The Adaptive SAR ATR problem set (AdaptSAPS)

Angela R. Wise*[a], Donna Fitzgerald[b], Timothy D. Ross[c]

[a]AFRL COMPASE Center, Jacobs Sverdrup ASG, WPAFB, OH 45433
[b]AFRL SDMS, General Dynamics AIS, WPAFB, OH 45433
[c]AFRL/SNAR, WPAFB, OH 45433

ABSTRACT

A strong and growing interest in systems that adapt to changing circumstances was evident in panel discussions at the "Algorithms for SAR Imagery" Conference of the AeroSense Symposium in April 2003, with DARPA, Air Force, industry and academia participation. As a result, Conference Co-Chair Mr. Ed Zelnio suggested producing a dynamic model to create problem sets suitable for adaptive system research and development. Such a problem set provides a framework for the overall problem, including organization of operating conditions, performance measures and specific test cases. It is hoped that this AdaptSAPS framework will help provide the community with a more concrete base for discussing adaptation in SAR imagery exploitation. AdaptSAPS Version 1.0 was produced by the AFRL COMPASE and SDMS organizations and posted on 5 August 2003. AdaptSAPS consists of over a dozen MatLab programs that allow the user to create "missions' with SAR data of varying complexities and then present that test data one image at a time, first as unexploited imagery and then later with the exploitation results that an ATR could use for adaptation in an operational environment. AdaptSAPS keeps track of performance results and reports performance measures. This paper describes AdaptSAPS – its application process and possible improvements as a problem set.

Keywords: problem set, adaptive systems, ATR, SAR

1. INTRODUCTION

A fundamental problem for ATR system development, actually for most any trained classifier system development, is that the training data may not be completely representative of the operational problem. One method for dealing with this problem is to train on the data available during development, but then to adapt to conditions as they are found operationally.

A growing interest in systems that adapt to changing circumstances was demonstrated in panel discussions, driven principally by Mr. Steve Welby (DARPA), at the Algorithms for SAR Imagery Conference of the AeroSense Symposium in April 2003, with DARPA, Air Force, industry and academia participation. As a result, Conference Co-Chair, Mr. Ed Zelnio, suggested producing a dynamic model to create problem sets suitable for adaptive system research and development. That is, instead of a one-shot batch training set and then a one-shot batch test set, we have an initial batch training set and then the remaining data be made available sequentially – initially as a test instance and then with "truth" so that the system may adapt. Such a problem set could encourage adaptive approaches to target detection in synthetic aperture radar (SAR) imagery by providing a framework for the overall problem, including organization of operating conditions, performance measures and specific test cases. It is hoped that this "Adaptive SAR ATR Problem Set" (AdaptSAPS) framework will help provide the community with a more concrete basis for discussing adaptation in SAR imagery exploitation.

To facilitate work in adaptive systems, AFRL COMPASE (for example, see Ref. 1) and SDMS organizations produced AdaptSAPS Version 1.0 and posted it at the public SDMS web site (Ref. 2) on 5 August 2003. AdaptSAPS consists of over a dozen MatLab programs that allow the user to create "missions" with SAR data of varying

* Angela.Wise@wpafb.af.mil; phone (937) 904-9315; fax (937) 656-7425

complexities and then present that test data one image at a time, first as unexploited imagery and then later with the exploitation results that an ATR could use for adaptation in an operational environment. AdaptSAPS keeps track of performance results and reports performance measures.

All elements of the AdaptSAPS package are approved for public release and are available from SDMS (Ref. 3). It is also important to be aware that Version 1.0 is populated with completely unsequestered data.

The AdaptSAPS providers are interested in suggestions for improving AdaptSAPS as a problem set. As an example, this initial version of AdaptSAPS leverages the previously released MSTAR data. Future versions may use other data sets or capabilities to synthesize data. Another area for improvement is in performance measures for self-assessed confidence, which is important in adaptive systems. The problem set will be managed by AFRL, based on discussions about future AdaptSAPS versions at the Algorithms for Synthetic Aperture Radar Imagery Conference of the SPIE International Symposium on Defense and Security.

This paper describes the overall AdaptSAPS package (Section 2), explains the recommended performance measures (Section 3), and provides additional characterization of the MSTAR data (Section 4). The paper concludes with a discussion of future work and a request for participation in improving AdaptSAPS (Section 5).

