Using Multisensor Occlusion Reasoning in Object Recognition *

Category: Object Recognition and Pose Estimation

Abstract

An iterative predict and match system renders CAD models to refine predictions about detectable features in both optical and range sensor data. Predicted features adapt to a scene specific interpretation which includes a time-of-day lighting model and reasoning about unmodeled occluding surfaces. During rendering, links between the rendered image and the 3D CAD model are retained to propagate information between object model, scene and sensors coordinate reference frames. Precise geometric matching of objects occluded by as much as 60% is demonstrated.

1 Introduction

Unconstrained object recognition in outdoor scenes is a daunting task. While the human visual system is adept at recognizing a wide range of objects in such a scene, for a machine to perform reasonably well an explicit geometric model of each object being sought is usually required [23, 21]. When detailed CAD models of objects are available they can provide a rich source of constraint. Thus object recognition can be viewed as finding evidence in imagery for various model constraints. Usually model properties are chosen such that they can easily be found in the data: points [37], lines [6], curves [29], corners [3] or planes [19].

Perhaps one of the most bedeviling complications for model-based recognition arises when one object partially occludes another. Traditional techniques either rely on error measures being robust to such occlusions [27, 23, 6, 44], or associate a likelihood of finding each feature based on off-line appearance analysis [14, 45, 36, 22]. In the former case, matching can be hindered by the lack of support for an occluded feature and continue to examine possibilities even when the feature will never be found. In the latter case, it is problematic when the off-line analysis can not adequately capture the variability of complex scenes.

We are taking a different approach to the problem of occlusion. When multiple, heterogeneous sensors are viewing the same scene, it is possible that occlusion may be inferred from one sensor and propagated into another. For our problem, we use both 2D color imagery as well as 3D range data. Range data has the nice property of allowing the clear determination of when a certain model feature is not visible because another object is lying in front of it [33]. The use of multiple sensors facilitates the process of reasoning about why certain model features are not being found in the data, and allows an adaptive method for removing model features as they are determined to be occluded.

To successfully reason about occlusion, we use graphic rendering techniques to dynamically predict CAD features for matching. As the object recognition system gains information about the location of the object in the world, it modifies the predicted features to fit the current scenario. Thus, more realistic appearance-based models for a specific scene are generated from the 3D CAD model. Furthermore, reasoning about which features to generate takes place in the CAD models native 3D coordinate system. Graphics techniques and hardware allow the movement back and forth between the scene imagery and this coordinate system in near real-time.

2 Prior Work

Our work incorporates a feature prediction capability based upon rendering CAD models within an object recognition system. Hence it draws upon two literatures: object recognition for which there are many surveys [35, 16, 5] and CAD rendering [38, 18, 31]. Most CAD-based recognition systems focus upon a single sensor. For example, models have been used for matching to 2D imagery [15, 9, 42], 3D range data [2, 4], as well as multi-spectral imagery such as IR [34] and SAR [12]. Typically these CAD systems rely on either 3D or 2D model geometry to constrain object location and appearance [28, 23, 21, 6, 10].

The more complex task of fusing data from sensors of different modalities, for instance range and optical, has also been addressed [30, 39, 43, 17, 24]. However, this research area is still young. Fusing information from different modalities is complicated by many fac-

*This work was sponsored by the Defense Advanced Research Projects Agency (DARPA) Image Understanding Program under grants DAAH04-93-G-422 and DAAH04-95-1-0447, monitored by the U. S. Army Research Office, and the National Science Foundation under grants CDA-9422007 and IRI-9503366

tors, including recovery of sensor alignment and registration [1]. Our solution is to use the CAD model geometry to suggest the proper registration [7].

3 Overview

By combining basic techniques of computer graphics with those of computer vision, it is possible to build highly robust model-based object recognition algorithms which deal properly with occlusion. This paper is one of a series presenting different aspects of our work in multisensor object recognition using repeated rendering of 3D CAD models as part of an iterative predict and match cycle [40, 32, 25]. The emphasis here is on using evidence from range data to infer occlusion in range and optical images and adjusting predicted model features accordingly.

