AFRL-RY-WP-TR-2010-1042

RANDOM SHAPE AND REFLECTANCE REPRESENTATIONS FOR 3D ASSISTED/AUTOMATED TARGET RECOGNTION (ATR)

B.M. Horowitz, P.A. Beling, 1.0. Reyes, and M.D. Devore

University of Virginia

FEBRUARY 2010
Final Report

AIR FORCE RESEARCH LABORATORY
SENSORS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7320
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RY-WP-TR-2010-1042 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

*//Signature//

KELLY MILLER, Program Manager
Assessment & Integration Branch

//Signature//

VINCENT VELTEN, Branch Chief
Assessment & Integration Branch

//Signature//

CHRIS RISTICH, Division Chief
Sensor ATR Technology Division

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

*Disseminated copies will show "//Signature//" stamped or typed above the signature blocks.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
February 2010	Final	05 March 2007 – 05 September 2009

4. TITLE AND SUBTITLE
RANDOM SHAPE AND REFLECTANCE REPRESENTATIONS FOR 3D ASSISTED/AUTOMATED TARGET RECOGNTION (ATR)

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA8650-07-1-1113

5c. PROGRAM ELEMENT NUMBER
62204F

6. AUTHOR(S)
B.M. Horowitz, P.A. Beling, I.0. Reyes, and M.D. Devore

5d. PROJECT NUMBER
6095

5e. TASK NUMBER
17

5f. WORK UNIT NUMBER
609517AG

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Virginia
1001 N. Emmet St.
Charlottesville ,VA 22903-4833

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Sensors Directorate
Wright-Patterson Air Force Base, OH 45433-7320
Air Force Materiel Command
United States Air Force

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/RYAA

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-RY-WP-TR-2010-1042

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report contains color.

14. ABSTRACT
This document is the final report for research on ATR Center RASER Grant FA8650-07-1-1113. The objective of this project was to expand the capabilities of model-based assisted/automated target recognition (ATR) systems by explicitly accommodating variation in shape and reflectance across elements of a broad target class. Work is set in the context of three-dimensional point-cloud data sets, such as LADAR or other structured light methods, and builds off a data representation model that represents measurement uncertainty probabilistically. Under this data model, the likelihood that a particular target gave rise to an observed point cloud can be computed using a collection of numerical integrations over the surface of a model of a target. Selection of the target with the largest likelihood then yields the classification result with the minimum probability of error (MPE) that can be achieved using a given sample of observed points. Our focus is on the study of anytime ATR algorithms, which are structured to support classification result queries that are placed at unknown, arbitrary times. A naïve anytime algorithm based on the MPE decision rule can be defined in terms of round-robin calculations of likelihoods for observed points.

15. SUBJECT TERMS
Model-based assisted/automated target recognition, LADAR, Minimum probability error

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor)
a. REPORT	b. ABSTRACT	c. THIS PAGE			Kelly Miller
Unclassified	Unclassified	Unclassified	SAR	28	19b. TELEPHONE NUMBER *(Include Area Code)* N/A

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

Table of Contents

This material is based on research sponsored by AFRL/RYAA under agreement FA8650-07-1-1113. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

RANDOM SHAPE AND REFLECTANCE REPRESENTATION FOR 3-D ASSISTED/AUTOMATED TARGET RECOGNITION

1. Summary

This document is the Final Report for research on ATR Center RASER Grant FA8650-07-1-1113. The objective of this project was to expand the capabilities of model-based assisted/automated target recognition (ATR) systems by explicitly accommodating variation in shape and reflectance across elements of a broad target class. Work is set in the context of three-dimensional point-cloud data sets, such as LADAR or other structured light methods, and builds off a data representation model that represents measurement uncertainty probabilistically. Under this data model, the likelihood that a particular target gave rise to an observed point cloud can be computed using a collection of numerical integrations over the surface of a model of a target. Selection of the target with the largest likelihood then yields the classification result with the minimum probability of error (MPE) that can be achieved using a given sample of observed points. Our focus is on the study of anytime ATR algorithms, which are structured to support classification result queries that are placed at unknown, arbitrary times. A naïve anytime algorithm based on the MPE decision rule can be defined in terms of round-robin calculations of likelihoods for observed points. Our approach to improving the performance of such a method is to use offline calculation of shape and reflectance-based performance estimates to to guide the sequence in which likelihoods are calculated and used. Both analytical results and numerical experiments on simulated data support the conclusion that pair-wise performance estimates can be used to achieve a substantial improvement in the classification accuracy as a function of computing time, relative to the naïve round-robin algorithm.

2. Introduction

For several years there has been a general interest in the assisted/automated target recognition (ATR) community in methods that permit a direct mapping of hierarchical target taxonomies into recognition systems. Efforts to develop such methods have met with varying success in specific cases, but as yet there seems to be no clear dominant approach that can be practically applied across a wide range of target families. The objective of our project, ATR Center RASER Grant FA8650-07-1-1113, is to contribute in this area by expanding the capabilities of model-based ATR systems using a method for explicitly accommodating variation in shape and reflectance across elements of a broad target class.

Classification of objects from three-dimensional measurement systems, such as LADAR, IFSAR, and optical stereo, is becoming increasingly important as development of these sensors advances. Our work is set in the context of three-dimensional point-cloud data sets and builds off a data representation model that represents measurement uncertainty probabilistically, with structure that can account for a variety of error sources, including target motion during imaging, nonzero optical footprint of the sensors, and misregistration of data from multiple sensors in a networked environment. Under this data model, the likelihood that a particular target gave rise to an observed point cloud can be computed using a collection of numerical integration over the

surface of a model of the target. Selection of the target with the largest likelihood then yields the classification result with the minimum probability of error (MPE) that can be achieved using a given sample of observed points. The researchers on this project have developed a body of work on the MPE decision rule. A basic classification algorithm for the 3D data sets is developed in [1] and analytical performance results are demonstrated. This basic algorithm is adapted to target recognition in highly cluttered environments in [2]. The result is a joint segmentation-classification algorithm that explicitly accounts for ground clutter and obscuring objects in the environment. In [3], basic design questions are addressed such as the required number of points on target, allowable pose uncertainty, and effect of standoff distance on target recognition from LADAR data. Related issues, such as evaluation of the kinds of approximations necesssary to apply dynamic search algorithms over pose resolution hierarchies are addressed in [4] and [5].

The MPE algorithm developed in [1, 2, 3, 4, 5] has a number of desirable theoretical properties, including immunity to peaking phenomena, robust behavior in the presence of clutter and small unknown location and pose errors, and the capability to be extended to account for completely unknown pose. In this project, our focus is on the study of *anytime* ATR algorithms, which are structured to support classification result queries that are placed at unknown, arbitrary times. Anytime algorithms are distinct from *contract* algorithms, which are structured to exploit a fixed amount of time (or operations) available for computation. A naïve anytime algorithm based on the MPE decision rule can be defined in terms of round-robin calculations of likelihoods for randomly ordered sample points. The contract MPE algorithm is theoretically optimal for a given number of sample points in the sense that no greater accuracy can be achieved, under the assumed data model. The algorithm uses likelihoods that are, in general, uniformly expensive to compute but need not be uniformly informative. This observation raises the possibility there are methods for sequencing likelihood calculations that are superior to the naïve approach in accuracy as a function of computing time.