2. AdaptSAPS CONCEPT OF OPERATION AND TOOLS

This section describes the basic concept of operation for AdaptSAPS and the tools that are provided as part of the AdaptSAPS package. The detection of targets in SAR imagery is a well-studied problem with a rich foundation (data sets, performance measures, exploitation systems, test results). Most of the current research with this foundation, or target detection in general, has been from the perspective of an exploitation system that is trained off-line and then used without adaptation on-line. The objective of AdaptSAPS is to facilitate research with systems that may be trained off-line, but then also adapt to improve their performance on-line.

There are a variety of steps involved in a real system that would detect targets in SAR imagery. AdaptSAPS is focused on only one of these steps. A "full-scene" SAR image is what is directly collected and may cover a square kilometer or more. Typically a pre-screener is run over a full-scene image and smaller "image chips" are selected for further processing. An image chip will have dimensions similar to that of a target, covering perhaps a few hundred square meters. The image chips selected by the pre-screener may or may not actually have a target in them. These chips are then individually processed to make a final decision about whether or not they contain a target. AdaptSAPS is concerned only with this second step. The phrase system-under-test (SUT) is used to refer to the exploitation system that accepts image chips as input and makes a target present/absent decision. It is this SUT that adapts in the AdaptSAPS setting. The AdaptSAPS package does not include the SUT, other than a trivial example version to demonstrate interfaces. The researcher using AdaptSAPS will provide the SUT.

AdaptSAPS is intended to emulate the following concept of operation. We image that the SUT is trained off-line with data that is limited in quantity and variety. The SUT is then deployed in a situation that was not perfectly represented in the off-line training data. The SUT must operate on image chips in this new operational situation. So far, that is all consistent with non-adaptive system use. The new element is that we assume that human analysts, perhaps fusing information from other sources and different times, further analyze the imagery. The results of this human exploitation are then provided to the SUT. This information allows the SUT to know where it made mistakes and to use the operational data to refine itself, i.e., to adapt. This process might be repeated as the SUT is taken to each new deployment. The different deployments are called "missions" in the AdaptSAPS terminology.

Ideally, AdaptSAPS would model the human exploitation and corrupt the feedback provided to the SUT to simulate any errors that might be made. This initial version of AdaptSAPS assumes that the human exploitation is performed perfectly and uses truth data (without corruption) as feedback. Although adaptation with perfect feedback represents a sufficiently challenging problem initially, future versions of AdaptSAPS may include a more realistic human exploitation model.

The AdaptSAPS tools may be used in any way a particular researcher finds useful, but a baseline process is recommended. In this baseline process, the researcher would perform off-line training of their SUT using the data and detailed procedures specified in the AdaptSAPS readme.txt file. The researcher then runs the AdaptSAPS main program (run_missions.m) specifying a list of missions. Pre-defined missions 1-10 are included (see Table 1). Other missions may be defined, typically as a collection of image chips with a common set of operating conditions. AdaptSAPS loops through each of the listed missions. With each mission it loops through randomly selected image chips. For each image chip, it is first provided to the SUT without truth information (i.e., [estTgtNontgt estTargetProb] = egSutExploit(filenameTestImage ,iMission)) and the SUT returns its decision (target or no target in the image chip) and an indication of its confidence in that decision (in the form of an estimate of the probability that the chip contained a target). The SUT may use any information included in the arguments (i.e., test image and mission number) for its initial exploitation. Note that any truth information that would not normally be available operationally has been redacted from the test image's header. The mission number is provided as an indication that the mission has changed, but the SUT should not have access to the operating conditions (e.g., the prior probability of a given target type) associated with each mission. The SUT should also not use prior knowledge about the MSTAR data collections to then use site, time, lat, long, etc. from the header to inform exploitation or adaptation.

After the SUT has made its estimates, AdaptSAPS then provides truth information (i.e., egSutAdapt(filenameTestImage, iMission, trueTgtNontgt)). This initial version of AdaptSAPS only provides truth concerning target presence. Future versions of AdaptSAPS may provide additional information (e.g., target orientation, target or false alarm object type, etc.); however as the feedback becomes more detailed it also becomes more important to properly reflect the errors that would inevitably be associated with such feedback.

AdaptSAPS accumulates performance measures as the missions and image chips are tested. These performance measures are described in Section 3.

Figure 1 summarizes the key elements of the AdaptSAPS tools.

