Computer Science
Technical Report

Colorado State University

Toward Target Verification Through 3-D Model-Based Sensor Fusion [*]

J. Ross Beveridge,
Mark R. Stevens
Anthony N. A. Schwickerath

February 2, 1996

Technical Report CS-96-105

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 work was sponsored by the Advanced Research Projects Agency (ARPA) under grants DAAH04-93-G-422 and DAAH04-95-1-0447, monitored by the U. S. Army Research Office.

Toward Target Verification Through 3-D Model-Based Sensor Fusion [*][†]

J. Ross Beveridge, Mark R. Stevens and Anthony N. A. Schwickerath
Colorado State University

February 2, 1996

Abstract

Most Automatic Target Recognition (ATR) algorithms operate in 2D image space. Even when using 3D models, these 3D models are typically translated off-line into sets of 2D representations, such as templates, which are then applied to imagery to perform detection, recognition and verification. In contrast to this approach, the work reported here takes steps toward direct matching of 3D models to range and optical imagery. The key idea is to exploit known 3D sensor and target geometry to drive a model-based sensor fusion process.

This process, which we call 'coregistration', resolves uncertainty in the 3D placement of the target relative to the sensors as well as uncertainty in the exact pixel registration between range and optical sensors. A specific coregistration algorithm is presented along with results on both synthetic and real data. Extending coregistration, two approaches are presented for finding locally optimal matches between 3D target models and multi-sensor data. Both are demonstrated on real data.

1 Introduction

Model-based object recognition work has long emphasized the importance of aligning 3D object models to features extracted from sensed imagery [BC82; Low91; GH91; HU90; HU88; BR92a; BR92b; Bev92a; Bev93; BR94; BHP94b; BR95]. While model-based approaches to Automatic Target Recognition have become much more common [DVD93; GJSL90], direct incorporation of alignment into the recognition process is rare [BJLP92]. This paper presents algorithms and results from a project at Colorado State University which is developing a new family of Target Verification algorithms based upon alignment.

Recognition problems in ATR typically involve complex objects viewed at great distances. Often they are partly obscured and backgrounds contain significant amounts of clutter. All these factors make typical ATR problems more difficult than those in which alignment to optical imagery alone has been demonstrated. To handle these more difficult problems, the work presented here

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

[†] Submitted to IEEE Transactions on Image Processing

exploits range as well as optical imagery. The direct measurement of range helps to resolve ambiguities which make robust recognition in optical imagery alone extremely difficult.

Working with two heterogeneous sensors raises many issues, including how to determine the registration mapping between them. In an ideal world, this mapping could be uniquely determined through off-line calibration. However, it is more prudent to assume that such estimates are accurate to within several pixels, but not more so. A system might try to resolve this ambiguity by only using low-level cues prior to applying model-based processing, but doing so would neglect valuable constraints implied by target geometry. Thus, we reach a key thesis of our work: **final refinement of the pixel-to-pixel registration between range and optical imagery should be an integral part of the model-based target recognition process.**

We use the term **coregistration** to describe the process in which sensor-to-target and sensor-to-sensor alignment parameters are simultaneously refined based upon constraints derived from known aspects of the global sensor and target geometry. Here, an algorithm for performing coregistration is presented. In addition, matching algorithms utilizing coregistration to establish globally consistent matches between 3D target and sensor features are presented and demonstrated on real data.

This more geometrically precise 3D framework for ATR promises more accurate match measures, and consequently, more robust target identification. Since the associated computational demands are considerable, less expensive detection [BDHR94] and hypothesis generation [Bev92b] algorithms are being used to provide focus of attention. Thus, it is assumed that the coregistration algorithms are being asked to rank and resolve quite specific hypotheses generated by up-stream processing. An example of such a hypothesis: 'there is an M113 at aspect 30 degrees and elevation 10 degrees at the following position in the scene'.

The scope of what we are undertaking is large, and only recently have the many pieces begun to fall together into a working, testable, system. Several key components had to be developed simultaneously before we could even begin testing on real data.

Dynamic Model Feature Generation: From stored 3D target models, derive for each hypothesized target instance the 3D features appropriate for matching to both range and optical imagery.

Model-Driven Feature Detection: Extract locally optimal linear edge features from highly cluttered, low resolution, optical imagery of camouflaged vehicles.

Coregistration: Refine the 3D pose (position and orientation) of a target model relative to both range and optical sensors while simultaneously refining the image-to-image registration between these sensors.

Correspondence-Space Search: Use local search in the space of discrete correspondence mappings between model and sensor features to find near optimal matches between 3D target features and sensor image features.

Coregistration-Space Search: Use local search in the space of coregistration parameters to find near optimal matches between 3D target features and sensor image features.

Experimentation with the complete system is in its early stages and has primarily focused upon careful study of test cases. In this paper, results for two target models and pairs of range and optical images are presented. While it is still too early to have statistical results over a large set of cases, these early results are suggestive of what these new 3D techniques can accomplish.

2 Motivation & Background

Our general, or conceptual, motivation derives from the belief that model-based algorithms which test for global consistency using alignment should and can be extended to multi-sensor domains. The more immediate motivation, and the one that has shaped the particulars of our problem formulation, are derived from the need to develop robust target verification algorithms for a sensor suite possessing separate LADAR, color and IR sensors.

2.1 Conceptual Motivation

A long tradition of work on object recognition has emphasized finding matches between object and image features for which there is a single globally consistent alignment of features. For example, Lowe utilized this approach subject to orthographic projection [LB85] and later demonstrated object tracking under perspective projection [Low91]. Huttenlocher [HU90] demonstrated alignment-based recognition under orthographic projection. Grimson's work [Gri90] on constraint base matching has emphasized local feature compatibility to prune tree search and thus minimize tests of global alignment.

Our past work [BWR90; BWR91; Bev93; BR95] emphasized global alignment as a basis for match ranking and optimal matching to CCD sensor data. Local search through the match space constructs a globally consistent match by finding a sequence of successively better match hypotheses until one which is locally optimal is found. For a given match hypothesis, the fit error is based upon alignment using Kumar's [Kum89; KH94] pose algorithm. Random trials of steepest-descent local search on CCD matching problems demonstrate that the technique finds, with high probability, matches which are both globally optimal and globally consistent [Bev93; BR95].

To extend this type of globally consistent matching for multisensor recognition, a global alignment or coregistration algorithm is needed. As an initial step, we investigate a combination of proven single sensor error formulations. Kumar [Kum89; KH94] has developed a succession of single sensor algorithms for a typical 'pin-hole' camera model which takes perspective into account. Section 6.1.1 gives a more complete description of Kumar's formulation and explains our decision to base our measure on Kumar's earlier, rather than later, error formulation. Horn [Hor86] details a technique for aligning two sets of 3D points. Each of these is a least-squares algorithm based upon known correspondence mappings between features. This paper describes a joining of these two least-squares problems into a single coupled least-squares optimization problem.

Others have worked on problems similar to the coregistration problem discussed in this paper. Herbert [HKK90] presents least-squares mechanism for computing the rigid transform between range and color CCD sensors based upon corresponding image points in the two sensor images. However, this work does not explicitly recover the associated pose of the 3D points relative to the sensors. Both Eason [EG92] and Hel-Or [HOW93] developed least-squares multisensor pose algorithms. However, these algorithms only solve for the six-degrees-of-freedom pose estimate.

They do not support simultaneous adjustment of the sensor registration parameters. In terms of constraints, all of these methods assume a known sensor-to-sensor registration.

More recent work by Hel-Or and Werman [Y. 94; Ho93] adds degrees of freedom to account for articulated objects and nicely handles variable constraints in a single extended Kalman filter formulation. Their general Kalman filter approach could be applied to the coregistration problem formulated here. However, for reasons of efficiency we have intentionally used what Hel-Or and Werman call the *parametric* approach. Since our coregistration algorithm is being developed to be run within the inner loop of an iterative local search matching procedure, efficiency takes precedence over generality. In future work, we will test the relative merits of our approach compared to the more general formulation of Hel-Or and Werman as well as more traditional photogrammetric approaches.

Aggarwal [Agg90] summarizes past sensor fusion work and makes two points particularly relevant to this paper. Aggarwal notes that past work on sensor fusion emphasized single modality sensors, with comparatively little work on different sensor modalities. The implied explanation is that relating data from different modalities is more difficult. While Aggarwal [MBCA85] and others [SG87] have examples of successful mixed-modality fusion, this is still a young research area.

Aggarwal also notes that to properly perform mixed-modality sensor fusion, **coordinate transformations between images need to be adaptively determined**. Coregistration, as presented in this paper, is a working example of precisely the kind of tool needed to support such an adaptive process. We believe that knowledge of sensor and object geometry should be used to constrain and adapt both the object pose and sensor-to-sensor coordinate transformations as part of object recognition.

2.2 Practical Motivation

While we expect coregistration to be important in many domains, the problem arises for us in the context of the RSTA (Reconnaissance, Surveillance and Target Acquisition) function of the UGV (Unmanned Ground Vehicle) program. In this application, a FLIR sensor for acquiring IR data, a color CCD sensor, and a LADAR sensor are to be mounted on a common pan-tilt head riding atop a HMMWV. The FLIR and color sensors have been fielded, but the LADAR sensor is still under construction. To fill in while no LADAR sensor is on the vehicle, we went to Fort Carson with Lockheed Martin in the Fall of 1993 and collected a large set of range, color and IR imagery [BPY94], and it is this data which is considered here. The overall goal is to utilize these three sensor modalities to improve the reliability of target identification algorithms.

Most of the prior work on target identification uses a fixed set of image-based templates or probes [Bev92b; VDL95; DVD93]. These techniques have their place, and we are using the Alliant Techsystems probing algorithms [Bev92b] to provide initial object-and-pose hypotheses. However, as the final step in verification, unless precise 3D coregistration is performed[1], it is neither possible to perform final cross-sensor model alignment nor is it possible to exploit precise 3D geometric relationships. Thus, the coregistration work presented here is our first step in developing a new generation of recognition algorithms which go beyond matching in single-sensor image spaces and instead geometrically couple 3D object models and sensors.

[1] We assume here applications in which precise pixel registration is unknown

3 A Quick Look at Some Real Data.

In this paper, we will be presenting our algorithms' performance on real ATR images. These images, as is common in the ATR domain, are difficult. The background consists of high texture clutter, such as grass and shrubs, which can partially occlude the target. Also, the goal of the vehicle designers is to paint them with similar colors and textures, inducing additional clutter. In addition, the objects being recognized are generally at a significant distance, and hence do not cover much of the image. This results in lower quality and less data.

From the set of hundreds of images collected at Fort Carson [BPY94], we selected three pairs of range and color. These were selected to represent some different qualities of the ATR domain. Shots 18 and 20 represent relatively close examples (approximately 50 meters) of two different vehicles, the M60 and M113, respectively. Shot 31 represents the M113 at about 160 meters. This image is significantly more difficult, due both to the small number of pixels on target and to the angle at which the vehicle is viewed, nearly head on.

Figures 1, 2 and 3, contain color and LADAR imagery for the two types of vehicles. The LADAR imagery is drawn as a set of 3D polygons at range sample depth. The visualization algorithm allows the 3D data to be rendered interactively from a different viewpoints, and provides a better feel for the characteristics of the data than the typical 2D range intensity image [GBSF94; GBSF95].

4 Deriving Target Model and Sensor Features

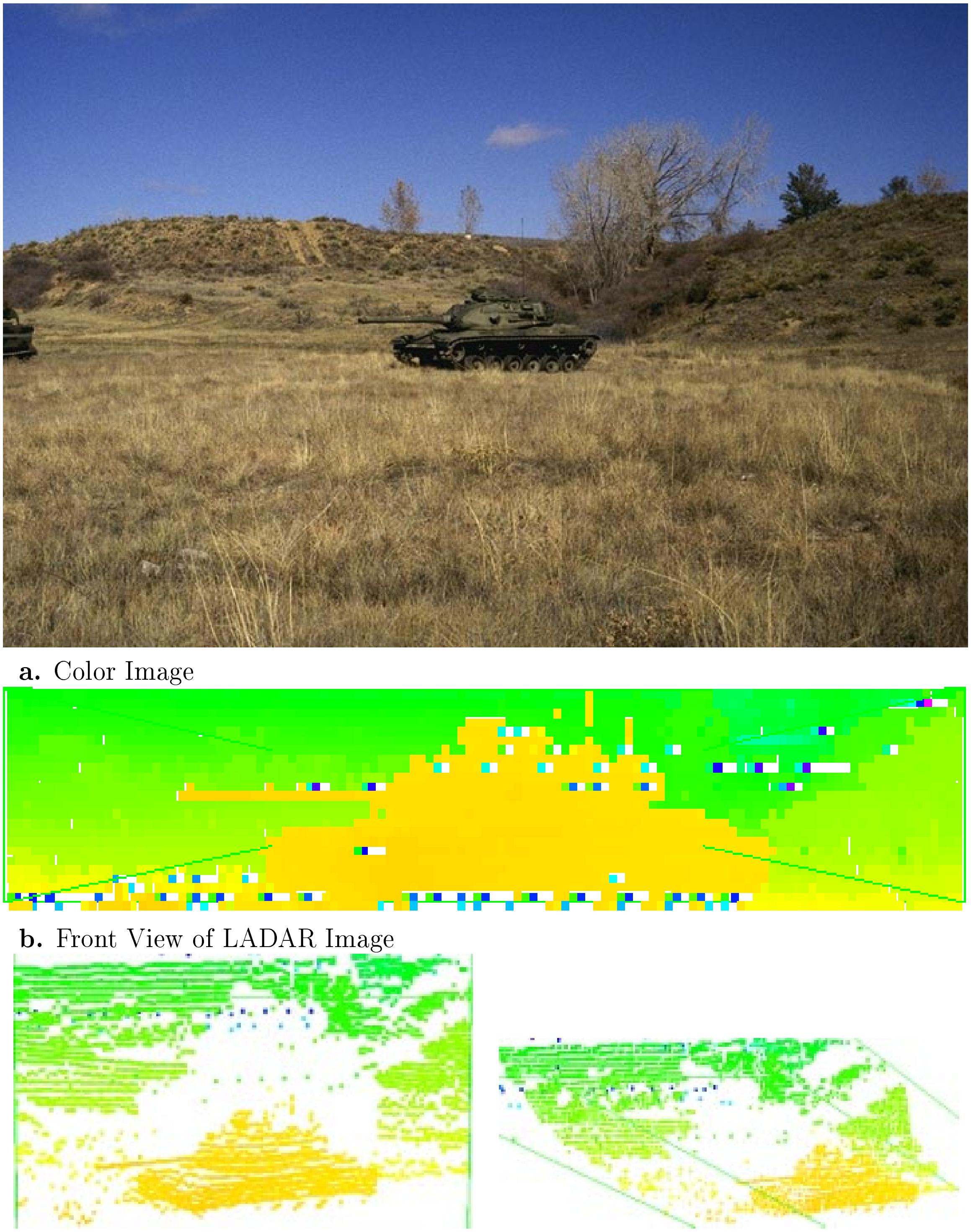
4.1 Dynamic Model Feature Generation

Highly detailed models of the vehicles in our Fort Carson dataset exist in the CAD model format known as BRL/CAD [U. 91]. Algorithms to reduce the model complexity to a level more closely related to the sensor granularity have already been developed [SBG95; Ste95]. From these simpler models, features to be used in the matching process are then obtained.

Common approaches to model feature generation have centered around an off-line model analysis in which visible features are determined for all viewpoints[PD87; Pla88]. The results are then grouped into regions of constant topology [KD87] and stored in an aspect graph representation [KvD76; KvD79]. The aspect graph is used at runtime to obtain the list of visible features for a given pose [SD92].

As an alternative to off-line compilation of the stored graph representation, our approach instead exploits the power of modern graphics hardware to achieve real-time generation of relevant model features [Ste95]. Two sets of model features are generated based upon an estimate of the target's position and orientation. The first set represents the model silhouette lines appropriate for matching to optical imagery. The second set represents the 3D sampled surface information for matching to the range data.