A critical aspect of our work is what we call *coregistration*. Introducing multiple sensors raises uncertainty in the pixel-to-pixel registration between sensors as well as the 3D pose of the object relative to the sensors. Consider data from two co-located sensors. Through calibration, an initial estimate of the pixel-to-pixel mapping can be developed. However, for nearly boresight aligned sensors, tiny rotations of one sensor relative to another can induce several pixel translations of one image relative to the other [26]. Therefore, to perform robustly despite such minor variations in image registration, an object recognition algorithm should refine sensor alignment as well as object pose. It is the combined process of determining pose and registration that we call coregistration.

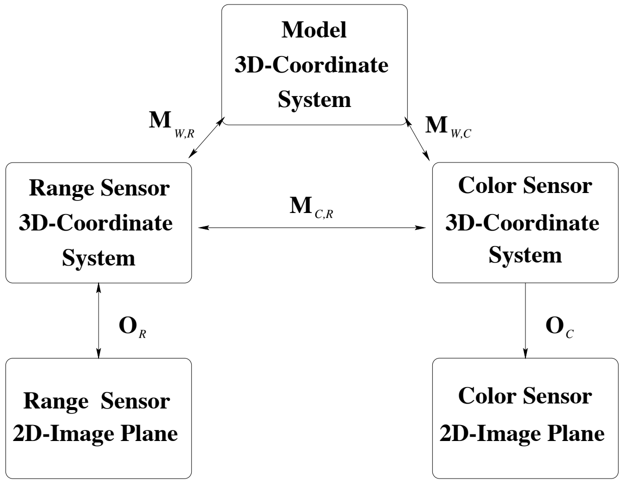

Figure 1: Coordinate Systems

In this paper, we will use data from a LADAR range sensor and a color camera. With two sensors there are eight degrees of freedom associated with coregistration: six encode the object pose relative to the sensor suite and two the relative translation of one imaging plane relative to the other. Figure 1 shows the coordinate systems and transformations associated with coregistration. Notice that the model provides a link between the two sensors transformations $M_{W,R}$, $M_{W,C}$ and that the two are not independent.

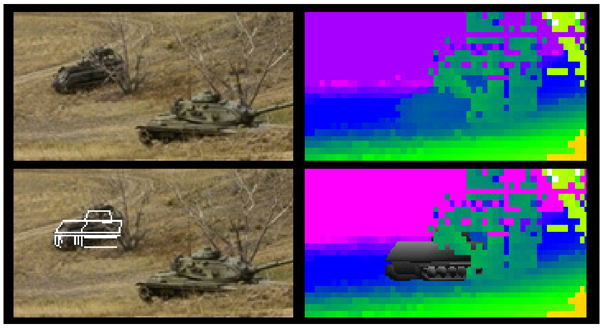

Figure 2: Example Data and Initial Hypothesis

Figure 2 presents an example recognition problem. The top row shows a portion of a color image and a portion of a range image covering roughly the same part of the scene. The data is from the Fort Carson Dataset, which contains range, IR and color imagery of military targets against natural terrain. The imagery and documentation [8] is available through our website at `http://www.cs.colostate.edu/~vision`. Sensor calibration data is available in [26].

The bottom row of Figure 2 shows an initial coregistration hypothesis for a specific CAD model. The hypothesized relationship between the model and the color image is shown as white lines in the color image. For the range data, a gray-scale rendering of the CAD model is embedded within a pseudo-colored rendering of the range image. This initial hypothesis has been automatically generated by two upstream algorithms described elsewhere. The first algorithm detects vehicles based on their color [13], and the second generates a vehicle and pose estimate based on the appearance of the occluding contour in a range image [11]. We will use this example to illustrate how our iterative predict and match system is able to refine the coregistration estimate while detecting and accounting for the partial occlusion of the vehicle by the tree.

The predict and match cycle at the heart of our coregistration matching system is illustrated in Figure 3. The process is initiated with a hypothesized model and a set of coregistration parameters (as already illustrated in Figure 2). Rendering of the object is used to predict features expected to be visible in the range and color imagery and then these features are

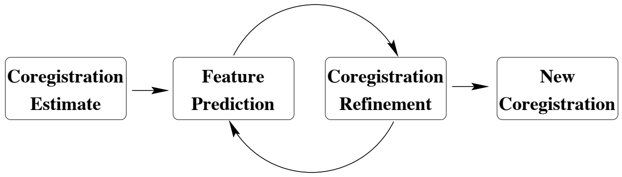

Figure 3: Interleaving Prediction and Coregistration

perturbed to test possible refinements of the vehicle-to-sensor coregistration. If improvements are found, the cycle repeats with new predicted features.