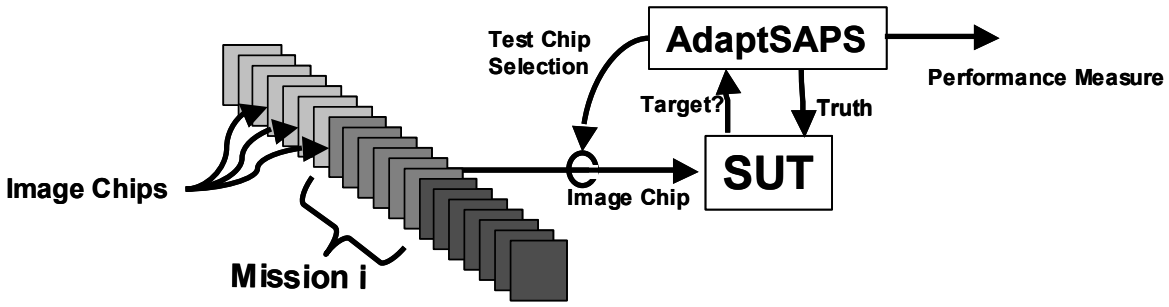

Figure 1. AdaptSAPS Operation

The baseline off-line (or "batch") training data is purposefully very limited, consisting of only 72 target chips of a single serial number (132) of a single type (T72) at a single depression angle (17 deg.) and 72 clutter chips from the most benign (set A) operating conditions. The readme.txt file lists the specific chips. This limited training data may result in artificially poor initial performance from the SUT, but it will also accentuate the adaptive capabilities of the SUT.

The AdaptSAPS baseline includes the ten pre-defined missions of Table 1. Additional missions may be defined by specifying a target set, a clutter set, a confuser set, prior probabilities, and the total number of images in a mission. As examples, a particular researcher may be interested in version variants only or use many more images per mission and would want to define missions accordingly.

AdaptSAPS Version 1.0 encourages consideration of one particular definition of “target”, i.e., as coded in create_DB.m, future versions may make this easier to change. The following vehicle types are only used as Targets: 2s1_gun, bmp2_tank, brdm2_truck, btr60_transport, btr70_transport, t62_tank, t72_tank, and zsu23-4_gun. The

following types are only used as confusers: d7_bulldozer, clutter, slicey, and zil131_truck. For the number of chips in an mission, multiples of 4 are convenient since quartiles of the test chips are scored. Note that the baseline missions all have 120 images, but work with larger numbers (even thousands) of images is also of interest.

Mission No.	Mission Name	Target Set	Clutter Set	Confuser Set	Priors (Tgt, Confuser, Clutter)	Total no. of mission images
1	Benign	T72 Nominal	A	None	0.4, 0.0, 0.6	120
2	Baseline	T72 EOC	B	Slicy	0.4, 0.1, 0.5	120
3	Target-Rich	T72 EOC	B	Slicy	0.6, 0.1, 0.3	120
4	Target-Poor	T72 EOC	B	Slicy	0.3, 0.1, 0.6	120
5	Hard Clutter	T72 EOC	D	Slicy	0.4, 0.1, 0.5	120
6	Confusers	T72 EOC	B	Slicy, Truck, and Bulldozer	0.4, 0.3, 0.3	120
7	Tracked Tgts	Tracked Types	B	Slicy	0.4, 0.1, 0.5	120
8	Wheeled Tgts	Combat Types	B	Slicy	0.4, 0.1, 0.5	120
9	Moderate	Tracked Types	C	Slicy and Truck	0.4, 0.2, 0.4	120
10	Hard	Combat Types	D	Slicy, Truck, and Bulldozer	0.4, 0.3, 0.3	120

Table 1. Pre-Defined Missions

In the pre-defined missions, we included 15-17 deg. depression angles and excluded greater than 17 deg. depression angles throughout. Articulation variants are not included since they do not occur at the included depression angles. We assumed that the Collection is not a significant operating condition and did not use it in partitioning the data. The offline training data is not considered to be a "mission", but Mission 1 (with similar OCs) is a Mission. Adaptation is desired on Mission 1.

3. PERFORMANCE MEASURES

This section defines recommended performance measures for the system's adaptability. Better adaptation might include many things, e.g., efficiency (learning with fewer sequential data points, taking fewer CPU cycles to perform each update, limiting growth of required memory, etc.), robustness (adapting to more and more extreme OCs), and post-adaptation accuracy (P_d/FAR/P_{id} and self –confidence accuracy). For scoring, the truth from the headers and reports from the system under test will be used. All testing is at chip level to avoid issues concerning location accuracy or truth-to-report "association" problems. An un-weighted averaging is being used, since we already control the population mix in each mission or experiment.