Not only is the on-line generation of 3D silhouette features novel in an ATR algorithm, it is also essential to the success of coregistration. Were silhouettes to be reduced to flat 2D templates, the ability to make subtle adjustments in appearance associated with small rotations of the target would be lost. The mapping from the target model to features appropriate for matching to range imagery is more obvious and less original. A simulation of the range sensor geometry is used to



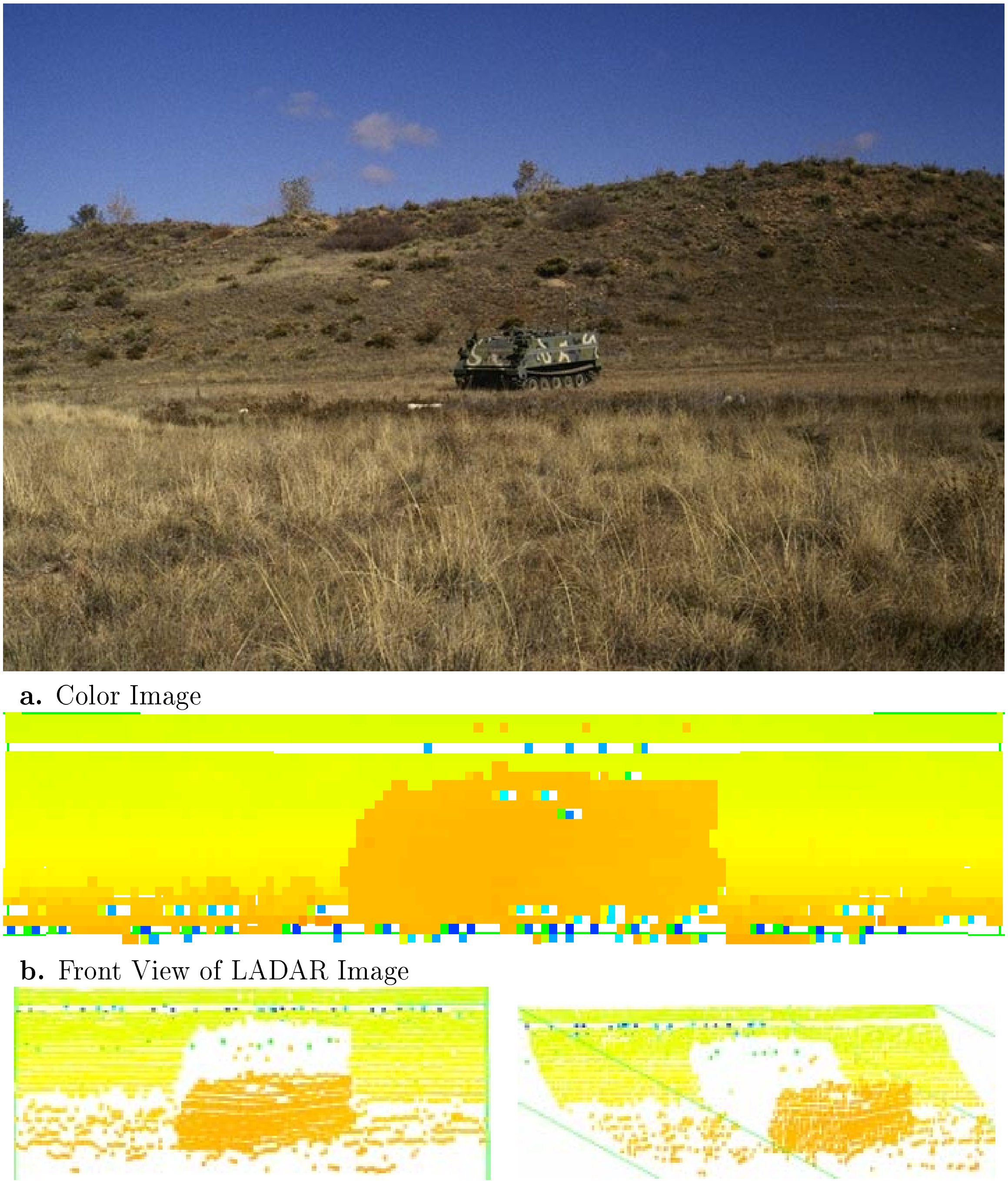
a. Color Image

b. Front View of LADAR Image

c. LADAR from another angle

d. LADAR from another angle

Figure 1: Data set for Shot 18, Array 5 (M60)



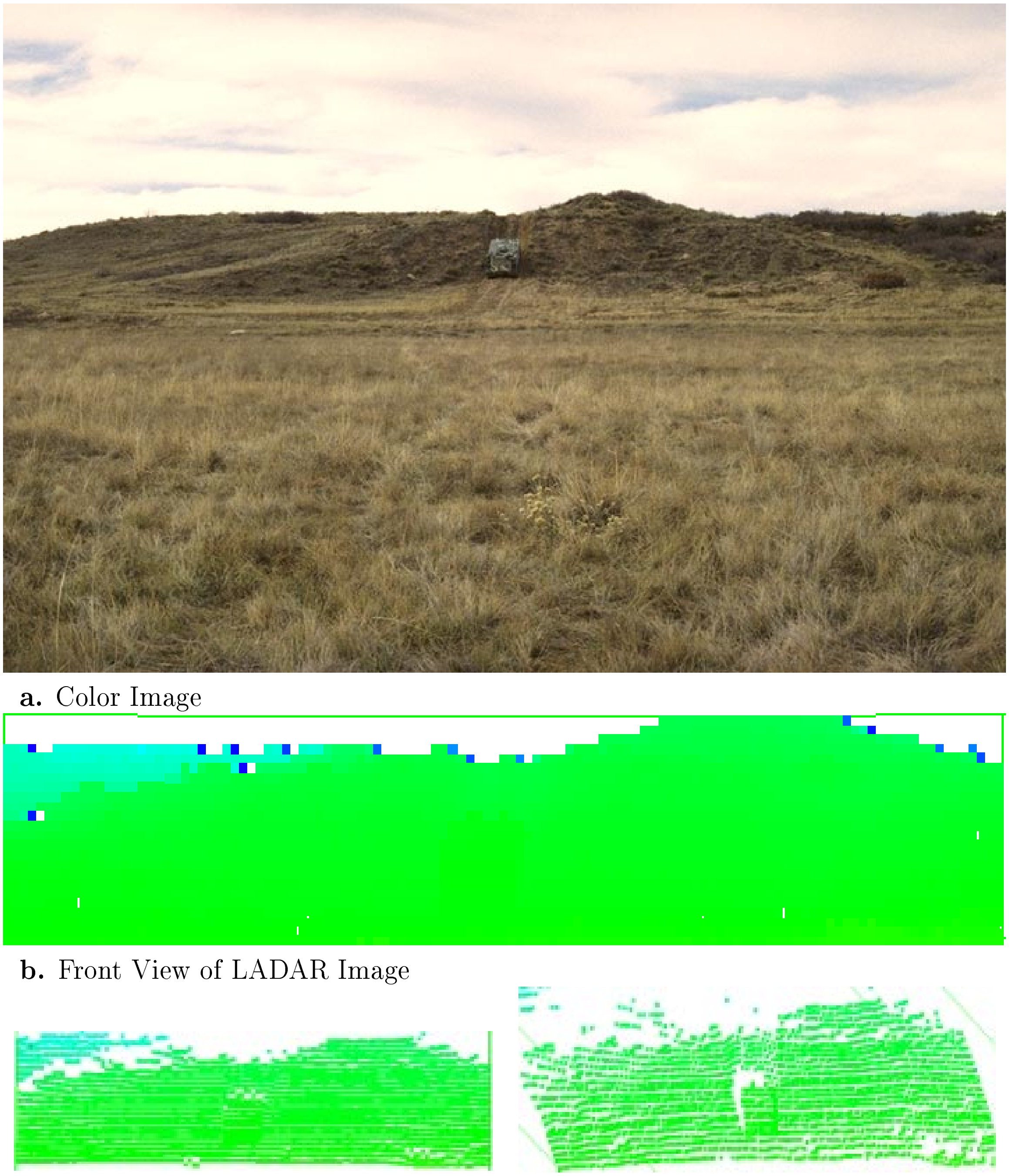
a. Color Image

b. Front View of LADAR Image

c. LADAR from another angle

d. LADAR from another angle

Figure 2: Data set for Shot 20, Array 5 (M113)



a. Color Image

b. Front View of LADAR Image

c. LADAR from another angle

d. LADAR from another angle

Figure 3: Data set for Shot 31, Array 8 (M113)

generate a sampled surface. The sampling parameters are chosen so that aspect and resolution reflects the actual data.

4.1.1 Generation of 3D Silhouette Edges

To determine what parts of the target model produce the silhouette, a unique color first is assigned to each face in the target model. This color acts as an index into a hash table of 3D faces. The model is then rendered from the hypothesized viewing angle. Rendering is performed on a hardware Z-buffer, and hence is very quick. Running on a Sparc 10 with a ZX accelerator, this process takes roughly 1.2 seconds for a target model containing 250 faces. The colors of resulting pixels indicate which faces are visible. Pixels adjacent to the background color, which is also unique, contribute to the target silhouette. Thus, if the background color appears in a pixel's eight-connected neighborhood, the associated face lies on the silhouette.

Further search determines which specific face boundaries (edges) generate the silhouette. An edge is a possible silhouette edge if only one of the two bounding faces is visible [SD92]. This step may leave some edges which are actually internal as hypothesized silhouette edges. It also does not deal with self-occlusion. A clipping algorithm is then used to discover and discard those edges, and portions of edges, which are not part of the silhouette. Because an orthographic projection is used to render the model, parametric end-point values measured in the rendered image during clipping may be applied directly to the corresponding 3D edges.

The final result is a list of 3D edges representing the silhouette of the target model for a given viewing angle. Using a user defined minimum line distance threshold, shorter lines can be discarded. Figure 4a shows a sample color image, and Figure 4b shows the silhouette obtained using this method.

4.1.2 Generation of Range Sampled Surfaces

A 3D sampled surface is generated in a manner which simulates, in simple terms, 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 position and orientation. Based on the characteristics of the range device, rays are cast into the scene and intersected with the 3D faces of the target model. The results of the rendering step used to extract the silhouette are used here to limit ray intersections to only those faces known to be visible. The closest face intersection is stored as the depth the current position. By design, noise factors are neglected when generating model features: the intention is to generate a high quality model. Noise is dealt with later when matching the model features to the sensor features.

4.2 Model-Directed Linear Feature Detection

Traditional methods for locating objects in optical imagery typically use edge detection algorithms. [MH80; Hil83] locate local edges which may then be grouped into larger features such as straight line segments [BHR86; LB83]. These linear features are, in turn, matched to linear features of stored object models [Low85; Low91; HU87; HU88; HU90; GH91; BWR89; BWR90; BWR91; BR92a; BR92b; Bev92a; Bev93; BR94; BHP94b; BR95]. These bottom-up feature extraction algorithms are prone to error [Cla89; BGK$^+$89], and often produce extraneous line segments, fragmented segments, and sometimes over-grouped segments. Our past work in other problem domains [Bev93] demonstrated local search, coupled with sound and efficient tests of global alignment, could overcome significant amounts of fragmentation, over-grouping and clutter. However, the features produced by the Burns algorithm [BHR86] on the Fort Carson imagery are of such poor quality that we have adopted a top-down rather than a bottom-up approach in the work developed here. This poor feature quality is a byproduct of the high texture content of the background and the vehicle camouflage, as well as the color similarity between the target and the background.

To realize a robust, model-driven, edge detection capability, two ideas from the literature have been combined: model-driven edge detection [FL87; FL88] and directionally tuned gradient filters [Can86]. The quality of a straight line segment denoting an extended edge is defined to be a function of the gradient magnitude under that edge. A gradient mask tuned to the specific expected orientation of the segment is used. The placement of the segment is perturbed until a locally optimal placement is found.

4.2.1 Placing Silhouette Edges in the Image

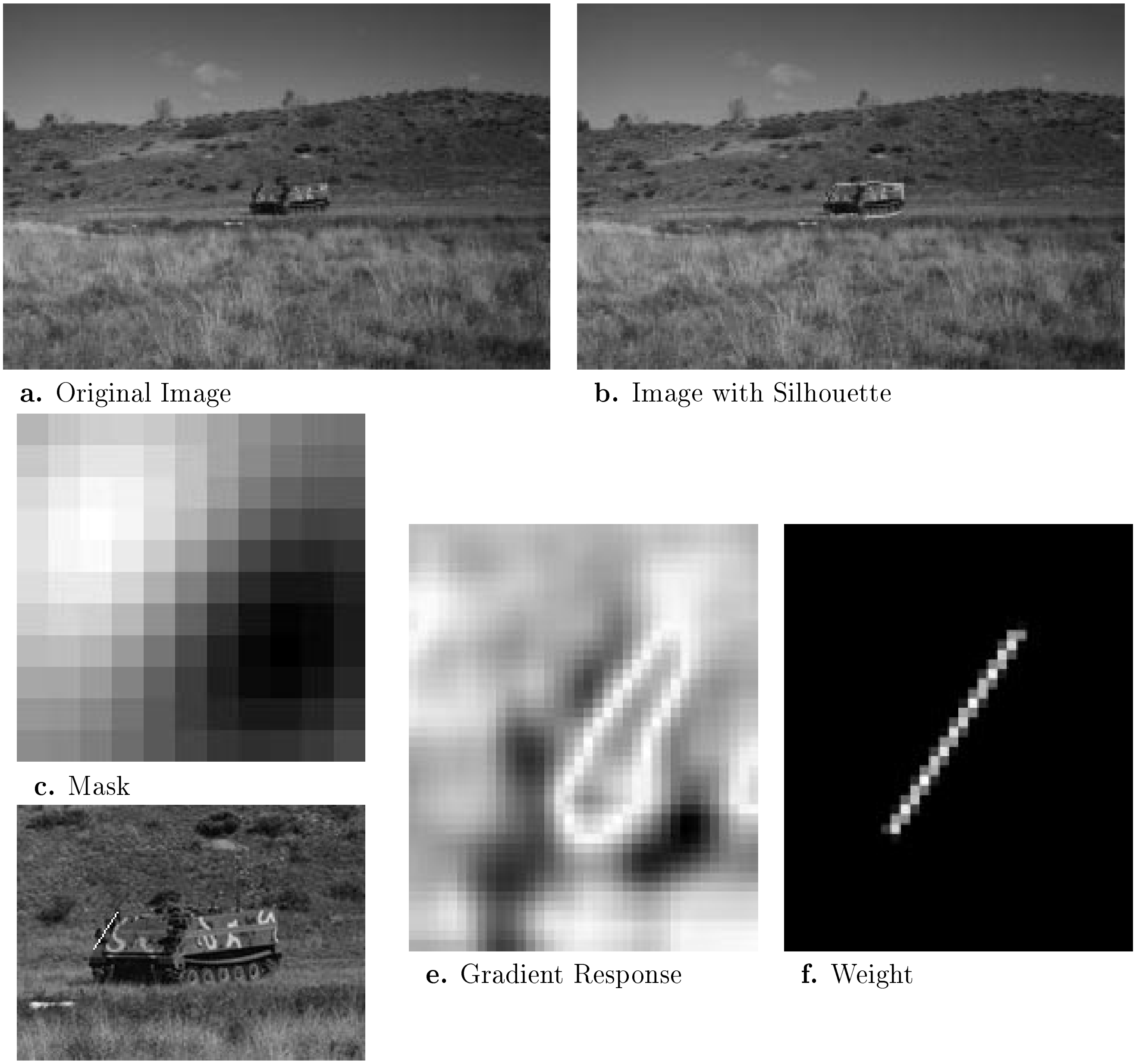
This model-driven approach is initialized by projecting the 3D silhouette edges into the color image. Projection is possible because both the intrinsic sensor parameters and the approximate pose of the target are known. The parameters for the color sensor have been determined off-line using calibration targets [BHP94a]. Thus, for each silhouette feature generated from the target model, a search is initiated in the image for the best corresponding line segment.

The first step in this search is to construct a gradient mask tuned to the particular expected orientation of each silhouette edge. This mask is formed by rotating the first derivative of a Gaussian to match the orientation of the current silhouette edge. There are many precedents for tuned edge masks including Canny [Can86] and Torres [TP86]. Others to develop and use such masks for bottom-up edge detection include [Shu94; FA91]. An example of such a filter, displayed as an image, is shown in Figure 4c.