The rest of this paper reviews the feature prediction process, emphasizing the original contributions on occlusion reasoning. The optimization procedure used to find a best coregistration estimate and associated match is described and results on several images with significant amounts of occlusion are presented. These results demonstrate how our optimization system converges upon an interpretation properly aligning CAD models to imagery as occluded portions of the model are obscured by foreground objects.

4 Multisensor Feature Prediction

We use on-line rendering techniques to dynamically alter the set of model features used for matching. Previously we have developed a CAD rendering package [41], as well as an iterative generated and test algorithm which uses the predicted features and an error function to accomplish multi-sensor object recognition [32, 40]. Here we expand on the use of rendering as a tool for coping with occlusion.

To reason about occlusion, there must be information available which suggests occlusion is present. There are two direct sources of such information: a detailed site-model from which occlusion can be inferred, or data from a sensor which indicates relative depth of surfaces. We take up the latter case, using range data to explain why certain model features will not be visible based on evidence of occluding surfaces. The process of predicting what features of a model will be visible is described in the following four sections.

4.1 Linking Model and Sensor Data

The orientation component of the current coregistration hypothesis is used to render an image of the CAD model. When the rendering is performed, each face in the model is assigned a unique color. After the model is rendered, the color of the pixel provides an index into the 3D model, and the face inducing that pixel can easily be determined. Figure 4 shows the initial rendering created for the hypothesized vehicle in Figure 2. Notice each face in the model has a unique color distinct from the background: colors are links which allow visibility information to be propagated between imagery and model.

To produce object model features that may be matched to range data, the visible faces are transformed into the range sensor coordinate system. Then a ray casting algorithm mimics the geometry of the range sensor in order to generate a set of sampled surface points. These points also link 3D model faces to range data and provide a basis for inferring occlusion.

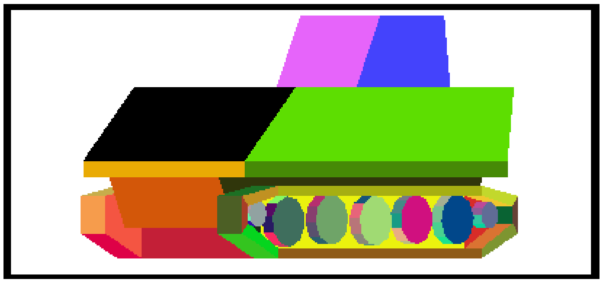

Figure 4: Initial Rendering

4.2 Mapping Model to Range Data

The next step is to examine how the ray traced sampled surface points from the model relate to the actual range data. A constrained nearest neighbor algorithm is used to find the data point lying closest to each model point. The constraint imposed is that the 3D data point, must be adjacent to the projected model point in the range image plane. This allows the nearest neighbor search to be performed in $O(n)$ time as opposed to the unconstrained nearest neighbor algorithm which takes $O(nm)$ where n is the number of model points and m is the number of data points.

Once the nearest neighbor is found it can be given one of three labels. The first label is matched. A matched point means the range data nearest neighbor lies within some threshold [1] distance of the model point. The second label is occluded. For this label to be chosen, the nearest range data point must lie the threshold distance in front of the model point. The final label is omitted. This label is given if neither of the other two conditions were meet, and accounts for unexplained outliers not found due to sensor noise.

Figure 5 shows the sampled surface generated for the coregistration estimate, and the label given to each point. The left image shows the visible faces transformed into the range coordinate system. The right image shows the sampled surface with the following color coding: black for occluded points, green for matched and blue for omitted.

[1] 2 meters in this work

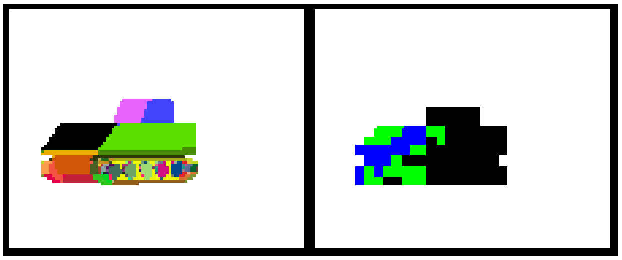

Figure 5: Model Sampled Surface

4.3 Inferring Occlusion

The 3D sampled range points can be transformed back into the 3D model coordinate system using the inverse transformation of the current coregistration estimate. Using these points in 3D model coordinates, and their associated label, it is possible to adapt the initial rendering to reflect the current state of occlusion in the scene.

