Machine Vision for the Battlefield

The development of vision systems for the autonomous land vehicle has been marked by a shift away from numerical techniques and toward symbolic computing, knowledge-based reasoning, and modeling.

Bir Bhanu (Systems and Research Center)
MN65-2300; HVN 782-7676

In 1983, the Defense Advanced Research Projects Agency announced the Strategic Computing Initiative, a 10-year plan to develop the technologies needed for machine intelligence. The plan includes programs to develop three intelligent systems: an autonomous land vehicle, a pilot’s associate, and an aircraft-carrier battle-management system. Each program is intended to be particularly attractive to one branch of service; the autonomous land vehicle, or ALV, is of particular interest to the Army.

The ALV is a robotic vehicle capable of completing a surveillance or resupply mission without assistance. The Army sees the ALV, other unmanned vehicles, and smart weapons as a means of countering the Soviet bloc’s superiority in men and conventional weapons. The Army’s scenario for land battle in the year 2000, called the Air-Land Battle 2000, envisions that automation will create “operational reserves” by taking over some battlefield tasks, giving U.S. forces the ability to increase the tempo of the battle to one characterized as “relentless attack” even though they are outnumbered. Before an ALV can be fielded, however, many difficult computing problems must be solved. Some of the most difficult have to do with vision.

Classical approaches to battlefield vision rely heavily on numerical techniques that reduce an object to a list of measurements and reduce movement to a displacement vector. They were devised for recognizing military targets from aircraft — a difficult task, but one that is inherently easier than the ALV’s. A recognition system aboard an aircraft sees targets from essentially one point of view and at a nearly constant range. Moreover, the background is flat and neither causes the aircraft to change its course, altering what the camera sees, nor provides cover for targets, preventing the camera from seeing them. The ALV, on the other hand, must deal with the full three-dimensional complexity of the natural terrain through which it is moving and which may conceal several moving targets at different ranges. Under such circumstances, the numerical techniques have proved brittle.

In devising a machine-vision system for the ALV, we have adopted many computing techniques from the field of artificial intelligence. Although each component of the system is unique, several trends can be distinguished. Instead of being characterized by overly precise numeric values, objects are characterized by more tolerant symbolic values. Although low-level image-processing functions remain largely quantitative, at intermediate and higher levels we are introducing qualitative reasoning. Instead of atomizing objects, or reducing them to a few features, we are modeling them in all their complexity. And instead of ignoring the local environment, we are modeling it as well. These new approaches promise vision systems that are more robust than the current target recognizers.

The machine-vision projects at the Systems and Research Center are extremely ambitious, and a production autonomous vehicle is probably still many years away. Recently, the Advanced Systems Center of the Defense Systems Group undertook the task of engineering semi-autonomous vehicles (able to be driven by remote control) that will be ready much sooner. We are collaborating with them on two projects to solve the easier, but no means trivial, vision problems these vehicles pose. These projects are briefly described at the end of this article.

A Mobile Platform

Clearly, the ALV must be able to recognize independent motion in a series of images if it is to respond appropriately to moving threats. But because the ALV is moving itself, recognizing independent motion is harder than it might seem. Stationary objects generally will not appear to be stationary in the images. Moreover, moving objects may not necessarily appear to be in motion. Before any useful conclusions can be drawn, the effects of the vehicle’s motion on the image must somehow be determined.

If the vehicle’s own motion can be removed from the images, they can provide information about the topology of the vehicle’s environment. For example, as the vehicle moves forward, nearby objects grow larger at a faster rate than distant ones. Indeed, information deduced from image motion can be used to construct a three-dimensional model of the local environment.

Previous work in motion analysis concentrated on numerical techniques for recovering three-dimensional structure from two-dimensional images.
Typically, the geometry of the environment and the motion of the vehicle were estimated simultaneously by solving a system of equations. This technique had three drawbacks. First, it was notoriously sensitive to noise. Second, it assumed the entire environment was stationary. If there was a moving object nearby, the system of equations might converge to a single, erroneous solution or it might exhibit a large residual error. In neither case would it allow the location of the moving object to be pinpointed. Third, the technique made no allowance for the ambiguity of visual evidence and the possibility that several interpretations of the evidence might temporarily be equally plausible.

