Verification of a Decision Level Fusion Algorithm Using a Proven ATR System and Measured SAR Data

THESIS

J. Douglas Thompson

AFIT/GE/ENG/06-60

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense or the United States Government.

AFIT/GE/ENG/06-60

Verification of a Decision Level Fusion Algorithm Using a Proven ATR System and Measured SAR Data

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

J. Douglas Thompson

March 2006

AFIT/GE/ENG/06-60

Verification of a Decision Level Fusion Algorithm Using a Proven ATR System and Measured SAR Data

J. Douglas Thompson

Approved:

Steven C. Gustafson, Ph.D. (Chairman)	8 Mar 06 date
Richard K. Martin, Ph.D. (Member)	7 Mar 06 date
John F. Raquet, Ph.D. (Member)	7 MAR 06 date

Abstract

Decision level fusion (DLF) algorithms combine outputs of multiple single sensors to make one confident declaration of a target. This research compares performance results of a DLF algorithm using measured data and a proven ATR system with results from simulated data and a modeled ATR system. This comparison indicates that DLF offers significant performance improvements over single sensor looks. However, results based on simulated data and a modeled ATR are slightly optimistic and overestimate results from measured data and a proven ATR system by nearly 10% over all targets tested.

Table of Contents

Page

List of Figures

List of Tables

VERIFICATION OF A DECISION LEVEL FUSION ALGORITHM USING A PROVEN ATR SYSTEM AND MEASURED SAR DATA

I. Introduction

1.1. Background

Current and historical U.S. combat rules of engagement (ROE) require most surface and aerial targets to be identified and confirmed as hostile prior to directing hostile fire against them. ROE adherence is driven by the need to eliminate friendly-fire casualties (fratricide) as well as to protect enemy non-combatants, and it has historically been accomplished visually. Depending on the intended target, however, visual confirmation requires a combat aircraft to move into close proximity with a suspected target before the final decision to expend ordnance can be made, and it reduces or eliminates the advantage of range for standoff weapons. Likewise, the ability to prosecute multiple targets during a given engagement is greatly restricted.

Reliable combat identification and fratricide prevention are important because U.S. troop strengths and numbers of combat aircraft have steadily declined, forcing greater reliance on force-multiplying technologies along with greater weapon precision and lethality. Mistaken target identity reduces combat effectiveness, and friendly-fire accidents have a proportionately greater adverse effect on force structure.

Although friendly force structures are smaller and more efficient, potential adversary weapon systems are becoming more prolific and more capable of destruction. Successful countermeasures require rapid threat identification, prioritization, and destruction. Likely combat scenarios involve concentrations of disparate hostile weapon systems and threats, taxing the un-assisted limits of human visual acuity and target prioritization. It is likely that mistakes in combat engagement, such as target miss-prioritization or miss-identification, will increase unless existing target recognition/prioritization capabilities are improved.

To this end, fusion has become a keyword in the automatic target recognition (ATR) community. This thesis research aims to verify the simulated results of the Decision Level Fusion (DLF) algorithm developed by MCA, Inc. [Cohen, 2005] by testing the algorithm with measured data and a fielded ATR system and comparing with results produced using simulated data and a modeled ATR system.

1.2. ATR Processes

The first step in the ATR process is to gather data. In the case of the system used here, the data is in the form of synthetic aperture radar (SAR) images collected by a sensor flying onboard a surveillance aircraft platform. Once the images are collected, several steps must be taken to isolate only the area of interest in the image, or more specifically only the target of interest. The process of selecting only the target of interest is referred to as chipping out the target. This process leads to the image of only the target, commonly referred to as a chip.

Throughout this research the measured data has already been processed into individual chips, which allows the full scenario to be simulated using only these pre-processed chips and the geometric positions of the strike platforms relative to the target of interest. Occasionally, a single sensor can provide enough information to make accurate declarations of targets through the ATR process. However, by combining the output of several single sensor ATR systems through a fusion algorithm, results are expected to improve significantly.

1.3. Objective

The objective of this research is to verify the performance of the DLF algorithm in support of the Air Force Research Laboratory (AFRL) Sensors Directorate. To accomplish this verification, theoretical models and data, similar to that used in the initial development [Cohen, 2005] are compared with performance results based on measured SAR data in a simulated scenario. The measured data is from sponsored collections, and the ATR system is proven in that it is currently used on several surveillance platforms. For this research the following components were provided:

- DLF algorithm and the assumptions made for its application [Cohen, 2005]
- Model used for the ATR system described in Section 4.2.1.2 [Gross, 2004a]
- Software for the actual ATR system [Sousa, 2004]
- Scenario information, including number of platforms, expected look ranges, and number of looks collected [Gross, 2004a]

The results and analysis in Section 5 are based on the provided items above and their integration with the data collected within the simulation. Significant contributions of this research are

- Conglomeration of over 500 GB of SAR data and sorting into usable sets
- MATLAB based simulation framework and implementation of the DLF algorithm
- Comparison of results based on simulated data and a modeled ATR system with results based on measured SAR data and an existing ATR system

This research does not endorse use of the DLF algorithm, nor is it intended to demonstrate the accuracy (or inaccuracy) of the ATR system. However, it does provide an unbiased analysis of the DLF algorithm in a real-time simulation based on conditions prescribed by AFRL.

II. Decision Level Fusion

This section provides an overview and derivation of the Decision Level Fusion (DLF) algorithm.

2.1 Overview

DLF works with the output of currently developed single sensor ATR algorithms, such as the RTMSTAR tool [Sousa, 2004]. The DLF algorithm provides a framework for combining the target identification (ID) of either the same or different sensor/algorithms over time and space. Each sensor performs preprocessing, including feature extraction and identity declaration, to develop a declared identity for the observed entity. Examples of decision-level fusion problems include:

- Threat-warning systems (TWS) onboard tactical aircraft that identify threats [Steinberg, 1987]
- Multiple-sensor target detection [Waltz, 1990]
- Data processing for robotic vision [Lus, 1988]

An advantage of DLF is that less detailed information is passed between multi-sensor platforms, resulting in low processing burden. The ATR ID and some historical performance knowledge are the only required inputs.

More complex and processing-intensive fusion methods, including Attribute Level Fusion or Feature Level Fusion, may provide better performance in the long run. These additional methods should be explored in the future, weighing the costs of performance

versus processing efficiency. However, as shown in Figure 1, DLF approaches require the least single sensor information for operation and therefore pass the least amount of data between sensors.

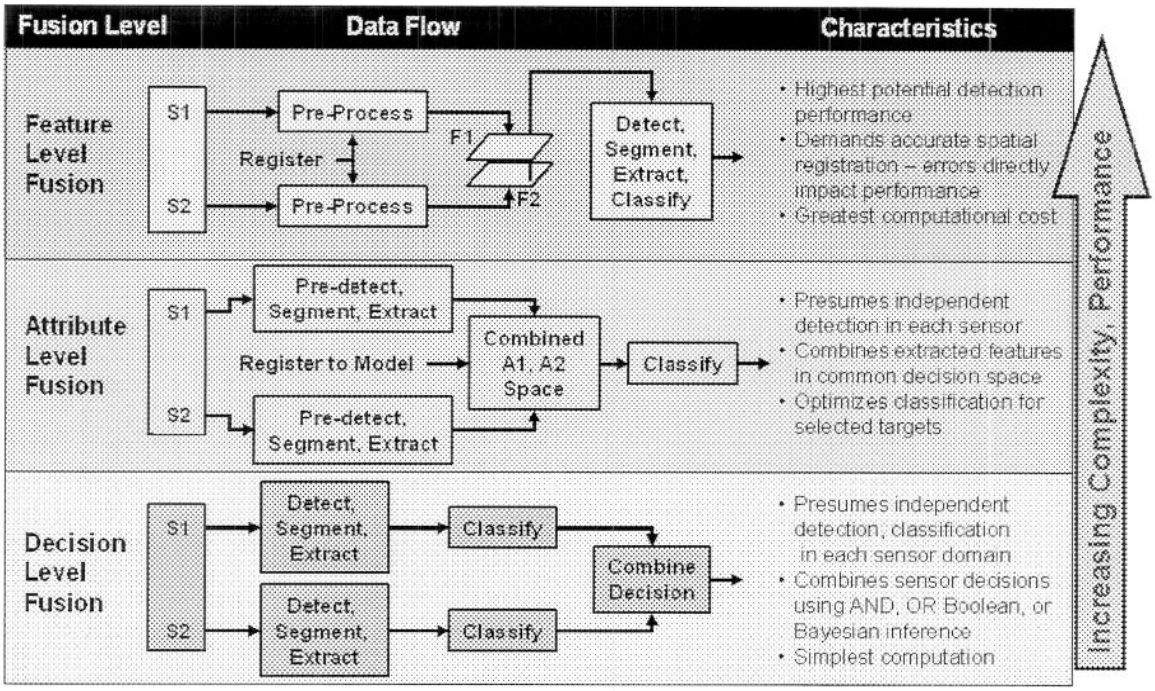


Figure 1. Comparison of different types of fusion levels [Gross, 2004a]

2.2 Derivation

2.2.1 Iterative Approach

The derivation of the DLF algorithm is, by design, quite simple. It assumes that a priori single-sensor performance estimates are available in the form of historical confusion matrices (HCMs). The DLF is designed to declare target T_i if this declaration is the most reliable decision that can be deduced from the single sensor.

The derivation that follows describes the most basic case, where two single sensor outputs are combined to declare a single target as the ID. The overall task of DLF is to estimate the probability

$$P(T_i \mid d_{1j}, d_{2k}), \text{ for all } i, j, \text{ and } k. \tag{1}$$

where d_{1j} indicates that sensor 1 declares target j, d_{2k} indicates that sensor 2 declares target k, and $P(T_i \mid d_{1j}, d_{2k})$ is the expression that the probability that target i is present given that these two declarations have occurred. By Bayes' Theorem, Expression (1) can be written

$$P(T_i|d_{1j}, d_{2k}) = \frac{P(d_{1j}, d_{2k}|T_i)P(T_i)}{\sum_i P(d_{1j}, d_{2k}|T_i)P(T_i)}, \text{ for all } i, j, \text{ and } k. \tag{2}$$

Assuming that the prior probabilities $P(T_i)$ are all equal allows Equation (2) to be simplified to

$$P(T_i|d_{1j}, d_{2k}) = \frac{P(d_{1j}, d_{2k}|T_i)}{\sum_i P(d_{1j}, d_{2k}|T_i)}, \text{ for all i, j, k.} \tag{3}$$

If it is assumed that sensor 1 and sensor 2 provide independent ID declarations, the joint probabilities in Equation (2) can be factored to yield

$$P(T_i \mid d_{1j}, d_{2k}) = \frac{P(d_{1j} \mid T_i)P(d_{2k} \mid T_i)}{\sum_i P(d_{1j} \mid T_i)P(d_{2k} \mid T_i)}, \text{ for all i, j, k.} \tag{4}$$

The right side of Equation (4) is now in terms of the entries of the single sensor HCM. Each index in this matrix represents a probability of ID for the given target (rows) versus the declared target (column), which allows definition of the declaration rule for the

DLF. Let DLF(j, k) represent the DLF declaration given that sensor 1 declares target j and sensor 2 declares target k, and define

$$DLF(j, k) = T_n, \tag{5}$$

$$n = arg\ max\{P(T_i \mid d_{1j}, d_{2k})\}.$$

The probability of target i is computed for all target declarations. The case that results in the highest probability is the DLF declaration as shown in Equation (5).

The performance of the so-defined DLF can be predicted by estimating, for each target of interest T_i, the frequency with which the single sensor algorithm makes the declarations j and k, and then assigning the appropriate declaration T_n from Equation (4), to the DLF algorithm. These frequencies can be directly estimated from the single-sensor confusion matrices (CMs) based on the independence assumption. In particular,

$$P\left(d_{1j}, d_{2k} \mid T_i\right) = P\left(d_{1j} \mid T_i\right) P\left(d_{2k} \mid T_i\right), \quad \text{for all } i, j, \text{ and } k. \tag{6}$$

Thus, for each T_i the frequency with which the DLF declares each T_j can be computed, and an estimate for the performance of the DLF in the form of a DLF confusion matrix can be derived [Cohen, 2005].

The assumption of two single sensors obviously makes the derivation simple. This research uses only the two sensor approach, which is accomplished by iterative application after each new look. Thus, the first two looks are fused to produce a single input into the DLF algorithm, the updated/fused look is then used in conjunction with a new look, and the process iterated until all looks are used or until the output produces

acceptable results for decisions on dispatching weapons. A second approach, the cumulative method is discussed in Section 2.2.2

2.2.2 Cumulative Approach

The approach described in Section 2.2.1 illustrates the simplest method, one in which each fused output acts as a single input. However, a more complicated and processing intensive method is described here. The cumulative approach allows for all the previous looks to be fused together when each new update is gathered. The derivation shown here, like that in Section 2.2.1, is theoretically simple, and attempts to take all biasness out of each new look.

Starting with Equation 4 from above, it is shown that as additional looks are cumulatively factored in, the conditional probabilities simply expand to result in Equation 7.

$$P(T_i \mid d_{1j}, d_{2k}, d_{3k}, d_{4k}, \ldots d_{nk}) = \frac{P(d_{1j} \mid T_i)P(d_{2k} \mid T_i)P(d_{3k} \mid T_i)P(d_{4k} \mid T_i)\ldots P(d_{nk} \mid T_i)}{\sum_i P(d_{1j} \mid T_i)P(d_{2k} \mid T_i)P(d_{3k} \mid T_i)P(d_{4k} \mid T_i)P(d_{nk} \mid T_i)}. \quad (7)$$

From Equation 7, it can be seen that as more looks are gathered the computation begins to grow very large. Although the output from Equation 7 may show optimal results, it has been indicated that the gain in performance does not outweigh the cost of processing efficiency.

2.3 Confusion Matrices

Confusion matrices are used to display which targets are declared for which templates. A template is defined as a stored image within an ATR system database that is the result of averaging multiple target images over an aspect range, providing an expected match to the target images available. Here targets, or truth, are displayed in rows and possible declarations, or templates, are shown at the heading of each column. The values for each index in the matrix correspond to the ratio of how many times the truth is declared the template out of the total number of data samples for a particular truth target.

The confusion matrices used in this research represent the historical performance of the ATR system. To compute them, data from each target is passed into the ATR system and compared against the available templates within the ATR system. Six total targets are used in this research. The ATR system contains templates on only four of these targets, leaving two targets without a template match. The two unmatched targets are the Not-In-Library targets described in Section 3.3

Confusion matrices often are used to describe the performance of an ATR system over the entire set of data. However, because it is well known that ATR prediction performance is aspect and depression angle dependent, for this research all data is divided into six aspect/depression regions – three at nine degrees depression and three at sixteen degrees depression. Each aspect region is based on the assumption of target symmetry, where it is assumed that targets are similar from the front and from the back, producing the head on/tail on region (HO), and that looking at the right side of a target is equivalent to looking at the left side, resulting in the broadside region (BS). These assumptions

leave all other targets grouped in the off-axis region (OA). Figure 2 illustrates the breakdown of aspect into the three regions.

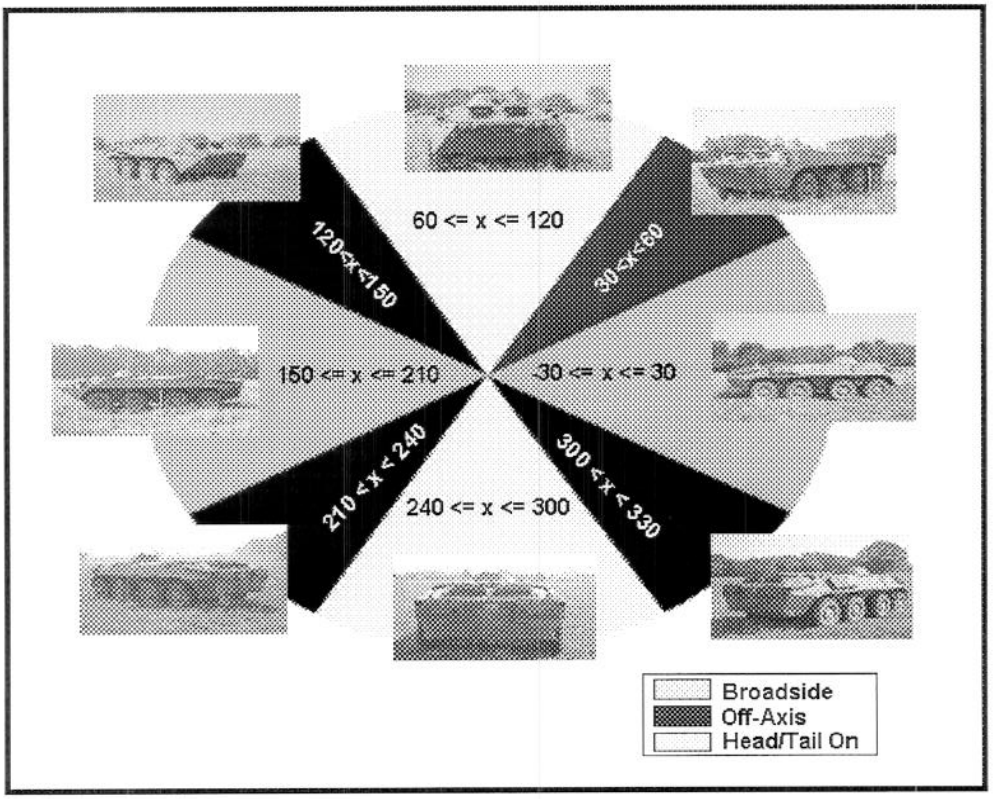


Figure 2. Aspect angle breakdown by region

A diagram illustrating the resulting six confusion matrices from the aspect and depression combinations is shown in Figure 3. The data used throughout this research is discussed in Section 3.1.

