Computer Science
Technical Report

A Coregistration Approach to Multisensor Target Recognition with Extensions to Exploit Digital Elevation Map Data [*]

J. Ross Beveridge
Colorado State University
ross@cs.colostate.edu

Bruce A. Draper
Colorado State University
draper@cs.colostate.edu

Mark R. Stevens
Colorado State University
stevensm@cs.colostate.edu

Allen Hanson
University of Massachusetts
hanson@cs.umass.edu

Kris Siejko
Alliant Techsystems
siejko@rtc.atk.com

September 8, 1997

Technical Report CS-97-107

Computer Science Department
Colorado State University
Fort Collins, CO 80523-1873

Phone: (970) 491-5792 Fax: (970) 491-2466
WWW: http://www.cs.colostate.edu

This document is the Final Technical Report for Contract DAAH04-93-G-0422 administered by the Army Research Office. This manuscript also appears in a collection of final reports prepared for DARPA titled "Reconnaissance, Surveillance, and Target Acquisition for the Unmanned Ground Vehicle" published by Morgan Kaufmann Publishers.

[*] This work was sponsored by the Defense Advanced Research Projects Agency (DARPA) Image Understanding Program under grants DAAH04-93-G-422 and DAAH04-95-1-0447, monitored by the U. S. Army Research Office

A Coregistration Approach to Multisensor Target Recognition with Extensions to Exploit Digital Elevation Map Data [*]

J. Ross Beveridge
Colorado State University
ross@cs.colostate.edu

Bruce A. Draper
Colorado State University
draper@cs.colostate.edu

Mark R. Stevens
Colorado State University
stevensm@cs.colostate.edu

Allen Hanson
University of Massachusetts
hanson@cs.umass.edu

Kris Siejko
Alliant Techsystems
siejko@rtc.atk.com

September 8, 1997

Abstract

A target recognition capability is described that performs: color target detection, target type and pose hypothesis generation, and target type verification by 3-D alignment of target models to range and electro-optical imagery. The term 'coregistration' is introduced to describe target, range and electro-optical (color and IR) sensor alignment correction. Online model feature prediction using 3-D military vehicle models is demonstrated for 3-D vehicle models. All phases of the recognition cycle are shown on near-boresight-aligned electro-optical and range imagery collected at Fort Carson, Colorado. As a step toward integrating constraints from Digital Elevation Maps (DEM), an automated terrain feature prediction and matching capability is demonstrated. This terrain matching is used to refine DEM to ground-looking imagery registration.

1 Introduction

The goal of this project has been the development of new Automatic Target Recognition (ATR) algorithms that are more robust with respect to scene clutter, target occlusion and variations in viewing angle. The heart of the approach is to fuse range and electro-optical imagery (color and/or IR) using global geometric constraints. These constraints derive from known sensor, target and scene geometry. This may be thought of as model-based sensor fusion, and contrasts with more traditional approaches that attempt to fuse data based upon low-level cues only [20].

The roots of our approach lie in past alignment-based object recognition research [41; 31; 8]. In this line of research, the value of varying 3-D object to sensor alignment during recognition has been clearly demonstrated. While this paradigm is popular in many domains, it is surprisingly absent from work on ATR. Instead, ATR has been dominated by systems which employ fixed sets of image space templates or probe sets: sets of templates span the cross product of target models and sampled viewpoints.

Our multisensor target identification algorithm goes beyond traditional alignment by using online 3-D rendering to predict how target signatures change as a function of target pose (3-D position and orientation), lighting and terrain occlusion. This rendering component is coupled with a novel optimization algorithm in order to find the best target match. On a test suite of 35 image triples: Range, IR and Color, the system correctly distinguishes between an M113, M901, M60 and pickup truck in 27 out of the 35 tests. Only one other RSTA group has performed target identification on this dataset, and they report that using a template approach on Range Imagery alone they will reliably solve only 4 out of the 35 test cases [32].

The geometrically precise multisensor identification algorithm is computationally demanding. To reduce processing, focus-of-attention algorithms are used to perform detection and suggest possible target type and pose hypotheses.

[*] This work was sponsored by the Defense Advanced Research Projects Agency (DARPA) Image Understanding Program under grants DAAH04-93-G-422 and DAAH04-95-1-0447, monitored by the U. S. Army Research Office

Each of these upstream processes is itself a major component of our project. The target detection effort, led by the University of Massachusetts, has demonstrated the ability to detect camouflaged vehicles against similarly colored natural terrain. The detection algorithm uses new non-parametric classification techniques from the field of machine learning. The target and pose hypotheses generation effort is being led by Alliant Techsystems and the algorithm being used here is an adaptation of their own mature LADAR ATR system.

While combined range and electro-optical (E-O) data provides one valuable source of constraint for ATR, digital elevation maps (DEM) provide another. Scene context, as expressed in a DEM, can be used to guide search for targets by suggesting more or less likely regions of a scene to examine. Terrain maps can also provide range-to-pixel estimates for E-O sensors, provided that the DEM has been accurately registered to the E-O imagery. To use DEM constraints to greatest advantage, an automated process must accurately provide this registration. Our project is developing robust algorithms for performing such registration and demonstrating these on SSV data collected at the UGV Demo C site.

1.1 Review of Accomplishments

At the outset of the of the project, several milestones were established. Below are high-level descriptions of each and pointers sections in this chapter where each is discussed.

Data Collection. Over 400 range, IR and color images of military targets against natural terrain were collected at Fort Carson, Colorado. The imagery and documentation [7] has been approved for unlimited public distribution and is available through our website at `http://www.cs.colostate.edu/~vision`. (Section 2.3)

Target Detection Using Color. A real-time target detection system which learns to discriminate between natural terrain coloration and military camouflage (both green and brown) has been developed and demonstrated running on the UGV programs vehicles. In formal evaluation on the Fort Carson Dataset, the system finds roughly 85% of all targets. After recent training, the system has performed even better running on the SSV at the Demo C test site. (Section 2.4)

Target & Multisensor Visualization. Two generations of interactive 3-D graphics systems have been built to visualize target models in the context of multisensor data. Visualization allows us to inspect the progress and results of recognition algorithms. The current system contains over $50,000$ lines of code and requires a Unix workstation with Open-GL [1]. (Section 2.6.2)

Least-median Squares Multisensor Coregistration. A new 3-D object pose determination algorithm simultaneously refines the least-squares best-fit 3-D pose of a target model as well as the sensor-to-sensor image registration. This combined process of refining pose and registration is called here **coregistration**. A least-median squares extension makes the algorithm robust to outliers. Sensitivity analyses on controlled synthetic data with known ground truth have been performed and the algorithm has been demonstrated on actual target model and multisensor data features. (Section 2.9)

Range Probing Hypothesis Generation. A mature ATR system for predicting the target type and pose based upon boundary probing has been adapted to provide target type and pose hypotheses for subsequent multisensor validation. Tests on 35 of the Fort Carson range images suggest the system does not reliably predict the single correct best hypothesis. However, with hand tuning for close versus distant targets, the correct vehicle type appears within the top five hypotheses on 33 out of 35 images. (Section 2.5 & Section 3)

Multisensor Target Identification. Our new system takes target type and pose hypotheses and refines these through integrated multisensor matching. It correctly identifies four classes of vehicle on 27 out of 35 range, IR and color image triples from the Fort Carson dataset. This system employs a sophisticated search procedure to locally refine the coregistration (pose plus sensor registration) between target and sensors. The features representing the target signature are dynamically predicted and refined during matching using 3-D graphics hardware. Thus the signature is adapted to match scene properties such as lighting. In addition, feature prediction uses occlusion cues in the range in order

[1] Additional detail on the visualization component of this work appears in the paper 'Visualizing Multisensor Model-Based Object Recognition' in the Appendix of this book.

to modify target signatures for all three sensors. This dynamic ‘occlusion reasoning’ is a major advancement for ground-based ATR and has enabled precise multisensor matches to be recovered for terrain occluded targets. (Section 2.6 & Section 3).

Terrain Feature Prediction and Matching. A critical practical problem for the UGV has proven to be the establishment of precise (within several pixels) registration between stored terrain maps and ground-looking imagery. A system for extracting terrain features from both rendered terrain and live video in real-time (under one second) has been developed and delivered to Lockheed-Martin for use in Demo II. In the lab, this system is being coupled with an optimal feature matching system to demonstrate the feasibility of automated registration refinement. (Section 4)

2 The Multisensor ATR System

A three-stage multisensor ATR system has been developed to test key innovations in multi-spectral target detection and multisensor target identification. The component technologies for each of these stages are summarized here along with results on imagery from the Fort Carson dataset.

2.1 The Recognition Testbed

A series of major software components has been brought together within a single testbed to test both components as well as the end-to-end capabilities of our ATR system. A summary diagram of the system architecture is shown in Figure 1. The inputs to the run-time system are sensor images from FLIR, LADAR and color sensors. Additional inputs come from off-line components that provide vehicle model information, time-of-day lighting, and decision-trees used for color-based target detection. The 3-D vehicle models are reduced from their full BRL-CAD detail to simpler 3-D representations appropriate for matching. In the future, the work on terrain maps described in Section 4 will be made part of the core recognition testbed.

2.2 The Alignment Approach to ATR

The Multisensor Target ID module in Figure 1 embodies the extension of the alignment-based recognition paradigm to the ATR domain. At first, this might seem a simple transfer of a well-understood paradigm from one application domain to another. However, such a view grossly underestimates the particular sources of difficulty and complexity inherent in ATR. To list just some of these factors: typically image resolution is low, targets viewed in color imagery are textured, in FLIR target appearance is highly variable, and in range imagery geometric form is often complex. Also, while CAD models of targets are typically available, they often contain excessive detail (6 to 12 thousand polygons). Terrain features in scenes often introduce structured clutter and targets are often partially occluded. These factors make the direct application of current algorithms infeasible.

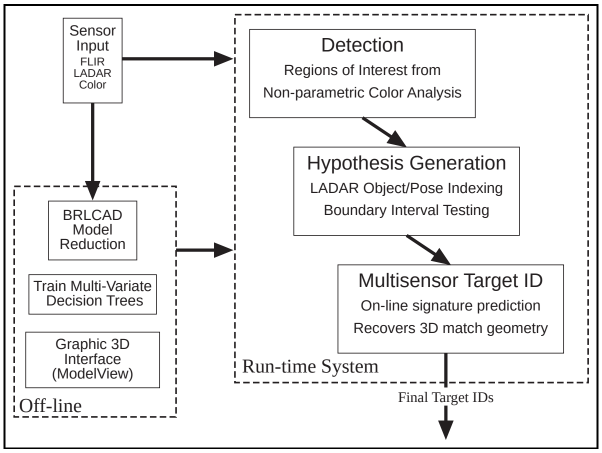

Sensor Input
FLIR
LADAR
Color
Detection
Regions of Interest from
Non-parametric Color Analysis
Hypothesis Generation
LADAR Object/Pose Indexing
Boundary Interval Testing
Multisensor Target ID
On-line signature prediction
Recovers 3D match geometry
Run-time System
Final Target IDs
BRLCAD Model Reduction
Train Multi-Variate Decision Trees
Graphic 3D Interface (ModelView)
Off-line

Figure 1: Overview of Recognition Testbed. The on-line system takes in imagery, performs detection, target type and pose hypothesis generation, and finally multisensor target identification. The off-line support software performs BRL-CAD model reduction, provides for training of the color detection system, and provides a full interactive 3-D graphical user interface for monitoring the multisensor identification system.

To overcome some of these difficulties, optical imagery must be supplemented with other types of scene information. A key tenet of our project is that 3-D range data resolves many ambiguities inherent in E-O imagery, and that E-O provides sources of constraint absent in the range imagery. Therefore, alignment-based recognition must couple together constraints from multiple sensors and target geometry. We accomplish this coupling with new algorithms that geometrically align and match 3-D target models with both range and E-O data. Through proper task formulation, global geometric constraints associated with known sensor and scene geometry are used to perform model-based sensor fusion.

Geometric constraints can be grouped into two categories: fixed intrinsic sensor properties, and variable scene attributes. Usually, the intrinsic parameters are calibrated off-line. However, variable attributes must be computed in the field. For example, the 3-D position and orientation of the target relative to the sensors is not known a-priori. Also, when separately mounted range and optical sensors are used, exact pixel registration between images can be expected to change. Thus, estimates of 3-D object pose as well as image registration must be allowed to vary during alignment. Thus, **coregistration** describes the process of simultaneously refining target-to-sensor-suite pose as well as sensor-to-sensor image registration.

2.3 Overview of Our Multisensor Dataset

At the start of the RSTA project, no three sensor (Range, IR and Color) data set was available. Therefore, a data collection effort was mounted by Colorado State University, Lockheed-Martin, and Alliant Techsystems. The collection took place in the first week of November 1993 at Fort Carson. The Fort Carson Colorado Army National Guard Depot made several vehicles available and provided drivers who placed the vehicles on the National Guard test range.

The data collection effort was highly constrained in terms of time, resources, vehicles and terrain. These limitations not withstanding, over 400 Range, IR and Color images were collected and this dataset has served as the primary dataset for all algorithm development and testing in this project. The Fort Carson data has been cleared for unlimited public distribution and Colorado State maintains a data distribution homepage (`http://www.cs.colostate.edu/∼vision`). To accompany the data, there is a 50 page report [7] describing each image, vehicle array, and ancillary information such as time of day and weather conditions. Additional information on sensor calibration may be found in [33].

The Fort Carson data meets all of our project's basic needs for algorithm development and testing. Specifically, it includes Range, IR and Color imagery for military vehicles positioned in natural terrain. The Alliant Techsystems LADAR used to collect range data generates 24 by 120 pixels with a 3 by 5 degree field of view. To simulate the nominal 1 foot per pixel range called for in the planned RSTA LADAR, vehicles were placed about 400 feet from the sensors at Fort Carson. Modestly wide angle lenses were used with the FLIR and color cameras so that 'pixels on target' values for these sensors would also be comparable to those expected in the 0.5 to 1.0 kilometer range using the RSTA sensor suite [2].

2.4 Recognition Stage 1: Detecting Targets in Multi-spectral Imagery

For the first stage of processing, a new machine learning algorithm [14] is applied to the problem of detecting camouflaged targets in multi-spectral (RGB) images. The goal of this module is not to identify the type or position of a target, but simply to detect where a target might be present, and to pass the resulting image chips (or "regions of interest" - ROIs) to the hypothesis generation module (which selects the target's type and approximate position) and eventually the coregistration matching module (which verifies the target type and refines the position estimate). Thus the goal of the color-based target detection module is to serve as a *focus-of-attention* mechanism that directs the system's resources toward parts of the image that contain potential targets. It should also be noted that although this work was designed for work on RGB images, the general approach is applicable to any multi-spectral image source, including multi-band IR or polarimetric imagery [39; 29].

2.4.1 Color Complements IR

In most ATR systems, targets are detected in 3-5 micron infra-red (IR) images. IR images have the advantage over color images (and many non-visible spectrums) that they can be used in either day or night operations, and that thermal signatures are comparatively difficult to hide (assuming the engine is running). By way of comparison, color images can only be acquired during the day, and any target detection system that uses them must be prepared to detect camouflage, an old but still very effective countermeasure.

Consequently, the goal of this project was not to develop a color-based target detection system that would supplant IR-based systems. Quite the opposite, our goal was to develop a color-based target detection system that would complement (and be used in conjunction with) existing

[2] The original plan for the RSTA system included a LADAR range sensor with a nominal one foot-per-pixel resolution at a range of $1,000$ meters

IR systems. Although generally useful, IR images exhibit certain problematic characteristics. The thermal properties of so-called “cold” targets whose engines are not running are difficult to predict, because their temperature (relative to the background) is a function of their recent history. If they are significantly warmer or colder than their background then they may be detected in IR images, but at times they may approximately match the background radiance and become difficult or impossible to spot in IR images. In addition, sunny days reflect solar thermal energy in the 3-5 micron range, creating false alarms and obscuring true targets in 3-5 micron IR images.

The problem with reflected solar energy on sunny days is one reason color detection complements IR. Color detection typically succeeds and fails independently of IR. For example, just when 3-5 micron IR sensors encounter their biggest problems on sunny days with lots of reflected thermal radiation; color-based systems are at their best. More generally, while IR systems have trouble with targets whose engines are not running, color-based systems are unaffected by such thermal properties. Conversely, color-based systems are useless at night, when 3-5 micron IR systems are at their best due to low background (thermal) radiation. There is one other good reason not to neglect color information: it is essentially free. Color cameras are by far the cheapest imaging sensors available, and many ATR systems already have color cameras on-board to aid human operators in verifying targets before firing.