In our approach, which is called Dynamic Reasoning from Integrated Visual Evidence (DRIVE), only the first step of motion analysis — determining the vehicle’s own motion — is numerical. The topology of the environment is reconstructed by qualitative reasoning rather than numerical computation. More than one model of the environment is maintained until ambiguities can be resolved. One indication of the greater robustness of this technique is that it is able to locate and track moving objects in a wide variety of ALV imagery.

Finding the Focus of Expansion

The ALV’s movement can be described as a combination of translation and rotation. Only translation supplies information about the topology of the environment, and so the first step in motion analysis is to remove the effects of vehicle rotation from the images. Suppose a camera-centered coordinate system is defined, whose origin lies at the center of the camera lens. Rotations of the vehicle about the axis that is perpendicular to the image plane and passes through the center of the camera lens would make a point in the scene trace a circle in the image. Rotations about the other two axes would make a point in the scene trace hyperbolic paths in the image.

Qualitative Scene Model is developed by deducing information from the expansion rates of tracked image points. At time \( t_0 \) there are three points in the model, all of which are assumed to be stationary. At time \( t_1 \), the subsystem has concluded that point \( a \) is closer than points \( b \) or \( c \) and that point \( c \) is closer than point \( b \). At time \( t_2 \), a conflict arises in this interpretation. A new piece of evidence indicates \( c \) is closer than \( a \) rather than the other way around. As a result, two new interpretations (2 and 3) are created, each of which considers one point as mobile (heavy circles). At time \( t_3 \), a conflict arises in interpretation 2. It appears point \( b \) may be closer than \( c \) rather than the other way around. Since there is an active interpretation (3) that can explain this conflict, interpretation 2 is collapsed. Only one valid interpretation remains.
Once these rotations are removed, the remaining image motion is due to translation. When the camera moves forward along a straight line, every point in the image seems to expand from a single point. This point, which is the intersection of the vehicle’s velocity vector with the image plane, is called the vanishing point in the graphic arts and the focus of expansion (FOE) in machine vision. The goal of derotation is to locate the FOE. Once the FOE has been located, the relative distance between each point in the scene and the camera can be determined from the rate of expansion. Nearby points appear to rush outward rapidly, whereas distant points hardly move at all.

There are several methods of locating the FOE, but experimentation has shown that a method called Rotation-from-FOE is one of the more robust. First, a set of points is extracted from each of two frames of imagery, and displacement vectors are computed for these points. The location of the FOE is then guessed. The initial guess might be that the FOE has the same location as the FOE in the last pair of frames to be analyzed. Next, the second image is “derotated”; in other words, the tracked points are mapped to the locations they would have if the vehicle had not rotated between frames. Derotation is guided by the attempt to arrive at a perfectly radial pattern of displacement vectors. If the second image is perfectly derotated, all displacement vectors will lie on straight lines passing through the FOE.

If the FOE is misplaced, however, a perfect pattern cannot be achieved. Then a measure of the error between the guessed location and the true location of the FOE is computed, the location of the FOE is guessed again, and the error the new location yields is calculated. The process is repeated until the FOE that yields the minimum error is found.

In a final step, the algorithm computes the region around this FOE within which the error is below some threshold. This “fuzzy” FOE embodies the assumption that noise and image distortion will prevent the location of the FOE from being determined exactly. The fuzzy FOE prevents a false precision from causing later processing steps to fail.

Once the FOE has been located, the vehicle’s velocity can be determined. The distance between the camera and the ground is known, and the camera’s angle of depression can be found from the location of the FOE in the image. Knowledge of this distance and angle allows the distance between a point in the scene and the vehicle to be estimated by triangulation. From the change in the absolute position of a point, the velocity of the vehicle can be determined.

**Constructing a Scene Model**

A three-dimensional camera-centered model of the scene is gradually built up from the information gleaned by motion analysis. The building blocks of this model are *entities*, which are the three-dimensional counterparts of the points tracked during motion analysis. The model also includes the properties of entities, such as whether they are stationary or mobile, and the relations between entities, such as which of two is closer to the camera. To distinguish this model from the geometric models that were the goal of earlier attempts at motion analysis, we call it the Qualitative Scene Model, or QSM.

