CAD-based Target Identification in Range, IR and Color Imagery Using On-Line Rendering and Feature Prediction [*]

J. Ross Beveridge Mark R. Stevens

Affiliation of Authors

Dr. J. Ross Beveridge received his B.S. degree in Applied Mechanics and Engineering Science from the University of California at San Diego in 1980 and his M.S. and Ph.D. degrees in Computer Science from the University of Massachusetts in 1987 and 1993 respectively. He has been an Assistant Professor in the Computer Science Department at Colorado State University since 1993. His present interests include object recognition, robot navigation, sensor fusion, image feature extraction, and software development environments for computer vision. He is a member of the ARPA Image Understanding Environment Technical Advisory Committee.

Mark R. Stevens received his B.S. degree in Computer Science from the University of Maine at Orono in 1993, and his M.S. in Computer Science from Colorado State University in 1995. He is currently working on his Ph.D. under Dr. Beveridge in the area of computer vision. His present interests include object recognition, multi-sensor visualization, solid-modeling and neural computing. He is a member of IEEE, AAAI and ACM.

Contact Information

J. Ross Beveridge
601 South Howes Street
Colorado State University
Computer Science Department
Fort Collins, Colorado 80523
ross@cs.colostate.edu

[*] 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, and the National Science Foundation under grants CDA-9422007 and IRI-9503366

Abstract

Results for a mutlisensor CAD-based object recognition system are presented in the context of Automatic Target Recognition using nearly boresight-aligned range, IR and color sensors. The system is shown to identify targets in test suite of 35 image triples. This suite includes targets at low resolution, unusual aspect angles, and partially obscured by terrain. The key concept presented in this work is that of using on-line rendering of CAD-models to support an iterative predict, match and refine cycle. This cycle optimizes the match subject to variability both in object pose and sensor registration. An occlusion reasoning component further illustrates the power of this approach by customizing the predicted features to fit specific scene geometry. Occlusion reasoning detects occlusion in the range data and adjusts the features predicted to be visible accordingly.

List of Symbols

Symbol	Meaning
$\mathcal{F}$	Coregistration parameters
$E_{\mathcal{M}}(\mathcal{F})$	Match error given coregistration $\mathcal{F}$
$\mathcal{S}$	A sensor
$\mathcal{C}$	Color sensor
$\mathcal{I}$	IR sensor
$\mathcal{R}$	Range sensor
$E_{\mathcal{M},\mathcal{C}}(\mathcal{F})$	Color error given coregistration $\mathcal{F}$
$E_{\mathcal{M},\mathcal{I}}(\mathcal{F})$	IR error given coregistration $\mathcal{F}$
$E_{\mathcal{M},\mathcal{R}}(\mathcal{F})$	Range error given coregistration $\mathcal{F}$
$\alpha_{\mathcal{C}}$	Weighting term for combining the color error
$\alpha_{\mathcal{I}}$	Weighting term for combining the IR error
$\alpha_{\mathcal{R}}$	Weighting term for combining the range error
$E_{fit,\mathcal{S}}(\mathcal{F})$	The fitness error for sensor $\mathcal{S}$ given coregistration $\mathcal{F}$
$E_{om,\mathcal{S}}(\mathcal{F})$	The omission error for sensor $\mathcal{S}$ given coregistration $\mathcal{F}$
$E_{oc,\mathcal{R}}(\mathcal{F})$	The range occlusion error given coregistration $\mathcal{F}$
$\beta_{\mathcal{S}}$	Weighting term for combining fitness and omission error
$E'_{\mathcal{M},\mathcal{S}}(\mathcal{F})$	The Scaled match error given coregistration $\mathcal{F}$
$w_{\mathcal{S}}(s)$	The scaling term for a given sensor $\mathcal{S}$
S_{min}	The minimum value for a given sensor $\mathcal{S}$ used for scaling
S_{max}	The maximum value for a given sensor $\mathcal{S}$ used for scaling
$\gamma_{\mathcal{S}}$	The Scale value given sensor $\mathcal{S}$

1 Introduction

The utility of CAD-based recognition techniques has long been recognized for industrial domains where detailed geometric models are available. It has also shown promise for Automatic Target Recognition [29, 13]. However, ATR is a highly challenging object recognition domain: targets typically appear at low resolution, sensor modalities other than visible light are typically most important, and targets are viewed against cluttered backgrounds. So far, many of the common techniques in CAD-based vision have adapted poorly to this domain.

To adapt the best features that CAD-based recognition has to offer to the task of identifying vehicles in multisensor data, we have developed a model-based ATR system [33] and tested it against what we consider to be a demanding and realistic dataset [8]. The system is based on the conviction that a CAD-based recognition system should utilize 3D models dynamically during recognition to make predictions about what features will be visible. Moreover, this prediction capability should be embedded within a local optimization framework which converges upon the most globally consistent interpretation relative to known object, sensor, and scene constraints.