2.4.2 Technical Issues

Given the reasons provided above, the ability to detect targets in color images is a potentially useful complement to IR sensors if an effective color-based detection system can be developed. The technical issues that must be addressed in order to build such a system are 1) the ability to recognize camouflage and 2) the ability to compensate for changes in apparent color due to changes in illumination, distance and viewing geometry.

Camouflage attempts to match the color and texture of a target to that of the background. Fortunately, the “background” color of the world is not a constant but rather changes daily, so that there is always a slight color distinction between a camouflage pattern and the true background; the target detection system we developed works by exploiting this fine distinction. This task is made easier by the multi-colored nature of most camouflage patterns – even if one color exactly matches a significant portion of the background, it is unlikely that the others will.

The more difficult issue is the variation over time of the apparent color of an object under natural lighting. The color of daylight changes as a function of the sun angle in the sky, which in turn depends on the time and location of the image. Since the apparent color of a target in an image is a combination of the surface color of the object and the color of the illuminant, the apparent color of targets changes with the daylight. This situation is further complicated by the observation that daylight is actually a combination of two distinct illuminants: direct sunlight (which tends toward yellow) and ambient skylight (which tends toward blue). The apparent color of a target depends on the ratio of sunlight to skylight falling on the surface, and therefore on the orientation of the target relative to the sun. Finally, weather conditions such as clouds and haze cause further changes in the apparent color of targets, including an apparent blue-shift as a function of target distance.

Attempts at color-based target detection using more traditional parametric classification techniques can be expected to fail. These traditional techniques would model variations in apparent color of a target as Gaussian noise around a “true” apparent color. Modeled in this way, the variations in apparent color will be much larger than the small distinctions between the color of camouflage and the color of the background.

Fortunately, shifts in the apparent color of targets are not random; there is a limited range of colors that natural daylight can assume[34], even given various ratios of sunlight to skylight, and a limited blue-shift created by atmospheric humidity. If we limit ourselves to a single sensor, therefore, we find that the apparent color of any single surface in outdoor images forms a continuous region in three-dimensional (RGB) color space (or a set of continuous regions if the object is multicolored, as are camouflaged targets.) Although we do not have sufficient information (i.e., about humidity) to predict the exact color of a surface in an outdoor image, we can fit a decision surface to the relatively smooth region of apparent colors that a target can assume.

We therefore train a non-parametric classifier to separate areas of color space that might be the image of the target under ‘normal’ conditions from those that cannot. Although such a classification scheme will always produce some false positives, it is very useful as a focus-of-attention mechanism to limit further processing by downstream recognition algorithms. Every image pixel can be classified as potential target or not according to whether it lies within the confines of the learned color region. The result is a binary region-of-interest image that marks all the pixels that lie within the object's color space; the target pixels in the binary images are then grouped to produce regions of interest (ROIs) around the targets. Examples are shown in Figure 2 (see color plates). This allows the system to use an RGB lookup-table for classification, enabling it to operate in almost real-time on inexpensive commercial hardware.

2.4.3 Multivariate Decision Tree Learning

The non-parametric classification technique we use is a multivariate decision tree (MDT) [14]. MDTs are a variant on traditional univariate decision trees [46] (a.k.a. regression trees[13]), in which a feature space is divided by selecting the feature and threshold value that best divides the target class from the background. This creates two feature subspaces, which are then recursively divided by another feature and threshold value, until each subspace contains samples that all belong to the same class (i.e., target or background). Geometrically, one can envision a univariate decision tree as a set of hyperplanes that successively divide the feature space into smaller and smaller regions, until each region contains elements of only one class.

The problem with traditional decision trees is that they divide the feature space by selecting a single feature and threshold, implying that the hyperplanes must be parallel to one of the feature axes. Multivariate decision trees recursively divide the feature space using the maximally separating hyperplane, regardless of its orientation. (This also implies that MDTs are impervious to linear transformations of the feature space, so that, for example, it makes no difference whether the data is presented in RGB or YIQ color space.) Another way of describing MDTs is that they fit a piecewise-planar function to a decision surface in a 3-D feature space.

It should be noted that other non-parametric classifiers could also be used for this task, including back-propagation neural networks. However, as discussed in [15], the decision surfaces for the apparent color of physical objects in an outdoor scene are well-described as piecewise planar functions in 3-D, and MDTs are therefore appropriate. Neural networks search for decision functions in higher-dimensionality function spaces, and therefore require more training instances to converge to a similarly reliable answer for this problem.

2.4.4 Operating Scenario

It is assumed that training imagery is obtained prior to a fielded mission, and based upon this training data the system learns to discriminate between color values produced by camouflaged vehicles and values produced by background terrain. Using the multi-variate decision tree learning algorithm discussed in the previous section, the result of training is a color lookup table (LUT) indicating, for each possible RGB color pixel value, whether it is more likely to be produced by a target or background.

In fielded operation, the system performs real-time color lookup on all pixels coming in and classifies them as target or background. Then, a region of interest (ROI) extraction process sums responses over fixed sized windows in the image and extracts ROIs: one ROI for each local maximum over a minimum threshold. When integrated with the RSTA package on the UGV, the results of the color detection were combined with those of a traditional FLIR detection algorithm.

Perhaps the most important factor in evaluating the usefulness of color detection concerns the degree to which training generalizes to variations in field conditions. The current system, using a single LUT, has been demonstrated to generalize across times of day, lighting conditions, weather, and vehicles. Results using the algorithm both on the vehicle and in the laboratory are discussed on the next two sections.

2.4.5 Experience Running on SSV-B

Before we look at evaluations of the color system conducted in a laboratory setting, let us briefly describe our experience in the field. Because of worries about reflected thermal radiation in 3-5 micron IR images at the Colorado Demo C site in July, the MDT target detection system was selected to run in conjunction with IR-based target detection as part of the RSTA package. After a significant software integration effort, the MDT system was finally integrated and debugged on board SSV-B in June, 1995.

On Tuesday, June 13, a handfull of training images were collected using this vehicle, and the following morning 14 of these images (3 indicating typical background colors and 11 showing vehicles) were used to train a color look-up table (LUT). Using this LUT, the system was tested from 1 to 5 PM on 51 new images. The results from these test are presented in Table 1. The 51 images included targets that were not in the training data, had both brown and green camouflage, and were viewed from vantage points differ-

Table 1: Detection Statistics on 51 Demo C Test Images. Training of the color detection system was performed using images collected the previous day. No true target was missed in this test.

Test Data Set, 1 - 5PM, June 14, 1995	
Total Number of Images	51
Target Types	5
Instances of Green Camouflage	34
Instances of Brown Camouflage	14
Missed Targets - False Negatives	
Target Instances	48
Targets Found	48
Targets Missed	0
False Negative Rate	0.0 (0/48)
False Detections - False Positives	
Total Number of Detections	766
Detections True	48
Detections False	718
False Positive Rate	0.94 (718/766)
Detection Statistics Per Image	
Minimum Detections	3
Maximum Detections	41
Median Detections	13
Mean Detections	15
Standard Deviation of Detections	9.5

ent from those in the training data.

The key result was that over the 4-hour period, under both cloudy and sunny conditions, viewing four different targets from two different vantage points, the system *never missed a target.* This first field test result was positive beyond our expectations. While perfect performance such as this is not a realistic expectation in general, it argues strongly for the merits of our approach.

Tight timing constraints associated with scheduling of SSV-B leading up to Demo C prevented further field testing or training. Consequently, there are no systematic results suggesting how performance changed with the changing terrain conditions. This is a major factor for the Denver site, where from June to July the natural grasses die and the predominant terrain color changes from green to brown. On the occasions in late July when the color detection system was run, it performed poorly compared to June. This is not surprising given the lack of re-training to account for seasonal changes.

Because the system was tuned to work in conjunction with a FLIR-based detection system, a high false positive rate was considered acceptable as a way of reducing the chance of missed targets. Observe the high false alarm rate in Table 1. To illustrate how these detection ROIs appear, the ROIs found for a typical image from the June tests at the Demo C site are shown in Figure 2a (see color plates). The summed response producing these ROIs are shown in Figure 2b (see color plates). Because each ROI is relatively small, even for those images with high numbers of detections, the color detection algorithm is focusing attention on a very small percentage of the total image.

2.4.6 Formal Lab Evaluation

The color focus-of-attention system for ATR has been formally evaluated on the Fort Carson data set, both by the authors and independently by Ted Yachik of Gilfillan Associates Inc. (LGA). Over 100 color images of military targets taken on 35mm film and then digitized onto Kodak CD were used in this evaluation. In [15], the authors evaluated the system at both a pixel and region-of-interest level. At the pixel level, it correctly identified target pixels 53.4% of the time and background pixels (which are much more common) 97.5% of the time, albeit with a high deviation from image to image. (The SD was 10.4% for target pixels and 1.6% for background.) Interestingly, this level of pixel-level performance was enough for very impressive region-level performance: the system identified 109 out of 112 targets with a total of 44 false alarms.

Independently, Ted Yachik evaluated the color FOA system on the Fort Carson data, using a slightly different sampling methodology for selecting training and test images, and a slightly different set of algorithm parameters. Although he did not analyze his results at the pixel level, his results at the region-of-interest level were roughly compatible: MDT found 49 out of 56 targets (87.5% of targets) while generating an approximately 6.5 false alarms per frame.

Subsequent to these initial evaluations, the authors set out to determine the optimal parameter settings of the MDT algorithm. In particular, they investigated the threshold used by the final step that converts the binary classification image into a selected set of ROIs. (This threshold determines what percent of the pixels in a window must be classified as target before a ROI is extracted.) As shown by Figure 3a, thresholds of above 70% caused the system detection rate (i.e., targets found) to decrease without significantly decreasing the number of false positive responses (i.e., false alarms). Conversely, threshold settings below 70% cause an increase in false positives without a significant decrease in false negatives. As a result, the 70% threshold was deemed optimal for this data set, and the false positive and false negative rates at this setting should be considered indicative of the current state of the system.

One technical point about the evaluations above should be discussed. Since the MDT system merges overlapping ROIs (i.e., overlapping detection rectangles), and because it is not given any depth information about the scene from which to infer target distance and therefore approximate target size, MDT returns ROIs of varying sizes. Both the author's original evaluation [15] and the LGA evaluation simply counted the *number* of true and false ROIs detected, and therefore the system was evaluated less harshly if it returned one large false positive region than if it returned two smaller ones, even if the sum of the areas of the smaller regions was less then the single large region.

In the context of a focus-of-attention mechanism, such an evaluation is flawed. If the system returns one large ROI, it is forcing the subsequent stages of the system to search a larger portion of the scene than if it returns two smaller ROIs. Consequently, when calculating the ROC curve we refined our study to measure false positives as a percent of image rather than a count of ROIs.

Another measure of MDTs effectiveness as a focus-of-attention mechanism can be found in its ability to filter out data. In Figure 3b we measure the percent of image data passed through to later stages of processing as an ROI (whether a target or a false positive) as a function of the ROI threshold discussed above. At the recommended 70% threshold, MDT filters out over 99% of the data, leaving the subsequent hypothesis generation and coregistration matching modules to search less than 1% of the image.

More evaluation of MDT is clearly needed. Both of the studies above were based on the same set of 35mm images. Evaluations on more data sets, including data sets of CCD images, are needed. The system is available for such tests, but color image sets of military targets from a single sensor are currently unavailable. (The Sept. 1994 Lockheed-Martin data set is not appropriate for this purpose because the color images were taken with an auto-white balancer, which effectively changed the sensor characteristics from image to image.) An evaluation of the color classification system in a non-military context is currently being carried out by UMass and General Motors.

2.4.7 Future Issues: Color Calibration and Portability Between Sensors

Two technical issues are still unresolved with regard to the practical use of color as an FOA mechanism for ATR. The first is sensor independence. The current system assumes that all training and test images are taken with a single sensor, an assumption that does not fit well into military scenarios. We believe that this assumption can be removed if the color transformation between one sensor and another is known, essentially by transforming the borders of the decision surface. However, this is an untested hypothesis.

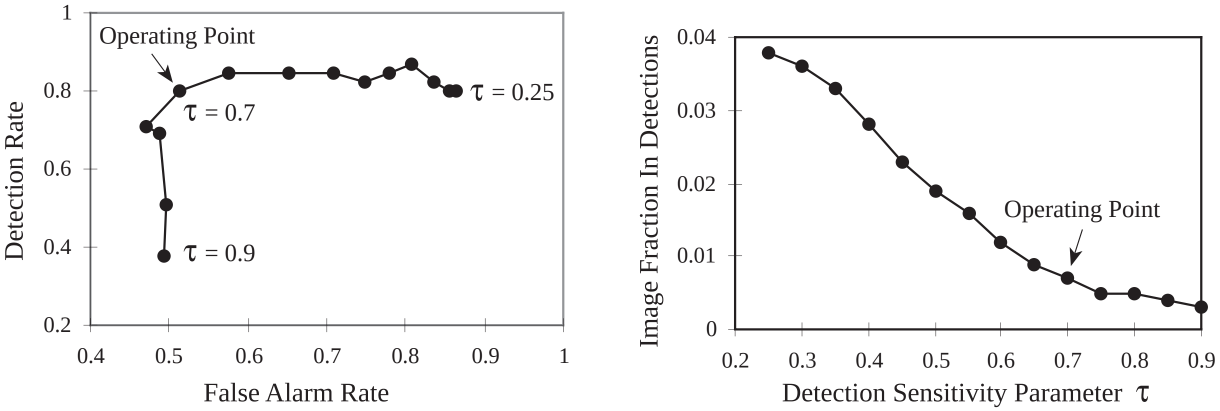
The second issue is the effective use of contextual information. Much of the variance in the apparent color of a target is the result of contextual factors that may be known, such as the time and location where the image was taken, the weather conditions, and/or the approximate distance to target. Such factors should allow us to restrict the expected appearance region in color space when they are known. We are currently collecting samples of natural illumination under different sun angles and weather conditions, with the aim of using this information to improve classifier performance in the future.

2.5 Recognition Stage 2: Hypothesizing Target Type and Pose

The multisensor target verification algorithm presented in Section 2.6 is powerful in terms of its ability to relate target model features to multisensor image features under widely varying target pose and image registration estimates. The algorithm is also very computationally demanding. To limit processing time, we use a less demanding algorithm to generate target and pose hypotheses. Therefore, by reducing the number of possibilities to examine during verification, processing time is reduced. While any number of algorithms might fill this role, including geometric hashing techniques [1], we have chosen to use an existing boundary probing algorithm [11] developed by Alliant Techsystems.

2.5.1 Range Boundary Probing

Alliant Techsystem's LADAR Recognition System (LARS) has demonstrated state-of-the-art target identification performance on hundreds of frames of both real and synthetic imagery. The LARS suite, summarized in Figure 4, uses a non-segmenting model-based approach, which efficiently exploits both the 2-D (boundary matching) and 3-D (surface matching) shape information contained in LADAR signatures. Templates are derived from BRL models of the expected target set, therefore no training imagery is required. Since LARS does not perform segmenta-



a. ROC Curve **b.** Focus-of-Attention Curve

Figure 3: Target Detection Algorithm Performance on Fort Carson Images.

tion, it avoids information loss and provides robust performance in low SNR (Signal to Noise Ratio) scenarios, an important consideration for low LADAR visibility conditions. In past tests on Tri-service LADAR data, LARS consistently attained target identification performance in the mid-to-upper 90% range.

As shown in Figure 4, LARS first processes the 2-D signature information. The off-line system generates a set of templates consisting of a list of probe points. Each probe point is designed to test for a discontinuity along the desired target boundary. Applying a probe to an image requires only a simple test to make sure the pixels at either end of the probe are greater in depth than some threshold. The likelihood of a match for each template is based on the number of passing probes in relation to the total number of probes in the template. This 2-D boundary matching process is referred to as BICOV (Boundary Interval Coincidence Verification).

BICOV operates on individual absolute range images corresponding to pre-cued ROIs. The BICOV output is an ordered list of the most likely target hypotheses at a specific pose, paired with a likelihood confidence ratio. In this project, the top hypotheses are passed onto the multisensor verification module.

In addition, the LARS system also contains a 3-D surface matcher (known as SUMMIT), which exploits the topography of a target's surface. The internal separation of the LARS matching stages is done primarily to achieve greater computational efficiency. A-priori knowledge of target class and aspect (as provided by BICOV) greatly constrains the 3-D surface matcher search space and simplifies the SUMMIT algorithm complexity as well. Since we are concerned with target hypothesis generation, we use only the more efficient BICOV algorithm.

When the existing LARS system is run in a stand-alone mode, both boundary and surface matching is performed and a certainty accrual mechanism is used to combine the BICOV and SUMMIT match scores. It is worth noting that this is an example of a weaker form of fusion, since the accrual mechanism does not actually couple the geometric constraints from boundary and surface information in a single geometric measurement process. Put simply, the two processes might both return high scores for a case where surface and boundary are mis-aligned. This decoupled fusion is in sharp contrast to the multisensor verification module presented below, for which geometric consistency is maintained through a single consistent manipulation of the multisensor and target geometry.