The model takes the form of a network of hypotheses that has the property of inheritance. Hypotheses common to all interpretations of the scene are found near the root of the network and are inherited by all interpretations below them. When a fact is received that is consistent with more than one hypothesis about the scene, the model branches. All interpretations are developed simultaneously as new information is received, and an interpretation is eliminated when a conflict arises between two of the hypotheses it contains.

The mechanism for developing the model is a rule-based blackboard. In other words, the network is accessible to several sets of rules, each of which can modify it. There are two major groups of rules. Generation rules examine the derotated images for significant changes and add new hypotheses to the model. For example, if two image points *a* and *b* lie on opposite sides of the fuzzy FOE and they are getting closer to each other, then one point must be in motion relative to the other. If an interpretation includes the hypothesis that one of these points is stationary, then the other point can be asserted to be mobile.

Verification rules review interpretations after they have been created and attempt to prove them false. They are typically rules that would produce too many conclusions if they were used as generators; a violation of such a rule often indicates an interpretation is implausible. For example, one verification rule is that if an object is lower than another in the image, it is closer to the camera. Suppose point *a* is lower than point *b* but an interpretation already includes the hypothesis that point *b* is closer to the camera than point *a*. If this verification rule fires, the interpretation will be marked as conflicting. Whether or not it is ultimately removed from the model will depend on the global state of the QSM.

**Locating Moving Objects**

A primary objective of motion analysis is to locate moving objects. One approach to this problem is frame differencing, or subtracting successive frames pixel-by-pixel, looking for areas where the values do not subtract to zero. If the camera itself is moving, however, this technique generates too many false alarms to be useful. A more sophisticated approach is to apply a two-dimensional transformation to the image, warping it to compensate for the motion of the background, before looking for anomalous regions. This approach works well, however, only if the object is moving in front of a relatively flat background. DRIVE avoids both pitfalls, taking account of the camera’s motion and allowing for the full three-dimensional structure of the background.

**DRIVE** detects motion in two ways. Some kinds of motion can be deduced directly from the pattern of the two-
IMAGE SEQUENCE is from a video tape used during the development of the DRIVE motion-analysis subsystem. The images contain two moving objects: a car that has passed the ALV and is barely visible in the distance, and a second car that is approaching in the opposite direction and is about to pass.

EDGES were obtained from the video imagery at the top of this page. The numbered circles indicate points that are being tracked from image to image. Note that points 24 and 33 are on the moving cars.

DISPLACEMENT VECTORS indicate the apparent motion of points from frame 195 to frame 196 (left) and from frame 196 to frame 197 (right). The shaded area marks possible FOE locations; the circle inside this area is the FOE with the lowest error value. The FOE ratio indicates the shape of the error function inside the shaded area. A low value indicates a flat function. The endpoints of selected vectors used to compute the vehicle's velocity and the distance it has advanced are marked with dots. The distance the vehicle has advanced is estimated in meters. Its rotation about two axes (by amounts less than one degree) is plotted. Rotations about the third axis are small enough to be neglected.

dimensional displacement vectors. For example, if a point is found to be moving toward the fuzzy FOE rather than away from it, then it must belong to a moving object. No other interpretation is possible.

Other kinds of motion are more subtle. For example, suppose the ALV is approaching a T intersection, that there is a building on the far side of the right branch of the T, that a van is approaching the intersection along the right branch, and that DRIVE is tracking a point on the building and one on the truck. As the ALV moves forward, the point on the building will appear to move outward toward the boundary of the image. If the truck is approaching the intersection as fast as the ALV, however, it will remain at the
same position in the image. From the expansion pattern alone, the ALV might conclude that the truck is a stationary object at infinite distance and so collide with it at the intersection.

The existence of the QSM, however, allows a more sophisticated kind of reasoning. Because the truck is occluding the building, it must be closer to the ALV than the building is. Since the image point on the building is moving outward, the building cannot be at infinite distance. Therefore, the truck cannot be at infinite distance, and there must be a different interpretation of its expansion pattern. The most probable interpretation is that the truck is moving toward the ALV.

**Labeling Terrain**

The development plan for the ALV requires the vehicle to negotiate terrain that becomes more challenging each year, graduating from a paved road, to open desert, to wooded areas. In 1987, it was to be able to travel across open desert and to demonstrate an understanding of types of soil and ground cover. The ultimate goal is that the ALV be able to label terrain regions specifically enough to be able to match them to regions in a digital map. An understanding of terrain will also allow the vehicle to plan a route that is traversable and provides good cover.

