QUALITATIVE TARGET MOTION DETECTION AND TRACKING

Bir Bhanu, Peter Symosek, John Ming, Wilhelm Burger, Hatem Nasr, and Jon Kim
Honeywell Systems and Research Center
3660 Technology Drive
Minneapolis, MN 55418

ABSTRACT

To detect and track moving targets using image information exclusively obtained from a vision system on-board an autonomous robotic vehicle, we present a comprehensive qualitative approach which allows for (a) determination of vehicle motion, (b) qualitative estimation of dynamic 3-D stationary structure, (c) detection and classification of the motion of individual objects in the scene using point, edge, and region features. The 3-D motion of targets is obtained from displacement vectors of point features without any knowledge about the underlying 3-D structure, discovering inconsistencies between the current state of the initial qualitative 3-D scene model and the changes actually observed in the scene, and by detecting moving edges and regions. We have also integrated map-based information into the system's reasoning framework. The digital map information, which consists of elevation, photographic, and terrain feature (roads, rivers, land cover, etc.), is used to predict target motion, track targets through occlusion, and assist in on/off road navigation by the robotic vehicle. The digital map information provides valuable clues when detecting moving targets in high clutter and low contrast environments. We present experimental results to demonstrate the capabilities of the system.

1. INTRODUCTION

Current multimode tracking approaches incorporate several possible techniques such as feature matching, correlation, centroid tracking, silhouette matching, and Kalman filtering for motion detection and tracking from a mobile platform.[2] These systems encounter problems in practical scenarios where conditions including arbitrary sensor platform motion, high clutter, low contrast, distant targets, sun angles (glare), variety of terrain features, vehicle maneuvers, countermeasures, target occlusion, and battlefield conditions pose significant threats to mission success. In some targeting scenarios, target location prediction is not enough for tracking; recognition of the target and clutter rejection algorithms may be required. At Honeywell, we are developing a comprehensive *qualitative* approach for target motion detection and tracking from a mobile platform.[3,4,6,8,10] Our objective is to achieve robust behavior in such a system. The key elements of our approach are shown in Figure 1.

The target motion detection and tracking problem can be viewed as the task of finding consistent and plausible 3-D interpretations for any change observed in the 2-D image sequence. Due to the motion of the Autonomous Land Vehicle (ALV), stationary objects in the scene generally do not appear stationary in the image, whereas moving objects are not necessarily seen in motion. The three main tasks in our approach for target motion detection and tracking are: (a) to estimate the vehicle's motion; (b) to derive the 3-D structure of the stationary environment; and (c) to detect and classify the motion of individual targets in the scene. These three tasks strongly depend on each other. The direction of travel (i.e. translation) and rotation of the vehicle are estimated with respect to stationary locations in the scene. The focus of expansion (FOE) is not determined as a particular image location, but as a region of possible FOE-locations called the *Fuzzy FOE*. We present a qualitative strategy of reasoning and modeling for the perception of 3-D space from motion information. Instead of refining a single quantitative description of the observed environment over time, multiple *qualitative interpretations* are maintained simultaneously.

The qualitative interpretations are built in three separate steps (see Figure 1). First, significant features (points, boundaries, corners, etc.) are extracted from the image and the 2-D displacement vectors are computed for this set of features. For the examples shown here, points were automatically selected and tracked between individual frames. The image database used to carry out the experiments is described in Section 2. The Interest Point detection and matching approach, which is a revised version of the Moravec interest point operator [18] and Barnard and Thompson's disparity analysis technique,[1] is optimized for low depression angle ALV imagery and is described in Section 3. During the second step, the vehicle's direction of translation, i.e. the Focus of Expansion (FOE), and the amount of rotation in space are determined. Almost all the necessary numerical computation is performed in the FOE computation stage, which is described in Section 4. The third step (2-D Change Analysis) constructs an internal 3-D model of the scene. Section 5 outlines the concepts and operation of this Qualitative Scene Model. Scene interpretations obtained from Qualitative Reasoning are validated with geometric reasoning and validation rules employing an auxiliary map database. The details of the map-based target tracking approach developed for the Scene Dynamics program are presented in Section 6. A preprocessing stage for the detection of rapidly moving objects in the field of view is incorporated.

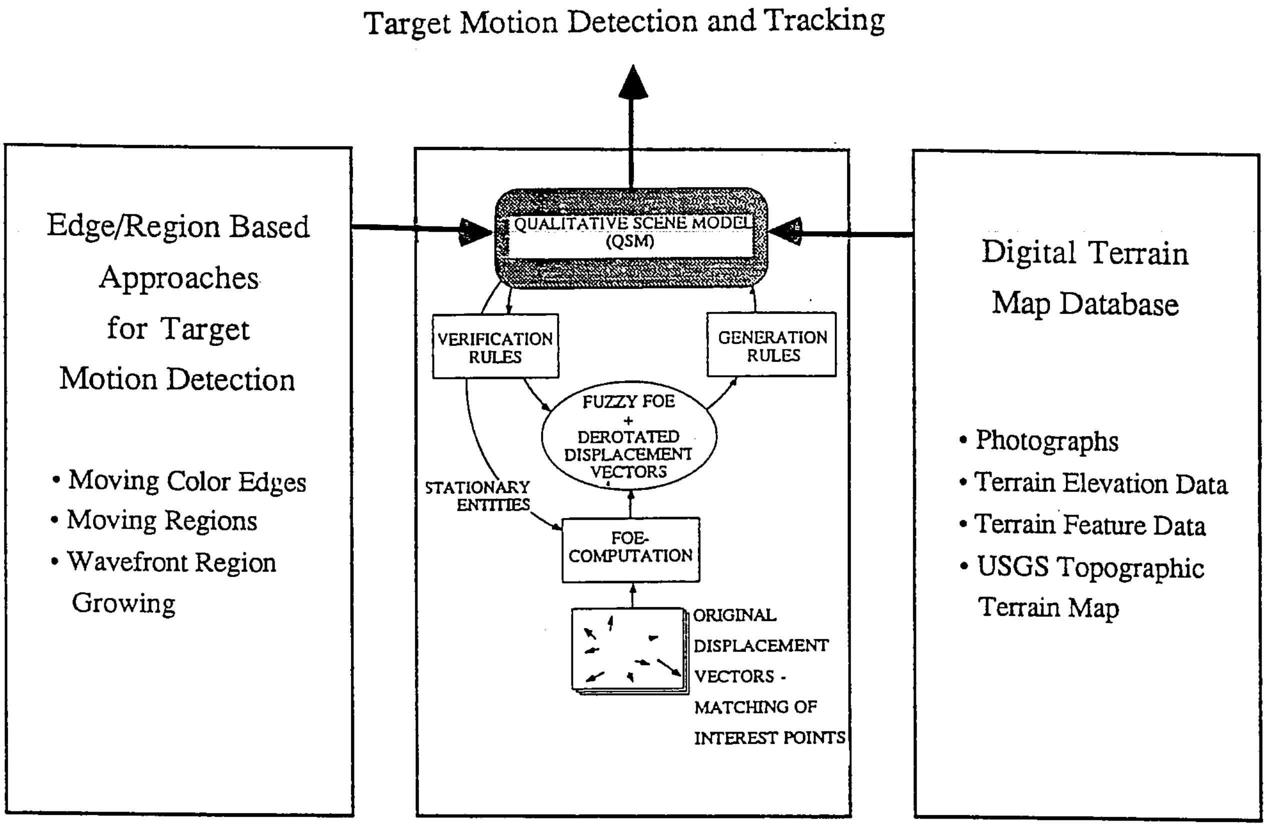

Target Motion Detection and Tracking
Edge/Region Based Approaches for Target Motion Detection
• Moving Color Edges
• Moving Regions
• Wavefront Region Growing
QUALITATIVE SCENE MODEL (QSM)
VERIFICATION RULES
GENERATION RULES
FUZZY FOE + DEROTATED DISPLACEMENT VECTORS
STATIONARY ENTITIES
FOE-COMPUTATION
ORIGINAL DISPLACEMENT VECTORS - MATCHING OF INTEREST POINTS
Digital Terrain Map Database
• Photographs
• Terrain Elevation Data
• Terrain Feature Data
• USGS Topographic Terrain Map

Figure 1: Qualitative Reasoning and Modeling approach for target motion detection and tracking. From the original displacement vectors (obtained by matching corresponding point features), the Fuzzy FOE and the derotated displacement field are computed. The Qualitative Scene Model (QSM) is built in a hypothesize-and-test cycle by two sets of rules. Generation rules search for significant image events and place immediate conclusions (hypotheses) in the QSM. A set of environmental entities that are believed to be stationary is supplied by the QSM for use in the FOE computation. A digital map database interacts with the QSM to detect and track moving targets in the image. Edge/region based approaches are used detect rapidly moving objects at close range.

We have performed experiments on edge and region-based approaches for target motion detection in color imagery, to assist in obtaining robust performance from the Qualitative Reasoning system. The technical details of these algorithms are described in Section 7. Finally, in Section 8, the conclusions of this paper and our plans for future enhancements of the qualitative motion detection and tracking approach are presented.

2. DATABASE OF IMAGES

In order to verify the capabilities of the system shown in Figure 1, a large image database was generated and processed. In subsequent sections of this paper, we will illustrate some of the results. The processing consisted of five stages:

(1) Interest Point Detection,

(2) Disparity Analysis,

(3) Qualitative Reasoning for Motion Detection and Tracking,

(4) Hypothesis Verification using Auxiliary Map Information, and

(5) Edge/Region Motion Detection at Close Ranges.

The image database contains five sequences which represent configurations of the imaging system and target that often prove to be extremely challenging for traditional tracking approaches. The five configurations are:

(1) Motion detection for targets traveling directly towards and away from the ALV.

(2) Tracking targets through total occlusion.

(3) Tracking targets in high clutter.

(4) Tracking targets at long ranges.

(5) Tracking system verification with auxiliary map information.

The images were obtained from the ERIM/Martin Marietta Collage I database. Examples of each of the five configurations could be found in multiple sequences, but the objective was to resolve just one problematic scenario per sequence. Three examples of images from one of the five sequences are presented in Figure 2. The images were digitized at 0.5 second intervals. The images were digitized as gray scale because the demonstration of target tracking requires only *Interest Points* or points of significant change in multiple directions in the intensity function of the image and these locations can be detected using the luminance or Y component of the NTSC television signal. When moving edges or regions are employed for matching by the Disparity Analysis algorithm, color information is used as well.

The images were preprocessed with a 3x3 window average filter to attenuate digitization noise. The noise was due to a lack of sufficient bandwidth in the digitizer and resulted in 1 pixel duration pattern noise in the signal output. The 3x3 window average filter was sufficient for the attenuation of the noise so that the Interest Point detection algorithm performed reliably.

3. COMPUTATION OF DISPLACEMENT VECTORS

3.1 SELECTION OF INTEREST POINTS

The first stage of the motion algorithm suite is the detection of *Interest Points* or points where the image's intensity demonstrates a discontinuity in multiple directions of the eight neighbor directions of the quadruled pixel grid. Discontinuities of this kind are indicative of corners in 3-space. These locations are detected with a modified Moravec Interest Operator.[17,18] The operator derives "interestingness" employing the same approach that the Moravec Operator utilizes, but the revised operator is adaptive for expected range-dependent image features. The window dimension for the modified Interest Operator is established with

$$H = W = w_0 + 2\left\lfloor \frac{i * Int\_SF}{512} \right\rfloor , \qquad (1)$$

where

H = the height of the window,

W = the width of the window,

w_0 = the window dimension for the first row of the image,

i = the row coordinate of the current row of the image, $1 <= i <= 512$,

Int_SF = a scale factor that controls how rapidly the detection operator window dimension varies with range,

$\lfloor x \rfloor$ = the greatest integer less than or equal to x.

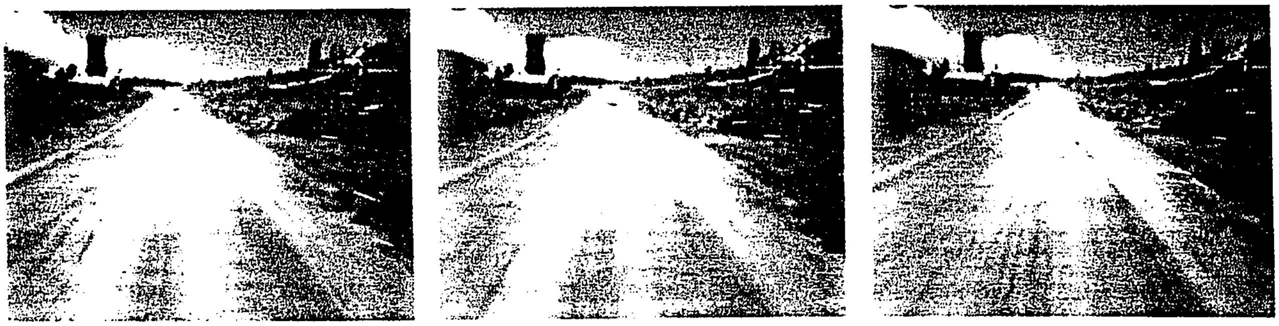


Figure 2: Three images of a sequence in which a car is approaching the ALV and another car is receding at a distance. This figure also shows the displacement vectors (by white lines) for Interest Points.