The second step is to measure the overall gradient response under the segment. A commonly used graphics anti-aliasing technique known as Pineda Arithmetic [Pin88], is used to determine with subpixel accuracy where the projected line crosses the sensor image. A weighting term is created (see Figure 4f) to scale the gradient at each pixel, and obtain the gradient for a single line. The gradient response under the line segment, $\hat{G}$, is normalized to lie in the range $[0, 1]$:

$$\hat{G}_{Line}(i) = \frac{\sum_{i=Line_{xa}}^{Line_{xb}} \sum_{j=Line_{ya}}^{Line_{yb}} |Gradient(i,j)| \cdot w(i,j)}{MaxResponse \cdot \sum_{i=Line_{xa}}^{Line_{xb}} \sum_{j=Line_{ya}}^{Line_{yb}} \cdot w(i,j)} \tag{1}$$

$Gradient(i,j)$ is the gradient magnitude measured using the directionally tuned mask at pixel (i,j), $w(i,j)$ is a weighting term proportional to the distance of the pixel from the true line, and $MaxResponse$ is the largest expected gradient in the image. The summations are defined over an extents box about the current position of the line segment, and $w(i,j)$ is 0 for pixels lying outside the radius of the line segment.



a. Original Image

b. Image with Silhouette

c. Mask

d. Silhouette Line

e. Gradient Response

f. Weight

Figure 4: Optical Data Error Function

To find the best placement for each line segment, the segment parameters are perturbed until a locally optimal placement is found. Perturbations consist of small translations and rotations of the line about its current position. Once a local optimum is reached, the perturbed edge is returned as the best segment in the image for the given model line.

5 Two Ways to Find Matches: An Overview

The previous section has provided a means to derive corresponding features from a target model and sensor data. This section outlines, in general terms, two quite different ways to go about determining whether there exists a good match between these features.

Consider an error function which evaluates the quality of a match between a target model and sensor data:

$$E_{match}(c, \mathcal{F}) > 0 \quad \forall c \in C,\ \mathcal{F} \in \Re^8 \tag{2}$$

The first argument to this function, c, represents a particular correspondence mapping between model and sensor features. The second argument, $\mathcal{F}$, represents the coregistration of the sensors relative to the target model.

In the most general case, the correspondence space C is the power set of all possible pairs of sensor and model features. Observe that C denotes correspondences between the target model and features derived from both range and optical imagery. Thus, c indicates the pairing between corresponding sampled surface features on the target model and sensed range points in the LADAR imagery. The mapping c also indicates corresponding features for the optical imagery.

The coregistration $\mathcal{F}$ represents the geometric relationship between the sensors and the target model. In the development below, this is an eight place vector: six values encode the pose of the target relative to the optical sensor (3 rotation and 3 translation), and two encode the planar translation of the optical image plane relative to the range sensor's image plane. The rationale for this choice is presented below.

Matching seeks to find the correspondence c^* and coregistration $\mathcal{F}^*$ which together minimize the match error:

$$E_{match}(c^*, \mathcal{F}^*) \ \leq\ E_{match}(c, \mathcal{F}) \quad \forall\, c \in C,\ \mathcal{F} \in \Re^8 \tag{3}$$

Since c and $\mathcal{F}$ are clearly related, it would seem that a choice of one leads to an obvious choice for the other. This intuition suggests two alternative approaches to matching.

Consider first the case in which one selects a specific set of corresponding features c. A fitting procedure can be imagined which determines an $\mathcal{F}^*$ such that:

$$E_{match}(c) \equiv E_{match}(c, \mathcal{F}^*) \ \leq\ E_{match}(c, \mathcal{F}) \quad \forall\, \mathcal{F} \in \Re^8 \tag{4}$$

In this formulation, correspondence c serves as an independent variable and $\mathcal{F}$ a dependent variable. Finding the best match may now be conceptualized as the process of searching C for the c^* which minimizes equation 4. As a shorthand, this approach will be called **correspondence-space search**. The coregistration algorithm developed below is precisely the type of fitting procedure required to perform correspondence-space search.

The converse may also be done. Select a specific coregistration $\mathcal{F}$, and use a local assignment procedure to determine a c^* such that:

$$E_{match}(\mathcal{F}) \equiv E_{match}(c^*, \mathcal{F}) \ \leq\ E_{match}(c, \mathcal{F}) \quad \forall\, c \in C \tag{5}$$

Now $\mathcal{F}$ is the independent variable and c the dependent one. Finding the best match may be framed as the problem of finding the coregistration $\mathcal{F}^*$ which minimizes equation 5. The term **coregistration-space search** will be used as a shorthand for this approach.

6 Coregistration & Correspondence-Space Search

In the specific formulation developed here, a coplanarity constraint limits the freedom of movement of the range sensor relative to the optical sensor. Thus, the range reference coordinate system may translate in the common x-y image plane of the two sensors, but otherwise the two sensors move together. This may, at first, seem an odd choice of constraint. Given two sensors on a common platform, it is their relative pointing angles, not their relative spacing, which is most likely to vary. Why should we use translation to express variability introduced by small rotations? The answer is that the pixel-to-pixel movement between the two image planes is virtually indistinguishable in the two cases when rotations are small. The advantage of the translation formulation is that it does *not* introduce a second rotation term into the coregistration formulation, which would in turn add unnecessary nonlinearities.

6.1 Coregistration

The explicit coregistration error term, its partial derivatives, and the resulting update equations for the coregistration algorithm are presented in Section 6.1.1. The range-to-model error is simply the squared Euclidean distance between corresponding features. The optical-to-model error is measured using a squared error between planes defined by image line segments and points on the 3D object model.

This measure of optical-to-model error is the first measure developed by Kumar [Kum89; KH94]. Kumar subsequently showed this measure to be sensitive to fragmentation noise in the image segments. To correct this weakness, Kumar subsequently developed a measure using 3D model features to define planes. We earlier observed an analogous result for 2D line fitting [BWR89]. While Kumar's later formulation is better when fragmentation occurs, we avoided it for two reasons. First, to use the more recent measure requires an additional renormalization of terms after each iteration. Second, our coregistration process is using a top-down approach to find segments in the optical imagery which will not induce fragmentation. Were fragmentation of optical segments to be an issue, investigation of Kumar's latter measure would be appropriate.

6.1.1 Deriving the Coregistration Update Equations

In our coregistration fit error, we consider a set of image data with independent Gaussian noise applied to the position information of each defining point. That is, point data in the range image is normally distributed about its true position and the endpoints of optical lines are individually perturbed. By assuming independent noise processes, we can define our fit error as the sum

$$E_{fit}(\mathcal{F}) = \alpha_{fit} E_{fit,o}(c,\mathcal{F}) + (1-\alpha_{fit}) E_{fit,r}(c,\mathcal{F}) \tag{6}$$

The constituent parts of E_{fit} are illustrated in Figure 5. 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 [Kum89; KH94] for computing camera-to-model pose. The second, $E_{fit,r}$, is simply the sum-of-squared Euclidean distances between corresponding model and range points. The coregistration parameters themselves are denoted by $\mathcal{F}$.

While the fit errors are dependent upon both the correspondence, c, and the coregistration parameters, $\mathcal{F}$, here we will be considering the case where c is fixed and an optimal $\mathcal{F}$ is being sought. This corresponds to the case in equation 4. For a more compact notation in this section, we will just write E_{fit}, leaving the fixed c and variable $\mathcal{F}$ implied.

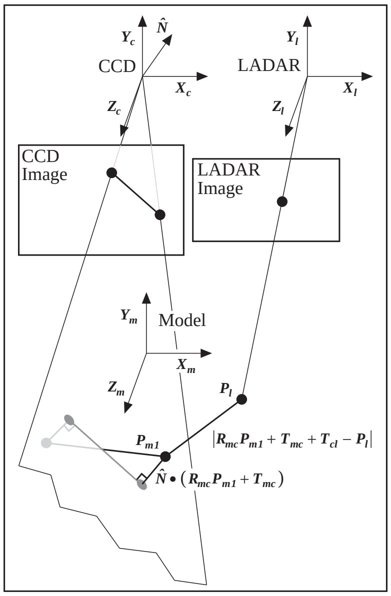


Figure 5: Illustrating distance errors which define optimal coregistration.

We use the weighting term $0 \leq \alpha_{fit} \leq 1$ to control the relative importance of the optical and the range data. Since each of the error terms, $E_{fit,o}$ and $E_{fit,r}$, generally falls in the range $[0, 1]$, we assume that E_{fit} also falls in this range. This normalization help us compare data from two separate sources.

The model-to-optical fit error, $E_{fit,o}$, measures the distance between endpoints of 3D line segments on the object model and 3D planes defined by corresponding line segments found in the optical image. These planes are defined by three points: the two endpoints of the optical line segment and the focal point of the optical sensor. If we have n_o model-to-optical correspondences, we write the model-to-optical fit error as:

$$E_{fit,o} = \frac{1}{2n_o\tau_{mo}^2 \sum_i \lambda_{oi}} \sum_{i=1}^{n_o} \sum_{j=1}^{2} \lambda_{oi} \left(\hat{N}_{oi} \cdot \left(R_{mo}\vec{P}_{mij} + \vec{T}_{mo}\right)\right)^2 \qquad (7)$$

where $\hat{N}_{oi}$ is the normal to the plane defined by the ith image line, R_{mo} is the rotation from model to optical sensor coordinates, $\vec{P}_{mij}$ is the jth endpoint of the ith model line segment in model coordinates, and $\vec{T}_{mo}$ is the translation from model to optical sensor coordinates. The six free variables defining the rigid 3D transformation indicated by R_{mo} and $\vec{T}_{mo}$ constitute the first six out of eight values in the coregistration vector $\mathcal{F}$. The weighting term λ_{oi} is typically 1, but can be used to bias some features over others. For example, λ_{oi} can be used to weight lines based on inverse distance to perform a normalization similar to that considered in Kumar's later measures [Kum89; KH94].

The weighting term $\frac{1}{2n_o\tau_{mo}^2\sum_i\lambda_{oi}}$ normalizes the $E_{fit,o}$ measure based upon the assumption that the image line endpoints fall within τ_{mo} of the associated model line. That is, we can set τ_{mo} to be 2 times the standard deviation of the Gaussian noise process. This will guarantee 98% of the points due to the model will fall within τ_{mo} of the perfect model projection. Notice that this τ_{mo} is in Euclidean units (such as meters) rather than in pixels. An exact upper bound can be found based upon a pixel-based τ, but the formulation is linear in the number of correspondences, rather than constant time. Hence, the simpler τ_{mo}, based on an estimate of total noise on target is computationally necessary in the current implementation. Also notice that while this normalization does not absolutely guarantee that $E_{fit,o}$ will remain in the range $[0,1]$, it does guarantee it assuming that the correspondence does not contain any pairs with image endpoint-to-model-line distances greater than the tolerance τ_{mo} (i.e., outliers).

For range data, 3D Cartesian points are formed by back-projecting range pixels into the scene. The model-to-range fit error, $E_{fit,r}$, is defined to be the squared Euclidean distance between each back-projected range point and its corresponding model point. We write this as:

$$\begin{aligned} E_{fit,r} &= \frac{1}{n_r\tau_{mr}^2\sum_i\lambda_{ri}}\sum_{i=1}^{n_r}\lambda_{ri}\left|\left(\left(\vec{P}_{moi}+\vec{T}_{or}\right)-\vec{P}_{ri}\right)\right|^2 \qquad (8)\\ \vec{P}_{moi} &= R_{mo}\vec{P}_{mi}+\vec{T}_{mo} \end{aligned}$$

where $\vec{P}_{moi}$ is the ith model point mapped into optical sensor coordinates, $\vec{P}_{ri}$ is the corresponding ith measured range point, and $\vec{T}_{or}$ (which is of the form $\left(t_{ol_x}, t_{ol_y}, 0\right)$) is the optical-to-range sensor registration parameter with the coplanarity constraint built in. The two non-zero elements in $\vec{T}_{or}$ constitute the remaining two values in the coregistration vector $\mathcal{F}$.

The lack of a rotation parameter between mc mapped points and the range sensor coordinate system constrains the sensor-to-sensor orientation. Notice that these constraints retain the same degree of nonlinearity (degree 2) found in the original Kumar [Kum89; KH94] and Horn [Hor86] equations. This is desirable, since increasing the nonlinearity of a system of equations tends to increase the instability of the solution. Again, we use a weighted form based on a threshold τ_{mr}, which keeps us generally in the range $[0,1]$.

If we treat R_{mo} as a 3×3 rotation matrix, solving for R_{mo} by minimizing Equation 6 and allowing all 9 terms to vary independently violates the constraint that R_{mo} be a rotation matrix. While the matrix terms $(r_{mc_{1,1}}\ldots r_{mc_{3,3}})$ could be constructed in such a way as to allow only rigid rotations, this would increase the degree of nonlinearity in the equation. Kumar [Kum89; KH94] suggests a better approach: Rodriguez's formula, which is an approximation appropriate for small rotations. To rotate a point $\vec{P_{mi}}$ by an amount R_{mo}, we decompose R_{mo} into an estimate R_{mo}^e and a small update, $\delta\vec{\omega}_{mo}$.

$$\begin{aligned} R_{mo}\vec{P}_{mi} &= R_{mo}^e\vec{P}_{mi}+\delta\vec{\omega}_{mo}\times\left(R_{mo}^e\vec{P}_{mi}\right)\\ &= \vec{P}_{mi}^e+\delta\vec{\omega}_{mo}\times\vec{P}_{mi}^e \qquad (9) \end{aligned}$$

(Note: to provide a more compact notation the vector $\vec{P}_{mi}^e$ is introduced as the current estimate of the transformed model point.) $\delta\vec{\omega}_{mo}$ is the small rotation update represented as a unit rotational axis scaled by the rotational magnitude.

The error terms in equations 7 and 8 may now be rewritten as follows.

$$\begin{aligned} E_{fit,o} &= \frac{1}{2n_o\tau_{mo}^2\sum_i\lambda_{oi}}\sum_{i=1}^{n_o}\sum_{j=1}^{2}\lambda_{oi}\left(\hat{N}_{oi}\cdot\left(\vec{P}^e_{mij}+\delta\vec{\omega}_{mo}\times\vec{P}^e_{mij}+\vec{T}^e_{mo}+\Delta\vec{T}_{mo}\right)\right)^2 \\ E_{fit,r} &= \frac{1}{n_r\tau_{mr}^2\sum_i\lambda_{ri}}\sum_{i=1}^{n_r}\lambda_{ri}\left|\left(\left(\vec{P}^e_{mi}+\delta\vec{\omega}_{mo}\times\vec{P}^e_{mi}+\vec{T}^e_{mo}\right)+\Delta\vec{T}_{mo}+\vec{T}^e_{or}+\Delta\vec{T}_{or}-\vec{P}_{ri}\right)\right|^2 \end{aligned}$$

In order to minimize E_{fit} in equation 6 with respect to $\delta\vec{\omega}_{mo}$, $\Delta\vec{T}_{mo}$ and $\Delta\vec{T}_{or}$, the partial derivatives with respect to each are set to zero and the resulting system of equations solved for the coregistration update parameters. These partial derivatives are shown in Table 1. In these equations, we simplify these by rewriting the weighting terms as

$$\begin{aligned} w_{mo} &= \alpha_{fit}\frac{1}{2n_o\tau_{mo}^2\sum_i\lambda_{oi}} \\ w_{mr} &= (1-\alpha_{fit})\frac{1}{n_r\tau_{mr}^2\sum_i\lambda_{ri}} \end{aligned}$$

The following matrix M_{moi} is introduced to simplify the expressions.

$$M_{moi} = \begin{pmatrix} 0 & \left(\vec{P}^e_{mi}\right)_z & -\left(\vec{P}^e_{mi}\right)_y \\ -\left(\vec{P}^e_{mi}\right)_z & 0 & \left(\vec{P}^e_{mi}\right)_x \\ \left(\vec{P}^e_{mi}\right)_y & -\left(\vec{P}^e_{mi}\right)_x & 0 \end{pmatrix} \tag{10}$$

The matrix M_{moi} is the vector product with $\vec{P}^e_{mi}$. In other words: $M_{moi}\vec{V} = \vec{V}\times\vec{P}^e_{mi}$. It is also the partial derivative of $\delta\vec{\omega}_{mo}\times\vec{P}^e_{mi}$ with respect to $\delta\vec{\omega}_{mo}$.

Setting the partial derivatives in Table 1 to zero yields 9 linear equations in 9 unknowns. These linear equations may be written as:

$$\begin{pmatrix} A & B & C \\ D & E & F \\ G & H & J \end{pmatrix}\begin{pmatrix} \delta\vec{\omega}_{mo} \\ \Delta\vec{T}_{mo} \\ \Delta\vec{T}_{or} \end{pmatrix} = \begin{pmatrix} \vec{K} \\ \vec{L} \\ \vec{M} \end{pmatrix} \tag{14}$$

where the constants A through M are defined in Table 2. Since Δ_{or} has the form $\left(\delta t_{ol_x}, \delta t_{ol_y}, 0\right)$, we can drop the rightmost column and bottom row of the 9×9 matrix in equation 10. The result is an 8×8 linear system which is used to iteratively solve for the optimal set of coregistration parameters.