Our approach to ordering likelihood calculation uses offline calculation of shape and reflectance-based performance estimates for pairs of objects from the target library. These performance estimates effectively provide an approximation to the accuracy versus computation time tradeoff in a classification problem defined over a pair of targets. The notion of the anytime algorithm is that, given a fixed amount of time available for computation, it is more efficient to spend less time on targets that are easy to differentiate and more time on targets that are difficult to differentiate. Our anytime algorithm, which we call the performance-based tree (PBT) algorithm, uses pairwise performance estimates to seed a tournament-style tree structure. We performed a number of numerical tests of the PBT algorithm using polygonal models of basic geometric shapes (such as spheres) and a variety of air and ground vehicles. Results and numerical experiments on simulated data support the conclusion that pair-wise performance estimates can be used to achieve a substantial improvement in classification accuracy as a function of computing time, relative to the naïve round robin algorithm.

The reminder of the report is organized as such: The following subsections outline the probability model for point cloud measurement, the minimum probability of error decision rule, and the PBT algorithm. Section 3 covers the assumptions, procedure, and simulation environment used to evaluate the PBT algorithm against the naïve round-robin approach. Section 4 presents the results from these experiments. Section 5 outlines a novel approach to classification that is particularly useful when the incremental cost of processing additional is

significantly greater than the incremental cost of collecting the data. Finally, Section 6 concludes the report with a discussion of the results and proposes future work based on these findings.

2.1 Data Model and Minimum Probability of Error Decision Rule

This section provides a brief introduction to the data model and decision rule that are the foundation for the research described later. Fuller treatment of these topics may be found in [1, 2, 3, 4, 5].

Let the set of points comprising the surface of an object be denoted by S, and let $X^* \in S$ be a point on the surface given 3D coordinates as $X^* = [X_1^*, X_2^*, X_3^*]^T$, where superscript T denotes transpose. We assume this point is selected from the surface for measurement according to a conditional probability density $p_{\mathbf{x}^*|\boldsymbol{\Theta},S}(\mathbf{x}^*|\boldsymbol{\Theta},S)$, where $\boldsymbol{\Theta}$ represents the pose (location and rotation) of the object relative to the measurement platform. Let the measured location of the surface point $\mathbf{X}^*$ be denoted by $\mathbf{X} = [X_1, X_2, X_3]^T$. We will refer to X^* as the point of origin for the measurement X, and model the observation as $X = \boldsymbol{\Theta} * \mathbf{X}^* + N$, where $\boldsymbol{\Theta} * \mathbf{X}^*$ represents the relative pose transform applied to $\mathbf{X}^*$ and gives the coordinates of the measured point in the same reference coordinate system as the measurement, and where N is a multivariate Gaussian distribution with zero mean. It follows that $\mathbf{X}$ is distributed as a multivariate Gaussian random variable with mean $\boldsymbol{\Theta} * \mathbf{X}^*$ and variance $\sum_{\Theta,S}$ is a 3 x 3 covariance matrix. The conditional probability density function for X is

$$p_{\mathbf{X}|\boldsymbol{\theta},S,\mathbf{X}^*}(\mathbf{x}|\boldsymbol{\theta},S,\mathbf{x}^*)=\frac{1}{(2\pi)^{3/2}|\Sigma_{\boldsymbol{\theta},S}|^{1/2}}\exp\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\theta}\star\mathbf{x}^*)^T\Sigma_{\boldsymbol{\theta},S}^{-1}(\mathbf{x}-\boldsymbol{\theta}\star\mathbf{x}^*)\}.$$

Equation 1.

The posterior distribution for a measured point from surface S with pose $\boldsymbol{\Theta}$ is

$$p_{\mathbf{X}|\boldsymbol{\theta},S}(\mathbf{x}|\boldsymbol{\theta},S)=\int_{\mathbf{x}^*\in S} p_{\mathbf{X}|\boldsymbol{\theta},S,\mathbf{X}^*}(\mathbf{x}|\boldsymbol{\theta},S,\mathbf{x}^*)p_{\mathbf{X}^*|\boldsymbol{\theta},S}(\mathbf{x}^*|\boldsymbol{\theta},S)d\mathbf{x}^*.$$

Equation 2.

A worst-case scenario is to assume no prior information about the sensor line-of-sight, and the measured points are modeled as uniformly distributed over the object's surface. Another worst-case assumption is that there is no directional preference in the measurement noise, which may be modeled by letting $\sum$ be a diagonal matrix with equal diagonal elements.

Given $\boldsymbol{\Theta}$, the pose, of the measured object relative to the measurement platform, and given that the set of measured surface points is independent, the likelihood of a point-cloud $\chi = \{\mathbf{X}_k\}^K_{k=1}$ is

$$p_{\mathcal{X}|\boldsymbol{\theta},S}(\mathbf{x}|\boldsymbol{\theta})=\prod_{k=1}^{K} p_{\mathbf{X}|\boldsymbol{\theta},S}(\mathbf{x}_k|\boldsymbol{\theta},S).$$

Equation 3.

The recognition problem of interest is to determine which object out of the target set

resulted in the set of measured surface point locations. The minimum probability of error (MPE) decision rule, which we adopt, chooses the object that maximizes the likelihood of the observations. That is, we select the surface index that solves

$$\max_{m=1,2,\ldots,M} P_m \prod_{k=1}^{K} p_{\mathbf{X}|\boldsymbol{\theta},S}(\mathbf{x}_k|\boldsymbol{\theta},S_m),$$

Equation 4.

where P_m is the prior associated with surface *m*. The assumption that $\boldsymbol{\Theta}$ is known may not be reasonable in all circumstances. In practice, it would be desirable to combine known pose algorithms with a search over the space of possible locations and orientations, as is described in [4] and [5]. Note that no distinction is made between points on an object and points on the background or other clutter. In [2] an approach to dealing with clutter is developed that uses the algorithms to jointly infer the presence of an object and the environment in which it resides (background structure, obscuration, etc.). An alternative approach is to employ a segmentation algorithm, eliminating points that are not likely from one of the objects under consideration [5].

2.2 Pairwise Performance Estimation

The approach to performance estimation that is outlined in this section was first proposed in [5], and is extended in [6]. This method is concerned with finding the conditional probability of correct classification between pairs of hypotheses, from which we build anytime algorithms for multiple hypothesis testing as described in Section 2.3. In some cases, closed-form expressions for the conditional probability of correct classification can be derived directly from the underlying statistical distributions. With synthetic aperture radar (SAR) imagery data, to take an example, DeVore considered a recognition model that allows analytic computation of the conditional probability of correct classification [7]. In general, however, we will be limited to finding estimators for the pair-wise conditional probability of correct classification, along with upper and lower bounds on this probability. We derived these estimators using log-likelihood ratios computed from sample data.

Let $p_{\mathbf{x}^*|\boldsymbol{\Theta},S}(\mathbf{x}^*|\boldsymbol{\Theta},S)$ be the probability density function for an observation $\mathbf{X}$ given that the surface S is measured with relative pose $\boldsymbol{\Theta}$. Let $\chi = \{\mathbf{X}_k\}_{k=1}^{K}$ be a point cloud that is measured from either S_1 or S_2, the two possible object surfaces in this problem. For simplicity, we assume the two objects have equal prior probability. Then the minimum probability of error decision rule can be written as a single inequality; i.e. we would choose S_1 if and only if $L(X_1, X_2,\ldots,X_K) \geq 0$, where

$$L(X_1,X_2,\ldots,X_K)=\sum_{k=1}^{K} L_k=\sum_{k=1}^{K}\left[\ln p_{\mathbf{X}|\boldsymbol{\theta},\mathcal{S}}(X_k|\boldsymbol{\theta},S_1)-\ln p_{\mathbf{X}|\boldsymbol{\theta},\mathcal{S}}(X_k|\boldsymbol{\theta},S_2)\right].$$

Equation 5.

