T72				1.05				98.95		
BTR60	1.03	1.03		1.03	2.05		1.03		93.33	0.51

Table 2: Results of using the multi-level tree architecture in Figure 6 to implement a 9-class ATR.

As shown in Table 2, the ATR achieved a very high P_c of 97.5%, P_r = 1%, and Pe = 1.5%. While ROC curves are necessary for comparing the two approaches, it can be at least said that the multi-level tree architecture performs comparable to the direct implementation with far fewer filters.

4. Conclusion

We have extended the application of the MACH/DCCF algorithm for 9- and 10-class SAR ATR. Preliminary results obtained using the MSTAR database suggest that the approach is robust and capable of achieving high recognition rates while maintaining low error rates. Two approaches were examined. The first is a direct implementation of a 10-class DCCF using multiple eigenvectors. The second method uses a multi-level ATR where targets are grouped together at higher nodes. These correlation based algorithm can be easily and efficiently implemented using FFT software in low-cost high speed DSPs.

5. References

1. A. Mahalanobis, B.V.K. Vijaya Kumar, D.W. Carlson, "Evaluation of MACH and DCCF correlation filters for SAR ATR using the MSTAR Public Data Base", *Proceedings of SPIE, Algorithms for Synthetic Aperture Radar Imagery V,* Vol.. 3370, pp. 460-468, Orlando, April 1998

2. A. Mahalanobis, B.V.K. Vijaya Kumar, S.R.F. Sims, and J. Epperson, "Unconstrained Correlation Filters," *Applied Optics* , Vol. 33, pp. 3751-3759, 1994.

3. A. Mahalanobis, B.V.K. Vijaya Kumar, and S.R.F. Sims, "Distance classifier correlation filters for distortion tolerance, discrimination, and clutter rejection," *Proc. SPIE*, Vol. 2026, pp. 325-335, 1993.

4. A. Mahalanobis, A. Forman, M. Bower, R. Cherry and N. Day, "Multi-class SAR ATR using shift-invariant correlation filters," special issue of *Pattern Recognition* on correlation filters and neural networks, Vol. 27, pp. 619-626, 1994.

5. A. Mahalanobis, B.V.K. Vijaya Kumar, D.W. Carlson, and S.R.F. Sims, "Performance evaluation of distance classifier correlation filters," *Proc. SPIE,* Vol. 2238, pp. 2-13, 1994.

6. L. M. Novak, G.J. Owirka, and C. M. Netishen, "Radar target identification using spatial matched filters", Special issue of *Pattern Recognition* on correlation filters and neural networks, Vol. 27, pp. 607-617, 1994