The approach, given by equation (1), essentially realizes a range adaptation capability by changing the window dimensions as a function of row coordinate. For low depression angle imagery, the range to 3-dimensional objects of the scene is approximately inversely proportional to the row coordinate of the image of the object. The resolution of scene regions that are seen in the lower portions of the image is greater than that of regions that are seen in the upper portions of the image because they are nearer to the ALV. Therefore, to detect features in these regions, the window dimension must increase to encompass enough image area for the entire neighborhood of the feature to be seen.

Because the scenes that comprise the NTSC video portion of the Collage I database are all on-road, the Moravec Interest Operator[18] detects fewer discontinuities in the higher resolution regions for several reasons. First, because of the rural nature of the Martin Marietta ALV Test Site, there are very few man-made objects by the side of the road (which normally exhibit good quality Interest Points), except for a few guard rails and telephone poles. Second, due to the scarcity of high quality interest points, the locations that are detected by the Moravec Interest Operator are often from regions displaying arbitrary textures such as small rocks or isolated stands of grass. The Moravec Interest Operator employs a cut off quota for point feature detection: The point features detected by the Interest Operator are stored in a list sorted in descending order of rating (a heuristic measure of the strength of the discontinuity). Only those interest points whose rating is greater than a cut off threshold are transferred to the next stage. Therefore, very few interest points for the lower regions of the image are archived in the locations of the list above the cut off quota.

In order to guarantee that the Interest Points the operator detects are not biased to any specific range, the revised operator does the following: The revised version of the algorithm uses a varying quota, where the quota is calculated as a function of the line coordinate of the image. This quota permits only $\frac{i}{512}$ * *Cut_Off* Interest Points to be detected for the first i lines of the image, where *Cut_off* is the cut off level of the original approach. When the number of Interest Points detected for the first i lines of the image is less than $\frac{i}{512}$ * *Cut_Off*, new Interest Points' quantified ratings are stored on the list in the usual fashion. When more than $\frac{i}{512}$ * *Cut_Off* Interest Points are detected for the first i lines of the image, these locations are integrated into the list only if their rating is greater than the ratings of the previously stored Interest Points. Otherwise, the new Interest Points are discarded.

3.2 DISPARITY ANALYSIS

The second stage of the Interest Point displacement estimation approach is disparity analysis. This algorithm is a revised version of the Barnard-Thompson Disparity Analysis algorithm.[1] The major difference between the Barnard-Thompson algorithm and the revised algorithm is enhanced feature matching that employs range adaptation. The phases of the algorithm where range adaptation is used are: 1) Neighbor Search, 2) Initial Match Likelihood Calculation, and 3) Relaxation. The revisions for each of these stages will be explained in the following sub-sections.

3.2.1 Neighbor Search

To restrain the size of the total set of candidate disparities and therefore, to diminish the computational complexity of the algorithm, the group of current frame Interest Points to which a previous frame Interest Point can be matched are those Interest Points lying in a W_S by H_S window centered at the previous frame location. The vertical and horizontal dimensions of this window change with the line coordinate of the previous frame Interest Point. The approach used to calculate the dimensions of this window is to estimate the maximum expected location change or disparity for an object at ground level for the imaging geometry of the ALV (as a function of line coordinate) and then empirically approximate the relation with an exponential function. The Neighbor Search window's dimension, obtained with this approach,[17] is:

$$W_S = H_S = 2\left\lfloor C_1 \exp[C_2 * i] + C_3 \right\rfloor + 1 \tag{2}$$

where

W_S = the width dimension of the Neighbor Search window,

H_S = the height dimension of the Neighbor Search window,

$C_1 = 8.32 * S_{MAX}^{(-0.018868)}$,

$$C_2 = \frac{\log\left[\frac{S_{MAX}}{8}\right]}{424} ,$$

$C_3 = w_0$, the Interest Operator for the first line of the image,

S_{MAX} = the maximum search window size for the near-field of the image (maximum expected object displacement),

i = the row coordinate of the Interest Point in the previous image.

3.2.2 Initial Match Likelihood Calculation

The window dimension for this stage is calculated by

$$W_M = H_M = 2\left[m_0 + \left\lfloor \frac{i * IM\_SF}{512} \right\rfloor\right] + 1 , \qquad (3)$$

where

m_0 = Initial Match Likelihood window dimension for the first line of the image,
IM_SF = the Initial Match scale factor, and
i = the line coordinate of the current line.

The Initial Match Likelihood is calculated as a function of a heuristic measure of the correlation of the image intensities of the previous and current frames in windows centered at the location of the Interest Point in each frame. This heuristic measure will be explained in the latter paragraphs of this section. Because the relaxation algorithm used to refine the estimates of the disparity likelihoods employs probabilities (i.e. each disparity $l = [l_x, l_y]$ is characterized by a number $p(l)$, where $p(l)$ is an element of the range [0,1] and $\sum_l p(l) = 1$), the Initial Match Likelihood is transformed into a probability using normalization. The heuristic measure is calculated for every candidate disparity l_j ; $j = 1,..., J_i$, where J_i is the total number of disparities identified in the neighbor search for a specific Interest Point i. Employing the notation of relaxation labeling schemes, each previous frame Interest Point represents a node or object in the object space of the iterative approximation procedure. Let a_i, $i = 1,...,Cut\_Off$ represent each node of the object space. The labels for each node a_i are l_j, $j = 1,..., J_i$ and an undefined disparity l^* (defined in the next paragraph). The set of all labels, l_j, $j = 1,..., J_i$, and l^* is denoted L_i. The objective of the iterative approximation procedure is to use evidence obtained from neighboring nodes of the object space to refine the estimated disparity likelihoods until one likelihood is approximately 1 and the rest 0, i.e. each object is uniquely labeled.

Because Interest Points are not detected with 100% certainty in each image, it is also possible that no valid match exists for each previous frame Interest Point. To account for this configuration, a non-match disparity l^* is defined. The likelihood $p_i(l^*)$ for node a_i represents the likelihood that the node i has no match. The first stage needed to transform heuristic correlation measures into probabilities is the calculation of the initial probability of the disparity $p_i(l^*)$. Using the approach of Barnard and Thompson,[1] the probability that an Interest Point is not matchable is approximated as $1 - w_i(l_{MAX})$, where $w_i(l_{MAX})$ is the heuristic measure of largest magnitude for node a_i. The assumption is justified due to the fact that the label of maximum weight is, in general, the correct one. We have verified this assumption for low-depression angle ground-to-ground imagery by empirically calculating the Initial Match Likelihood heuristic measure for a large number of frames and then tabulating the number of times that the true disparity was the one with the greatest measure.

The next stage is the application of Bayes Rule to obtain an initial estimate of the probability that a_i should be labeled l for labels other than l^*. This calculation is carried out as follows:

$$p_i^0(l_j) = p_i(l_j \mid i) * (1 - p_i^0(l^*)); \quad l_j \neq l^*. \qquad (4)$$

where

$p_i^0(l_j)$ = the Initial Match Likelihood for disparity l_j of node a_i,
$p_i(l_j \mid i)$ = the conditional probability that a_i has label l_j given that a_i is matchable,
$(1 - p_i^0(l^*))$ = the probability that a_i is matchable.

The quantities $p_i(l_j \mid i)$ are estimated with

$$p_i(l_j \mid i) = \frac{w_i(l_j)}{\sum_{l_k \neq l^*} w_i(l_k)}. \qquad (5)$$

In order to guarantee that the revised Disparity Analysis algorithm would perform correctly for a wide variety of scenarios, the following was done: A variety of match measures were evaluated for sequence 1 and the measure which produced the best results in terms of a qualitative visual evaluation of the estimated disparities was used to process the remaining four sequences. The qualitative visual evaluation was carried out by creating a pseudo-colored image (where one frame is displayed with the color green, the second is displayed with the color red and the estimated disparities are displayed with the color blue) and then visually checking the validity of the matches. The Initial Match measure which was judged to be the best, because it produced the fewest errors for a wide variety of scenarios, was

$$w_i(l_j) = \frac{1}{1 + C * S}\left[\frac{1 + L * I_k}{1 + L}\right]; \; i = 1,...,Cut\_Off; \; j = 1,...., J_i , \qquad (6)$$

where

$w_i(l_j) =$ the initial match evidence of a candidate disparity l_j,

$I_k =$ the k'th Interest Point's rating, where the k'th Interest Point is the current frame location of the feature that defines disparity l_j, and $0 \leq I_k \leq 1$,

$Cut\_Off =$ number of Interest Points detected in the previous frame,

$J_i =$ total number of potential disparities for Interest Point i,

$C, L =$ constants, and

$$S = \frac{\sum_{j,k \,\varepsilon\, N} \left[g_{j_0+j,\, k_0+k} - h_{j_1+j,\, k_1+k} \right]^2}{\sum_{j,k \,\varepsilon\, N} 1} ,$$

where

$g_{m,n} =$ intensity of the previous frame at the location $\{m,n\}$,

$h_{m,n} =$ intensity of the current frame at the location $\{m,n\}$,

$\{j_0, k_0\} =$ previous frame coordinate of the point feature for disparity l_j,

$\{j_1, k_1\} =$ current frame coordinate of the point feature for disparity l_j, and

$N =$ Initial Match region for disparity l_j.

3.2.3 Relaxation Region

With a relaxation labeling scheme, the valid disparities with low initial likelihoods are elevated in magnitude by the correlated evidence of their neighbors in the object space of the scene. By correlated evidence, it is meant that the disparities of neighboring nodes exhibit approximately the same magnitude and direction displacement. A node is a neighbor of another if it lies within the relaxation window centered at that node. Therefore, the correlated evidence for a specific disparity l_j of a_i is calculated as the sum of the disparity likelihoods for all neighbors of a_i, where the disparities of the neighbors have approximately the same magnitude and orientation as l_j. The degree of mismatch between the neighboring node's disparities and l_j is defined in terms of specific error thresholds for orientation and displacement. The Relaxation Region window's dimensions are calculated with this equation:

$$W_R = H_R = 2 \left[r_0 + \left\lfloor \frac{i * R\_SF}{512} \right\rfloor \right] + 1 , \tag{7}$$

where

$W_R =$ the width dimension of the Relaxation Region window,

$H_R =$ the height dimension of the Relaxation Region window,

$r_0 =$ the Relaxation Region window dimension for the first line of the image,

$R\_SF =$ the scale factor for adaptation of the Relaxation Region window dimension with respect to range, and

$i =$ the line coordinate of the current line of the image.

The iterative update rule for disparity probabilities, employed at each stage of the relaxation algorithm, is

$$\tilde{P}_i^k(l_j) = P_i^{k-1}(l_j) * (B + A * G^k(l_j)) \; ; j = 1,\ldots, J_i \tag{8}$$

$$\tilde{P}_i^k(l^*) = P_i^{k-1}(l^*)$$

$$P_i^k(l_j) = \frac{\tilde{P}_i^k(l_j)}{\sum_{l \,\varepsilon\, L_i} \tilde{P}_i^k(l_m)}$$

where

$P_i^k(l_j) =$ the probability that a specific hypothesized disparity l_j, at the k'th iteration, is the true disparity,

$P_i^{k-1}(l_j) =$ the probability that a specific hypothesized disparity l_j, at the (k-1)st iteration, is the true disparity,

$G^k(l_j) =$ the correlated evidence obtained from other disparities lying in the Relaxation Region at iteration k,

$A, B =$ constants that control the rate of convergence of the iterative procedure, and

$R_i =$ the W_R x H_R Relaxation Region window centered at the previous coordinate of the point feature.

For the preceding equation, terms with tildes (˜'s) are evidences. Evidences are not constrained to be elements of the range [0,1] and therefore are not probabilities. Normalizing these terms by the total weight of all the evidences for a specific domain transforms evidences into probabilities.