2.5.2 Avoiding Exhaustive Probing

Boundary interval probing algorithms suffer from a problem common to most all template matching [4] approaches: exhaustive search in an explosive space of probes/templates is impractical. What is needed are control strategies to select probes only when they are likely to convey meaningful and helpful information, i.e., when their respective scores will be high. Past work on this general problem has developed hand-coded heuristics for avoiding exhaustive probing [11] and at least one algorithm has developed probe hierarchies [12] to control probe use.

In a recently initiated joint project with Professor Charles Anderson, also at Colorado State University, we have begun to explore the use of

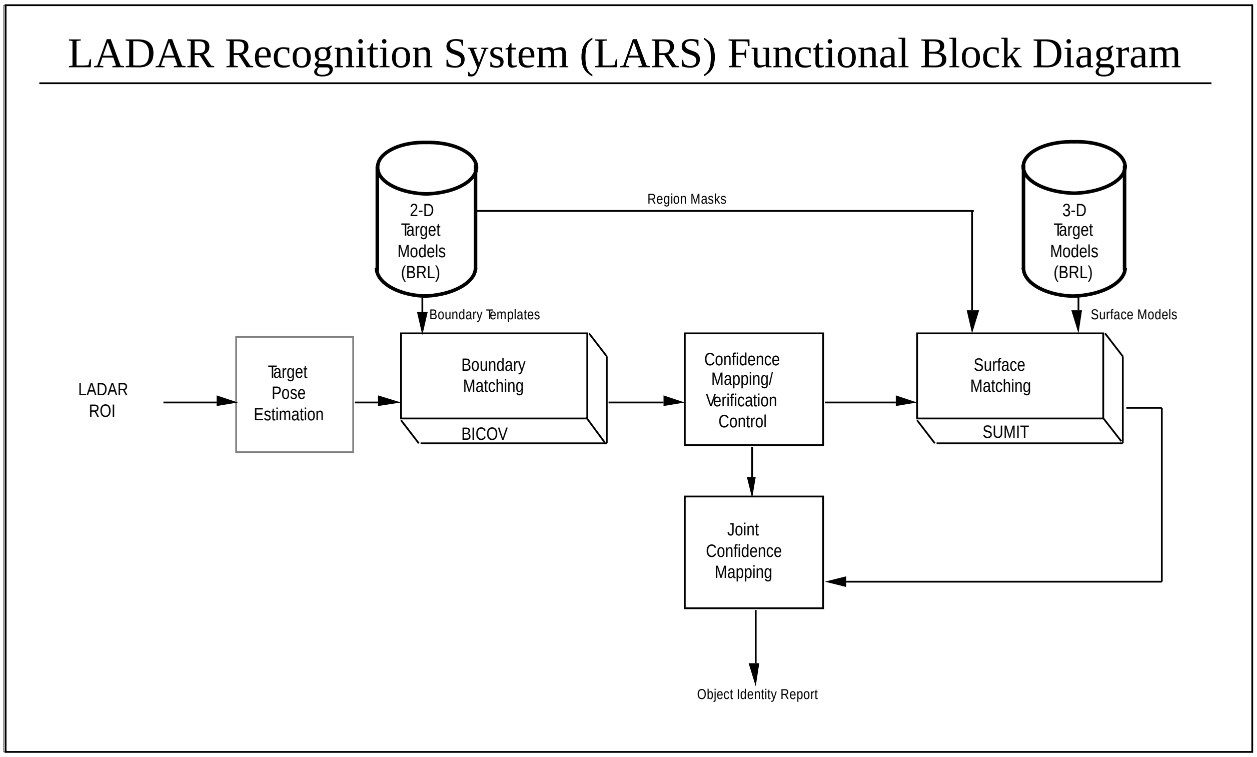


Figure 4: Block Diagram of Alliant Techsystems LADAR Recognition System.

a Neural Network (NN) as a device for efficiently selecting which probes to apply and when. While there is little new about using a NN in the context of ATR [59; 17; 22; 35; 45], what is novel about our approach is that the NN is being used primarily as a control mechanism rather than a pattern classification tool.

The goal is to teach a relatively compact and efficient NN to predict the responses generated by a large set of probes applied to a given window in an image. The NN is being trained to learn a clear and explicitly defined mapping: the mapping from image pixel values in the image window to the probe score generated by a boundary interval probe. Since exhaustive probing is the default, the NN may be trained in a supervised fashion simply by 'watching' exhaustive probe applications.

A proof of concept system has been tested on $5,400$ probes developed for three of the vehicles (M60, M113 and pickup truck) in the Fort Carson Dataset. A two layer NN has been trained on synthetic LADAR data generated from BRL-CAD models and then tested on 15 Fort Carson LADAR images. The NN reliably predicts the top 25 out of $5,400$ probes to apply at any given pixel. After only 10 training epochs over synthetic data the hybrid neural network approach was shown to perform virtually equivalent to the brute-force apply-all-probes technique on the real data imagery. The NN learning converges quickly, suggesting that these mapping are linear and therefore not difficult to learn. Of key importance, even though it is a statistically uncommon event for probes to return a high score, the learning is generalizing to capture these cases.

The most significant result relates to the run-time savings using the NN to selectively apply probes. Running on a Sparc 10, it takes almost 24 hours to apply all $5,400$ probes to all the LADAR pixels in the dataset.. In contrast, running the NN and the resulting 25 best probes across the takes roughly 2 hours. Both these run-times are long, and it must be stressed these results are for a new research system with no optimizations. The next phase in this work will be to tie this learning into the actual LADAR probing system developed by Alliant Techsystems.

In this very first pass at this approach, it appears the NN reduces run-times by an order of magnitude. It is not unreasonable to think much greater savings are possible. Work on this project will clearly be expanded and continued. One such extension will be to switch from range to E-O data using registered DEM data for range-to-pixel estimates. This extension will also exploit the DEM to ground-looking imagery registration work presented in Section 4. Without such registration, probing on E-O data is infeasible.

2.6 Recognition Stage 3: Multisensor Target Identification Using Coregistration

This section introduces the target identification system that fuses Range, IR and Color imagery onto the 3-D target model and thereby makes a determination as to the true type and pose of the target. To begin, the concept of coregistration is carefully explained, followed by a description of the 3-D visualization environment developed for the project. Then the two key ideas of multisensor matching are explained: on-line model feature prediction and iterative search through the space of globally consistent relationships between sensors and target. This section concludes with a description of our recently developed occlusion reasoning component and three matching examples from the Fort Carson dataset. A full evaluation of the target identification system is presented in Section 3.

2.6.1 What is Coregistration?

Appearance of 3-D object models varies with viewpoint, and pixels from multiple sensors typically are not in a one-to-one correspondence. Knowledge of sensor parameters and relative sensor positions can provide moderately accurate estimates of the pixel-to-pixel registration. However, small variations in relative sensor position can lead to significant mis-registration between pixels. This is of concern when matching objects, such as targets, which are small in terms of absolute image size. To get around registration problems and 3-D variations in appearance, ATR systems commonly assume sensor registration is exactly known or determined using low-level correlation. Variation in 3-D appearance is typically accounted for by sampling expected viewpoints to produce a set of templates represented in image space.

Our approach is different. Rather than assuming perfect registration obtained prior to matching object models or building a suite of viewpoint specific templates, we have developed new methods to simultaneously refine alignment between sensors and 3-D object models. This process refines the pose (position and orientation) estimate of the target model relative to a sensor suite as well as the sensor-to-sensor alignment from which the sensor registration is derived.

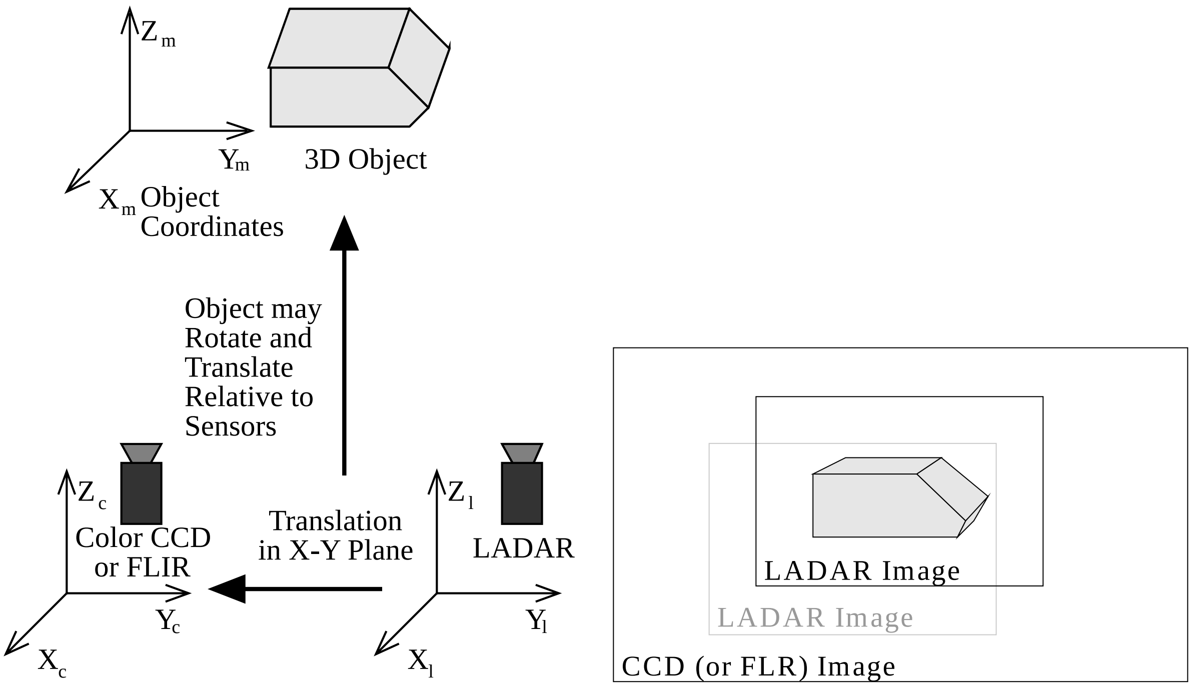
In the general case, an entire family of coregistration problems can grow out of different assumptions regarding the relative placement of the sensors. At one extreme, if sensors are assumed to be perfectly registered, then coregistration devolves into sensor to object pose computation. At the other extreme, if sensors move freely and independently, then there is no coupling and the result is an independent sensor pose problem for each sensor. The specific problem of interest in the context of RSTA is that of near-boresight-aligned sensors.

A detailed study of different sources of uncertainty in alignment for near-boresight-aligned sensors appears in [33]. Briefly, a useful heuristic falls out of this study: over small rotations and restricted depth ranges, sensor-to-sensor rotation may be approximated with simpler coplanar sensor-to-sensor translation. This approximation is illustrated for two sensors in Figure 5. Figure 5a illustrates the 3-D geometry of an object, a FLIR or Color sensor and a LADAR sensor. The sensors, together, are free to rotate and translate relative to the object. The sensors are constrained to permit only translation in a common image plane. These 3-D constraints permit translation of FLIR or color images relative to LADAR images as illustrated in Figure 5b. In all the coregistration work developed below, this co-planar translation constraint between sensors is imposed.

2.6.2 The Testbed Visualization Component

An interactive 3-D interface has proven essential to visualize and inspect relationships between 3-D object models and sensor data. Two generations of interactive 3-D graphical user interfaces have been developed under this project [26; 25; 54].

Figure 6 (see color plates) shows a sampling of views from our original RangeView system, and it provides a visual overview of the data, object models, and relationships of interest in our work. A FLIR image of an M60 tank is shown in the upper left corner of the figure. The thermal readings have been given a color coding in which hotter, or higher value samples are red, and colder, lower value samples are blue. The next frame shows the same thermal information texture mapped onto the range data. The subsequent two panels show similar information for the color image. The right column shows the CAD model visualization capabilities of RangeView. The M60 model can be rendered in the scene along with the range information. The user has the ability to interactively



a. 3-D scene geometry and transformation constraints

b. Resulting allowed pixel translation between images

Figure 5: Coregistration with sensor-to-sensor planar translation

modify the viewpoint to gain a better understanding as to how the range features match to the model information.

While RangeView used the LADAR coordinate system as the central reference frame, its successor ModelView uses the 3-D target model frame. Figure 7 graphically shows how the various reference frames relate via the given transformations. Each of the boxes represents a unique reference frame. At the center is the canonical representation: the model coordinate system. Each arc in the graph represents a mapping from one coordinate system to the next. All of the M transformations are mappings between 3-D reference frames and are invertible.

ModelView is primarily used to visualize relationships between target models and sensor data. However, it also supports combined visualization of range and E-O data. In ModelView, the sensors themselves are not iconically represented. Instead, changes in 3-D relationships between them are expressed through visual overlays of one type of sensor output on another. A user of ModelView will typically create multiple windows showing different types of model and sensor data displays. Examples of such screen visualizations, as well as a more detailed description of the ModelView system appears in the Appendix of this book.

2.7 Interleaving Feature Prediction and Multisensor Target Matching

The search process developed for coregistration matching uses an iterative generate-and-test loop (Figure 8) in which the current coregistration hypothesis, denoted as $\mathcal{F}$, is used to predict a set of model features which are, in turn, used in an error evaluation function. A neighborhood of moves is then examined and the best move, the one with the lowest error, is taken. The features are regenerated for the new coregistration estimate and the process continues. The three key elements in this process are: feature prediction, match evaluation, and local search. Each of these elements is described below.

2.7.1 On-line Model Feature Prediction

Highly detailed Constructive Solid Geometry (CSG) models of target vehicles are available in BRL-CAD format [58]. We have already developed algorithms to convert these models to a level of detail more appropriate for matching to the given sensor data [53; 52]. Another system, summarized here and fully described in [42], has been developed to extract edge and surface information from these models.

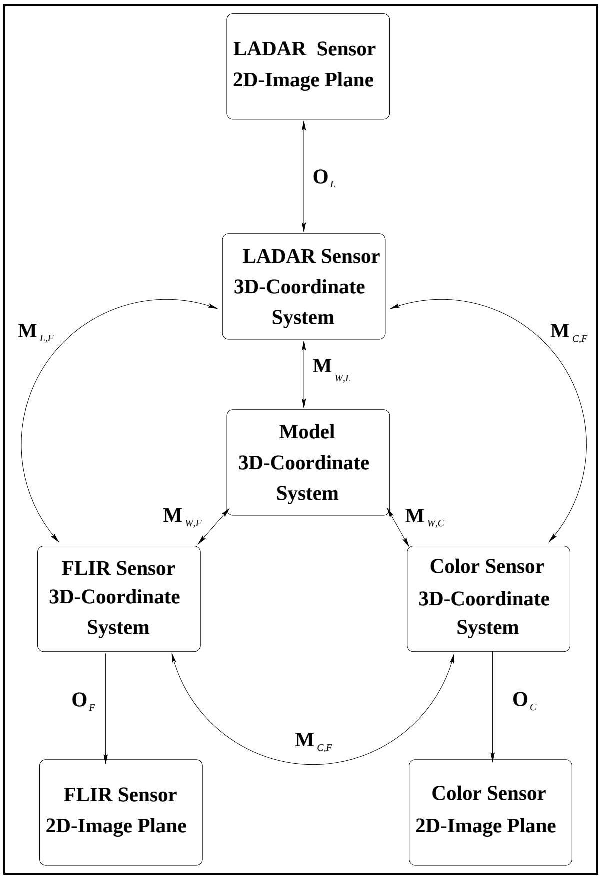

LADAR Sensor 2D-Image Plane
O_L
LADAR Sensor 3D-Coordinate System
M_L,F
M_C,F
M_W,L
Model 3D-Coordinate System
M_W,F
M_W,C
FLIR Sensor 3D-Coordinate System
Color Sensor 3D-Coordinate System
O_F
O_C
M_C,F
FLIR Sensor 2D-Image Plane
Color Sensor 2D-Image Plane

Figure 7: Coordinate Systems of Model and Data Sets

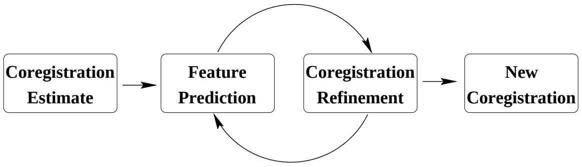


Figure 8: Interleaving Feature Prediction, Coregistration Refinement and Matching

The feature prediction algorithm renders the vehicle using the current pose and lighting estimates to infer which 3-D components of the target will generate detectable features in the specific scene. Each rendered 3-D surface is given a unique tag and the resulting image carries precise information about surface relationships as seen from the hypothesized viewpoint. From this information, the feature prediction algorithm identifies those elements of the 3-D model that generate the target silhouette. Prediction also takes account of lighting from the sun to identify significant internal structure.