The process begins by projecting each 3D occluded point onto the 2D image plane of the rendered image. Remember from Figure 1 that transformations between coordinate systems are invertible, and therefore the inverse of the transformation moving the model faces into the range coordinate system can be used to move range points into the model coordinates. We assume that the occlusion in the scene is induced by some object also lying on the ground plane. This assumption allows us to use a simple fill operation to replace all pixels below the current occluded point, in model coordinates, to the bottom of the initial rendered image of the model with the unique background color. Due to the resolution of the range sensor, each projected point covers a certain area on the model. Thus the fill is actually a rectangle based on the estimated area covered by the given range point.

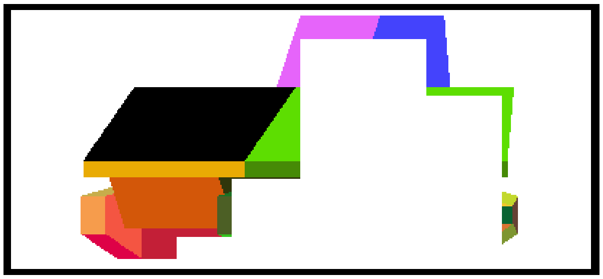

Figure 6: Updated rendering for scene occlusion

Figure 6 shows the result of modifying the rendered model image to reflect the current knowledge about occlusion in the scene. It shows the system has adapted the rendering to reflect where occlusion by the tree has been detected. From the new rendered image, visible edges can be obtained for matching to the color image.

4.4 Predicting Model Edges

Two types of model edges are predicted for matching to the color imagery: any edges lying on the occluding contour (the model silhouette), and any edges likely to be visible due to illumination conditions in the scene. Furthermore, since the rendered image will be used to infer these edges, the set of edges used for matching will be clipped to only the portion expected to be visible and not occluded.

The edge selection process begins with analysis of the visible set of faces that have been projected into the range sensor and ray traced. That list is pruned to remove any faces that do not have at least one ray intersection point which was matched to a data point. Each of the remaining visible faces are labeled as either lying on the silhouette or illuminated. To be on the silhouette, one of the pixels in the rendered image of the model belonging to that face must have an eight-connected neighbor which is the unique background color. To be illuminated, the face must be visible to the main source of illumination: the sun.

For our models, each edge can only be shared by two faces. Edges are discarded unless either one face is illuminated and the other is not visible, or one face is on the silhouette and the other is not visible. Once the subset of edges meeting these two criteria is derived, each edge is clipped to account for self-occlusion with other parts of the same object. The clipping process projects each edge likely to be matched into the image plane. The line is then traversed from beginning to end, and the portions of the edge for which the correct face color is present in the image are retained. Because orthographic rendering is used to generate the image, the clipped endpoints can be related back to the 3D model endpoints. This allows the derivation of 3D model edges for matching to the 2D color imagery. Details of clipping have been presented elsewhere [41].

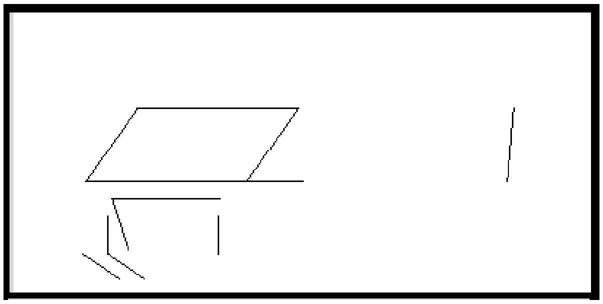

Figure 7: Set of clipped 3D model edge segments

Figure 7 shows the set of edge segments derived from the current coregistration estimate. Remember that these segments are actually available in the 3D coordinates of the model. Also notice that initially most of the segments to be matched lie on the front portion of the model. In addition, the internal features of the model have been clipped to the areas where occlusion is most likely to not be present.

5 Coregistration Matching

The matching algorithm uses a variant of tabu search [20], the set of features predicted for matching, and a match error to refine the initial coregistration estimate. The search process, as outlined in Figure 3, is a simple generate and test algorithm. After the features are predicted, a neighborhood of moves about the current point in coregistration space is defined.