Our system uses graphics hardware to efficiently render a 3D object model using the believed camera position. From the rendered image, those features most likely to be detectable in imagery are selected for matching. A tabu search uses these features to explore variations in the object's 3D pose (position and orientation) relative to a sensor suite as well as the registration between heterogeneous, near-boresight-aligned sensors. Search converges upon a precise 3D match between the CAD target model and range, IR and color imagery.

The feature prediction process, being based upon graphical rendering, is quite flexible and easily extended to include additional salient constraints. One such constraint is the sun angle, which is needed to predict what aspects of an objects internal structure will stand out in visible light imagery. In a prior paper [28], we described a system which uses the steps just described to precisely recover the pose of a known 3D object given near-boresight-aligned range, IR and color imagery.

This paper focuses on a significant extension for reasoning about scene occlusion and presents results where the system is used to distinguish between modeled objects and thereby perform target identification. Occlusion reasoning takes advantage of the fact that we are fusing range and electro optical (EO) data. Since range data indicates directly which features are being obscured by objects closer to the sensor, occlusion need not be treated as signal dropout. The feature prediction algorithm detects evidence of occlusion in range data and modifies the predicted set of features for range, IR and color sensors accordingly.

Of great importance is how well this CAD-based system is able to recognize targets. A series of experiments have been conducted on a test suite of 35 range, IR and color image triples containing four different military vehicles. From the study we have learned that the system is performing identification quite well, 77% correct, across the complete test set. Moreover, based on the conclusions of MIT Lincoln Laboratory [38], this level of performance is unlikely to be reached using only traditional ATR image-space representations (templates).

The paper is broken down into six distinct sections. Section 2 contains a literature review describing the works of other researchers as they pertain to our work. This is followed by a overview of the the goals of our current system. Sections 4 and 5 overview the system including the occlusion reasoning component. The next section presents results on our data set, and then conclusions and future work are discussed.

2 Relation to Prior Work

Our work draws upon the sensor fusion and CAD-based recognition literature. Most uses of CAD-based recognition focus upon a single sensor. For example, CAD models have been used for matching to 2D imagery [14, 35, 9], 3D range data [3, 4], as well as multispectral imagery such as IR [29] and SAR [11]. Typically these CAD systems rely on either the 3D or 2D geometry of the model to constrain the location and appearance of that object in the imagery [25, 19, 20].

With respect to sensor fusion, Aggarwal [1] nicely summarizes past work and notes that sensor fusion has tended to emphasize single modality sensors. There is comparatively little work focusing on different sensor modalities. He states that relating data from different modalities is more difficult, in part because of issues of sensor alignment and registration. While Aggarwal [26] and others [31, 21, 16, 36] have examples of successful mixed-modality fusion, this is still a young research area.

One solution to the problem of imperfectly aligned sensors is to use the CAD model geometry to suggest how image registration needs to be adjusted [7]. Our current system employs such constraints within a closed loop rendering system for dynamic feature prediction. Our use of rendering to perform feature prediction within a matching loop exemplifies an approach advocated by Besl [5] when he stressed the potential value of graphical rendering to support object verification.

Others have recognized the value of graphical rendering in support of object recognition. Wells has used graphics hardware for the computation of model points for use in tracking faces in video sequences [22]. Sato has used computer graphics to recover reflectance models of objects spinning on a turntable [30]. Others have used rendering for the generation of imagery for statistical modeling [17, 39].

Alternatively, we advocate an on-line prediction capability which performs the mapping from stored model to predicted features dynamically as part of the recognition process [28, 33]. A key to making this approach feasible is the development of algorithms which run many, if not all, computations in parallel on standard graphics acceleration hardware. This on-line capability supports a tight coupling between feature prediction and matching: modifying the features expected to be visible as matching progresses.

3 Multisensor Viewing Geometry: Defining Coregistration

Underlying our whole approach is a concept we call *coregistration.* When performing sensor fusion from multiple heterogeneous sensors, seldom if ever will the pixels from the different sensors be in a one-to-one correspondence. Knowledge of sensor parameters and relative sensor positions provides 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.

In our problem of fusing ground-looking range, IR and color sensor data, it is safe to assume that our sensors are approximately boresight-aligned. Put simply, they are likely to be positioned quite close to each other and to be looking in the same direction. We have prepared a detailed study of different sources of uncertainty in sensor-to-sensor alignment for near-boresight-aligned sensors [23]. A useful heuristic from this study is that over small rotations and restricted depth ranges, sensor-to-sensor rotation may be approximated with simpler co-planar translation. Together the sensors are free to rotate and translate relative to the object, but are constrained to permit only translation in a common image plane.