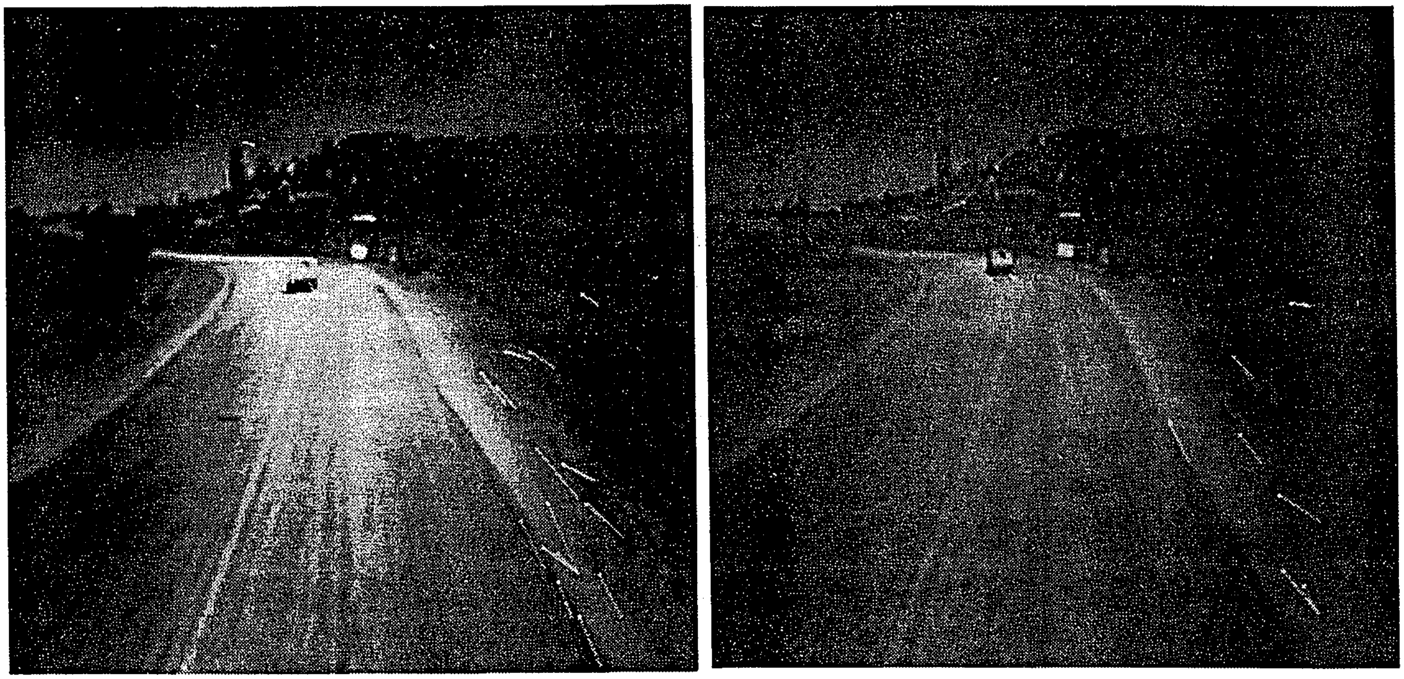
3.3 RESULTS FOR DISPLACEMENT VECTORS

The adaptive window approach was validated with several sequences of real ALV imagery. Figure 2 shows the displacement vectors (white lines) for a few frames of a sequence in which the ALV was traveling along a paved road and, during the first half of the 8 second sequence, a vehicle passes the ALV on the left and, during the last half of the sequence another vehicle approaches the ALV in the opposite lane. Another example, given in Figure 3, demonstrates the estimated disparities for the points employed for matching for two successive frames of a sequence. The Interest Points were ranked on the basis of their interestingness, and the best 50 were selected, subject to the constraint that for each line of the image with coordinate *i*, no more than (*i* / 511) * 50 points were selected. By choosing a greater number of Interest Points for the lower half of the image than the upper half, the Interest Points detected represent valid object features and not arbitrary textures. The interest operator window size varied from 4x4 to 8x8. For a specific y coordinate, the interest operator window size is computed by

$$\text{Window Size}_X = \text{Window Size}_Y = 4 + \frac{4 * i}{512} \tag{9}$$

The Neighbor Search window size varied from 4x4 to 120x120, the Initial Match Likelihood Calculation window size varied from 5x5 to 7x7, and the Relaxation Region window size varied from 32x32 to 128x128.

Another series of experiments was carried out with imagery obtained from a video camera on-board a low flying rotorcraft. The detection of obstacles such as low-hanging branches, trees, and power lines is critical for rotorcraft engaged in flying in Nap-Of-the-Earth (NOE) courses due to the substantial clearance required by their rotors. Figure 4 demonstrates the automated detection of Interest Points, the derivation of the valid disparities, and the calculation of the Fuzzy Focus of Expansion (FOE) (discussed in the next section). Each frame of the 246 frame database is processed with the revised Moravec Interest Operator and the revised Disparity Analysis algorithm for the estimation of disparities. The array of disparities for each pair of frames is transferred to the FOE computation stage.



(a)

(b)

Figure 3: Selection of Interest Points and disparity analysis. Displacement vectors are indicated by white lines. Note the displacement vectors present on the moving cars.

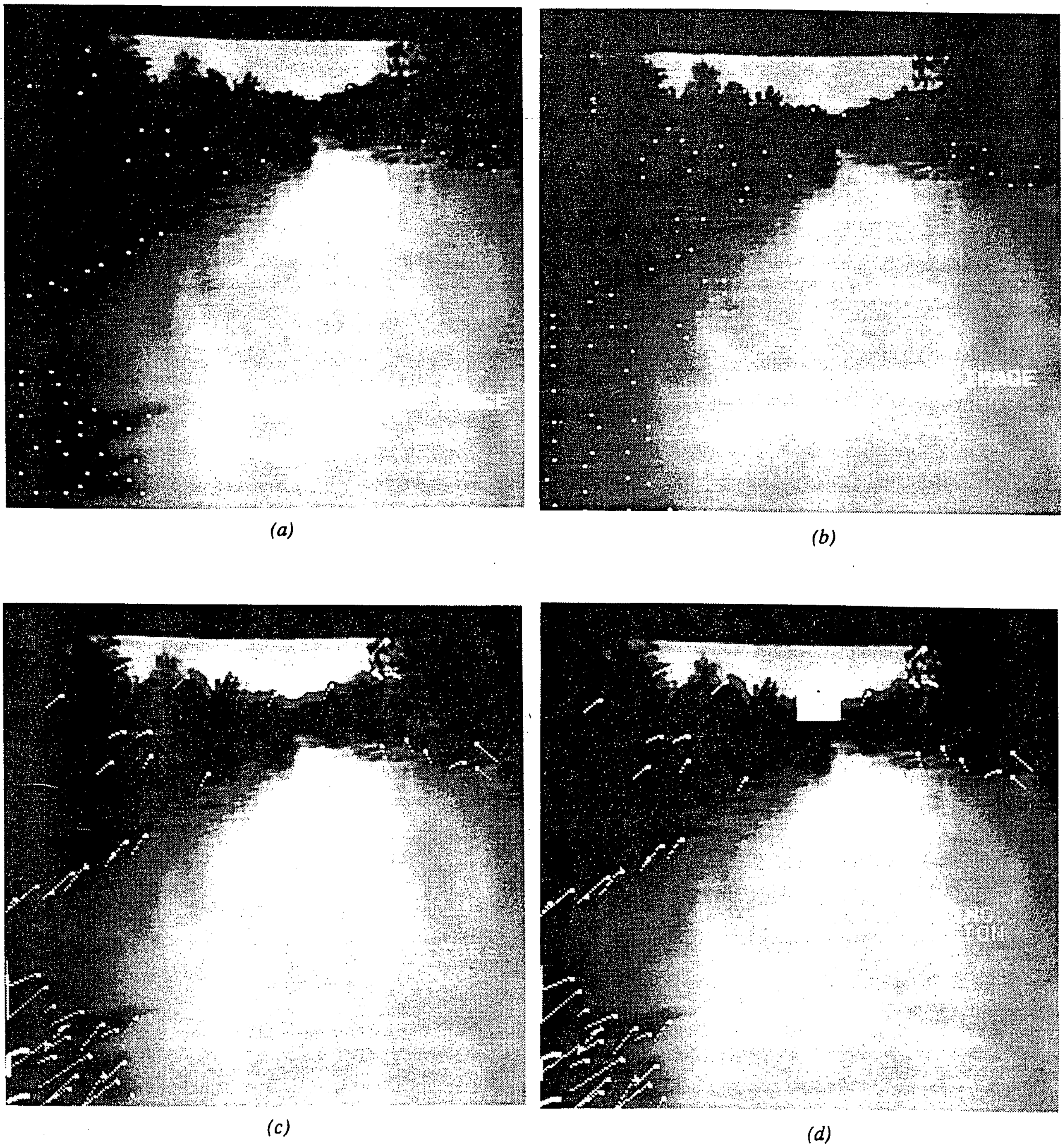


Figure 4: Disparity estimation for Nap-Of-the-Earth (NOE) rotorcraft data. *(a)* Interest Points in frame 1. *(b)* Interest Points in frame 2. *(c)* Displacement vectors. *(d)* Computation of Fuzzy Focus of Expansion. Fuzzy FOE is shown by the white area near the horizon. The black dot in the center indicates the best estimate of the FOE.

4. COMPUTATION OF THE FUZZY FOCUS OF EXPANSION

For short time intervals, the 3-D motion **M** of the ALV can be modeled by a translation **T** followed by a rotation $\mathbf{R}_\theta$ about the Y-axis and a rotation $\mathbf{R}_\phi$ about the X-axis:[9, 10]

$$\mathbf{M} = \mathbf{R}_\phi \, \mathbf{R}_\theta \, \mathbf{T} \ . \tag{10}$$

This results in a mapping **d** from the original image $\mathbf{I}_0$ at time t_0 into the new image $\mathbf{I}_1$ at time t_1.

$$\mathbf{d} : \mathbf{I}_0 \rightarrow \mathbf{I}_1 \ = \ \mathbf{r}_\phi \, \mathbf{r}_\theta \, \mathbf{t} \, \mathbf{I}_0 \ = \ \mathbf{r}_\phi \, \mathbf{r}_\theta \, \mathbf{I}_0' \ . \tag{11}$$

The intermediate image $\mathbf{I}_0'$ in (11) is the result of the translation component of the vehicle's motion and has the property of being a radial mapping, which deterministically is represented as:

$$\mathbf{t} = \{ \ (\mathbf{x}_i \, , \mathbf{x}_i') \in \mathbf{I} \times \mathbf{I}' \mid \mathbf{x}_i' = \mathbf{x}_i + \mu_i \, (\mathbf{x}_i - \mathbf{x}_f) \ , \mu_i \in R, \ \mu_i \geq 0 \ \} \ . \tag{12}$$

Unlike the two images $\mathbf{I}_0$ and $\mathbf{I}_1$, which are actually given, the image $\mathbf{I}_0'$ is generally not observed, except when the camera rotation is zero. It serves as an intermediate result to be reached during the separation of translational and rotational motion components. The fact that

$$\mathbf{I}_0' = \mathbf{r}_\theta^{-1} \, \mathbf{r}_\phi^{-1} \, \mathbf{I}_1 = \mathbf{t} \, \mathbf{I}_0 \tag{13}$$

suggests two different strategies for separating the motion components:

(1) **FOE from Rotation**: Successively apply combinations of inverse rotation mappings $\mathbf{r}_{\theta_1}^{-1} \, \mathbf{r}_{\phi_1}^{-1}$, $\mathbf{r}_{\theta_2}^{-1} \, \mathbf{r}_{\phi_2}^{-1}$,... $\mathbf{r}_{\theta_k}^{-1} \, \mathbf{r}_{\varphi_k}^{-1}$ to the second image $\mathbf{I}_1$, until the resulting image $\mathbf{I}'$ is a radial mapping with respect to the original image $\mathbf{I}_0$. Then locate the FOE $\mathbf{x}_{f_k}$ in $\mathbf{I}_0$.

(2) **Rotation from FOE**: Successively select FOE-locations (different directions of vehicle translation) $\mathbf{X}_{f_1}$, $\mathbf{X}_{f_2}$,... $\mathbf{X}_{f_l}$ in the original image $\mathbf{I}_0$ and then determine the inverse rotation mapping $\bar{\mathbf{r}}_{\theta_l}^{-1} \, \bar{\mathbf{r}}_{\phi_l}^{-1}$ that yields a radial mapping with respect to the given FOE $\mathbf{x}_{f_l}$ in the original image $\mathbf{I}_0$.

Both alternatives were investigated under the assumption of restricted, but realistic vehicle motion. It turned out that the major problem in the **FOE-from-Rotation** approach is to determine if a mapping of image points is (or is close to being) radial when the location of the FOE is unknown. Of course, in the presence of noise, this problem becomes even more difficult. The second approach was examined after it appeared that any method which extends the given set of displacement vectors *backwards* to find the FOE is inherently sensitive to image degradations.

Although there have been a number of suggestions for FOE-algorithms in the past,[15, 20, 22] no results of implementations have been demonstrated on real outdoor imagery. One reason for the absence of useful results might be that most researchers have tried to locate the FOE in terms of a single, distinct image location. In practice, however, the noise generated by merely digitizing a perfect translation displacement field may keep the resulting vectors from passing through a single pixel. Even for human observers it seems to be difficult to determine the exact direction of heading (i.e., the location of the FOE on the retina). Average deviation of human judgement from the real direction has been reported[21] to be as large as 10° and up to 20° in the presence of large rotations.

It was, therefore, an important premise in this work that the final algorithm should determine an *area* of potential FOE-locations (called the *Fuzzy FOE*) instead of a single (but probably incorrect) point. The method described below avoids the problem mentioned above by guessing an FOE-location first and estimating the optimal derotation for this particular FOE in the second step.

4.1 FUZZY FOE ALGORITHM

Given the two images $\mathbf{I}_0$ and $\mathbf{I}_1$ of corresponding points, the main algorithmic steps of this approach are:[9]

(1) Guess an FOE-location $\mathbf{x}_f^{(i)}$ in image $\mathbf{I}_0$ (for the current iteration i).

(2) Determine the derotation mapping $\mathbf{r}_\theta^{-1}$, $\mathbf{r}_\phi^{-1}$ which would transform image $\mathbf{I}_1$ into an image $\mathbf{I}_1'$ such that the mapping $(\mathbf{x}_f^{(i)}, \mathbf{I}_0, \mathbf{I}_1')$ deviates from a radial mapping with minimum error $\mathrm{E}^{(i)}$.

