Computer Science
Technical Report

Colorado
State
University

Visualizing Multisensor Model-Based Object Recognition [*]

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

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

Michael E. Goss
Hewlett-Packard Laboratories
goss@hpl.hp.com

March 28, 1997

Technical Report CS-97-106

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 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

Visualizing Multisensor Model-Based Object Recognition [*]

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

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

Michael E. Goss
Hewlett-Packard Laboratories
goss@hpl.hp.com

Abstract

A difficult problem when designing automatic object recognition algorithms is the visualization of relationships between sensor data and the internal models used by the recognition algorithms. In our particular case, we need to coregister color, thermal (infrared), and range imagery, to 3-D object models in an effort to determine object positions and orientations in three-space.

In this paper we describe a system for interactive visualization of the various spatial relationships between the heterogeneous data sources. This system is designed to be closely linked to the object recognition software such that it allows detailed monitoring of each step in the recognition process. We employ several novel techniques for visualizing the output from an imaging range device. Our system also incorporates sensor models which can simulate sensor data for visible features of stored object models, and display these features in the proper position relative to the appropriate sensor.

1 Introduction

When multiple sensors view a common scene, it is extremely helpful to fuse the information into a single common interactive display. We have developed a visualization system which integrates information from multiple sensors along with 3-D CAD object models for use in model-based object recognition. This tool enables us to readily uncover and understand subtle relationships between our stored object models and sensor data. Three specific sensors used are: a color camera, a Forward-Looking Infrared (FLIR) camera, and a range imaging laser RADAR (LADAR).

This visualization system has become an invaluable tool for data exploration and understanding. In addition, the interface illustrates the importance of coupling visualization with object recognition algorithms. As recognition algorithms improve in capability and scope, the interpretations they produce include increasingly rich and detailed information about relationships between stored object models and the scenes being viewed. The information present in these interpretations is most compelling when presented to a person using interactive 3-D visualization. In addition, the system allows a user to better estimate ground truth for the data. Since ground truth is often hard to determine at the time of data collection, some method of establishing a baseline is imperative.

Each of the three sensors we use provides qualitatively different information about the nature of the world being viewed. Color imagery provides visual cues in the form to which humans are accustomed. Infrared provides thermal information, a useful sense missing from our human repertoire. Finally, range data directly reveals part of the 3-D structure of the scene. Due to the varying nature of the sensor information, the visualization system is responsible for showing how the different pieces of sensor data relate to one another. The problem becomes compounded when different levels of resolution and detail must be overlaid. While 3-D visualization might be helpful for working with CAD

[*] 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

models even when range data is not available, the ability to visualize 3-D relationships in the presence of such data is paramount.

Our efforts began in 1993 when we collected roughly 400 color, IR and range images[5], available through our web site [1]. It was immediately apparent that viewing these images with separate, unrelated 2-D displays was woefully inadequate. A means was needed to visually fuse the imagery while simultaneously bringing to life the 3-D character of the range data. This need led us to build our first visualization tool: *RangeView* [11; 10]. RangeView had limited capacity for embedding and manipulating 3-D CAD object models in the range sensor's native 3-D coordinate system. However, the choice of this coordinate system as the master reference frame limited RangeView, and subsequently a newer more flexible system was built. This system, called *ModelView*, allows greater integration of object models with sensor data. This paper describes the ModelView system, and the improvements made over our earlier work.

2 Motivation

The Computer Vision Group at Colorado State University is developing new object recognition algorithms to identify military vehicles in multisensor imagery. The research is part of the Unmanned Ground Vehicle (UGV) Program, a joint project of the Advanced Research Project Agency (ARPA) and the Office of the Secretary of Defense. For the UGV project, four Semi-autonomous Surrogate Vehicles (SSVs) have been designed to survey surrounding terrain and locate possible targets.

To achieve these target recognition goals, each vehicle has been designed to carry three sensors (color, FLIR, LADAR) mounted upon a single pan-tilt platform. The ModelView visualization system has been built to assist us in understanding this data and its relationship to stored 3-D vehicle models. ModelView system also serves as a prototype for how raw data from an autonomous vehicle might be fused and displayed to a remote operator.

Many of the visualization concepts illustrated in ModelView are applicable to a wide variety of application domains. Numerous sensor modalities, including stereo [3; 6; 26], Scanning and Scannerless LADARs, and IFSAR [40] produce range imagery. Whenever range data is integrated with stored 3-D object and terrain models, 3-D visual feedback is important. In addition, when separate range and optical sensors view the same scene, the multisensor visualization aspects illustrated by ModelView are essential.

Figure 1 (see color plates) shows a sampling of views from our system, as well as providing a visual overview of the data, object models, and relationships of interest in our work. The vehicle in the image is roughly 50 meters from each sensor, and is almost centered in the scene. This particular image is not a representative of the types of images the ATR algorithm is designed to deal with, but the large number of on-target pixels is useful for illustration of the visualization. For the Fort Carson data, a more typical range to vehicle would be between 100 and 150 meters from the sensor. The close ranges are partially due to limits of the prototype LADAR sensor. Wide-angle lenses were used on the other two sensors to approximate the performance of a modern vehicle sensor suite operating at 1km.

Starting in the left column of Figure 1, we see an example color image followed by an enlarged portion with the model overlaid. The right column shows the FLIR image, and its corresponding overlay. The top of the center column shows a standard view of a range image and the next lower frame shows the CAD model rendered in the scene. The bottom two center images contain different views of the scene with the range data re-projected from a different viewpoint and optical imagery texture mapped onto the range data. This form of texture mapping allows the relationships between the sensor images to be observed.

The remainder of the paper is broken into sections describing the sensor model, the CAD models used in the system, the coordinate systems used, and the methods used to visualize the sensor data relationships.

3 The Sources of Information

3.1 Sensor Models

A simple pinhole camera model is used to describe the 3-D world-to-image plane mapping for the range, IR and color sensors. Hence the mapping from a point in the sensor coordinate system, $P_{\mathcal{A}}$, to an image point, $p_{\mathcal{A}}$, for a sensor $\mathcal{A}$ may be written in homogeneous coordinates

[1] http://www.cs.colostate.edu/~vision

Table 1: Intrinsic Sensor Parameters for Fort Carson Color, FLIR and LADAR

	FOV $f_{u,\mathcal{A}}$		FOV $f_{v,\mathcal{A}}$		Dimensions		Scale		Center	
Sensor	Rad.	Deg.	Rad.	Deg.	$d_{u,\mathcal{A}}$	$d_{v,\mathcal{A}}$	$s_{u,\mathcal{A}}$	$s_{v,\mathcal{A}}$	$t_{u,\mathcal{A}}$	$t_{v,\mathcal{A}}$
Color	0.705	40.4	0.496	28.4	720	480	978	947	345.0	227.8
FLIR	0.434	24.9	0.401	23.0	256	256	580	630	127.5	127.5
range	0.271	15.5	0.060	3.4	120	24	440	400	59.5	11.5

as:

$$p_{\mathcal{A}} = O_{\mathcal{A}} P_{\mathcal{A}} = \begin{vmatrix} s_{u,\mathcal{A}} & 0 & t_{u,\mathcal{A}} \\ 0 & s_{v,\mathcal{A}} & t_{v,\mathcal{A}} \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} X \\ Y \\ Z \end{vmatrix} \qquad (1)$$

where $O_{\mathcal{A}}$ is the perspective projection matrix and $\mathcal{A}$ is either a point from a color, $\mathcal{C}$, FLIR, $\mathcal{F}$, or LADAR, $\mathcal{L}$, sensor ($\mathcal{A} \in \{\mathcal{C}, \mathcal{F}, \mathcal{L}\}$). Converting $p_{\mathcal{A}}$ into normalized 2-D homogeneous coordinates effects the perspective transformation:

$$p_{\mathcal{A}} = \begin{vmatrix} s_{u,\mathcal{A}} \frac{X}{Z} + t_{u,\mathcal{A}} \\ s_{v,\mathcal{A}} \frac{Y}{Z} + t_{v,\mathcal{A}} \\ 1 \end{vmatrix} \qquad (2)$$

The parameters $s_{u,\mathcal{A}}$ and $s_{v,\mathcal{A}}$ are typically called the scale factors for the sensor and the parameters $t_{u,\mathcal{A}}$ and $t_{v,\mathcal{A}}$ define the point of intersection between the optical axis and the image plane. Together, these four values define the intrinsic parameters of the sensor, $\mathcal{A}$.