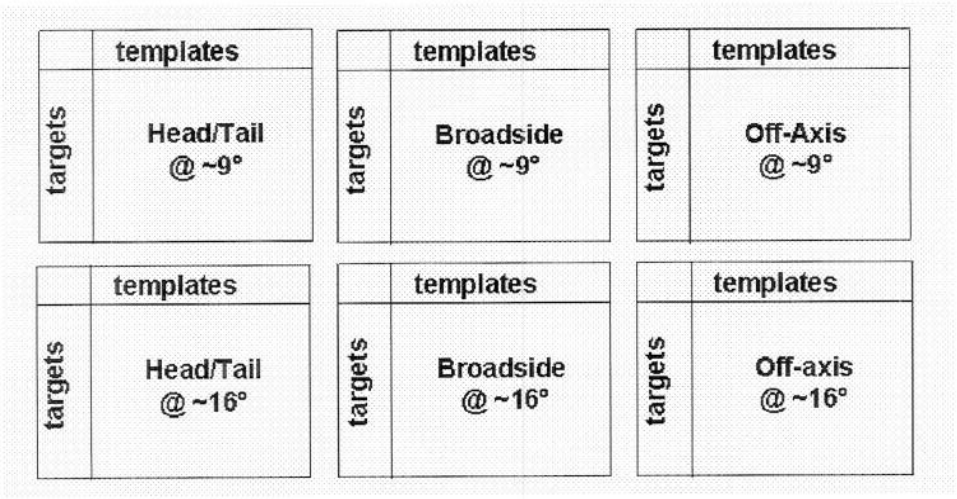


Figure 3. Confusion matrix illustration by aspect/depression regions

The measured depression angle is dependent on data collections, and the collections of interest are chosen based on their expected depression angle. The scenarios of interest to AFRL require multiple looks from multiple platforms. Therefore, data chosen at a high depression angle (15° - 17°) represents a near range look, and data chosen at a low depression angle (8° - 10°) represents a long range stand-off distance.

Once all the data is run through the ATR prediction system, the training-based confusion matrices (which act as historical performance indicators or HCMs) for each aspect/depression region are created as shown in Tables 1 – 6. These tables are the underlying performance indicators used throughout the DLF simulation based on measured data.

Table 1. Head/Tail On confusion matrix at 9° depression

HO 9	BRDM2	BTR70	T72	ZSU	NIL
BRDM2	0.5556	0.0000	0.0000	0.0046	0.4398
BTR70	0.2344	0.6641	0.0039	0.0469	0.0508
T72	0.0099	0.0000	0.9205	0.0596	0.0099
ZSU	0.0000	0.0186	0.2814	0.6744	0.0256
NIL	0.0000	0.4769	0.3333	0.0509	0.1389

Table 2. Off-Axis confusion matrix at 9° depression

OA 9	BRDM2	BTR70	T72	ZSU	NIL
BRDM2	0.6473	0.0145	0.0000	0.0000	0.3382
BTR70	0.0607	0.8237	0.0549	0.0376	0.0231
T72	0.0147	0.0189	0.9600	0.0063	0.0000
ZSU	0.0346	0.0035	0.1626	0.7820	0.0173
NIL	0.0000	0.7174	0.2500	0.0000	0.0326

Table 3. Broadside confusion matrix at 9° depression

BS 9	BRDM2	BTR70	T72	ZSU	NIL
BRDM2	0.8831	0.0584	0.0000	0.0065	0.0519
BTR70	0.0459	0.9083	0.0367	0.0000	0.0092
T72	0.0056	0.2000	0.7944	0.0000	0.0000
ZSU	0.1201	0.2205	0.1376	0.5087	0.0131
NIL	0.0000	0.9612	0.0243	0.0000	0.0146

Table 4. Head/Tail On confusion matrix at 16° depression

HO 16	BRDM2	BTR70	T72	ZSU	NIL
BRDM2	0.8711	0.0000	0.0000	0.0444	0.0844
BTR70	0.0417	0.9177	0.0094	0.0292	0.0021
T72	0.0000	0.0000	0.8713	0.1287	0.0000
ZSU	0.0221	0.0354	0.7257	0.2168	0.0000
NIL	0.1020	0.5204	0.0204	0.2075	0.1497

Table 5. Off-Axis confusion matrix at 16° depression

OA 16	BRDM2	BTR70	T72	ZSU	NIL
BRDM2	0.9553	0.0122	0.0000	0.0000	0.0325
BTR70	0.0192	0.9220	0.0289	0.0260	0.0038
T72	0.0000	0.0057	0.9583	0.0359	0.0000
ZSU	0.0772	0.0041	0.3089	0.6098	0.0000
NIL	0.0460	0.8161	0.1303	0.0077	0.0000

Table 6. Broadside confusion matrix at 16° depression

BS 16	BRDM2	BTR70	T72	ZSU	NIL
BRDM2	0.8717	0.1176	0.0000	0.0107	0.0000
BTR70	0.0012	0.9698	0.0157	0.0121	0.0012
T72	0.0000	0.0189	0.9726	0.0086	0.0000
ZSU	0.0214	0.0000	0.2834	0.6952	0.0000
NIL	0.0330	0.9104	0.0519	0.0000	0.0047

2.4 PID vs. Reliability

Probability of Identification (PID) requires that the truth be known. Testing a system for the average PID over a set of targets is acceptable as long as the results are used to form conclusions on the usefulness of the system. However, during simulation and demonstration of an ATR system another metric is needed, one that is independent of truth. Here this metric is referred to as a measure of reliability (REL). The difference between PID and REL can be seen from the sample confusion matrix shown in Figure 4.

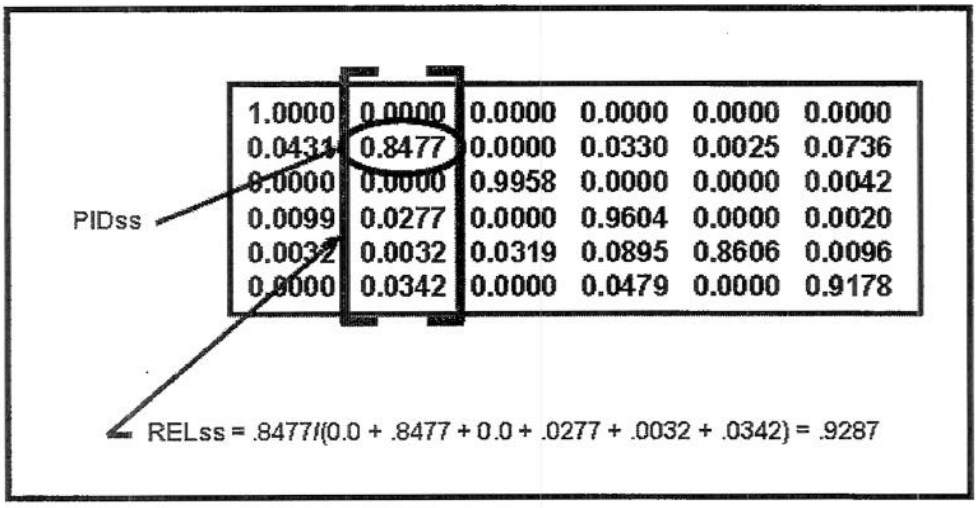


Figure 4. Definition of probability of ID (PID) vs. definition of reliability (REL)

In the confusion matrix of Figure 4 the rows represent the target set and the columns represent the templates used in the experiment, and, for example, the ATR system determines the target of interest to be in column 2. The REL for this single sensor look evaluates the (j, j) index of the confusion matrix, where j is the index of the single sensor declaration, and divides by the sum of the entire column to obtain a REL of .9287. Probability of ID is calculated by evaluating the (t, j) index of the confusion matrix, where t is the row corresponding to the truth target, and divides by the sum of the entire row. Here (as in all confusion matrices shown) the confusion matrix is normalized by row, and therefore the sum of the row is 1, resulting in a PID equal to the value of the (t, j) index itself.

In a perfect sensor/ATR system the PID and REL values are both equal to 1, in which case there are no errors, but values greater than 90% are generally considered

acceptable. The DLF algorithm described in this research attempts to maximize REL, sometimes at the sacrifice of PID.

III. Data and ATR System

This section considers the data sets, including test/template separation, not-in-library targets, and the ATR system.

3.1 Data Sets

Data used in this research was gathered based on available targets, type of collecting sensor, and relative depression angle. The initial goal was to use SAR data from as many identical targets as possible. To demonstrate the AFRL test plan, data must be collected for all targets in two distinct depression regions and must cover the full range of aspect angles. The process of gathering data and sorting available sets led to the initial determination of targets of interest. Similar vehicles were available in each of the data collections described, but only vehicles that were consistent in each of the sets were used. The targets of interest were the 2S1, BRDM-2, BTR-70, M978, T-72, and ZSU-23/4, and data from the following collections was obtained: DARPA Sponsored Moving and Stationary Target Acquisition and Recognition (MSTAR '95 and MSTAR '96), Dynamic Database – Multisensor All-Source Data 1998 (DDB-MAD '98), and AFRL sponsored Affordable Moving Surface Target Engagement – Moving Target Feature Phenomenology (AMSTE-MTFP). From this list, data from the two MSTAR collections provided the high depression region (15°-17°) and data from MTFP and MAD '98 provided the low depression region (8°-10°).

3.1.1 MSTAR

The MSTAR program has had three separate data collections. Data for this research was from both the MSTAR '95 and MSTAR '96 collections. All data was collected using the Sandia National Laboratory (SNL) STARLOS sensor operating at X-band at two separate sites. First in August 1995 Scud (surrogate) data was collected at Estancia, NM and had just one scene with two Scud surrogate targets in various articulation states. The second part of the collection was at Redstone Arsenal, AL, where data was collected over several days and had three scenes, each scene with 18 targets (12 target types) in various states of articulation, camouflage, and obscuration. In addition, clutter imagery was collected in the Redstone Arsenal area. Data from this collection was at both 15° depression and 17° depression, and targets of interest for this research from this collection were the T-72 and BTR-70.

A second MSTAR collection was in November 1996 at Eglin Air Force Base, FL, and contained baseline data on nine additional vehicles at a wide verity of squint angles and aspect angles. As in the first MSTAR collection, the SNL STARLOS sensor was used, and nearly all other operating conditions were the same. However, the addition of several vehicles made this collection important for this research. From the MSTAR '96 collection, data for the 2S1, BRDM-2, BTR-70, M978, T-72 and ZSU-23/4 were used in this research [Fitzgerald, 2005].

3.1.2 AMSTE-MTFP

The objective of the AMSTE-MTFP collection that took place at Eglin in 2000 was to obtain high range and Doppler resolution data and ground truth data on moving military targets to support the analysis and exploitation of target features for feature aided tracking. This collection used the Veridian/General Dynamics DCS sensor to collect the SAR data. All data was collected at full 360° aspect ranges and at a depression angle of 8° or 10°. Of the nine targets in the collection, data for the 2S1, T-72 and ZSU-23/4 were used in this research [Fitzgerald, 2005].

3.1.3 DDB –MAD '98

Similar to the AMSTE-MTFP collection, DDB-MAD '98 data was collected using the Veridian/General Dynamics DCS sensor at Eglin AFB, FL. All data from DDB-MAD '98 is in a depression range between 8° and 10° and contains data for over 40 targets. Many targets have different configurations and articulations of 25 vehicle types. As is the case throughout the data in each of the collections described, any target for a particular vehicle that is visually similar to other configurations for that vehicle is grouped to form the entire data set. Thus, although the serial number or configuration of the T-72 used in the DDB-MAD '98 collection differs from that of the T-72 used in the MSTAR collections, for this research they are grouped together. The scale of this collection allowed for data from all 6 targets to be used [Sensor, 2005].

Figure 5 provides images of all the targets used in the experiments and shows some of the subtle differences present in like targets with different serial numbers.

Figure 5. Target photo collage

Table 7 shows all the data used by aspect and depression regions. Although some targets are heavily populated with data, several targets are data limited in certain regions and therefore may not be sufficient in providing statistically significant information.

Table 7. Complete data summary of targets used for DLF verification

Confusion Matrix populating data								
Vehicle Name	Serial Number	broadside 9° chips	broadside 16° chips	head/tail on 9° chips	head/tail on 16° chips	off-axis 9° chips	off-axis 16° chips	total chips
BRDM2	E71, E72	177	211	257	261	243	286	1435
BTR70	C70,C71,C72,C73	128	856	289	995	394	1077	3739
T72	A07, A10, 132	381	612	547	722	505	729	3496
ZSU	D06, D08, D13	491	212	473	261	313	286	2036
M978	222rf, npogsn	56	212	99	263	53	279	962
2S1	B01	226	213	187	264	93	282	1265
Total Chips		1459	2316	1852	2766	1601	2939	12933

3.2 Test/Train Separation

To ensure that the algorithm is not trained on data it uses during testing, all available data is separated into two groups. The test group is used during the simulation and in the ATR system. Additionally, a train group is used to create the Training Confusion Matrices (TCM) and consists of the excess measured SAR data. The test group is selected from all data by choosing 100 data chips for each target in both the 9° depression region and the 16° depression regions. To gather the 100 chips from each target in each depression region, all data for each target and depression region is binned according to aspect angle. The first bin contains any data with an aspect of -1.8 degrees to 1.8 degrees, with each of the following bins covering 3.6°, resulting in 100 bins. Within each bin a random data chip is selected. The selected chips, 100 in all for each depression region, form the test group. Figure 6 shows a histogram of all available data of one target in the 16° depression region.

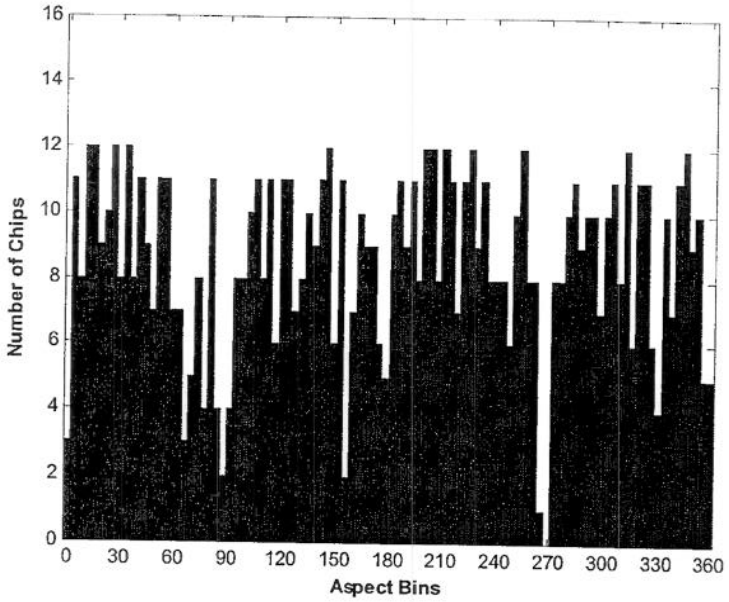


Figure 6. Histogram distribution of data samples

Initially the test group is created by selection of the data sample closest to the center of the bin, so that of all data available (nearly 13,000 chips) only 600 are used. Two issues with this test group are, first, cases where there is no data in a particular region (such as near 270° in Figure 6), and second, increasing the test set given that only about 5% of the data is used for testing.

To address the first issue the search for the random chip is simply expanded, which is necessary in several target depression combinations. The other solution is a Monte Carlo selection where the separation process is enhanced by allowing the chip selection process to be redone any number of times, thus enabling the use of a different test set for each test run.

3.3 *Not-In-Library Targets*

This section considers issues related to not-in-library targets.