The following measures are proposed as something that encourages the desired adaptive behavior, but does so imperfectly. Again, the AdaptSAPS providers encourage suggestions and comments for better and/or simpler performance measures. Further development of the AdaptSAPS measures is reported in Ref. 4.

Now, from one perspective, a given set of test data and a given system under test (SUT) produce two distributions on the reported Probability of target (ProbTgt). One distribution is for the target test data and the second is for the non-target test data. As always, the desire is that the two distributions be well separated. This separation might be measured

in one of several ways, e.g., one, the probabilistic distance measure (e.g. Bhattacharyya distance), two, probability of false alarm (P_{fa}) at a fixed probability of detection (P_d), three, P_{fa} or ($1-P_d$) when they are equal, or four, measure the area under the ROC curve. The later is the measure used here to reflect discrimination performance.

It is also desired that the reported ProbTgt to be accurate. In other words, of all the reports with the confidence of ProgTgt, the fraction of those that are actually targets should be about ProbTgt. This accuracy may be measured as the difference between actual and reported probabilities or the mutual information between reported probabilities and the correctness of decisions. Here, the providers use the difference in probabilities to reflect the confidence accuracy.

It is often beneficial to have a single summary performance measure and we propose MOP_A as $MOP_A = (E + (1-D))/2$. Where error (E) is the average across the five bins (containing equal numbers of test instances) of the difference (RMS) between the average reported ProbTgt in the bin and the actual target fraction in the bin. And where discrimination (D) is the area under the $P_d - P_{fa}$ ROC curve. Figure 2 illustrates examples where E and D would be large and small.

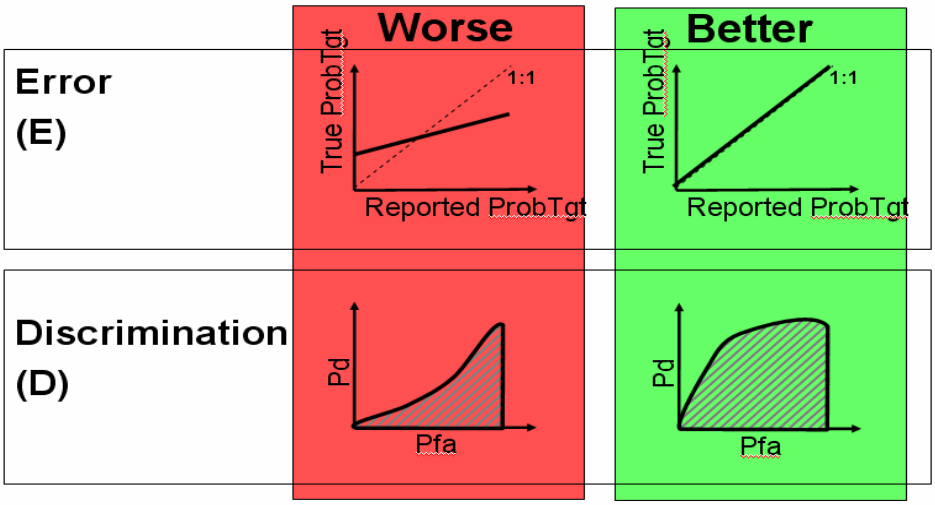

Figure 2. MOP_A Relationship

There are some general considerations related to MOP_A. The smaller the MOP_A, the better and the score should always be in [0, 1]. If the score set does not include both target and non-target entries then D is undefined and therefore the MOP_A is undefined and if the score set does not have at least one entry per bin then E is undefined and again the MOP_A is undefined. Note that the error term includes a sample size dependent bias, so it will vary especially at small sample sizes. Comparisons should only be made between similar sample sizes. Note also that the current MOPs depend solely on the SUT reported score (estimated ProbTgt) and do not use the SUT's target/non-target decision (estimated TgtNontgt).

The measure of performance of adaptation (MOP_A) should be reported for the overall experiment, each mission, and each quartile of each mission. The objective of doing this is to encourage the SUT to have *improving* self-assessed confidence and differentiation of targets and non-targets.

4. MSTAR DATA SET

The public MSTAR data set can be found on SDMS (Ref. 5), including a recently compiled list of 150 papers that have made use of the MSTAR data. The public data set is measured one foot resolution X-band complex SAR images and includes target chips from MSTAR collections 1 and 2 and clutter full scene imagery from MSTAR collection 1.