For range imagery, sampled surfaces are extracted from the 3-D model using a process that simulates the operation of the actual range sensor. The target model is transformed into the range sensor's coordinate system using the initial estimate of the target's pose, and rays cast into the scene are intersected with the 3-D faces of the target model. The same rendering step used to predict optical features is used to filter the number of visible features for this range feature extraction algorithm.

2.7.2 Match Evaluation

The goal of the search process is to find an optimal set of coregistration parameters based upon measures of fidelity between target model features predicted to be visible and corresponding features in the optical and range imagery. This measure of fidelity is expressed as a match error, which is lower for better matches. This match error may be written as:

$$E_{\mathcal{M}}(\mathcal{F}) > 0 \quad \mathcal{F} \in \Re^K \tag{1}$$

The argument, $\mathcal{F}$, represents the coregistration of the sensors relative to the model. For a sensor triple of IR, color and range, $\mathcal{F} \in \Re^{10}$ with 6 degrees-of-freedom (DOF) encoding the pose of the sensor suite relative to the target; 2 DOF encoding the co-planar translation of each optical sensor relative to the range sensor.

The error, $E_{\mathcal{M}}(\mathcal{F})$, is divided into three main components: two weighted terms representing how well the 3-D predicted edge structure matches the current color, $E_{\mathcal{M},\mathcal{C}}(\mathcal{F})$, and IR, $E_{\mathcal{M},\mathcal{I}}(\mathcal{F})$, imagery, and a weighted term representing how well the predicted sampled surface fits the range, $E_{\mathcal{M},\mathcal{R}}(\mathcal{F})$, data. These terms may be combined to form the overall match error:

$$\begin{aligned} E_{\mathcal{M}}(\mathcal{F}) &= \alpha_{\mathcal{C}} E_{\mathcal{M},\mathcal{C}}(\mathcal{F}) \\ &+ \alpha_{\mathcal{I}} E_{\mathcal{M},\mathcal{I}}(\mathcal{F}) \\ &+ \alpha_{\mathcal{R}} E_{\mathcal{M},\mathcal{R}}(\mathcal{F}) \end{aligned} \tag{2}$$

where $(\alpha_{\mathcal{C}}+\alpha_{\mathcal{I}}+\alpha_{\mathcal{R}} = 1.0)$. Each sensor term can be further broken down into two weighted terms: an omission error and a fitness error.

$$\begin{aligned} E_{\mathcal{M},\mathcal{S}}(\mathcal{F}) &= \beta_{\mathcal{S}} E_{fit,\mathcal{S}}(\mathcal{F}) \\ &+ (1-\beta_{\mathcal{S}}) E_{om,\mathcal{S}}(\mathcal{F}) \end{aligned} \tag{3}$$

The subscript $(\mathcal{S})$ is replaced with either $\mathcal{C}, \mathcal{I}, \mathcal{R}$. The fitness error $E_{fit,\mathcal{S}}(\mathcal{F})$ represents how well the strongest features match (as determined by a threshold), and the omission error $E_{om,\mathcal{S}}(\mathcal{F})$ penalizes the match in proportion to the number of model features left unmatched. Omission introduces a bias in favor of accounting for as many model features as possible [5]. The fitness error values are summarized below and detailed in [44].

The optical fitness error represents the fidelity of match between the 3-D edge features and the underlying image. The process of determining the error begins by projecting the predicted 3-D model edges into the optical imagery. Projection is possible because both the intrinsic sensor parameters and the pose of the target are known. The gradient under each line is then estimated and converted to an error normalized to the range $[0, 1]$. Lines with weak gradient estimates are omitted.

The range fitness error represents how well the predicted 3-D sampled surface model points fit the actual range data. The error is based on the average distance from each model point to the corresponding nearest Euclidean neighbor. To reduce computation, only a subset of the range data is examined at any one time. A bounding rectangle around the hypothesized target is formed within the 2-D coordinate system of the range image. A 3-D enclosing box is then derived by back-projecting the rectangle into the 3-D range sensor coordinate system. When seeking points to match to the 3-D target model, only the data points lying inside this box (within some margin of error) are examined. Matched points having too great a Euclidean distance are omitted.

2.7.3 Finding Locally Optimal Matches

Match error is locally minimized through iterative improvement. The local improvement algorithm samples each of the 10 dimensions of the coregistration space about the current estimate. Sampling step-size is important and a general strategy moves from coarse to fine sampling as the algorithm converges upon a locally optimal solution. The initial scaling of the sampling interval is determined automatically, based upon moment analysis applied to the current model and sensor data sets.

A variant on local search, called *tabu* search, is used to escape from some local optima [23]. Tabu search keeps a limited history and will explore 'uphill' for a short duration to climb out of local optima. In this problem, it turns out that the regeneration of predicted target features changes the error landscape after each move. This can, in turn, induce local optima which tabu search readily escapes.

When tabu search fails to find improvement in the current neighborhood, the resulting 10 values are returned as the locally optimal coregistration estimate. Initial results of the search have shown that the local optima in color, IR, and range space do not usually coincide. By searching for the model in both the optical and range imagery, local optima in each will be rejected in favor of a more jointly consistent solution.

2.8 Occlusion Reasoning

One of the main benefits of multisensor ATR is the ability to reason about model feature occlusion. Since the range sensor provides an estimated range to the target, the following observation can be made: having a range pixel located much closer to the sensor than expected supports the belief that the feature is occluded.

The addition of occlusion reasoning to the existing system was fairly simple. We modified the system to retain the model face associated with the sampled surface point predicted for matching. Then the closest Euclidean neighbor to each model point was found using the same method discussed in Section 2.7.2. If the nearest neighbor lies some fixed distance (3 meters in our experiments) in front of the target, then it is labeled as occluded.

Once the point has been labeled as occluded, the match error for the range data is adjusted to remove this point from the predicted target signature. To accomplish this change, the match error was changed as follows:

$$\begin{aligned} E_{\mathcal{M},\mathcal{R}}(\mathcal{F}) &= \beta_{\mathcal{R}} E_{fit,\mathcal{R}}(\mathcal{F}) \\ &+ (1-\beta_{\mathcal{R}}) \max(E_{om,\mathcal{R}}(\mathcal{F}), E_{oc,\mathcal{R}}(\mathcal{F})) \end{aligned} \tag{4}$$

where $E_{oc,\mathcal{R}(\mathcal{F})}$ is a non-linear function of the ratio, r, of occluded versus the total possible visible features:

$$E_{oc,\mathcal{R}(\mathcal{F})} = \begin{cases} 0 & \text{if } r \leq 0.4 \\ (r-0.4)/0.6 & \text{if } 0.4 < r < 0.6 \\ 1 & \text{if } r \geq 0.6 \end{cases} \tag{5}$$

Initial experiments showed it was not enough to simply remove the features from the match that were believed to be occluded. The matching system quickly discovered the benefit of moving vehicles completely behind a hillside, thus occluding all of the features and sending the error measure to zero.

Once the changes to the range error were made, it again became obvious that we needed to remove features from the set used in matching to the optical imagery. Using the established link between the model face and the associated sampled feature, we simply remove all lines from consideration for which the associated face is occluded. These edge features are completely neglected in the optical error computation.

Figure 9 (see color plates) shows an example of the multisensor matching algorithm with the occlusion reasoning. In this image, the bottom half of the M901 is occluded by the terrain. In the center of the Figure are two range images, the top has the range with a grey-scale rendering of the vehicle and the bottom has the color image textured over range data. The left image shows the color image with the features determined to be occluded in black. Similarly the IR image is on the right with the occluded features in white. All other features were matched.

2.9 Least-Median Squares Coregistration

The algorithm described above searches in the space of coregistration estimates for a best match between target model and image features. This section describes a least-squares multisensor pose algorithm which, given a set of corresponding model and image features, recovers the associated best coregistration estimate. This algorithm extends single sensor pose work [28; 36; 20; 38] by imposing constraints on both sensor and object geometry.

Our target identification system does not currently include the least-squares multisensor pose algorithm. The pose algorithm was developed to provide a basis for an alternative form of multisensor matching in which the globally best least-squares coregistration would guide a search through the space of possible discrete matches between target and image features. This proved infeasible given the great number of sample surface points predicted for a target.

The multisensor pose algorithm is useful for final highly precise refinement of the coregistration parameters and it is intended that it should be utilized as a final refinement for output of the multisensor matching algorithm presented in the previous section. There is also promise that if extended to work with more highly structured range features, the least-squares approach may yet be a useful basis for a search algorithm which operates in the space of discrete mappings between target and sensor features. Additional details on the algorithm presented here, along with one such possible extension, are described in [3; 2; 50].

2.9.1 The Least-Squares Fitting Function

The best coregistration estimate minimizes a quadratic error of fit between corresponding object model and sensor features.

$$E_{fit} = \alpha_{fit} E_{fit,o} + (1 - \alpha_{fit}) E_{fit,r} \qquad (6)$$

The constituent parts of E_{fit} are illustrated in Figure 10. The first term, $E_{fit,o}$, measures distance between corresponding optical and model features. This term is precisely the point-to-plane error criterion defined by Kumar [36; 38] for computing camera-to-model pose [3]. The second, $E_{fit,r}$, is the sum-of-squared Euclidean distances between corresponding model and range points. The edge features in the optical imagery are found using a model-directed edge extraction technique described in [43].

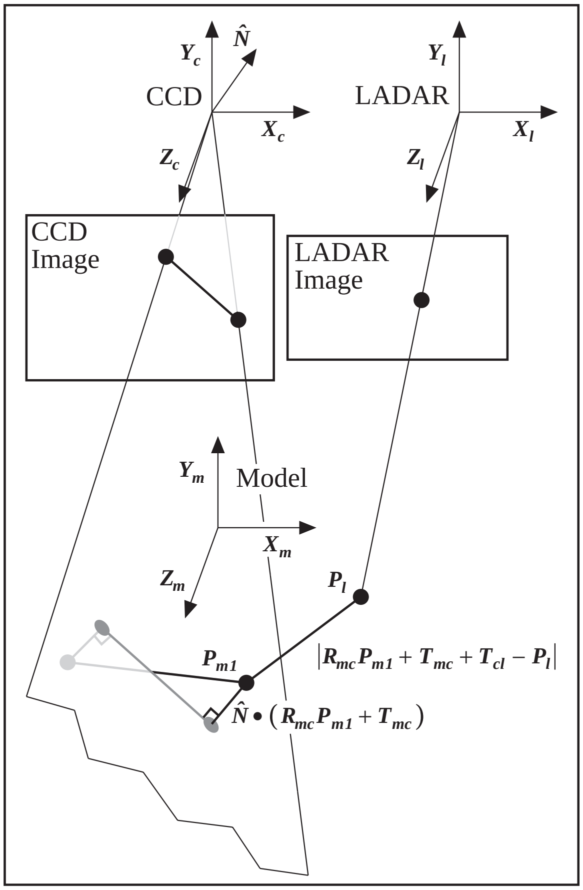


Figure 10: Illustrating distance errors which define optimal coregistration.

[3] After developing this measure, Kumar developed others which are more robust but which also require additional normalization.

The weighting term $0 \leq \alpha_{fit} \leq 1$ controls the relative importance of the optical and the range data. The terms $E_{fit,o}$ and $E_{fit,r}$ are normalized between $[0, 1]$ based upon the expected amount of noise present in the features [4], and consequently E_{fit} also falls in this range. This normalization allows comparison of data from two separate sources. The exact derivation of the least-squares fitting function and the associated iterative update equations used to minimize the non-linear error term are presented in [51; 3].

2.9.2 Median Filtering Extension

Median filtering [49] handles outliers by fitting to the subset of the data which minimizes the ensemble median error value. It is a robust statistic when there are less than 50% outliers. This is in contrast to the mean around which least-squares algorithms are based, where a single outlier can radically shift the result. The subset which minimizes the median error will contain no outliers. Including an outlier in a subset results in a poor estimate of the true curve (statistical model) and, in turn, will increase the median error.

The space of subsets is combinatoric and hence typically large. To avoid exhaustive search, the space is randomly sampled. Given sufficient samples, the probability of seeing at least one outlier-free subset is very high. This yields the optimal fit, and allows us to discard all data not accounted for by the Gaussian assumption (i.e., outside of two standard deviations of the best fit function, since this will contain 98% of the data effected by Gaussian noise).

The subsets must be at least large enough to cover the degrees of freedom, so at least three optical lines and one range point are needed. However, Kumar [37] found that selecting a minimal number of features caused the solution to be sensitive to the Gaussian noise that we assume is overlaid onto the true data. As a consequence, it is better to select a larger subset to stabilize the optimal pose against noise. If we select too large a subset size, however, we greatly reduce our chances of selecting a subset with no outliers. A compromise must be made between probability and stability.

Once we have minimized the error, we need to select a cutoff point, above which we will consider correspondences to be outliers. We can achieve this either by selecting some *a priori* threshold or by computing one based on the median. We choose the later method. Assuming a normal distribution, we can set `cutoff` $= (a \times s)^2$ where $s = \frac{\min \tilde{E}_{fit}}{0.6745}$ is an approximation of the standard deviation for a Gaussian distribution based upon the interquartile range. Setting a to 2.0 filters out data which lies more than two standard deviations above the error, so that the majority of the Gaussian data will be retained.

2.9.3 Least-Squares Study on Controlled Data

The synthetic optical sensor has a 4° field of view and generates a 512×512 image; the range images, 6 pixels per meter at 500 meters. The sensors are separated by 1 meter. Each model is located 500 meters from the sensors along the focal axis of the optical sensor. The ground truth image data for these tests is obtained for each sensor by projecting the appropriate model features (lines for optical, points for range) onto the sensor image plane.

Algorithm tuning parameters such as error weighting terms and convergence criteria are constant throughout both experiments. The weights in the coregistration error, λ_{oi}, λ_{ri}, w_{mo} and w_{mr} [5], are all set to 1.0. The convergence threshold for E_{fit} is 10^{-4}. The maximum number of iterations is 20.

Two sets of experiments were conducted: 1) sensitivity to noise in initial coregistration estimate, and 2) sensitivity to noisy image data. Both tests were run on four synthetic models. The models exhibit different geometric characteristics including planarity or lack of planarity, symmetry or lack of symmetry, and few versus many features. Complete results of these tests are reported in [51].

In Test 1, we found that, given perfect image data, the algorithm could reliably recover and correct coregistration given up to a 30° error in orientation. The correct solution was often found even for orientation errors as large as 50° and initial translation errors up to 100 meters. This suggests that, given good data, the algorithm reliably converges upon the optimal set of coregistration parameters. Test II shows that, given modest image noise ($\sigma = 1$ for both sensors), the final rotation error was within 1° of the correct value. With significantly higher errors, though, ($\sigma = 5$ for both

[4] While a rigorous and complete noise model is not developed, the Gaussian noise assumption underlies least-squares.

[5] The weights are the combined threshold and α_{fit} term described in [51]

sensors), coregistration yielded a final rotation error around 5°.

2.9.4 Least-Median Squares on Real Data

In our previous work [51], instabilities and pathological behavior were found when running coregistration on hand-picked features. This behavior has been traced to outliers present in the hand-picked data. To address this issue, median filtering is used to construct outlier-free correspondences.

For both of the example runs shown here, we started with the same initial pose (shown in Figures 11a and c). The initial correspondence was built based upon the initial coregistration hypothesis. In the CCD, lines with average distance less than 30 pixels and orientation difference less than 15° were included in the initial pose. For the LADAR, points within 0.5 pixels in the x and y dimensions and 10.0 meters in distance were paired. While these values provide a relatively small and mostly correct initial correspondence, an enlarged initial correspondence would include significantly more than 50% outliers and median filtering would fail.

Figures 11a and c show the initial positioning of the model features (shown in black) and the data features (shown in grey). In the final results, a similar scheme is used, with the addition that features included in the match are filled. In the optical images, the features (both model and data), included in the match are shown in grey, and the unmatched features are shown in black. The correspondence between individual features is not explicitly shown. In Figures 11b and d, the final results of median filtering show a generally good match, indicating the absence of significant outliers. Notice that the LADAR points (Figure 11d) generated by the top of the vehicle are not included in the match, since they match poorly.

3 Evaluating Target Identification Results for the Fort Carson Dataset

This section first introduces the dataset we use for testing. It then summarizes how well targets are identified on 35 test cases.