This neighborhood is defined over the eight DOF of the coregistration-space. These parameters are: a rotation about the x axis, a rotation about the y axis, a rotation about z, a translation in the z axis (depth), two translations in x and y for the color coordinates and two translations in x and y for the range coordinates. Line search is used to generate perturbations of the current estimate from coarse-to-fine sampling in each dimension in order to reduce the algorithms sensitivity to step sizes.

For each discrete perturbation, the match error is evaluated. This match error, formally defined in [32], is summarized here. The match error, $E_{\mathcal{M}}$, is the weighted sum of errors from each of the two sensors:

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

The argument, $\mathcal{F}$, represents the coregistration of the sensors relative to the model: $\mathcal{F} \in \Re^8$. The error terms for range and color data are $E_{\mathcal{M},c}(\mathcal{F})$ and $E_{\mathcal{M},r}(\mathcal{F})$ respectively. Their relative importance is controlled by the weight $\alpha_{\mathcal{M}}$. For all experiments $\alpha_{\mathcal{M}} = 0.4$.

The color error term is further broken down into a fitness, $E_{fit,c}$ term and an omission, $E_{om,c}$, term. The fitness term is the sum of the gradient under each of the 3D predicted model edges as they are projected into the 2D color image using the current perturbation. If the response for an edge is weak, as set by a threshold, the line is said to be omitted and does not contribute to the fitness sum. Thus the fitness term is the average gradient response for only the strongly matched edges. The omission is just the ratio of unmatched features to the total number predicted [6]. The resulting error is:

$$E_{\mathcal{M},c}(\mathcal{F}) = \beta_c E_{fit,c}(\mathcal{F}) + (1 - \beta_c) E_{om,c}(\mathcal{F}) \quad (2)$$

for the color, with the weighting term, β_c, determined empirically to also be 0.4.

The range error measures how well the sampled surface information fits the range data, can also be broken down to:

$$\begin{aligned} E_{\mathcal{M},r}(\mathcal{F}) &= \beta_r E_{fit,r}(\mathcal{F}) \quad (3) \\ &+ (1 - \beta_r) max(E_{om,r}(\mathcal{F}), E_{oc,r}(\mathcal{F})) \end{aligned}$$

where the weighting term, β_r, was also determined empirically to be 0.4. The labeling scheme described in Section 4.2 is used to find the nearest constrained neighbor to a given model point after being transformed by the current perturbation. The fitness is the average constrained Euclidean distance between the model points and the nearest neighbor data points that are labeled matched. The omission is the ratio of points labeled omitted to those predicted but not occluded.

The occlusion term is based on the ratio of unmatched points to those predicted and not omitted:

$$E_{oc,r}(\mathcal{F}) = \begin{cases} 0 & \text{if } r \leq 0.3 \\ (r - 0.3)/0.7 & \text{if } 0.3 < r < 0.7 \\ 1 & \text{if } r \geq 0.7 \end{cases} \quad (4)$$

where the r represents the ratio of occluded model points to those predicted to be visible and not omitted. Initial experiments showed the necessity of using the occlusion term as it was not enough to simply remove the features from the match that were believed to be occluded. The matching system quickly discovered the benefit of moving vehicles completely behind a hillside, thus occluding all of the features, in order to send the error measure to zero.

As the error term for each discrete neighborhood is examined, the best move, or position in coregistration space with the lowest error, is retained. The best move then replaces the current one, and the features are regenerated. The process continues until no moves are considered better than the current one. At this point the current position is returned as the best coregistration for the initial estimate.

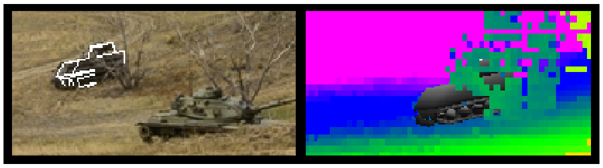

Figure 8: Results for Shot 15 Array 4

Figure 8 shows the solution converged to after 20 iterations through the generate-and-test tabu search process. As can be seen, the system was able to correct for the rotation and translation errors present in the initial coregistration estimate.

6 Additional Examples

We have run the matching system just described on 35 images from the Fort Carson dataset. These experiments are described in [25]. On these 35 cases, the system correctly identifies the vehicle present (out of four possible vehicles) in roughly 80 percent of the cases. While the system performed adequately on occluded targets, properly identifying about half, it often did not correctly recover the true object pose.