Each time through the loop, the resulting updates ($\delta\vec{\omega_{mo}}$, $\Delta\vec{T_{mo}}$, and $\Delta\vec{T_{or}}$) are added to the current estimate (R^e_{mo}, $\vec{T^e_{mo}}$, and $\vec{T^e_{or}}$). The constants in Table 2 are recomputed each time through the loop. The algorithm converges when the amount by which E_{fit} drops between successive iterations falls below a preset threshold. Unsuccessful termination occurs if the total number of iterations exceeds a maximum number of iterations. The Levenberg-Marquardt [PFTV88] method has been found to be robust in our past single sensor pose work [BR92b; BR94], and it is used here to find the optimal coregistration parameters.

$$\begin{aligned}\frac{\partial E_{fit}}{\partial \delta\vec{\omega}_{mo}} = {} & w_{mo}\sum_{i=1}^{n_o}\sum_{j=1}^{2}\lambda_{oi}\left(\left(\vec{P}^e_{mij}\times\hat{N}_{oi}\right)\cdot\delta\vec{\omega}_{mo}+\hat{N}_{oi}\cdot\Delta\vec{T}_{mo}\right)\left(\vec{P}^e_{mij}\times\hat{N}_{oi}\right)\\ & + w_{mr}\sum_{i=1}^{n_r}\lambda_{ri}\left(M^2_{moi}\delta\vec{\omega}_{mo}+M_{moi}\Delta\vec{T}_{mo}+M_{moi}\Delta\vec{T}_{or}\right)\\ & + w_{mo}\sum_{i=1}^{n_o}\sum_{j=1}^{2}\lambda_{oi}\left(\hat{N}_{oi}\cdot\left(\vec{P}^e_{moij}+\vec{T}^e_{mo}\right)\right)\left(\vec{P}^e_{mij}\times\hat{N}_{oi}\right)\\ & + w_{mr}\sum_{i=1}^{n_r}\lambda_{ri}M_{moi}\left(\vec{P}^e_{mi}+\vec{T}^e_{mo}+\vec{T}^e_{or}-\vec{P}_{ri}\right)\end{aligned}\tag{11}$$

$$\begin{aligned}\frac{\partial E_{fit}}{\partial \Delta\vec{T}_{mo}} = {} & w_{mo}\sum_{i=1}^{n_o}\sum_{j=1}^{2}\lambda_{oi}\left(\left(\vec{P}^e_{mij}\times\hat{N}_{oi}\right)\cdot\delta\vec{\omega}_{mo}+\hat{N}_{oi}\cdot\Delta\vec{T}_{mo}\right)\hat{N}_{oi}\\ & + w_{mr}\sum_{i=1}^{n_r}\lambda_{ri}\left(M_{moi}\delta\vec{\omega}_{mo}+\Delta\vec{T}_{mo}+\Delta\vec{T}_{or}\right)\\ & + w_{mo}\sum_{i=1}^{n_o}\sum_{j=1}^{2}\lambda_{oi}\left(\hat{N}_{oi}\cdot\left(\vec{P}^e_{mij}+\vec{T}^e_{mo}\right)\right)\hat{N}_{oi}\\ & + w_{mr}\sum_{i=1}^{n_r}\lambda_{ri}\left(\vec{P}^e_{mi}+\vec{T}^e_{mo}+\vec{T}^e_{or}-\vec{P}_{ri}\right)\end{aligned}\tag{12}$$

$$\begin{aligned}\frac{\partial E_{fit}}{\partial \Delta\vec{T}_{or}} = {} & w_{or}\sum_{i=1}^{n_r}\lambda_{ri}\left(M_{moi}\delta\vec{\omega}_{mo}+\Delta\vec{T}_{mo}+\Delta\vec{T}_{or}\right)\\ & + w_{or}\sum_{i=1}^{n_r}\lambda_{ri}\left(\vec{P}^e_{mi}+\vec{T}^e_{mo}+\vec{T}^e_{or}-\vec{P}_{ri}\right)\end{aligned}\tag{13}$$

Table 1: Partial derivatives of coregistration error with respect to free variables

$$A = w_{mo}\sum_{i=1}^{n_o}\sum_{j=1}^{2}\lambda_{oi}\left(\vec{P}^e_{mij}\times\hat{N}_{oi}\right)\left(\vec{P}^e_{mij}\times\hat{N}_{oi}\right)^T + w_{mr}\sum_{i=1}^{n_r}\lambda_{ri}M^2_{moi}\tag{15}$$

$$B = w_{mo}\sum_{i=1}^{n_o}\sum_{j=1}^{2}\lambda_{oi}\left(\vec{P}^e_{mij}\times\hat{N}_{oi}\right)\hat{N}^T_{oi} + w_{mr}\sum_{i=1}^{n_r}\lambda_{ri}M_{moi}\tag{16}$$

$$C = w_{mr}\sum_{i=1}^{n_r}\lambda_{ri}M_{moi}\tag{17}$$

$$D = w_{mo}\sum_{i=1}^{n_o}\sum_{j=1}^{2}\lambda_{oi}\left(\hat{N}_{oi}\left(\vec{P}^e_{mij}\times\hat{N}_{oi}\right)^T\right) + w_{mr}\sum_{i=1}^{n_r}\lambda_{ri}M_{moi}\tag{18}$$

$$E = w_{mo}\sum_{i=1}^{n_o}\sum_{j=1}^{2}\lambda_{oi}\hat{N}_{oi}\hat{N}^T_{oi} + w_{mr}\sum_{i=1}^{n_r}\lambda_{ri}I_3\tag{19}$$

$$F = w_{mr}\sum_{i=1}^{n_r}\lambda_{ri}I_3\tag{20}$$

$$G = w_{mo}\sum_{i=1}^{n_r}\lambda_{ri}M_{moi}\tag{21}$$

$$H = w_{mo}\sum_{i=1}^{n_r}\lambda_{ri}I_3\tag{22}$$

$$J = w_{mo}\sum_{i=1}^{n_r}\lambda_{ri}I_3\tag{23}$$

$$\begin{aligned}\vec{K} = {} & -w_{mo}\sum_{i=1}^{n_o}\sum_{j=1}^{2}\lambda_{oi}\left(\hat{N}_{oi}\cdot\left(\vec{P}^e_{mij}+\vec{T}^e_{mo}\right)\right)\left(\vec{P}^e_{mij}\times\hat{N}_{oi}\right)\\ & -w_{mr}\sum_{i=1}^{n_r}\lambda_{ri}M_{moi}\left(\vec{P}^e_{mi}+\vec{T}^e_{mo}+\vec{T}^e_{or}-\vec{P}_{ri}\right)\end{aligned}\tag{24}$$

$$\begin{aligned}\vec{L} = {} & -w_{mo}\sum_{i=1}^{n_o}\sum_{j=1}^{2}\lambda_{oi}\left(\hat{N}_{oi}\cdot\left(\vec{P}^e_{mij}+\vec{T}^e_{mo}\right)\right)\hat{N}_{oi}\\ & -w_{mr}\sum_{i=1}^{n_r}\lambda_{ri}\left(\vec{P}^e_{mi}+\vec{T}^e_{mo}+\vec{T}^e_{or}-\vec{P}_{ri}\right)\end{aligned}\tag{25}$$

$$\vec{M} = -w_{mr}\sum_{i=1}^{n_r}\lambda_{ri}\left(\vec{P}^e_{mi}+\vec{T}^e_{mo}+\vec{T}^e_{or}-\vec{P}_{ri}\right)\tag{26}$$

Table 2: Constant matrices and vectors in linear update equation.

6.1.2 Coregistration Sensitivity Analysis

The robustness and accuracy of the coregistration algorithm is tested on a controlled set of synthetic data. This data approximates the viewing of vehicle-sized objects at 500 meters and thus reflects the expected RSTA conditions. 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 synthetically projecting the appropriate model features (lines for optical, points for range) onto the sensor.

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}[2], 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: I) sensitivity to noise in initial coregistration estimate, and II) 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. The image features were simply projections of the vertices and edges.

The goal of test I is to probe the sensitivity of the algorithm to noise in the initial coregistration estimate. The convergence properties of the algorithm are tested with regard to both the number of iterations and the quality of the final solution for a set of noisy initial coregistration estimates. Noise is introduced into the model-to-sensors orientation estimate, the model-to-sensors translation estimate, and the relative translation between the sensors.

In test I[3] we observe that, given perfect data and a reasonable initial hypothesis, our system accurately recovers the true pose. In test II we find that the system is relatively stable for small amounts of noise. We also find that the sensitivity of the system to Gaussian noise in the data is dependent upon the Euclidean error introduced rather than the pixel error. Combined, these tests indicate that, when applying coregistration to real data, the hypothesis doesn't need to be perfect, if the extracted features are valid. They also motivate the use of Euclidean τ weights.

6.2 Correspondence-Space Search

The process of generating the best correspondence can be defined to be a combinatorial optimization problem. Consider expanding all of the possible combinations of the potential correspondences, performing coregistration on this subset, and choosing the correspondence with the lowest error. This is impractical, as it is combinatorially explosive. Median filtering, and more sophisticated local search techniques, can be thought of as searching the space of possible correspondence mappings C for one which is best. Median filtering, as performed here, does this by sampling the space of small subsets with the expectation that the best one will probably consist of good correspondences. Local search, as developed in section 6.2.3,is more general in two ways. First, a match error is developed which evaluates coverage of the model as well as quality of fit.

[2] The weights are the combined threshold and α_{fit} term described in [SB94]

[3] A more detailed analysis of the results from test I and II is available in [SB94].

Second, the definition of a discrete neighborhood of alternate solutions guides the search to a local optimum, increasing the chance of finding a good solution in a large combinatorial space over unguided random sampling.

6.2.1 Building Matches: Median Filtering

Least-squares methods, such as our E_{fit} measure, assume that the data has Gaussian random noise added to it. If, however, the correspondence data contains outliers, our method will behave unstably. Median filtering is a robust statistic for detecting and removing outliers [RL87].

Median filtering 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 statistic. The subset which minimizes the median error must contain no outliers, otherwise it would skew the error, increasing the median. And since the median is insensitive to up to 50% outliers, so is median filtering.

The down side is that, for non-differentiable error functions, a combinatorial search of the subset space needs to be explored. To approximate the complete combinatorial search, we can select a number of small subsets, assuming that we have a high probability of sampling at least one subset which contains no outliers. This yields the optimal fit, and allows us to throw out 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 need to be at least large enough to cover the degrees of freedom, so we would need to select at least 3 optical lines and 1 range point. However, Kumar [Kum92] 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 upon the median. We choose the latter method. Assuming a normal distribution, we can set $\texttt{cutoff} = (a \times s)^2$ 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.

6.2.2 Results Using Median Filtering

Three images were used to demonstrate typical behavior of our median filtering matching. The images were presented in Section 3. Parameters were not tuned to specific images, and unless there were specific reasons to do otherwise, defaults reflecting a lack of *a priori* knowledge were selected.

For the fit error, we weighted the CCD and LADAR data equally, since we have no *a priori* knowledge of sensor importance. The only two values we actually set using our test set were the thresholds τ_{mo} and τ_{mr} from equations 7 and 8. We set the CCD threshold $\tau_{mo} = 0.25$ meters, since this is about how far off the worst line features are in any of our test suites. Similarly we use $\tau_{mr} = 5.0$ meters, since there can be such large errors when scanning a surface nearly perpendicular to the view plane, such as the M113's roof in shot 20. We run median filtering on subsets of 10 correspondences for stability reasons. This, however, would indicate that huge number of subsets would need to be examined for initial correspondences of several hundred, such as we are considering. For tractability reasons, we only consider subsets of 300, hypothesizing that most of the data is outlier-free.

When we generate the initial correspondence set, we assume that the pose is nearly correct in the LADAR, but that there is some error in the CCD. We include a pair of model and CCD lines if they have an average distance of less than 30 pixels and are within 15° of each other. For the range correspondences, we consider a significantly narrower band, looking for points that are within 0.5 pixels of each other in both x and y. In the range, however, we have set our threshold to 10 meters. The range tolerance is set to a relatively large value in order to find, with reasonable certainty, corresponding points even when they lie on oblique surfaces.

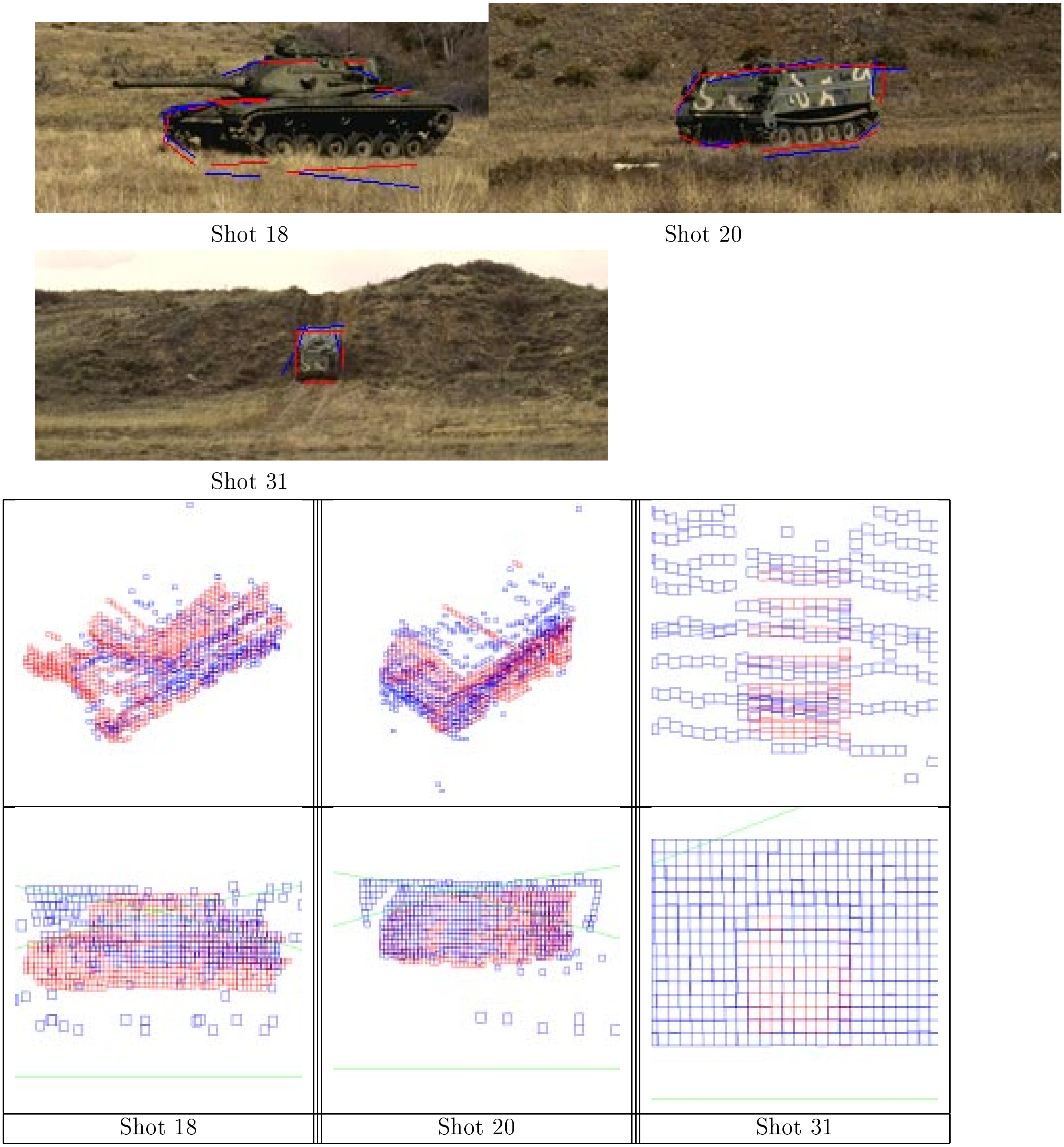
For each of the three images considered, we matched it to its appropriate model: shot 18 to the M60 and shots 20 and 31 to the M113. For these runs, the initial pose is shown in figure 6. These are nearly perfect poses, due to the parameters used to generate the initial correspondence. Some of them contain slight rotations that are corrected for. We will see in section 7 that coregistration space search can deliver nearly this quality of initial hypotheses.