(3) Repeat steps *(1)* and *(2)* until an FOE-location $\mathbf{x}_f^{(k)}$ with the lowest minimum error $\mathrm{E}^{(k)}$ is found.

An initial *guess* for the FOE-location is obtained from knowledge about the orientation of the camera with respect to the vehicle. For subsequent pairs of frames, the FOE-location computed from the previous pair can be used as a starting point. Once a particular $\mathbf{x}_f$ has been selected, the problem is to compute the rotation mappings $\mathbf{r}_\theta^{-1}$ and $\mathbf{r}_\phi^{-1}$ which, when applied to the image $\mathbf{I}_1$, will result in an optimal radial mapping with respect to $\mathbf{I}_0$ and $\mathbf{x}_f$.

To measure how close a given mapping is to a radial mapping, the perpendicular distances between points in the second image $(\mathbf{x}_i')$ and the "ideal" displacement vectors is measured. The sum of the squared perpendicular distances d_i

is the final error measure. For each set of corresponding image points ($\mathbf{x}_i \in \mathbf{I}$, $\mathbf{x}_i' \in \mathbf{I}'$), the error measure is defined as:

$$E\ (\mathbf{x}_f) = \sum_i E_i = \sum_i d_i^2 = \sum_i \left[\frac{1}{|\overrightarrow{\mathbf{x}_f\mathbf{x}_i}|}\ \overrightarrow{\mathbf{x}_f\mathbf{x}_i} \times \overrightarrow{\mathbf{x}_f\mathbf{x}_i'} \right]^2 . \tag{14}$$

The final algorithm for determining the direction of heading as well as horizontal and vertical camera rotations is the following:

(1) Guess an initial FOE $\mathbf{x}_f^0$, for example the FOE-location obtained from the previous pair of frames.

(2) Starting from $\mathbf{x}_f^0$, search for a location $\mathbf{x}_f^{opt}$ where $E_{min}(\mathbf{x}_f^{opt})$ is a minimum. A technique of *steepest descent* is used, where the search proceeds in the direction of least error.

(3) Determine a region around $\mathbf{x}_f^{opt}$ in which the error is below some threshold.

The error function $E(\mathbf{x}_f)$ is computed in time proportional to the number of displacement vectors *N*. The final size of the FOE-area depends on the local shape of the error function and can be constrained not to exceed a certain maximum *M*. Therefore, the time complexity is *O(MN)*.

4.2 RESULTS FOR FOE COMPUTATION

Figure 4(d) shows the computation of Fuzzy FOE for the data taken from a rotorcraft during Nap-Of-the-Earth flight. Fuzzy FOE is shown by the white area near the horizon. The black dot in the center indicates the best estimate of the FOE.

5. QUALITATIVE REASONING AND MODELING FOR MOTION DETECTION AND TARGET TRACKING

5.1 QUALITATIVE REASONING TECHNICAL APPROACH

The choice of a suitable scheme for the internal representation of the scene is of great importance. The *Qualitative Scene Model* (QSM) is a 3-D camera-centered interpretation of the scene that is built incrementally from visual information gathered over time. The nature of this model, however, is *qualitative* rather than a precise geometric description of the scene. The basic building blocks of the QSM are *entities*, which are the 3-D counterparts of the 2-D *features* observed in the image. For example, the point feature *A* located in the image at *x,y* at time *t*

(Point_Feature *A t x y*)

has its 3-D counterpart in the model as

(Point_Entity *A*).

Since the model is camera-centered ("retinocentric"), the image locations and 2-D movements of features are implicitly part (i.e., known facts) of the model. Additional entries are the properties of entities (e.g. "stationary" or "mobile") and relationships between entities (e.g. "closer"), which are not given facts but hypotheses about the real scene. This is expressed in the model as either

(Stationary *entity*) or (Mobile *entity*) .

It is one of the key features of the QSM that it generally contains not only one interpretation of the scene, but a (possibly empty) *set* of interpretations which are all pursued simultaneously. At any point in time, a hypothesis is said to be "feasible" if it exists in the QSM and is not in conflict with some observation made since it was established.

Interpretations are structured as an inheritance network of partial hypotheses. Individual scene interpretations are treated as "closed worlds", i.e., a new conclusion only holds within an interpretation where all the required premises are true. Interpretations are also checked for internal consistency, e.g. entities cannot be both stationary *and* mobile within the same interpretation. The QSM is maintained through a generate-and-test process as the core of a rule-based blackboard system. The two major groups of rules are: *Generation Rules* and *Verification Rules.*

Generation Rules

Generation rules examine the (derotated) image sequence for significant changes and modify each interpretation in the QSM. Some of these observations have unconditional effects upon the model. For example, if an image feature is found to be moving *towards* the Fuzzy FOE (instead of diverging away from it), then it belongs to a moving entity in 3-D space. The actual rule contains only one premise and asserts (MOBILE ?x) as a global fact (i.e., it is true in every interpretation):

```
(defrule DEFINITE_MOTION
  (MOVING_TOWARDS_FOE ?x ?t)
 =>
  (at ROOT (assert (MOBILE ?x)))) /*a global fact*/
```

The directive "at ROOT" places the new fact at the root of the interpretation graph, i.e., it is inherited by all existing interpretations.

Other observations depend upon the facts that are currently true in a "world" and, therefore, may have only local consequences inside particular interpretations. For example, if two image features *A* and *B* lie on opposite sides of the Fuzzy FOE and they are getting closer to each other, then they must be in relative motion. If an interpretation exists that considers at least one of the two entities (x,y) stationary, then (at least) the other entity cannot be stationary (i.e., it must be mobile). The following rule "fires within" each interpretation that considers the first entity (x) stationary:

```
(defrule RELATIVE_MOTION
  (OPPOSITE_FOE ?x ?y ?t) /* first observation */
  (CONVERGING ?x ?y ?t)   /* second observation */
  (STATIONARY ?x) /* true inside an interpretation */
 =>
  (assert (MOBILE ?y)))   /* local to this interpretation */
```

Verification Rules

While the purpose of the generation rules is to establish new hypotheses and conclusions, the purpose of *verification rules* is to review interpretations after they have been created and, if possible, prove that they are false. When a hypothesis is found to be inconsistent with some new observation, it is usually removed from the QSM. Any interpretation that is based on such a hypothesis is removed simultaneously. Since we are always trying to come up with a single (and hopefully correct) scene interpretation, this mechanism is important for pruning the search tree.

Verification rules are typically based on image observations that, used as generators, would produce a large number of unnecessary conclusions. For example, the general layout of the scene seen from the top of a land-based vehicle suggest the rule of thumb that things which are *lower* in the image are generally closer to the camera. Although this rule is not strong enough to draw direct conclusions, it may be used to verify existing hypotheses:

```
(defrule LOWER_IS_CLOSER_HEURISTIC
  (CLOSER ?x ?y)
  (BELOW_THE_HORIZON ?x ?t)
  (BELOW_THE_HORIZON ?y ?t)
  (BELOW ?y ?x  ?t)
 =>
  /*mark this interpretation as conflicting*/
  (assert (CONFLICT LOWER/CLOSER ?x ?y))).
```

Whenever an existing hypothesis (CLOSER ?x ?y) violates the above rule of thumb, this rule fires and marks the interpretation as conflicting. How the conflict is eventually resolved depends upon the global state of the QSM. Simply removing the afflicted interpretation would create an empty model if this interpretation was the only one. This task is handled by a set of dedicated *conflict resolution rules*.[3]

The kind of rules described up to this point are mainly based upon the geometry of the imaging process, i.e., perspective projection. Other important visual clues are available from occlusion analysis, perceptual grouping, and semantic interpretation. *Occlusion* becomes an interesting phenomenon when features of higher dimensionality than points are employed, such as lines and regions. Similarities in form and motion found by *perceptual grouping* allow us to assemble simple features into complex objects. Finally, as an outcome of the recognition process, *semantic* information may help to disambiguate the scene interpretation. If an object has been recognized as a building, for example, it makes every interpretation obsolete that considers this object mobile. For all these various lines of reasoning, the QSM serves as a common platform.

Meta Rules

In summary, the construction of the QSM and the search for the most plausible scene interpretation are guided by the following meta rules:

- Always tend towards the "most stationary" (i.e. most conservative) solution. By default all new entities are considered stationary.
- Assume that an interpretation is feasible unless it can be proved to be false (the principle of "lack of conflict").
- If a new conclusion causes a conflict in one but not in another current interpretation, then remove the conflicting interpretation.
- If a new conclusion cannot be accommodated by any current interpretation, then create a new, feasible interpretation and remove the conflicting ones.

More details about QSM and rules are given in the Dynamic Reasoning using Integrated Visual Evidence (DRIVE) technical report.[3]

5.2 QUALITATIVE REASONING RESULTS

The computation of the Fuzzy Focus of Expansion and Qualitative Scene Model are implemented on a Symbolics 3670. Qualitative Reasoning is implemented in a knowledge-based system development environment, called the Automated Reasoning Tool (ART), which is supplied by Automated Inference Corp.

Figure 5 presents four frames from the Collage I database, where a vehicle is seen traveling down on the very distant roadway in the top right corner of the image. The distance from the ALV to the other vehicle is several hundred feet. The four frames were processed and the estimated Interest Point disparities between the first and second frames, the second and third frames and the third and fourth frames are shown in Figures 6(a)-(c). Note that the target is detected in the images even though it is quite far away from the ALV.

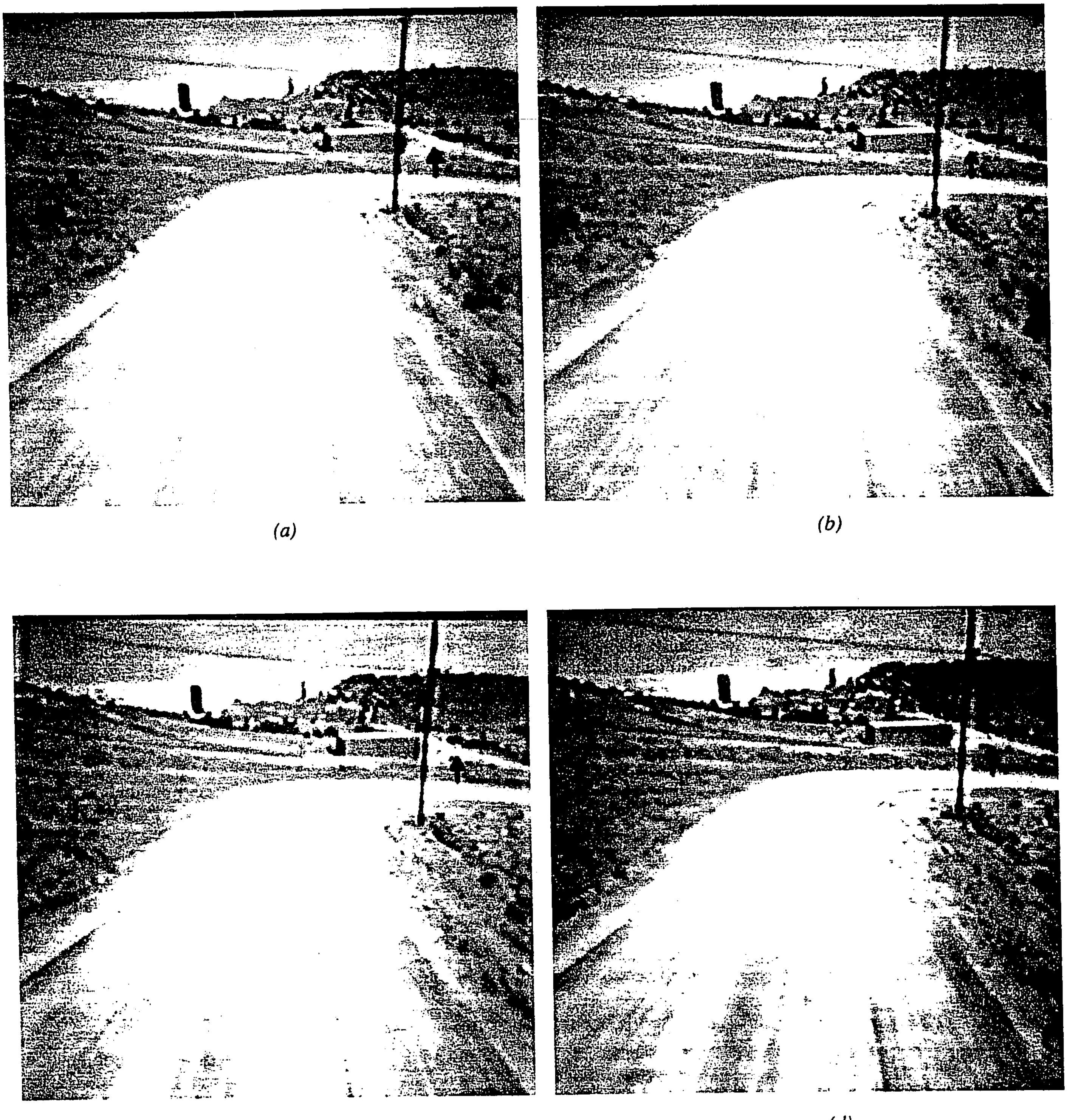
The estimated disparities for each pair of images are then transferred to the FOE location estimation stage. The results obtained for the first pair of images (frames 15 and 16) in Figure 5 are shown in Figure 7, where Figure 7(a) depicts the Interest Point locations in the coordinate frame of the second image of the pair along with the estimated disparity of the point (shown as a line attached to the apex of the pointer to the Interest Point location) and Figure 7(b) is the result of Qualitative Scene Model Calculation. The results obtained for the second and third and third and fourth frames of this example are presented in Figures 8 and 9, respectively.