The standard approach to terrain labeling is to use image statistics such as texture values to distinguish terrain types. Like other statistical solutions to vision problems, this one is too rigid to be generally successful. Little may be known in advance about a terrain type's feature values, which in any case are likely to vary widely both because natural terrain displays infinite variety and because the viewing conditions have a significant effect on the feature values.

Our approach, which is called Hierarchical Symbolic Grouping for Multi-Spectral Data (HSGM), combines algorithmic calculations with knowledge-based reasoning. The imagery for HSGM comes from the Multi-Spectral Scanner, an instrument that is sensitive to 12 bands in the visible and infrared regions of the spectrum. The knowledge base contains general world knowledge, such as the fact that the top part of the image is likely to be the sky. In the future, HSGM might also use information about the local environment derived from the map in the PRACTE subsystem, which is described below.

HSGM first identifies large regions in the image, labeling them as sky, forest, field, road, or unknown. The same processing steps are then repeated to identify subregions and sub-subregions until no further subdivision appears to be necessary. At the lowest level, region labels can be as specific as gravel road, snowberry shrub, gambel oak, or rocky ledge.

In the first processing step, the image is segmented into regions by a texture-gradient operator. The operator calculates the local rate of change of an attribute that can be loosely described as the amount of fluctuation in intensity. By measuring the signatures of different types of terrain, we determined which scanner channels would yield the steepest gradients between pairs of terrain types, and only these channels are processed. The settings of the gradient operator's parameters are chosen for the range at which each terrain type is likely to appear. For example, given the ALV's current mission profile, fields will usually be between five and 100 yards away from the vehicle.

The evidence from the processed channels is combined to produce a single gradient image. A second algorithm then finds the edges that mark region boundaries and links incomplete edges. Gaps in the edges derived by applying the highest threshold to

**Scene Interpretations** are based on the apparent motion of points in the imagery. Stationary points are indicated by numbers in circles, and moving points are indicated by numbers in squares. A line between two points indicates that the one enclosed by the larger circle is closer to the ALV than the one enclosed by the smaller circle. A point that is near the bottom of the interpretation is near the bottom of the image. After frame 195, two interpretations are created. Interpretation 1, (left) according to which points 24 and 33 are moving, is ranked higher than interpretation 2, (center) according to which point 33 is stationary, because interpretation 1 contains more stationary points. If point 33 is stationary, its rapid expansion indicates it must be at least as close to the vehicle as point 76. Since point 76 is much lower in the image than point 33, this doesn't make sense. As a result of this conflict, interpretation 2 is eliminated, leaving only the correct interpretation (right).

Summer 1988

61
the gradient image are filled in by incrementally lowering the threshold. Once the regions have been outlined, statistics are calculated for each region.

The statistics are used by a knowledge-based region-labeling algorithm to determine how similar each segmented region is to each class of terrain. Three types of features are compared. The first is spectral characteristics, such as the mean and the standard deviation of the intensity values in a region. The algorithm also examines the correlation between a region’s location in the image and its expected location, given the orientation of the camera. Finally, the algorithm tests for valid relations among regions. A set of adjacency rules codifies which types of regions can be neighbors. This combination of calculation and reasoning promises to be much more robust than the strictly statistical approach.

Using Maps
A feature common to many of the subsystems we are describing is the attempt to supply information in advance, instead of deriving everything from raw data. In the case of the ALV, the most useful form of advance information is a digital map of the area it will be traversing. The digital map is more detailed and records different

MULTI-SPECTRAL SCANNER records the same image in 12 bands in the visible and infrared regions of the spectrum starting with the band from 0.44 to 0.49 microns (upper left) and extending to the band from 2.0 to 2.6 microns (lower right). No data were available for channel 9 because of an electronics malfunction.