3.3.1 In-Library Targets vs. Not-In-Library Targets

As described in Section 3.1, the targets used in this research include the 2S1, BRDM2, BTR-70, M978, T-72, and ZSU-23/4. Data from each of these targets are collected and processed through the ATR system. However, there are two categories of targets of the six listed. The BRDM2, BTR-70, T-72 and ZSU-23/4 are considered In-Library targets. These are so defined because the ATR system is trained on these targets and therefore has templates for matching. Not-In-Library (NIL) targets act as confusers to the ATR system, thus providing valuable information regarding how the ATR systems handles targets that it has no prior experience in identifying. The remaining two targets, 2S1 and M978, are NIL. The NIL targets do not have corresponding templates that the ATR system has been trained on, but are declared by the ATR system to be one of the In-Library targets due to the forced decision nature of the ATR system. The NIL targets used as inputs are considered here as "NILs by target". Another type of NIL target class, "NILs by threshold", is described in the following section.

3.3.2 Not-In-Library by Threshold

One of the outputs of the ATR system for each chip being processed is the score metric. Although the score metric can be changed to optimize different criteria, for this research the metric is restrained to the Classification Logic Test Statistic (CLTS). The CLTS is a linear sum of several other scoring options and consistently produces the best

overall PID compared to all other metrics when applied to the MSTAR data sets [Sousa, 2004]. The score is then compared against a set threshold to determine if the target declaration is retained or declared a NIL. If the score is below threshold, regardless of the declaration by the ATR system, the target is called a NIL.

Figure 7 shows the sample output of the ATR system. A score metric is computed for all available templates compared to the input data sample, where the highest score determines the declaration of the ATR system. In the case shown, if the threshold is set below .779 the test chip is declared a BRDM2, otherwise the post-processing step declares a NIL (by threshold) target.

```
target discrimination score      BRDM2 232.000 --   0.779
target discrimination score      BTR70  48.000 --   0.599
target discrimination score        2S1 228.000 --   0.513
target discrimination score    ZSU23-4  42.000 --   0.513
target discrimination score        T72  42.000 --   0.513

Stage Timings: 0.230 0.090 0.030 0.050 0.170 0.000 (0.570)
<< Metric labels complete. >>
<< Successfully processed 1 chip. >>
```

Figure 7. Sample output of the ATR system

The difference between the types of NIL targets can be summarized by examining relevance within a confusion matrix. In a standard confusion matrix the columns indicate the templates and each row represents a truth target. A row indicated by NIL refers to NIL by target since it is known that the truth target must have been either a 2S1 or M978.

However, the column marked by NIL indicates that the best template match had a score less than the determined threshold. An example confusion matrix is shown in Table 8.

Table 8. Sample confusion matrix

BS 16	BRDM2	BTR70	T72	ZSU	NIL
BRDM2	0.8768	0.1090	0.0047	0.0095	0.0000
BTR70	0.0012	0.9685	0.0163	0.0128	0.0012
T72	0.0000	0.0179	0.9739	0.0082	0.0000
ZSU	0.0189	0.0000	0.2736	0.7075	0.0000
NIL	0.0187	0.7944	0.1612	0.0000	0.0257

3.3.3 Threshold Determination

As described above, the threshold plays a pivotal role in NIL determination. If the threshold is set too low, too many of the NIL by target inputs result in false positives for the In-Library targets. Likewise, if the threshold is set too high, too many correctly identified targets are declared NIL because the score is not high enough. There are several approaches to setting the threshold, but for this research, where priority is on maximizing the probability of correct identification for the In-Library targets, the threshold is chosen to be two standard deviations below the mean of all available scores, resulting in a threshold of .6053. To verify that this value is reasonable, Figure 8 shows that near .61 a difference in targets correctly identified from targets misidentified as NIL begins monotonically increasing.

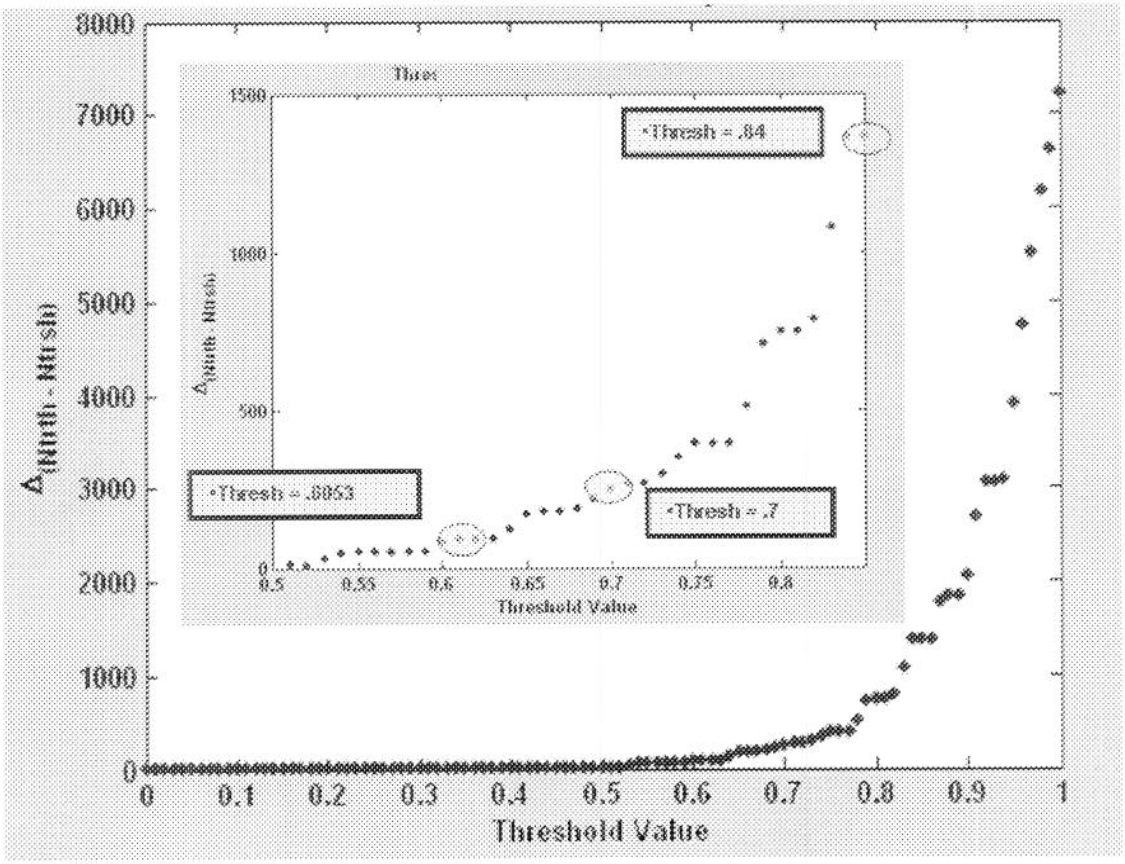
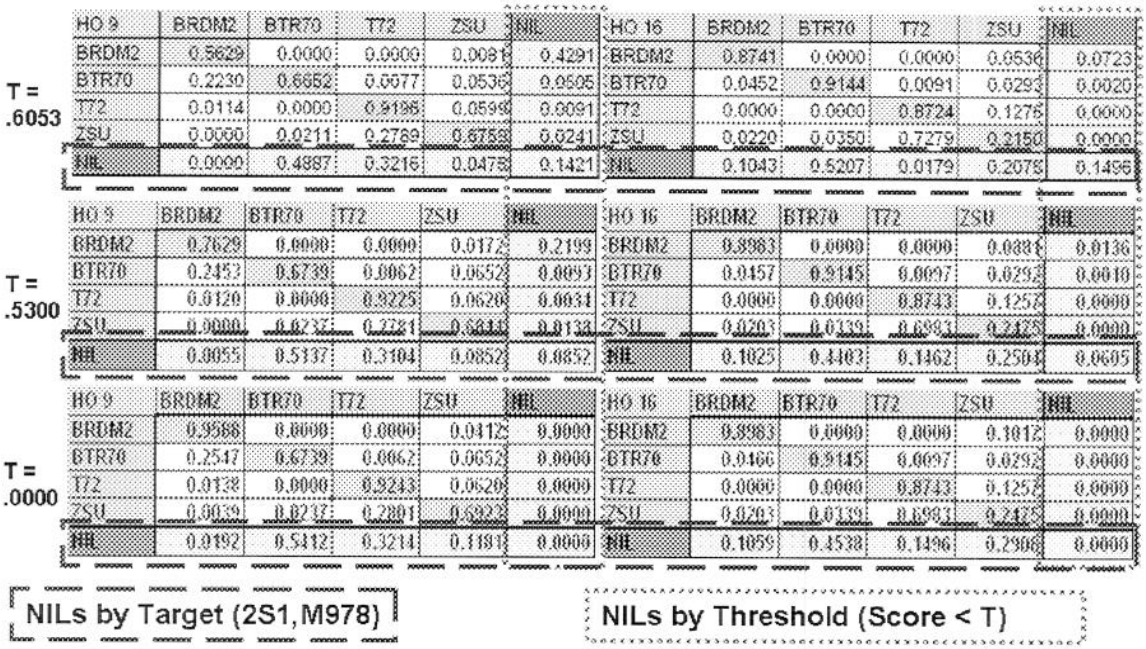


Figure 8. Graphical verification of statistically determined threshold

Figure 8 shows the difference in errors in determining NIL by threshold and by target. The y-axis is the absolute difference in the number of times a target is declared correctly but (due to a low score) is called a NIL and the number of times a NIL by target chip is not called a NIL (because of a high score).

Table 9 also shows the effects of changing the threshold values compared to the baseline, where the threshold is set to zero.

Table 9. Threshold effects on confusion matrices



T = .6053

HO 9	BRDM2	BTR70	T72	ZSU	NIL
BRDM2	0.5629	0.0000	0.0000	0.0081	0.4291
BTR70	0.2230	0.6652	0.0077	0.0536	0.0505
T72	0.0114	0.0000	0.9196	0.0599	0.0091
ZSU	0.0000	0.0211	0.2789	0.6759	0.0241
NIL	0.0000	0.4887	0.3216	0.0475	0.1421

HO 16	BRDM2	BTR70	T72	ZSU	NIL
BRDM2	0.8741	0.0000	0.0000	0.0536	0.0723
BTR70	0.0452	0.9144	0.0091	0.0293	0.0020
T72	0.0000	0.0000	0.8724	0.1276	0.0000
ZSU	0.0220	0.0350	0.7279	0.2150	0.0000
NIL	0.1043	0.5207	0.0179	0.2075	0.1496

T = .5300

HO 9	BRDM2	BTR70	T72	ZSU	NIL
BRDM2	0.7629	0.0000	0.0000	0.0172	0.2199
BTR70	0.2452	0.6739	0.0062	0.0652	0.0093
T72	0.0120	0.0000	0.9225	0.0620	0.0034
ZSU	0.0000	0.0237	0.2781	0.6814	0.0138
NIL	0.0055	0.5137	0.3104	0.0852	0.0852

HO 16	BRDM2	BTR70	T72	ZSU	NIL
BRDM2	0.8983	0.0000	0.0000	0.0881	0.0136
BTR70	0.0457	0.9145	0.0097	0.0293	0.0010
T72	0.0000	0.0000	0.8743	0.1257	0.0000
ZSU	0.0203	0.0139	0.6983	0.2475	0.0000
NIL	0.1025	0.4403	0.1462	0.2504	0.0605

T = .0000

HO 9	BRDM2	BTR70	T72	ZSU	NIL
BRDM2	0.9588	0.0000	0.0000	0.0412	0.0000
BTR70	0.2547	0.6739	0.0062	0.0652	0.0000
T72	0.0138	0.0000	0.9243	0.0620	0.0000
ZSU	0.0039	0.0217	0.2801	0.6923	0.0000
NIL	0.0192	0.5412	0.3214	0.1181	0.0000

HO 16	BRDM2	BTR70	T72	ZSU	NIL
BRDM2	0.8983	0.0000	0.0000	0.1012	0.0000
BTR70	0.0466	0.9145	0.0097	0.0292	0.0000
T72	0.0000	0.0000	0.8743	0.1257	0.0000
ZSU	0.0203	0.0139	0.6983	0.2475	0.0000
NIL	0.1059	0.4538	0.1496	0.2906	0.0000

NILs by Target (2S1, M978)

NILs by Threshold (Score < T)

Table 9 displays confusion matrices over the same aspect/depression regions at several different threshold values. The baseline case, with T = 0.000 illustrates the forced decision nature of the ATR system. As the threshold is increased many calls declaring targets BTR-70 and ZSU-23/4 are pushed to the NIL column. The other extreme is at T = 1.000 (not shown), which would result in all targets being placed in the NIL column.

3.4 ATR System

This section describes the ATR system, including its setup and its outputs.

3.4.1 Description

The ATR system used for this research is a model-based system that aims to achieve accurate and robust performance in unconstrained scenarios. The system is designed to locate and recognize military targets in SAR imagery. The scene conditions are realistic

situations characterized by arbitrary viewing angles, variable radar squints, variable radar depression angles, modest obscuration levels in both flat and hilly terrain, and multiple target configurations and articulations. Figure 9 gives a general overview of the representative ATR system.

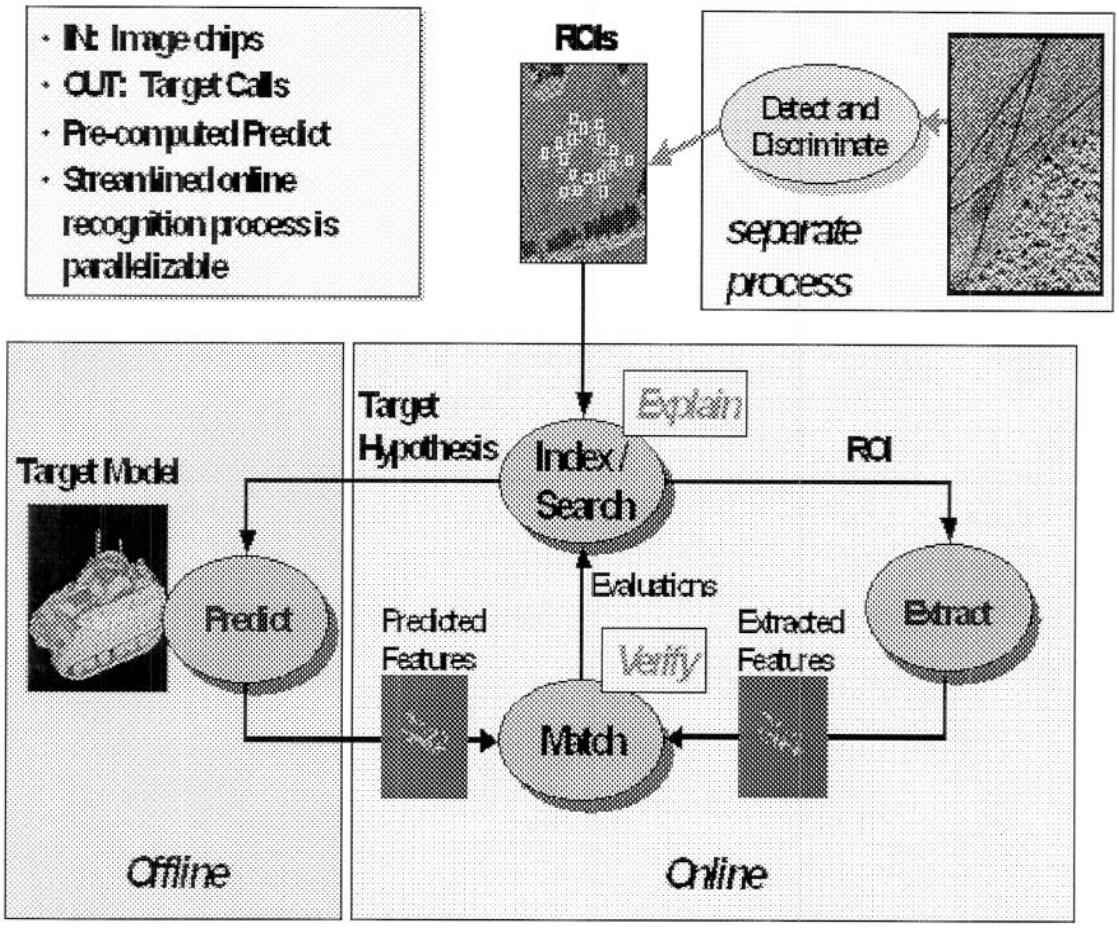


Figure 9. General overview of characteristics of the ATR system [Sousa, 2004]

A central tenet of the ATR system is that the combinatorically large number of targets, sensors, and backgrounds found in realistic scenarios precludes the use of algorithms that rely on a priori measured images. Statistically significant amounts of data

may not be available for template based ATR training. The system architecture is thus designed as a hypothesis and test paradigm in which scene hypothesis are generated and refined based on the degree to which the viewed imagery supports the model based hypotheses [Sousa, 2004].