6. MAP-BASED TARGET TRACKING

Target motion detection and tracking is essential for the potential military applications of a robotic vehicle. However, purely image-based target motion information may have restricted use in many practical military scenarios such as reconnaissance, where map location (latitude, longitude, and elevation) of the moving targets, precise information on their direction of movement, and knowledge about nearby roads and terrain may be crucial. We have developed and implemented the Map Assisted Tracking System (MATS), which integrates the digital terrain map information with Honeywell's Qualitative Reasoning system,[5,7] which was described in the last three sections. At present, MATS is loosely integrated with the Qualitative Reasoning system to provide a comprehensive set of information about the map location of the moving objects, the road label that the targets are possibly traveling on, and neighboring landmarks. Beyond practical mission considerations, digital terrain map information can be very helpful in detecting and tracking moving targets in high clutter and low contrast scenes.

Figure 10 provides a high level view of the MATS system. As shown at the top of the figure, Qualitative Reasoning provides the motion direction, x and y image location, and relative range of the moving targets. MATS performs an image-to-map correspondence for the individual targets and determines the approximate location of the targets in the map. An uncertainty area is computed for each target. Then, the digital roads file is searched to locate nearby roads. A search algorithm determines the most likely roads and nearby landmarks. This computation requires that the vehicle's position in the map must be given, which can be obtained from the Land Navigation System (LNS) or an Inertial Navigation System (INS), for future robotic vehicles. MATS assumes a given camera view angle, which can be obtained from the gimbal controller. Road and terrain information is used to predict obscuration so that the Qualitative Reasoning system is able to track targets in high clutter scenarios.

The prototype of the MATS system has been implemented and initial tests have been performed in conjunction with the Qualitative Reasoning system. Currently, MATS is functional, although it needs to be fully integrated with the Qualitative Reasoning system in an end-to-end experiment. The image-to-map correspondence algorithm must be refined and further testing needs to be done on the entire system. The remainder of this section describes the characteristics of the digital map data that we have used and explains the details of its use in the MATS system and experimental results for two representative scenarios.



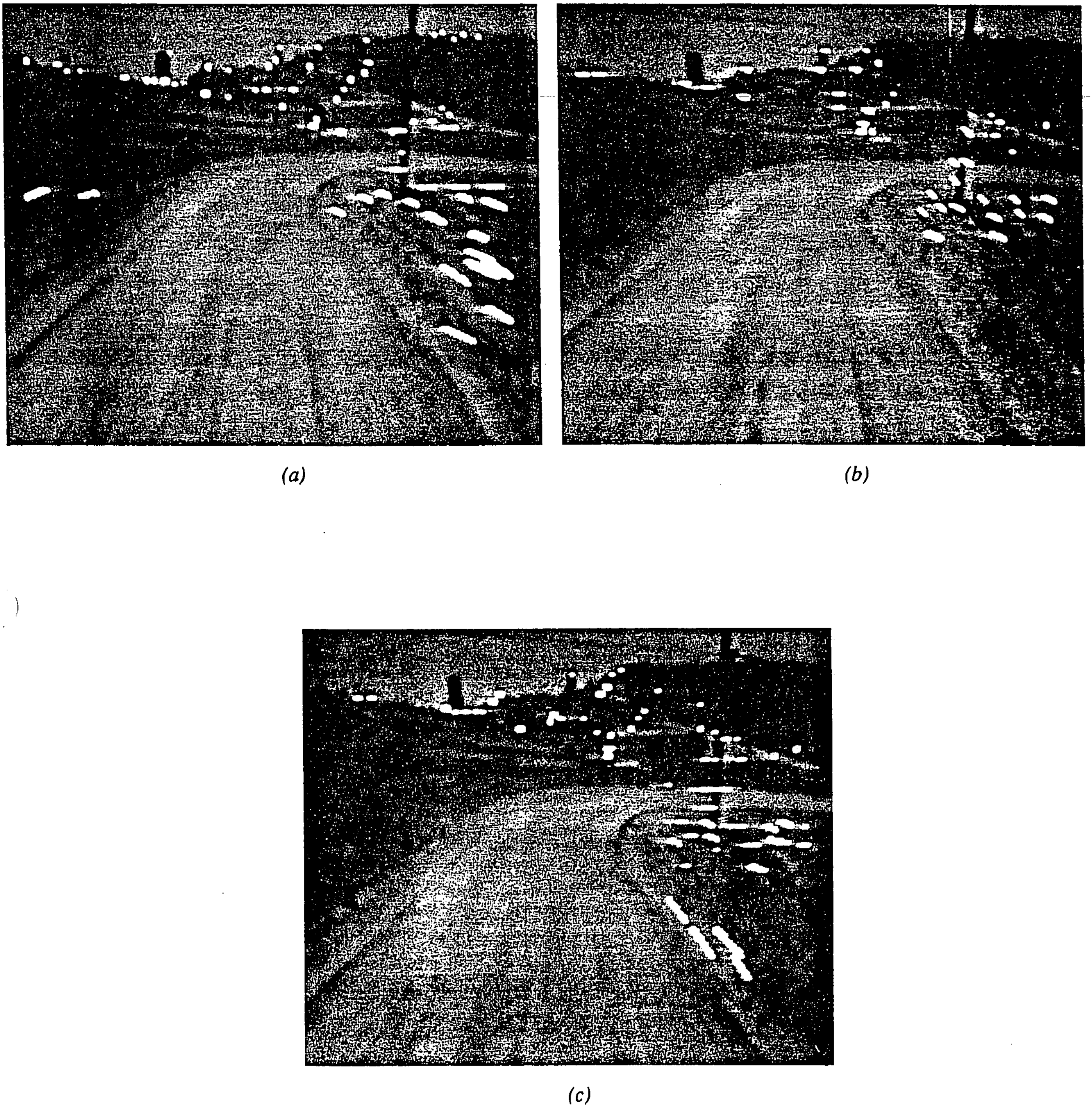
(a)

(b)

(c)

(d)

Figure 5: Four frames (15, 16, 17, 18) from the Collage I database, where a vehicle is seen traveling down in the very distant roadway in the top right corner of the image. *(a)* Frame 15. *(b)* Frame 16. *(c)* Frame 17. *(d)* Frame 18.

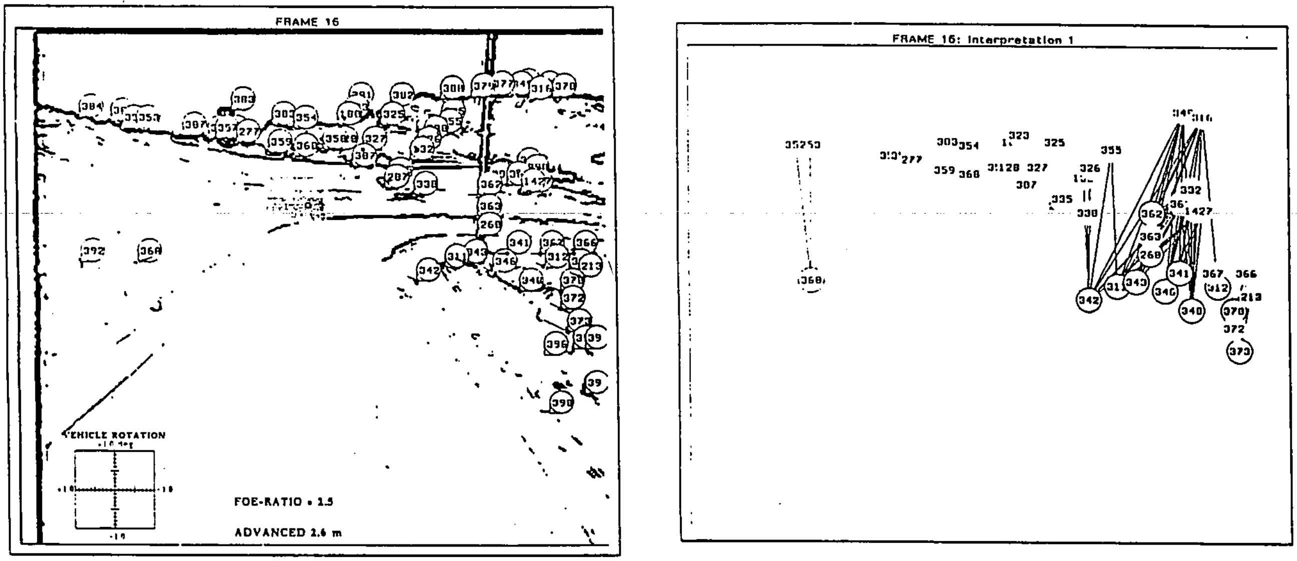


(a)

(b)

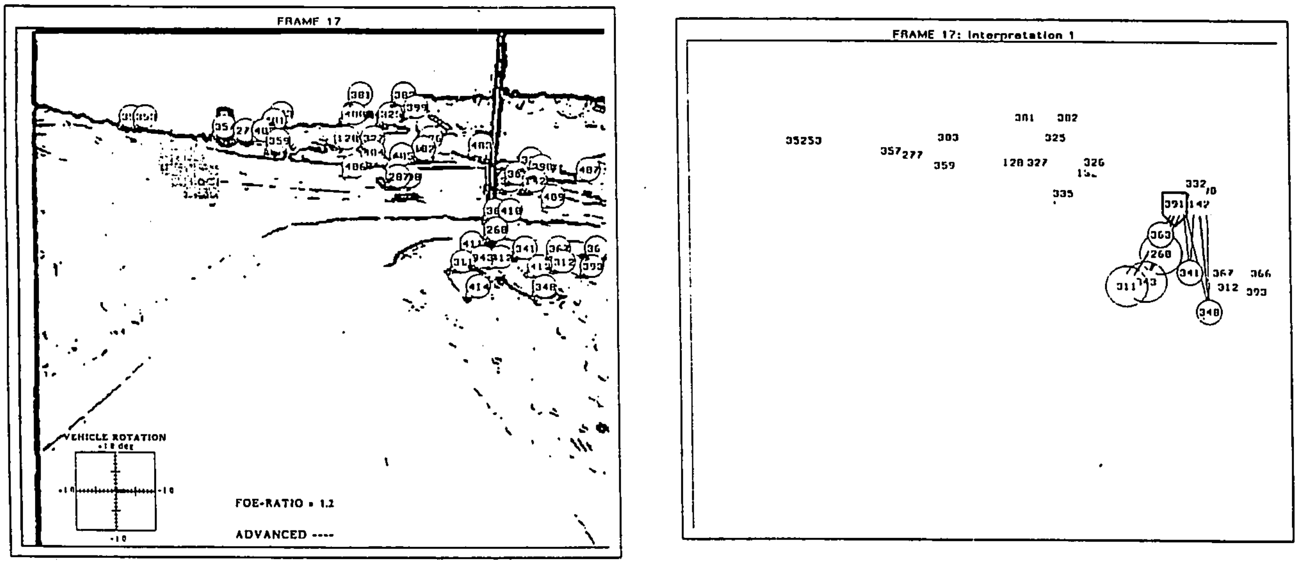
(c)

Figure 6: Estimated disparities between the four frames in Figure 5. *(a)* The displacement vectors for Interest Points in frames 15 and 16. Note that a displacement vector has been derived for the car traveling downward even though this object is several hundred feet from the ALV and its image has low contrast. *(b)* The displacement vectors for Interest Points in frames 16 and 17. The car was also detected in this image pair. *(c)* The displacement vectors for Interest Points in frames 17 and 18.