In our results, we show the range features which are not included in the correspondence as hollow, red for model and dark blue for image data. The color lines not in the correspondence are also red and dark blue. The model features in the correspondence are rendered in yellow, while the the data features in the correspondence are colored light blue. The correspondence between individual features is not explicitly shown.

In figures 7, 8, 9 we can see that median filtering constructed a match suitable for recovering an accurate coregistration. With respect to the LADAR data, the planar faces of the M113 (shots 20 and 31) are well aligned, even in shot 31 (figure 9) where there are relatively few pixels on the target faces. In shot 18, we can see that the curvature of the turret and the planar surfaces of the body match well between model and data. This corrects for some initial rotational error, since the initial pose was a head-on view, coming down the hill, while the correct pose was slightly turned towards the left.

In the CCD images, features which arose from background processes (e.g., grass) were removed from the matches. Due to the small number of CCD features, which account for half of the fit error, a single outlier will produce a large error in the coregistration parameters. As a result, the removal of these features (such as the line below the M60 tread in figure 7) accounts for much of the significant improvement.

While the extreme outliers in the CCD data were removed from the match, so were some which would be expected to remain (such as the line along the top of the M60's turret in figure 7). It is important to remember that, when the median filtering is actually being performed, the coregistration parameters are being recovered from a small sample (10 correspondences). While an infinitely large sample of data may display a perfect Gaussian error distribution, a finite sample will only approximate this. As a result, the small sample, while drawn from a Gaussian population, only approximates the true statistics to an accuracy reflected by the sample size.



Shot 18

Shot 20

Shot 31

Shot 18

Shot 20

Shot 31

Figure 6: Initial poses for median filtering.

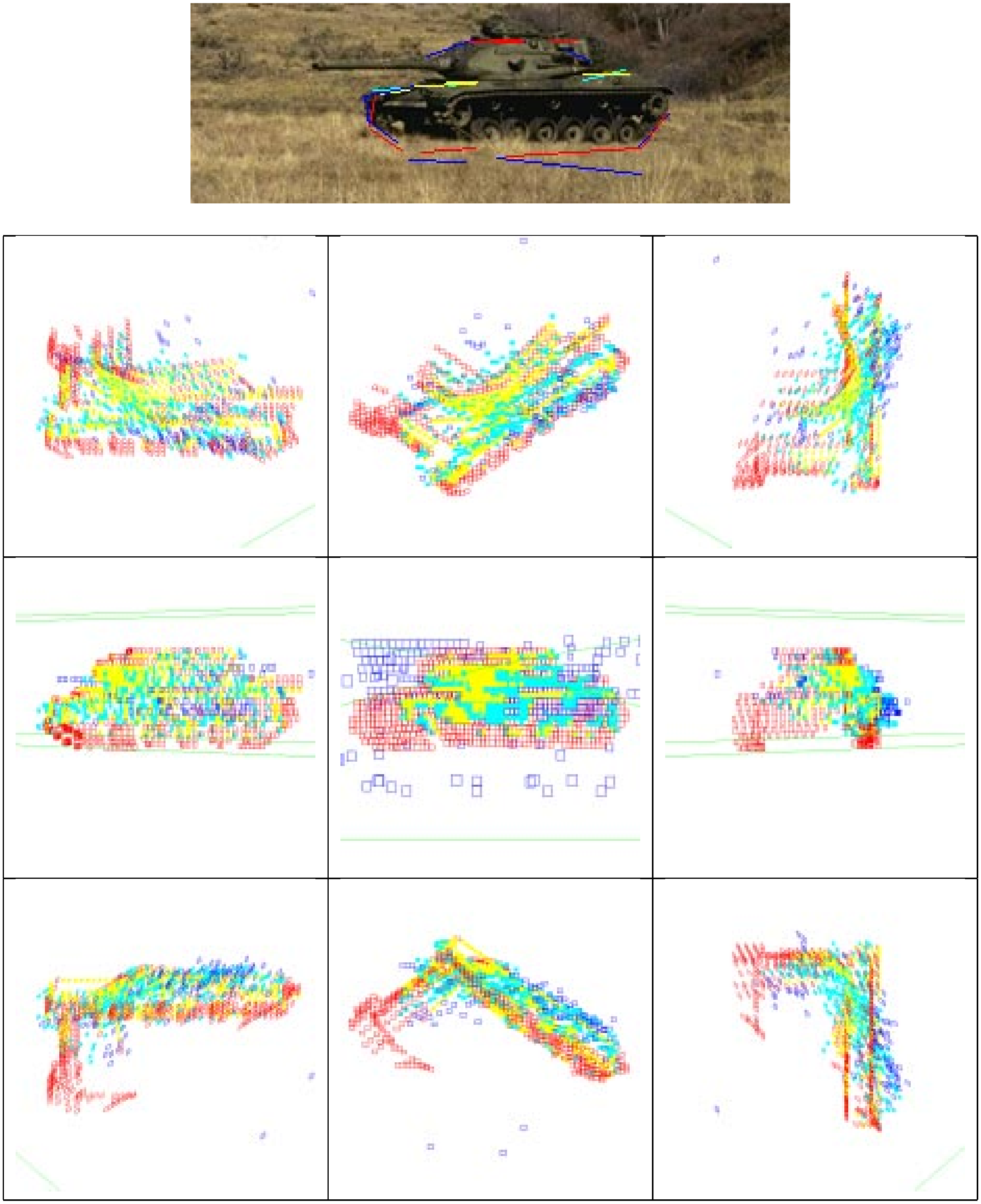


Figure 7: Median Filtering Results on Shot 18, Array 5 using the M60 model.

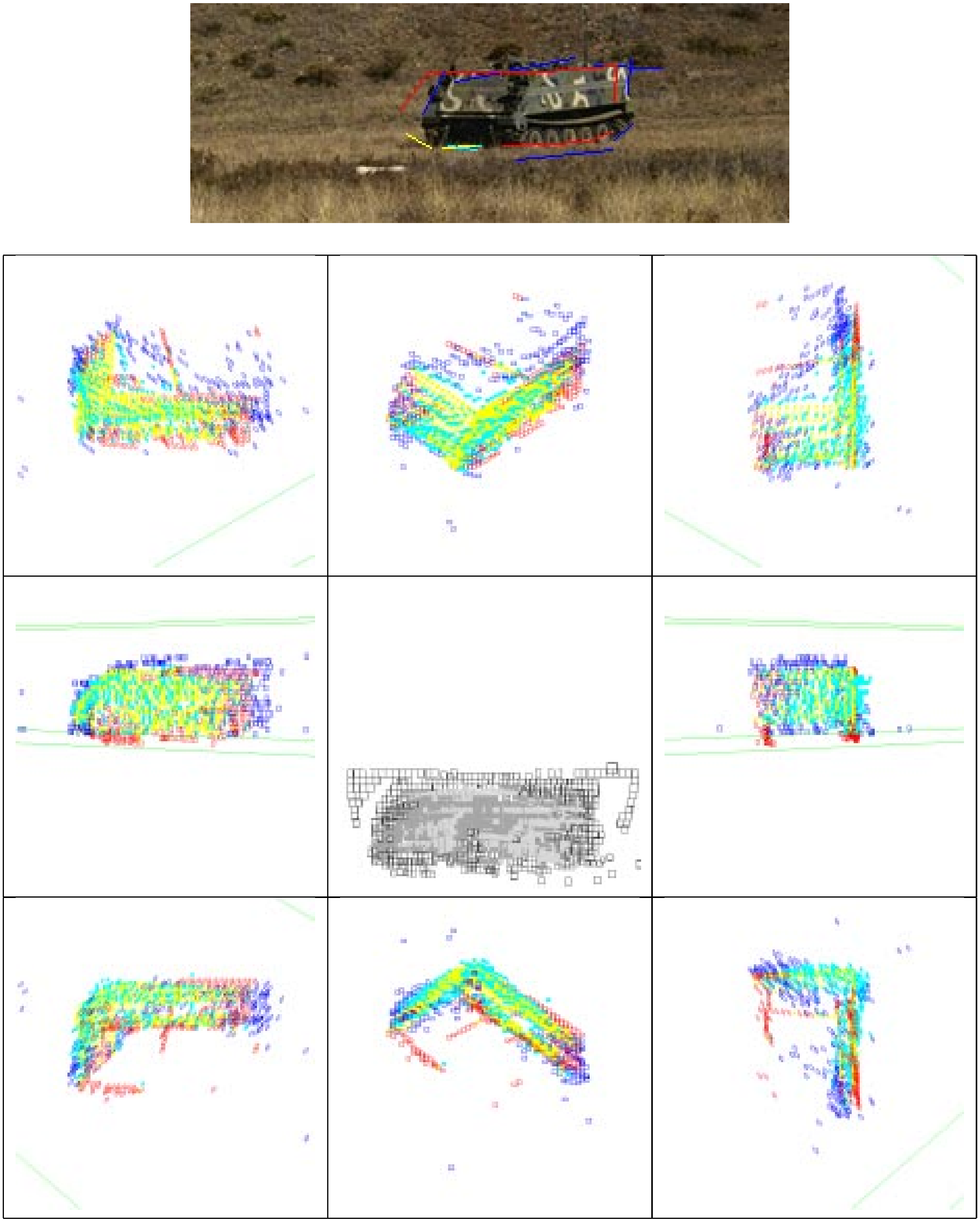


Figure 8: Median Filtering Results on Shot 20, Array 5 using the M113 model.

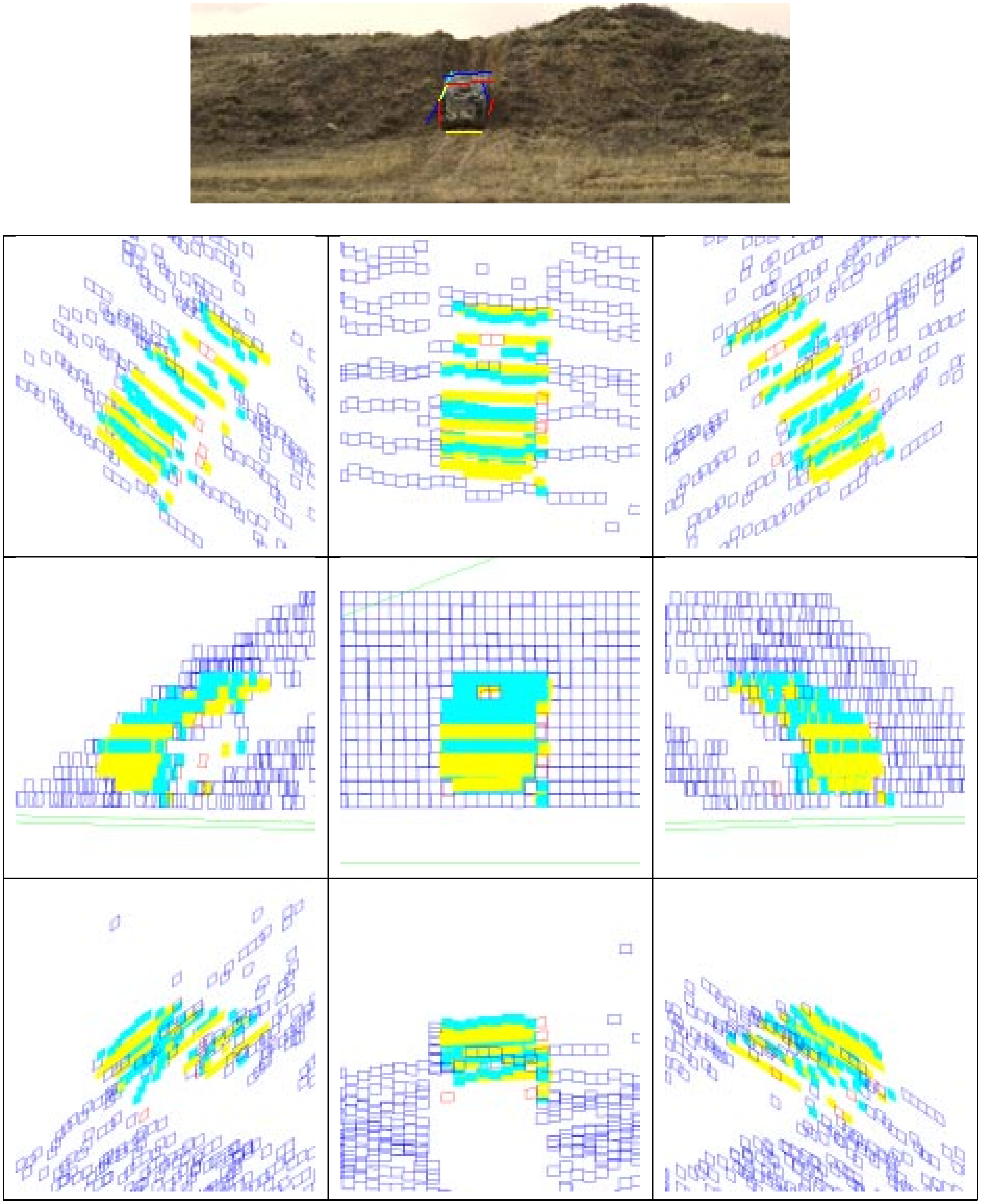


Figure 9: Median Filtering Results on Shot 31, Array 6 using the M113 model.

Shot	Array	Model	E_{fit}	Final Features
18	5	M60	0.0513775	351
20	5	M113	0.045898	393
31	6	M113	0.00483954	74

Table 3: Median filtering results.

These small samples, therefore, are likely to produce slightly incorrect poses, even though they contain no outliers. When the final coregistration is reconstructed from the filtered correspondence set, we are considering a significantly larger sample population. As a result, the coregistration parameters will be slightly different, resulting in the exclusion of some features which should have been included in the match. This is not intolerable, since 1) we are searching for a *near* optimal correspondence from which we can recover accurate coregistration parameters and 2) if we want a more optimal correspondence, we can use local search in correspondence space to fill out the match. The first of these is considered next, and the second is discussed in section 7.1.

The fidelity of the final match is reflected in the quality of the fit error: E_{fit}. Actual final values for E_{fit} generated by median filtering are given in table 3. Typical values for E_{fit} tend to around 0.5 when outliers are included in the subset. In contrast, good subsets typically generate near or below 0.05. This suggest a general rule of thumb for discriminating good matches from bad.

6.2.3 Local Search in Correspondence-Space

This section introduces a more general framework for search in correspondence space. It sets forth a general definition of the problem and relates it to our own previous work in this area. While this approach has yet to yield significant results, it is presented for completeness, as a counterpoint to the coregistration search presented below, and finally because with some modest enhancements, it is expected to become a significant and useful tool. These enhancements are: a local neighborhood in which pairs of features may be added as well as deleted, and some initial grouping of range features.

To begin, the process described above in Section 6.2.2 demonstrates a process by which proximity constraints may be used to define a set of potentially matching features S. In the notation introduced in Section 5, once S is determined, the space of possible matches C is the power set of S. While Median filtering as described above is not typically thought of as an optimal matching algorithm, it can be thought of that way, since it partitions S into two sets: correct feature pairs and outliers. As such, it selects an element of the correspondence space C. The objective function is the median fit value over set of features S based upon the different subsets used to define the coregistration.

A more general formulation of the correspondence problem is to find the best $c \in C$ based upon a more general measure of optimality than fit alone. Fit, as the sole measure, is not adequate, and one way to see this is to consider the null set. By fit alone, the null set is optimal: you cannot have less than 0 squared error. To correct this failing, a match error may be defined which weights both quality of fit and quality of model coverage.

$$E_{match,sensor}(c) = E_{fit,sensor}(c) + E_{om,sensor}(c) \tag{27}$$

We have made extensive use of this *fit-plus-omission* formulation in prior work on matching to optical image features [Bev93]. The term $E_{fit,sensor}$ measures how well the features fit. Here it is defined to be a function of the residual error from the coregistration process. The omission error measures how many of the target features are matched to features in the sensor data: more matches are better.

A simple and general strategy for attempting to find the correspondence c^* which minimizes equation 27 is to pick an initial correspondence c_{start} and then iteratively refine it by adding and removing individual pairs of features. Search stops when no further progress can be made. This constitutes a local search algorithm in correspondence space. For matching problems using line segments derived from models and optical images, we've shown this to be a general and effective technique [Bev93].

A first pass algorithm for minimizing the error defined in equation 27 has been implemented. This algorithm is highly restricted: pairs of features may be removed but not added. Not surprisingly, this restriction limits the algorithms usefulness. Our immediate goals are to extend this neighborhood definition to include addition and deletion of paired features. It is also important that some form of grouping be applied to the range features. Using raw points quickly leads to a combinatorial explosion in the correspondence space.