3.4.2 Processing Setup

The ATR system was developed and tested on UNIX based computer systems. Therefore, the first task arranged for an ability to pass data back and forth between a Windows based workstation and the UNIX based ATR system, which was accomplished by setting up a networked file transfer protocol (FTP) between the two computers. The data was stored in a directory structure on a workstation by target followed by subdirectories for depression region and then another subdirectory that isolated aspect region. Figure 10 illustrates the directory structure.

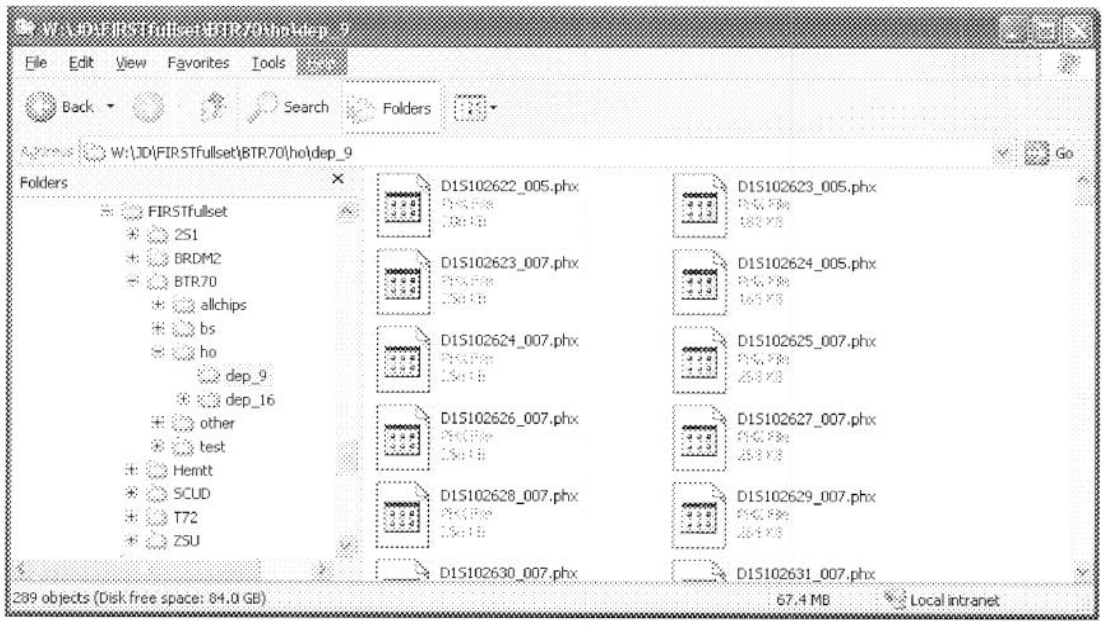


Figure 10. Directory structure setup for measured data

With the directory structure in place, scripts were written within UNIX shell scripts to process all chips for each target at the specified depression and aspect regions.

The most efficient way to proceed with the experiment is to process all available data and save the results, which allows for the data set to change and also enables an efficient process. The processing of all chips takes over 10 hours. If all available data is processed, all the code needed to run an experiment can be written entirely in MATLAB. With everything running from the same software, the simulation has a much more realistic flow and can be done in one simple execution step.

Before the data is applied to the ATR system, several issues had to be addressed. Problems arose when either the DDB-MAD 98 or AMSTE-MTFP data was passed to the ATR system. Data collected under DDB-MAD 98 and AMSTE-MTFP was collected in the form of Video Phase-History by the DCS sensor and processed into National Image Transmission Format (NITF) [Fitzgerald, 2005]. This data format first needed conversion to Phoenix format (PHX), in which header information is stored as text before the image data in a binary format, for the ATR system to read the data. Initially this was a simple task that was accomplished using a conversion tool developed by the AFRL Sensor Data Management System (SDMS) group [Fitzgerald, 2005]. The conversion updated the header into the Phoenix style, but there were differences in field names and units that needed resolution. For MSTAR and AMSTE-MTFP data results were very good. However, it was determined that the current data from the DDB-MAD 98 collection had a convention difference in determining how the platform collected the data from the MSTAR and AMSTE-MTFP. By rotating the data 90° the chips were finally

ready to be processed by the ATR system to produce the confusion matrices used throughout this research and shown in Section 2.2.1.

3.4.3 Outputs

The outputs from the ATR system include the truth target, the declared target, the actual pose angle (aspect), the estimated pose angle, and the metric score. After all the data is run through the ATR system a text file is created, where each row of text represents the results of one chip. This master text file is then parsed into individual files for each line. These output result files are then stored in a directory structure similar to the raw data (separated by target/depression/aspect). For each target the filenames representing each individual chip reflect the aspect angle of the actual measured SAR chip. With the data stored in this fashion the simulation can easily query the folder of the individual target for results at a given aspect and depression.

IV. Experimental Setup

This section describes the experiment scenario along with the simulation framework and the DLF flow.

4.1 Scenario Description

A simple case is considered where two fighters are flying in tandem. Each platform is offset about the target, with headings such that the target is centered. When the platforms are within 50 km the SAR sensors are turned on and the first look is gathered. While this look is processed the platform diverges by 15° from its current path, and when the target is within 30 km a second look is gathered from each platform, resulting in four total looks – two from each platform. Figure 11 describes the process.

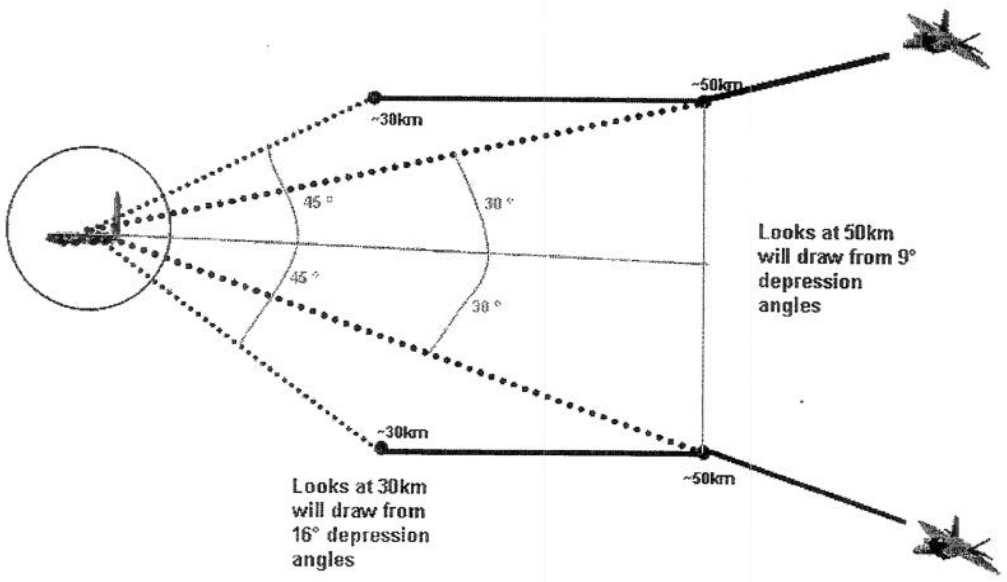


Figure 11. Scenario description

The divergence after the first look allows the platforms to gather target SAR data at a significant aspect change (a change of greater than 10°), which is important because performance may depend on aspect angle. By gathering data at different aspects, a wider range of search criteria for the ATR system is examined, and independence between looks is guaranteed. For this research data collected under AMSTE-MTFP and DDB-MAD 98 collections (8°-10°) is used to represent the 50 km looks, and data from the MSTAR collections (15°-17°) is used to represent looks gathered at 30 km.

4.2 Simulation Flow

This section describes the experimental cases. An experiment can be run as a single case, where the test chips are evaluated by initializing the target heading incrementally, or through a Monte Carlo setup, where multiple test sets are used. Here 'iteration' is

defined as a four-look DLF process, where the target and the corresponding heading are kept constant. Likewise, a 'run' is defined as the process of cycling through each target at each heading for a set of test data. To verify the DLF algorithm two types of experiments are performed. First is the simulated experiment, which uses simulated data and an ATR model used in the initial DLF development. The second experiment uses a real ATR system with measured SAR data. In both simulated and measured cases, the framework remains the same and is shown in Figure 12.

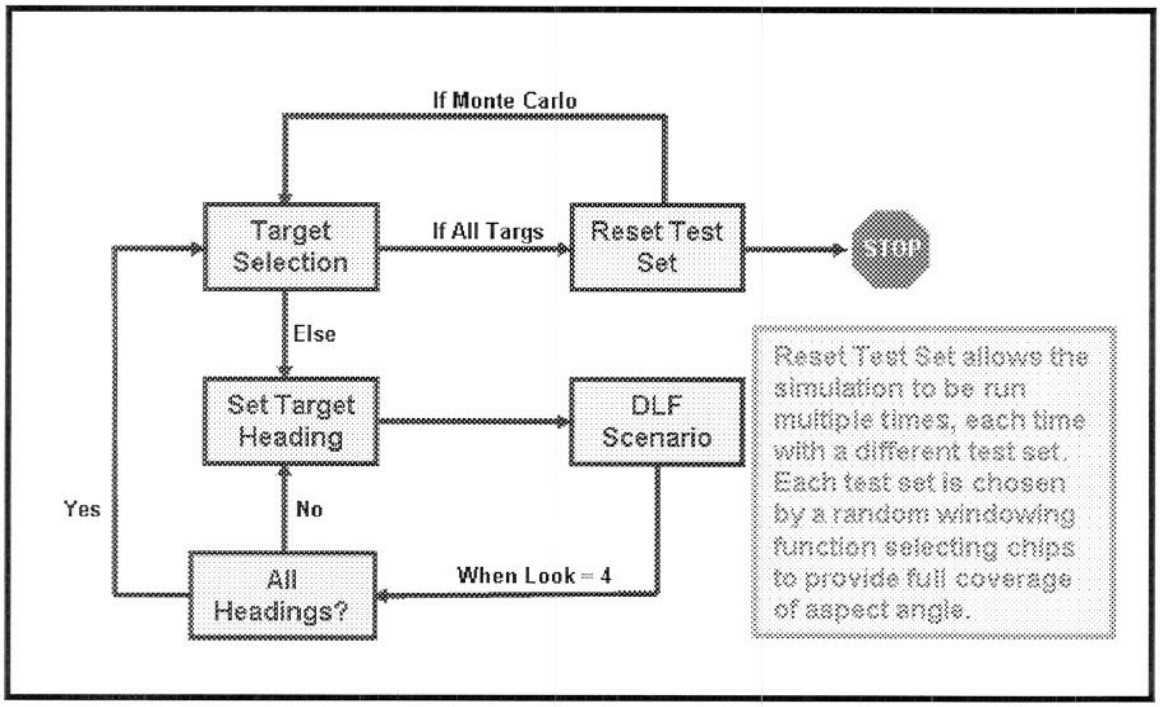


Figure 12. Simulation framework flowchart

4.2.1 Simulated Experiment

For a simulated experiment no measured data is used. The data comes directly from the HCMs described in Section 4.2.1.1, and the ATR system is a simple probabilistic

model that bases target declaration on the probabilities of the confusion matrices and compares them to a random variable between zero and one. The ATR system model used in the simulated experiment is described in detail in Section 4.2.1.2. The only difference between the simulated experiment and the real experiment is that the modeled ATR system is exchanged for a real ATR system and estimated data is replaced by measured SAR data.

4.2.1.1 Confusion Matrix Estimation

Key to DLF is reliable estimation of sensor/ATR performance. As described in Section 2.2, DLF is based on confusion matrix results. The confusion matrices used in the simulated testing of the DLF algorithm are based on historical performance over SAR data for a general SAR sensor/ATR system. Examples of the simulated confusion matrices used are shown in Tables 10-15. [Gross, 2004b]

Table 10. Head/tail 9° historical confusion matrix

HT 9°	BRDM2	BTR70	T72	ZSU	NIL
BRDM2	0.8500	0.0000	0.0000	0.0000	0.1500
BTR70	0.0000	0.8500	0.0000	0.0000	0.1500
T72	0.0000	0.0000	0.9500	0.0000	0.0500
ZSU	0.0000	0.0000	0.2000	0.7500	0.0500
NIL	0.0000	0.4750	0.0000	0.0000	0.5250

Table 11. Broadside 9° historical confusion matrix

BS 9°	BRDM2	BTR70	T72	ZSU	NIL
BRDM2	0.8000	0.0000	0.0000	0.0000	0.2000
BTR70	0.1500	0.6500	0.0000	0.0000	0.2000
T72	0.0000	0.0000	0.8000	0.1000	0.1000
ZSU	0.0000	0.0000	0.3000	0.5500	0.1500
NIL	0.0250	0.1500	0.0000	0.0500	0.7750

Table 12. Off-axis 9° historical confusion matrix

OA 9°	BRDM2	BTR70	T72	ZSU	NIL
BRDM2	0.9000	0.0500	0.0000	0.0000	0.0500
BTR70	0.0700	0.8500	0.0000	0.0000	0.0800
T72	0.0000	0.0550	0.8950	0.0000	0.0500
ZSU	0.0000	0.0000	0.1000	0.8000	0.1000
NIL	0.0250	0.5000	0.0000	0.0000	0.4750

Table 13. Head/tail 16° historical confusion matrix

HT 16°	BRDM2	BTR70	T72	ZSU	NIL
BRDM2	0.9110	0.0000	0.0000	0.0000	0.0890
BTR70	0.0000	0.9060	0.0000	0.0000	0.0940
T72	0.0000	0.0000	0.9670	0.0000	0.0330
ZSU	0.0000	0.0000	0.1780	0.8220	0.0000
NIL	0.0000	0.5900	0.0000	0.0000	0.4100

Table 14. Broadside 16° historical confusion matrix

BS 16°	BRDM2	BTR70	T72	ZSU	NIL
BRDM2	0.8980	0.0000	0.0000	0.0000	0.1020
BTR70	0.1780	0.6830	0.0000	0.0230	0.1160
T72	0.0000	0.0000	0.8180	0.1820	0.0000
ZSU	0.0000	0.0000	0.3270	0.5910	0.0820
NIL	0.0610	0.0710	0.0000	0.1330	0.7350

Table 15. Off-axis 16° historical confusion matrix

OA 16°	BRDM2	BTR70	T72	ZSU	NIL
BRDM2	0.9340	0.0330	0.0000	0.0000	0.0330
BTR70	0.0580	0.8720	0.0000	0.0000	0.0700
T72	0.0000	0.0550	0.9240	0.0110	0.0100
ZSU	0.0000	0.0000	0.1000	0.9000	0.0000
NIL	0.0320	0.5920	0.0000	0.0160	0.3600

The tables show the performance of the ATR system when used with SAR data. One difference between the estimated confusion matrices here and the confusion matrices of the measured data (Tables 1-6) is that the latter show performance based solely on the

data available while the CMs shown in tables 10-15 are estimates of the actual sensor performance without regard to the characteristics of the measured data.

4.2.1.2 ATR System Modeling

The model used for the ATR system is a probability-based predictor with a random threshold. When used with historical confusion matrices, which also are probability based, all needed information is available. The first step is to randomly define a minimum probability threshold, which allows for the possibility of mis-identification and the inherent presence of error. Once the threshold is set, the model examines the row of the confusion matrix corresponding to the selected target truth. The process incrementally moves across the row, summing total values. When the sum exceeds the threshold, a declaration is made corresponding to the current column [Gross, 2004a]. Figure 13 illustrates an example case processed by the model ATR system and used in the simulated experiment.

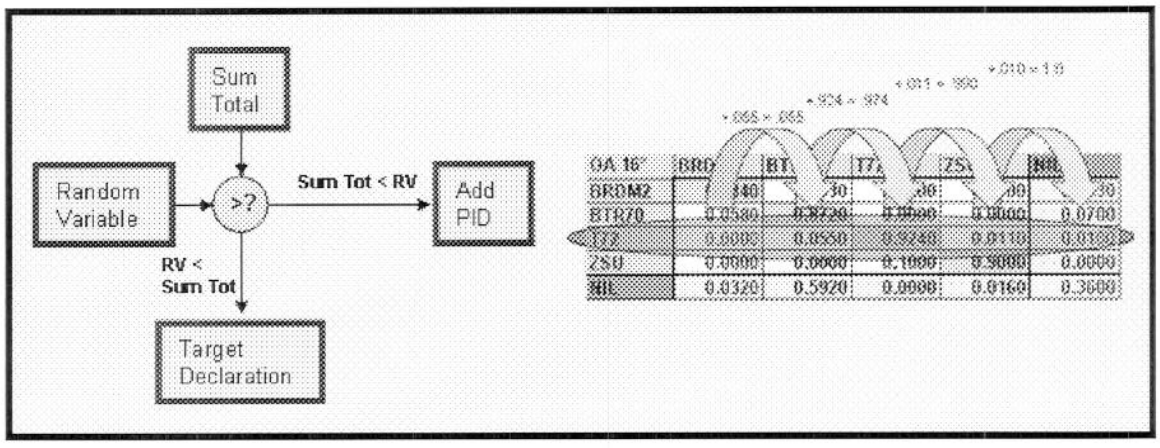


Figure 13. Modeled ATR system flow and example case

This illustration indicates that the model is likely very good if the historical confusion matrices are perfect. Since real world characteristics such as clutter and signal to noise ratio cause unforeseen error in image based ATR systems, the model does not perform as an actual system. Verification using measured data and a real ATR system is needed.