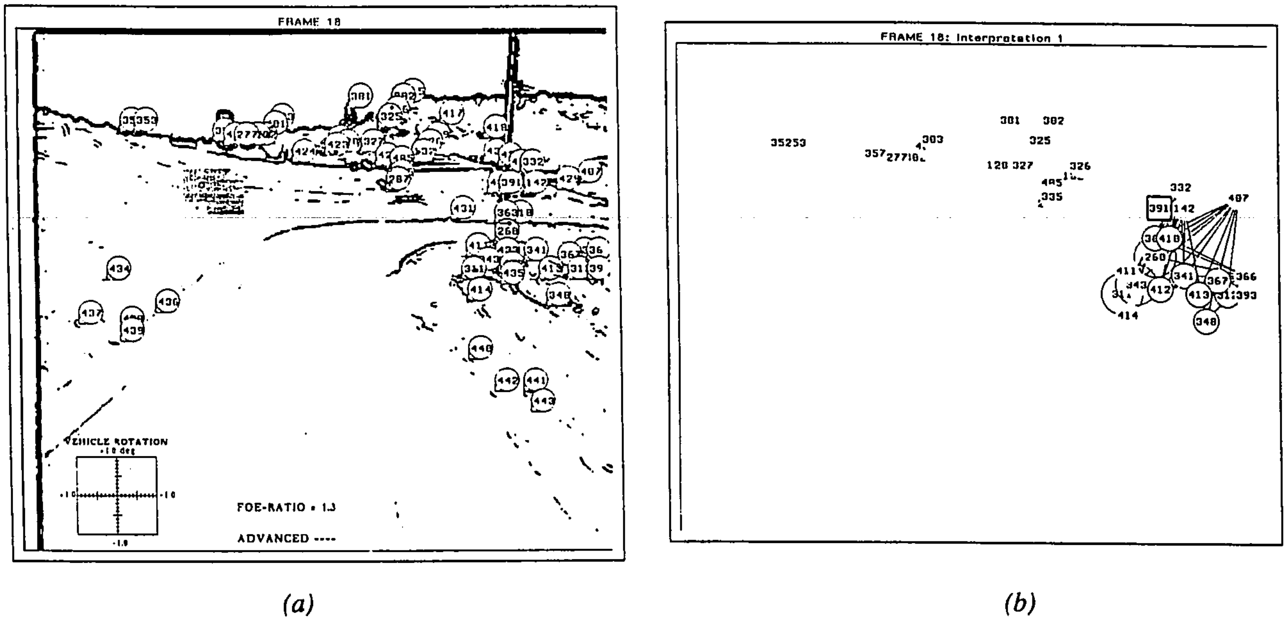
(a) (b)

Figure 7: Fuzzy FOE Computation and Qualitative Scene Model. *(a)* The Fuzzy FOE is shown by the shaded area. The circle within the shaded area is the most probable location for the FOE. *(b)* Qualitative Scene Model, where the size of one circle denotes its distance from the vehicle and links between circle or points indicate that closer relationships have been established between stationary entities.



(a) (b)

Figure 8: Fuzzy FOE Computation and Qualitative Scene Model. *(a)* The Fuzzy FOE is shown by the shaded area. The circle within the shaded area is the most probable location for the FOE. *(b)* Qualitative Scene Model, where the size of one circle denotes its distance from the vehicle and links between circle or points indicate that closer relationships have been established between stationary entities. Note that object 391 has been detected as moving. The direction of the arrow indicates that the target is moving downward.



(a) *(b)*

Figure 9: Fuzzy FOE Computation and Qualitative Scene Model. *(a)* The Fuzzy FOE is shown by the shaded area. The circle within the shaded area is the most probable location for the FOE. *(b)* Qualitative Scene Model, where the size of one circle denotes its distance from the vehicle and links between circle or points indicate that closer relationships have been established between stationary entities. Note that the target has been detected as moving, but its direction of movement could not be determined.

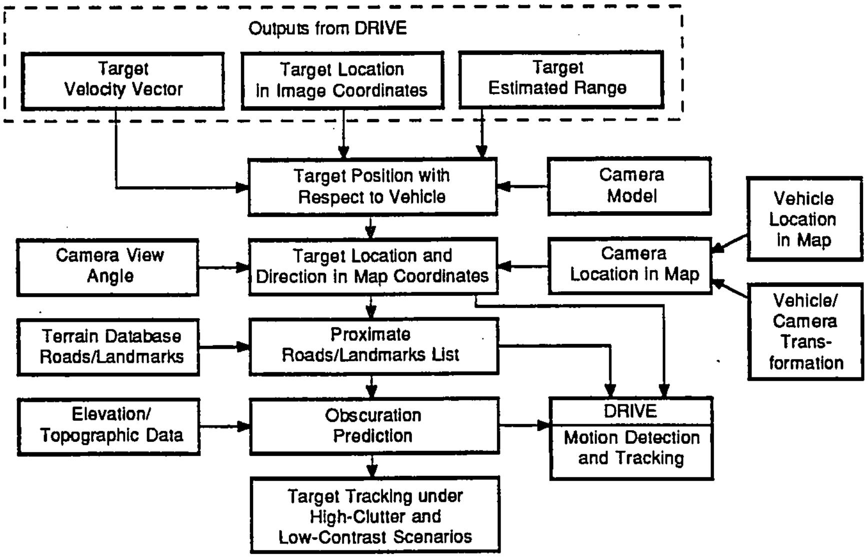

Outputs from DRIVE
Target Velocity Vector
Target Location In Image Coordinates
Target Estimated Range
Target Position with Respect to Vehicle
Camera Model
Vehicle Location In Map
Camera View Angle
Target Location and Direction In Map Coordinates
Camera Location In Map
Vehicle/ Camera Trans- formation
Terrain Database Roads/Landmarks
Proximate Roads/Landmarks List
Elevation/ Topographic Data
Obscuration Prediction
DRIVE
Motion Detection and Tracking
Target Tracking under High-Clutter and Low-Contrast Scenarios

Figure 10: Map Assisted Tracking System (MATS) which has been integrated with the DRIVE system for motion detection and tracking.

6.1 DIGITAL TERRAIN MAP DATABASE

The existing digital terrain map database of the Autonomous Land Vehicle (ALV) test site contains elevation, photographic, and terrain feature data. The elevation data specifies the elevation in meters above sea level at a given map position. The horizontal resolution of the elevation data is five meters per pixel. The photographic data includes a digitized aerial photograph of the ALV test site. The terrain data includes road, river, land cover, and soil information for the area. Only the road feature data from the terrain database was used in our MATS experiments. The road data consists of unique identifiers for each road segment. The roads are represented by their width and segment endpoint coordinates.

Each elevation data point in the original database was represented as 16-bit data. Since the actual variation in elevation over the mapped region was only 901 feet, we derived an 8 bit per pixel representation of the data that had units of meters. After the elevation data was derived, we transformed the feature data into the correct scale.

The terrain feature data files are provided as character files. Each line segment lists a brief header which includes a brief description, a unique identifier, and the number of line segments that belong to that segment, followed by the segment end points. The first step was to scale the endpoints to the elevation data and to run a line tracing algorithm to convert the data into vector form.

Using the digitized photograph at 4096 x 4096 resolution, a USGS 7.5 minute topographical terrain map, and the elevation data, prominent landmarks were identified and the correspondence was manually established. The road features were primarily used to establish the correspondence. Registration was done manually by selecting four corresponding points in each file. The four points were selected at the outer boundaries. Although the digitized aerial photographs were not orthophotos, they were treated as such for the purpose of these initial experiments. Future experiments would be better served by the usage of sensor and platform data to eliminate the effects of relief distortion and the resulting misalignment between the map and the image. Since the photograph data had a much finer resolution than the elevation data, the elevation data was bi-linearly interpolated to match the resolution of the photographic data. This resulted in a 0.7 meter spacing between posts in the elevation data. The same interpolation was performed on the road data. However, there was a trade-off to be made here between processing time and achievable resolution. It was decided that very fine resolution was not critical to the experiment and the data was sampled at a 1.4 meter resolution between posts.

Figures 11-13 show the digital terrain database which includes grid elevation data, elevation intensity data, and terrain features overlays.

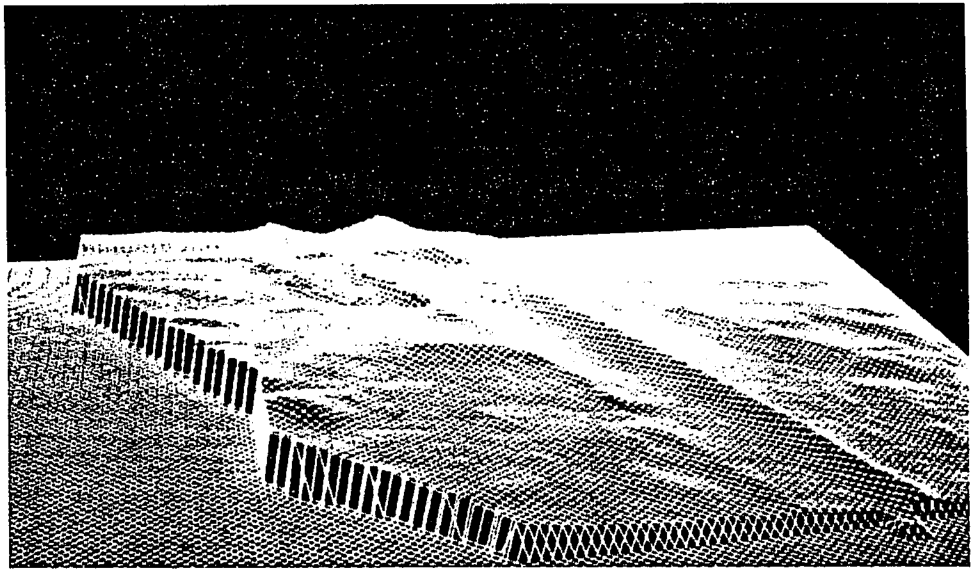


Figure 11: Digital elevation map of the ALV test site area displayed in a grid format.

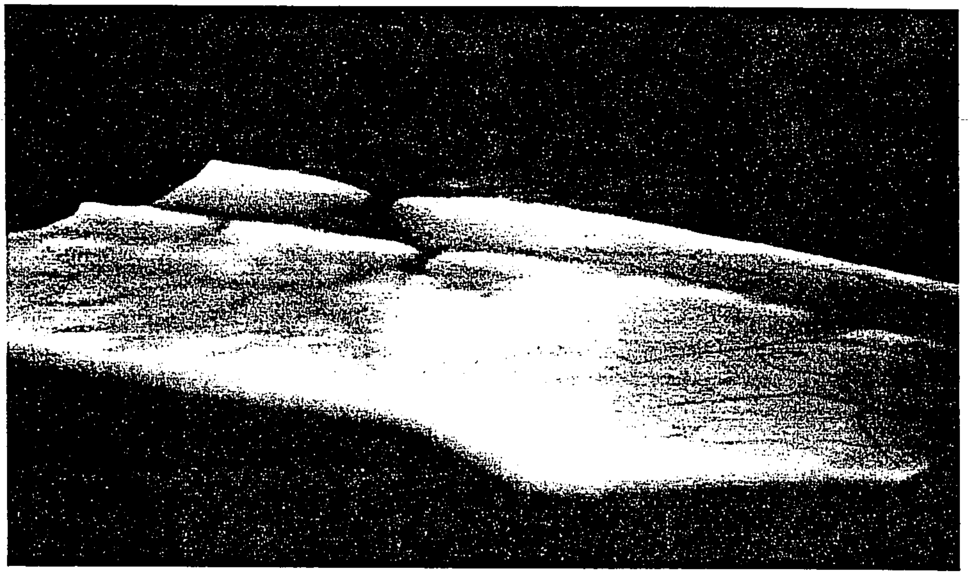


Figure 12: Digital Elevation Model (DEM) with overlays showing roads (black) and rivers.

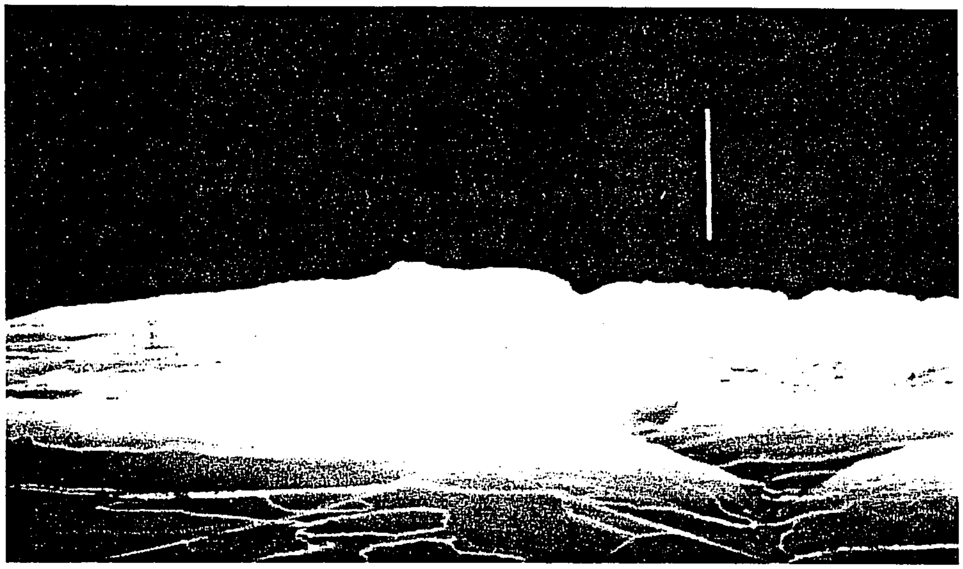


Figure 13: Close-up of digital elevation map with all feature data overlayed.

6.2 MAP ASSISTED TRACKING SYSTEM DESCRIPTION

The MATS functional objectives are: 1) locate the 3-D map coordinates of moving targets, 2) identify the roads or terrain that they are traveling on, and 3) identify nearest landmarks to the targets. MATS assumes that the following information is given:

- Target (x,y) image location and traveling direction in image coordinates
- Range to target
- Camera model and a fixed view angle
- Digital terrain map database

The first task that MATS performs is to find a view vector from the vehicle's camera center to the target. As shown in Figure 14, the target is detected at T(x,y) location and the center of the image C(256,242) is where the view vector coincides with the optical axis of the camera. The equation of an arbitrary view vector can be determined given the imaging system's 3-D coordinates (this is obtained from the Inertial Navigation System data available on a full-scale robotic vehicle) and orientation.