TERRAIN CLASSIFICATIONS (right) were derived by the HSGM subsystem. For the purpose of comparison, a luminance image of the scene is shown to the left. The luminance image is the Y image of the NTSC television standard for the transmission of color imagery. In the segmented image, s stands for sky, f is for forest, g is for grass, r is for road and u is unknown. All major regions have been accurately detected except for the left and right forks of the road. These are missed only because they are distant, and the processing algorithms had been tuned to detect the road boundary at closer ranges.
information than the qualitative scene model discussed earlier. Whereas the qualitative scene model is a coarse range map of the scene, the digital map records terrain elevations and land cover.

A digital map would make many vision-related tasks easier; one notable example is target tracking. Early trackers tended to lose targets frequently, particularly when the image was noisy or had low contrast, and they had difficulty picking the targets up again once they were lost. It becomes much easier to track a target if the target can be shown to be traveling on a road. In this case, its path and even its disappearances can be predicted.

We have implemented a map-reasoning subsystem for target tracking that will be integrated with DRIVE. DRIVE provides the location of a moving target, its velocity and heading, and its approximate range. Given this information, the ALV’s location, and a model that describes the relation between the vehicle’s location and what the camera sees, the subsystem establishes an image-to-map registration and returns the target’s location in map coordinates to DRIVE. It then searches the map database for a description of the roads or terrain the target is traversing and returns this information to DRIVE. Information on nearby landmarks is also obtained. The final step is to predict when the target will disappear. Using techniques developed at Honeywell, digital topographical maps can be transformed into digital visibility maps that allow predictions of what can be seen from a given viewpoint. The map is used to predict obscurance; these predictions are also fed back to DRIVE.

Recognizing Landmarks
To find its way, the ALV relies on a land navigation system not unlike those aboard aircraft. Even when an inertial-navigation system is aboard an aircraft in smooth and level flight, it has an error rate of roughly one or two nautical miles per hour. Clearly, a vehicle that relied on such a system alone would soon be lost. To deal with this problem, the ALV has the ability to recognize selected landmarks, such as telephone poles, storage tanks, buildings, and gates. A landmark provides a position fix that allows the navigational system’s accumulated error to be corrected.

Landmark recognition is simply an instance of the more familiar problem of target recognition. Target recognition becomes very difficult if the point of view from which the target is seen and the target’s range are uncontrolled. To deal with this problem, early target-recognition systems emphasized the need for target models composed of rotation-invariant and range-independent features. Size, for example, is a range-dependent target feature, but the ratio of the square of an object’s perimeter to its area is range-independent. It soon became apparent, however, that too few features met these criteria. As a result, the models were weak and did not adequately discriminate between different classes of objects. Compounding the difficulty was the fact that segmentation, the step in which homogeneous regions are isolated from an image, is notoriously faulty. As a result, the recognizer was often in the position of attempting to match a mis-shaped image region to a weak model.

We have developed a knowledge-based landmark-recognition subsystem that sidesteps these problems. It is called PRACTE, which stands for Perception, Reasoning, Action, and Expectation. Given a vehicle’s location, velocity, and heading, PRACTE uses an internal map to generate an Expected Site Model, or ESM. The map consists of models of the road, the camera (its viewing geometry), and landmarks. The ESM amounts to a hypothesis about the next site and the location and appearance of the landmarks it contains. The segmented image is then compared with the site model, the objective being to label regions in the image that correspond to the expected landmarks.

PRACTE's map knowledge base takes the form of a hierarchical relational network whose primitives are schemata, or collections of attributes possessed by objects. The slots in a site schema include the approximate latitude, longitude, and elevation of the site, the next site, and a spatial model, which describes the location of landmarks relative to the road and to each other. The next-site's value depends on the vehicle's heading. The road plays a prominent role in the spatial model because a paved road is easily segmented and because the vehicle was not yet required to negotiate open terrain when PRACTE was designed.

Each landmark is also represented as a schema or a collection of schemata. The slots in a landmark schema may contain values for attributes such as the landmark’s color, texture, and gross shape. Other possible attributes, however, are a spatial model of the landmark’s local environment, and a geometric or semantic model of the landmark. A semantic model describes the position of the landmark’s features (parts) relative to one other.

Because the image content changes as the vehicle approaches the next site, the landmark models and the search-and-match strategy must be dynamic. The matching technique we have developed is called Dynamic Model Matching, or DMM. Recognition takes place in three stages. At distances greater than about 35 meters, only a few attributes of a landmark, such as its color, may be distinguishable. However, the position of the landmark with respect to the road can be extracted from the spatial model in the site schema. A dominant-feature landmark model and the spatial information generally allow a landmark to be detected but not to be recognized with a high degree of confidence.