7 Coregistration-Space Search

As outlined in Section 5, instead of letting the structure of the correspondence mappings between features drive the search, an alternative is to search directly in the space of coregistration parameters. Coregistration parameters $\mathcal{F}$ are perturbed about a current estimate in order to test for better estimates. The process repeats until no improvement is possible. This general approach has been used by others, including some recent work by Sullivan [SWF95] in which vehicles are tracked in video imagery.

Since the space of coregistration parameters is continuous, gradient methods suggest themselves. Sampling about the current estimate might be likened to sampling the gradient, but this interpretation is problematic. The reason is that the error function is neither continuous nor smooth. Here, a simple local search strategy of sampling about the current estimate is used. Future work will explore whether more gradient descent-like methods are more efficient in this space.

Coregistration-space search has one great advantage over correspondence-space search. For correspondence-space search, increased numbers of potentially matching features exponentially increase the size of the correspondence space. In contrast, the coregistration space is fixed: $\Re^8$ for the case treated here. The use of raw sampled range points exacerbates this problem because the absolute numbers of such features tends to be large. Consequently, opening up the uncertainty bounds that define potentially matching features quickly leads to intractably large search spaces. Future work will explore grouping range features to reduce this problem. Until this is done correspondence-space search is limited to problems with very accurate initial coregistration estimates.

In contrast, our early experiments using coregistration-space search suggest an ability to correct quite large errors in the initial coregistration parameters. If any failing has been observed so far, it is a weakness in generating the final highly precise match. In other word, search leads to a much better but not perfect coregistration estimate. This suggests a complementarity between the two approaches which will be examined further in future work.

7.1 A Match Error for Coregistration-Space Search

Once the features expected to be visible for a given pose are generated from the target model data base, they can be used to determine an error function expressing how well these features fit the sensor data. The error function is broken down into two distinct pieces: an error term for the optical data and an error term for the range data. They are combined together using the equation:

$$E_{match}(\mathcal{F}) \;=\; \alpha_{match} E_{match,o}(\mathcal{F}) \;+\; (1 - \alpha_{match}) E_{match,r}(\mathcal{F}) \tag{28}$$

where α_{match} weights the relative importance of the error terms derived from the optical and range imagery. The optical image error term $E_{match,o}$ and range image error term $E_{match,r}$ are are discussed in the next two sections.

7.1.1 Optical Image Error Function

The model-driven technique for placing individual line segments described in Section 4.2 has a weakness for which coregistration-space is a natural cure. Individual features projected from a common target silhouette and independently optimized can 'wander-off' into configurations completely impossible given the global geometric constraints associated with the 3D shape of the target.

Coupling all the features and perturbing them in ways consistent with movement of the target as a whole solves the wandering feature problem. The evaluation for each line is combined into one error term representing how well the silhouette lines fit the underlying image. The summation normalizes the result based on the length of each of the line segments and is given by:

$$E_{match,o}(\mathcal{F}) = \frac{\sum\limits_{i \in lines} E_{Line}(i) \cdot Length(i)}{\sum\limits_{i \in eachline} Length(i)} \tag{29}$$

where

$$E_{Line}(i) \;=\; 1 - \hat{G}_{line}(i) \tag{30}$$

and $\hat{G}_{line}(i)$ is the gradient response defined in equation 1 and $Length(i)$ represents the Euclidean length of each line. The coregistration which minimizes equation 29 maximizes the gradient response under the project silhouette features of the target model.

7.1.2 Range Image Error Function

Using the same basic fit-plus-omission form introduced earlier, the range match error, $E_{match,r}(\mathcal{F})$, is a function of a measure of spatial fit and a measure of omission:

$$E_{match,r}(\mathcal{F}) = \beta_{range} E_{fit,r}(\mathcal{F}) + (1 - \beta_{range}) E_{om,r}(\mathcal{F}) \tag{31}$$

A correspondence mapping c is generated between 3D sampled surface points generated from the target model and 3D points in the range data based upon Euclidean distance between points. Distance is measured with the model placed relative to the data using the current coregistration estimate $\mathcal{F}$. Model points are matched to their nearest neighbors in the range data unless the nearest point is outside a sphere of radius τ, in which case the model point is left unmatched.

Consequently, the fit error $E_{fit,r(i)_{i \in ModelPoints}}$ for a single model point i may be written as:

$$E_{fit,r}(i) = \begin{cases} NearestNeighbor(i) & if NearestNeighbor(i) < \tau \\ 0 & otherwise \end{cases} \tag{32}$$

where:

$$NearestNeighbor(i)_{i \in ModelPoints} = Min((i_x - j_x)^2 + (i_y - j_y)^2 + (i_z - j_z)^2) \forall_{j \in DataPoints} \tag{33}$$

The average over all model points for which there is a corresponding data point is the fit error for the range data:

$$E_{fit,r} = \frac{\sum_{i=0}^{ModelPoints} E_{Fit}(i)}{N \cdot \tau} \tag{34}$$

where N is the number of non-omitted model points, and τ is the minimum distance parameter.

Omission is based on the number of points, p, which do not fall below the minimum threshold τ. The omission term not only accounts for the omitted model points, but also the omitted data points. The reason for the additional term is that the nearest neighbor approach allows many model features to be matched to one data feature. For particularly bad pose hypothesis, all the model features will be paired with a single data point. The omission is given by:

$$E_{om,r} = \left\{ \begin{array}{rl} \frac{1}{2} \cdot (\frac{e^{\alpha p}-1}{e^{\alpha}-1} + \frac{e^{\alpha q}-1}{e^{\alpha}-1}) & \alpha \neq 0 \\ p & \alpha = 0 \end{array} \right\} \tag{35}$$

where the α is defined in terms of the attenuation parameter a:

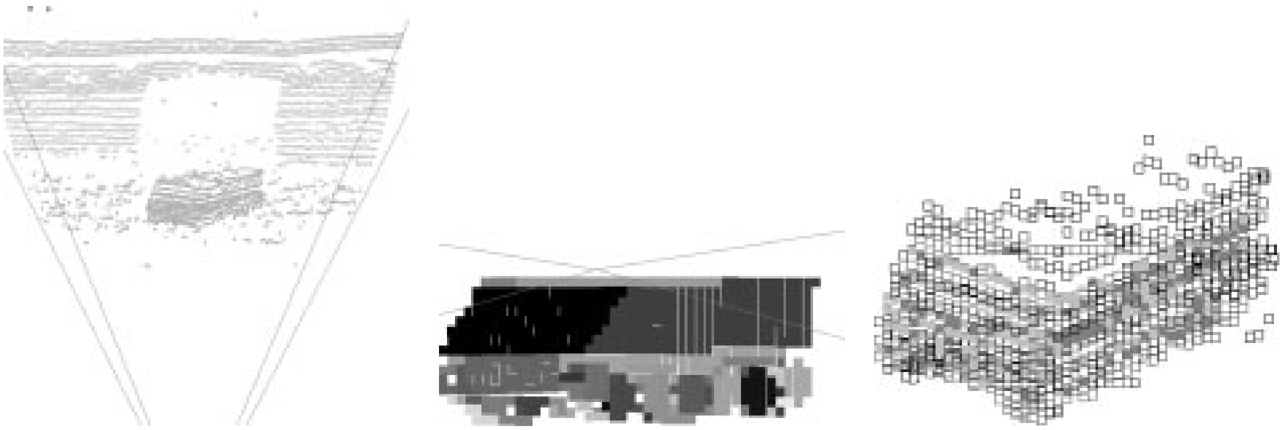
$$\alpha = 2ln(\frac{2}{a} - 1) \tag{36}$$

p is the number of omitted model points, and q is the number of omitted data points.

Figure 10c shows the relationship of the model features generated for the model (Figure 10b), as compared to the sensor data points (Figure 10a). The sensor points are shown as hollow polygons in black. The model features are shown as the lightest gray and hollow. Model points which were not omitted, along with the nearest neighbor data point, are shown rendered solid. For this example τ, the minimum distance parameter is set to $0.01m$, all the hollow model points shown add to the omission error term.

7.2 Local Search in Coregistration-Space

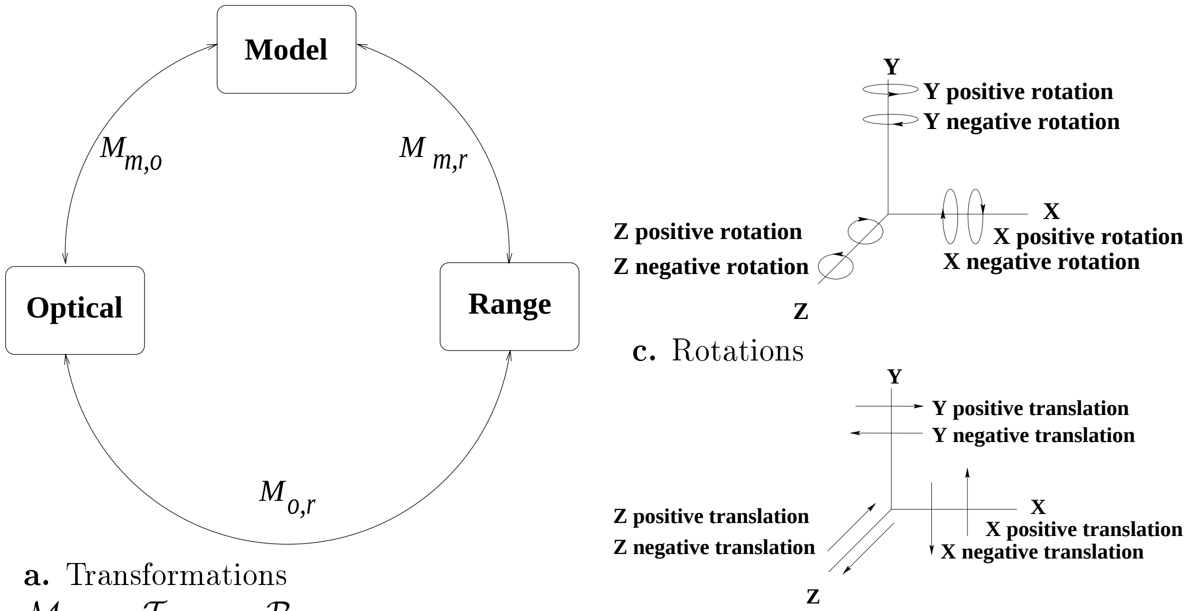
The goal of coregistration-space search process can be viewed as trying to find the optimum set of transformations to move the model into the different sensor coordinate systems (Figure 11a). As



a. LADAR Sensor Image **b.** Sampled Surface **c.** Non-omitted points

Figure 10: Range Data Error Function

stated previously, we must account for eight degrees of freedom in the local search process. For coregistration-space search, we will capture the rotations and translations of Figures 11c and 11d in two sets of matrices: $\mathcal{M}_{m,r}$ and $\mathcal{M}_{r,o}$. The matrix $\mathcal{M}_{m,r}$ relates the target model to the range sensor, and the matrix $\mathcal{M}_{r,o}$ relates the range sensor to the optical sensor. This transformation is restricted to permit only translation in the common image plane. The relationships between these transformations, as well as the transformation between target model and optical sensor, are shown in Figure 11b.



a. Transformations

$\mathcal{M}_{m,r} = \mathcal{T}_{m,r_{x,y,z}} \cdot \mathcal{R}_{m,rx,y,z}$

$\mathcal{M}_{r,o} = \mathcal{T}_{t,ox,y}$

b. Matrices

c. Rotations

d. Translations

Figure 11: Degrees Of Freedom

The goal is to then optimize the match error relative to the eight degrees of freedom in the coregistration $\mathcal{F}$. A coregistration may be written in terms eight values representing the three rotations and five translations:

$$\mathcal{F} \in \Re^8 = \{T_{m,r_x}, T_{m,r_y}, T_{m,r_z}, R_{m,r_x}, R_{m,r_y}, R_{m,r_z}, T_{r,o_x}, T_{r,o_y}\} \tag{37}$$

An initial coregistration estimate $\mathcal{F}_{init}$ is assumed. In the experiments below, this estimate is produced using a LADAR boundary template matching algorithm. Given such an estimate, the goal becomes somewhat less far reaching than that of finding the globally best coregistration. Instead, the goal is to find the best coregistration in some extended neighborhood relative to the initial estimate. This can be done using local search.

As formulated below, local search repeatedly examines a discrete neighborhood of alternatives defined relative to the current best estimate. The neighborhood is obtained by sampling a predetermined number of points along a bounded interval centered about the current estimate. This is done for each of the 8 dimensions of the coregistration space. However, rather than consider all dimensions together, search is divided into alternating steps in which improvement is first sought in the $\Re^6$ subspace representing the pose of the sensors relative to the target, and then in the $\Re^2$ subspace representing translation (registration) between the optical and range sensors.

The outer bounds of these intervals along each dimension of the coregistration space may be defined by homogeneous transformation matrices. Applying a matrix to the current estimate generates a new coregistration at one of the outer boundaries of an interval in which local search will seek a better estimate. Definitions for these matrices are provided in equation 38. The transformations are divided into two sets: $\mathcal{M}_{pose}$ and $\mathcal{M}_{reg}$. This is done because these two sets are used independently in the two alternating stages of the local search process.

$$\mathcal{M}_{pose} \in \Re^6 = \left\{ \begin{array}{rcl} \mathcal{M}_1 & = & +T_{m,r_x}, \\ \mathcal{M}_2 & = & -T_{m,r_x}, \\ \mathcal{M}_3 & = & +T_{m,r_y}, \\ \mathcal{M}_4 & = & -T_{m,r_y}, \\ \mathcal{M}_5 & = & +T_{m,r_z}, \\ \mathcal{M}_6 & = & -T_{m,r_z}, \\ \mathcal{M}_7 & = & +R_{m,r_x}, \\ \mathcal{M}_8 & = & -R_{m,r_x}, \\ \mathcal{M}_9 & = & +R_{m,r_y}, \\ \mathcal{M}_{10} & = & -R_{m,r_y}, \\ \mathcal{M}_{11} & = & +R_{m,r_z}, \\ \mathcal{M}_{12} & = & -R_{m,r_z} \end{array} \right\} \qquad \mathcal{M}_{reg} \in \Re^2 = \left\{ \begin{array}{rcl} \mathcal{M}_{13} & = & +T_{r,r_x}, \\ \mathcal{M}_{14} & = & -T_{r,o_x}, \\ \mathcal{M}_{15} & = & +T_{r,o_y}, \\ \mathcal{M}_{16} & = & -T_{r,o_y}, \\ \mathcal{M}_{17} & = & +T_{o,r_x}, \\ \mathcal{M}_{18} & = & -T_{o,r_x}, \\ \mathcal{M}_{19} & = & +T_{o,r_y}, \\ \mathcal{M}_{20} & = & -T_{o,r_y} \end{array} \right\} \tag{38}$$

The simplest way to use these moves to generate new states would be to apply them directly, neglecting any intermediate states. This would generate a discrete neighborhood consisting of only those coregistration values at the bounds of the intervals defined by the matrices in (38). However, this can lead to missed chances for improvement when a better estimate lies within the interval, but not at its upper or lower bound. To protect against such missed opportunities, coregistration values internal to the interval are sampled. This modestly increases the size of the discrete neighborhood, but greatly improves the performance of the algorithm.

The specific strategy used to sample the intervals is loosely based upon binary search. The underlying idea is that the sampling interval should be successively reduced by half as samples approach the current estimate. This is done by starting with a move matrix in equation 38, and repeatedly halving it. For translation matrices, this is done by simply multiplying the matrix by 0.5. This process stops when the spacing drops below an absolute threshold.

The upper and lower bounds for each dimension are determined heuristically based upon experience with the domain. For translation of the sensors relative to the target, the distance between the centroids of the sampled model and range sensor points is used to initialize this bound [4]. The upper bound on rotation depends upon the expected degree of error in the initial estimate. Values of 10 and 45 degrees respectively were used in the experiments below. To capture translation between sensors, an upper bound of 0.5 meters has been used below.

The local search algorithm uses the transformations defined in equation 38, along with the internal sampling strategy just described, to generate alternative coregistration estimates which are then evaluated using the match error defined in Section 7.1. As mentioned above, search alternates between considering moves from $\mathcal{M}_{m,r}$ and moves from $\mathcal{M}_{r,o}$. Once a local optimum is found, new model features are developed using the algorithm described in Section 4.1. This regeneration of features is critical to the overall success of this approach when local search is correcting for significant errors in the initial coregistration estimate.