3.2 Intrinsic Sensor Parameters

The intrinsic parameters, as well as the field of view and image dimensions, for the three sensors whose imagery is presented in this paper are indicated in Table 1. For the color sensor, the intrinsic parameters have been recovered using a known calibration target. For the IR sensor, the intrinsic parameters were estimated based upon the manufacturers specifications for the sensor and subsequently refined using the ModelView tool (see Section 5.3 for details).

Finally, the range sensor parameters were recovered from calibrated imagery. The maximum range measured by the LADAR is 1074 feet, and hence multiplying a raw pixel value by the ratio 1074/4095 yields a range measurement in feet. The standard deviation of the range measurement is approximately 1 foot [4]. Additional details on how the sensors were calibrated, the relationships between parameters, and the justification for the use of the perspective projection mapping for the range sensor can be found in [14].

3.3 CAD Object Models

Highly detailed models of the vehicles in our Fort Carson dataset exist in the model format known as BRL/CAD [36]. Algorithms to reduce the model complexity to a level more closely related to the sensor granularity have already been developed [33]. From these simpler models, features to be used in the matching process can be predicted.

Visible feature prediction is essential due to the combinatorially explosive nature of model based recognition [12]. The naive approach to model matching would be to determine a match error between all possible combinations of model and data features, and then choose the smallest error set as the optimal object match. However in a system where only 10 model features are being compared against 10 data features, there exist 2^{100} possible combinations. Being able to pick a small set of model features for matching, let us say 5, can greatly simplify the the correspondence space (2^{50}).

Common approaches to model feature generation have centered around an off-line model analysis in which visible features are determined for all viewpoints [27]. The results are then grouped into regions of constant topology [16] and stored in an aspect graph representation [15]. The aspect graph is used at runtime to obtain the list of visible features for a given pose [31]. For these systems, visualization is mainly centered around showing which particular model features are chosen, rather than relating them to any particular sensor image.

Our current feature prediction algorithms use graphics hardware to achieve real-time generation of relevant model features [33]. The feature prediction is fully integrated into the ModelView system and is central to the visualization paradigm. When visible model features are determined, they are rendered into each sensor image. This allows the user to assess not only how accurately the features were predicted, but also how well they fit the data.

Two sets of model features are generated based upon an estimate of an object's position and orientation. The first set represents the model silhouette and visible internal edges based on simple illumination constraints, the second set represents the 3-D sampled surface information for matching to the range data. The edges are defined by a set of 3-D endpoints stored in the BREP model. The sampled surface points are generated by ray tracing the model, using the range sensor geometry, for a given pose.

4 Related Work

Related work falls into three broad and distinct categories: fusing information provided by multiple sensors, work related to multisensor and object model visualization, and computer vision work on object recognition. The discussion is followed by our previous work on the RangeView system.

4.1 Sensor Fusion

Sensor fusion is the process of finding commonalities in heterogeneous sensors information. The goal is to provide a semi-autonomous agent with an internal representation of the surrounding world [23]. While the goals are very different from those of visualization, they need to solve many of the same problems associated with combining information from different sensor modalities.

Tong has used LADAR masks to segment FLIR imagery, resulting in simple sensor fusion [34; 35]. Unfortunately, the visualizations are rendered only as single 2-D images. Others have used neural networks to learn a mapping which relates two images [37; 38]. If the two images are analyzed as simple input signals, a network can be used to find a correlation between the two signals. In addition to neural networks, wavelets [17], edges extracted from the imagery [3], fuzzy logic [30], and image contours [18] have been used to determine the pixel-to-pixel correlation mapping between various types of imagery.

4.2 Sensor and Object Visualization

FLIR and color images can be effectively viewed using conventional image display techniques. More interesting is the case of imaging range sensor data which are conventionally displayed as either 2-D grey scale images, or as a 2-D overhead "scatter plots" [39]. For a grey scale representation, the grey level of each pixel corresponds to the distance of the sample from the sensor. Work has also been done in the display of range images as 3-D surfaces with illumination and shading, but this requires the prior extraction of surfaces from the raw range data [25; 29].

Previous systems [28; 39] which locate a 3-D target relative to a LADAR image typically render an image of the model in the image plane of the sensor. This 2-D image does not allow a complete understanding of how well the model has been located. With our verification system the 3-D model and sensor data can be interactively examined to assess the 3-D fidelity of the match. We have found the ability to arbitrarily change viewing parameters invaluable in the development of our object recognition algorithms.

Model-based visualization tools generally allow wire-frame representations to be interactively manipulated until the desired point of view is obtained. Ray tracing can then be used to render highly detailed images of this scene. In addition, terrain models and sensor imagery can be used to generate highly realistic scenes [24].

4.3 Model-Based Recognition

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. Using multiple sensors instead of a single sensor complicates the alignment task.

There are good examples of successful mixed-modality fusion [32; 7; 13], but this research area is still young. Aggarwal notes, and we agree, that to properly perform mixed-modality sensor fusion, coordinate transformations between images need to be adaptively determined [19]. Our recent work [22] emphasizes global alignment as a basis for optimal matching to multisensor data using local search.

The ModelView visualization system presented in this paper displays all the necessary relationships to visualize the sensor fusion process, as well as to interact with the model and data while the recognition is taking place. The local search process finds a sequence of successively better match hypotheses until one which is locally optimal is found. As researchers developing these algorithms, we find the ability to visualize each step of the local search process invaluable. The visualization not only aids in testing the matching code, but also provides the user with an understanding of how well the algorithm is operating, and whether or not it is converging upon a good solution.

4.4 RangeView

Our previous efforts at visualizing multisensor data within the model-based vision framework utilized a data-centered paradigm. The RangeView system [11; 10] contained several of the sensor-to-sensor relationships discussed in Section 6. The system was centered around the LADAR coordinate system, and allowed a user to interactively view the output of the color and FLIR images textured onto the range data.

The texture mapping used by RangeView was not as well developed as that presented here: RangeView averaged all the color pixels lying under a corresponding range pixel. The appearance was a smoothed color image on top of the LADAR. ModelView uses the known 3-D relationships of sensors to accurately back-project range data into the scene and then re-project this data into the color image plane. Consequently, the texture mapping accurately reflects the true underlying 3-D geometry of the scene.

RangeView also did not utilize sensor geometry when determining the correspondence between range and optical imagery. Instead, separate affine transformations were used to warp the optical pixels to the range data coordinates [9]. The 2-D affine mappings were determined by solving for the affine parameters which minimize the squared Euclidean distance between hand selected control points. The true mapping between sensors is often not affine and this affine mapping was only a coarse approximation of the true mapping. In previous work [14] we explore in considerable detail the conditions under which the mapping is affine and to what extent our mappings are valid.

RangeView was further limited by the lack of model-to-sensor relationships. The only technique for visualizing the object recognition results was to render the BRL/CAD models into the LADAR window. Then the texture mapped optical-to-range representation is used to determine how well the model fit the optical data. It was soon discovered that the model-to-color and model-to-FLIR relationships were essential for rating the matching system performance.

