Interleaving 3D Model Feature Prediction and Matching to Support Multi-Sensor Object Recognition *

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

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

Abstract

The object recognition system presented combines on-line feature prediction with an iterative multisensor matching algorithm. Matching begins with an initial object type and pose hypothesis. An iterative generate-and-test procedure then refines the pose as well as the sensor-to-sensor registration for separate range and electro optical sensors. During matching, object features predicted to be visible are updated to reflect changes in hypothesized object pose and sensor registration. The match found is locally optimal in terms of the complete space of possible matches and globally consistent in the sense of preserving the 3D constraints implied by sensor and object geometry. Results on real data are presented which demonstrate the algorithm correcting for up to 30° *errors in initial orientation and* $5m$ *errors in initial translation.*

1 Introduction

Common approaches to model feature prediction use static data structures to store feature visibility information associated with object geometry alone [12, 4]. Typically, all possible viewpoints are divided into sets of constant model topology [6] and visible face and occluding contour information are stored for each topological region [14, 7]. Statistical modeling information for predicting feature visibility [19, 3] may also be stored.

We have developed an on-line prediction capability which performs the mapping from stored model to predicted features dynamically as part of matching. A key to making this approach feasible is the development of algorithms which run many, if not all, computations in parallel on standard graphics acceleration hardware. This on-line capability permits us to develop a tight coupling between feature prediction and local search matching.

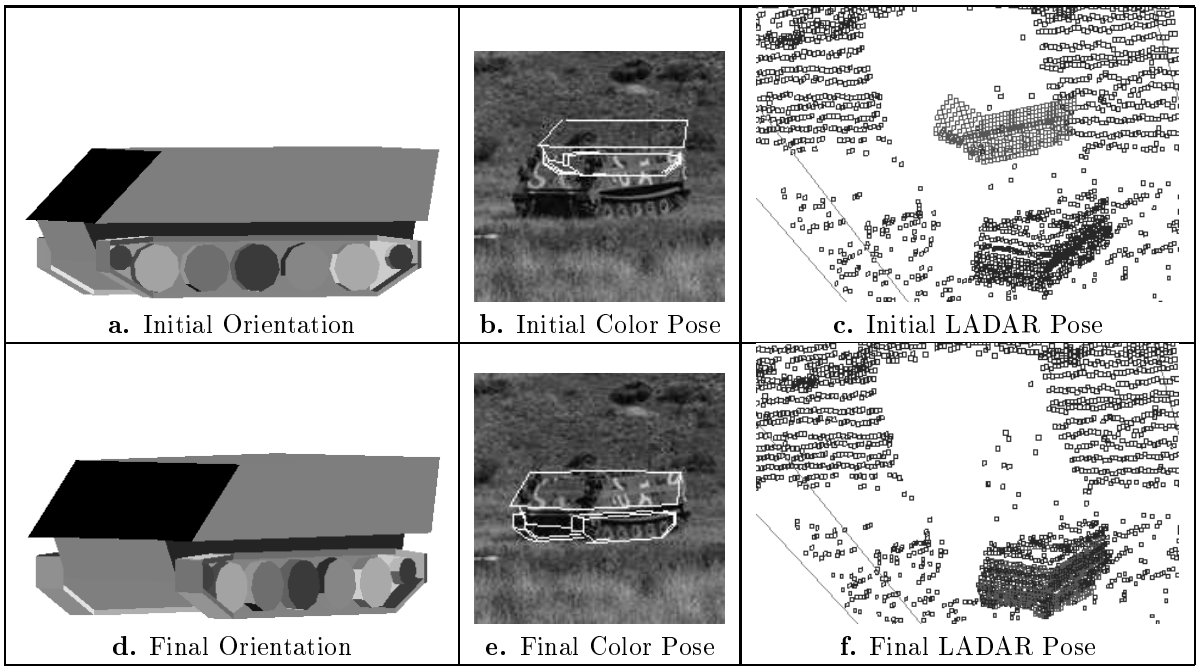
The local search algorithm matches 3D object model features to features from an optical and a range sensor. The range image provides critical 3D constraints missing in optical imagery alone. However, matching must refine the estimated 3D pose of the object relative to the sensor suite as well as correct for small errors in the pixel-to-pixel registration estimate between the two sensors. We will use the term *coregistration* to describe the combined object pose and multisensor image registration refinement process.