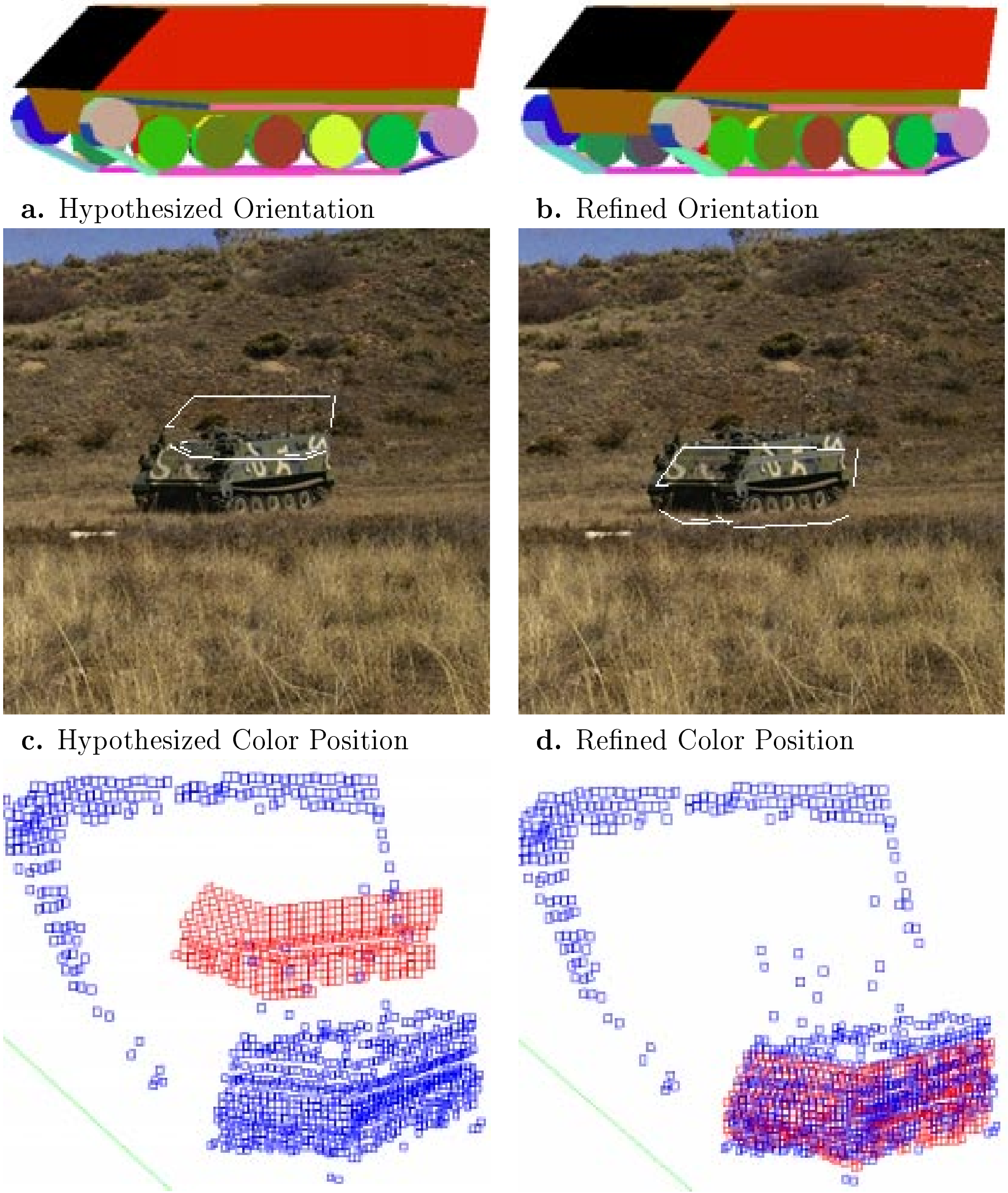
7.3 Results

The Coregistration-space search has been applied to two sets of image from the Fort Carson data set. While the results are preliminary, they are promising. The method is able to significantly improve the initial estimate provided by the range template matching system we are using to generate our initial coregistration estimate. The estimates have significant errors in both rotation and translation which interfere with the model driven feature detection algorithms used to provide features for the Correspondence-space search. Therefore it is essential to run the Coregistration-space search to refine the estimates so more accurate features can be extracted from the data.

The left column of Figures 13 and 12 show the initial estimates provided by the template matching algorithm. The model features are shown in red, and the data features in blue. The right columns show the pose-refinement achieved with the Coregistration-based search. The results are not perfect: in both images local minima in the search space were reached which prevented the algorithm from finding the best global solution. However, the algorithm was able to correct a significant amount of rotation and translation error in the estimate. The refined pose reflects a more accurate estimate of where the vehicle is located in the image.

In order to achieve the results for Shot 20, five moves needed to be made. These moves are summarized in Table 4. Figure 14 shows each error term versus the number of error evaluations. The Figure also shows the error determined for the vehicle when hand placed in the desired position. The main problem with the current search space is immediately obvious from Figure 4c. The color error for the optimum position is not the global optimum for the image. We currently believe the color error to be the weak link in the process, and it may be a result of relying only on the model silhouette lines as the key model features. We are pursuing other methods of extracting features based on vehicle illumination characteristics. While the results for this method

[4] A crude proximity heuristic is used to restrict attention to LADAR range points in the vicinity of the model.



a. Hypothesized Orientation

b. Refined Orientation

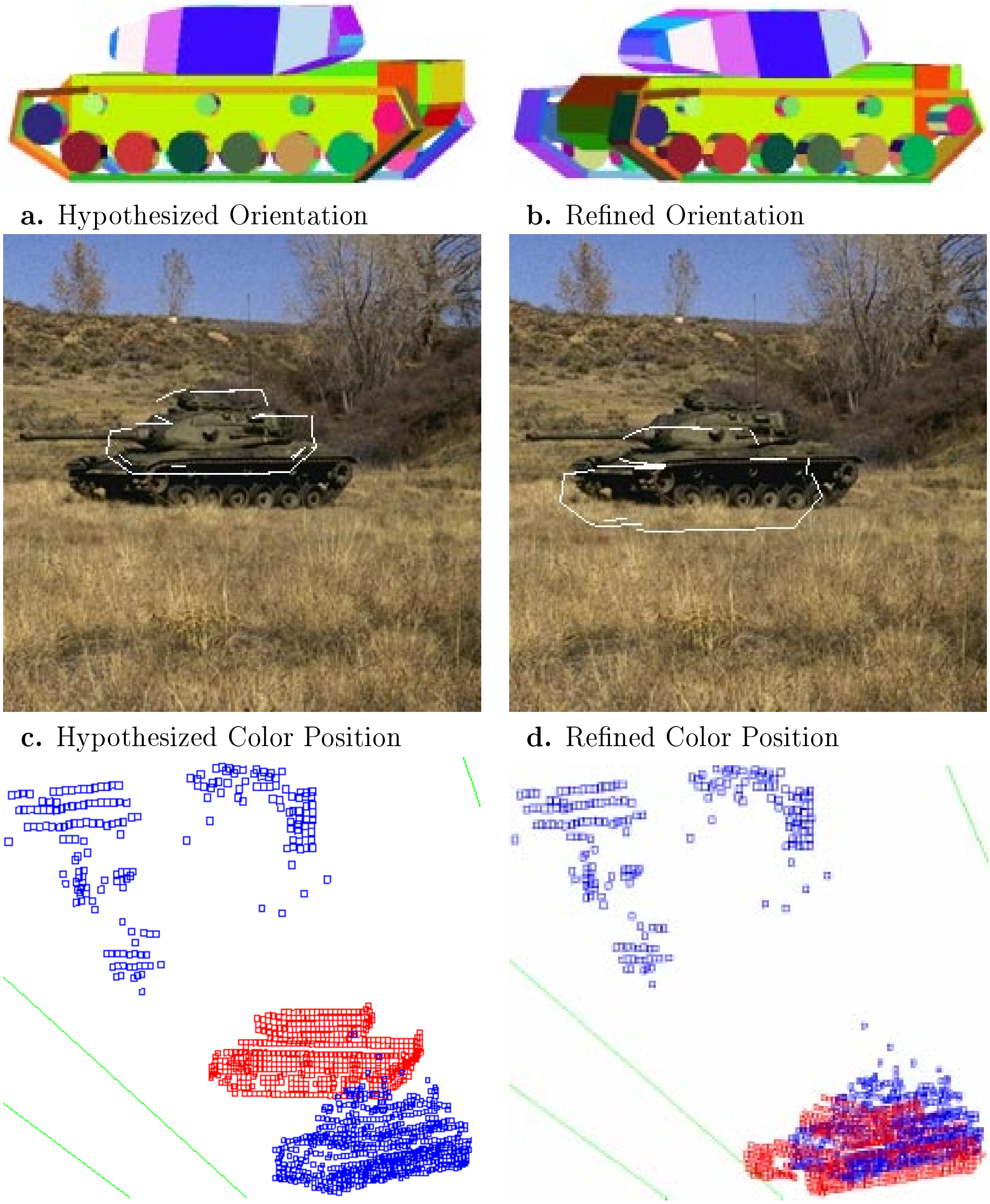
c. Hypothesized Color Position

d. Refined Color Position

e. Hypothesized LADAR Position

f. Refined LADAR Position

Figure 12: Coregistration-space Local Search Results for Shot 20



a. Hypothesized Orientation

b. Refined Orientation

c. Hypothesized Color Position

d. Refined Color Position

e. Hypothesized LADAR Position

f. Refined LADAR Position

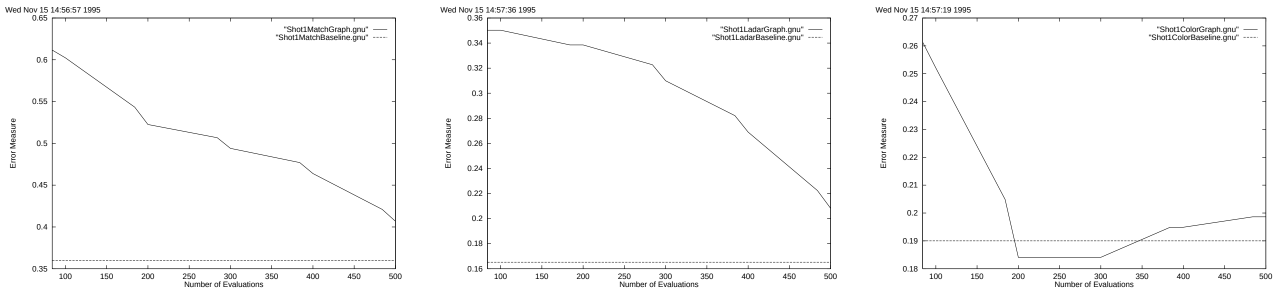
Figure 13: Coregistration-space Local Search Results for Shot 18

are preliminary, we believe they will improve the optical error function and allow search to find better solutions.

The parameters used to achieve the results for Shot 20 were determined through empirical study of how each affects the results of the search. It was found that $E_{match,r}$ was a more stable error measure than $E_{match,o}$, and so α_{match} from equation 28 for $E_{match}(\mathcal{F})$ was set to 0.4. It was also found that the omission term for $E_{match,r}$ had less effect than the fitness term, so β_{range} from equation 31 for $E_{match}(\mathcal{F})$, was also set to 0.4. The last three significant parameters, the omission attenuation, τ, and the initial rotation were set to 1.0, 0.25, and 10° respectively.

Move	Matrix	Translation	Rotation	E_{match}	$E_{match,r}$	$E_{match,o}$
1	$-T_{m,r_z}$	10.06235	0	0.61161	0.35026	0.26135
	$-T_{o,r_y}$	0.2	0	0.60232	0.35026	0.25206
2	$-T_{m,r_y}$	1.25779	0	0.54336	0.33850	0.20486
	$-T_{o,r_y}$	0.1	0	0.52261	0.33850	0.18411
3	$-T_{m,r_z}$	0.62890	0	0.50679	0.32270	0.18409
	$-T_{r,o_x}$	0.1	0	0.49404	0.30995	0.18409
4	$-R_{m,r_y}$	0	10.0	0.47701	0.28211	0.19490
	$-T_{r,o_x}$	0.1		0.46387	0.26897	0.19490
5	$-R_{m,r_y}$	0	10.0	0.42104	0.22241	0.19864
	$-T_{r,o_y}$	0.2	0	0.40680	0.20817	0.19864

Table 4: Moves Made during Hypothesized Pose Refinement Shot 20



a. Match Error **b.** Range Error **c.** Color Error

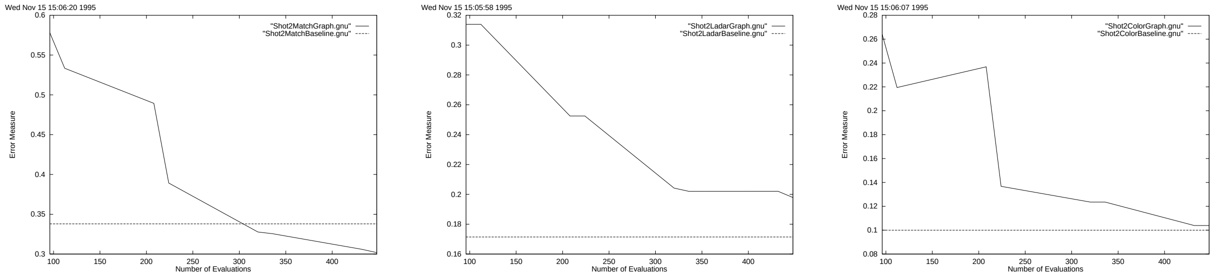
Figure 14: Coregistration-space Local Search Results for Shot 20

On the Shot 18 data, four moves needed to be made, and they are summarized in Table 5. Figure 15 shows each error term in relation to the error determined for the vehicle when hand-placed in the image. Again problems exist with the color error term. The match found is visibly off in translation from the desired position, however it does reflect a considerable refinement from the hypothesized pose. The parameters used to achieve this result are identical to that used in the Shot 20 trial.

The final set of results show Coregistration-space search applied to an initial estimate in which the incorrect vehicle type is hypothesized to be in the image. Figure 16 shows a M60 placed in the image with the same pose estimate used for the M113. The search was able to move the vehicle

Move	Matrix	Translation	Rotation	E_{match}	$E_{match,r}$	$E_{match,o}$
1	$-T_{m,r_z}$	10.71222	0	0.57800	0.31391	0.26409
	$-T_{o,r_y}$	0.2	0	0.53338	0.31391	0.21947
2	$-T_{m,r_y}$	1.33903	0	0.48930	0.25251	0.23679
	$+T_{o,r_y}$	0.2	0	0.38923	0.25251	0.13673
3	$-R_{m,r_y}$	0	22.5	0.32774	0.20419	0.12354
	$+T_{r,o_y}$	0.1		0.32555	0.20201	0.12354
4	$-R_{m,r_y}$	0	22.5	0.30597	0.20206	0.10391
	$+T_{r,o_y}$	0.1	0	0.30179	0.19789	0.10391