Finally, RangeView did not have any mechanisms for visualizing which model features were being used during the matching process. In fact, the feature prediction was completely decoupled from the visualization system. This made it difficult to develop the object recognition system because it was not easy to determine which model features were being matched to which data features. ModelView fully integrates the visualization of predicted features and their relation to the sensor data as the recognition system is converging to a solution.

5 Reference Frames

ModelView may be thought of as a tool for visualizing 3-D relationships between sensors and their relationships to the object models. While the sensors themselves are not iconically represented, changes in 3-D relationships between them are expressed through visual overlays of one type of sensor output on another. The following two sections address how to obtain the 3-D mappings between the sensors, and then how to visualize the results.

5.1 Sensor Placement

The positioning of the sensors used in the Fort Carson data collection mimics the setup which would be present on the UGV vehicle: recall Figure 1 (see color plates) and accompanying text introduced this dataset. The sensors were placed as physically close to each other as was practical and care was taken to make sure each sensor was level. This setup approximates boresight alignment of the sensors for objects viewed at a distance beyond approximately 45 meters.

From the given sensor placement, a series of constraints may be inferred for the relative 3-D mapping between sensors. These constraints are described in the following section and the underlying rationale and associated approximations are further justified in Section 5.4.

5.2 Relating Sensors and Objects

Consider transformations between three distinct 3-D coordinate reference frames: a world reference $\mathcal{W}$ and the three sensor reference frames $\mathcal{C}$ (color), $\mathcal{F}$ (FLIR), and $\mathcal{L}$ (LADAR).

The world reference frame origin and axes coincide with the model coordinate system. The model is centered at the origin, and the positive y axis points upward. The 3-D transformation of a point $P_{\mathcal{W}}$ in the world coordinates $\mathcal{W}$ to a point $P_{\mathcal{A}}$ in the reference frame of a sensor $\mathcal{A}$ may be defined as:

$$P_{\mathcal{A}} = M_{\mathcal{W},\mathcal{A}} P_{\mathcal{W}} \tag{3}$$

where $M_{\mathcal{W},\mathcal{A}} = T_{\mathcal{W},\mathcal{A}} T_{\mathcal{W}} R_{\mathcal{W}}$ and $\mathcal{A} \in \{\mathcal{C}, \mathcal{F}, \mathcal{L}\}$. All coordinate systems are measured in meters, so no scaling is needed to transform between coordinate systems.

The rotation $R_{\mathcal{W}}$ rotates points about the origin of the world (model) coordinate system in order to alter the relative orientation of the object model and the sensors. Note that the same rotation matrix is used for all sensor coordinate systems. This is consistent with an assumption that the sensors are mounted upon a common platform.

The transformations, $T_{\mathcal{W}}$ and $T_{\mathcal{W},\mathcal{A}}$, translate the points in the world relative to the sensors. The ModelView system distinguishes between translation in depth relative to the sensors and translation in a common XY image plane shared by all three sensors. The XY plane translations $T_{\mathcal{W},\mathcal{A}}$ are independent for each sensor. The depth, or Z axis translation $T_{\mathcal{W}}$, is the same for all three sensors.

Given these constraints, the world ($\mathcal{W}$) to sensor ($\mathcal{A}$) transformation may now be expanded from ($T_{\mathcal{W},\mathcal{A}}\ T_{\mathcal{W}}\ R_{\mathcal{W}}$), and $M_{\mathcal{W},\mathcal{A}} =$

$$\begin{vmatrix} 1 & 0 & 0 & T_{\mathcal{A},x} \\ 0 & 1 & 0 & T_{\mathcal{A},y} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} r_{xx} & r_{xy} & r_{xz} & 0 \\ r_{yx} & r_{yy} & r_{yz} & 0 \\ r_{zx} & r_{zy} & r_{zz} & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{4}$$

Using the constraints specified by Equation 4, the mapping between each sensor frame has two degrees of freedom; the mapping between any two sensors $\mathcal{A}$ and $\mathcal{B}$ may be written as:

$$M_{\mathcal{A},\mathcal{B}} = T_{\mathcal{A},\mathcal{B}} \tag{5}$$

This translation between two sensors may be expressed in terms of the translation of each sensor relative to the world.

$$T_{\mathcal{A},\mathcal{B}} = T_{\mathcal{W},\mathcal{A}} - T_{\mathcal{W},\mathcal{B}} = \begin{vmatrix} 1 & 0 & 0 & T_{\mathcal{A},x} - T_{\mathcal{B},x} \\ 0 & 1 & 0 & T_{\mathcal{A},y} - T_{\mathcal{B},y} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{6}$$

The three sensor-to-world transformations are not independent: knowing any two determines the third.

$$\begin{aligned} M_{\mathcal{W},\mathcal{C}} &= M_{\mathcal{L},\mathcal{C}} M_{\mathcal{W},\mathcal{L}} & M_{\mathcal{W},\mathcal{C}} &= M_{\mathcal{F},\mathcal{C}} M_{\mathcal{W},\mathcal{F}} \\ M_{\mathcal{W},\mathcal{F}} &= M_{\mathcal{L},\mathcal{F}} M_{\mathcal{W},\mathcal{L}} & M_{\mathcal{W},\mathcal{F}} &= M_{\mathcal{C},\mathcal{F}} M_{\mathcal{W},\mathcal{C}} \\ M_{\mathcal{W},\mathcal{L}} &= M_{\mathcal{C},\mathcal{L}} M_{\mathcal{W},\mathcal{C}} & M_{\mathcal{W},\mathcal{L}} &= M_{\mathcal{F},\mathcal{L}} M_{\mathcal{W},\mathcal{F}} \end{aligned} \tag{7}$$

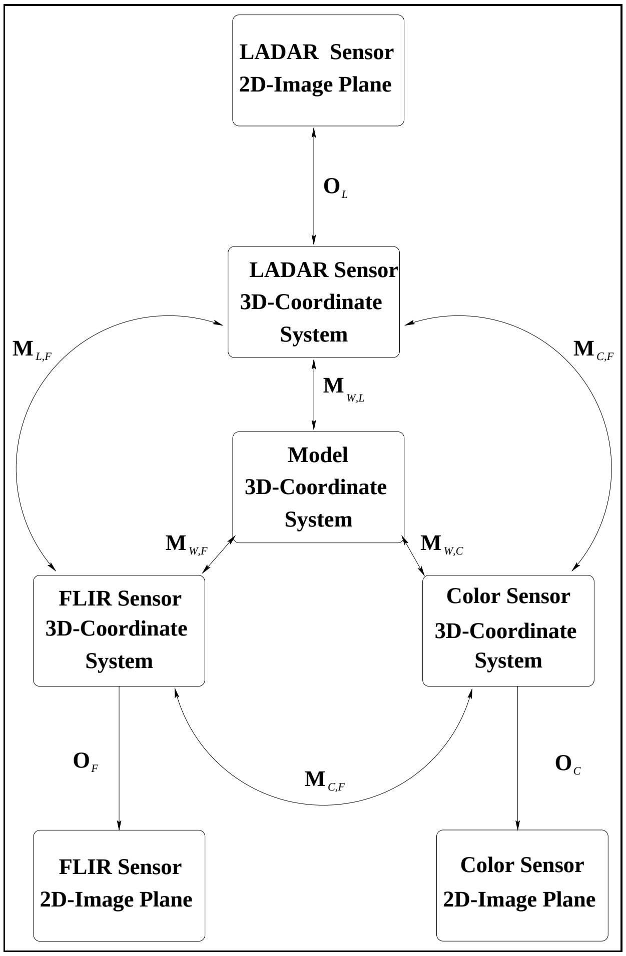

LADAR Sensor 2D-Image Plane
$\mathbf{O}_{L}$
LADAR Sensor 3D-Coordinate System
$\mathbf{M}_{L,F}$
$\mathbf{M}_{C,F}$
$\mathbf{M}_{W,L}$
Model 3D-Coordinate System
$\mathbf{M}_{W,F}$
$\mathbf{M}_{W,C}$
FLIR Sensor 3D-Coordinate System
Color Sensor 3D-Coordinate System
$\mathbf{O}_{F}$
$\mathbf{M}_{C,F}$
$\mathbf{O}_{C}$
FLIR Sensor 2D-Image Plane
Color Sensor 2D-Image Plane

Figure 2: Coordinate Systems of Model and Data Sets

Figure 2 graphically shows how the various reference frames relate via the given transformations. Each of the boxes represents a unique reference frame. At the center is the canonical representation: the model (world) coordinate system. Each arc in the graph represents a mapping from one system to the next. All of the M transformations are mappings between 3-D reference frames and are invertible.