The coregistration matching algorithm is part of a larger Automatic Target Recognition (ATR) system [16] which refines and verifies target type and pose estimates provided by up-stream processes. Range and color imagery collected at Fort Carson, Colorado in November 1993 [1] is used to demonstrate our algorithm. This data is publicly available through our web site: `http://www.cs.colostate.edu/~vision`. Figure 1 presents a pair of range and color images in which the target is roughly $50m$ away in a fairly open area. The second image set, Figure 2, is more difficult with the vehicle approximately $100m$ away on a hillside. The matching shown in the figures is explained below.

2 3D Model Feature Prediction

To *coregister* an object model to optical and range imagery, prediction extracts features suitable for matching from the CAD model. Feature prediction takes 3D object models derived from BRL-CAD [17] constructive solid geometry format. The algorithms used to reduce model complexity are presented elsewhere [15]. The CAD model for the M113 APC is used for illustration in this paper. Matching has also been done with an M60 model [16].

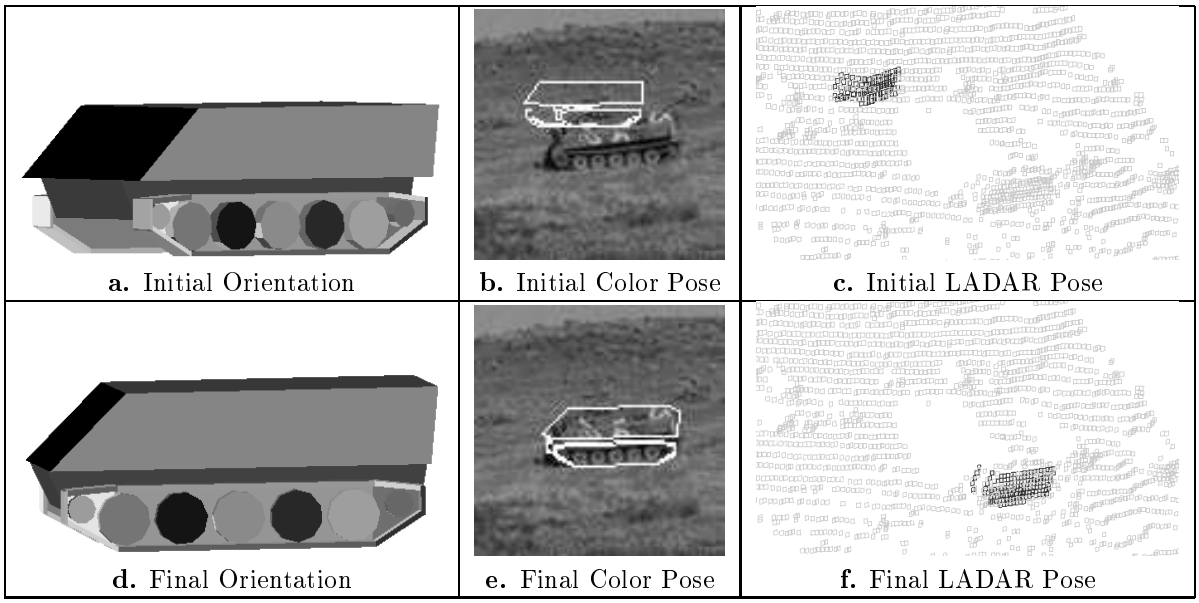
*This work was sponsored by the Defense Advanced Research Projects Agency (DARPA) Image Understanding Program under grants DAAH04-93-G-422 and DAAH04-95-1-0447, monitored by the U. S. Army Research Office



a. Initial Orientation | **b.** Initial Color Pose | **c.** Initial LADAR Pose

d. Final Orientation | **e.** Final Color Pose | **f.** Final LADAR Pose

Figure 1. Local Search Results for Shot 20 Array 5 (M113 APC)



a. Initial Orientation | **b.** Initial Color Pose | **c.** Initial LADAR Pose

d. Final Orientation | **e.** Final Color Pose | **f.** Final LADAR Pose

Figure 2. Local Search Results for Shot 35 Array 9 (M113 APC)

Two sets of features are used in matching: model edges and discrete sampled surfaces. An object's silhouette is one valuable recognition cue when dealing with two-dimensional optical imagery [10]. Many systems recognize 3D objects using stored 2D silhouettes [8, 18] while 3D edge silhouette representations such as we produce are less common [2]. In our imagery, internal features as well as silhouette features are required to overcome ambiguity. Extracting 3D features rather than 2D features enables matching to explore minor adjustments to 3D pose.