Table 5: Moves Made during Hypothesized Pose Refinement of Shot 18



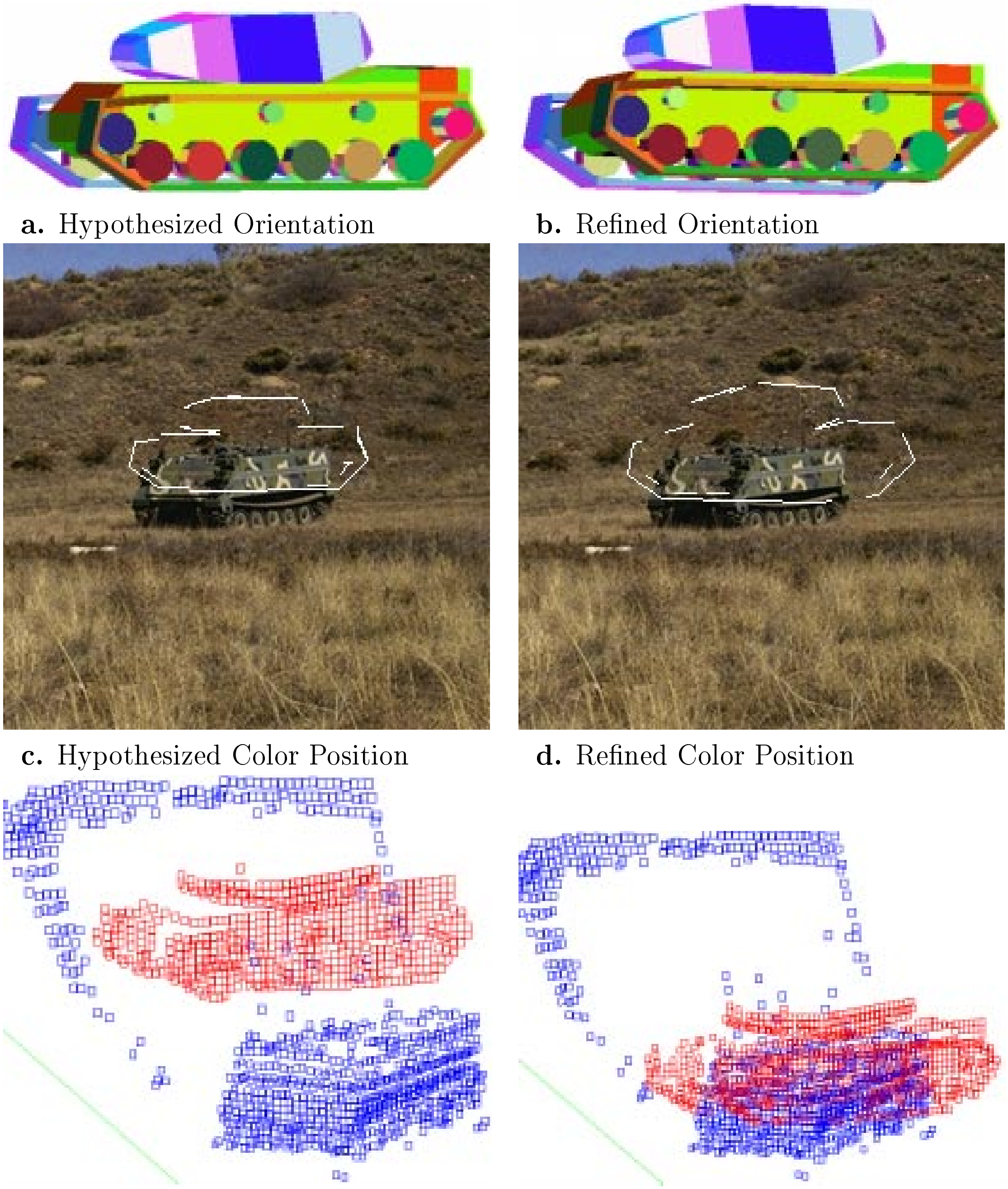
a. Match Error **b.** Range Error **c.** Color Error

Figure 15: Coregistration-space Local Search Results for Shot 18

closer to the correct position, but was not able to do anything else. A local minimum was achieved, and the match error measure was significantly higher. The search correctly distinguished between the two vehicle types.

	M113				M60			
Move	Matrix	E_{match}	$E_{match,r}$	$E_{match,o}$	Matrix	E_{match}	$E_{match,r}$	$E_{match,o}$
0	$-T_{m,r_z}$	0.61161	0.35026	0.26135	$-T_{m,r_z}$	0.60395	0.38116	0.22279
	$-T_{o,r_y}$	0.60232	0.35026	0.25206	$-T_{r,o_y}$	0.58675	0.36396	0.22279
1	$-T_{m,r_y}$	0.54336	0.33850	0.20486	$-R_{m,r_x}$	0.55671	0.35547	0.20124
	$-T_{o,r_y}$	0.52261	0.33850	0.18411	$-T_{r,o_y}$	0.55320	0.35196	0.20124
2	$-T_{m,r_z}$	0.50679	0.32270	0.18409				
	$-T_{r,o_x}$	0.49404	0.30995	0.18409				
3	$-R_{m,r_y}$	0.47701	0.28211	0.19490				
	$-T_{r,o_x}$	0.46387	0.26897	0.19490				
4	$-R_{m,r_y}$	0.42104	0.22241	0.19864				
	$-T_{r,o_y}$	0.40680	0.20817	0.19864				

Table 6: A comparison of Moves Made during Hypothesized Pose Refinement



a. Hypothesized Orientation

b. Refined Orientation

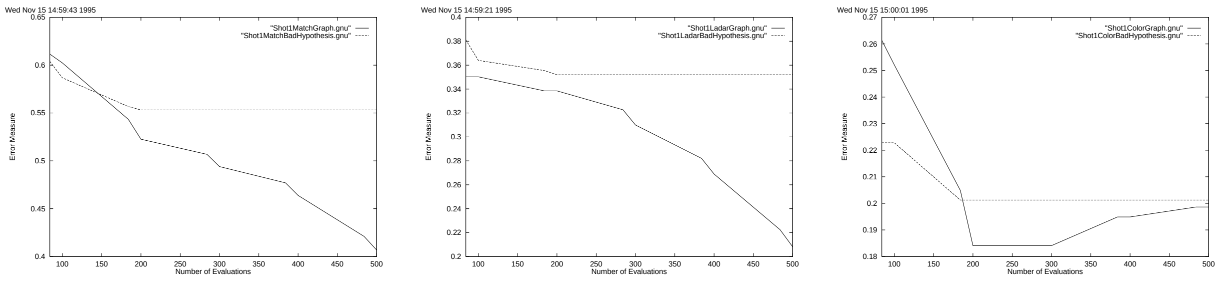
c. Hypothesized Color Position

d. Refined Color Position

e. Hypothesized LADAR Position

f. Refined LADAR Position

Figure 16: Coregistration-space Local Search Results for Shot 20 with incorrect vehicle Hypothesis



a. Match Error **b.** Range Error **c.** Color Error

Figure 17: Coregistration-space Local Search Results for Shot 20

References

[Agg90] 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.

[BC82] R. C. Bolles and R. A. Cain. Recognizing and Locating Partially Visible Objects: The Local-Feature-Focus Method. *International Journal of Robotics Research*, 1(3):57 – 82, 1982.

[BDHR94] 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.

[Bev92a] J. Ross Beveridge. Comparing Subset-convergent and Variable-depth Local Search on Perspective Sensitive Landmark Recognition Problems. In *Proceedings: SPIE Intelligent Robots and Computer Vision XI: Algorithms, Techniques, and Active Vision*, volume 1825, pages 168 – 179. SPIE, November 1992.

[Bev92b] James E. Bevington. Laser Radar ATR Algorithms: Phase III Final Report. Technical report, Alliant Techsystems, Inc., May 1992.

[Bev93] J. Ross Beveridge. *Local Search Algorithms for Geometric Obejct Recognition: Optimal Correspondence and Pose.* PhD thesis, University of Massachuesetts at Amherst, May 1993.

[BGK$^+$89] J. Ross Beveridge, Joey Griffith, Ralf R. Kohler, Allen R. Hanson, and Edward M. Riseman. Segmenting images using localized histograms and region merging. *International Journal of Computer Vision*, 2(3):311 – 347, January 1989.

[BHP94a] 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.

[BHP94b] J. Ross Beveridge, Allen Hanson, and Durga Panda. RSTA Research of the Colorado State, University of Massachusetts and Alliant Techsystems Team. In *Image Understanding Workshop (separate addendum)*, page (to appear). ARPA, November 1994.

[BHR86] J. Brian Burns, Allen R. Hanson, and Edward M. Riseman. Extracting Straight Lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(4):425–455, July 1986.

[BJLP92] James Bevington, Randy Johnston, Joel Lee, and Richard Peters. A modular target recognition algorithm for ladar. In *Proc of the 2nd Automatic Target Recognizer Systems and Technology Conference*, pages 91 – 104, Fort Belvoir, VA, mar 1992.

[BPY94] 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.

[BR92a] J. Ross Beveridge and Edward M. Riseman. Can Too Much Perspective Spoil the View? A Case Study in 2D Affine Versus 3D Perspective Model Matching. In *Proceedings: Image Understanding Workshop*, pages 665 – 663, San Mateo, CA, January 1992. Morgan Kaufmann.

[BR92b] J. Ross Beveridge and Edward M. Riseman. Hybrid Weak-Perspective and Full-Perspective Matching. In *Proceedings: IEEE 1992 Computer Society Conference on Computer Vision and Pattern Recognition*, pages 432 – 438. IEEE Computer Society, June 1992.

[BR94] 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).

[BR95] 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).

[BWR89] J. Ross Beveridge, Rich Weiss, and Edward M. Riseman. Optimization of 2-dimensional model matching. In *Proceedings: Image Understanding Workshop*, pages 815 – 830, Los Altos, CA, June 1989. DARPA, Morgan Kaufmann Publishers, Inc (Also a Tech. Report).

[BWR90] 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.

[BWR91] J. Ross Beveridge, Rich Weiss, and Edward M. Riseman. Optimization of 2-Dimensional Model Matching. In Hatem Nasr, editor, *Selected Papers on Automatic Object Recognition (originally appeared in DARPA Image Understanding Workshop, 1989)*, SPIE Milestone Series. SPIE, Bellingham, WA, 1991.

[Can86] John Canny. A computational approach to edge detection. *PAMI*, 8(6):679–698, November 1986.

[Cla89] James J. Clark. Authenticating Edges Produced by Zero-Crossing Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11(1):43–57, January 1989.

[DVD93] Richard L. Delanoy, Jacques G. Verly, and Dan E. Dudgeon. Machine Intelligent Automatic Recognition of Critical Mobile Targets in Laser Radar Imagery. *The Lincoln Laboratory Journal*, 6(1):161–186, Spring 1993.

[EG92] 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.

[FA91] William T. Freeman and Edward H. Adelson. The Design and Use of Steerable Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13(9):891–906, September 1991.

[FL87] Pascal Fua and Yvan G. Leclerc. Finding Object Boundaries Using Guided Gradient Ascent. In *Proceedings: Image Understanding Workshop – 1987*, pages 888–891. DARPA, Morgan Kaufmann, February 1987.

[FL88] Pascal Fua and Yvan G. Leclerc. Model Driven Edge Detection. In *Proceedings: Image Understanding Workshop – 1988*, pages 1016–1020. DARPA, Morgan Kaufmann, April 1988.

[GBSF94] Michael E. Goss, J. Ross Beveridge, Mark R. 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. ARPA, nov 1994.

[GBSF95] Michael E. Goss, J. Ross Beveridge, Mark R. Stevens, and Aaron D. Fuegi. Three-dimensional visualization environment for multi-sensor data analysis, interpretation, and model-based object recognition. In Georges G. Grinstein and Robert F. Erbacher, editors, *Proceedings: Visual Data Exploration and Analysis II*, pages 283–291. SPIE Vol. 2410, feb 1995.

[GH91] W. Eric L. Grimson and Daniel P. Huttenlocher. On the Verification of Hypothesized Matches in Model-Based Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(12):1201 – 1213, December 1991.

[GJSL90] Jeffrey Gillberg, Randy Johnston, Kris Siejko, and Joel Lee. Laser Radar ATR Algorithms. Technical Report DAABO7-87-C-F109, Honeywell Systems and Research Center, Minnesota, May 1990.

[Gri90] W. Eric L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints.* MIT Press, Cambridge, MA, 1990.

[Hil83] Ellen C. Hildreth. The Detection of Intensity Changes by Computer and Biological Vision Systems. *Computer Vision, Graphics, and Image Processing*, 22:1–27, 1983.

[HKK90] Martial Hebert, Takeo Kanade, and InSo Kweon. 3-D Vision Techniques for Autonomous Vehicles. In Ramesh C. Jain and Anil K. Jain, editors, *Analysis and Interpretation of Range Images*, chapter 7. Springer-Verlag, 1990.

[Ho93] Y. Hel-or. *Model Based Pose Estimation from Uncertain Data.* PhD thesis, Hebrew Univeristy in Jerusalem, 1993.

[Hor86] B. K. P. Horn. *Robot Vision.* The MIT Press, 1986.

[HOW93] Y. Hel-Or and M. Werman. Absolute Orientation from Uncertain Data: A Unified Approach. In *Proceedings: Computer Vision and Pattern Recognition*, pages 77 – 82. IEEE Computer Society Press, June 1993.

[HU87] Daniel P. Huttenlocher and Shimon Ullman. Object Recognition Using Alignement. In *Proc. First International Conference on Computer Vision*, pages 102–111, London, England, June 1987. Computer Society Press of the IEEE.

[HU88] Daniel P. Huttenlocher and Shimon Ullman. Recognizing Solid Objects by Alignment. In *Proc. of the DARPA Image Understanding Workshop*, pages 1114 – 1124, Cambridge, April 1988. Morgan Kaufman Publishers, Inc., New York.

[HU90] 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.

[KD87] Matthew R. Korn and Charles R. Dyer. 3D Multiview Object Representations for Model-Based Object Recognition. *Pattern Recognition*, 20(1):91–103, 1987.

[KH94] Rakesh Kumar and Allen R. Hanson. Robust methods for estimating pose and a sensitivity analysis. *CVGIP:Image Understanding*, 11, 1994.

[Kum89] Rakesh Kumar. Determination of Camera Location and Orientation. In *Proceedings: Image Understanding Workshop*, pages 870 – 881, Los Altos, CA, June 1989. DARPA, Morgan Kaufmann Publishers, Inc.

[Kum92] Rakesh Kumar. *Model Dependent Inference of 3D Information From a Sequence of 2D Images.* PhD thesis, University of Massachusetts, Amherst, February 1992.

[KvD76] J. J. Koenderink and A. J. van Doorn. The Singularities of Visual Mapping. *Biological Cybernetics*, 24:51–59, 1976.

[KvD79] J. J. Koenderink and A. J. van Doorn. The Internal Representation of Shape with Respect to Vision. *Biological Cybernetics*, 32:211–216, 1979.

[LB83] David G. Lowe and T. O. Binford. The Perceptual Organization of Visual Images: Segmentation as a Basis for Recognition. In *Proceedings Image Understanding Workshop, Stanford*, pages 203 – 209, June 1983.

[LB85] David G. Lowe and Thomas O. Binford. The Recovery of Three-Dimensional Structure from Image Curves. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7(3):320 – 325, 1985.

[Low85] David G. Lowe. *Perceptual Organization and Visual Recognition.* Kluwer Academic Publishers, 1985.

[Low91] David G. Lowe. Fitting Parameterized Three-Dimensional Models to Images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(5):441 – 450, May 1991.

[MBCA85] 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.

[MH80] David Marr and Ellen C. Hildreth. Theory of Edge Detection. *Proceedings of the Royal Society of London*, B207:187–217, 1980.

[PD87] Harry Platinga and Charles Dyer. Visibility, Occlusion, and the Aspect Graph. Technical Report 736, University of Wisconsin - Madison, December 1987.

[PFTV88] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, Cambridge, 1988.

[Pin88] Juan Pineda. A Parallel Algorithm for Polygon Rasterization. In *Proceedings of Siggraph '88*, pages 17–20, 1988.

[Pla88] William Harry Plantinga. *The ASP: A Continuous, Viewer-Centered Object Representation for Computer Vision.* PhD thesis, University of Wisconsin at Madison, 1988.

[RL87] Peter J. Rousseeuw and Annick M. Leroy. *Robust Regression and Outlier Detection.* Wiley, 1987.

[SB94] 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.

[SBG95] Mark R. Stevens, J. Ross Beveridge, and Michael E. Goss. Reduction of BRL/CAD Models and Their Use in Automatic Target Recognition Algorithms. In *BRL/CAD Symposium 95*, 1995.

[SD92] W. Brent Seales and Charles R. Dyer. Modeling the Rim Appearance. In *Proceedings of the 3rd International Conference on Computer Vision*, pages 698–701, 1992.

[SG87] 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.

[Shu94] Alexander Shustorovich. Scale Specific and Robust Edge/Line Encoding with Linear Combinations of Gabor Wavelets. *Pattern Recognition*, 27(5):713–725, 1994.

[Ste95] Mark R. Stevens. Obtaining 3D Silhouettes and Sampled Surfaces from Solid Models for use in Computer Vision. Master's thesis, Colorado State University, September 1995.

[SWF95] 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.

[TP86] Vincent Torre and Tomaso A. Poggio. On Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(2):147–164, March 1986.

[U. 91] U. S. Army Ballistic Research Laboratory. *BRL-CAD User's Manual*, release 4.0 edition, December 1991.

[VDL95] J. G. Verly, D. E. Dudgeon, and R. T. Lacoss. Progress Report on the Development of the Automatic Target Recognition System for the UGV/RSTA LADAR. Technical Report 1006, Massachusetts Institute of Technology, Lincoln Laboratory, March 1995.

[Y. 94] Y. Hel-Or and M. Werman. Constraint-Fusion for Interpretation of Articulated Objects. In *Proceedings: Computer Vision and Pattern Recognition*, pages 39 – 45. IEEE Computer Society Press, June 1994.