For the current implementation of MATS, the estimates for the ALV's Universal Transverse Mercatur (UTM) coordinates are obtained as follows: The image coordinates of three landmarks, observed in the image, are noted and the view vectors to the landmarks in a vehicle-centered coordinate frame are calculated. Because the absolute location of each of the landmarks in terms of UTM coordinates is known, the distance between each pair of landmarks is known and the subtended angle between any pair of neighboring legs of the triangle defined by the landmarks is known. Because the geometry of this triangle is known, it is possible to estimate the distance from the imaging system to each of the landmarks. The coordinate of the ALV in the UTM coordinate space is obtained by solving for the unique 3-D location that is the prerequisite distance from each of the three landmarks. This also establishes the orientation of the ALV in UTM coordinates. Improved accuracy is obtainable by using 4-10 landmarks for calculation of the ALV's coordinates (employing least squares techniques).

When the orientation and location of the ALV is determined in terms of the UTM coordinate system, the location of the target is estimated with the following approach. Let the orientation of a view vector in terms of the camera coordinate system be expressed by the pure horizontal rotation R_θ and pure vertical rotation $\mathbf{R}_\phi$ as shown in Figure 15. The θ and ϕ required to bring the target location to align with the optical axis are obtainable by estimating where the line of sight ray (defined by the view vector) intersects the digital terrain elevation map.

Let (Δx, Δy, Δz) be the displacement vector from the target to the vehicle, then the target's real location in the map coordinates, (X_w, Y_w, Z_w), is:

$$X_w = X_V + \Delta x \ , \quad Y_w = Y_V + \Delta y \ , \quad Z_w = Z_V + \Delta z \qquad (15)$$

where (X_V, Y_V, Z_V) is the coordinate of the current vehicle location in the map.

There are obviously uncertainties about the current vehicle map location and the location of the detected moving targets. In addition, there are uncertainties about the estimated range to the target obtained from Qualitative Reasoning. The terrain map information is used to correct such uncertainties. The initial hypothesis that targets are moving on roads allows MATS to search for the roads nearest to the computed target map location. The road file is represented as

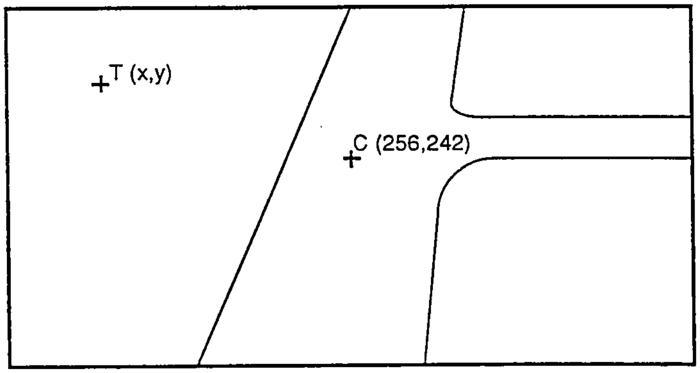


Figure 14: Orientation for locating target view vectors.

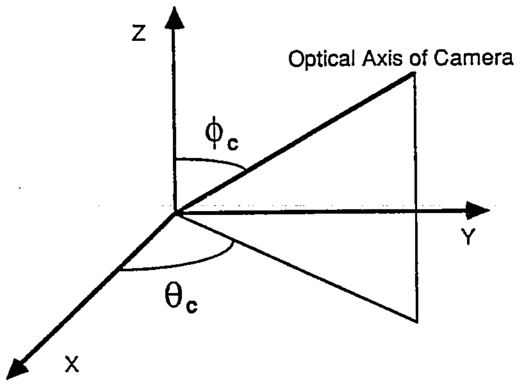


Figure 15: Orientation of the camera system.

a 2-D image file, where pixel values represent roads labels. Pixel values of zero indicate no roads, otherwise pixels values correspond to roads labels. From this road map representation, another file is generated which contains the shortest distances from each pixel to a road, and this file is pre-compiled to allow very rapid search. Once the target location is estimated in the map, MATS quickly searches for the x and y coordinates values in the road map representation and infers the road that the target is traveling on.

6.3 EXPERIMENTAL RESULTS

Experiments with MATS have been conducted on two scenarios (Figures 5 and 16). In both experiments, the targets detected were moving at a relatively far distance from the vehicle. Typically, the ranges were several hundred feet. Both scenarios contained high clutter and the contrast of the targets was low. In both experiments, only one target was moving in the image. Figures 17 and 18 show results calculated by MATS for the first example scenario and Figures 19 and 20 display the results for the second experiment.

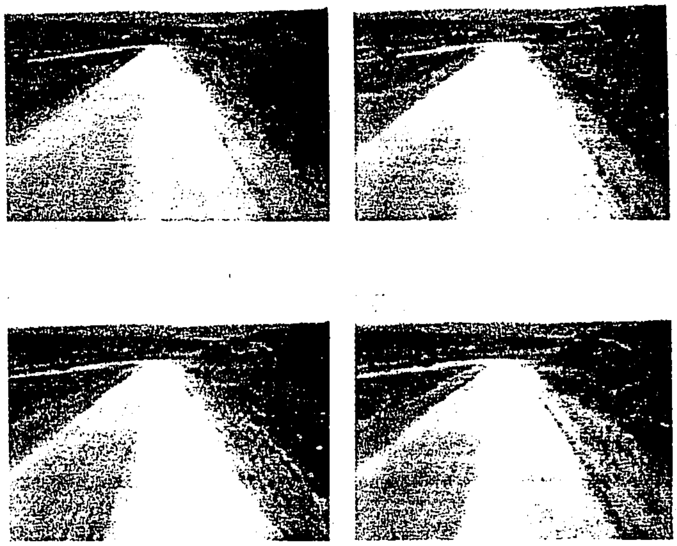


Figure 16. The second experiment shows another target moving across the scene with a range of several hundred feet.

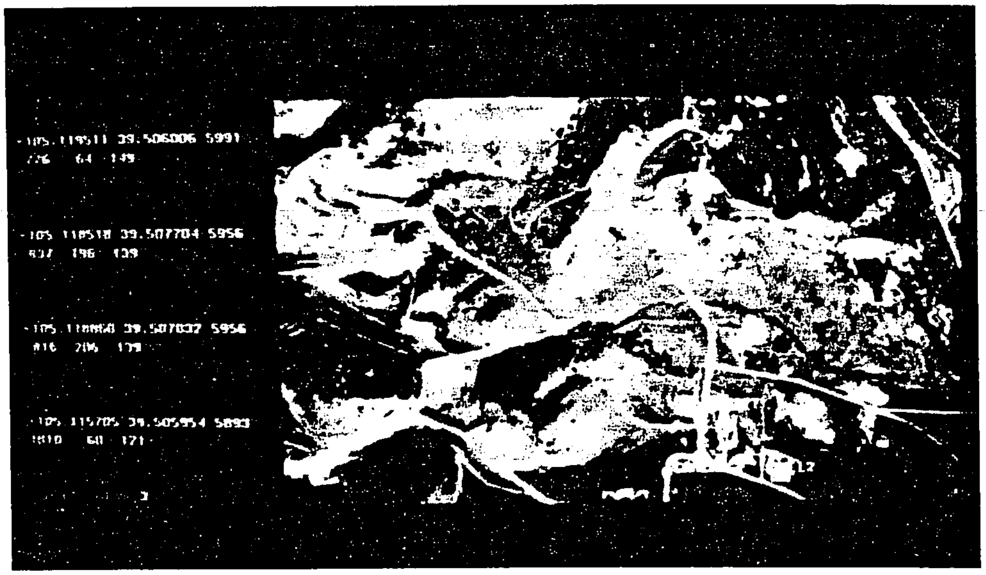


Figure 17: In the first experiment, MATS identifies the target location in the map and highlights the vehicle's position.

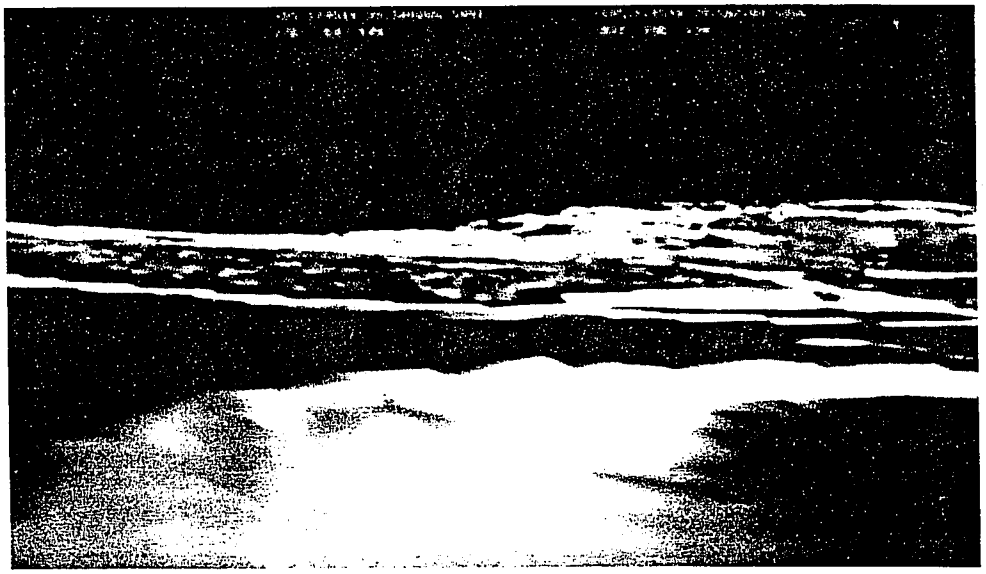


Figure 18: MATS generates a side view of the first scene showing what the vehicle should expect to observe and highlights the target in red, which corresponds very well to the location of the target in the actual images.

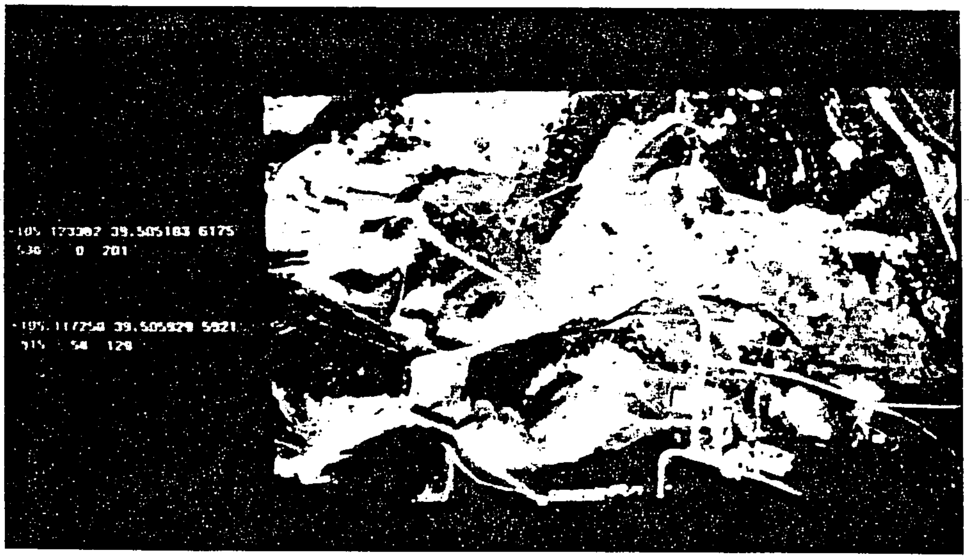


Figure 19. Target location detected in the second experiment with a direction vector overlaid on the target indicating its traveling course.

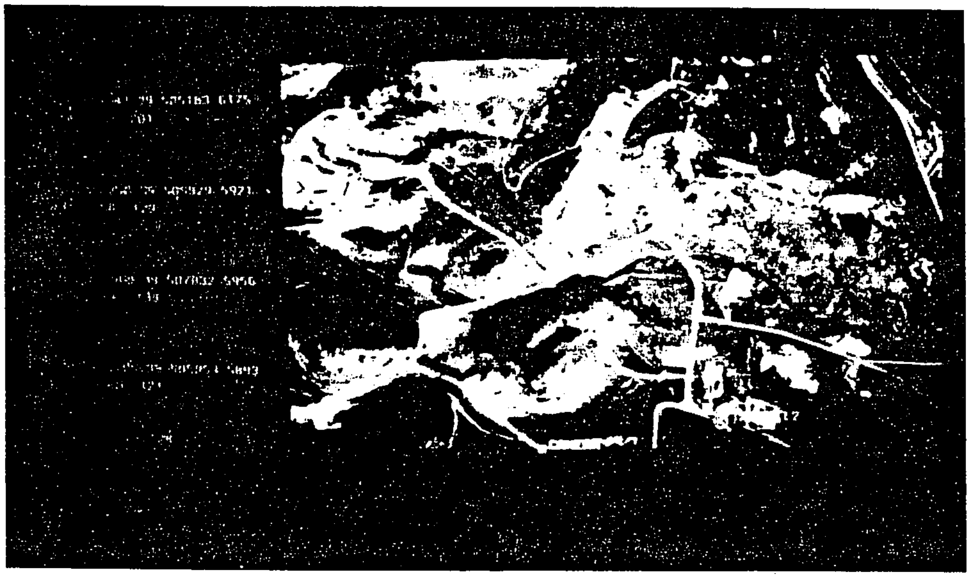


Figure 20. Final results of the second experiment which highlight the road segment the target is traveling on.