4.2.2 Measured Data Experiment

The framework described at the beginning of this section and the flow outlined in Figure 12 remains the same when measured SAR chips are used as the input data and replace the simulated confusion matrices. The ATR system is also altered by replacing the modeled system with a fielded ATR system. In the following sections the process is described for measured data and a real ATR system.

4.2.2.1 Single Test Set

For a single test set, the geometric scenario is run for each target at each angle, which is accomplished by simply rotating the heading of the target by 3.6 degrees for each iteration. Due to the geometry of the scenario, when the target heading is kept constant throughout one iteration, each look (of the four total) is at a different aspect angle. For the long range looks the aspects angles are offset by 60°. For the second look from each platform the aspect angle changes by 15° and the overall offset of the near range looks is 90°.

A single test set experiment is completed when each of the 5 targets (BRDM-2, BTR-70, T-72, ZSU-23/4, NIL) is initialized at the heading increments, each with the

same test set, which results in 500 iterations, or one run. However, performance and validation should not be based on such a limited sample size (500 chips) – especially if it is heavily dependent on the test data. Thus an expansion of the data set into the Monte Carlo scenario is made.

4.2.2.2 Monte Carlo Scenario

To overcome instances where a single test set is chosen such that the confusion matrix results for each test set are dissimilar to that of the training set, a Monte Carlo scenario is used. A simulation run in Monte Carlo mode takes over 10 hours to run the entire scenario if 100 different test sets are considered, but a better representation of the results is obtained. The Monte Carlo version of the simulation chooses a new test set for each run. Once the test set is chosen as described in Section 3.2, aspect/depression region training confusion matrices (TCM) are calculated, and these TCMs are then used in the DLF algorithm. After the completion of each run the test set is reset and the process starts over.

The Monte Carlo method is the method of choice and generates the results of Section 5. Variations in case by case results are obvious, but by combining results of many runs (100), most of the outlier data points are masked by the law of averages. The results reflect a more general case than if the randomly drawn test set does not coincide with the historical performance of the ATR system.

4.3 DLF Flow

The DLF portion of the simulation is performed within each iteration. The simulation processes the looks in sequence, where the first fighter gets "Look 1". The first step is to calculate the aspect angle from the platform to the target. Once the aspect angle is calculated, the data chip to be read from the results database is known. Each data file contains information that is saved during processing in the ATR system. From each file the truth target, the target declared by the system, the actual aspect angle that is the basis, the aspect angle estimated by the ATR system, and the scoring metric value are read. From this information the declared target determines the target ID, the estimated aspect angle for the lookup determines which TCM is used, and the scoring metric determines if the target is called NIL. The TCM that contains the estimated pose of the measured data is then stored as the single sensor confusion matrix. However, for this first look the single sensor results also represent the initial DLF result. Therefore, the ATR declaration for the single sensor look is identical to the initial DLF identification.

Once the first look is processed, the second platform has a first look at the target. A process similar to that described above allows the second platform to obtain single sensor results. In this case and those that follow for look 3 (from the first platform again) and look 4 (from the second platform), the single sensor results are fused with the previous DLF results using the method described in Section 2.2 to produce an updated DLF output. After the completion of look 4, the resulting DLF identification is used as the final decision. The flowchart of Figure 14 describes a single iteration using the DLF algorithm.

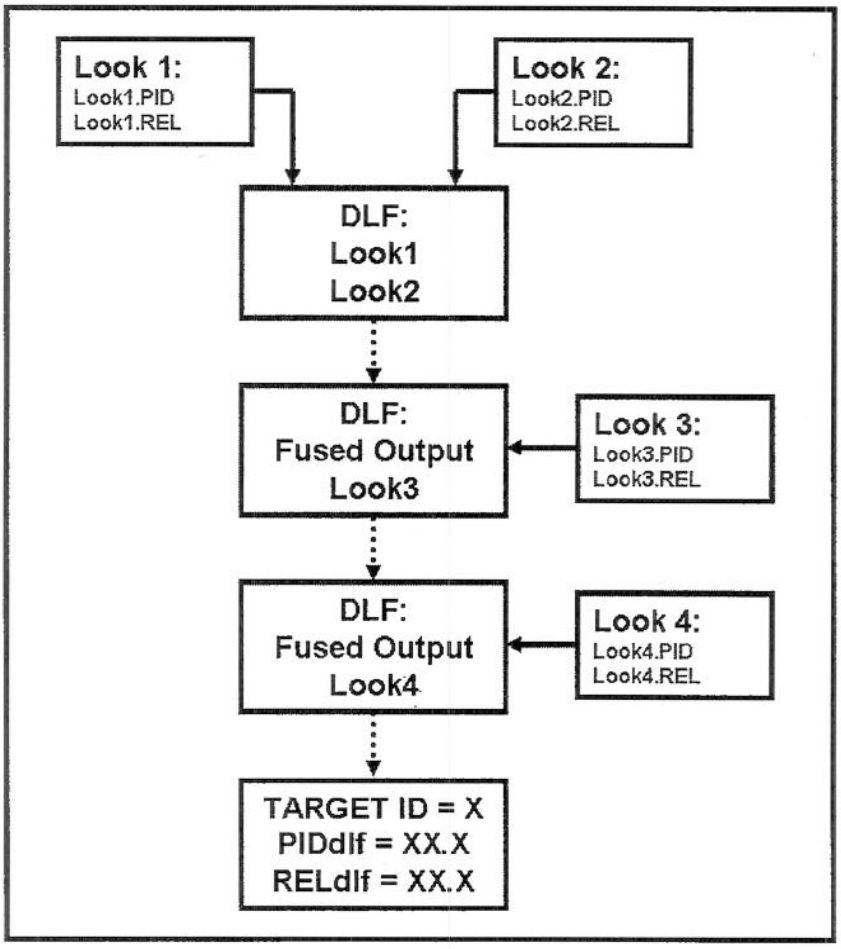


Figure 14. DLF flowchart

V. Results

The goal of any ATR system is to correctly identify targets as quickly as possible, as accurately as possible, and from as far away as possible. Improved sensor technology and on-board computing allows better and more complicated systems to be placed on many kinds of platforms. Advancements in fusion, specifically less complicated approaches such as DLF, can only enhance an already proven ATR system. However, as shown below, the benefits of DLF on simulated data may be slightly optimistic compared with the performance found using a real ATR system and measured data.

5.1 Simulated Results

This section shows examples of results from a simulated ATR system used in coordination with estimated data, and it provides baseline observations for further research. The results show that DLF yields significant improvement over a single sensor working on each platform. Figures 15 – 16 show that DLF provides a median PID and REL for all targets near 95% or greater.

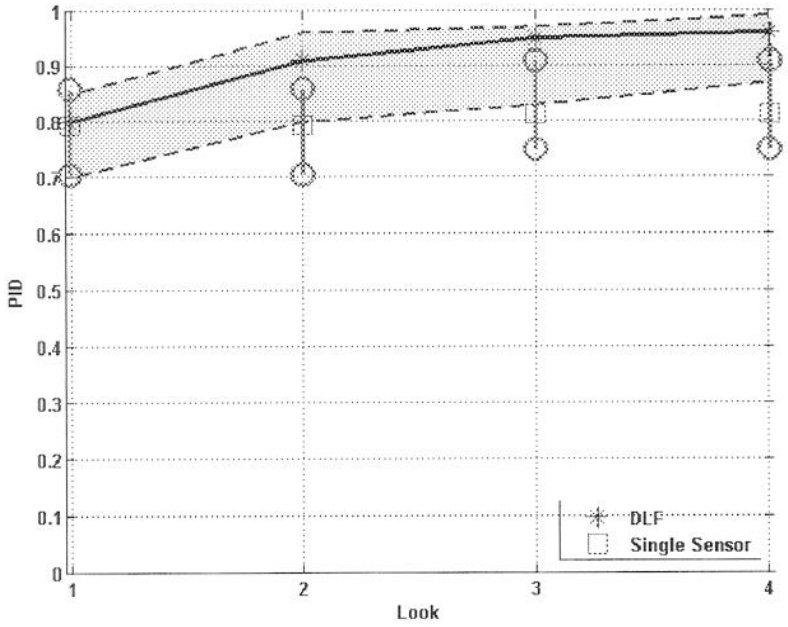


Figure 15. Median PID values with percentile ranges over all DLF looks - Simulated

Each DLF look is made up of fused single sensor looks. However, since each sensor performs differently, it is important to show the single sensor results separated by range and depression angle. Thus in the single sensor comparison there are two data points. The first data point, which is the same for both look 1 and look 2, represents single sensor looks from a further range and uses data collected at the lower depression angle. The other single sensor value is the averaged PID (or REL in Figure 16) over the last two looks, which are at a closer range and higher depression.

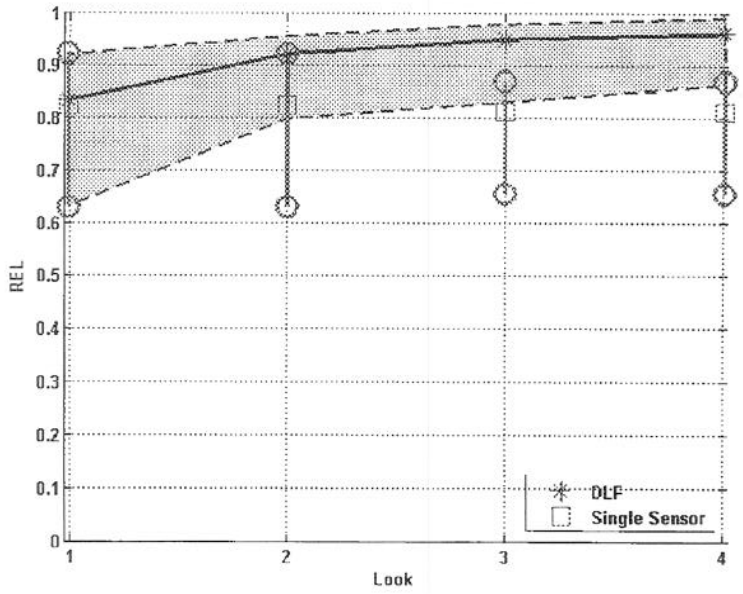


Figure 16. Median reliability with percentile ranges over all DLF looks – Simulated

In the typical plots of Figures 15 and 16, the shaded area represents the range of values in the 25th and 75th percentile. The vertical lines at each look represent the single sensor results. Where the box is the median value and the circles show the 25th and 75th percentile points. A significant performance improvement is shown over the single sensor median values. Not only does the median value continue to increase as additional looks are gathered, but the size of the middle fiftieth percent of the data grows considerably smaller. Table 16 compares the improvements made through the DLF process over single sensor looks.

Table 16. DLF vs. single sensor over all looks – Theoretical

		Look 1	Look 2	Look 3	Look 4
PID	DLF	0.8	0.91	0.95	0.96
	SS	0.795	0.795	0.81	0.81
	Improvement %	0.62%	12.64%	14.74%	15.63%
REL	DLF	0.82	0.91	0.95	0.96
	SS	0.81	0.81	0.805	0.805
	Improvement %	1.22%	10.99%	15.26%	16.15%

From the table it can be seen that simply adding a second look and fusing it with the first look yields significant improvements in capabilities. By using all four looks the algorithm is able to increase both the probability of identification and the reliability in the declaration to over 95%. Also shown in the table is that the improvement over the single sensor performance is over 15% in both PID and REL

Figures 17-21 show how the individual targets, when isolated, perform in the DLF process.

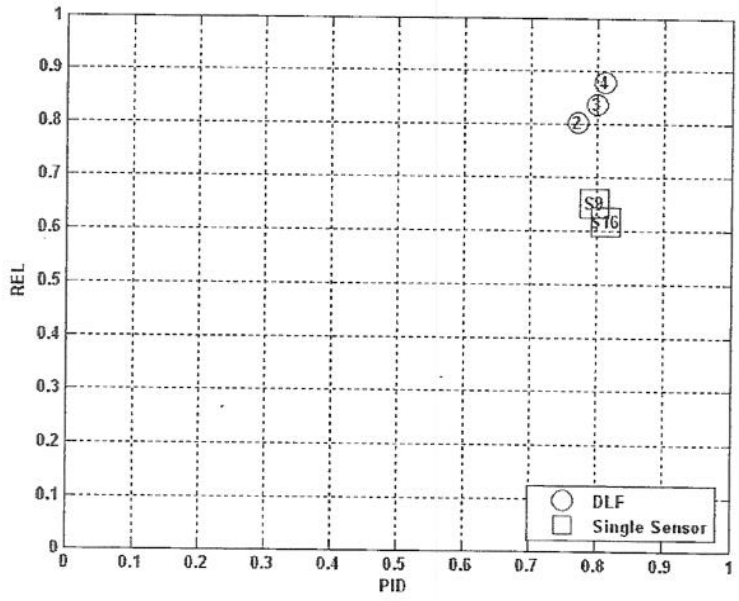


Figure 18. PID vs. reliability for BTR-70 - Simulated

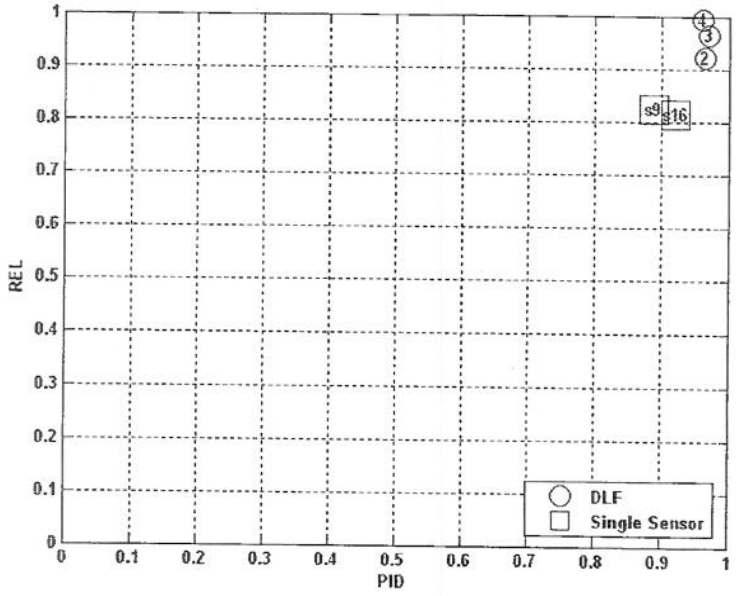


Figure 19. PID vs. reliability for T-72 - Simulated

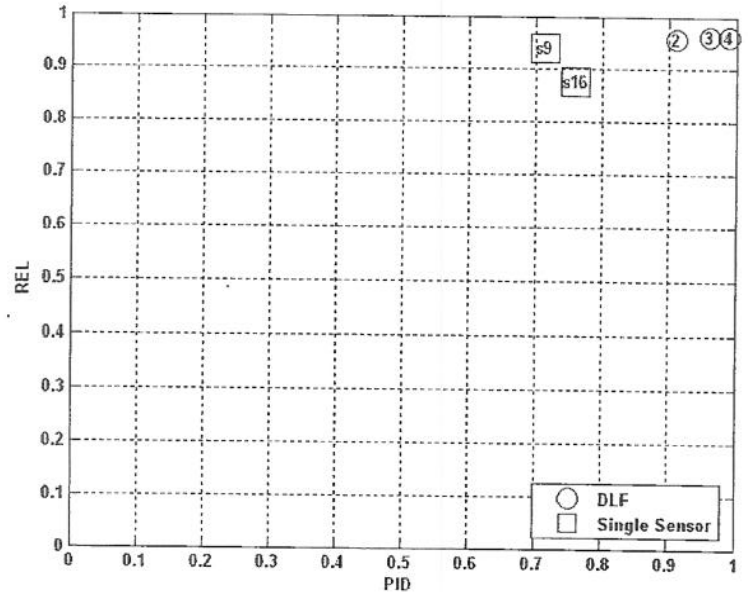


Figure 20. PID vs. reliability for ZSU-23/4 – Simulated

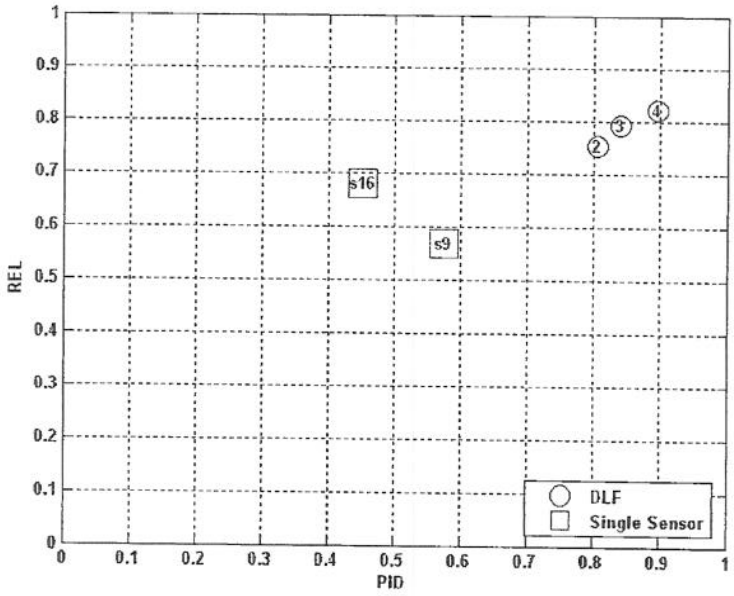


Figure 21. PID vs. reliability for NIL - Simulated

These plots show PID vs. REL for each target. As the system gathers additional looks and DLF is performed, both the PID and REL move on the plot toward the upper right corner. These results depend on expected performance as characterized in the historical confusion matrices or training confusion matrices. For the BTR-70 and the NIL targets the results are slightly below the other targets because the expected PID based on the HCMs (Tables 10-15) used is also less.