In the remainder of this paper, the term coregistration will be used to describe the process of simultaneously adjusting both the 3D pose of the sensor suite relative to a modeled object as well as the planar translation between sensors. For three sensors: range, color and IR, there are 10 coregistration parameters: 6 encode the pose of the sensor suite relative to the target and 4 encode the relative planar translation between pairs of sensors.

4 Cuing the Multisensor Target Identification System

While the focus of this paper is our multisensor identification system, in practice it must be used in conjunction with other algorithms which focus attention on a tractable number of possible target types and pose estimates. We have developed two upstream processes which together provide this cuing information. To provide context, these are briefly summarized.

4.1 Cuing Step 1: Color Detection

Targets are first detected using a new machine learning algorithm [12, 15] geared towards finding camouflaged targets in multi-spectral (RGB) images. The goal of this stage 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 regions of interest (ROIs) to the hypothesis generation stage. In essence, the target detection algorithm serves as a *focus of attention* mechanism that directs the system's resources only towards those parts of the image that contain potential targets. Typically over 95% of an image can be dismissed as not containing targets after this step.

4.2 Cuing Step 2: Hypothesis Generation

Color detection passes to the hypothesis generation algorithm a list of regions of interest (ROIs): boxes bounding possible targets. Each ROI is then analyzed and a list of most likely target types and poses is generated. A number of algorithms might fill this role, including perhaps geometric hashing techniques [2]. In our work, an existing boundary probing algorithm [10] developed by Alliant Techsystems has been adapted to this hypothesis generation task. The LARS suite uses a non-segmenting model-based approach, which efficiently exploits the 2-D (boundary matching) shape information contained in range signatures. Templates are derived from BRL/CAD models of the expected target set, thus no training imagery is required.

5 Multisensor Target Identification

The multisensor target identification stage fuses the imagery from all three sensors with the 3D CAD target model in order to make a final determination as to the type of the target. This section will first review the overall approach as presented in [28]. Next, the new extension to this system which enables it to reason about scene occlusion will be presented.

5.1 Interleaving Feature Prediction and Multisensor Target Matching

The search process developed for coregistration matching uses an iterative generate-and-test loop (Figure 1) 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 re-generated 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.

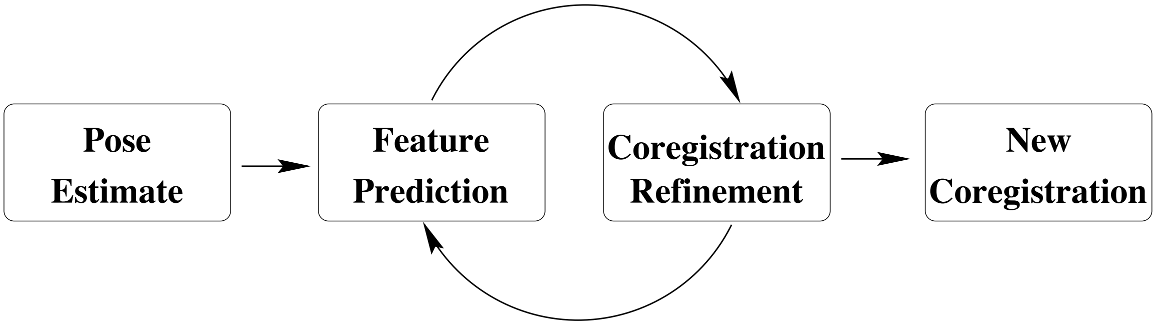


Figure 1: Interleaving Feature Prediction, Coregistration Refinement and Matching

5.1.1 On-line Model Feature Prediction

Highly detailed Constructive Solid Geometry (CSG) models of target vehicles are available in BRL-CAD format [37]. We have already developed algorithms to convert these models to a level of detail more appropriate for matching to the given sensor data [34, 32]. Another system, summarized here and fully described in [27], has been developed to extract edge and surface information from these models.

The feature prediction algorithm renders the vehicle using the current pose and lighting estimates to infer which 3D components of the target will generate detectable features in the specific scene. Each rendered 3D 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 3D 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 3D 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 3D 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.

5.1.2 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) encode the pose of the sensor suite relative to the target; 2 DOF encode 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 3D 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:

$$E_{\mathcal{M}}(\mathcal{F}) \; = \; \alpha_{\mathcal{C}} E_{\mathcal{M},\mathcal{C}}(\mathcal{F}) \; + \; \alpha_{\mathcal{I}} E_{\mathcal{M},\mathcal{I}}(\mathcal{F}) \; + \; \alpha_{\mathcal{R}} E_{\mathcal{M},\mathcal{R}}(\mathcal{F}) \tag{2}$$

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.