The O mappings in Figure 2 indicate projection onto the associated sensor image plane. These transformations are of the form described in equation 1 and use the intrinsic parameters presented in Table 1. These projection transformations are of course not typically invertible. The exception is the LADAR mapping $O_{\mathcal{L}}$, which may be inverted due to the 3-D nature of the depth information.

ModelView supports a C++ object class of transformations which implements a network of coupled transformations. This network is constructed such that if any transformation is altered, any others effected are updated appropriately. For instance if the transformation $M_{\mathcal{W},\mathcal{C}}$ is modified, then the $M_{\mathcal{W},\mathcal{F}}$ and $M_{\mathcal{W},\mathcal{L}}$ are adjusted so the constraints between reference frames are not violated.

Within the boundary representation (BREP) [20] used for the CAD object models are structures used to encode the object in each of the major coordinate reference frames. The essential properties of the model representation capture the typical relationships needed by the model matching system (i.e., which (V)ertices bound an (E)dge, which edges bound a (F)ace, and which faces belong to a (S)urface). The vertex structure contains the 3-D model point in each of the various coordinate systems. Modifying an arc in Figure 2 causes a subsequent update to each vertex point to reflect changes in model position relative to each sensor.

5.3 Calibration Refinement

Once the model features have been predicted, they can be directly related to the data imagery using the camera calibration parameters. Often the camera calibration parameters may contain slight errors. Therefore it is useful to allow the user to interactively refine the initial sensor parameters in an effort to correct for slight inaccuracies. The left side of Figure 3 shows the initial color calibration parameters used to render the model features into the scene; these parameters cause the vehicle to be slightly too small. The right portion of the figure shows the result of interactive user modification of those parameters to reflect the actual imagery. The modifications are typically very small since the calibration parameters tend to be highly accurate.

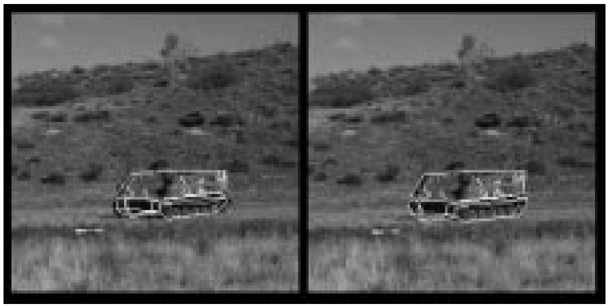


Figure 3: Before and after interactively adjusting the calibration parameters

5.4 Sensor-to-Sensor Constraints

In the Fort Carson data, what each sensor sees relative to another sensor depends upon minor differences in sensor placement, small differences in sensor pointing angles, different fields of view and different levels of resolution. In principle, this suggests the transformations $M_{\mathcal{W},\mathcal{A}}$ developed earlier ought to permit all possible relative motions between sensors. However, it not necessary nor desirable in our application to allow 6 degrees of freedom between sensors. This is because the sensors are approximately bore-sight aligned, and hence their relative placement is constrained.

It is worthwhile to consider briefly the implications of near bore-sight alignment. Perfect bore-sight alignment means that the transformations $M_{\mathcal{W},\mathcal{A}}$ and $M_{\mathcal{W},\mathcal{B}}$ are identical for two sensors $\mathcal{A}$ and $\mathcal{B}$. For simplicity, also assume each has the same intrinsic parameters so that image space coordinates should be identical for common 3-D points. Now, consider the following possible deviations from this initial configuration:

Rotation About the Optical Axis: Sensor $\mathcal{B}$ is rotated about Z, the optical axis, by an amount ϕ relative to sensor $\mathcal{A}$. If both sensors have equal horizontal and vertical scale factors, then the image plane for $\mathcal{B}$ is simply rotated by ϕ relative to the image plane for $\mathcal{A}$. However, given unequal horizontal and vertical scale factors, warping occurs and a full 6 degree of freedom 2-D affine transform is needed to represent the mapping between the two image planes.

Translation Along the Z axis: Sensor $\mathcal{B}$ translates ahead or behind sensor $\mathcal{A}$ along the common optical axis. Under the highly restricted case of viewing points all lying in a plane of constant Z, the scaling is constant for all points and the mapping reduces to a simple 2-D scale change. However, in general no single 2-D affine transformation can capture the mapping for all 3-D points.

Translation in a Common Image Plane: Sensor $\mathcal{B}$ translates in the common XY image plane. Under the restricted case of viewing points all at a common depth, this case reduces to simple 2-D translation of one image plane relative to the other. However, more generally the amount a 3-D point translates in one image plane relative to another is inversely proportional to distance from the sensors.

Rotation About the Horizontal and Vertical Axes: Rotate sensor $\mathcal{B}$ about the X and Y axes relative to sensor $\mathcal{A}$. Equivalently, this may be thought of as introducing uncertainty in the relative pan and tilt angles of one sensor relative to the other. In general, this case introduces a significant degree of non-linearity in the mapping between the two sensor image planes. However, under some conditions discussed below, this case may be approximated as translation in the common XY image plane.

Rotation about the optical axis does not appear to be a factor in the Fort Carson data. Movement of one sensor ahead or behind the other also does not appear to be a practical concern and is unlikely to be a factor in any data where the sensors are placed close to each other and are used to view distant objects. For small pan and tilt angles, common image plane translation is an adequate approximation to rotation about the X and Y axes. Consequently, the transformations in ModelView enforce a common rotation for all three sensors.

One way to quantitatively evaluate the adequacy of this simplifying assumption is to ask what happens if two sensor models are setup to track a common central 3-D point at a depth $D = 100$ meters. One sensor tracks the point by panning and the other by translating in the XY plane. Using the parameters for the color sensor, it is possible to determine the maximum deviation between pixel coordinates for identical 3-D points under some different assumptions. For example, consider first a sensor rotated by 1/5 of a degree and points lying within 1 meter of the tracking depth and near the tracking point on the image. In this case, the translating sensor approximates the mapping of the rotating sensor to within 0.06 pixels. Clearly this is more than adequate for 3-D points on a moderately sized object at 100 meters.

For points beyond the tracking depth, the pixel deviation is bounded. For example, given 1/5 of a degree of rotation, the maximum deviation between the rotated and translated sensor for points out to $1,000$ meters is less than 4 pixels. For the LADAR, which has a lower pixel resolution and smaller field of view, the comparable pixel deviation is about 1 pixel. A much more detailed development of near bore-sight aligned sensors and the use of image plane translation to approximate pan and tilt may be found in [14].

6 Visualizing the Model-to-Sensor Relationships

ModelView may be used to visualize different 3-D relationships between an object model and multisensor data. ModelView visualizes the effects rather than the relationships themselves because the system never actually presents icons of the sensors or their relative 3-D scene position. Instead, ModelView creates a synthetic visualization environment in which changes to these 3-D relationships manifest themselves as changes in a combined display of model and sensor data. For the color and FLIR sensors, this is done with two different displays. One maps 3-D object model features into the sensor image space and the other maps the image data onto the 3-D object model. For the range sensor, both range and object model features are displayed in a common 3-D interactive viewer.