The results shown here are encouraging; however, there are still many robustness and accuracy concerns. The next step is to compare the simulated results using the model ATR system with results from a proven ATR system used with measured data.

5.2 Measured Data Results

Once the modeled ATR system is replaced with the proven ATR system and the expected historical confusion matrices are exchanged with calculated training confusion matrices from the actual ATR system, results can be compared. Similar to the plots shown in the previous section, the measured data indicates strong trends that endorse the use of DLF in ATR prediction. However, the measured data and real ATR system do not produce results as conclusive as those using simulated data.

Figure 22 shows that the median value for the PID increases as the looks increase.

matrices from the actual ATR system, results can be compared. Similar to the plots shown in the previous section, the measured data indicates strong trends that endorse the use of DLF in ATR prediction. However, the measured data and real ATR system do not produce results as conclusive as those using simulated data.

Figure 22 shows that the median value for the PID increases as the looks increase.

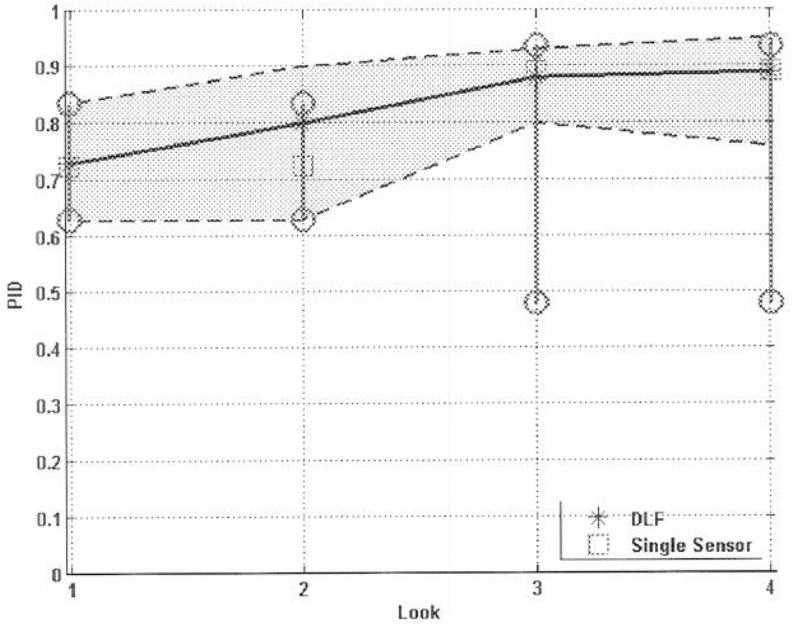


Figure 22. Median PID values with percentile ranges over all DLF looks – Measured

When comparing with Figure 16 (the PID vs. looks plot based on simulated data), Figure 22 shows that in both cases the improvement from the first to the second looks increases by nearly 10%, with the simulated results starting from a higher PID.

The REL vs. look plot shown in Figure 23 again shows that as more looks are input to the DLF algorithm, reliability improves. A slightly more dramatic improvement than

in Figure 16 (simulated REL vs. Looks plot) from look 1 to look 2 is apparent, with the final two looks adding nearly a 3% increase in performance over the preceding look.

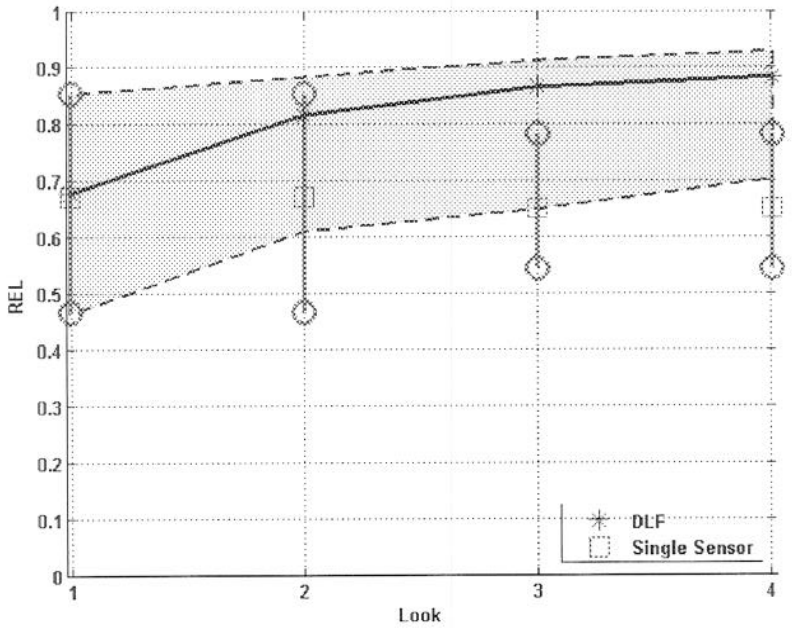


Figure 23. Median reliability values with percentile ranges over all DLF looks - Measured

Figures 24-28 show how each target performed for the measured data.

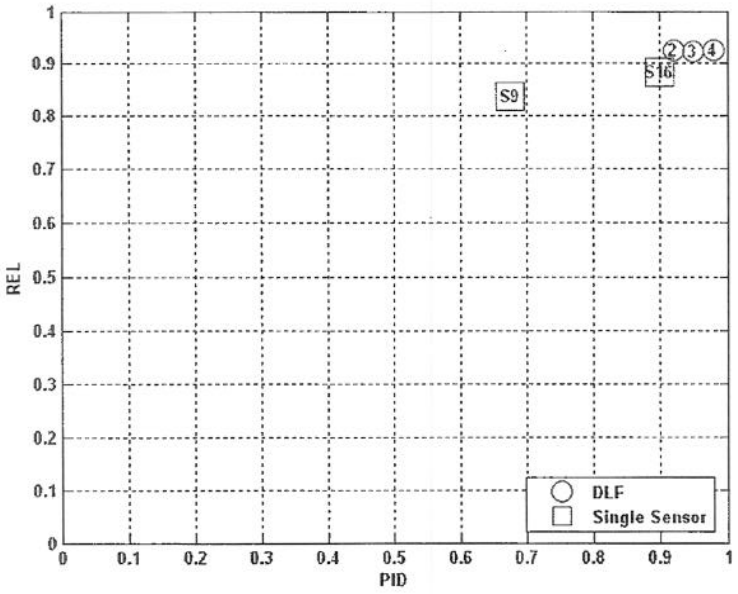


Figure 24. PID vs. reliability for BRDM2 – Measured

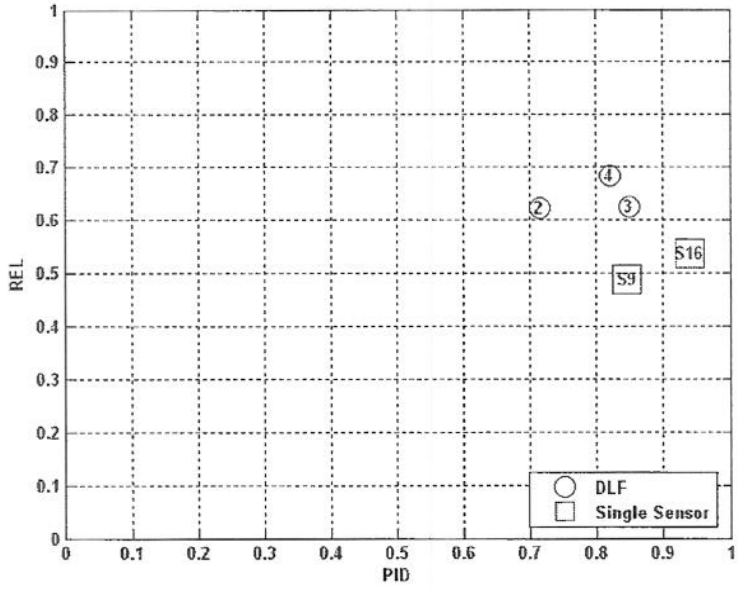


Figure 25. PID vs. reliability for BTR-70 - Measured

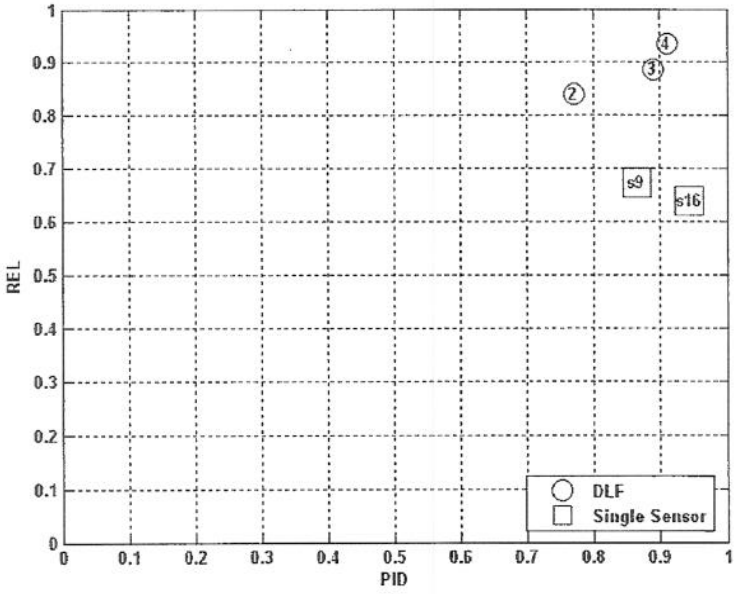


Figure 26. PID vs. reliability for T-72 – Measured

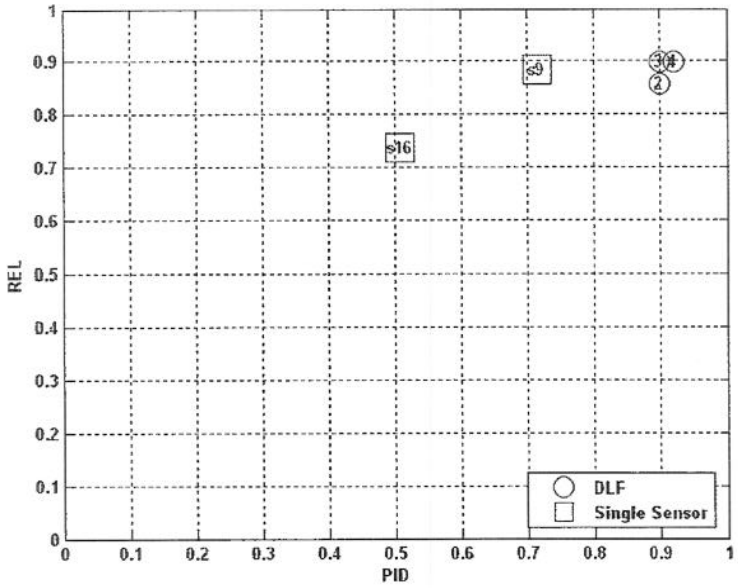


Figure 27. PID vs. reliability for ZSU-23/4 - Measured

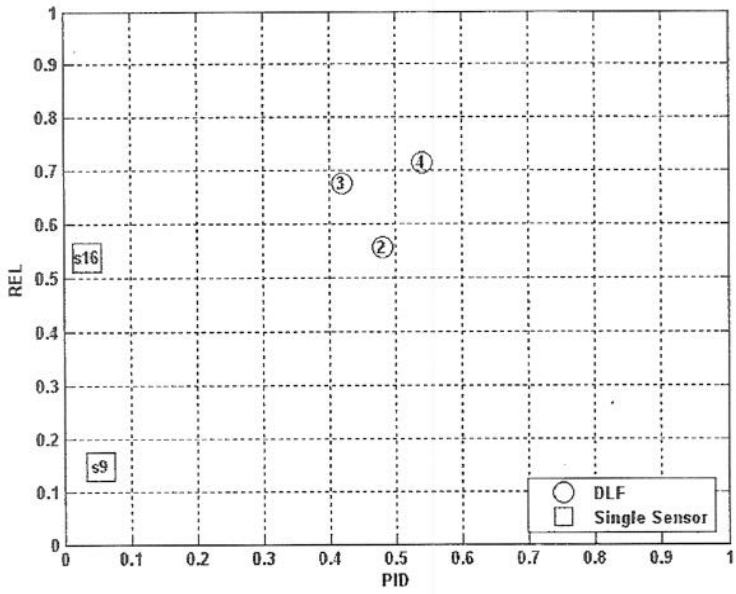


Figure 28. PID vs. reliability for NIL - Measured

Again it is shown that the BTR-70 and the NIL targets provide results that underperform the other targets considerably. Each of these two targets provides different insight into the performance of the DLF process. In the case of the BTR-70, the PID values approach 80%, but the REL values never rise above 70%. This low REL value can be directly traced to the training confusion matrices. For example, consider any of the TCMs shown in Tables 1-6. As described in Section 2.2.2, PID is calculated across a row, while REL is based on the indices in the column of the target declaration. Therefore, since the value corresponding to the BTR-70 column for the NIL truth target is also a large value for an off-diagonal index, the REL associated with that target is constrained by the largest value in any of the other indices in the column. When all the TCMs are considered, the

maximum REL of the BTR-70 is only 62%. Therefore, it is shown that through the process of DLF over four looks the REL associated with the target is increased over 11%.

In the case of the NIL, it is shown that DLF has greatly improved performance over the four look scenario. Of course improvement in performance is relative considering that the percentage improvement in PID is over 90% as the value is raised from 5% to 55%. The improvement in REL nearly matches that of the PID metric, where the values improve 78% from 15% to 70%.

The improvement over the single sensor looks as DLF proceeds, as shown in Table 17, shows that (similar to the simulated experiment) there is great improvement in reliability as looks progress – to over 26% after the fourth look.

Table 17. DLF vs. single sensor over all looks - Measured

		Look 1	Look 2	Look 3	Look 4
PID	DLF	0.73	0.8	0.88	0.89
	SS	0.725	0.725	0.892	0.8925
	Improvement %	0.68%	9.38%	-1.36%	-0.28%
REL	DLF	0.6785	0.817	0.866	0.885
	SS	0.6725	0.6725	0.653	0.653
	Improvement %	0.88%	17.69%	24.60%	26.21%

The overall value of PID increases nearly 18% from the first look to the fourth DLF iteration, but the final PID is actually below that of the single sensor looks at the closer range. This is an ideal example that illustrates the impact of maximizing REL at the expense of PID.

VI. Conclusions

The previous two sections show that DLF significantly improves ATR performance versus single sensor looks. However, the purpose of this research is not to prove that fusion is a useful tool in the ATR arena (this statement is a forgone conclusion). While results of the measured data within the framework of an actual ATR system slightly underperform those that were modeled, they are still encouraging. Figure 29 compares the effects of DLF on both performance of the simulated portion of this research with performance based on measured data and a real ATR system.