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

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 [6]. The fitness error values are summarized below (see [28] for a through discussion).

The optical fitness error represents the fidelity of match between the 3D edge features and the underlying image. The process of determining the error begins by projecting the predicted 3D 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 3D 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 2D coordinate system of the range image. A 3D enclosing box is then derived by back-projecting the rectangle into the 3D range sensor coordinate system. When seeking points to match to the 3D 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.

5.1.3 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 stepsize 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 [18]. 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.

5.2 Occlusion Reasoning

One of the main benefits of multisensor ATR is the ability to reason about model feature occlusion. Since 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 5.1.2. 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:

$$E_{\mathcal{M},\mathcal{R}}(\mathcal{F}) = \beta_{\mathcal{R}} E_{fit,\mathcal{R}}(\mathcal{F}) + (1 - \beta_{\mathcal{R}}) MAX(E_{om,\mathcal{R}}(\mathcal{F}), E_{oc,\mathcal{R}}(\mathcal{F})) \quad (4)$$

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

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

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, in order to send the error measure to zero.

Once the changes to the range error was 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 2 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.

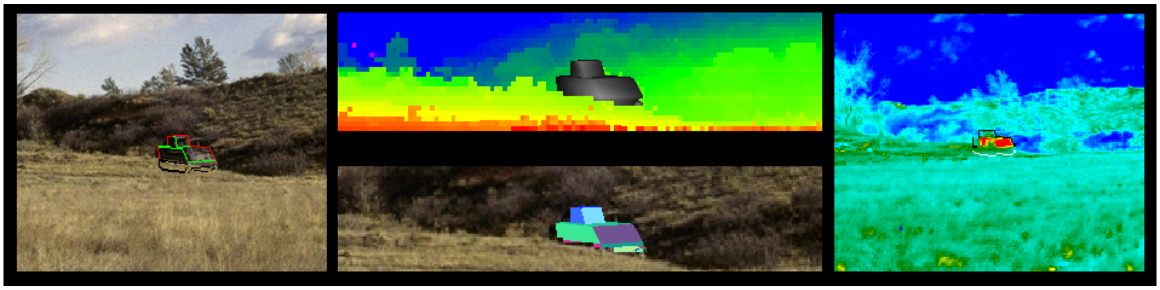


Figure 2: Shot34 Occlusion Example

6 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.

6.1 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 [8] 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 [23].

6.2 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 [24].

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 712 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.

6.3 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

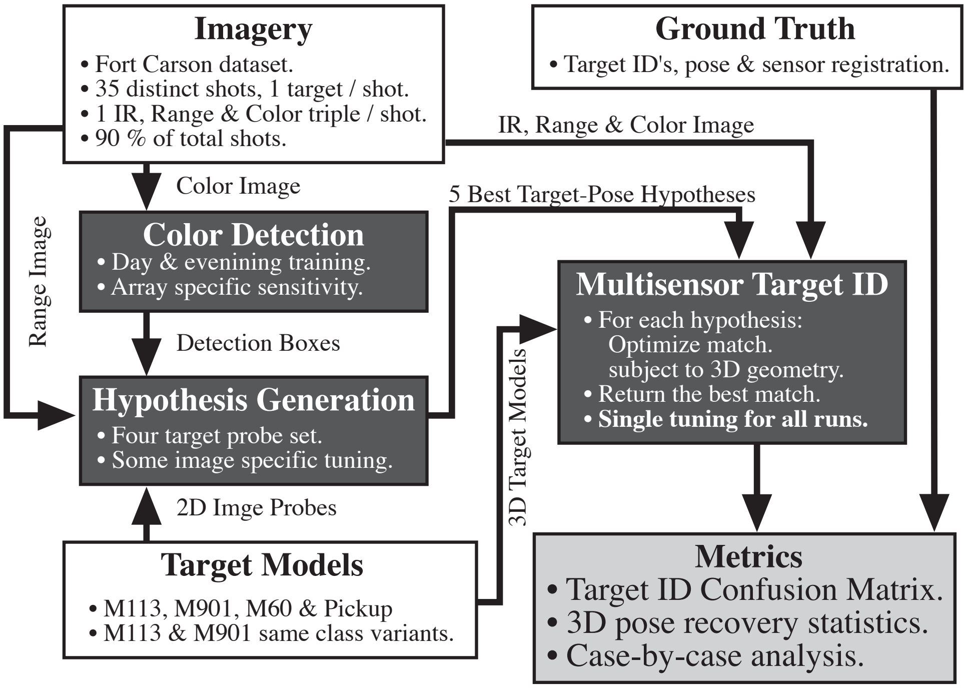

Imagery
• Fort Carson dataset.
• 35 distinct shots, 1 target / shot.
• 1 IR, Range & Color triple / shot.
• 90 % of total shots.
Ground Truth
• Target ID's, pose & sensor registration.
IR, Range & Color Image
Color Image
5 Best Target-Pose Hypotheses
Range Image
Color Detection
• Day & evenining training.
• 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
Hypothesis Generation
• Four target probe set.
• Some image specific tuning.
3D Target Models
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 3: Diagram of End-to-end ATR System Test.