The AdaptSAPS package includes a tool to create clutter chips from the full scenes. MSTAR collection 1 occurred in September 1995 near Huntsville, Alabama and MSTAR collection 2 occurred in November 1996 at Eglin Air Force Base, Florida. Both collections utilized Sandia National Laboratory's STARLOS sensor.

The public MSTAR data set contains four packages (MSTAR Clutter, MSTAR Targets, MSTAR/IU T-72 Variants, and MSTAR/IU Mixed Targets) making up a total of seven CDs. The data contained on the CDs comprise of 11 target types and 23 unique serial numbers and are from 15, 17, 30, and 45 degree depression angles with varying aspect angles. Each file is constructed with a prepended, variable-length, Phoenix formatted header that contains the details of the ground, sensor, and image truth for each chip. The Phoenix header is then followed by the magnitude and phase data blocks. Tools for reading, viewing and manipulating the MSTAR data can be found online.

Candidate target operating conditions (OCs) for this data set can be separated into three dimensions, target, sensing, and environment (see Ref. 6 for background on Ocs). Target dimension OCs include serial number variations, version variations, articulation/configuration variations, target type variations, class variations, target dimensions, and prior probabilities. Sensing OCs include synthetic noise and depression angles. And lastly, the environment OCs include collection variations. There are a total of 17,096 target chips with a varying number of pixels. The smallest chips are 54 x 54 (range x cross-range) and the largest is 192 x 193 pixels in size. Table 2 is a summary of the OCs present on the three public MSTAR target CD sets; MSTAR Targets, MSTAR/IU T-72 Variants, and MSTAR/IU Mixed Targets. Table 3 is a count of the number of target instances across these OCs. The columns in Tables 2 and 3 are labeled to indicate the collection number (C), scene (S) and depression angle.

Target Type	Bumper Number	C1S1_15	C1S1_17	C2S1_15	C2S1_16	C2S1_17	C2S1_29	C2S1_30	C2S1_31	C2S1_43	C2S1_44	C2S1_45	C2S2_30	C2S2_45	C2S3_30	C2S3_45
2S1	B01			N		N		N				N				
BMP2	9563	N	N													
BMP2	9566	N	N													
BMP2	C21	N	N													
BRDM2	E71			N		N		N				N			Afs	Afs
BTR60	K10YT7532	N	N													
BTR70	C71	N	N													
D7	92v13015			N		N										
slicy	1			N	N	N	N	N	N	N	N	N				
T62	A51			Cf		Cf										
T72	132	N	N													
T72	812	Cf	Cf													
T72 M	A04			VCf		VCf										
T72 M1	A05			V		V										
T72 M	A07			V		V										
T72 M	A10			V		V										
T72 AV	A32			VCfr		VCfr										
T72 B	A62			VCf		VCf										
T72 B	A63			VCf		VCf										
T72 BE	A64			V		V		V				V			VAth	VAth
T72	S7	N	N													
ZIL131	E12			N		N										
ZSU23/4	D08			N		N		N				N	Atgd	Atgd		

N=Nominal
A=Articulation (t=turret, g=gun, h=hatch, f=firing rack, s=sight port, d=dish)
C=Configuration (f=fuel barrels, r=reactive armor)
V=Version Variant

Table 2. Public MSTAR target OC summary

Target Type	Bumper Number	C1S1_15	C1S1_17	C2S1_15	C2S1_16	C2S1_17	C2S1_29	C2S1_30	C2S1_31	C2S1_43	C2S1_44	C2S1_45	C2S2_30	C2S2_45	C2S3_30	C2S3_45
2S1	B01			274		299		288				303				
BMP2	9563	195	233													
BMP2	9566	196	232													
BMP2	C21	196	233													
BRDM2	E71			274		298		287				303			133	120
BTR60	K10YT7532	195	256													
BTR70	C71	196	233													
D7	92v13015			274		299										
slicy	1			274	286	298	210	288	313	255	312	303				
T62	A51			273		299										
T72	132	196	232													
T72	812	195	231													
T72 M	A04			274		299										
T72 M1	A05			274		299										
T72 M	A07			274		299										
T72 M	A10			271		296										
T72 AV	A32			274		298										
T72 B	A62			274		299										
T72 B	A63			274		299										
T72 BE	A64			274		299		288				303			133	120
T72	S7	191	228													
ZIL131	E12			274		299										
ZSU23/4	D08			274		299		288				303	118	119		