Between 10 meters and 35 meters, the landmark’s image is likely to have enough resolution that lines and vertices can be extracted from it. As a result, it is possible to match the landmark’s image with the appropriate projection of a geometric model. A match confirms the identity of the landmark.

At close ranges, parts of the landmark can be identified, and it becomes
MAP KNOWLEDGE BASE in the PREACTE subsystem is a hierarchical network whose primitives are schemata. In the example shown here the vehicle is at site 31 and, given its heading, the next site is predicted to be site 32. The schema for site 32 is shown in the upper right corner. Its attributes include the view of the site the vehicle will have (given its direction of approach), the landmarks located at the site, and a spatial model. The spatial model, which is shown in greater detail below the site schema, defines the "expectation zone" of a landmark, or its position relative to the road and to other landmarks. Also shown is the schema for building 32, one of the landmarks expected to be at site 32. Landmarks are characterized by symbolic attributes, each of which is assigned a discriminant strength. For example, building 32's most distinctive attribute is likely to be linear boundaries. Additional attributes can be extracted from the geometric model that is one attribute of the building schema. As the vehicle approaches site 32, PREACTE attempts to match attributes of image regions to the left of the road with attributes of building 32.
possible to use a semantic model for matching. At this stage, PREACTE’s success or failure in matching image regions to landmarks allows it to confirm that the vehicle has arrived at the predicted site or to quantify the discrepancy between the vehicle’s expected location and its actual location.

Learning New Targets
Like any system that makes extensive use of knowledge, the ALV is subject to the knowledge-acquisition bottleneck. One of the more time consuming knowledge-engineering tasks is the creation of target models. Machine-learning technology, which should allow target models to be created semi-automatically, is one solution to this problem. Another shortcoming of many vision systems is their failure to adapt to varying light conditions, image noise, occlusion, or other minor problems. An intelligent matching strategy that could decide whether to look harder for certain target features or to create a new target model would make vision systems less fragile.

To demonstrate the feasibility of a target-recognition system with an integrated machine-learning capability, we are implementing a system called Target Recognition Incorporating Positive Learning Expertise (PREACTE). LANDMARK-RECOGNITION stages are shown in this sequence of screen images from the PREACTE subsystem. Site 105, which contains a gate and a telephone pole, has been predicted to be the next site the vehicle will traverse. In the left windows of the images, the subsystem has drawn projections of the three-dimensional geometric models of these landmarks. The right windows contain the processed image. The boxes are the minimum bounding rectangles of potential landmarks. During the detection stage (top), landmark attributes that are relatively insensitive to range, such as intensity, are matched. At the end of this stage, the site uncertainty is fairly high. The recognition stage (middle) uses new landmark models that emphasize a different set of attributes. The new positive evidence that results from these matching attempts reduces the site uncertainty. The requirements for a match at close range are stringent and, at the end of the verification stage (bottom), the site uncertainty has increased slightly.

Summer 1988

65
TRIPLE is a target-recognition system that uses two machine-learning techniques to maintain a knowledge base of target models. In this example, the targets are Soviet vehicles, and the system has just encountered a vehicle which does not match any in the knowledge base (top left). The first machine-learning component, EBL, takes the schema for the unknown vehicle and selects the best attributes for target recognition (top right). The selection is guided by inference rules such as: "Vehicle length, height, and wheelbase are always used in characterizing a target." The second machine-learning component, SCC, takes the new schema and integrates it into a classification tree (bottom). In this case the new schema was accommodated by adding the number-of-wheels branch to the tree.

The system combines two powerful learning methodologies with a knowledge-based matching technique for recognizing examples of known target types. The two methodologies are Explanation-Based Learning (EBL), which is used to create and refine target models, and Structured Conceptual Clustering (SCC), which is used to arrange models in an efficient classification tree. TRIPLE is one of the first attempts we know of to combine learning techniques. Neither EBL nor SCC, used separately, would be very effective, but the methods complement one another, and the hybrid approach is more robust than either approach alone.