targets present in these images are: M113, M901 (M113 with missile launcher), M60 and a pickup truck.

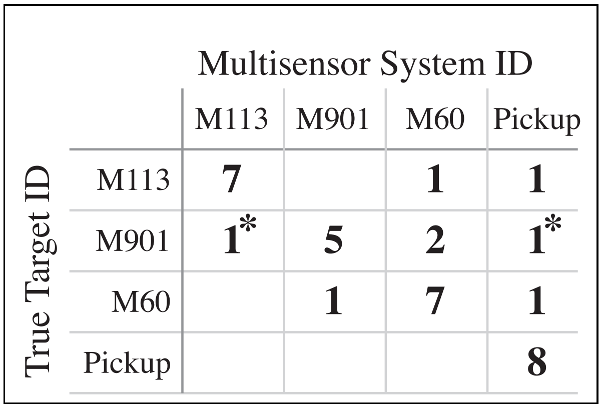
The overall design and flow of this experiment is summarized in Figure 3. The upstream detection and hypothesis generation algorithms were used to generate realistic input for the multisensor matching system. However, these upstream algorithms are not the focus of this particular experiment and they 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.

6.4 How Well are Targets Identified



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

Table 1: 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 2.

Table 1 presents a confusion matrix summarizing how well the multisensor identification system performed on the 35 test cases. A detailed case-by-case breakdown is presented in Table 2. The second column indicates the vehicle shot number and vehicle array as identified in the Fort Carson data collection report [8]. 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.

The next column represents the target ID returned by the multisensor matching system. A $\surd$ indicates the correct target has been identified. The fifth 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 2. One is that identification performs perfectly on the high resolution data from Array 5: #17 through #20. Another not surprising

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

Table 2: 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.

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 mis-identified. 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, and about 16% on occluded vehicles. However, it must be remember it has been hand tuned for each vehicle array.

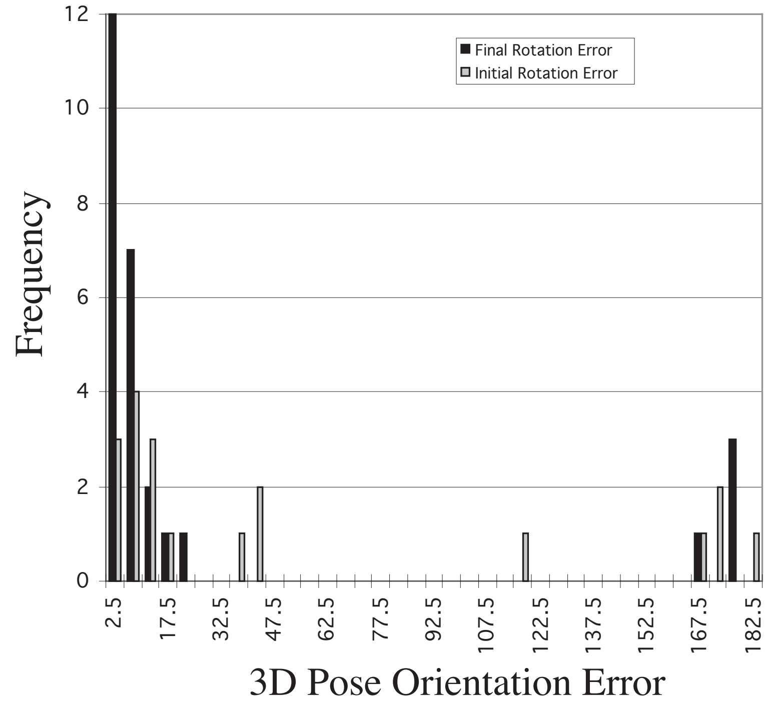
Unfortunately the table does not say anything about the pose of the best match found by either algorithm. Figure 4 shows the histogram comparison of the best result found for each image, when the system found the correct vehicle. In several cases, both systems placed the vehicle 180° from the true orientation. However, for the most part the multisensor algorithm was able to correctly estimate the pose of the vehicle.