3.1 The Fort Carson Dataset

In November 1993, data was collected by Colorado State University, Lockheed-Martin, and Alliant Techsystems at Fort Carson, Colorado. Over 400 range, IR and color images were collected and this imagery has been cleared for unlimited public distribution and Colorado State maintains a data distribution homepage (`http://www.cs.colostate.edu/~vision`). This homepage also includes a complete data browser for the color imagery. A 50 page report [7] describes each image, vehicles present, and ancillary information such as time of day and weather conditions. Additional information on the sensor calibration may be found in [33].

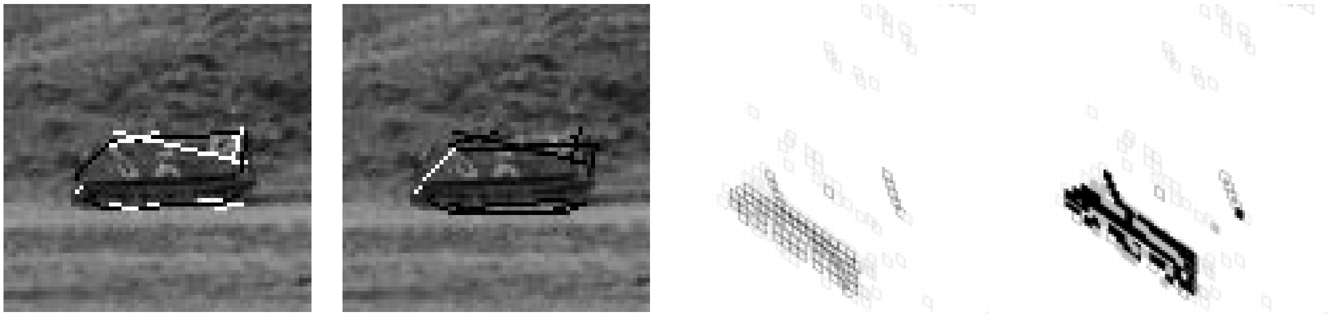
3.2 How Difficult is the Fort Carson Dataset?

The Fort Carson dataset was designed to contain challenging target identification problems requiring advancements to the state-of-the-art in ATR. We believe this goal has been met. To our knowledge, only one other organization has carried out target identification on this data, and that is the group from MIT Lincoln Laboratory. The Fort Carson dataset has been used in part of the evaluation of their own range-only ATR system [32].

The MIT group has also developed a set of correct-recognition performance curves that allow them to predict the best performance they can expect to achieve for given operating parameters (range, depression angle, noise, etc). In the case of the Fort Carson dataset, their curve of correct recognition versus range (which translates into a number of pixels on target for any given angular pixel size) indicates that their ATR system should be capable of achieving close to 100% correct recognition on the easiest imagery of the datasets where the vehicles occupy about 700 pixels. The same curve also predicted poor results on all the other images in the Fort Carson datasets where the numbers of pixels on target are much smaller. Even worse performance is expected due to the number of less than ideal conditions, such as obscurations and unusual viewing angles.

3.3 Our Experiment Design

Thirty five distinct range, IR and color image triples from the Fort Carson dataset were used in this test. These image triples represent over 90% of the total target views available in the dataset. The four targets present in these images are: M113, M901 (M113 with missile launcher), M60 and a pickup truck.



a. Initial Lines **b.** Final Lines **c.** Initial Points **d.** Final Points

Figure 11: Least-median squares results on real data. a) & b) are the initial estimates, c) & d) are the the results using median filtering.

The overall design and flow of this experiment is summarized in Figure 12. The upstream detection and hypothesis generation algorithms were used to generate realistic input for the multisensor matching system. However, these upstream algorithms are not the focus of this particular experiment and they were run in such as way as to maximally exercise the multisensor matching system. Put simply, we did not want to miss a chance to test the identification system due to a failure upstream. Different thresholds were used for the color system on different vehicle arrays.

For each region-of-interest produced by the target detection algorithm, the range boundary probing system was run using a four target probe-set. Since the conversion of the ROI from the color image to the range image is dependent upon knowing the current alignment between those two sensors, the process was repeated three times. In the first set, no alignment error was assumed. In the second set, random noise in the range $[0, 0.75]$ was added to each alignment dimension. The last set used noise in the range $[0, 1.5]$.

Our goal was to find a configuration for this probing system which gave us at least one 'reasonable' hypothesis in the top five ranked hypotheses. A reasonable hypothesis is one where the true target type is identified and the vehicle pose is within 60 degrees of correct. Using different probe-sets for near versus distant targets and hand generated tuning for each vehicle array, the system returned such 'reasonable' hypotheses in 33 out of the 35 cases.

While we did allow upstream tuning for specific vehicle arrays, we did not allow such tuning for the multisensor target ID system. As the focus of this evaluation, the ground rule was one configuration for all tests. All system input parameters were set to the same values for all 35 image triples.

3.4 How Well are Targets Identified

Table 2 presents a confusion matrix summarizing how well the multisensor identification system performed on the 35 test cases. The table shows the majority of the targets were correctly classified (27/35 or 77%). In two of the incorrect cases, hypothesis generation failed to suggest the correct target type.

Table 2: Confusion matrix for Multisensor Target Identification. Correct identification rate is 27/35 (77%). The two entries marked with '*' are cases where hypothesis generation failed to suggest the correct target type: entries #14 and #29 in Table 3.

		Multisensor System ID			
		M113	M901	M60	Pickup
True Target ID	M113	**7**		**1**	**1**
	M901	**1***	**5**	**2**	**1***
	M60		**1**	**7**	**1**
	Pickup				**8**

A detailed case-by-case breakdown is presented in Table 3. The second column indicates the vehicle shot number and vehicle array as identified in the Fort Carson data collection report [7]. The third column indicates the true target. The next five columns show the performance of the probing system, with the first four being the number of vehicle types returned out of 15 possible trials run. The fifth column shows the best probing output. A $\surd$ indicates the correct target has been identified.

The next column shows the target ID returned by the multisensor matching system. The fifth

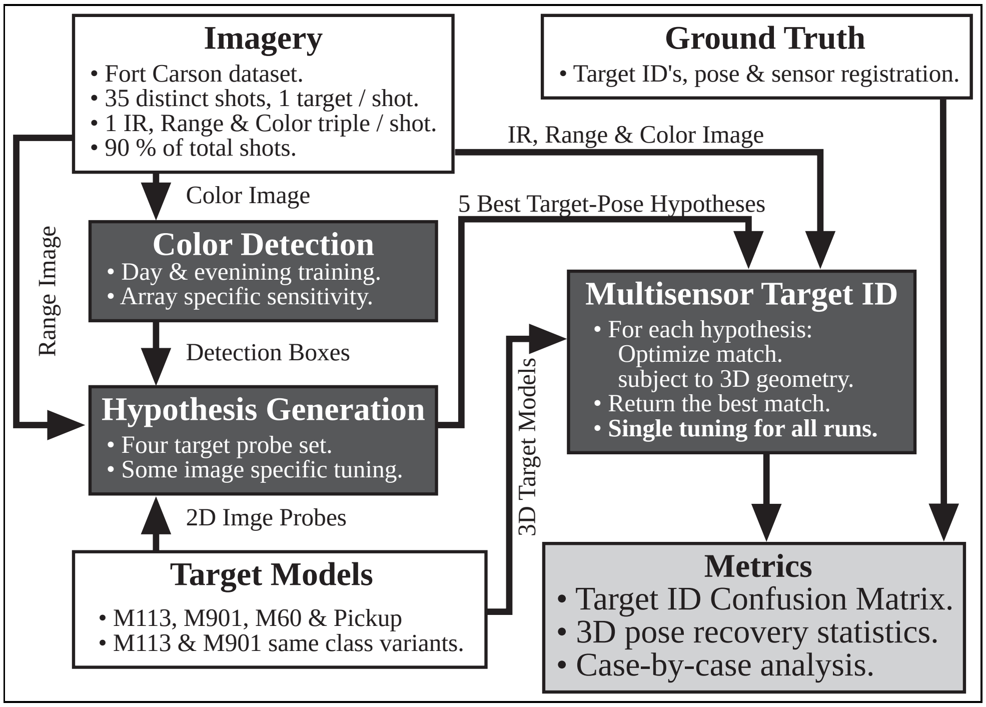

Imagery
• Fort Carson dataset.
• 35 distinct shots, 1 target / shot.
• 1 IR, Range & Color triple / shot.
• 90 % of total shots.
Ground Truth
• Target ID's, pose & sensor registration.
IR, Range & Color Image
Color Image
5 Best Target-Pose Hypotheses
Range Image
Color Detection
• Day & evenining training.
• Array specific sensitivity.
Multisensor Target ID
• For each hypothesis:
Optimize match.
subject to 3D geometry.
• Return the best match.
• Single tuning for all runs.
Detection Boxes
3D Target Models
Hypothesis Generation
• Four target probe set.
• Some image specific tuning.
2D Imge Probes
Target Models
• M113, M901, M60 & Pickup
• M113 & M901 same class variants.
Metrics
• Target ID Confusion Matrix.
• 3D pose recovery statistics.
• Case-by-case analysis.

Figure 12: Diagram of End-to-end ATR System Test.

column indicates the percentage of the target occluded in ten percent increments: blank indicates no occlusion. The final column indicates the number of range pixels on target.

In most cases, the system correctly distinguishes between very different targets, i.e., M60 versus M113. It also successfully discriminates between two variants of the same underlying vehicle. The M113 and M901 are identical except for the presence of a missile launcher mounted on the top of the M901. In one case where these two targets are confused, #14, the M901 is labeled an M113 because the missle launcher is completely obscured by an occluding tree.

Some other observations can be made looking at the data in Table 3. One is that identification performs perfectly on the high resolution data from Array 5: #17 through #20. Another not surprising observation is that even with our occlusion reasoning component, performance is better on non-occluded targets. There are 23 instances of non-occluded targets. Of these, only 2 are misidentified. That represents a better than 90% identification rate.

There are 12 occluded targets, of which 6 are correctly identified. Thus, even with our occlusion reasoning during matching, the identification rate is 50%. However, a related factor is the number of pixels on target, and of the 8 occluded targets with more than 50 pixels on target, 6 are correctly identified: an identification rate of 75%. While it is risky to conclude too much from so few instances, it appears that identification is breaking down at around 50 pixels on target.

The final observations to be made are about the performance of the multisensor system as compared to the probing algorithm. In many of the cases, the probing algorithm provided a wide range of vehicle types to the multisensor algorithm, and in only two instances was the correct vehicle type not present. The probing algorithm is operating at about 57% accuracy over all tests, and about 16% on occluded vehicles. However, it must be remembered it has been hand tuned for each vehicle array.

3.4.1 Analysis of Pose Recovery

Table 3 does not provide information about the pose of the best match found by either algorithm. Since both algorithms rely on object pose to identify targets, it is essential to compare the accuracy of the pose recovered by each. Pose can be broken down into two parts: rotation and translation. For simplicity, only the rotation error is analyzed here; a comparison over translation showed similar results.

A rotation error measure is formed by comparing a given pose to a pre-determined ground truth pose. The rotation error value can be thought of

Table 3: Case-by-case Breakdown of Target ID Results. The probing system required some image specific tuning in order to generate the results shown here. The Multisensor target recognition system used the same setting for all images.

Image	Shot	True	Hypothesis Generation					Multisensor	%	Target
#	Array	Target	M60	M901	M113	Pickup	Best	ID	Occlusion	Size
1	S01/A01	M60	3	2	6	4	M113	✓	-	62
2	S02/A01	M901	0	13	0	2	✓	✓	-	68
3	S04/A01	M113	3	4	4	4	✓	✓	-	54
4	S05/A01	Pickup	0	6	7	2	M113	✓	-	46
5	S06/A02	M60	8	4	0	3	✓	✓	-	144
6	S07/A02	M901	0	15	0	0	✓	✓	-	107
7	S08/A02	M113	4	1	10	0	✓	✓	-	91
8	S09/A02	Pickup	0	0	0	15	✓	✓	-	75
9	S10/A03	M60	5	4	0	6	✓	Pickup	-	129
10	S11/A03	M901	4	4	3	4	M113	M60	-	130
11	S12/A03	M113	0	0	13	2	✓	✓	-	113
12	S13/A03	Pickup	0	0	6	9	✓	✓	-	83
13	S14/A04	M60	7	1	4	3	M113	✓	20	181
14	S15/A04	M901	0	0	7	8	M113	M113	60	40
15	S16/A04	M113	0	0	9	6	Pickup	Pickup	50	25
16	S17/A04	Pickup	0	1	2	12	✓	✓	-	84
17	S18/A05	M60	15	0	0	0	✓	✓	-	683
18	S19/A05	M901	0	13	2	0	Pickup	✓	-	469
19	S20/A05	M113	0	0	15	0	✓	✓	-	691
20	S21/A05	Pickup	1	0	2	12	✓	✓	-	246
21	S22/A06	M60	5	0	6	4	M113	✓	10	180
22	S23/A06	M901	0	15	9	9	✓	✓	10	63
23	S24/A06	M113	0	0	15	0	✓	✓	-	85
24	S25/A06	Pickup	0	0	5	10	✓	✓	-	61
25	S26/A07	M60	5	5	5	0	M901	M901	10	101
26	S27/A07	M901	4	9	0	2	✓	M60	10	120
27	S28/A07	M113	9	1	5	0	✓	✓	-	122
28	S29/A08	M60	7	0	8	0	Pickup	✓	40	143
29	S30/A08	M901	1	0	2	12	Pickup	Pickup	80	20
30	S31/A08	M113	1	6	8	0	M60	M60	10	45
31	S32/A08	Pickup	0	3	0	12	✓	✓	-	118
32	S33/A09	M60	3	0	7	5	M113	✓	10	95
33	S34/A09	M901	2	8	0	5	M35	✓	60	80
34	S35/A09	M113	0	0	15	0	✓	✓	-	159
35	S36/A09	Pickup	0	2	3	10	M113	✓	-	48

as the amount of rotation required to rotate from estimated orientation to the ground truth orientation. For instance, a 30° error means a given pose estimate is 30° from the true orientation. Using this measure, a distribution is formed from the output of the hypothesis generation phase and also from the output of multisensor target identification algorithm.

Figure 13 shows the histogram comparing orientation error for the best target match in each of the 35 image triples reported in Table 3. Observe from the leftmost pair of bars in the histogram that multisensor matching increases four fold the number of matches within 5° of ground truth. It increases by a factor of two those between 5° and 10° of ground truth.

Hypothesis generation sometimes confuses vehicle fronts and backs. Note the right hand most

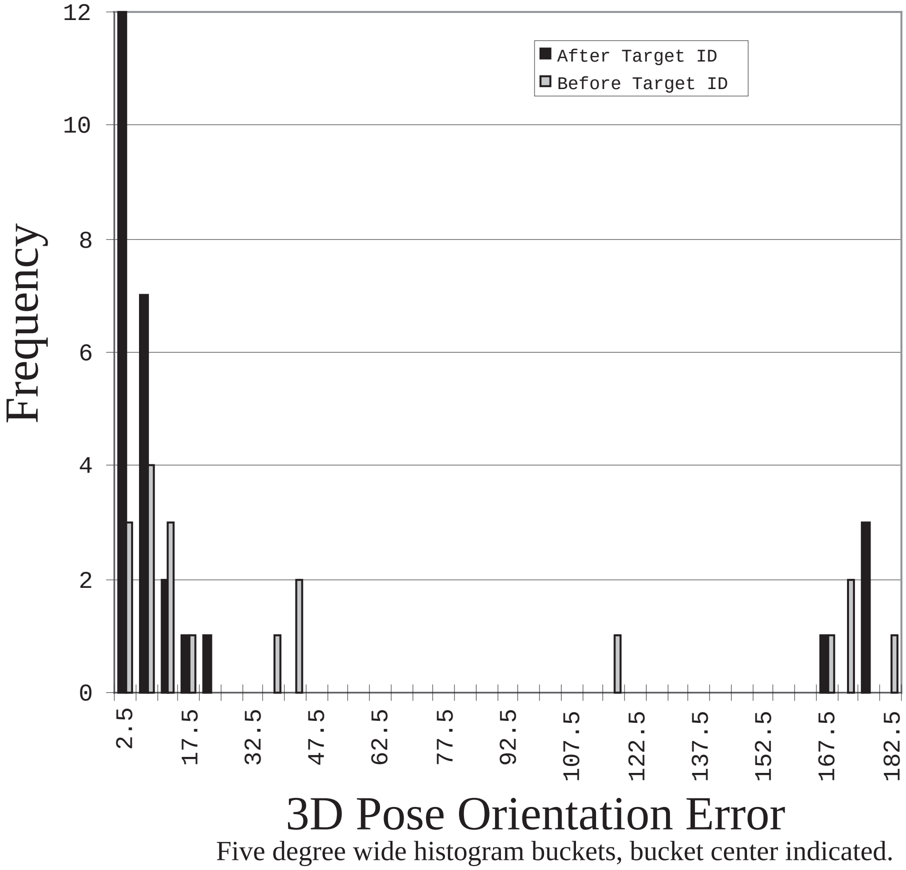


Figure 13: Difference between estimated and true 3-D target orientation before and after running of the multisensor identification algorithm. Bars labeled 'Before Target ID' indicate pose estimates coming out of the hypothesis generation algorithm, and those labeled 'After Target ID' are after multisensor matching has been run. Frequency is the count of trials with pose estimates differing from ground truth by the indicated amount.

histogram bars. This is in part because the algorithm relies solely upon the occluding contour. Currently, multisensor matching is unable to reverse orientations by 180° and these errors therefore persist in the final matches.