6.1 Relating Model to Optical

From the standpoint of visualizing the relationship between a sensor and the model, the needs and methods for the FLIR and color sensors are essentially equivalent. For simplicity, most of the development will be cast in terms of the color sensor, and mention of the FLIR will only be made when specific differences arise.

Of the two ways to visualize the relationship between the sensor image and the object model, projecting object features into the image space is perhaps more commonly seen in the context of object recognition, while texture mapping imagery onto a 3-D model is more common in computer graphics. For the former case, the features predicted to be visible for the object model are projected into the image using the 3-D transformation $M_{\mathcal{W},\mathcal{C}}$ and the projection $O_{\mathcal{C}}$ for the color sensor. Similarly, $M_{\mathcal{W},\mathcal{F}}$ and $O_{\mathcal{F}}$ are used for the FLIR image.

When mapping imagery onto the models, the same transformations are used to determine a relationship between points on the object model surfaces and points in the color image. Once it is known which color pixel belongs to each 3-D point in the model's native $\mathcal{W}$ environment, it is possible to texture map the color imagery onto the 3-D model. Figure 4 displays the various model to optical relationships just discussed.

6.1.1 Model Over Image Display

Section 3.3 briefly described the process used to predict a set of features expected to be visible for a given object model. For the color sensor, the factors considered when making this prediction are the object model orientation and the direction of the light source. The center image in the bottom row of Figure 4 (see color plates) shows an example of the features projected into the color image. The lines shown in red were determined to lie on the object silhouette. Those shown in green represent significant internal lines predicted based on the lighting model. To the right of the color image the predicted features, all in black, are shown projected into the FLIR image.

The complete mapping from object reference frame $\mathcal{W}$ to the color image plane, based upon equations 1 and 4, may be written as:

$$p_{\mathcal{C}} = O_{\mathcal{C}} M_{\mathcal{W},\mathcal{C}} P_{\mathcal{W}} \tag{8}$$

where the intrinsic camera parameters for the color sensor are given above in Table 1. A similar equation using the FLIR transformation and intrinsic parameters defines the mapping for the FLIR visualization. In both cases, the line displayed over the image is produced by mapping the 3-D endpoints of the model line features to the image plane.

6.1.2 Image Over Model Display

For visualizing the color or FLIR as it maps back into the 3-D object model reference frame, the original BREP model is used in conjunction with the current transformations. The imagery is mapped onto visible faces using what are essentially standard texture mapping techniques [1]. For each face, three vertices are selected to define a face specific 2-D coordinate reference frame which is used to map between image and face reference frames.

The top row of Figure 4 (see color plates) illustrates the results of the color texture mapping process with the model rendered from several different viewpoints. The middle row of Figure 4 shows the same model rendered with the FLIR texture map. To formally define the image mapping process for a 3-D planar face, let P_o be a 3-D face vertex selected to serve as the origin of the face coordinate reference frame. Pick two additional vertices P_1 and P_2 so as to define the basis vectors of the face specific 2-D reference frame:

$$U_{\mathcal{M}} = P_1 - P_o \qquad V_{\mathcal{M}} = P_2 - P_o \tag{9}$$

Basis vectors $U_{\mathcal{M}}$ and $V_{\mathcal{M}}$ are not usually orthogonal, which is not a problem so long as they are not co-linear (or nearly so).

To map between the color (or FLIR) image plane and the face, comparable basis vectors may be defined on the image plane using the projection of the 3-D vertices into the image:

$$\begin{aligned} U_{\mathcal{C}} &= (O_{\mathcal{C}} M_{\mathcal{W},\mathcal{C}} P_1) - (O_{\mathcal{C}} M_{\mathcal{W},\mathcal{C}} P_o) \\ V_{\mathcal{C}} &= (O_{\mathcal{C}} M_{\mathcal{W},\mathcal{C}} P_2) - (O_{\mathcal{C}} M_{\mathcal{W},\mathcal{C}} P_o) \end{aligned} \tag{10}$$

The basis vectors $(U_{\mathcal{M}}, V_{\mathcal{M}})$ and $(U_{\mathcal{C}}, V_{\mathcal{C}})$ allow us to define a simple invertible mapping between the image space and the object model face. To go back and forth between 3-D points P on the face and 2-D points p on the image:

$$p = (U_{\mathcal{M}} \cdot P)\, U_{\mathcal{C}} + (V_{\mathcal{M}} \cdot P)\, V_{\mathcal{C}} \tag{11}$$

$$P = (U_{\mathcal{C}} \cdot p)\, U_{\mathcal{M}} + (V_{\mathcal{C}} \cdot p)\, V_{\mathcal{M}} \tag{12}$$

To map the pixel grid onto the 3-D object face, first the k vertices of the convex face polygon are mapped to the image plane using equation 11. Next, a standard active edge list polygon fill algorithm [8] is used to enumerate the pixels falling within the face polygon. Finally, the pixels, or more specifically the coordinates of the vertices at the four corners of each pixel rectangle, are mapped back onto the 3-D object face using equation 12.

6.2 3-D Model and LADAR Display

We have developed a new method for displaying data from a range imaging device (LADAR) and relating this data to a 3-D object model. This technique provides a three-dimensional view of the data which allows the viewer to interactively explore 3-D relationships from any distance and viewpoint. The range data is displayed as a partial 3-D model of the scene in a coordinate system relative to the sensor. Each range data sample is displayed as a rectangle at the measured depth, occupying the angular space covered by the LADAR pixel.

The 3-D coordinates ($X_{\mathcal{L}}$, $Y_{\mathcal{L}}$, $Z_{\mathcal{L}}$) for a pixel (u, v) with range value D must satisfy the following constraints:

$$u = s_{u,\mathcal{L}} \frac{X_{\mathcal{L}}}{Z_{\mathcal{L}}} + t_{u,\mathcal{L}}$$

$$
\begin{aligned}
v &= s_{v,\mathcal{L}} \frac{Y_{\mathcal{L}}}{Z_{\mathcal{L}}} + t_{v,\mathcal{L}} \qquad (13) \\
D &= \sqrt{X_{\mathcal{L}}^2 + Y_{\mathcal{L}}^2 + Z_{\mathcal{L}}^2}
\end{aligned}
$$

Using these constraints, an inverse mapping $\mathcal{D}$ may be written as:

$$
\left| \begin{array}{c} X_{\mathcal{L}} \\ Y_{\mathcal{L}} \\ Z_{\mathcal{L}} \\ 1 \end{array} \right| = \left| \begin{array}{c} \alpha\gamma \\ \beta\gamma \\ \gamma \\ 1 \end{array} \right| = \mathcal{D}(u,\, v,\, D) \qquad (14)
$$

where:

$$
\begin{aligned}
\alpha &= \frac{u - t_{u,\mathcal{L}}}{s_{u,\mathcal{L}}}, \\
\beta &= \frac{v - t_{v,\mathcal{L}}}{s_{v,\mathcal{L}}}, \qquad (15) \\
\gamma &= \frac{D}{\sqrt{\alpha^2 + \beta^2 + 1}}
\end{aligned}
$$

Using $\mathcal{D}$, the 3-D corners of a LADAR pixel are defined by:

$$
\mathcal{D}(u \pm 1/2,\, v \pm 1/2,\, D) \qquad (16)
$$

Figure 5 (see color plates) illustrates different aspects of the model and LADAR display. The top left image of Figure 5 shows the BREP object model in the range sensor reference frame as determined by the transformation $M_{\mathcal{W},\mathcal{L}}$. The next image in the same row shows the sampled surface of the model generated by the feature prediction algorithm discussed in Section 3.3. Each range point is rendered in the color of the face which caused the sampled point.