Table 3. Public MSTAR target instance counts

The clutter candidate OC dimensions fall into three categories. These include imaging geometry (depression, squint, etc.), clutter features, and confusers (slicy, D7, Zil). There are a total of 100 MSTAR public release full scene clutter images that have been run through a nominal ATR prescreener to produce 1,160 "target-like" clutter chips. Each chip is 128 x 128 pixels. Each chip possesses the following features: clutter type, score (as assigned by the prescreener), mean, variance, standard deviation, RMS, skewness, kurtosis, maximum, total integral (sum of pixel magnitudes across entire chip), and zero-valued points. The clutter type has been assigned 6 categories based on the features of the chip. They are as follows: C1 = Cultural Isolated Object False Alarm (small building, vehicle, etc); C2 = Natural Isolated Object False Alarm (tree, rock, etc); C3 = Cultural Edge / Corner False Alarm (things from fences, roads, etc); C4 = Natural Edge / Corner False Alarm (things from tree lines, streams, etc); C5 = Cultural Homogenous Area False Alarm (on a large building, parking lot, etc); and C6 = Natural Homogenous Area False Alarm (on a grass field, forest canopy, etc). There are 345, 310, 189, 73, 122, and 120 chips in the six categories respectively. Figure 3 gives examples of these six categories by type and Figure 4 gives examples for clutter based on the score.