To give another indication of how well the matching system is correcting inaccurate initial pose estimates coming from the hypothesis generation algorithm, Figure 5 shows the distribution of error in orientation relative to ground truth before and after matching. The histogram considers all inputs to multisensor matching with the correct target identified and an initial pose estimate within 90 degrees of true.

6.5 An Example Image

Figure 6 shows an example of a single image out of the set. For this image, the color detection algorithm successfully found the 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 match found by the multisensor matching algorithm. Recall that the best match is that which minimizes the match error defined in Section 5.1.2 and Section 5.2.

Figure 6a shows the initial starting hypothesis for the matching algorithm. Starting from the top left



Five degree wide histogram buckets, bucket center indicated.

Figure 4: Histogram of 3-D pose errors for the probing and multisensor identification algorithm. Frequency is the count of trials with less than or equal to indicated orientation error.

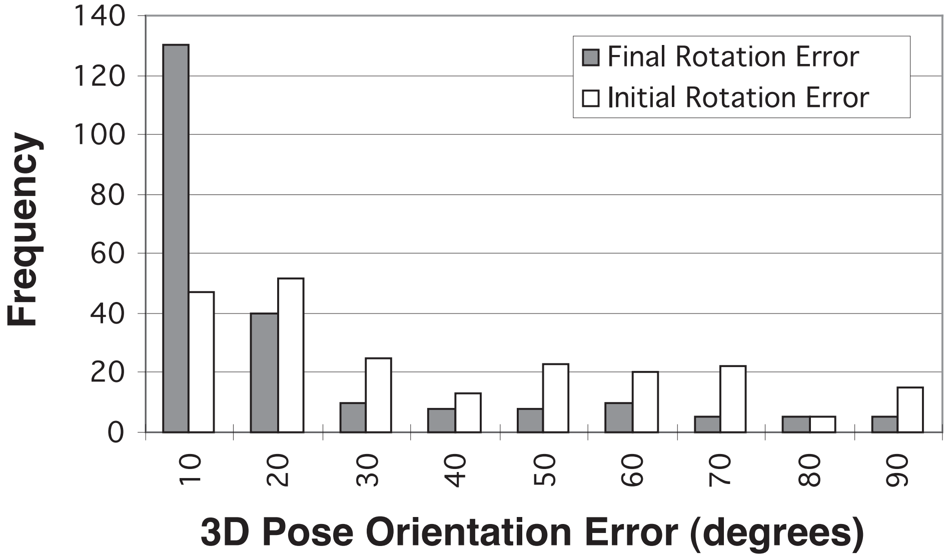


Figure 5: Histogram of 3-D pose errors before and after matching. Frequency is the count of trials with less than or equal to indicated orientation error.

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 boxes are predicted model points (red is non-omitted). The rightmost chip represents the range data with a IR texture map.

Figure 6b shows 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 to 90 seconds to converge from the initial to final estimates on Shot26.