Both the range data itself and the sampled model features may be shown simultaneously in the model and LADAR display. The third image from the left in the top row of Figure 5 shows the predicted model features in red and the LADAR range features in blue. The rectangles in this display are drawn in outline so one can actually see through a sample point to those which are behind. While the effect on a printed page is only moderately informative, interactively moving the viewpoint effectively displays the spatial relationships. The next image in the row uses the color coding to show the feature set. Again, red and blue are used to distinguish between model and data features, and now two new colors (yellow and cyan) have been introduced to represent which pixels are being used during the matching process. The cyan pixels are the data features used, and yellow are the corresponding model features being used.

The bottom row of Figure 5 show the model drawn with the range data in a common frame of reference. By allowing different viewpoints, the user can easily determine how well the range data fits the model features. The second image on the bottom row shows a view looking down on the model and data. From this perspective, the outline of the data points forms an L-shape corresponding to the general shape of the vehicle. The next image shows a different perspective which is more difficult to interpret. The final image shows a side view of the model and range data. Again, this figure is difficult to interpret, but if one looks closely, a general trend in the data can be seen which represents the front portion of the vehicle.

Display of the range data in three-dimensional space gives the user a more complete representation of the spatial layout of the data. Interactive control of viewing parameters allows the user to examine the range data from many different viewpoints at any magnification level. The user may select a point of interest in the data using the cursor in the viewer window, or by specifying a 3-D location directly using sliders in the control panel. The control panel also provides sliders for the user to change the view azimuth, elevation, and distance relative to the point of interest. User controlled de-cluttering of the image is accomplished by specifying the range of depth values displayed relative to the viewer, the sensor, or both, using sliders on the control panel. Grey scale or pseudo-color values are rescaled to represent the selected depth range. Several built-in color palettes are provided, as well as the ability to load a color palette from a user-specified file. A workstation with hardware graphics acceleration is used to perform display update at interactive rates. Current implementations are for Sun Sparc10/24ZX systems using the PEX and OpenGL graphics libraries.

7 Sensor to Sensor Relationships

As a visualization interface for object recognition algorithms, the model-to-sensor visualizations just presented are invaluable. However, another very important feature of ModelView is the ability to to visualize changes in pixel-to-pixel alignment between sensors. Sensor-to-sensor visualization is important for two reasons. First, one product of recognition [2; 21] is an adjustment to the sensor-to-sensor transformations and it helpful to see this adjustment. Perhaps more importantly, fusing the display of multisensor data provides the user with the opportunity to see relationships not evident when each sensor is viewed separately.

7.1 Displaying Color and FLIR

The transformation network specified in Figure 2 suggests that perhaps enough is known to compute a mapping between the color and FLIR sensors. Figure 6 (see color plates) illustrates such a combined visualization using the simplifying assumption that all points in the scene lie at the depth of the object model.

7.1.1 Approximating Pixel Mappings

Since the color and FLIR sensors are constrained to translate in a common image plane, there exists a fairly simple expression for the mapping of points from one image plane to the other. Begin with the equation for all points which project to a single color pixel:

$$P_{\mathcal{C}} = \begin{vmatrix} Z\frac{(u_{\mathcal{C}} - t_{u,\mathcal{C}})}{s_{u,\mathcal{C}}} \\ Z\frac{(v_{\mathcal{C}} - t_{v,\mathcal{C}})}{s_{v,\mathcal{C}}} \\ Z \end{vmatrix} \tag{17}$$

As Z varies, $P_{\mathcal{C}}$ traces out points lying on the ray passing through the sensor focal point and image point $u_{\mathcal{C}}$ and $v_{\mathcal{C}}$. Since the FLIR reference frame differs by only an XY translation, the same set of points in the FLIR reference frame may be written as:

$$P_{\mathcal{F}} = \begin{vmatrix} Z\frac{(u_{\mathcal{C}} - t_{u,\mathcal{C}})}{s_{u,\mathcal{C}}} + T_x \\ Z\frac{(v_{\mathcal{C}} - t_{v,\mathcal{C}})}{s_{v,\mathcal{C}}} + T_y \\ Z \end{vmatrix} \tag{18}$$

The projection of this point into the FLIR image plane may be written as:

$$\begin{aligned} u_{\mathcal{F}} &= s_{u,\mathcal{F}}\left(\frac{(u_{\mathcal{C}} - t_{u,\mathcal{C}})}{s_{u,\mathcal{C}}}\right) + t_{u,\mathcal{F}} + s_{u,\mathcal{F}}\frac{T_x}{Z} \\ v_{\mathcal{F}} &= s_{v,\mathcal{F}}\left(\frac{(v_{\mathcal{C}} - t_{v,\mathcal{C}})}{s_{v,\mathcal{C}}}\right) + t_{v,\mathcal{F}} + s_{v,\mathcal{F}}\frac{T_y}{Z} \end{aligned} \tag{19}$$

As expected, equation 19 shows how a point maps between color and FLIR depends upon the Z coordinate of the 3-D point being viewed. In the absence of knowledge of Z for each pixel in the color image, some assumptions must be made. The simplest assumption, and the one used here, is to assume all points are at a constant depth $Z = D$. Since there is typically an object model of interest at a known depth D, the choice is not entirely arbitrary. Clearly it is far better to use pixel specific Z values when possible, for example when these estimates are available from a range sensor.

7.1.2 Combining Imagery With HSV

Once the mapping between the two image planes is obtained, the sensor information can be combined. It is useful to represent the combinations in HSV space. For example, by modulating different channels in HSV it is possible to make pixels either colorful or grey-scale based upon the response of another sensor. Two different combination schemes have been found to be useful.

To define these combinations, let O_h, O_s, O_v be the HSV components of the output for a pixel which is displayed to the user. Let I_h, I_s, I_v be the input values for the corresponding pixel in the color and let F be the value for the corresponding pixel in the FLIR image. Since the correspondence is not exact, the nearest pixel is selected based upon the geometric mapping discussed in the previous section. For the FLIR, it is also helpful to define normalized FLIR $F_{[0,255]}$ compressed to the ranges $[0, 255]$ (as well as $F_{[0,1]}$ in the range $[0, 1]$).

The two combinations used by ModelView and shown in Figure 6 (see color plates). The first uses FLIR intensities combined with the value component of the color pixel in HSV space:

$$\begin{vmatrix} O_h \\ O_s \\ O_v \end{vmatrix} = \begin{vmatrix} I_h \\ I_s \\ \gamma F_{[0,255]} + (1.0 - \gamma)I_v \end{vmatrix} \tag{20}$$

and the FLIR intensity mapped onto the saturation component of the color HSV pixel:

$$\begin{vmatrix} O_h \\ O_s \\ O_v \end{vmatrix} = \begin{vmatrix} I_h \\ \gamma F_{[0,1]} + (1.0 - \gamma)I_s \\ I_v \end{vmatrix} \tag{21}$$

where γ is weighting term used to control the FLIR intensities influence over the resulting pixel values. For Figure 6 γ was set to 0.4.

Figure 6 shows the FLIR mapped on the value component of the color pixel (equation 20). In the first part of the figure, the mapping or alignment is perfect, and in the next, the images are slightly mis-aligned. Notice the faint impression of the false vehicle present, and the noticeable discrepancy along the horizon. The next two parts of the figures show the FLIR mapped on to the saturation component of the color pixel (equation 21). The result is an image which has vivid, highly saturated colors in areas of interest due to high thermal intensity (such as the vehicle) and has dull, unsaturated (grey) colors in areas of low thermal intensity (for example the trees on the skyline). In the mis-aligned image, observe how clearly the natural terrain fades from color to grey-scale as you look up toward the horizon. The FLIR is offset downward relative to the color, and the black sky in the FLIR is causes the terrain to loose its color.