Before and after pose recovery results for a larger set of runs are shown in Figure 14. Where as in Figure 13, only the best match results are presented, Figure 14 shows before and after results for all runs where the initial target hypothesis is of the correct type and the pose within 90 degrees of the true orientation. As can be seen, the multisensor algorithm is able to substantially correct erroneous pose estimates.

The reliability of any model-based target identification procedure is clearly related to how well pose is recovered: without accurate pose the pairing of predicted model features to image measurements will be erroneous. For the most part, 23 out of 27 best matches, the multisensor algorithm recovers the true pose to within 20°. Because multisensor matching improves the pose estimates, it achieves more reliable target identification.

3.5 Two Examples

Before and after multisensor matching results for two specific images are shown in Figures 15 and 16 (see color plates), shots 20 and 26 repectively. For each of these images, the color detection algorithm successfully found each target. The pose hypothesis algorithm then provided a sequence of possible target type and pose hypotheses. The multisensor matching algorithm then refined the estimate to correct for pose and alignment errors. The results illustrated below show the best

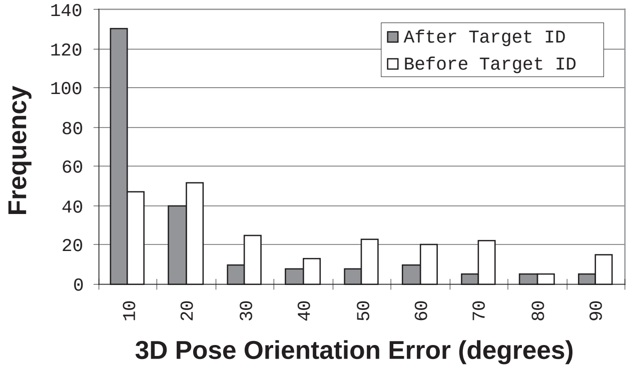


Figure 14: Difference between estimated and true 3-D target orientation before and after running of the multisensor identification algorithm.

match found by the multisensor matching algorithm. Recall that the best match is that which minimizes the match error defined in Section 2.7.2 and Section 2.8.

Figures 15a and 16a (see color plates) show the initial starting hypothesis for the matching algorithm. Starting from the top left corner of the image and moving clockwise, each image chip represents either different sensor-to-model relationships, or the sensor-to-sensor alignment. The upper left image shows the color image with the predicted model edges drawn in red and blue (red represents a non-omitted model feature). The next image shows the model in the initial orientation, followed by the IR image with the lines in white and black (black is non-omitted).

In the bottom row, the leftmost image shows the wireframe model in relation to the range data. The range data has been texture mapped with the color imagery, which allows the alignment between sensors to be visually assessed. The middle image shows the predicted model features in relation to the range sensor data. The blue boxes are data points and the red and yellow are predicted model points (red is non-omitted). The rightmost chip represents the range data with an IR texture map.

Figures 15b and 16b (see color plates) show the resulting pose and alignment after the multisensor matching system has refined these transformations. As can be seen from careful examination of the before and after imagery, the matching algorithm was able to substantially improve upon the model-to-sensor as well as the sensor-to-sensor relationships.

The multisensor matching algorithm took roughly 45 seconds to converge from the initial to final estimates for Shot26. Shot20 took longer, at 120 seconds, due largely to the greater number of range data points on target. Shot20 required 10 iterations of the local search algorithm and roughly 700 match error evaluations.

3.6 Removing a Bias for Small Targets

Our heuristic match evaluation function, the match error, is carefully normalized so as to not vary with target size. By design, the measure returns a value between zero and one regardless of whether the target is tiny (10 pixels on target) or large ($1,000$ pixels on target). A side effect of this normalization is that smaller target models tend to score slightly better than large target models. Speaking broadly, it is probably a consequence of the fact that smaller numbers of features are more likely to accidentally fit image clutter, including internal portions of larger targets.

To correct for the small target bias, i.e., the bias for pickup truck matches over M60 matches, a final linear adjustment is made to match errors based upon the predicted number of pixels on target. To perform this adjustment, the largest (S_{max}) and smallest (S_{min}) expected number pixels on target are determined for all the targets combined. Then, match errors for specific target instances are assessed a penalty proportional to match size s measured in pixels: smaller target matches incur a greater penalty.

$$E'_{\mathcal{M},\mathcal{S}}(\mathcal{F}) \; = \; w_{\mathcal{S}}(s) E_{\mathcal{M},\mathcal{S}}(\mathcal{F}) \tag{7}$$

$$w_{\mathcal{S}}(s) \; = \; \frac{(\gamma_{\mathcal{S}} - 1)\, s + S_{min} - \gamma_{\mathcal{S}} S_{max}}{S_{min} - S_{max}} \tag{8}$$

The scaled match error $E'_{\mathcal{M},\mathcal{S}}(\mathcal{F})$ for sensor $\mathcal{S}$ is adjusted by weight $w_{\mathcal{S}}(s)$, where

$$\begin{aligned} w_{\mathcal{S}}(S_{Min}) &= \gamma_{\mathcal{S}} \\ w_{\mathcal{S}}(S_{Max}) &= 1.0 \\ \gamma_{\mathcal{S}} &\geq 1.0 \end{aligned}$$

In the experiments reported above, the penalty for the smallest matches $\gamma_{\mathcal{S}}$ is 1.5, 1.5 and 1.1 for the range, color and IR sensors respectively. This simple modification has dramatically improved identification by correctly classifying the M60 7 times instead of 2 times without scaling. With the correction just explained, the system shows little or no bias in favor of smaller versus larger targets.

3.7 General Approach & Relative Sensor Weighting

A variety of thresholds, weights and step-size parameters are associated with the match error and the tabu-search process. Our general approach to tuning these parameters is to begin with what appears to be a 'common-sense' choice, and then to not vary the choice unless there is evidence of a problem. It has not been our goal in these early phases of work to explore the myriad possible tuning refinements.

Our one ground rule has been that whatever tuning we select, it must remain constant over the entire dataset being evaluated. Consequently, all the identification results reported above are for a single tuning of the multisensor target identification system. Since we have not yet explored the space of possible tunings, it is likely that a better tunings exists, and future refinements will probably lead to more robust target identification.

One set of weights is of special interest: the relative weight assigned to each sensor. All our experiments to date use a 50%, 30% and 20% weighting for range, color and IR respectively. However, changing these weights, for instance leaving out a sensor entirely, would allows us to assess the comparative value of sensors in terms of more or less reliable target identification.

We hope in the near future to begin to systematically explore the importance of each sensor by varying these weights and noting changes in performance. Our experience to date, given only a small amount of study, suggests that both the range and color data are important. There is less evidence that IR is helping. However, too much should not be read into this statement. Our current use of IR is somewhat naive: computing gradients rather than using a statistical measure of target/background differences. Enhancing our match quality measure for IR must go hand-in-hand with our aim of more thoroughly studying the relative value of each sensor.

4 Using Terrain Context

Terrain context plays a critical role in determining where targets can appear, how they will appear, what they may be doing, and perhaps most obviously, how far they are from a scout vehicle. A long-term goal of the RSTA project has been to introduce constraints derived from the analysis of the terrain. Examples in which terrain context is used include a terrain-guided search process developed by Lockheed-Martin [6] This algorithm directs the RSTA sensor suite to survey first those regions of a scene most likely to contain targets.

Another use of terrain is to derive range-to-target estimates. It may not at first be apparent, but most of the ATR algorithms used in RSTA require initial range-to-target estimates. These estimates are required to scale the templates and probe-sets used for recognition. This is true of the LADAR probing algorithm used above for hypothesis generation. Of course, when using LADAR, the range data itself provides the range-to-target estimate. More importantly, the IR detection and recognition systems developed by Lockheed-Martin and Hughes [19] require such an estimate. When working with IR or color imagery, it is less obvious how to generate good range-to-target estimates. Such information is not explicitly present in the optical imagery itself.

In the current RSTA system, range-to-target estimates for any pixel in an image are derived from the Digital Elevation Map (DEM) of the terrain. So long as the DEM is accurately registered to the optical imagery, it is a simple matter to intersect a ray passing through a given pixel with the terrain map and record the distance. The distance from the vehicle to the point of intersection with the terrain is the estimated range-to-target for a target centered at this pixel. However, there is an obvious weakness in this approach. Small errors in registration between the DEM and imagery can produce wildly incorrect range-to-target estimates.

On the current UGV vehicles, human intervention is required to accurately refine these estimates. Our project has begun work on automating this process. Results are presented for data collected at the UGV DEMO C site showing how automated matching of features extracted from the DEM to features extracted from imagery can register the DEM to imagery.

4.1 Vehicle Orientation Correction

When one of the SSVs using GPS and inertial guidance stops, small errors in pointing angle lead to large errors in pixel registration between imagery and the DEM. Orientation estimates can be off by one or more degrees [48]. The resulting uncertainty precludes terrain guided visual search and target recognition. To correct this uncertainty in pointing angle, the SSV transmits imagery from a sweep of the terrain to the operator work station. The operator then hand selects corresponding features on the stored terrain map and in the imagery. These corresponding control points enable the SSV to refine its estimated pointing angle relative to the terrain.

To automate this process, a matching system is provided features extracted from the DEM and

[6] Lockheed-Martin's Chapter 3 of this book includes additional detail on terrain-guided search.

features extracted from the imagery. The matching system establishes a 'best' correspondence between the two sets of features. Several different approaches to matching are being investigated for this problem. Foremost are a family of local search algorithms which find, with arbitrarily high probability, the optimal correspondence mapping and geometric transformation between a model and image data [10; 5; 6]. Two other techniques are also being studied. The first is a form of Genetic Algorithm called a 'Messy GA' [24]. The second is the Hausdorff matching algorithm developed by Huttenlocher [30], who has graciously provided us with the code for this algorithm.

Using any of these three approaches, the basic outline of the automated orientation correction procedure is the same:

- Render 3-D terrain using the estimated vehicle pose.
- Extract matchable features from the rendered terrain and actual imagery.
- Match the two sets of features.
- Use matched features in place of hand selected control points to correct the orientation estimate of the vehicle.

While the very specific problem of feature matching for orientation correction does not appear to have received much attention in the literature, there has been prior work on terrain feature matching which deserves mention. Matching of line segments representing dominant image features was proposed by Clark [18]. Levitt [40] proposed a way to select salient landmarks from terrain data [40] for navigation. Stein [21] uses panoramic horizon curve matching for vehicle localization. Thompson and Sutherland [56; 57; 55] have built a sophisticated expert system with a domain specific image feature extraction algorithm for abstracting structural terrain descriptions.

These prior efforts typically address the general problem of vehicle localization anywhere on a map, while the work presented here considers the more constrained problem of orientation correction. That said, the work here places a higher premium on precise matching of features from single narrow FOV imagery to support accurate orientation correction.

4.1.1 Terrain Rendering

The 5m digital elevation map (DEM) for the Demo C test site was obtained from Lockheed-Martin. This site was selected because test imagery taken directly from the SSV is available along with ground truth indicating vehicle position and pointing angle relative to fixed targets [47].

A terrain-rendering system has been developed using Open-GL which simulates the FOV of the CCD sensor used on the SSV. A simple lighting model is used and terrain is rendered from positions at which the vehicle actually acquired imagery. The vehicle pointing angle is derived from recorded vehicle and target positions: the targets are other military vehicles. Because target ground truth is being used to derive pointing angles, only images with targets near the image center are used. Figures 17a and 17b show two rendered terrain images for which matching is tested below.

4.1.2 Extracting Terrain and Image Features

The local search and messy GA matching algorithms match sets of line segments. For this problem, model and data segments are extracted from the rendered DEM and actual images respectively. An in-house implementation of the Burns algorithm [16] [7] is used to extract the line segment features. High frequency texture in these scenes prevents horizon features from being extracted unless the imagery is first smoothed: a 7x7 smoothing kernel has been used here. Even with smoothing, the horizons are still sometimes difficult to extract, and significant fragmentation occurs. Figures 17c and 17d show the images themselves along with the segments extracted by the Burns algorithm.

4.1.3 Image Feature Extraction on the SSV

Before saying more about matching extracted line segment features, it is worth noting that line extraction code was installed on the SSV vehicles. This was done to support horizon line orientation corrected carried out by an operator without the need for transmitting large amounts of data

[7] This version has a simple single Glyph interface and is publicly available from our FTP site: `ftp.cs.colostate.edu` in directory `/pub/vision/khoros-v1.0.5/CSUtools/csuExtrLn`

between the vehicle and the operator workstation (OWS). Colorado State provided Lockheed-Martin with an implementation of the Burns Line Extraction algorithm specifically tailored for use on the SSV and Lockheed-Martin integrated this software with the RSTA executive. The algorithm extracts the segments from imagery captured by the vehicle, sorts the features on the vehicle based upon a saliency measure, and then encodes the best 256 (or 1024) for transmission back to the OWS via the packet switched radio system.

Recognition of the need for this 'iconic' means of representing horizon features grew out of an analysis of how long it would take to transmit raw imagery for by-hand orientation correction at the OWS. With three vehicles all stopping to perform wide-area surveillance, it was estimated that during Demo II, it could take upwards of 15 minutes to transmit back all the required imagery for one round of orientation correction for 3 vehicles. It was decided that it is operationally unacceptable to introduce a 15 minute delay into operations each time all the vehicles stop to perform surveillance. To overcome this problem, we realized a person could designate control points to guide the orientation correction operation by looking at the 'iconic' terrain representation provided by a set of straight line segments. Moreover, the time required to transmit a set of straight line segments is nearly an order of magnitude less than that required to ship a raw image. Hence, the 15 minutes may be reduced to just under 2 minutes: a much more reasonable number from an operations standpoint.

4.1.4 Matching Using Local Search

A complete explanation of local search matching appears in Beveridge's dissertation [5] and 3-D matching results appear in [8]. A controlled performance analysis of 2-D matching appears in [9]. To briefly review the approach, an iterative generate-and-test strategy moves from a randomly selected initial match to one that is locally optimal. A global least-squares fitting process *always* aligns model and data for *any* correspondence tested. Thus, global geometry implicitly directs search. A match error takes account both of spatial fit and omission: how much of the model is un-matched.