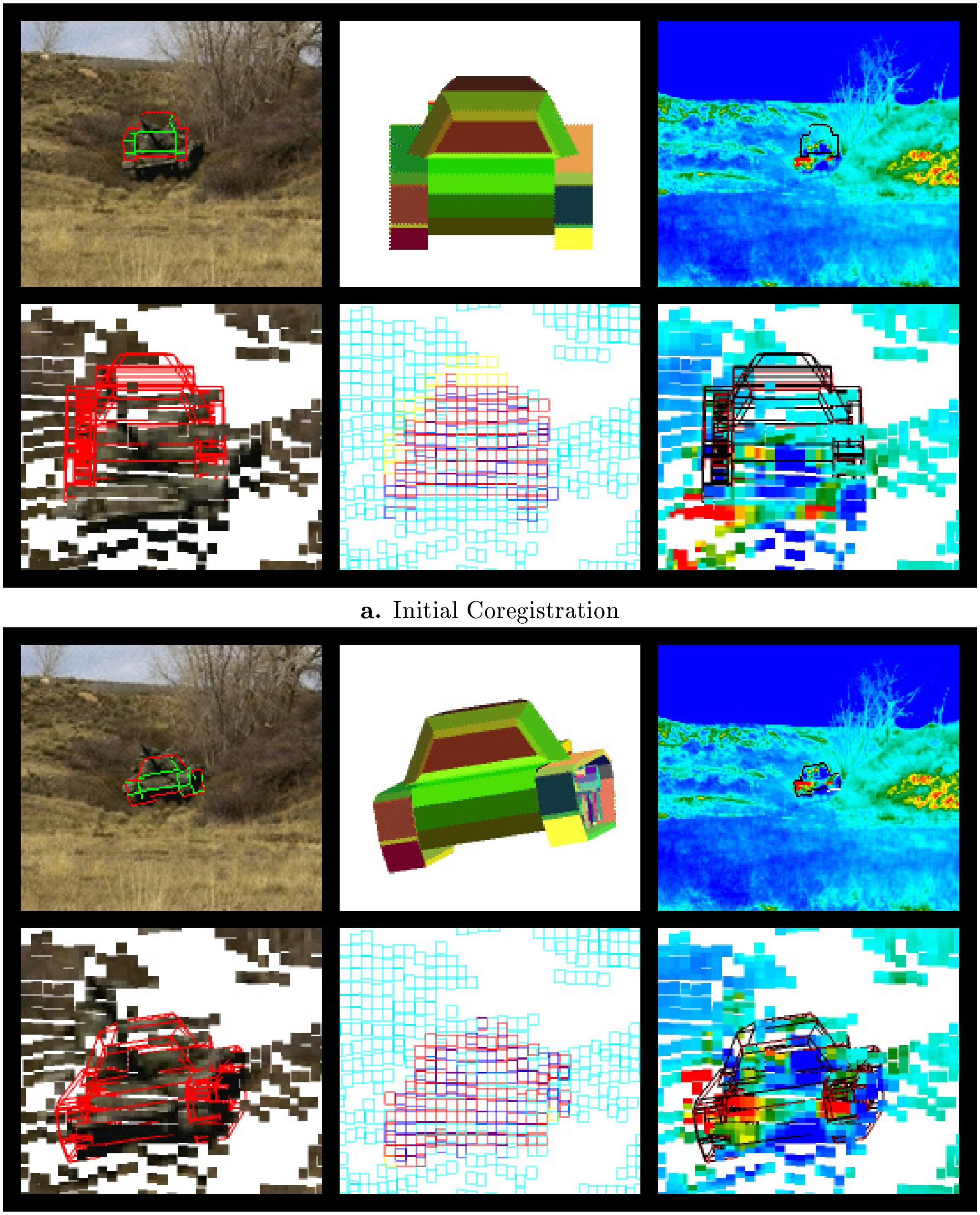
6.6 Removing a Bias for Small Targets

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, e.g. 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.

$$E'_{\mathcal{M},\mathcal{S}}(\mathcal{F}) = w_{\mathcal{S}}(s) E_{\mathcal{M},\mathcal{S}}(\mathcal{F}) \tag{6}$$

$$w_{\mathcal{S}}(s) = \frac{(\gamma_{\mathcal{S}} - 1)\, s + S_{min} - \gamma_{\mathcal{S}} S_{max}}{S_{min} - S_{max}} \tag{7}$$



a. Initial Coregistration

b. Refined Coregistration

Figure 6: Shot26 Multisensor Target Matching Results

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 correctly 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.

6.7 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 more 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.

7 Conclusions and Future Work

Whether the results presented above are considered full or meager is somewhat a matter of perspective. We have demonstrated that a CAD-based model approach to recognition can function well in a challenging object recognition domain: by all indications, these results are better than could be expected of a system exploiting only image-based representations. We have provided a strong argument for the value of on-line object feature prediction based upon the rendering of 3D models, and extensions which allow scene constraints such as occlusion to be worked into the feature generation process. We hope to see these same themes further explored in domains other than ATR.

Our test suite of 35 images is larger than that against which many algorithms are tested. However it is too small to allow us to characterize meaningful statistical dependencies between interesting dimensions of problem variability, such as occlusion, and target identification reliability. Future work will focus on obtaining more data to study these connections.

References

[1] J. K. Aggarwal. MultiSensor Fusion for Automatic Scene Interpretation. In Ramesh C. Jain and Anil K. Jain, editors, *Analysis and Interpretation of Range Images*, chapter 8. Springer-Verlag, 1990.

[2] Alexander Akerman and Ronald Patton and Walter Delashmit and Robert 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 Kaufman.

[3] F. Arman and J.K. Aggarwal. Cad-based vision: Object recognition in cluttered range images using recognition strategies. *Image Understanding*, 58:33–48, 1993.

[4] P. J. Besl and R. C. Jain. Invariant surface characteristics for 3D object recognition from range data. *cvgip*, 33:33 – 80, 1986.

[5] Paul J. Besl and Ramesh C. Jain. Three-dimensional object recognition. *ACM Computing Surveys*, 17(1):75 –145, March 1985.

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

[7] J. Ross Beveridge, Allen Hanson, and Durga Panda. Integrated color ccd, flir & ladar based object modeling and recognition. Technical report, Colorado State University and Alliant Techsystems and University of Massachusetts, April 1994.

[8] 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.