2.1 Silhouette Lines

To find the parts of the CAD model inducing the silhouette, a unique color is assigned to each 3D face: this color value then serves as a face index. Next, the

model is rendered from the hypothesized viewing orientation. Rendering is performed on a hardware Z-buffer, and hence can be done quickly. Running on a Sparc 10 with a ZX accelerator, rendering takes roughly 0.3 seconds for a model containing 250 faces. Pixel color indicates the visible face and the pixels within an 8-adjacent neighbor to the unique background color are noted as participating in the silhouette.

Next, it is determined which individual face boundaries (edges) generate the silhouette. An edge is a possible silhouette edge only if one of the two bounding faces is visible [14]. This step may leave some edges which are actually internal as hypothesized silhouette edges, so a clipping algorithm discovers and discards edges and portions of edges not on the silhouette. Clipping projects the 3D model edge endpoints onto the image plane. A line following algorithm then traverses the segment to find the parametric end-points which correspond to the beginning and ending portion of the silhouette edge. Because an orthographic projection is used to render the model, parametric end-point values may be applied directly to the corresponding model edges to produce the resulting 3D silhouette edges.

2.2 Internal Lines

Whether an edge causes a significant change in illumination depends upon object geometry and the angle of the sun. The sun is modelled as an area light source and the vector to the sun is calculated using a long/lat estimate, time of day, date, and compass orientation [11] (we have this collateral information [1]). The dot product of the sun direction vector and the normal of each face is determined. For all visible edges, if the sign of these dot products differ for two adjoining faces then the edge is marked as likely to be visible. This simple test determines edges for which light will be cast onto only one of the adjoining faces.

Clipping, similar to that used in obtaining silhouettes, is used detect self-occlusion. After both silhouette and internal edges are determined for a given pose hypothesis, shorter lines are discarded using a user specified minimum distance threshold. Figures 1b and 1e show the silhouette and internal edges used in the matching process for the pose hypothesis given in Figures 1a and 1d.

2.3 Predicting Sampled Surfaces

A 3D sampled surface is generated by simulating the operation of the LADAR range sensor. The CAD model is transformed into the range sensor's coordinate system using the current target pose estimate. Based on the characteristics of the range device, rays are cast into the scene and intersected with the 3D faces of the CAD model. Face rendering used in optical feature prediction limits intersection computation to only visible faces. The closest face intersection is stored as the depth of the specific range pixel. Figures 1c and 1f show the sampled surface information used in the matching process for the pose hypothesis given in Figures 1a and 1d.

3 Local Search for Coregistration

An error function relates model features to the data and a search mechanism finds a coregistration estimate which minimizes that error. The match error function defining the quality of a match $\mathcal{M}$ may be written as:

$$E_{\mathcal{M}}(c, \mathcal{F}) > 0 \quad \forall c \in C,\ \mathcal{F} \in \Re^8 \tag{1}$$

The first argument, 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 model. In the most general case, the correspondence space C is the super-set of all possible combinations of pairs of sensor and model features.

The coregistration, $\mathcal{F}$, represents the geometric relationship between the sensors and the model. Here, it 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 values encode the planar translation of the optical image plane relative to the range sensor's image plane. The rationale for this planar translation constraint is explained in [5].

One can develop algorithms which emphasize search in either the space of possible correspondence mappings C or the space of coregistration estimates $\mathcal{F} \in \Re^8$. In other work, we consider search in C [13], while here we address search in $\mathcal{F}$.

3.1 Coregistration Matching

Matching uses an iterative generate-and-test procedure (Figure 3) in which the current coregistration hypothesis $\mathcal{F}$ is used to predict a set of model features which are in turn used in the error evaluation function. A neighborhood of moves is then examined and the best move, the one with the lowest error, is taken. The features are re-generated for the new coregistration estimate and the process continues.