The system enhanced with the occlusion reasoning capability described above has been re-run on the imagery containing occluded targets. In four out of the six cases, the modified system was able to correctly recover the object pose with far more accuracy than the previous system [2]. The two cases for which the new system did not increase performance contained approximately 70 to 80 percent occlusion: this is more occlusion than we currently permit based upon the occlusion term defined equation 4.

Figures 9, 10, and 11 show the results of the occlusion reasoning system as applied to three of the four successful images (the final of the four was shown in the previous section). Each figure contains several images. The top row of each figure shows the two data images. The next row shows the initial estimates provided to the system, and the final row shows the solution to which the system converged. The first column represents the color sensor, the second the range sensor, and the third shows the model alone to get a better feel for the orientation.

As can be seen, the imagery are of varying difficulty. The most important information is contained in the first column of each figure, where the color lines used for matching have been changed significantly during the search process. These examples illustrate the system successfully adapting the feature set used for matching in response to occlusion cues from the range data.

7 Conclusions and Future Work

The current occlusion reasoning system performs well for up to 60% occlusion. Given enough pixels remain in view, we hope to extend performance to higher levels of occlusion. We also hope to strengthen the system's ability to cope with the tendency of the occluding contour to induce a strong edge that misleads matching. Either this edge could be suppressed or introduced as a new feature to match.

[2] While we've not quantified this improvement, it is immediately evident through visual inspection using our our on-line 3D visualization system

References

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

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

[3] Benjamin Bell and L. F. Pau. Contour tracking and corner detection in a logic programming environment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):913–917, September 1990.

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

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

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

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

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

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

[10] 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).

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

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

[13] 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.

[14] Octavia I. Camps, Linda Shapiro, and Robert Haralick. *Three Dimensional Object Recognition Systems*, chapter Image Prediction for Computer Vision. In Three-Dimensional Object Recognition Systems. Elsevier Science Publishers, 1993.

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

[16] Roland T. Chin and Charles R. Dyer. Model-based recognition in robot vision. *ACM Computing Surveys*, 18(1):67–108, March 1986.