[9] J. Ross Beveridge and Edward M. Riseman. Optimal Geometric Model Matching Under Full 3D Perspective. In *Second CAD-Based Vision Workshop*, pages 54 – 63. IEEE Computer Society Press, February 1994. (Submitted to CVGIP-IU).

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

[11] Bir Bhanu and Grinnell Jones and Joon Ahn and Ming Li and June Yi. Recognition of Articulated Objects in SAR Imagery. In *Proceedings: Image Understanding Workshop*, pages 1237–1250, Los Altos, CA, February 1996. ARPA, Morgan Kaufman.

[12] C. E. Brodley and P. E. Utgoff. Goal-directed Classification Using Linear Machine Decision Trees. *Machine Learning*, page (to appear), 1994.

[13] Major Tom Burns. Moving and stationary target acquisition (mstar). In *Image Understanding Technology Programs*, Fort Belvoir, 1996. DARPA.

[14] Jin-Long Chen, George C. Stockman, and Kashi Rao. Recovering and tracking pose of curved 3d objects from 2d images. In *Proceedings Computer Vision and Pattern Recognition*, pages 233–239, June 1993.

[15] B. Draper, C. E. Brodley, and P. Utgoff. Goal-directed Classification Using Linear Machine Decision Trees. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(9):(to appear), September 1994.

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

[17] T.L. Gandhi and O.I. Camps. Robust feature selection for object recognition using uncertain 2d image data. *Computer Vision and Pattern Recognition*, pages 281–287, 1994.

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

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

[20] Daniel P. Huttenlocher and Shimon Ullman. Object Recognition Using Alignment. In *Proc. First International Conference on Computer Vision*, pages 102–111, London, England, June 1987. Computer Society Press of the IEEE.

[21] Alexander Akerman III, Ronald Patton, Walter H. Delashmit, and Robert Hummel. Multisensor fusion using FLIR and LADAR identification. Technical Report NRC-TR-94-052, Nichols Research Corporation, April 1994.

[22] William Wells III, Michael Halle, Ron Kikinis, and Paul Viola. Alignment and tracking using graphics hardware. In *Image Understanding Workshop*, pages 837–842. DARPA, 1996.

[23] J. Ross Beveridge and Mark R. Stevens and Zhongfei Zhang and Mike 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.

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

[25] David G. Lowe. Three-dimensional Object Recognition from Single Two-dimensional Images. *Artificial Intelligence*, 31, 1987.

[26] M. J. Magee, B. A. Boyter, C. H. Chien, and J. K. Aggarwal. Experiments in Intensity Guided Range Sensing Recognition of Three-Dimensional Objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7(6):629 – 637, November 1985.

[27] 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 Kaufman.

[28] Mark R. Stevens and J. Ross Beveridge. Precise Matching of 3-D Target Models to Multisensor Data. *IEEE Transactions on Image Processing*, page (to appear), January 1997.

[29] N. Nandhakumar and J. K. Aggarwal. Integrated analysis of thermal and visual images for scene interpretation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):469–481, July 1988.

[30] Yoichi Sato and Katsushi Ikeuchi. Refectance analysis for 3d computer graphics model generation. Technical Report CMU-CS-95-146, Carnegie Mellon University, June 1995.

[31] A. Stentz and Y. Goto. The CMU Navigational Architecture. In *Proceedings: Image Understanding Workshop*, pages 440–446, Los Angeles, CA, February 1987. ARPA, Morgan Kaufmann.

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

[33] Mark R. Stevens and J. Ross Beveridge. Interleaving 3d model feature prediction and matching to support multi-sensor object recognition. In *International Conference on Pattern Recognition*, volume 13, Austria, August 1996. Internation Association of Pattern Recognition.

[34] 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.

[35] G.D Sullivan, A.D. Worrall, and J.M Ferryman. Visual Object Recognition Using Deformable Models of Vehicles. In *Workshop on Context-Based Vision*, pages 75–86, june 1995.

[36] C.W. Tong, S.K. Rodgers, J.P. Mills, and M.K. Kabrinsky. Multisensor data fusion of laser radar and forward looking infared for target segmentation and enhancement. In R.G. Buser and F.B. Warren, editors, *Infared Sensors and Sensor Fusion.* SPIE, 1987.

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

[38] Jacques G. Verly, Dan E. Dudgeon, and Richard T. Lacoss. Model-Based Automatic Target Recognition System for the UGV/RSTA Ladar: Status at Demo C. In *Proceedings: Image Understanding Workshop*, pages 549–583. ARPA, February 1996.

[39] Mark D. Wheeler and Katsushi Ikeuchi. Sensor modeling, probabilistic hypothesis generation, and robust localization for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(3):252–265, 1995.