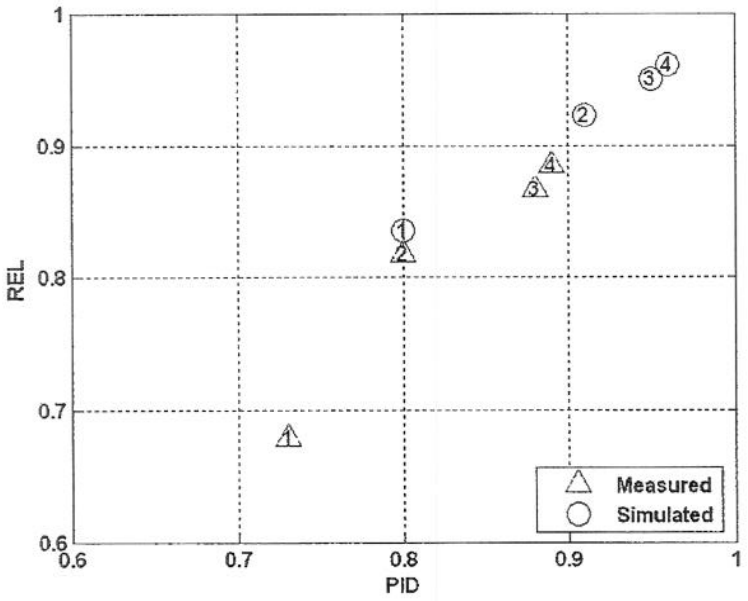


Figure 29. Comparison of PID vs. REL with measured and simulated results

Figure 29 illustrates that although DLF is able to increase performance in both PID and REL there are still differences that factor into the two cases described. Figure 30 extends the plot shown in Figure 29 to include the percentile variances at each look.

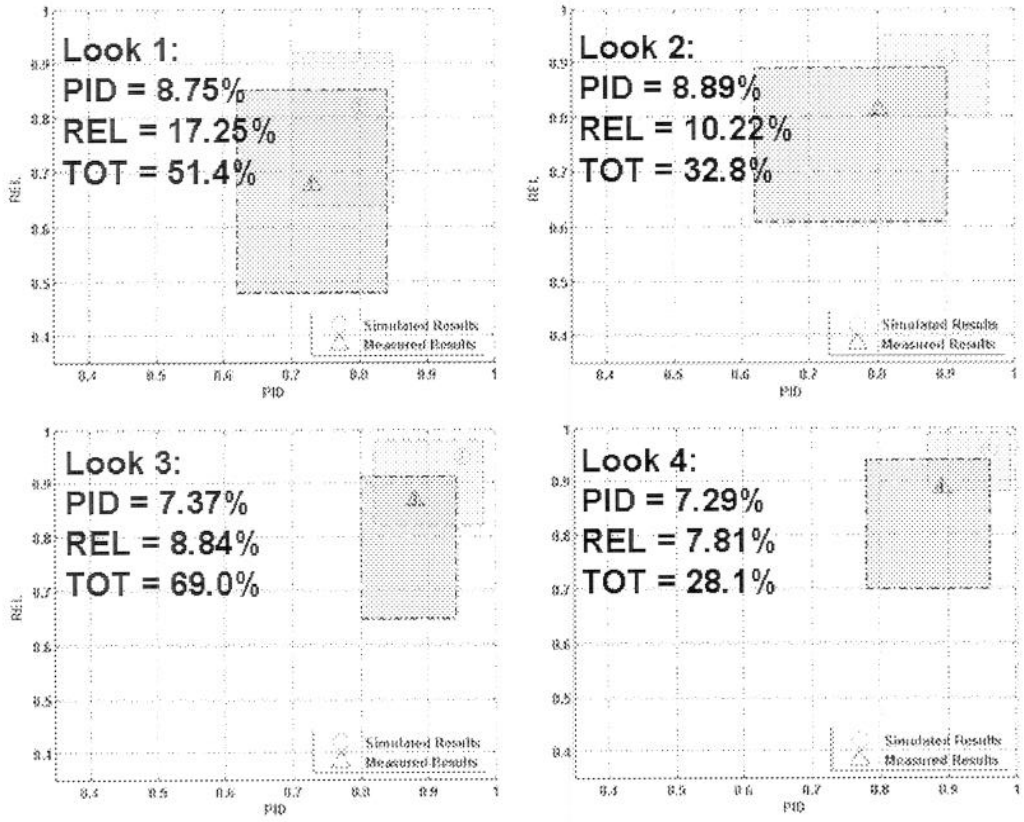


Figure 30. Percentile variance area plots for each look

The difference between the PID and REL at each look is shown on the individual axes. The term TOT refers to the total area in percentile variance. As shown, in Look 1 the area of the simulated variance is 51% the size of the percentile variance of the measured results. As the looks at the target increase the discrepancy in PID and REL is reduced. However, by adding the percentile variances it is seen that even though the median values

are closer, the majority of the data is bounded by a significantly tighter box after the fourth look.

Table 18 compares results in a form that highlights the difference between the use of measured data with a real ATR system and results using simulated data with a modeled ATR system. The most significant difference in performance is in REL of the first look.

Table 18. Table of median PID and REL for measured and simulated results

		Look 1	Look 2	Look 3	Look 4
Measured Data	PID	0.7300	0.8000	0.8800	0.8900
	REL	0.6785	0.8169	0.8664	0.8848
Simulated Data	PID	0.8000	0.9100	0.9500	0.9600
	REL	0.8351	0.9231	0.9505	0.9614
% difference	PID	9.59%	13.75%	7.95%	7.87%
	REL	23.08%	13.00%	9.71%	8.66%

The discrepancy between the simulated and measured results is mostly due to the difference in the ATR predictor used. The modeled ATR system is probabilistic according to expected performance as given in the HCMs, not taking into account that certain aspect regions are more likely to produce errors. For example, each BTR70 within the head-on or tail-on aspect region (-30° – 30°, and 150°-210°) is expected to correctly identify the target 85% of the time. However, if this region had been divided into finer regions, there could be significant changes in results.

The limited data supply (of over 14,000 samples) leads to the initial assumption that the targets are symmetrical for similar angular approaches. Some leeway is taken in

defining symmetry, but the aspect regions must be described by enough data to be statistically significant.

This research provides a critical bridge between the use of components provided by outside sources, including the DLF algorithm and its associated assumptions, the defined scenario, the simulated data in the form of confusion matrices, the developed ATR model, and the intermediate steps of integration with measured data and existing software into a experimental framework that can easily be adapted to the needs of future DLF studies.

This research illustrates the benefits of a simple fusion algorithm such as DLF, but also motivates several questions and possible experiments to further the technology. Some of these questions are as follows.

- How would a variable threshold for setting the NIL affect overall performance?
- If enough data were available, how would a more precise separation of aspect regions affect performance?
- What about a larger target set?
- Would applying DLF to different sensor types, such as a EO/IR camera with an image based ATR and the SAR based ATR system described in this research, be advisable?
- Can related but more complicated fusion algorithms such as Attribute Level Fusion or Feature Level Fusion be pursued?

Each question is worthy of future study as the search for reliable algorithms that maximize the capabilities of multiple sensors, each with different strengths, continues.

In summary, it is shown here that the DLF algorithm is a good approach for the fusion of independent sensor looks on a target. Although the initial experiments that

involved simulated ATR and historical performance illustrated the benefits of the algorithm, actual testing with a proven ATR system shows that performance is adequate, with room for improvement.

VII. References

M. N. Cohen, T. V. Le, and J. Reinebold, "An Approach to Decision Level Identification Fusion," *Combat Identification Systems Conference*, Portsmouth, VA, June 2005.

D. Fitzgerald, ed., Sensor Data Management System, *https://restricted.vdl.afrl.af.mil/sdms/index.htm*, Oct. 2005.

D. Gross and J. Schmitz, "RCIST Sensor System Integration and Performance Modeling," *Report for AFRL/SN*, WPAFB, OH, October 2004a.

D. Gross, "CM_F.m," MATLAB m-file, *Personal Correspondance*, July 2004b.

R. C. Lus and M. H. Lin, "Robot Multi-Sensor Fusion and Integration: Optimum Estimation of Fused Sensor Data," *Conference on Robots and Automation*, Philadelphia, PA, April 1988.

M. Sousa and D. Turner, ed, "RTMSTAR Software User Manual," Version 1.2, *Alphatech Inc.*, Burlington MA, June 2004.

A. N. Steinberg, "Threat Management System for Combat Aircraft," *Tri-Service Data Fusion Symposium,* Vol.1, Warminster, PA, June 1987.

E. Waltz and L. Llinas, *Multisensor Data Fusion*, Artech, 1990.

VIII. Appendix A

Quick-Reference: ATR System Run Guide[1]

1. Installation

The installation disk includes one file, "atrsys.tar". Copy this file to a working location, such as the user HOME directory.

1.1 "Untar" the file as follows:

```
tar -xvf atrsys.tar -C /usr/local
```

This will create a directory "atrsys" under the "/usr/local" directory. Another destination directory can be substituted, if desired, for "/usr/local".

1.2 Modify the "Makefile" file.

```
cd /usr/local/atrsys/source
```

Open "Makefile" with an editor to remove "-lgen" from the "libs" variable:

Change: `libs = -lgen -lm -lz` `glib-config --libs`

To:

```
# libs = -lgen -lm -lz `glib-config --libs`
  libs = -lm -lz `glib-config --libs`
```

1.3 Create the executable as follows:

```
cd /usr/local/atrsys/source
make clean
make atrsys
```

Ensure that no errors were reported.

1.4 The executable is now ready to use:

```
/usr/local/atrsys/source/atrsys
```

---

[1] All references to the actual software package have been replaced by *atrsys* in this text.

2. Running the Application

The steps below describe the process to set up and run data through the ATR system.

2.1 Set Templates

The ATR system allows the user to define which templates the system will attempt to compare the measured data with. By editing the file named *rc_mstar.txt*, the user can set the templates for comparison.

2.2 Set Search Path

Data that is to be processed by the ATR system can be a single file or multiple files contained in a single director. A folder named chips is initially created during the install process. This is the default folder in which the system searches for data. If subdirectories are placed in this folder the user must specify the folder to read.

2.3 Run Application

The application is run once the templates are defined and the search path is set. To begin the process, enter the following at the command line.

```
./atrsys -rc rc_mstar.txt -cp ../chips -numres 1 -pid 10
```

Additional inputs can be added to set search thresholds such as maximum depression, azimuth, and squint angles by adding:

```
-dep <percentage>
-sq <percentage>
-az <percentage>
```

By default these values are set to 100%, allowing for all possible template matches.

The input *–numres* defines the number of outputs. Target declaration is based on the best match. However, in some applications it is beneficial to also see how all templates scored against the truth target that was input. In this case the user would set *–numres* to the total number of template choices and the score of each is displayed.

2.4 Read Results

As the application runs the results are displayed to the screen. One way to capture the results is simple create a file to write the text displayed on the screen to a separate text time and then parse it according to the information that user desires. This method is best suited for cases where *–numres* is greater than one. In the simple, target declaration case, the application can save the results to a file by adding the following command when calling the application.

-o <filename>

The added *–o* input stores the results in a binary file that is then read by an additional application that ships with the software and is installed by default under the normal install process.

To read the binary file and convert it to text type

```
./results  -i <results file from atrsys>  >  newfilename.txt
```

Once the line shown above is executed, a text tile is created that is formatted into a table that can easily be parsed by other additional applications such as MATLAB or EXCEL. An example of the formatted output is shown below.

File	GT Target Call	Y/N	Est Pose	GT Pose	Pose Dlta	ATR Target Call	ATR Score	ERPS	DLTS
hb03914.0015	t72	1	114.000	112.791	-1.209	T72_scnofuel	2.959	5.336	0.977
hb15182.0019	t72	1	238.000	240.084	2.084	T72_scnofuel	0.200	4.217	0.775
hb03907.0015	t72	1	70.000	72.791	2.791	T72_scnofuel	-1.664	4.808	0.907
hb19887.0019	t72	0	176.000	178.084	2.084	ZSU23-4_sc	1.832	4.267	0.780
hb03335.0015	t72	1	252.000	250.791	-1.209	T72_sc	-2.018	4.737	0.901
hb03884.0015	t72	1	310.000	312.791	2.791	T72_scnofuel	1.892	4.998	0.949
hb15142.0020	t72	1	36.000	32.515	-3.485	T72_scnofuel	-0.213	5.190	0.965
hb03877.0015	t72	1	274.000	272.791	-1.209	T72_sc	-1.241	5.399	0.982
hb15120.0019	t72	1	226.000	229.084	3.084	T72_scnofuel	-0.250	4.740	0.901
hb03848.0015	t72	1	62.000	61.791	-0.209	T72_scnofuel	2.071	5.429	0.985
hb15106.0020	t72	1	156.000	156.515	0.515	T72_scnofuel	0.927	5.914	1.000
hb03840.0015	t72	0	26.000	21.791	-4.209	ZSU23-4_sc	0.459	5.167	0.963
hb17642.0019	t72	1	120.000	120.084	0.084	T72_scnofuel	-0.005	4.856	0.911
hb03822.0015	t72	1	258.000	261.791	3.791	T72_sc	-2.023	4.629	0.865
hb15076.0020	t72	1	356.000	356.515	0.515	T72_scnofuel	2.553	4.849	0.911
hb15257.0020	t72	1	282.000	283.515	1.515	T72_sc	-3.555	4.723	0.899
hb15072.0019	t72	1	336.000	338.084	2.084	T72_scnofuel	3.591	5.001	0.949