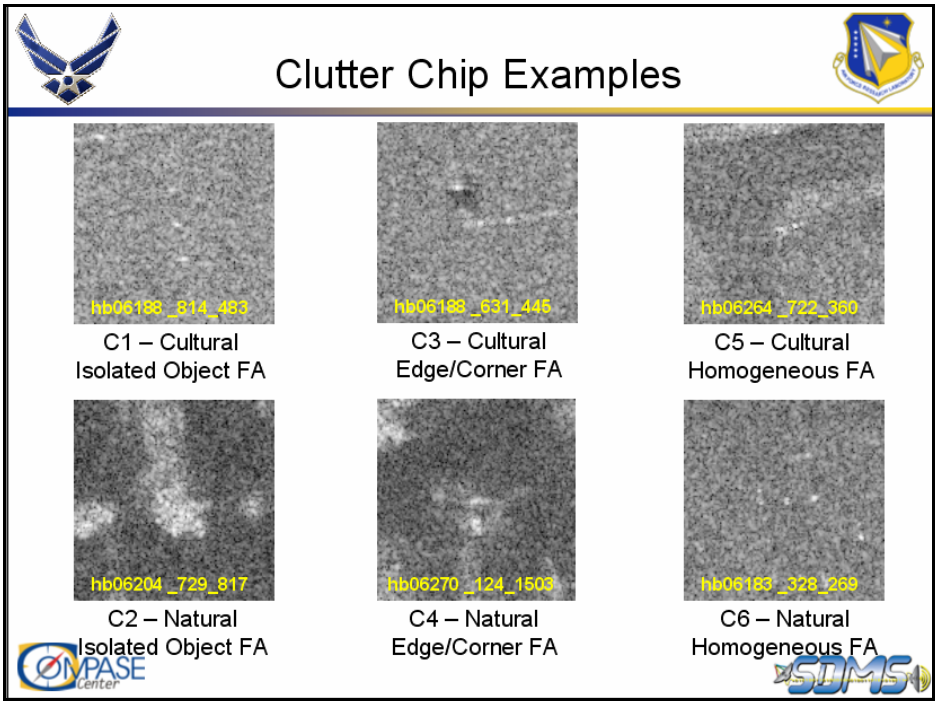

Figure 3. Clutter Chip Examples by Clutter Type

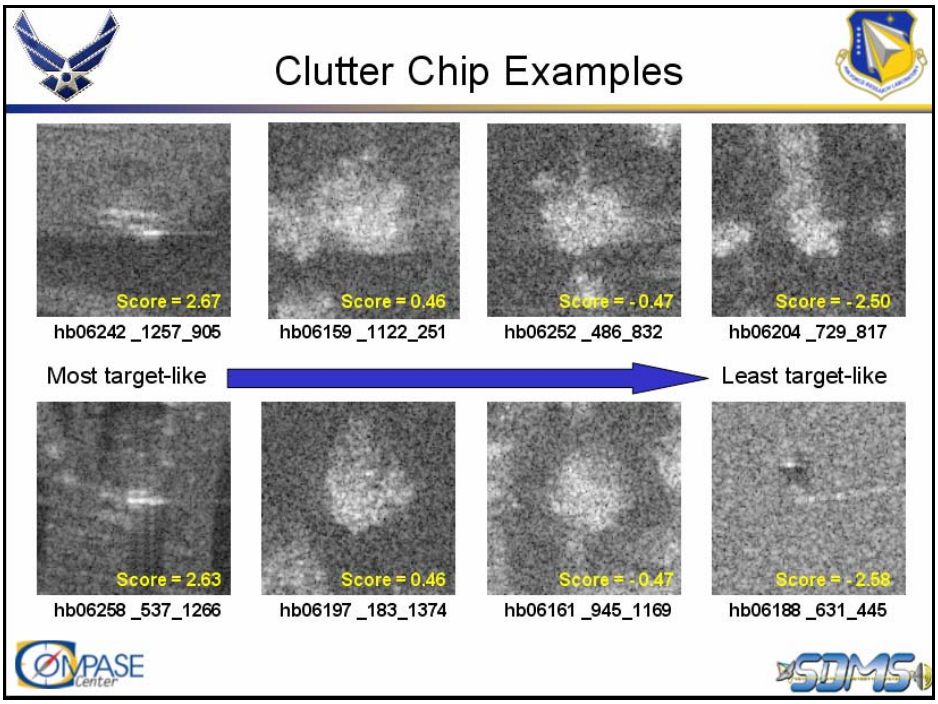

Figure 4. Clutter Chip Examples by Score

The six classifications of clutter are then broken down into four clutter sets – A, B, C, and D. Set A includes the C6 clutter equaling 120 chips. Set B includes the C3, C4, and C5 clutter totaling 384 chips. Set C contains the 310 C2 clutter chips. And Set D is the C1 clutter which adds up to 345 chips.

5. FUTURE WORK

The AdaptSAPS has been publicly released to the sensor development community using websites, such as SDMS and Mathworks File Exchange, and email blasters.

The AdaptSAPS download page on the SDMS website contains a request form to capture some basic information about the customers who were downloading AdaptSAPS Version 1.0. A total of 41 customers downloaded from the SDMS website as of 15 January 2004. There were also 295 downloads from the Mathworks File Exchange (Ref. 7), however no information is available on those customers. The SDMS customer base consisted of government organizations, DoD companies, universities, and others from outside of the country. The U. S. based customers were sent a simple questionnaire to fill out to help gather inputs and comments on the usability AdaptSAPS. Listed below is a sampling of the results.

The "Purpose for Downloading" comments are:

- "Primarily looking for open source of data with target features to perform ATR/sensor fusion experiments", Major Trevor Laine, AFIT/ENS

- **"**Research for my final year project which refers to the signal and image processing area. Utilize wavelets purposed to apply the Winger distribution decomposition in order to constract an algorithm for ground penetrating radar for a specific class of targets", Petroula Karle

- "Interested in seeing calibrated images from target and clutter, and knowing what I was looking at in the clutter", Anonymous

In the "Summary of Results", the comments from a good number of customers were that they were experimenting with the data set and had no real conclusions.

The Comments and Improvements are:

- "The clutter-chip generator and the clutter excel sheet were very useful adjuncts to the MSTAR public data", Anonymous

- "Thanks for developing this and making it available. It was a big time saver", Anonymous

The providers of AdaptSAPS hope to implement this type of user feedback and move toward more advanced versions of AdaptSAPS. Future versions may implement the idea of the exploitation model to provide more realistic feedback to the SUT. Synthetic effects on the target, confuser and clutter data may be used. The noise level could be adjusted or the shadow altered in an attempt to test the SUT. The entire methodology surrounding the AdaptSAPS package has the possibility for improvement. Future versions may or may not provide the initial batch training set discussed earlier. The level of truth provided to the SUT may be decreased to nothing (unsupervised) or increased to include target type, aspect angle, etc. On the same note, the AdaptSAPS may score more detailed reports to include the target type, aspect angle, etc. Users may prefer a different mechanism for interfacing between AdaptSAPS, the data, and the SUT. The imagery and truth may be provided on a predetermined schedule rather than on-demand. Users may even rather have imagery provided in image sets rather than individually.