8 Visualizing Recognition

Not all relationships for which visualization displays have been developed are equally important from the standpoint of understanding how an object recognition algorithm is performing. While monitoring the progress of the recognition process, only a few relationships are critical. Those displays tend to be left up on the screen so the researcher can simultaneously view the progress of the recognition using these few selected visualization techniques. Figure 7 (see color plates) shows an example of how such a screen might appear.

Typically, three relationships are shown with four different windows. The first shows only the model. This window is helpful to determine how well the recognition system has determined the proper model orientation across all sensors. The next four windows visualize the model relative to each sensor. The predicted silhouette and internal edges are rendered into the optical sensors, and the predicted range features are shown against the range data. Finally, once the recognition algorithm has converged to a solution, the other relationships can be examined to determine how well the recognition system corrected the pixel-to-pixel alignment between sensors.

9 Conclusion

Visualization of multisensor model-based object-recognition algorithms is an essential component for determining how well the system performs. Since ground-truth is not always available, visualization becomes the key component of verification. Once multiple sensors are integrated into the matching system, the visualization system must be able to show complex sensor-to-sensor relationships as well as model-to-sensor relationships. ModelView is a system which has been shown to meet many of the needs of our current matching system. Its model-centered visualization paradigm is a step above the data-centered approach utilized in our previous work. Furthermore, ModelView shows promise as a general framework for visualizing relationships between heterogeneous sensor data.

References

[1] Alan Watt and Mark Watt. *Advanced Animation and Rendering Techniques*, chapter Mapping Techniques: Texture and Environment Mapping. ACM Press: Addison-Wesley, New York, 1992.

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

[3] Nicholas Ayache and Bernard Faverjon. Efficient registration of stereo images by matching graph descriptions of edge segments. *International Journal of Computer Vision*, 1(2), 1987.

[4] Mark Bellrichard. Alliant techsystems LADAR field calibration. Personal Correspondence, November 1993.

[5] 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.

[6] R.C. Bolles, H.H. Baker, and M.J. Hannah. The jisct stereo evaluation. *IUW*, 93:263–274, 1995.

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

[8] J. D. Foley and A. Van Dam. *Fundamentals of Interactive Computer Graphics.* The Systems Programming Series. Addison–Wesley, Reading, Massachusetts, 1982.

[9] Aaron D. Fuegi. A tool for visualization of multi-sensor data for automatic target recognition. Master's thesis, Colorado State University, Fort Collins, Colorado, September 1994.

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

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

[12] W. E. L. Grimson. The Combinatorics of Object Recognition in Cluttered Environments Using Constrained Search. *Artificial Intelligence*, 44(1):121 – 165, July 1990.

[13] 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.

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

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

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

[17] H. Li, B.S. Manjunath, and S.K. Mitra. A contour-based approach to multisensor image registration. *Ieee transactions on image processing*, 4(3):320, march 1995.

[18] H. Li, B.S. Manjunath, and S.K. Mitra. Multisensor image fusion using the wavelet transform. *Graphical models and image processing*, 57(3):235, may 1995.

[19] 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.

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

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

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

[23] Robin R. Murphy. Robust sensor fusion for teleoperations. In *IEEE International Conference on Robotics and Automation*, 1993.

[24] Michael John Muuss. Towards real-time ray-tracing of combinatorial solid geometric models. In Keith Applin, editor, *Symposium of BRL/CAD '95.* Army Research Laboratories, June 1995. http://ftp.arl.mil:80/ mike/.

[25] R. Nevatia, K. Price, and G. Medioni. USC image understanding research: 1993-1994. In *Proceedings: Image Understanding Workshop*, pages 69–80. ARPA, November 1994.

[26] M. Okutomi and T. Kanade. A multiple-baseline stereo. *PAMI*, 15:353–363, April 1993.

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

[28] Steven K. Rodgers, Carl W. Tong, Matthew Kabrisky, and James P. Mills. Multisensor fusion of ladar and passive infared imagery for target segmentation. *Optical Engineering*, 28(8):881–886, August 1989.

[29] Hillel Rom and Gérard Medioni. Part decomposition and description of 3D shapes. In *Proceedings: Image Understanding Workshop*, pages 1505–1522. ARPA, November 1994.

[30] F. Russon and G. Ramponi. Fuzzy methods for multisensor data fusion. *Ieee transactions on instrumentation and measurement*, 43(2):288, april 1994.

[31] 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.

[32] 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.

[33] 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.

[34] C.W. Tong. Target segmentation and image enhancement through multisensor data fusion. Master's thesis, Air Force Institute of Technology, 1986.

[35] 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.

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

[37] J. W. M. van Dam, B. J. A. Kruse, and F. C. A. Groen. *Artificial Neural Networks*, chapter Optimizing local Hebbian learning: use the δ-rule, pages 631–634. Springer Verlag, 1994.

[38] J. W. M. van Dam, B. J. A. Kruse, and F. C. A. Groen. Transforming the egocentered internal representation of an autonomous robot with the cascaded neural network. *IEEE*, pages 667–674, October 1994.

[39] Jacques G. Verly, Dan E. Dudgeon, and Richard T. Lacoss. Model-Based Automatic Target Recognition System for the UGV/RSTA LADAR. In *Proceedings: Image Understanding Workshop*, pages 559–583. ARPA, November 1994.

[40] H.Q. Wu, D. Grenier, Y.G. Delisle, and D.G. Fang. Translational motion compensation in isar image-processing. *Image Processing*, 4(11):1561–1571, November 1995.

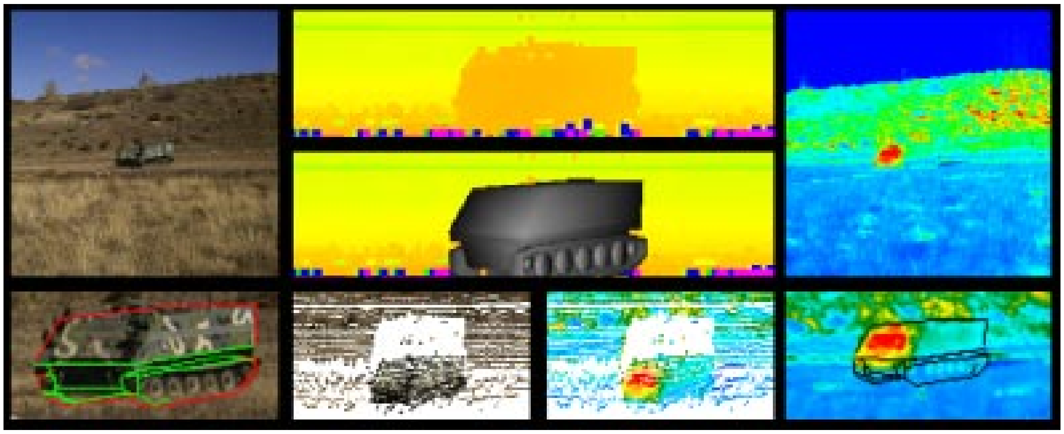


Figure 1: Sample screen images of the ModelView system. The top left column contains a color image with the model projected into the scene on bottom. The top right column shows a FLIR image with the model projection beneath. The center column shows four views of a LADAR image.