Because L_k are independent and identically distributed, $\mathbf{E}[L|\boldsymbol{\Theta}, S_m]$, and $\text{var}(L|\boldsymbol{\Theta},S_m) = K\text{var}(L_k|\boldsymbol{\Theta},S_m)$. The conditional mean and variance of L_k given S_m and $\boldsymbol{\Theta}$ can be written as

$$\mathbf{E}[L_k|\boldsymbol{\theta},S_m]=\int_{X_k\in\mathbf{R}^3}L_k p_{\mathbf{X}|\boldsymbol{\theta},S_m}(X_k|\boldsymbol{\theta},S_m)dX_k$$
$$\mathrm{var}(L_k|\boldsymbol{\theta},S_m)=\int_{X_k\in\mathbf{R}^3}L_k^2 p_{\mathbf{X}|\boldsymbol{\theta},S_m}(X_k|\boldsymbol{\theta},S_m)dX_k-\mathbf{E}^2[L_k|\boldsymbol{\theta},S_m].$$

Equation 6.

The variable L is a sum of independent, identically distributed random variables and so by the central limit theorem can be approximated as a Gaussian distribution with mean $\mathbf{E}[L|\boldsymbol{\Theta}, S_m]$ and variance $\mathrm{var}(L|\boldsymbol{H},S_m)$ for large K. When the object being observed is S_1, we make a correct decision if $L > 0$. Thus the approximate conditional probability of a correct decision is

$$\Pr[\mathrm{correct}|\boldsymbol{\theta},S_1]\approx 1-\Phi\left(-\frac{\mathbf{E}[L|\boldsymbol{\theta},S_1]}{\sqrt{\mathrm{var}(L|\boldsymbol{\theta},S_1)}}\right)=1-\Phi\left(-\frac{\sqrt{K}\mathbf{E}[L_k|\boldsymbol{\theta},S_1]}{\sqrt{\mathrm{var}(L_k|\boldsymbol{\theta},S_1)}}\right),$$

Equation 7.

where Φ is the cumulative distribution function for a standard Gaussian random variable.

From the above, we know that an estimate of the probability of correct classification can be formed from $\mathrm{E}[L_k\,|\,\theta,S_1]/\sqrt{\mathrm{var}(L_k\,|\,\theta,S_1)}$. To obtain this ratio, we generate N sample points $\{X_n\}$, $n = 1, \ldots N_1$ from S_1 and then compute the log-likelihood ratio

$$l_n=\ln p_{\mathbf{X}|\boldsymbol{\theta},\mathcal{S}}(X_n|\boldsymbol{\theta},S_1)-\ln p_{\mathbf{X}|\boldsymbol{\theta},\mathcal{S}}(X_n|\boldsymbol{\theta},S_2),$$

Equation 8.

for $n = 1,\ldots,N$. The sample mean and sample variance of the log-likelihood ratio are then respectively

$$\bar{l}=\frac{1}{N}\sum_{n=1}^{N}l_n \quad \text{and} \quad \sigma_l^2=\frac{1}{N}\sum_{n=1}^{N}(l_n-\bar{l})^2$$

Equation 9.

Then $\mathrm{E}[L_k\,|\,\theta,S_1]/\sqrt{\mathrm{var}(L_k\,|\,\theta,S_1)}\approx \bar{l}/\sigma_l$. From [8], a 1 – α confidence interval is

$$\Pr\left[\bar{l}/\sigma_l+\Phi^{-1}(\alpha/2)/\sqrt{N}<\frac{\mathbf{E}[L_k|S_1]}{\sqrt{\mathrm{var}(L_k|S_1)}}<\bar{l}/\sigma_l-\Phi^{-1}(\alpha/2)/\sqrt{N}\right]=1-\alpha.$$

Equation 10.

Therefore, we can compute the estimated accuracy $\hat{p}$ (i.e. conditional probability of correctly classifying S_1), and the lower bound $\hat{p}_l$ and upper bound $\hat{p}_u$ (i.e. the two bounds together provide the 1 - α confidence interval for the accuracy) as follows

$$\hat{p}=1-\Phi\left(-\sqrt{K}\frac{\bar{l}}{\sigma_l}\right),$$
$$\hat{p}_l=1-\Phi\left(-\sqrt{K}\left(\bar{l}/\sigma_l+\Phi^{-1}(\alpha/2)/\sqrt{N}\right)\right),$$
$$\hat{p}_u=1-\Phi\left(-\sqrt{K}\left(\bar{l}/\sigma_l-\Phi^{-1}(\alpha/2)/\sqrt{N}\right)\right).$$

Equation 11.

The estimations outlined above are based on log-likelihood ratio samples, which can be conveniently simulated based on the targeted operation scenario or acquired from actual measurements. This sampling method, though it does not produce estimates as accurate as those that may be found by extensive measurements, requires little computation and therefore may be suitable when a quick and rough answer is needed. Creating pair-wise estimates is an operation that can be done offline ahead of time and stored in a database before deployment of a system. If a system has multiple sensors or a single sensor that operates at various ranges causing various levels of measurements, then a set of performance estimates must be created for each sensor scenario. These performance estimates may be used repeatedly in the algorithm described in Section 2.3, provided the target library and sensors do not change.

We tested the proposed performance estimators on simulated data from three different operational scenarios of varying noise characteristics (low, medium, and high) and for different pairs of vehicles. The results show that the estimated accuracy is generally lower than the accuracy obtained through extensive simulations in the low and medium noise scenarios. This effect is a consequence of the fact that our central limit theorem based method is inclined to underestimate actual performance when most of the log-likelihood ratio samples are greater than 0, as occurs in these two scenarios when the number of points on target are small. This tendency to underestimate also causes the estimated lower bound in the high noise scenario to dominate the estimated lower bounds in the low and medium scenarios when classifying pairs of vehicles that have very different sizes and shapes. The results also have shown that more log-likelihood ratio scenarios can help improve precision of the estimated accuracy and reduce the confidence interval length. The lower bounds produced by this method tend to underestimate system performance, and so are often suitable when conservative estimates are needed.

2.3 Anytime algorithms

This subsection describes two anytime ATR algorithms, each of which is structured to support classification result queries that are placed at unknown, arbitrary times. Anytime algorithms are distinct from contract algorithms, which are structured to exploit a fixed amount of time (or operations) available for computation.

2.3.1 UC-MPE Algorithm

A naïve anytime algorithm based on the MPE decision rule can be defined in terms of round robin calculations of likelihoods for randomly ordered sample points. We term this the uniform computation, minimum probability of error (UC-MPE) algorithm. When queried for an answer, UC-MPE will return the MPE target, which is the target with the maximum aggregate likelihood with respect to the sample points considered up to the time of the query. Note that, if

run sufficiently long, UC-MPE will converge in output to the contract MPE algorithm of Section 2.1. UC-MPE serves as a baseline against which to measure the performance of the PBT tree algorithm introduced below.

2.3.2 PBT Multi-tree Algorithm

The contract MPE algorithm is theoretically optimal for a given set of sample points in the sense that no greater accuracy can be achieved, under the assumed data model. The algorithm uses likelihoods which are, in general, uniformly expensive to compute but need not be uniformly informative. This observation raises the possibility there are methods for sequencing likelihood calculations that are superior to the naïve approach in accuracy as a function of computing time. Our approach to ordering likelihood calculation uses offline calculation of pairwise performance estimates in the manner described in Section 2.2. These performance estimates effectively provide an approximation to the accuracy versus computation time tradeoff in a classification problem defined over a pair of targets. The notion of our anytime algorithm, which we call the performance-based tree (PBT) algorithm, is to use pairwise performance estimates to seed a tournament-style tree structure, with the motivation that given a fixed amount of time available for computation it is more efficient to spend less time on targets that are easy to differentiate and more time on targets that are difficult to differentiate. The PBT algorithm is further described in [23, 24].