As suggested, user feedback and suggestions is highly encouraged and will determine the nature of AdaptSAPS Version 2.0. Currently, the authors know of at least two papers to appear in the 2004 SPIE conference which leverage AdaptSAPS (by D. Parker and A. Williams) and would appreciate receiving notice on others papers or forums where feedback on AdaptSAPS might be gathered.

6. SUMMARY AND CONCLUSIONS

This paper introduces the AdaptSAPS package. This package suggests processes and provides tools to encourage research with adaptive target detection systems. Self-assessed confidence is an important part of adaptation and of benefit in itself, so the package encourages research in that direction also. The package provides a new operating condition characterization and a chipping tool for the MSTAR public release clutter data that may be of interest aside from adaptation. The AdaptSAPS package (Ref. 3) consists of a briefing (with content similar to that of this paper), the MSTAR Public Release data, a baseline batch training set specification, a spreadsheet with clutter characterization information, and Matlab tools. The MSTAR data is available from SDMS (Ref. 5) and includes MSTAR Clutter, MSTAR Targets, MSTAR/IU T-72 Variants, and MSTAR/IU Mixed Targets packages. The batch training set is defined in the readme.txt file and lists target and clutter chip identifiers. The tools are described in the readme.txt file and by the high-level code itself. The tools for installation and setup enable clutter chip generation from full scene clutter images, database generation for target, confuser, and clutter operating conditions, and for defining missions with a script that generates image lists from the parameters for enumerated missions. The tools for execution include the main (run_missions.m), an example SUT (egSutInit.m, egSutExploit.m, egSutAdapt.m), the server of test images and truth (sarOracle.m) and performance measure computations (getMOPs.m). The readme.txt file contains step-by-step instructions on set-up, initialization and execution of AdaptSAPS, a list of the options available for each field in mission definition, the baseline list of batch training data, a summary of AdaptSAPS in the format suggested for the UCI Repository of machine learning databases, and example text output for Missions 1-10. Future versions of AdaptSAPS will evolve with feedback and other contributions from its users.

ACKNOWLEDGEMENTS

- Steve Welby, DARPA
    - Provided impetus for this problems set in comments during a SPIE '03 panel discussion, particularly that static problems have become less relevant to the problems faced by today's military.

- Ed Zelnio, AFRL/SNA
  - Originated the idea of wrapping static problem sets with procedures that create dynamic problem sets.
- Lanson Hudson, AFRL COMPASE Center, JE Sverdrup
  - Developed code for clutter chip generation, display, and analysis.
- Mike Bryant, AFRL/SNA
  - Principally responsible for the original MSTAR data public release, suggested the exploitation emulation component, and allowed us to use chip database/server code he developed at Wright State University.
- Ron Dilsavor, AFRL/SNA
  - Provided guidance on methods for SAR image manipulation, inclusion of synthetic effects, and constructive criticism of MOPs.
- Mark Minardi, AFRL/SNA
  - Contributed area-under-ROC curve algorithm and MOP guidance generally.
- ATRWG/DUSD
  - Developed guidelines for Problem Set definition which were considered here.
- Capt. Dave Parker, AFIT
  - Improvements in code and documentation based on AdaptSAPS beta testing.
- Arnold Williams, Science Applications International Corporation (SAIC)
  - Took the AdaptSAPS package to get initial results of the utility of AdaptSAPS for adaptive ATR development and assessment.

REFERENCES

1. Ross, T., L. Westerkamp, R. Dilsavor, and J. Mossing. "Performance Measure for Summarizing Confusion Matrices – The AFRL COMPASE Approach", Proc. SPIE, Vol. 4727, Algorithms for Synthetic Aperture Radar Imagery VI, April 2002.
2. AFRL Sensor Data Management System (SDMS) web site: https://www.mbvlab.wpafb.af.mil/public/sdms/.
3. AdaptSAPS site on SDMS: http://www.mbvlab.wpafb.af.mil/public/sdms/datasets/adaptsaps/overview.htm.
4. Ross, T., M. Minardi, "Performance Assessment of Detector Reported Confidences", Proc. SPIE, Vol. 5427, Algorithms for SAR Imagery XI, April 2004.
5. MSTAR Public Data site on SDMS: https://www.sdms.afrl.af.mil/public/sdms/datasets/mstar/overview.htm.
6. Ross, T., J. Bradley, L. Hudson, M. O'Conner, "SAR ATR – So What's The Problem? – An MSTAR Perspective", Proc. SPIE, Vol. 3721, April 1999.
7. MathWorks File Exchange: http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId-3807&objectType=file#.