IX. Appendix B

Simulation Code

```
% Updates for version 4:
% NIL calls are determined by score

% Initialize Variables
minheading = 0;
maxheading = 359;
NumTruth = 5;
RTMSTAR = 0;

plat1loc = [43.3 25;
    16.6 25];
plat2loc = [43.3 -25;
    16.6 -25];
epsi = 4e-2;
its = 100;
h = waitbar(0);
headingchoices = [3.6:3.6:360];

storeCMho9 = zeros(5,5);
storeCMoa9 = zeros(5,5);
storeCMbs9 = zeros(5,5);
storeCMbs16 = zeros(5,5);
storeCMho16 = zeros(5,5);
storeCMoa16 = zeros(5,5);

totlooks = 0;

% Clean the folders in preparation for next test set
if length(dir('C:\FIRST\ResultsData\BRDM2\bs\dep9')) > 3
    CleanFolders
end

for largecount = 1:100 %100 differenct Test Sets
    tic
    if RTMSTAR == 1 %if using "real system" calculate CMs
        NewVerificationGroup(num2str(largecount)) %Choose test set
        CMho9 = FirstCMs_8_25(10,1);
        CMho16 = FirstCMs_8_25(10,3);
        CMbs9 = FirstCMs_8_25(90,1);
        CMbs16 = FirstCMs_8_25(90,3);
        CMoa9 = FirstCMs_8_25(45,1);
        CMoa16 = FirstCMs_8_25(45,3);

        storeCMho16 = storeCMho16+CMho16;
        storeCMho9 = storeCMho9+CMho9;
        storeCMoa16 = storeCMoa16+CMoa16;
        storeCMoa9 = storeCMoa9+CMoa9;
        storeCMbs16 = storeCMbs16+CMbs16;
        storeCMbs9 = storeCMbs9+CMbs9;
    end
    for targcount = 1:5 %[BRDM2 BTR70 T72 ZSU NIL] Target Loop
        targVal = targcount;
        for loopcount = 1:its %Target Heading loop
            totlooks = totlooks+1;
            waitbar(totlooks/(its*6),h,['Iteration ' num2str(totlooks) ...
                ' of ' num2str(its*6)]);
            locX(totlooks) = 0;
            locY(totlooks) = 0;

            %setup Heading for each trial
            targheading = headingchoices(loopcount);

            %Initialize
            CMdlf = zeros(6);
            IDdlf = 0;
            CMdlf_a = zeros(6);
            IDdlf_a = 0;

            AZlist = []; %%Reset each run
```

```
%%%%%%%     Look 1    %%%%%%%%
[Look1aspect,sensor_az,target_az] = aspectCalc(plat1loc(1,1),...
            plat1loc(1,2),locX(totlooks), locY(totlooks),...
            locX(totlooks),locY(totlooks),0,0,targheading);

if RTMSTAR == 1
    %if using "real" ATR system queary results folders for the
    %corresponding result that is the closest match to the
    %Look1aspect for the current truth target

    targDirs = {'BRDM2', 'BTR70', 'T72', 'ZSU', 'NIL'};
    resultdir = fullfile('C:\FIRST\ResultsData',targDirs{targVal},...
                         'test','dep9');
    D = dir(resultdir);
    D = D(3:end);
    fnames = {D(:).name};
    for chipcount = 1:length(fnames)
        AZlist(chipcount) = str2num(fnames{chipcount}(4:end-4));
    end

    nameind = nearest(Look1aspect,AZlist);
    readfile = fullfile(resultdir,fnames{nameind});

    %%%% Here we read the results file to get the ATR call %%%%
    %%%%%%%%%% use RTMSTAR results for ATR prediction %%%%%%%%%%
    [IDss ATRaz] = ATRcall_scorebasedNILs(readfile);

    %Use the correct CM region
    if (ATRaz <= 30) | (ATRaz >= 330) | ((ATRaz >= 150)&(ATRaz <= 210))
        CMss = CMho9;
    elseif ((ATRaz > 60)&(ATRaz < 120)) | ((ATRaz > 240)&(ATRaz < 300))
        CMss = CMbs9;
    else
        CMss = CMoa9;
    end

    CMss(find(CMss == 0)) = epsi;
    for i = 1:length(CMss(:,1))
        CMss(i,:) = CMss(i,:)/sum(CMss(i,:));
    end
    %% IDss is the column in the confusion matrix that corresponds to
    %% the targets
    %%  [BRDM2    BTR70    SCUD    T72    ZSU    NIL] - these also
    %%  correspond to targVal.

    PIDss = CMss(targVal,IDss);
    RELss = CMss(IDss,IDss)/sum(CMss(:,IDss));
    [CMdlf, IDdlf, PIDdlf, RELdlf, DLFconf] = FindDLF(1, targVal, ...
                                              CMss, IDss, CMdlf, ...
                                              IDdlf,0); %Perform DLF
else
    %%%%%%% Use CMs for ATR predition %%%%%%%%%%%%%%%%%%%%%
    CMss = FirstCMs(Look1aspect,1);
    ATRaz = Look1aspect;
    tgt = rand;
    cumProb = 0; % reset the cumulative Probability

    %Modeled ATR system
    for ii = 1:size(CMss,2)
        cumProb = cumProb + CMss(targVal, ii);
        if tgt <= cumProb
            IDss = ii;
            PIDss = CMss(targVal,IDss);
            RELss = CMss(IDss,IDss)/sum(CMss(:,IDss));
            break
        end
    end

    [CMdlf, IDdlf, PIDdlf, RELdlf, DLFconf] = FindDLF(1, targVal, ...
                                              CMss, IDss, CMdlf, ...
                                              IDdlf,0); %Perform DLF
end

%Store outputs
Look{1}.piddlf(totlooks) = PIDdlf;
Look{1}.pidss(totlooks) = PIDss;
Look{1}.relss(totlooks) = RELss;
Look{1}.reldlf(totlooks) = RELdlf;
Look{1}.range(totlooks) = sqrt((plat1loc(1,1) - locX(totlooks))^2 ...
                          + (plat1loc(1,2) - locY(totlooks))^2);
Look{1}.aspect(totlooks) = Look1aspect;
```

```
Look{1}.ATRaz(totlooks) = ATRaz;
Look{1}.truth(totlooks) = targVal;
Look{1}.time(totlooks) = 0;
Look{1}.platform(totlooks) = 1;
Look{1}.sensor(totlooks) = 2;
Look{1}.ptype(totlooks) = 16;
Look{1}.idss(totlooks) = IDss;
Look{1}.iddlf(totlooks) = IDdlf;

if IDss == targVal
    look1ss(totlooks) = 1;
else
    look1ss(totlooks) = 0;
end

%%%%%%%     Look 2    %%%%%%%%%
[Look2aspect,sensor_az,target_az] = aspectCalc(plat2loc(1,1),...
            plat2loc(1,2),locX(totlooks), locY(totlooks),...
            locX(totlooks),locY(totlooks),0,0,targheading);

if RTMSTAR == 1
     %if using "real" ATR system queary results folders for the
    %corresponding result that is the closest match to the
    %Look2aspect for the current truth target

    targDirs = {'BRDM2', 'BTR70', 'T72', 'ZSU', 'NIL'};
    resultdir = fullfile('C:\FIRST\ResultsData',targDirs{targVal},...
                         'test','dep9');
    D = dir(resultdir);
    D = D(3:end);
    fnames = {D(:).name};
    for chipcount = 1:length(fnames)
        AZlist(chipcount) = str2num(fnames{chipcount}(4:end-4));
    end

    nameind = nearest(Look2aspect,AZlist);
    readfile = fullfile(resultdir,fnames{nameind});

    %%% Here we read the results file to get the ATR call %%%
    %%%%%%%%%% use RTMSTAR results for ATR prediction %%%%%%%%%
    [IDss ATRaz] = ATRcall_scorebasedNILs(readfile);

    %Use the correct CM region
    if (ATRaz <= 30) | (ATRaz >= 330) | ((ATRaz >= 150)&(ATRaz <= 210))
        CMss = CMho9;
    elseif ((ATRaz > 60)&(ATRaz < 120)) | ((ATRaz > 240)&(ATRaz < 300))
        CMss = CMbs9;
    else
        CMss = CMoa9;
    end

    CMss(find(CMss == 0)) = epsi;
    for i = 1:length(CMss(:,1))
        CMss(i,:) = CMss(i,:)/sum(CMss(i,:));
    end
    %% IDss is the column in the confusion matrix that corresponds to
    %% the targets
    %%  [BRDM2    BTR70    SCUD    T72    ZSU    NIL] - these also
    %%  correspond to targVal.

    PIDss = CMss(targVal,IDss);
    RELss = CMss(IDss,IDss)/sum(CMss(:,IDss));
    [CMdlf, IDdlf, PIDdlf, RELdlf, DLFconf] = FindDLF(2, targVal, ...
                                              CMss, IDss, CMdlf, ...
                                              IDdlf,0); %Perform DLF
else
    %%%%%%% Use CMs for ATR predition %%%%%%%%%%%%%%%%%%%%%
    CMss = FirstCMs(Look2aspect,2);
    ATRaz = Look1aspect;
    tgt = rand;
    cumProb = 0; % reset the cumulative Probability

    %modeled ATR system
    for ii = 1:size(CMss,2)
        cumProb = cumProb + CMss(targVal, ii);
        if tgt <= cumProb
            IDss = ii;
            PIDss = CMss(targVal,IDss);
            RELss = CMss(IDss,IDss)/sum(CMss(:,IDss));
            break
        end
```

```
    end
    [CMdlf, IDdlf, PIDdlf, RELdlf, DLFconf] = FindDLF(2, targVal,...
                                                CMss, IDss, CMdlf,...
                                                IDdlf,0); %Perform DLF
end

if IDdlf ~= Look{1}.iddlf(totlooks)
    adsf = 0;
end

%Store outputs
Look{2}.piddlf(totlooks) = PIDdlf;
Look{2}.pidss(totlooks) = PIDss;
Look{2}.relss(totlooks) = RELss;
Look{2}.reldlf(totlooks) = RELdlf;
Look{2}.range(totlooks) = sqrt((plat2loc(1,1) - locX(totlooks))^2 ...
                        + (plat2loc(1,2) - locY(totlooks))^2);
Look{2}.aspect(totlooks) = Look2aspect;
Look{2}.ATRaz(totlooks) = ATRaz;
Look{2}.truth(totlooks) = targVal;
Look{2}.time(totlooks) = 0;
Look{2}.platform(totlooks) = 2;
Look{2}.sensor(totlooks) = 2;
Look{2}.ptype(totlooks) = 16;
Look{2}.idss(totlooks) = IDss;
Look{2}.iddlf(totlooks) = IDdlf;

if IDss == targVal
    look2ss(totlooks) = 1;
else
    look2ss(totlooks) = 0;
end

if IDdlf == targVal
    look2dlf(totlooks) = 1;
else
    look2dlf(totlooks) = 0;
end

%%%%%%%     Look 3    %%%%%%%%
[Look3aspect,sensor_az,target_az] = aspectCalc(plat1loc(2,1),...
            plat1loc(2,2),locX(totlooks),locY(totlooks),...
            locX(totlooks),locY(totlooks),0,0,targheading);

if RTMSTAR == 1
    %if using "real" ATR system queary results folders for the
    %corresponding result that is the closest match to the
    %Look3aspect for the current truth target
    targDirs = {'BRDM2', 'BTR70', 'T72', 'ZSU', 'NIL'};
    resultdir = fullfile('C:\FIRST\ResultsData',...
                         targDirs{targVal},'test','dep16');
    D = dir(resultdir);
    D = D(3:end);
    fnames = {D(:).name};
    for chipcount = 1:length(fnames)
        AZlist(chipcount) = str2num(fnames{chipcount}(4:end-4));
    end

    nameind = nearest(Look3aspect,AZlist);
    readfile = fullfile(resultdir,fnames{nameind});

    %%%% Here we read the results file to get the ATR call %%%%
    %%%%%%%%%% use RTMSTAR results for ATR prediction %%%%%%%%%
    [IDss ATRaz] = ATRcall_scorebasedNILs(readfile);

    %Use correct CM region
    if (ATRaz <= 30) | (ATRaz >= 330) | ((ATRaz >= 150)&(ATRaz <= 210))
        CMss = CMho16;
    elseif ((ATRaz > 60)&(ATRaz < 120)) | ((ATRaz > 240)&(ATRaz < 300))
        CMss = CMbs16;
    else
        CMss = CMoa16;
    end

    CMss(find(CMss == 0)) = epsi;
    for i = 1:length(CMss(:,1))
        CMss(i,:) = CMss(i,:)/sum(CMss(i,:));
    end
    %% IDss is the column in the confusion matrix that corresponds to
    %% the targets
    %%  [BRDM2    BTR70    SCUD    T72    ZSU    NIL] - these also
```

```
    %%  correspond to targVal.

    PIDss = CMss(targVal,IDss);
    RELss = CMss(IDss,IDss)/sum(CMss(:,IDss));
    [CMdlf, IDdlf, PIDdlf, RELdlf, DLFconf] = FindDLF(3, targVal, ...
                                                CMss, IDss, CMdlf, ...
                                                IDdlf,0); %Perform DLF
else
    %%%%%%% Use CMs for ATR predition %%%%%%%%%%%%%%%%%%%%%
    CMss = FirstCMs(Look3aspect,3);
    ATRaz = Look1aspect;
    tgt = rand;
    cumProb = 0; % reset the cumulative Probability

    %modeled ATR system
    for ii = 1:size(CMss,2)
        cumProb = cumProb + CMss(targVal, ii);
        if tgt <= cumProb
            IDss = ii;
            PIDss = CMss(targVal,IDss);
            RELss = CMss(IDss,IDss)/sum(CMss(:,IDss));
            break
        end
    end

    [CMdlf, IDdlf, PIDdlf, RELdlf, DLFconf] = FindDLF(3, targVal, ...
                                                CMss, IDss, CMdlf,...
                                                IDdlf,0); %Perform DLF
end

if IDdlf ~= Look{2}.iddlf(totlooks)
    adsf = 0;
end

%Store outputs
Look{3}.piddlf(totlooks) = PIDdlf;
Look{3}.pidss(totlooks) = PIDss;
Look{3}.relss(totlooks) = RELss;
Look{3}.reldlf(totlooks) = RELdlf;
Look{3}.range(totlooks) = sqrt((plat1loc(2,1) - locX(totlooks))^2 ...
                          + (plat1loc(2,2) - locY(totlooks))^2);
Look{3}.aspect(totlooks) = Look3aspect;
Look{3}.ATRaz(totlooks) = ATRaz;
Look{3}.truth(totlooks) = targVal;
Look{3}.time(totlooks) = 0;
Look{3}.platform(totlooks) = 1;
Look{3}.sensor(totlooks) = 2;
Look{3}.ptype(totlooks) = 16;
Look{3}.idss(totlooks) = IDss;
Look{3}.iddlf(totlooks) = IDdlf;

if IDss == targVal
    look3ss(totlooks) = 1;
else
    look3ss(totlooks) = 0;
end

if IDdlf == targVal
    look3dlf(totlooks) = 1;
else
    look3dlf(totlooks) = 0;
end

%%%%%%%     Look 4    %%%%%%%%
[Look4aspect,sensor_az,target_az] = aspectCalc(plat2loc(2,1),...
            plat2loc(2,2),locX(totlooks), locY(totlooks),...
            locX(totlooks),locY(totlooks),0,0,targheading);

if RTMSTAR == 1
    %if using "real" ATR system queary results folders for the
    %corresponding result that is the closest match to the
    %Look4aspect for the current truth target

    targDirs = {'BRDM2', 'BTR70', 'T72', 'ZSU', 'NIL'};
    resultdir = fullfile('C:\FIRST\ResultsData',targDirs{targVal},...
                         'test','dep16');
    D = dir(resultdir);
    D = D(3:end);
    fnames = {D(:).name};
    for chipcount = 1:length(fnames)
        AZlist(chipcount) = str2num(fnames{chipcount}(4:end-4));
```

```
    end

    nameind = nearest(Look4aspect,AZlist);
    readfile = fullfile(resultdir,fnames{nameind});

    %%% Here we read the results file to get the ATR call %%%

    %%%%%%%%%% use RTMSTAR results for ATR prediction %%%%%%%%%
    [IDss ATRaz] = ATRcall_scorebasedNILs(readfile);
    if (ATRaz <= 30) | (ATRaz >= 330) | ((ATRaz >= 150)&(ATRaz <= 210))
        CMss = CMho16;
    elseif ((ATRaz > 60)&(ATRaz < 120)) | ((ATRaz > 240)&(ATRaz < 300))
        CMss = CMbs16;
    else
        CMss = CMoa16;
    end

    CMss(find(CMss == 0)) = epsi;
    for i = 1:length(CMss(:,1))
        CMss(i,:) = CMss(i,:)/sum(CMss(i,:));
    end
    %% IDss is the column in the confusion matrix that corresponds to
    %% the targets
    %%  [BRDM2    BTR70    SCUD    T72    ZSU    NIL] - these also
    %%  correspond to targVal.

    PIDss = CMss(targVal,IDss);
    RELss = CMss(IDss,IDss)/sum(CMss(:,IDss));
    [CMdlf, IDdlf, PIDdlf, RELdlf, DLFconf] = FindDLF(4, targVal, ...
                                                CMss, IDss, CMdlf,...
                                                IDdlf,0); %Perform DLF
else
    %%%%%%%% Use CMs for ATR predition %%%%%%%%%%%%%%%%%%%%%%
    CMss = FirstCMs(Look4aspect,4);
    ATRaz = Look1aspect;
    tgt = rand;
    cumProb = 0; % reset the cumulative Probability

    %modeled ATR system
    for ii = 1:size(CMss,2)
        cumProb = cumProb + CMss(targVal, ii);
        if tgt <= cumProb
            IDss = ii;
            PIDss = CMss(targVal,IDss);
            RELss = CMss(IDss,IDss)/sum(CMss(:,IDss));
            break
        end
    end

    [CMdlf, IDdlf, PIDdlf, RELdlf, DLFconf] = FindDLF(4, targVal, ...
                                                CMss, IDss, CMdlf, ...
                                                IDdlf,0); %Perform DLF
end

if IDdlf ~= Look{3}.iddlf(totlooks)
    adsf = 0;
end

%Store outputs
Look{4}.piddlf(totlooks) = PIDdlf;
Look{4}.pidss(totlooks) = PIDss;
Look{4}.relss(totlooks) = RELss;
Look{4}.reldlf(totlooks) = RELdlf;
Look{4}.range(totlooks) = sqrt((plat2loc(2,1) - locX(totlooks))^2 ...
                          + (plat2loc(2,2) - locY(totlooks))^2);
Look{4}.aspect(totlooks) = Look4aspect;
Look{4}.ATRaz(totlooks) = ATRaz;
Look{4}.truth(totlooks) = targVal;
Look{4}.time(totlooks) = 0;
Look{4}.platform(totlooks) = 2;
Look{4}.sensor(totlooks) = 2;
Look{4}.ptype(totlooks) = 16;
Look{4}.idss(totlooks) = IDss;
Look{4}.iddlf(totlooks) = IDdlf;

if IDss == targVal
    look4ss(totlooks) = 1;
else
    look4ss(totlooks) = 0;
end
```

```
            if IDdlf == targVal
                look4dlf(totlooks) = 1;
            else
                look4dlf(totlooks) = 0;
            end
            Azchoice(totlooks) = targheading;
        end
    end

    loc.X = locX;
    loc.Y = locY;
    %Clean directories for new test set
    if RTMSTAR == 1
        CleanFolders
    end
    toc
end
```

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
23-03-2006	**Master's Thesis**	August 2004 - March 2006

4. TITLE AND SUBTITLE

Verification of a Decision Level Fusion Algorithm Using a Proven ATR System and Measured SAR Data

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Thompson, James D.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/06-60

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/SNZT
Attn: Mr. Michael J. Gauder
Area B; Bldg 620; Rm 1BI41
2241 Avionics Circle
WPAFB OH 45433-7303
email: michael.gauder@wpafb.af.mil
Phone: 255.4794 x4276

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Decision level fusion (DLF) algorithms combine outputs of multiple single sensors to make one confident declaration of a target. This research compares performance results of a DLF algorithm using measured data and a proven ATR system with results from simulated data and a modeled ATR system. This comparison indicates that DLF offers significant performance improvements over single sensor looks. However, results based on simulated data and a modeled ATR are slightly optimistic and overestimate results from measured data and a proven ATR system by nearly 10% over all targets tested.

15. SUBJECT TERMS

Automatic Target Recognition, Synthetic Aperture RADAR, Sensor fusion, Decision Level Fusion, Multi-Look Sensing

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
REPORT	ABSTRACT	c. THIS PAGE			Steven C. Gustafson, PhD. (ENG)
U	U	U	UU	80	19b. TELEPHONE NUMBER *(Include area code)* (937) 255-3636, ext 4598; e-mail: steven.gustafson@afit.edu