The PBT algorithm produces a final classification result by running a hierarchical tournament based on hypothesis tests on pairs of targets. Pairwise performance estimates (computed using the method of Section 2.2) are used to determine the amount of computation to devote to each hypothesis test. This is done by fixing a desired classification accuracy level $\rho \geq 0.5$ and then, for each target pair, retrieving the estimated number of likelihoods needed for to achieve that accuracy level for the pairwise classification problem. The base level for the tournament tree consists of all targets in the library, organized into pairs (methods for choosing base level pairs are described below). Winners of each pairwise comparison advance to the next level of the tree. Because they are expensive to compute, likelihoods that were calculated in prior levels of the tree are saved and reused where possible. The final pairwise comparison yields the classification answer for the entire problem.

Figure 1 shows an example of the tournament structure in the PBT algorithm. The desired pairwise classification accuracy, ρ, is 0.75. The pairwise performance curve for Car A and Tank C dictates likelihoods for 5 sample points be used to achieve this accuracy. Likewise, likelihood for 25 sample points are used for Tank A and Tank B. If Tank B and Tank C are the maximum likelihood winners of their pairwise comparisons in the base level, then they are compared at the next level, using likelihood for 50 sample points to again attempt to achieve a pairwise accuracy of 0.75. If Tank B wins that comparison it becomes classification output by the algorithm.

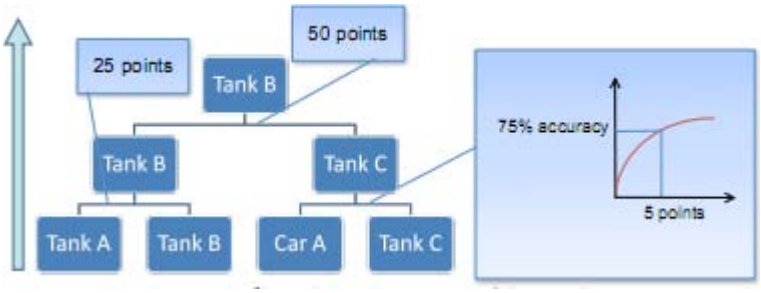

Figure 1. Single Pass Tree Algorithm

The multi-pass version of the PBT algorithm involves a sequence of tournaments in which the desired pairwise accuracy, ρ, in increased in each tournament by a fixed amount δ. If δ is sufficiently small then multi-pass PBT may be considered to be an anytime algorithm, as a negligible amount of computation will be wasted if the algorithm is queried for an output in the midst of a pass through the tree. Each completed pass through the tree yields a classification. Just as likelihoods were reused from level to level of the tree in the single pass PBT algorithm, likelihoods will be reused from one tree pass (tournament) to the next. The process continues until the algorithm is stopped on the basis of a query for a final answer or until all sample points have been exhausted. Figure 2 shows an example of the tournament structure in the multi-pass PBT algorithm and Figure 3 illustrate the notion of likelihood reuse between tournaments.

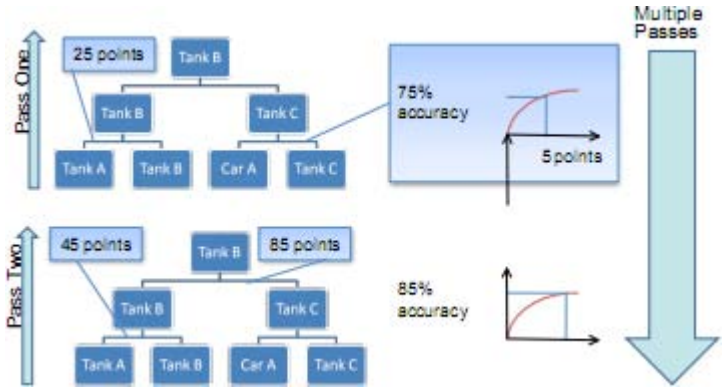

Figure 2. Multiple Pass Tree Algorithm

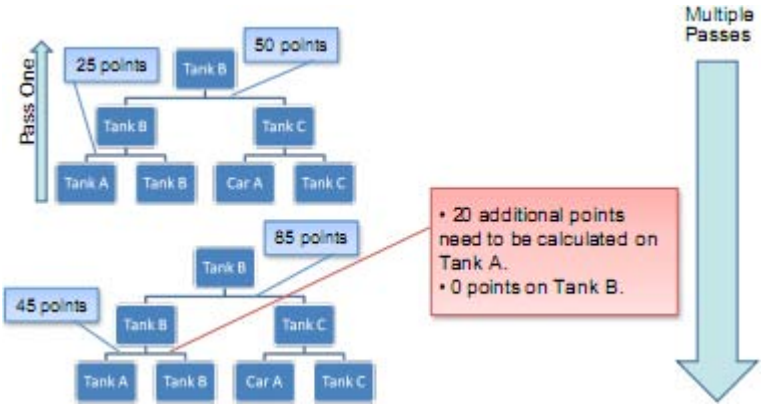

Figure 3. Likelihood Reuse in Multiple Pass Tree Algorithm

The base level for the tournament tree consists of all targets in the library, organized into pairs. Pairings can be done randomly or arbitrarily, but given the availability of pairwise performance estimates it may be desirable to create pairs so that the overall number of likelihoods used in the base level is as small as possible. This can be done by finding a minimum weight perfect matching on a complete graph with targets as the nodes and edge weights that are the performance estimate of the number of likelihoods needed to achieve the desired accuracy for the corresponding pairwise classification problem. A perfect matching in a graph G is a subset of edges such that each node in G has exactly one incident edge from the subset. Given a real weight w_e for each edge e of G, the minimum-weight perfect matching problem is to find a perfect matching of minimum weight. Minimum-weight perfect matching has low-order polynomial computational complexity [9, 10].

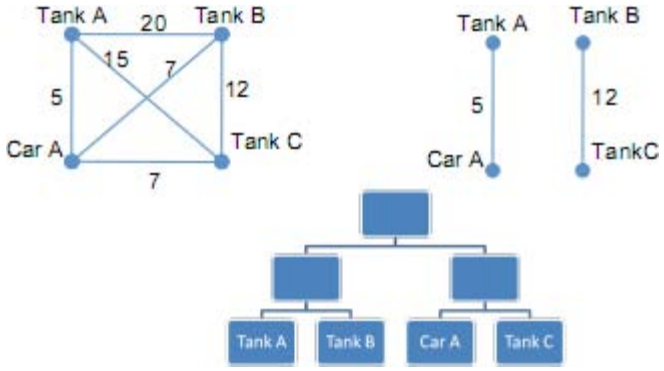

Figure 4. Target Pairing in PBT Algorithm

3. Methods, Assumptions, and Procedures

Simulations were run using as targets both spheres and CAD models of a variety of land and air vehicles. Spheres with radial Gaussian noise were used because the basic shape trivializes likelihood calculations. Sphere likelihood calculations run approximately 150 times faster than CAD model calculations, which allowed for a greater variety of scenarios to be tested in a reasonable amount of time. Performance estimates for spheres can be derived analytically and are exact. CAD models have more complex shapes so there is more of an argument for algorithms detecting shape differences in addition to size differences. CAD models may be used with a LADAR simulator, which allows for the future possibility of incorporating more realistic noise models.