The neighborhood decouples the 8 dimensions of the coregistration space into three distinct sets of possible moves. The first set represents the 6 dimensions which encode the 3D pose of the sensors relative to the object. The other two represent changes in registration between the two sensors. Search examines pose moves first until a local optimum is reached. Then and only then are changes in sensor registration considered. If a

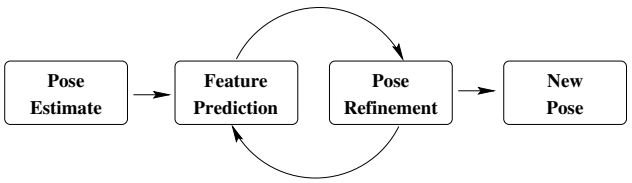


Figure 3. Pose Refinement

change is made in sensor registration, then additional changes in the pose are again considered.

When no further progress is possible along any dimension, the resulting 8 values are returned as the locally optimal coregistration estimate. Initial results of the search have shown that the local optima in color space, and the local optima in range space, do not usually coincide. By searching for the model in both the optical and range imagery, local optima in each will be rejected in favor of a globally better solution.

3.2 Error Terms

The error function, $E_{\mathcal{M}}(\mathcal{F})$, consists of two components: a weighted term representing how well the 3D model line segments fit the current color image, and a weighted term representing how well the sampled surface information fits the range data. These two terms are combined to form the overall match error:

$$E_{\mathcal{M}}(\mathcal{F}) = \alpha_{\mathcal{M}} E_{\mathcal{M},o}(\mathcal{F}) + (1-\alpha_{\mathcal{M}}) E_{\mathcal{M},r}(\mathcal{F}) \quad (2)$$

Each sensor term can be further broken down into two weighted terms: an omission error and a fitness error.

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

The subscript ($\mathcal{S}$) can be replaced with either o for optical or r for range. The fitness error $E_{fit,\mathcal{S}}(\mathcal{F})$ represents how well the strongest features (as determined by a threshold) match, and the omission error $E_{om,\mathcal{S}}(\mathcal{F})$ penalize the match wherein model features are left unmatched. This happens when no adequate matching features can be found in the sensor data.

3.2.1 Optical Fitness Error: $E_{fit,o}(\mathcal{F})$

The optical fitness error represents how well each model line fits the underlying image. To compute the error, each 3D model edge is projected into the color image and a gradient mask tuned to the precise expected orientation is applied to the pixels lying under the line. The gradient response, $\hat{G}_{Line}(k)$, for each line k is normalized to the range $[0,1]$. The derivation of the response is presented in [9].

A threshold (v) is used to discard lines with weak responses, and the gradient is inverted for each line:

$$E_{Line}(k) = \begin{cases} (1-\hat{G}_{Line}(k)) & \hat{G}_{Line}(k) > v \\ 1(omitted) & otherwise \end{cases} \quad (4)$$

The fitness error is then formed by summing the error terms, and normalizing by the number of lines l not discarded (omitted lines):

$$E_{fit,o}(\mathcal{F}) = \frac{1}{l} \sum_{k \in Lines} E_{Line}(k) \quad (5)$$

3.2.2 Range Fitness Error: $E_{fit,r}(\mathcal{F})$

The range fitness error represents how well the predicted 3D sampled surface model points fit the actual range data. Let (χ) be the set of all predicted model points and (ψ) the set of range data points within a bounding box defined relative to (χ). Fitness is a function of the Euclidean distance between model points $i \in \chi$ and data points $j \in \psi$. The distance is measured with the model points (χ) placed relative to the data using the current coregistration estimate $\mathcal{F}$:

$$\overline{D}(i,j)_{i\in\chi, j\in\psi} = ||i-j|| \quad (6)$$

For each model point i, H_{point} is the distance to the nearest neighbor:

$$H_{point}(i) = \overline{D}(i,j) \quad : \forall k \in \psi \, \overline{D}(i,j) \leq \overline{D}(i,k) \quad (7)$$

and Fitness depends upon these distances

$$E_{point}(i) = \begin{cases} H_{point}(i) & H_{point}(i) < \tau \\ 0 & otherwise \end{cases} \quad (8)$$

The threshold (τ) places an upper bound on the distance between matching features, and is set to discard points considered too far away to match. The total fitness is summed over the matched points and normalized to lie in the interval $[0,1]$. Normalization takes into account of the number of matched points p and the maximum allowable distance τ:

$$E_{fit,r}(\mathcal{F}) = \frac{1}{p \cdot \tau} \sum_{i \in \chi} E_{point}(i) \quad (9)$$