The background information for the task of target modeling is organized into a network called a goal-dependency network. The network includes rules for deriving new attributes from input attributes and rules for determining which attributes of targets may be relevant to the recognition task.

The initial input to TRIPLE is a set of target examples represented as schemata. EBL examines the schemata, selects the best attributes for the recognition task, and tags them. The schemata are then sent to SCC for integration into the classification tree. Using simplicity as a quality measure, SCC takes the target schemata and constructs a tree. The classification tree is then passed to the model-matching component for use in identifying candidate image regions.

The model-matching component monitors the matching process and, when a failure occurs, decides what action is appropriate. If the model-matching component successfully traverses the classification tree, matching all necessary attributes, and arrives at a leaf node that contains a target model, the object has probably been correctly identified. If the matching component correctly parses the high-level nodes of the tree but exhausts the image data before it arrives at a leaf node, the problem probably lies in the image processing. The matching component can ask the segmentation and feature-extraction algorithms to re-examine a specific portion of the image,
using more relaxed parameter settings. If the matching component is not able to match any high-level nodes, but does match many low-level nodes, the target may be occluded. In this case, a tentative identification is accompanied by a confidence level that reflects the discriminant strength of the features that can be seen.

Cases that appear to call for model refinement or the acquisition of a new model lead to the re-invocation of the EBL component. If the matching component parses the tree without missing more than a few attributes but the values stored in the attributes do not match those derived from the image, the target model should probably be refined. If the model-matching component cannot match any nodes of the tree, it may be that a new target has been encountered. In both cases, EBL is provided with the object description that caused the failure and the failure type. In the first case, the EBL component may remove the tag from an attribute in a target schema, modify the value of an attribute, or add a new attribute. In the second case, a new target schema is created and the classification tree is re-ordered to accommodate the new schema.

**Semi-Autonomous Vehicles**

Once the autonomous vehicle has been given its marching orders, it is on its own. In contrast, the semi-autonomous vehicles being developed at Honeywell can be driven by remote control. In a cooperative effort, engineers at the Advanced Systems Center and at the Systems and Research Center have devised a tele-operation system for these vehicles called the Human Engineering Remote Driving System (HERDS).

The operator, who sits in a distant control center, receives video images from the vehicle's camera and navigational data from its inertial-reference unit over a narrow-bandwidth communications link. Although the video data are compressed before transmission, an image can be transmitted over the link only at three-second intervals. Because the vehicle must be able to travel at high speeds, this rate isn't high enough for manual control. On the other hand, there is space on the communications channel to transmit position updates as often as 50 times a second.

The gaps in the imagery are filled in by a warp processor developed by Honeywell that is located in the control center. The control center compresses the image data and then warps it to reflect the difference between the vehicle's position at the time the image was taken and its current position. The image is warped 30 times a second in real time, providing the operator with apparently continuous imagery.

One goal for semi-autonomous vehicles is that they be able to find their own way back to a home position after they have been driven out under remote control. The solution we are working on under a program called FOCUS can be regarded as a simplified version of PREACTE. The landmark-recognition subsystem will not be supplied with an internal map; instead it will learn landmarks on its outward trip, recording their features together with data from the inertial-reference unit. And instead of analyzing video imagery, FOCUS will analyze laser-radar images. Because we have done extensive work with such imagery, we hope to be able to make rapid progress with FOCUS.

**Acknowledgements**

The following researchers have contributed to the ALV work described in this report: Hatem Nasr, Peter Symo-sek, John Ming, Wilhelm Burger, Jon Kim, Stephanie Schaffer, David Krig, Aaron Larson, Chris Meier, Marc Gluch, Ajay Jain, Sung Kee Lee, Jeff Landay, Henry Lai, Tim Wittenburg, Dave Robohm, and Wing Au. Durga Panda, Steve Savitt, and Raj Aggarwal have provided strategic direction, managerial assistance, and support. The team members for the HERDS project include: Bill Barrett, Greg Cary, Jerry Coffel, Larry Linde, Pat Narendra, Dave Opheim, Steve Pratt, Brian Schipper, Chi Vu, Rand Whillock, and Sue Williams. Those working on the FOCUS program include: John Budenske, Mike Klein, Russ Richardson, Greg Saunders, Kris Siejko, and Sue Williams.

Summer 1988

67