CAD models of approximately 50 air and land vehicles were supplied by the Air Force Research Lab for its 2003 3-D Challenge Problem. Based on shape and size, the targets in this library can naturally be divided into classes (e.g., cars, aircraft, tanks). Note that the pairwise performance estimates described in Section 2 provide a structured method for class division. Standard techniques were used to either combine polygons, in cases where the fidelity of the CAD model exceeded that required for acceptably accurate numerical surface integration in likelihood calculation (as determined by empirical studies in [5]), or split them in converse cases. Simulated point cloud samples were obtained by using these CAD models as inputs to a LADAR simulator [11] set to use spherical Gaussian noise. Repeated sampling from the CAD model at

varying poses was used to produced point clouds that were both of the desired size and uniformly sampled over the surface. For each target in the library, we generated a point cloud sample of 50,000 simulated measured points, and then computed the likelihood of each point with respect to each target in the library; thus a pool of 50,000 x 50 likelihoods were available for experimentation. Performance estimates for the CAD model targets were computed using 1000 sample points and the estimation method described in Section 2.2.

Experiments with the PBT multi-tree algorithm were run by first setting the variables: single pass or multi pass tree, standard decision rule or alternate decision rule, and minimum weight perfect matching pairings or randomized pairings. The number of replications performed on each experiment was chosen on the basis of the degree of smoothness desired in plots of averaged accuracy versus computation time, but 500 to 10,000 replications were typical. Experiments with the UC-MPE algorithm were performed in a similar way, with all likelihood coming from the same pre-computed pool. Figure 5 provides an example output showing recognition accuracy as a function of likelihoods consumed for the PBT multi-tree Tree and the UC-MPE algorithm on a target library containing four vehicle classes, each with four members.

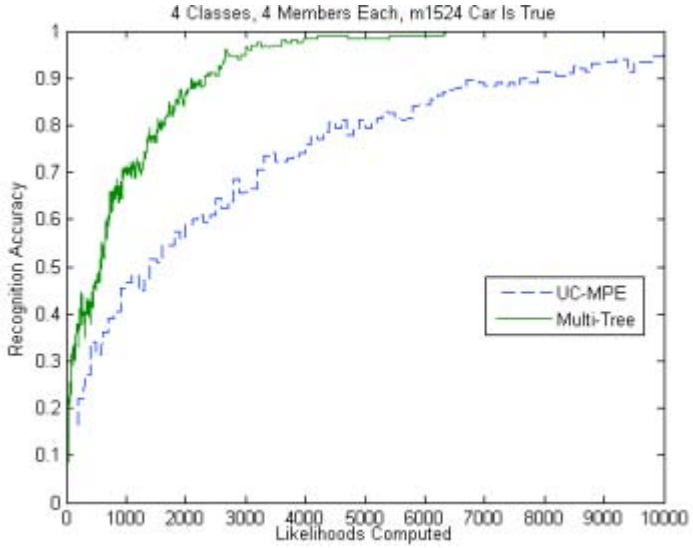

Figure 5. Recognition Accuracy as a Function of Likelihoods Consumed for a Target Library Containing Four Vehicle Classes, Each With Four Members

4. Results and Discussions

Numerical tests in the simulation environment yield several conclusions. Minimum weight perfect matchings give better performance than a random pairing, as one would expect. The trade-off of using minimum weight matchings is highly depending on the size of the target library, the average savings (influenced by the degree of differences in targets) provided by the minimum weight perfect matchings, and the complexity of the surface models (number of polygons in the CAD models). The single pass PBT algorithm did outperform the multi-pass variant, however when the minimum weight perfect matchings were used, the difference was nonexistent in some scenarios and much smaller in other scenarios. The multi-pass algorithm also has the significant benefit of being an anytime algorithm. Both versions of PBT generally

outperform UC-MPE. Using performance estimates to choose which targets have likelihoods computed on them indeed increased performance over the alternative method of using equal computation on all targets.

Using spheres as targets showed that as the target library size increases, the PBT algorithm has increasingly larger advantages in performance over the UC-MPE algorithm. The hypothesis is that if there are more target classes in the library, then more targets can be easily eliminated because all classes except the one containing the true target will be easy to distinguish from the true target. The sphere experiments suggested that a larger number of distinct target classes in the library will cause the BPT algorithm to increasingly outperform the UC-MPE algorithm. Some unexpected issues and complications were encountered after adapting the experimental test bench from one whose target library consisted of simulated spherical targets of varying radii to one with a library of 3D CAD models of a variety of real vehicles. The transition process itself is straightforward: in the experimental simulation scripts, replace all references to runtime-generated sphere data with offline-compiled vehicular target information (i.e., from sphere generation and online likelihood calculations to reference CAD models and likelihood tables produced from simulated LADAR point clouds). Running these benchmarks on real targets, however, produced unforeseen results that under certain circumstances significantly diverged from those observed with the sphere libraries. In particular, although the sphere tests demonstrated that the tree algorithm always outperformed the UC-MPE approach in terms of recognition accuracy as a function of likelihoods computed, experiments on real reference targets showed that this is not always the case.

It is inferred from this observation that the targets' (both the measured object and the reference library) respective geometries may lead to situations where the tree algorithm's motivation of increasing the number of low-likelihood-cost comparisons actually produces worst-case comparisons far more costly than its UC-MPE counterpart. For example, suppose that the measured point cloud arose from a tank and that the tree algorithm is attempting to determine in one branch which of two highly similar pickup trucks most likely produced the cloud; the performance estimation curves between those two trucks will cause the algorithm to request a large number of likelihood computation due to those vehicles' geometric similarity, and because the measured data came from neither of those reference targets, all that additional computation will be for naught.

---

5. Sequential Hypothesis Testing

As part of this effort, the research team developed a novel approach to classification that is particularly useful when the *incremental cost of processing additional is significantly greater than the incremental cost of collecting the data.* This situation can occur when high-level inferences are to be drawn from massive data collections. For example, the incremental cost of collecting one image from a high-resolution sensor on a loitering UAV platform is very small compared to the cost of extracting and fusing relevant information from that image. When timing is critical, it may make sense to selectively *not* process (or not fully process) all the data that has been collected.

This is a situation that apparently has not been previously discussed in the literature on statistical hypothesis testing. Many authors in both the statistics and pattern recognition communities have developed sequential inferencing algorithms that allow, among their possible

outcomes, a decision that more data must be collected to meet pre-specified accuracy requirements. However, these algorithms fully process all the data available at each stage. In contrast, while algorithms developed as part of this project can similarly infer that more data must be collected, they can subsequently limit the extent to which those data are processed.

This is a concept that is implicitly incorporated into most practical pattern recognition systems. For example, vision-based systems frequently segment out the most relevant regions of an image for analysis and disregard the rest. The algorithms developed in this project formalize the concept in the context of target classification, in which additional measurement data may be irrelevant with respect to some target hypotheses but highly relevant for others. The closest problem addressed in the statistics literature seems to be that of choosing the best available treatment. In these problems, multiple trials are run in parallel involving different treatments, and the goal is to decide which treatment provides the best result. There is often a desire to terminate any trial as soon as sufficient evidence exists that it is not the best treatment, but to allow other trials to continue. This is different from the classification problem addressed in this research project, which seeks the hypothesis that best explains a given set of sensor data.

The approach developed by the research team is to construct hierarchical classification algorithm that operates by repeatedly comparing pairs of target hypotheses. At each stage, the number of hypotheses under consideration is reduced by half as the unlikely targets are eliminated from consideration. In each comparison, a table lookup provides the quantity of 3D data that must be processed to select between the two hypotheses at a user-specified correct classification rate, and only this much data is processed. Target hypotheses are paired for comparison at each stage of the algorithm in an attempt to minimize the total number of computations that must be performed at each stage. Great computational savings are possible because the early stages involve pairings of easily distinguished targets, and processing large quantities of data is postponed for later stages that involve only a small number of hypotheses.