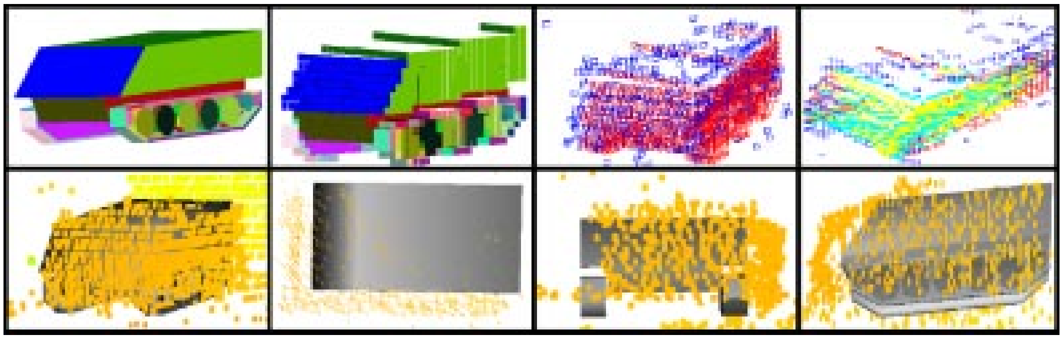


Figure 5: Model To LADAR Relationships. The top left image shows the CAD model. Next is the sampled surface points generated from the model. The next two images show the sampled surface points in relation to the actual data. The bottom row contains images of the model rendered within the 3-D range scene.

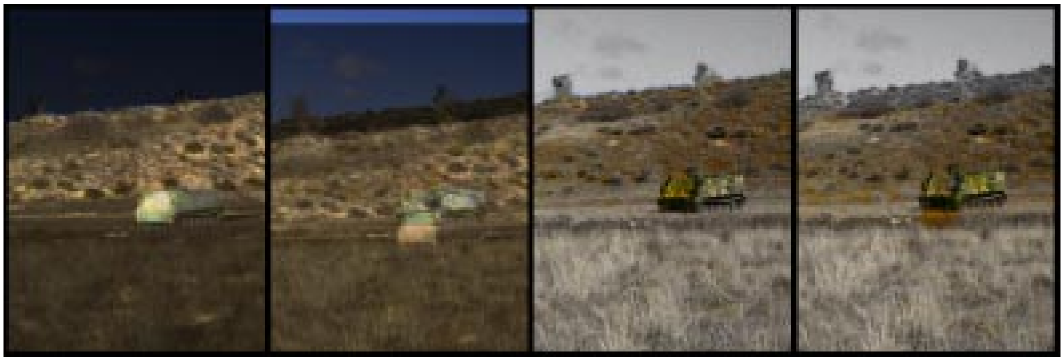


Figure 6: Color To FLIR Relationship. The left-most image shows the FLIR mapped to the intensity component of the color image for the correct transformation. Next, the same mapping is used, but the transformation is incorrect. The final two images show the FLIR mapped to the color saturation component for both the correct and incorrect transformations.

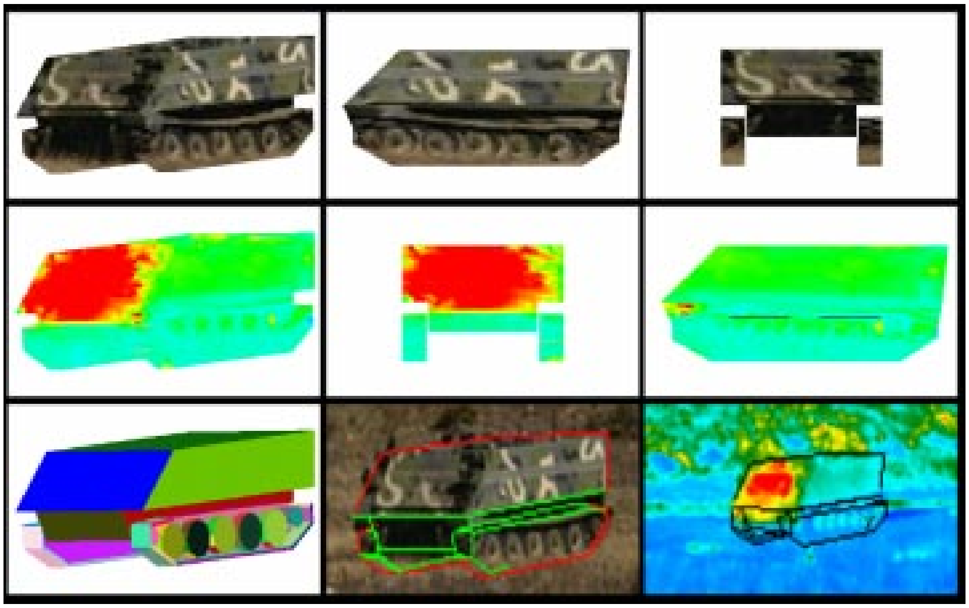
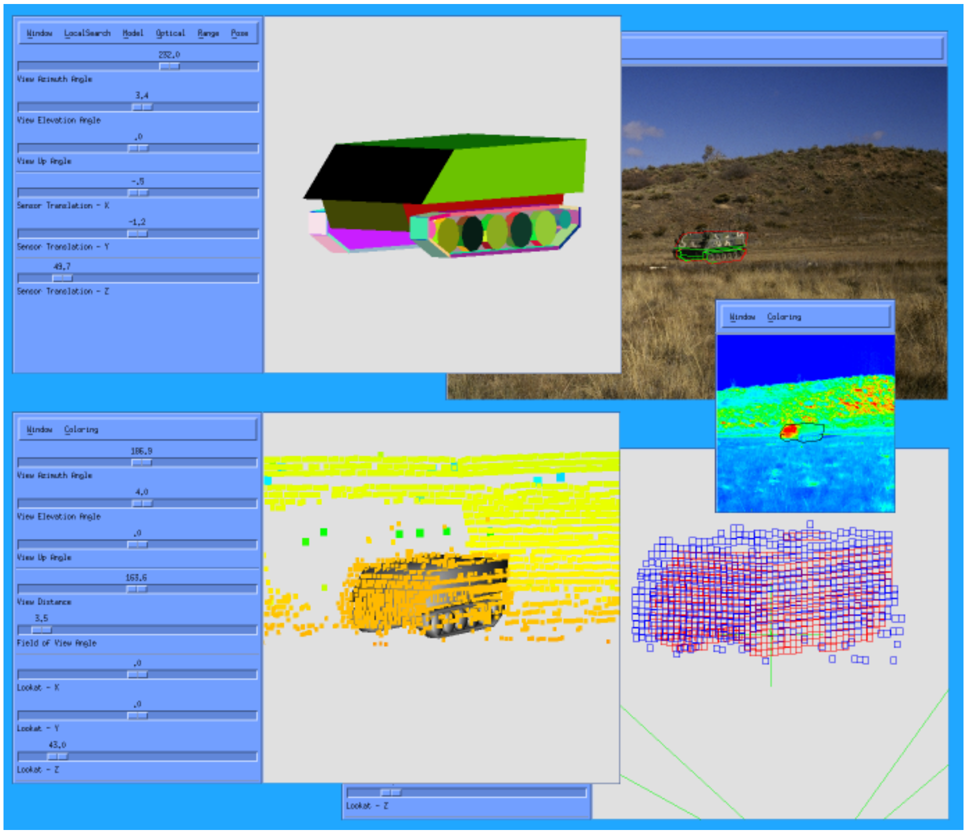


Figure 4: Model To Optical Relationships. The top row contains the model with the color image texture map. The middle row has the model with the FLIR texture map. The last row shows the model, the model lines projected into the color image, and the model lines projected into the FLIR.



232.0
View Azimuth Angle
3.4
View Elevation Angle
.0
View Up Angle
-.5
Sensor Translation - X
-1.2
Sensor Translation - Y
49.7
Sensor Translation - Z

186.9
View Azimuth Angle
4.0
View Elevation Angle
.0
View Up Angle
163.6
View Distance
3.5
Field of View Angle
.0
Lookat - X
.0
Lookat - Y
43.0
Lookat - Z

Lookat - Z

Figure 7: A Sample Screen Layout of ModelView