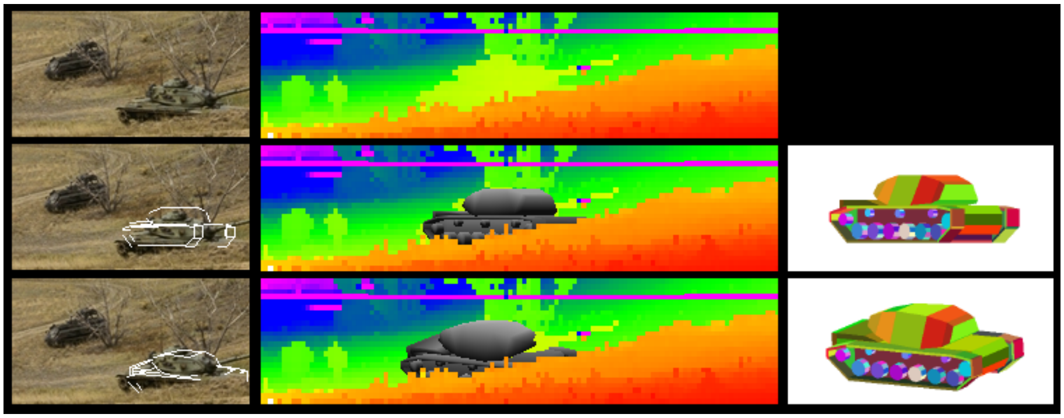

Figure 9: Results for Shot 14 Array 4

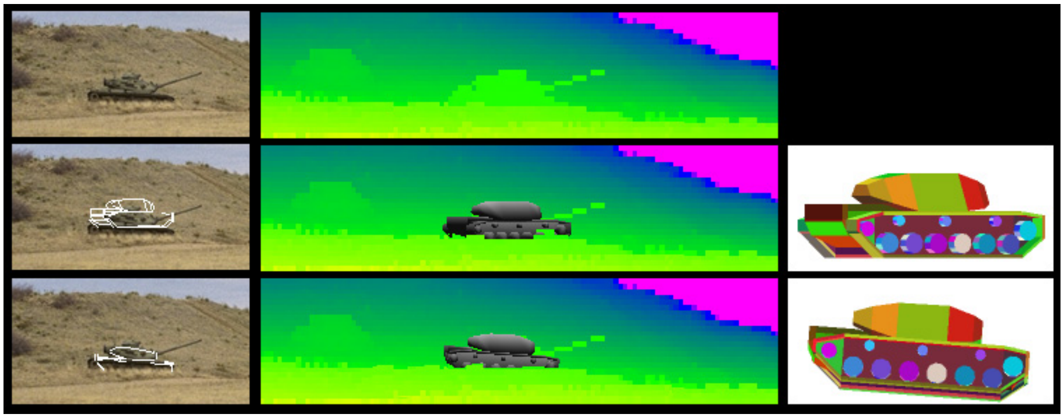

Figure 10: Results for Shot 29 Array 8

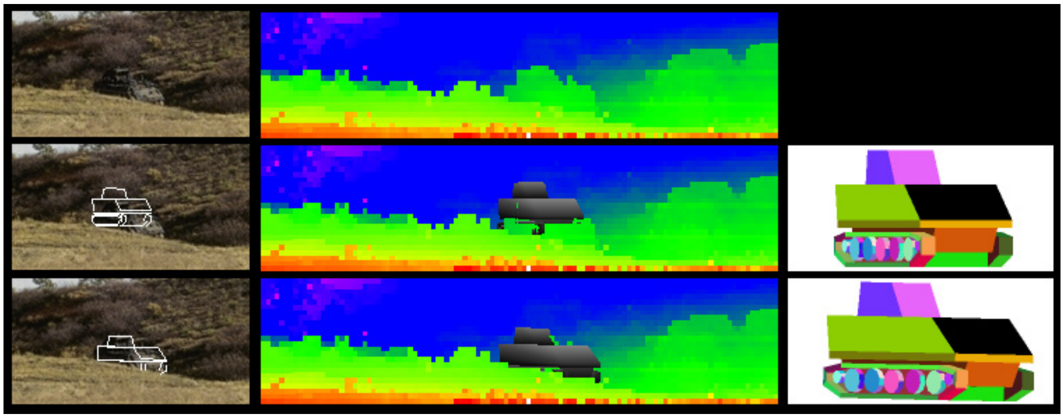

Figure 11: Results for Shot 34 Array 9

[17] 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.

[18] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley Publishing Company, New York, 1990.

[19] P. C. Gaston and T. Lozano-Pérez. Tactile recognition and localization using object models: The case of polyhedra on a plane. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI – 6:721 – 741, May 1984.

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

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

[22] Anthony Hoogs and Ruzena Bajcsy. Model-based learning of segmentations. In *International Conference on Pattern Recognition*, volume 4, pages 494–499, Vienna, August 1996. AIPR, IEEE.

[23] 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.

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

[25] J. Ross Beveridge and Mark R. Stevens. CAD-based Target Identification in Range, IR and Color Imagery Using On-Line Rendering and Feature Prediction. Technical report, Computer Science, Colorado State University, 1996. Submitted to special issue on CAD-based vision.

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

[27] David G. Lowe. The Viewpoint Consistency Constraint. *International Journal of Computer Vision*, 1(1):58 – 72, 1987.

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

[29] 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.

[30] 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.

[31] Martti Mantyla. *An Introduction to Solid Modeling*. Computer Science Press, 1990.

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

[33] Alan M. McIvor. Segmentation of 3d surface data. In *In Image and Vision Computing New Zealand*, pages 79–84, Lower Hutt, August 1996.

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

[35] Arthur R. Pope. Model-Based Object Recognition. Technical report, University of British Columbia, January 1994.

[36] Arthur R. Pope. *Learning to Recognize Objects in Images: Acquiring and Using Probablistic Models of Appearance*. PhD thesis, University of British Columbia, 1995.

[37] Sanjay Ranade and Azriel Rosenfeld. Point pattern matching by relaxation. *Pattern Recognition*, 12:269 – 276, 1980.

[38] Scott D. Roth. Ray casting for modeling solids. *IEEE Computer Graphics and Image Processing*, 18:109–144, 1982.

[39] 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.

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

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

[42] 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.

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

[44] William M. Wells. *Statistical Object Recognition*. PhD thesis, Massachusetts Institute of Technology, 1993.

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