5.1 Sequential Hypothesis Testing Theory

The goal of a hypothesis testing procedure is to choose from among several possible hypotheses the one that best accounts for an observed set of data, if any. A fixed sample size procedure makes this choice after processing all available data. A sequential procedure, on the other hand, processes a variable amount of data, making a decision at each stage as to whether the collection of additional data is warranted.

More formally, suppose that a measurement process yields a sequence of random variables X_1, X_2, ..., drawn independently from some distribution, which could be one of a $M < \infty$ known distributions or some other unknown distribution. The possibility that the data were drawn from the mth distribution is referred to as hypothesis m, denoted H_m. The possibility that the data was drawn according to some other unknown distribution is referred to as the null hypothesis, denoted H_0.

A fixed sample size test consists of a pair (N_F, ϕ_F), where $N_F < \infty$ is an integer constant indicating the number of sample observations to be processed, and ϕ_F is a decision rule (i.e., a function) mapping the collection of observations $X_1, X_2, \ldots, X_{N_F}$ to one of the M distributions. The decision rule has the following interpretation: If $\phi_F(X_1, X_2, \ldots, X_{N_F}) = m$, then H_m is asserted to be the best explanation for the observations. Of course, this assertion may be

incorrect, and the probability that the observations from class *j* are incorrectly declared to be from class *m* is $P_{j,m} = \Pr[\phi_F = m \mid H_j \text{ is true}]$. The *class conditional probability of error* is $\varepsilon_j = \Pr[H_j \text{ not chosen} \mid H_j \text{ not true}] = \sum_{m \neq j} P_{j,m}$.

In a sequential test the value N_S is itself a random variable, being a function of the observations. That is, a sequential test is a pair (N_S, ϕ_S), where $N_S(X_1, X_2, \ldots)$ is called a *stopping* rule, and ϕ_S, called the *final decision rule*, maps observations $X_1, X_2, \ldots, X_N$ to one of the *M* distributions. As with a fixed sample size test, we are concerned with the various error probabilities $P_{i,j}$ and ε_j. Additionally, we are concerned with the distribution of N_S when hypothesis H_j is true. Most sequential tests are characterized in terms of the average sample number (ASN), defined as $E[N_S]$.

For the two-hypothesis case, the sequential probability ratio test (SPRT) of Wald [12] has been shown to minimize the ASN over all tests that have class conditional probabilities of error no greater than some specified ε_1 and ε_2, regardless of which hypothesis is true. At each stage, the SPRT calculates the likelihood ratio of all available data and compares the result against two threshold values. If the likelihood ratio is smaller than the minimum threshold or greater than the maximum threshold, the corresponding hypothesis is selected. If the ratio lies between the two thresholds, the data are taken to be ambiguous, and additional data is collected. The test then repeats with additional data.

Many authors have reported on direct multi-hypothesis extensions of the basic SPRT [13, 14, 15, 16]. Unfortunately, it can also be shown that this type of optimality cannot extend to problems with more than two hypotheses. That is, no sequential test can minimize the ASN, subject to upper bounds on the probabilities of error, simultaneously across all true hypotheses [17, Sec. 9.2]. It has recently been shown that the Mulithypothesis SPRT (MSPRT), which is similar in from to earlier SPRT extensions, is asymptotically optimal for arbitrarily small error probabilities, ε_j [18, 19, 20]. Like the SPRT, all these tests compute the likelihood of the entire data collection under each hypothesis as every stage.

5.2 Sequential classification with Reject Option

In this section, we describe a new classification algorithm based on an alternative formulation of sequential multi-hypothesis test, in which a hypothesis is dropped from further consideration as soon as there is significant evidence that it is not correct. By dropping unlikely hypotheses from further consideration, the number of hypotheses, and thus the amount of processing for each new data sample, decreases with each stage. For some specific examples of this general approach, see [21,22].

Suppose that when H_m is true, and observed data sample *X* has probability density function (PDF) $f_m(x)$. Given a collection of conditionally independent observed data $X_1, X_2, \ldots, X_N$, denote the log-likelihood of H_m as the sum

$$L_m^n = \sum_{n=1}^{N} log(f_m(X_n))$$

Equation 12. Log-likelihood of H_m

Note that L_m^n is a function of the observed data, and so it is itself a random variable. Let $F_m^n(l)$ denote the cumulative distribution function (CDF) of L_m^n under the assumption that H_m is true. Optimal hypothesis testing algorithms are based on the fact that when H_m is not true, we expect the log-likelihood hypothesis L_m^n to be small. This implies that when H_m is not true, we expect $F_m^n(L_m^n)$ to be close to zero.

Working the other way, assuming that we want to determine whether or not H_m is true, we note that if $F_m^n(L_m^n) = \alpha$, then there is a 100α percent probability that *by pure chance alone* an even less likely sequence of data would be observed if H_m were in fact true. This can form the basis of a discrimination test for whether or not to drop H_m at stage *N*. For example, suppose we pick some arbitrarily small quantity $\alpha = 0.01$ and drop H_m at stage *N* from further consideration if $F_m^n(L_m^n) = \alpha$. We can then be confident that there is less than a 1% chance that we have incorrectly rejected H_m.

With this in mind, we define the sequential target classification procedure as follows.

1. Define α to be the largest tolerable false rejection rate for the problem

2. Initialize the stage number $n = 0$ and let $M^0 = \{1,2,\dots,M\}$ be the set of target classes initially under consideration

3. Increment *n* by one, collect observation X_n and compute L_m^n for each $m \in M$

4. Let $M^n = \{m \in M^{n-1} | F_m^n(L_m^n) > \alpha\}$ be the set of target classes still under consideration after stage *n*

5. If $|M^n| > 0$, report

   (a) $\hat{m} = argmax_{m \in M^n} L_m^n$ as the most likely hypothesis found at the end of stage *n*;
   (b) $F_{\hat{m}}^n(L_{\hat{m}}^n)$ as the *significance* of that hypothesis; and
   (c) M^n as the set of feasible alternatives.

6. If $|M^n| = 0$, report that all of the known target hypotheses have been rejected

7. If $|M^n| > 1$ go to step 3, otherwise terminate.

Note that, while the formulation is slightly different, the algorithm shares many properties of the pairwise hierarchical algorithm originally developed by the research team. In particular, it has an anytime capability, reporting the most likely target at each stage as well as the set of feasible alternative targets (i.e., those that cannot be ruled out at 100α significance). The sets M^n, for $n = 1,2,\dots$ for a dynamically constructed, nested set of target super-classes, similar to the nested classes produced by the pairwise algorithm. Also, it can be executed in a multi-pass function, with an initial α assignment that is gradually reduced.

Unlike the pairwise algorithm, this algorithm comes with a performance guarantee: When the algorithm has terminated there is less than a 100α percent chance that the correct hypothesis

was rejected, regardless of which hypothesis was actually correct. Moreover, this algorithm can correctly reject all target hypotheses in the case that observations were produced by a previously unknown target class.

5.3 Distribution of Log-Likelihoods

For the algorithm of the previous section to be practical, one must be able to evaluate the CDFs of the log-likelihoods L_m^n for each *m* and arbitrary values of *n*. In general, closed-form expressions of these functions will be difficult to determine. As an alternative, we can assume an approximate form and use sample log-likelihood values to find best-fit distribution consistent with this form. In this section we consider three possible alternatives that adopt this strategy.