3.2.3 Omission Error: $E_{om,\mathcal{S}}(\mathcal{F})$

Omission accounts for weak responses in optical and unmatched points in range. Omission prevents fixation upon very small numbers of strongly matched features by introducing a bias in favor of accounting for as many

model features as possible. The general form of the omission error is:

$$E_{om,o}(\mathcal{F}) = \begin{cases} \frac{e^{\alpha \cdot w}-1}{e^{\alpha}-1} & \alpha \neq 0 \\ w & \alpha = 0 \end{cases} \quad (10)$$

where w is the ratio of unmatched model features over the total number of model features. The parameter α introduces a non-linear bias which reduces the penalty of small amounts of omission while increasing the penalty for large amounts of omission [16]. For the optical omission error, $E_{om,o}(\mathcal{F})$, w is the number of unmatched lines over the total number of lines.

For the range data, omission is measured in both directions: model-to-data and data-to-model:

$$E_{om,r}(\mathcal{F}) = \begin{cases} \frac{1}{2} \cdot (\frac{e^{\alpha p}-1}{e^{\alpha}-1} + \frac{e^{\alpha q}-1}{e^{\alpha}-1}) & \alpha \neq 0 \\ \frac{p+q}{2} & \alpha = 0 \end{cases} \quad (11)$$

where p is the ratio of unmatched model points over the total number of model points, and q is the number of unmatched data points (those lying above the threshold τ) over the total number of data points.

4 Results

Figures 1 and 2 show both the initial starting conditions and final matches obtained using our combined feature prediction and matching algorithms. Figures 1a, 1b, 1c show the initial pose estimate in each sensor coordinate system for shot 20. Figures 1d, 1e, 1f show the results of the local search algorithm. As can be seen, the algorithm corrects for a substantial amount of error in the initial estimate. It took the search algorithm approximately 30 moves to come to the solution shown. On the order of 1500 evaluations of the error function were performed. Figure 2 shows the results of the algorithm applied to Shot 35. Again, the algorithm was able to account for a substantial amount of error in the initial estimate.

References

[1] J. R. Beveridge, D. P. Panda, and T. Yachik. November 1993 Fort Carson RSTA Data Collection Final Report. Technical Report CSS-94-118, Colorado State University, Fort Collins, CO, January 1994.

[2] C. H. Chien and J. K. Aggarwal. Shape recognition from single silhouettes. In *International Conference on Computer Vision*, pages 481–490, 1987.

[3] A. Hoogs and D. Hackett. Model-supported exploitation as a framework for image understanding. In *Proceedings: Image Understanding Workshop*, pages 265–268. ARPA, nov 1994.

[4] K. Ikeuchi. Precompiling a geometrical model into an interpretation tree for object recognition in bin-picking tasks. In *Proc. DARPA Image Understanding Workshop*, pages 321–330, Feb. 1987.

[5] J. Ross Beveridge and Bruce A. Draper and Kris Siejko. Progress on Target and Terrain Recognition Research at Colorado State University. In *Proceedings: Image Understanding Workshop*, page (to appear), Los Altos, CA, February 1996. ARPA, Morgan Kaufman.

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

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

[8] C.-H. Liu and W.-H. Tsai. 3D curved object recognition from multiple 2d camera views. *Computer Vision, Graphics and Image Processing*, 50:177–187, 1990.

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

[10] D. Marr. Analysis of occluding contour. *Proceedings of the Royal Society of London*, B197:441–475, 1977.

[11] G. Paltridge and C. Platt. *Radiative Processes in Meteorology and Climatology*. Elsevier Scientific Publishing Company, 1976.

[12] H. Platinga and C. Dyer. Visibility, Occlusion, and the Aspect Graph. Technical Report 736, University of Wisconsin - Madison, December 1987.

[13] A. N. A. Schwickerath and J. R. 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.

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

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

[16] M. R. Stevens and J. R. Beveridge. Precise matching of 3D models to multisensor data. *IEEE Transactions on Image Processing*, page to appear, 1997.

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

[18] Y. F. Wang, M. J. Magee, and J. K. Aggarwal. Matching three-dimensional objects using silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:513–518, 1984.

[19] M. Wheeler and K. Ikeuchi. Sensor modeling, markov random fields, and robust localization for recognizing partially occluded objects. *IUW*, 93:811–818, 1993.