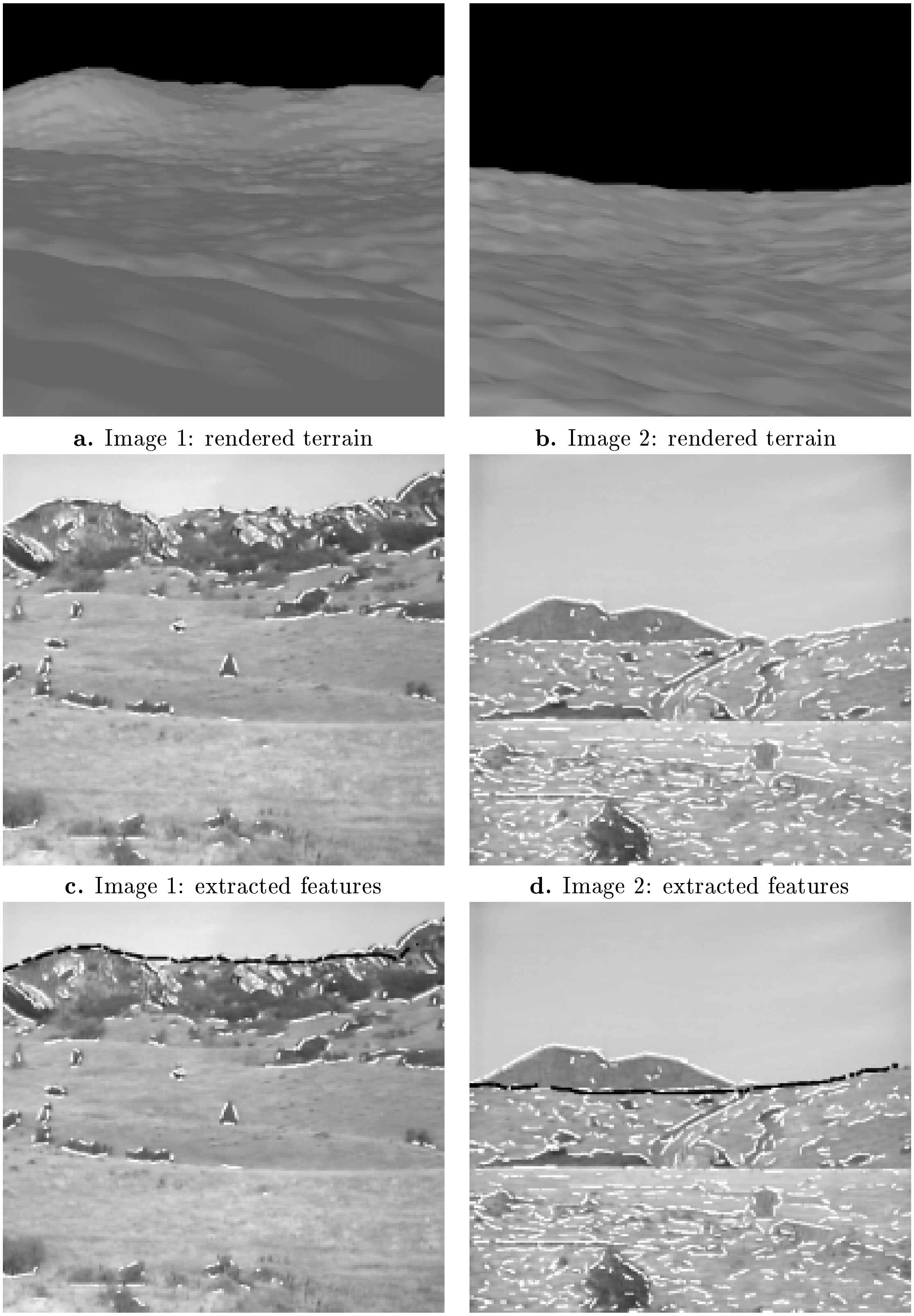
Search is conducted over a space of correspondence mappings C: C is the powerset of possibly matching features S. Most other algorithms consider one-to-many matches [27] while our C includes many-to-many matches. Without many-to-many mappings, properly matching piecewise approximations to curves with non-coincident breakpoints is impossible. This point is important here because horizon lines involve such non-coincident breakpoints.

While at first the initialization of search from *randomly* chosen matches may seem foolish, it is a strength of the approach. By running multiple trials from independently chosen initial matches, the probability of seeing the best (or near best) at least once may be made arbitrarily high. Past experience has demonstrated 100 trials is adequate to solve most difficult problems [8; 9]. Another benefit of multiple trials is the structure and frequency of alternative solutions tells us much about the difficulty of a particular problem.

Not all possible pairs of line features need be considered in matching. Two constraints limit the space of possible matches between horizon model and image line segments. First, it is safe to assume the horizon lies somewhere in a band that is half of the height of the image centered about the true position. Second, the relative orientation between segments must be less than 17 degrees for them to match. With these constraints, the sets of potentially matching features is still large: 1183 for Image 1 and 1577 for Image 2. The resulting search spaces C contain 2^{1183} and 2^{1577} states respectively.

To explore the space of possible matches, 500 trials of subset-convergent local search were run on each problem. The best match found in each case is shown superimposed in black in Figures 17e and 17f. In both cases, visual inspection shows these to be essentially correct matches. For Images 1 and 2, the best matches were found in a single trial with probabilities 0.056 and 0.036 respectively. Based upon this probability, it follows this match may be found with better than 95% confidence running 59 and 90 trials respectively. Being conservative, 100 trials is more than sufficient.

These are large problems, in terms of search space, for local search matching. To find these matches reliably, the current C implementation running on a Sparc 20 requires on the order of 20 minutes for Image 1 and an hour for Image 2. Clearly, either some domain-specific tuning or use of parallel hardware is required to bring run-times down. Both of these are very reasonable options for future work. Use of a better feature extraction algorithm would dramatically simplify the combinatorics, and parallel local search is trivial due to the independence of trials.



a. Image 1: rendered terrain

b. Image 2: rendered terrain

c. Image 1: extracted features

d. Image 2: extracted features

e. Image 1: horizon match

f. Image 2: horizon match

Figure 17: Results of Local Search Matching on Two Horizon Images

4.1.5 Matching With a Messy Genetic Algorithm

A messy genetic algorithm is one which can grow and shrink the 'chromosomes' which represent states in a search space. In practical terms, this means the strings representing individuals in the population can vary in size. For matching, the strings are made up from pairs of model and data line segments. The worth of any given string is the same match error used above by the local search procedure. The match error is defined for all possible combinations of model and data features.

Messy genetic algorithms are initialized with small strings of high relative value. For a matching problem with n potentially matching pairs of model and data segments, on the order of n triples of model and data line segments are introduced into the initial population. Triples are built using proximity constraints to avoid enumerating all n^3 possibilities. These triples are then ranked by match error. Triples with low match error rise to the top of the population and are favored in subsequent recombination operations.

In the recombination phase of the messy GA, strings with better fitness values, i.e., low match error, are more likely to be chosen. The combination operator takes two parent strings, cuts each parent at a random point to form four substrings, and creates two new strings by merging substrings from different parents. After recombination, one of the two merged children is selected at random and inserted back into the population. The population is maintained in sorted order from best to worst, and after insertion, the worst string is dropped from the population.

Because the match evaluation function favors larger strings of consistent paired features, the overall string length in the population tends to increase. In order to force a bounded run-time on the messy GA, after a set number of recombinations, the overall population size is reduced by dropping the worst individual. The algorithm terminates when the population size drops below a preset threshold: ten strings in the experiments run here. The messy GA used here is also a hybridized form of genetic algorithm in the following sense: local search is used sparingly to improve individuals within the evolving population.

Early results suggest that the messy GA is performing much better than the random starts local search algorithm. Qualitatively, it has expanded the practical threshold on the size of problems being solved. The messy GA is reliably solving large problems, including horizon problems similar to those shown in Figure 17, with over $6,000$ pairs of potentially matching features. This is three times as many pairs as the largest problem solved using random-starts local search. For these larger problems, the messy GA requires on the order of 1 hour on a Spar 10. The messy GA has not yet been tested on the exact same set of pairs for used in the local search results shown in Figure 17. However, based upon the performance with larger problems the messy GA should solve the smaller problems in several minutes.

4.1.6 Matching with the Hausdorff Metric

A modest effort was made to match horizons using the Hausdorff metric using a software package graciously provided to us by Huttenlocher [30]. Two binary images are input to the system where 'on' pixels represent local edges in the imagery. We used local edges extracted from the thresholded (sky/ground) rendered terrain and from the grey-scale imagery.

Initially, we were optimistic about Hausdorff matching, since the use of edge images rather than extracted straight line features ought to make representing curved horizons easier. However, our efforts quickly ran into problems. We learned through the optimal transformations found by local search matching that there were small scale changes (on the order of 5%) between our rendered terrain and actual images. These are due to imperfect sensor calibration [8]. There were also small rotations due to the vehicle being parked on irregular ground. The Hausdorff system exhaustively samples over possible scales and rotations, and configuring it to account for these variations slowed it down considerably: very roughly speaking run-times were comparable to the random starts local search.

A greater problem with matching horizons using the Hausdorff metric involved how to preselect the desired match quality. The horizon line images have a lot of edge clutter under the true horizon. This clutter can easily fall within the matching threshold and lead to matches of 'quality' comparable to the true match. More study and refined tuning might clear up these problems, but in our initial tests the system often missed the true horizon. While these initial experiences were not encouraging, it is critical to understand that we did not devote a great deal of effort to overcoming our initial problems. Consequently, it would be quite unfair to conclude too much.

[8] Such changes in scale are not a problem for the local search matching, which best fits the model to the data subject to 2-D rotation, translation and scale. As suggested, the optimal matching recovers the scale change between rendered terrain and imagery.

In future, perhaps in collaboration with Huttenlocher [30], we hope to devote the time and energy to conduct a more thorough study.

4.2 Automated Orientation Correction Tests on Demo C Site Data

To study the ability of the full system described above to improve vehicle orientation estimates, an early test-of-concept experiment has been conducted for a single vehicle placement in a data collection conducted at the Lockheed-Martin Demo C site in September of 1994. The imagery was collected to test RSTA target recognition algorithms. Fortunately, sufficient ground truth data was collected to allow this same dataset to serve as a testbed for automated orientation correction.

For this vehicle placement the vehicle position is known using the SSV GPS system. The vehicle orientation has been recovered by measuring pointing angles to surveyed points on the test site. To test the ability of automated feature matching to recover true vehicle pointing angle, these ground truth estimates are perturbed in a controlled fashion to generate 27 estimated orientations. The precise perturbations are: pan angle -2, 0 and 2 degrees, tilt angle -1, 0 and 2 degrees, and roll angle -5, 0, and 5 degrees.

For all 27 cases, the DEM terrain was rendered, model features were extracted from this rendered image, and these were matched to the image features. The corresponding features returned by local search matching were used to update the vehicle orientation estimate. The initial orientation estimates are off by up to 7 degrees, while the recovered orientation estimates are never off by more than 0.8 degrees. The average difference from ground truth after orientation correction is about 0.5 degrees.

5 Summary

Colorado State, with its team members Alliant Techsystems and the University of Massachusetts, has built an end-to-end target identification system. This research testbed has demonstrated what we consider to be a qualitative advance in the the state-of-the-art for ground-based, multi-sensor ATR.

In particular, we have demonstrated the value of developing target signature predictions on-line to fit specific scene contexts. Prediction exploits known collateral knowledge such as time-of-day. Prediction also infers information about target occlusion from the range data. Within the target verification module, iterative use of the prediction algorithm develops precise scene specific target signatures. These capabilities have been demonstrated over 90% of the available target instances in our Fort Carson dataset, and target identification rates are above 90% for unoccluded targets. More importantly, for occluded targets and low resolution targets (100 pixels on target), our system is performing better than any other system known to us.

We have also demonstrated the value of color as a cue for target detection. Unlike most IR detection algorithms, color detection requires no collateral knowledge about range to targets. It does, of course, presuppose daytime operations and it requires training imagery. The success of color target detection has two short term implications. First, the underlying multivariate classification algorithm is equally applicable to other forms of multivariate data, such as multispectral IR or polarized E-O imagery. Second, the technology demonstrated on the SSV vehicles could be immediately used for a remote sentry where training takes place at the time of deployment.

Overall we have tried in this project to balance basic research and technology demonstration, and much of what has been demonstrated rests upon new algorithms. At many points in the development of the target identification system, choices had to be made between alternative paths. In most cases, the simplest or most obvious path was followed and the examination of alternatives put off to a later date. In so doing, we accomplished our most important goal: demonstrating the value of on-line feature prediction by showing that a complete system could solve difficult real world problems. However, since we have postponed careful study of many critical decisions, it is important that work continues on systems such as ours.

References

[1] A. Akerman and R. Patton and W. Delashmit and R. Hummel. Target Identification Using Geometric Hashing and FLIR/LADAR fusion. In *Proceedings: Image Understanding Workshop*, pages 595 – 618, Los Altos, CA, February 1996. ARPA, Morgan Kaufmann.

[2] Anthony N. A. Schwickerath and J. Ross Beveridge. Coregistering 3D Models, Range, and Optical Imagery Using Least-Median Squares Fitting. In *Proceedings: Image Understanding Workshop*, pages 719–722, Los

Altos, CA, February 1996. ARPA, Morgan Kaufmann.

[3] Anthony N. A. Schwickerath and J. Ross Beveridge. Coregistration of Range and Optical Images Using Coplanarity and Orientation Constraints. In *1996 Conference on Computer Vision and Patter Recognition*, pages 899 – 906, San Francisco, CA, June 1996.

[4] Dana H. Ballard and Christopher M. Brown. *Computer Vision.* Prentice-Hall, Inc, Englewood Cliffs, New Jersey, 1982.

[5] J. Ross Beveridge. *Local Search Algorithms for Geometric Object Recognition: Optimal Correspondence and Pose.* PhD thesis, University of Massachusetts at Amherst, May 1993.

[6] J. Ross Beveridge, Allen Hanson, and Durga Panda. Model-based Fusion of FLIR, Color and LADAR. In Paul S. Schenker and Gerard T. McKee, editors, *Proceedings: Sensor Fusion and Networked Robotics VIII*, Proc. SPIE 2589, pages 2 – 11, October 1995.

[7] J. Ross Beveridge, Durga P. Panda, and Theodore Yachik. November 1993 Fort Carson RSTA Data Collection Final Report. Technical Report CSS-94-118, Colorado State University, Fort Collins, CO, January 1994.

[8] J. Ross Beveridge and Edward M. Riseman. Optimal Geometric Model Matching Under Full 3D Perspective. *Computer Vision and Image Understanding*, 61(3):351 – 364, 1995. (short version in IEEE Second CAD-Based Vision Workshop).

[9] J. Ross Beveridge, Edward M. Riseman, and Christopher Graves. Demonstrating Polynomial Run-Time Growth for Local Search Matching. In *Proceedings: International Symposium on Computer Vision*, pages 533 – 538, Coral Gables, Florida, November 1995. IEEE PAMI TC, IEEE Computer Society Press.

[10] J. Ross Beveridge, Rich Weiss, and Edward M. Riseman. Combinatorial Optimization Applied to Variable Scale 2D Model Matching. In *Proceedings of the IEEE International Conference on Pattern Recognition 1990, Atlantic City*, pages 18 – 23. IEEE, June 1990.

[11] James E. Bevington. Laser Radar ATR Algorithms: Phase III Final Report. Technical report, Alliant Techsystems, Inc., May 1992.

[12] Elie Bienstock, Donald Geman, Stuart Geman, and Donald E. McClure. Development of laser radar atr algorithms: Phase ii - military objects. Technical report, Mathematical Technologies Inc., Providence, Rhode Island, October 1990. Prepared under Harry Diamond Laboratories Contract No. DAAL02-89-C-0081.

[13] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. Classification and regression. Technical report, Wadsworth International Group, Belmont, CA, 1984.

[14] C. E. Brodley and P. E. Utgoff. Goal-directed Classification Using Linear Machine Decision Trees. *Machine Learning*, 1994.

[15] Shashi Buluswar, Bruce A. Draper, Allen Hanson, and Edward Riseman. Non-parametric Classification of Pixels Under Varying Outdoor Illumination. In *Proceedings: Image Understanding Workshop*, pages 1619–1626, Los Altos, CA, November 1994. ARPA, Morgan Kaufmann.

[16] J. B. Burns, A. R. Hanson, and E. M. Riseman. Extracting straight lines. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI–8(4):425 – 456, July 1986.

[17] T.W. Chen and W.C. Lin. A neural-network approach to csg-based 3-d object recognition. *pami*, 16(7):719–726, july 1994.

[18] C.S. Clark, D.K. Conti, W.O. Eckhardt, T.A. McCulloh, R. Nevatia, and D.Y. Tseng. Matching of Natural Terrain Scenes. In *ICPR80*, pages 217–222, 1980.

[19] David M. Doria and Daneil P. Huttenlocher. Progress on the Fast Adaptive Target Detection Program. In *Proceedings: Image Understanding Workshop*, pages 719–722, Los Altos, CA, February 1996. ARPA, Morgan Kaufmann.

[20] R. O. Eason and R. C. Gonzalez. Least-Squares Fusion of Multisensor Data. In Mongi A. Abidi and Rafael C. Gonzalez, editors, *Data Fusion in Robotics and Machine Intelligence*, chapter 9. Academic Press, 1992.

[21] Fridtjof Stein and Gerard Medioni. Map-based Localization using the Panoramic Horizon. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pages 2631 – 2637, Nice, France, May 1992.

[22] S. Ghosal and R. Mehrotra. Range surface characterization and segmentation using neural networks. *Pattern Recognition*, 28(5):711–727, may 1995.

[23] F. Glover. Tabu search – part i. *ORSA Journal on Computing*, 1(3):190 – 206, 1989.

[24] David E. Goldberg, Kalyanmoy Deb, Hillol Kargupta, and Georges Harik. Rapid, accurate optimization of difficult problems using fast messy genetic algorithms. In Stephanie Forrest, editor, *Proc. 5th International Conference on Genetic Algorithms*, pages 56–64. Morgan-Kaufmann, 1993.

[25] M. E. Goss, J. R. Beveridge, M. Stevens, and A. Fuegi. Three-dimensional visualization environment for multisensor data analysis, interpretation, and model-based object recognition. In *IS&T/SPIE Symposium on Electronic Imaging: Science & Technology*, pages 283 – 291, February 1995.