When H_m is true, the quantity L_m^n is the sum of *n* independent, identically distributed random quantities, each with a distribution identical to that of L_m^1. Each of the methods below attempts to characterize the distribution L_m^1 and then turns that characterization into a corresponding characterization of L_m^n for arbitrary values of *n*.

5.4 Johnson Family

In this approach, we assume L_m^n follows a Johnson distribution for each value of *n*. The Johnson family is a four-parameter distribution that includes the Gaussian and log-normal distribution families as special cases. It has the property that for any valid combination of the first four central moments (mean, variance, skewness, kurtosis), a unique distribution exists within the family that possesses these moments.

To apply this method, we begin with the first four central moments of L_m^1, defined as

$$\begin{aligned}\hat{\mu}_1(L_m^1) &= E[L_m^1] \\ \hat{\mu}_2(L_m^1) &= E\left[\left(L_m^1 - \hat{\mu}_1(L_m^1)\right)^2\right] \\ \hat{\mu}_3(L_m^1) &= E\left[\left(L_m^1 - \hat{\mu}_1(L_m^1)\right)^3\right] \\ \hat{\mu}_4(L_m^1) &= E\left[\left(L_m^1 - \hat{\mu}_1(L_m^1)\right)^4\right]\end{aligned}$$

Equation 13. Central moments of L_m^1

In practice, these can be estimated from sample log-likelihood values. For other values of *n*, the first four moments of L_m^n can be found as

$$\begin{aligned}\hat{\mu}_1(L_m^n) &= n\hat{\mu}_1(L_m^1) \\ \hat{\mu}_2(L_m^n) &= n\hat{\mu}_2(L_m^1) \\ \hat{\mu}_3(L_m^n) &= n\hat{\mu}_3(L_m^1) \\ \hat{\mu}_4(L_m^n) &= n\hat{\mu}_4(L_m^1) + 3n(n-1)\hat{\mu}_2(L_m^1)^2\end{aligned}$$

Equation 14. Central moments of L_m^n

These estimated moments are then used to find a corresponding member of the Johnson family, and the significance values are calculated from the CDF of the resulting distribution.

5.5 Polynomial Chaos Distribution Families

One alternative to the approach described above is to replace the Johnson family, which has a fixed parameterization, with a family of truncated polynomial chaos distributions. Such a family can be truncated at any level, allowing for an arbitrary number of free parameters with which to fit the distribution of L_m^n. A polynomial chaos expansion of a random variable *L* with finite variance can be written as

$$L_m^n = \sum_{i=0}^{\infty} l_i \psi_i(Z)$$

Equation 15.

where equality is in the mean-square sense, the l_i are deterministic constants, *Z* is a random variable, and the functions ψ_i are polynomials forming a complete orthonormal basis with respect to the probability density function for *Z*. Many choices of *Z* and corresponding ψ_i are available, and the series above is provably convergent for each, as long as L_m^n has finite variance. Throughout this report, *Z* will represent a Gaussian random variable with zero mean and variance 1/2. The function ψ_i will be the *i*th normalized Hermite polynomial. In practice, we will truncate the above sum to a total of *I* + 1 terms,

$$L_m^n \approx \sum_{i=0}^{I} l_i \psi_i(Z)$$

Equation 16.

With this formulation, the approximating distribution of L_m^n is completely governed by the choice of *I* and coefficients l_i. A method of moments formulation that parallels the Johnson family approach can be developed for the polynomial chaos approximation. It can be shown that the *j*th raw moment of the approximation in (5) is

$$\mu_j' = \sum_{i_1=0}^{I} \sum_{i_2=0}^{I} \cdots \sum_{i_j=0}^{I} l_{i_1} l_{i_2} \cdots l_{i_j} E[\psi_{i_1}(Z)\psi_{i_2}(Z) \cdots \psi_{i_j}(Z)]$$

Equation 17.

The expected values above are constants and can be computed off-line. The moments are thus polynomial functions of the l_i for any given choice of *I.* In what follows we choose *I* = 3 to give us four free parameters to match the original Johnson method. Note that the polynomial chaos approach allows for an arbitrary choice of *I* and thus an arbitrary representation accuracy.

The moments of the polynomial chaos approximating distribution with *I* = 3 can be shown to be

$$\mu_1' = l_0$$
$$\mu_2' = l_0^2 + l_1^2 + l_2^2 + l_3^2$$

$$\mu_3' = l_0^3 + 3(l_1^2 + l_2^2 + l_3^2) + l_0 + l_2\left(3\sqrt{2}l_1^2 + 6\sqrt{3}l_3l_1 + \sqrt{2}(2l_2^2 + 9l_3^2)\right)$$
$$\mu_4' = l_0^4 + 6(l_1^2 + l_2^2 + l_3^2) + l_0^2 + 4l_2\left(3\sqrt{2}l_1^2 + 6\sqrt{3}l_3l_1 + \sqrt{2}(2l_2^2 + 9l_3^2)\right)l_0$$
$$+3l_1^4 + 15l_2^4 + 93l_3^4 + 36\sqrt{6}l_1l_3^3 + 42l_1^2l_3^2 + 4\sqrt{6}l_1^3l_3 + 6l_2^2(5l_1^2 + 8\sqrt{6}l_3l_1 + 31l_3^2)$$

Equation 18.

Equating these with the estimated moments of L_m^n from (3) yields a system of four polynomial equations in four unknowns, which can be solved to yield the desired coefficients. These coefficients completely characterize the approximation of L_m^n, and the conditional cumulative distribution function $F_m^n(L_m^n)$ can be evaluated in terms of them. A variety of software tools exist for solving systems of polynomials, and any of these can be used to automate this procedure.

A potential drawback of the above approach is that we are not guaranteed to find a set of real-valued coefficients that yield the desired moments (i.e., the l_i may be complex). One solution is to introduce an additional coefficient L_{I+1}, creating an overdetermined system of equations, and look for a real solution. Another is to reformulate the problem as one of minimizing $\left(\mu_4' - \mu_4(L_m^n)\right)^2$ subject to the constraints that all coefficients are real and $\mu_1' = \mu_1(L_m^n)$, $\mu_2' = \mu_2(L_m^n)$, and $\mu_3' = \mu_3(L_m^n)$. This approach may not be highly robust, and an alternative criterion may be preferred.

An alternative to the method of moments approach is based on the fact that the CDF is a sum of independent, identically distributed random variables can be expressed as an *n*-fold convolution. In general, if *X* and *Y* are independent random variables, the CDF of their sum can be expressed as

$$\begin{aligned} F_{X+Y}(a) &= \int_{-\infty}^{\infty}\int_{-\infty}^{a-x} fx(x)fy(y)dy\,dx \\ &= \int_{-\infty}^{\infty} fx(x)\int_{-\infty}^{a-x} fy(y)dy\,dx \\ &= \int_{-\infty}^{\infty} fx(x)F_Y(a-x)dx \\ &= (f_x * F_Y)(a) \end{aligned}$$

Equation 19.

Thus, for *n* = 2, the CDF $F_m^2(\lambda) = (f_m^1 * F_m^1)(\lambda)$, where f_m^1 is the PDF of L_m^1. We can generalize this result for any *n* as

$$F_m^n(\lambda) = \left[(f_m^1)^{*(n-1)} * F_m^1\right](\lambda)$$

Equation 20.

which involves the (*n* - 1)-fold self-convolution of f_m^1. This expression for the CDF can be efficiently computed via Fourier transform techniques.

5.6 Kernel Density Distribution Families

The approximation approaches discussed in the previous section are semi-parametric, meaning that they represent the desired distribution in terms of a finite collection of parameters, but the total number of parameter may be arbitrarily large. In this section we briefly discuss the family of kernel density distributions, originally introduced by Parzen. This method is based on a theorem stating that for a sequence of independent identically distributed random variables $\lambda_1, \lambda_2, \ldots$ and for any probability density function $g(l)$ that is symmetric, bounded, and has tails that go to zero sufficiently fast, the mixture density

$$\hat{f}_K(\lambda) = \frac{1}{K h_K} \sum_{k=1}^{K} g\left(\frac{\lambda_k - \lambda}{h_K}\right)$$

Equation 21.

converges in a mean square sense to the distribution of the λ_k whenever the constants h_k are chosen to go to zero sufficiently slowly. A similar statement can be made about the cumulative distribution function. A common choice of $g(.)$ is the standard Gaussian density function.

This suggests the following approach to modeling the distribution of L_m^n. First, use (10) to find the Parzen estimate for the distribution of L_m^1 based on samples of the single-point log-likelihood when H_m is true. Then, for arbitrary *n* use the expression in (9) to find a Parzen estimate CDF of L_m^n. The (*n* - 1)-fold self-convolution in that equation can be easily computed form the samples used to construct the distribution of L_m^1, so there is no need to employ Fourier transform techniques.

---

6. Conclusions

Our empirical experiments suggest that more accurate performance estimates will amplify the benefit of the PBT algorithm. Future research might focus on the use of more general distributional forms to represent sums of log likelihoods. A variant algorithm worth exploring would be to use performance estimates to reduce the number of CAD model polygons used in a likelihood computation. In the extreme case only one polygon would be used and the method will converge to the Minimum Sum of Squared Distances (MSSD) algorithm. The MSSD algorithm has some undesirable properties detailed in [5]. It would be worth investigating if utilizing performance estimates mitigates the undesirable properties, while retaining the performance gains. The performance estimates could be generated using a small trial result set, similarly to the performance estimates described in Section 2.1.

The sequential hypothesis testing approach has the side effect of dynamically constructing a nested set of target super-classes. That is, target hypotheses that survive until the latest stages of the algorithm tend to be very similar in shape to one another and the target itself. Additionally, by making several passes through the algorithm while gradually increasing the specified correct classification rate, an anytime capability can be realized. When operated in this way, a sequence of classification results are produced with increasing confidence levels at each pass through the target hierarchy. This approach also includes a *reject option* that allows the algorithm to assert that none of the known target hypotheses corresponds to a given set of data.

This is an important addition, because practical target recognition systems can be expected to be confronted with data from objects that were not anticipated when the algorithm was implemented. Additionally, the amount of data processed for each hypothesis is a function of a data itself. This is in contrast to the PBT algorithm, in which the quantity of 3D data processed for each pairwise comparison is statically determined. Finally, the significance of results produced by the new method are easier to interpret because they follow directly from a user-specified maximum error rate. For example, the user could directly specify that the probability of wrongly rejecting the correct target hypothesis must be less than 1%, regardless of which hypothesis is actually correct. As before, a multi-pass variant of this algorithm is possible, in which the maximum allowable error rate is gradually reduced.

---

7. References

1. Zhou, X. and DeVore, M. (2007), "An analysis of data and model accuracy requirements for target classification using LADAR data", *IEEE Transactions on Aerospace and Electronic Systems*, under review.
2. DeVore, M. and Zhou, X. (2006), "Minimum probability of error recognition of three-dimensional laser-scanned targets", In F. A. Sadjadi (Ed.), *Automatic Target Recognition XVI, Proc. of SPIE*, 6234, pp. 43-53.
3. Zhou, X. and DeVore, M. (2005), "Shape recognition from point measurement with range and direction uncertainty", Optical Engineering, **44** (12).
4. DeVore, M. and Zhou, X. (2006), "Time-dependent valuation of maximum-likelihood approximation sequences", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, under review.
5. Zhou, X. (2008), *Statistical Model-Based Object Recognition from Three-Dimensional Point Cloud Data*, Ph.D. Dissertation, University of Virginia.
6. Zhou, X., Beling, P., DeVore, M., and Horowitz, B. (2009), "Estimation of pairwise classification performance in model-based ATR algorithms", manuscript, Department of Systems and Information Engineering, University of Virginia.
7. DeVore, M. (2004), "Analytical performance evaluation of SAR ATR with inaccurate or estimated models", In E. G. Zelnio and F. D. Garber, editors, *Algorithms for Synthetic Aperture Radar Imagery XI, Proc. of SPIE*, **5427**, pp. 407-417.
8. Sharma, K. and Krishna, H. (1994), "Asymptotic sampling distribution of inverse coefficient-of-variation and its applications", *IEEE Transactions on Reliability*, **43** (4), pp. 630-633.
9. Cook, W. and Rohe, A. (1999), "Computing minimum-weight perfect matchings", *INFORMS Journal on Computing*, **11** (2).
10. Gabow, H. N. (1990), "Data structures for weighted matching and nearest common ancestors with linking", *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 434-443.
11. Dixon, J. H. (2007), *LADAR Simulator User Manual* (Retrieved from http://users.ece.gatech.edu/~jdixon/ladar_sim/).
12. Wald, A. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117-186, June 1945.
13. Sobel, M. and Wald, A., A sequential decision procedure for choosing one of three

hypotheses concerning the unknown mean of a normal distribution. *The Annals of Mathematical Statistics*, 20(4):502-522, December 1949.
14. Armitage, P., Sequential analysis with more than two alternative hypotheses, and its relation to discriminant function analysis. *Journal of the Royal Statistical Society, Series B*, 12(1):137-144, 1950.
15. Simons G. A sequential three hypothesis test for determining the mean of a normal population with known variance. *The Annals of Mathematical Statistics*, 38(5):1365-1375, October 1967.
16. Lorden G. 2-SPRT's and the modified Kiefer-Weiss problem of minimizing an expected sample size. *The Annals of Statistics*, 4(2):281-291, March 1976.
17. Ghosh, B.K. and Sen, P.K., editors. *Handbook of Sequential Analysis*. Marcel Dekker, 1991.
18. Baum, C.W. and Veeravalli, V.V. A sequential procedure for multihypothesis testing. *IEEE Transactions on Information Theory*, 40(6):1994-2007, November 1994.
19. Dragalin, V.P., Tartakovsky, A.G., and Veeravalli, V.V. Multihypothesis sequential probability ratio test - part I: Asymptotic optimality. *IEEE Transactions on Information Theory*, 45(7):2448-2461, November 1999.
20. Dragalin, V.P., Tartakovsky, A.G., and Veeravalli, V.V. Multihypothesis sequential probability ratio test - part II: Accurate asymptotic expansions for the expected size. *IEEE Transactions on Information Theory*, 46(4):1366-1383, July 2000.
21. Lorden, G. Likelihood ratio tests for sequential k-decision problems. *The Annals of Mathematical Statistics*, 43(5):1412-1427, October 1972.
22. Hsu, J.C. and Edwards, D.G. Sequential multiple comparisons with the best. *Journal of the American Statistical Association*, 78(384):958-964, December 1983.
23. Redard, D. (2008), *A Performance-Estimation-Based Tree Algorithm for Reducing Computation in Automatic Target Recognition*, M.S. Thesis, University of Virginia.
24. Reyes, I., Beling, P., Devore, M., and Horowitz, B. (2009), “An anytime algorithm for ATR based on pairwise performance estimation”, manuscript, Department of Systems and Information Engineering, University of Virginia.