[26] Michael E. Goss, J. Ross Beveridge, Mark Stevens, and Aaron Fuegi. Visualization and Verification of Automatic Target Recognition Results Using Combined Range and Optical Imagery. In *Proceedings: Image Understanding Workshop*, pages 491 – 494, Los Altos, CA, November 1994. ARPA, Morgan Kaufmann.

[27] W. Eric L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints.* MIT Press, Cambridge, MA, 1990.

[28] B. K. P. Horn. *Robot Vision.* The MIT Press, 1986.

[29] Hua Chen and Lawrence B. Wolff. A Polarization Phase-Based Method For Material Classification And Object Recognition in Computer Vision. In *Proceedings: Image Understanding Workshop*, pages 1297 – 1303, Los Altos, CA, February 1996. ARPA, Morgan Kaufmann.

[30] Daniel P. Huttenlocher, Gregory A. Klanderman, and William J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 850 – 862, September 1993.

[31] Daniel P. Huttenlocher and Shimon Ullman. Recognizing Solid Objects by Alignment with an Image. *International Journal of Computer Vision*, 5(2):195 – 212, November 1990.

[32] Jacques G. Verly and Richard T. Lacoss. Automatic Target Recognition for LADAR imagery Using Functional Templates Derived From 3-D CAD Models. In Oscar Firschein, editor, *Reconnaissance, Surveillance, and Target Acquisition (RSTA) for the Unmanned Ground Vehicle.* Morgan Kaufmann, 1997.

[33] Zhongfei Zhang J.R. Beveridge, M.R. Stevens and M.E. Goss. Approximate Image Mappings Between Nearly Boresight Aligned Optical and Range Sensors. Technical Report CS-96-112, Computer Science, Colorado State University, Fort Collins, CO, April 1996.

[34] D. Judd, D. MacAdam, and G. Wyszecki. Spectral distribution of typical daylight as a function of correlated color temperature. *Journal of the Optical Society of America*, 54(8):1031–1040, 1964.

[35] Alana Katz and Philip Thrift. Hybrid neural network classifiers for automatic target detection. *Expert Systems*, 10(4):243, November 1993.

[36] Rakesh Kumar. Determination of Camera Location and Orientation. In *Proceedings: Image Understanding Workshop*, pages 870 – 881, Los Altos, CA, June 1989. DARPA, Morgan Kaufmann.

[37] Rakesh Kumar. *Model Dependent Inference of 3D Information From a Sequence of 2D Images.* PhD thesis, University of Massachusetts, COINS TR92-04, Amherst, February 1992.

[38] Rakesh Kumar and Allen R. Hanson. Robust methods for estimating pose and a sensitivity analysis. *CVGIP:Image Understanding*, 11, 1994.

[39] Lawrence B. Wolff. Reflectance Modeling for Object Recognition and Detection in Outdoor Scenes. In *Proceedings: Image Understanding Workshop*, pages 799 – 803, Los

Altos, CA, February 1996. ARPA, Morgan Kaufmann.

[40] T.S. Levitt, D.T. Lawton, D.M. Chelberg, and P.C. Nelson. Qualitative landmark-based path planning and following. In *AAAI-87*, pages 689–694, 1987.

[41] David G. Lowe. *Perceptual Organization and Visual Recognition.* Kluwer Academic Publishers, 1985.

[42] Mark R. Stevens and J. Ross Beveridge. Interleaving 3D Model Feature Prediction and Matching to Support Multi-Sensor Object Recognition. In *Proceedings: Image Understanding Workshop*, pages 699–706, Los Altos, CA, February 1996. ARPA, Morgan Kaufmann.

[43] Mark R. Stevens and J. Ross Beveridge. Optical Linear Feature Detection Based on Model Pose. In *Proceedings: Image Understanding Workshop*, pages 695–697, Los Altos, CA, February 1996. ARPA, Morgan Kaufmann.

[44] Mark R. Stevens and J. Ross Beveridge. Precise Matching of 3-D Target Models to Multisensor Data. *IEEE Transactions on Image Processing*, 6(1):126–142, January 1997.

[45] L.I. Perlovsky, J.A. Chernick, and W.H. Schoendorf. Multisensor atr and identification of friend or foe using mlans. *Neural Networks*, 8(7-8):1185–1200, 1995.

[46] J.R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[47] Ray Rimey. RSTA Sept94 Data Collection Final Report. Technical report, Martin Marietta Astronautics, Denver, CO, January 1995.

[48] Ray Rimey and Darrell Hougen. Discussion of SSV Orientation Correction with Lockheed-Martin RSTA Group. Personal Correspondence, 1995.

[49] Peter. J. Rousseeuw and Annick. M. Leroy. *Robust regression & outlier detection.* John Wiley & Sons, New York, 1987.

[50] Anthony N. A. Schwickerath. Simultaneous refinement of pose and sensor registration. Master's thesis, Colorado State University (TechReport CS-97-101), 1996.

[51] Anthony N. A. Schwickerath and J. Ross Beveridge. Model to Multisensor Coregistration with Eight Degrees of Freedom. In *Proceedings: Image Understanding Workshop*, pages 481 – 490, Los Altos, CA, November 1994. ARPA, Morgan Kaufmann.

[52] Mark R. Stevens. Obtaining 3D Silhouettes and Sampled Surfaces from Solid Models for use in Computer Vision. Master's thesis, Colorado State University, Fort Collins, Colorado, September 1995.

[53] Mark R. Stevens, J. Ross Beveridge, and Michael E. Goss. Reduction of BRL/CAD Models and Their Use in Automatic Target Recognition Algorithms. In *Proceedings: BRL-CAD Symposium.* Army Research Labs, June 1995.

[54] Mark R. Stevens, J. Ross Beveridge, and Mike Goss. Visualization of multi-sensor model-based object recognition. *IEEE Transactions on Visualization and Computer Graphics*, (submitted).

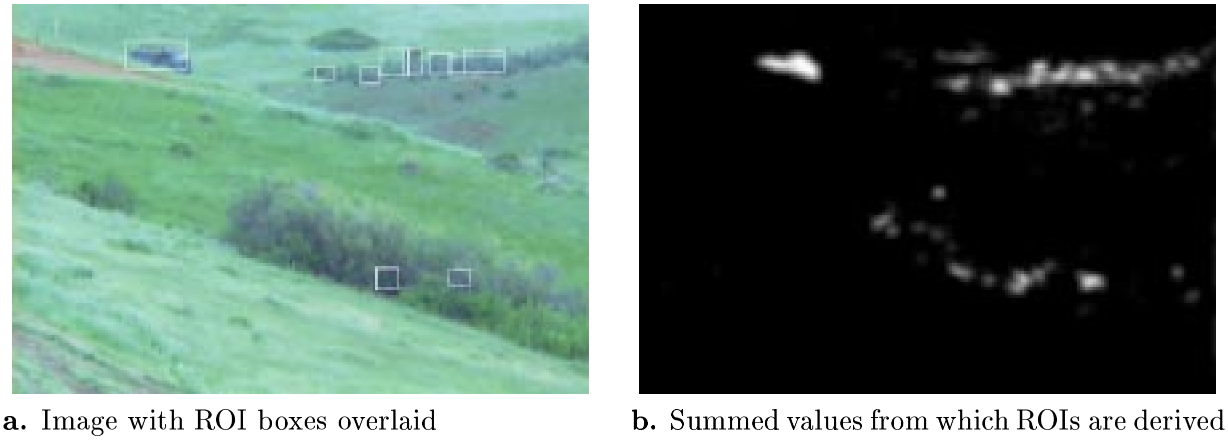
[55] Sutherland, K.T. and Thompson, W.B. Localizing in Unstructured Environments: Dealing with the Errors. *Robotics and Automation*, 10:740–754, 1994.

[56] W.B. Thompson, T.C. Henderson, T.L. Colvin, L.B. Dick, and C.M. Valiquette. Vision-Based Localization. In *Proceedings: Image Understanding Workshop*, pages 491–498, Los Altos, CA, 1993. ARPA, Morgan Kaufmann.

[57] Thompson, W.B. and Pick, Jr., H.L. Vision-Based Navigation. In *Proceedings: Image Understanding Workshop*, pages 127–131, Los Altos, CA, 1993. ARPA, Morgan Kaufmann.

[58] U. S. Army Ballistic Research Laboratory. *BRL-CAD User's Manual*, release 4.0 edition, December 1991.

[59] Allen M. Waxman, Micheal C. Seibert, Alan Gove, David A. Fay, Ann Marie Bernardon, Carol Lazott, William R. Steele, and Robert K. Cunningham. Neural processing of targets in visible, multispectral ir and sar imagery. *Neural Networks*, 8(7/8):1029, may 1995.



a. Image with ROI boxes overlaid **b.** Summed values from which ROIs are derived

Figure 2: Color Detection Example in UGV Data from Demo C Test Sit

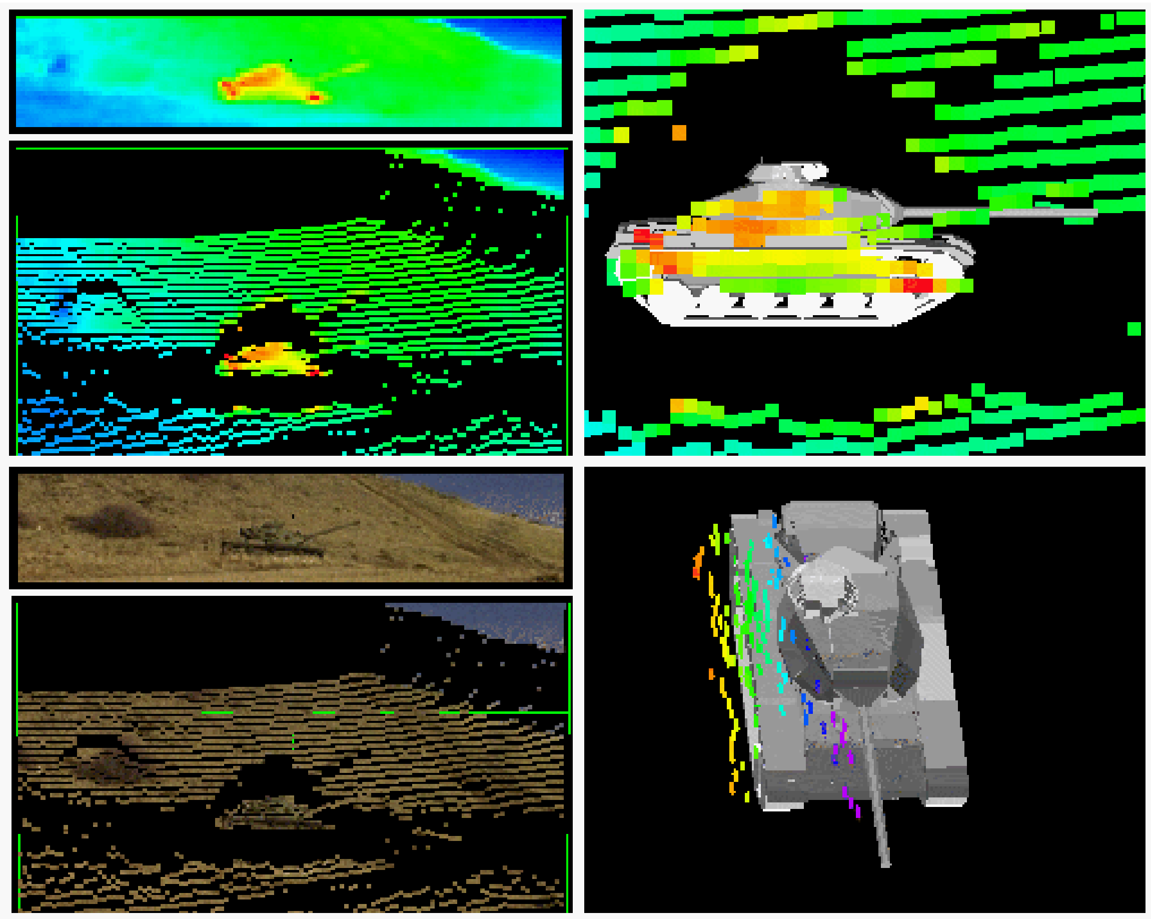


Figure 6: Sample Screen Images of the RangeView System

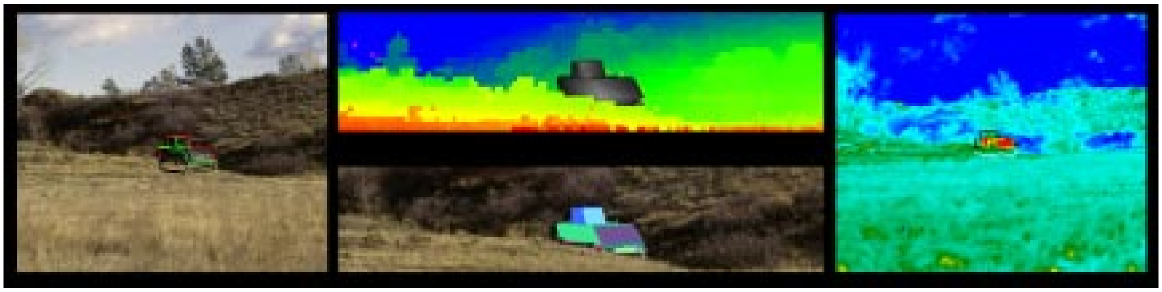
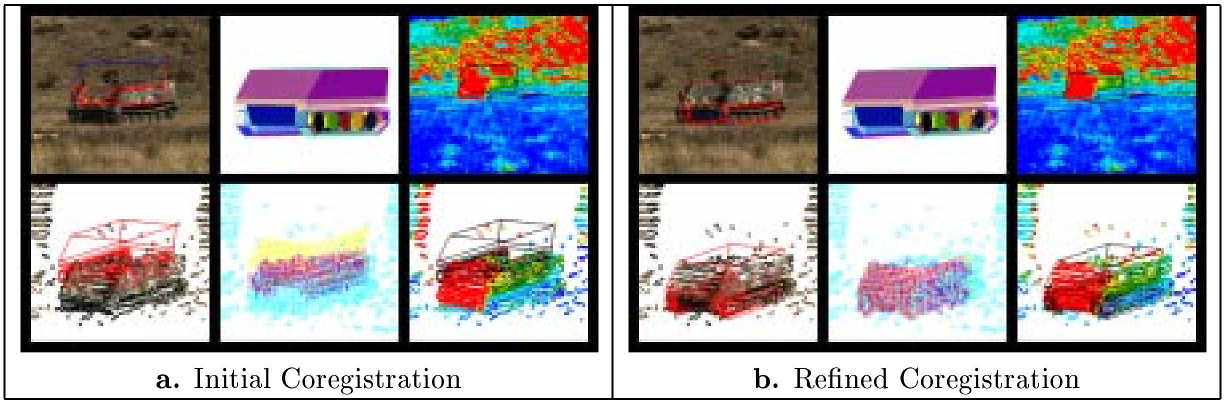


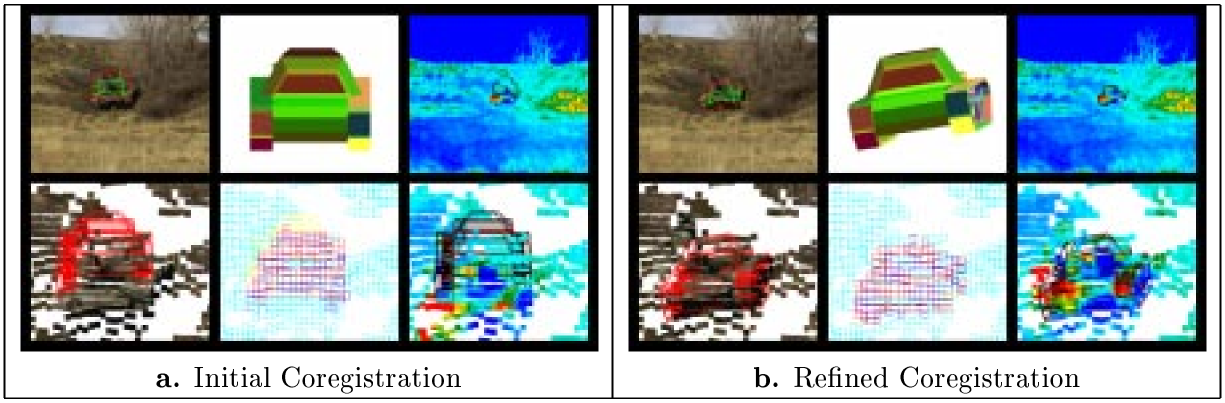
Figure 9: Shot34 Occlusion Example



a. Initial Coregistration

b. Refined Coregistration

Figure 15: Shot20 Multisensor Target Matching Results



a. Initial Coregistration

b. Refined Coregistration

Figure 16: Shot26 Multisensor Target Matching Results