The objective of the MATS effort is to demonstrate the practical aspects of target tracking in a real-world scenario, where scenes are usually highly cluttered, low in contrast, and contain targets at long ranges. MATS has shown that it can enhance target detection and tracking performance and provide useful outputs such as a target's 3-D map location and the road it is traveling on. Currently, MATS is still a prototype system which needs further development. More experiments are planned to establish the performance capabilities of this system.

7. REGION MOTION DETECTION USING COLOR DIFFERENCE PICTURES

When the 3-D world is projected onto a 2-D image, valuable information is lost. Motion information, along with stereopsis and range maps, is a well known information source for the reconstruction of 3-D representations from 2-D images. Since motion information is useful for other image processing stages, it is desirable to perform motion analysis at an *early* stage of scene understanding. This early stage is referred to as the *peripheral process* by Jain[14] and the *short-range process* by Ullman.[23] The long range or attentive processes are correspondence schemes in which high-level, symbolic features are matched and tracked over time. Low-level processing stages for motion interpretation include gradient-based methods,[13] cross-correlation methods,[1] and spatio-temporal filtering methods.[24]

For real-time motion analysis, the algorithms employed are constrained to be efficient as well as dependable. Jain[14] has experimented with a simple method of using a difference picture accompanied with a simple decision tree to extract motion information in the peripheral phase. A difference picture is generated by comparing two frames of the same dynamic scene on a point by point basis. In subsequent experiments, he showed that the difference picture, in combination with the edge and corner image, could be used effectively to detect motion in the scene.[14,16] Features such as temporal-edges and interest points are often used in motion detection algorithms. Examples are the edge features used by Hildreth[12] to compute optical flow and the region images used by Bhanu and Burger[4] to compute disparity vectors. However, one deficiency of the Jain approach is that the interiors of constant intensity level regions do not generate a difference signal, even if the corresponding surface is moving. Thus, the determination of surface boundaries requires the observation of longer sequences or the application of more sophisticated, high-level analysis.

We present a similar scheme using color images to obtain a more reliable difference picture for use with standard region-based motion detection schemes. We have successfully demonstrated the detection of motion for the complex ALV imagery, where Jain's algorithm normally is not robust enough due to the diverse nature of this imagery.

7.1 COLOR DIFFERENCE PICTURE

Motion analysis techniques use various assumptions about the scene characteristics to decrease the complexity of the calculations of 3-D features. One such assumption is that the illumination of an object does not change from scene to scene. For the ALV scenario, this assumption may not hold because the changing location of the imaging system causes the orientation and location of objects, relative to the ALV, to continually change. Thus, the use of invariant scene characteristics is necessary. It has been reported that changes in the ambient illumination level does not alter the human perception of color. By using the individual color components of the image, instead of the luminance component, gradient-based motion algorithms will be less sensitive to local changes in average object intensities. For example, the hue of the image is calculated as a function of a ratio of linear combinations of the three primary image intensities, red, green, and blue.

$$Hue = \cos^{-1}\left\{ \frac{½[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}} \right\} \tag{16}$$

If $B > G$, then $Hue = 2\pi - Hue$

Therefore, a change in the average intensity of a specific region does not effect the magnitude of the hue.

The temporal derivative $\frac{\partial f(x,y,t)}{\partial t}$ in the discrete domain is approximated by the difference operation $f(x,y,t_2) - f(x,y,t_1)$, where $f(x,y,t)$ is the image intensity at the location $\{x,y\}$, at time t. The entries in the difference picture are significant only at pixel locations where an object has moved. The difference picture may be used in combination with the edge image to obtain a time-varying edge detector.[16] When more than two frames are used to approximate the temporal derivative, observation of a time-varying edge permits the detection of moving edges.[16] This distinction is made since the motion of an object detected with a Moving Edge Detector results from *persistent change* in the sequence of the frames representing the scene. Jain argues that if the change exists only in two frames, then the change is, most probably, not due to motion. The use of multiple frames helps to resolve the ambiguity problems due to noise that occur for frame-to-frame differencing techniques.

In the Moving Edge Detection method, the difference picture is refined using a syntactic labeling scheme. Because the criteria for this scheme are derived for noiseless imagery, they work accurately when there are only minor changes in the average intensity level of regions and the edges of regions are sharp. We have developed a similar, but simpler, method of obtaining a difference picture which uses color images in order to make the algorithm less sensitive to these inconsistencies.

Our algorithm is comprised of three steps:

(1) The point by point subtraction is done for each primary image, resulting in three conventional difference pictures.

(2) A symbolic mapping is applied to the red, green, and blue difference values to obtain a single symbolic difference image. (See Table 1). Only the sign of the difference picture for each primary image is used to derive the symbolic difference image, because *how* the region changed is more important than *how much* it changed.

(3) The connected components of the symbolic difference image are extracted. Isolated points and small regions are eliminated. Small regions in the difference picture are either caused by noise or by objects that are too far from the ALV to be of interest.

The computational complexity of this algorithm is low. Since the change in the region intensity, rather than the magnitude of the change, for each of the three primary images is used for moving object detection, the scheme is robust for outdoor imagery, when the camera rotation component is not significant. This procedure effectively removes most of the noise that often exists in difference pictures. Only those regions which are caused by targets moving at a *significant rate* of speed remain. The results obtained with this approach demonstrated far less noise than the difference pictures obtained from the Jain algorithm alone.

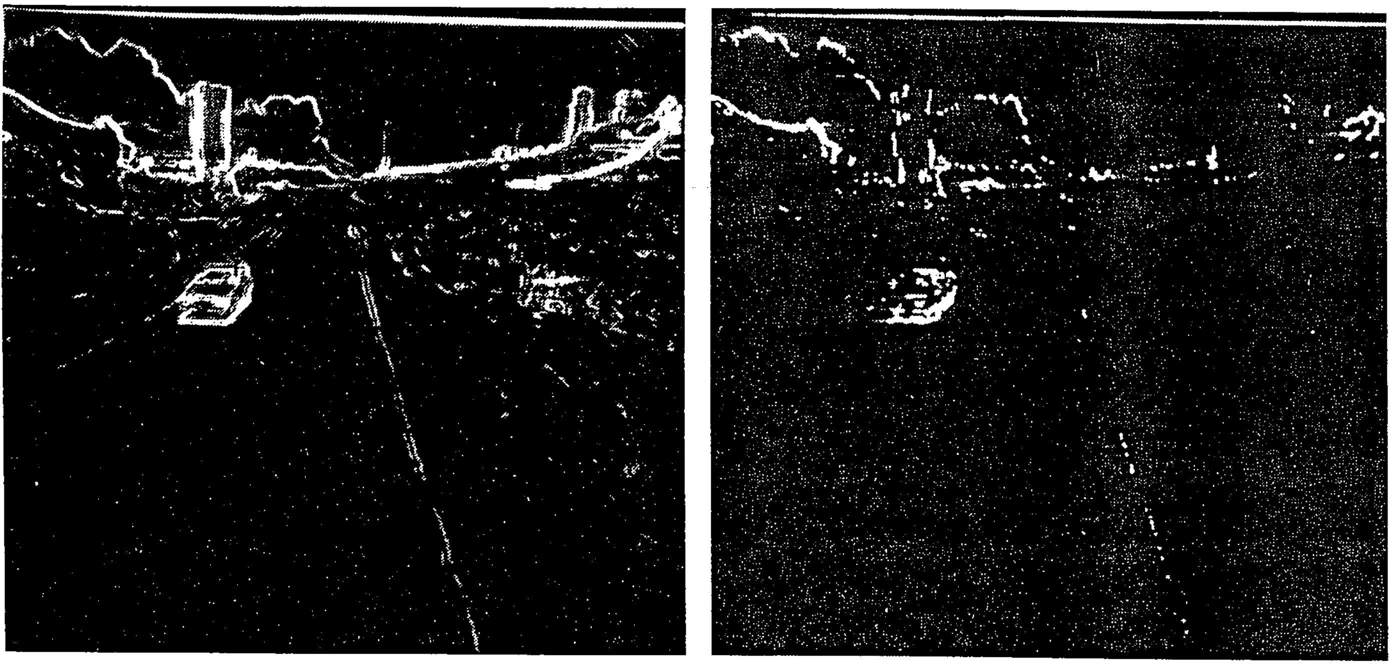
The performance of the Color Difference Picture motion detection algorithm was evaluated for a representative set of dynamic images. The purpose of the evaluation was to empirically appraise the effects of changing the algorithm's parameters on the resultant number of false detections, for imagery with varying amounts of sensor-induced motion. It was found that the approach performs well for sequences of images where the imaging system's motion is approximately linear (a large forward component, plus a significantly smaller rotational component), with only moderate sensitivity to the selected parameter values. For image sequences demonstrating more complex motion (significant rotational components), the algorithm was more dependent on the parameter values selected and it was not possible to identify a set of parameters for the algorithms which would be optimal for all cases.

7.2 MOVING COLOR EDGE DETECTION

Moving targets can be detected in a series of images on the basis of multiple clues. A few examples are moving edges, moving corners, and spatio-temporal frequency disparities. For Moving Color Edge Detection, the color difference picture is combined with the color edge magnitude image to identify moving color edge points in a sequence of images. This technique detects image locations where there is a high degree of *edginess* and a high temporal rate of change in intensity. Regions that display a large temporal rate of change are those regions of the color difference picture that remain after small regions are discarded. The edge image is obtained with the DiZenzo color image edge operator.[11] The magnitude of the DiZenzo color image edge operator for a scene where a vehicle is rapidly approaching the ALV is presented in Figure 21. The result of deriving the conjunctive evidence of the multi-image gradient magnitude and the Color Difference Picture is presented in Figure 21(b). Figure 22 provides another example of Moving Color Edge Detection.

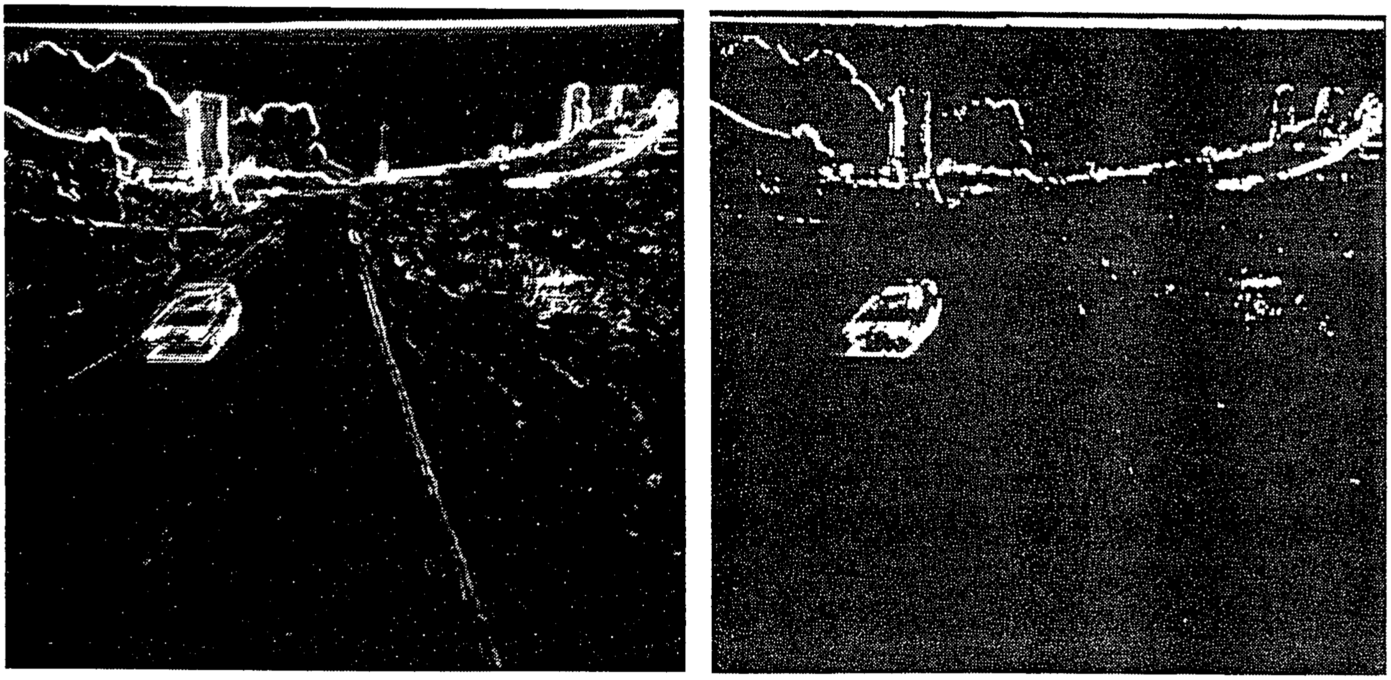
Class	Red	Green	Blue	Meaning
0	0	0	0	No change or changed negatively in all colors
1	0	0	1	changed positively in Blue direction
2	0	1	0	changed positively in Green direction
3	0	1	1	changed positively in Blue and Green directions
4	1	0	0	.
5	1	0	1	.
6	1	1	0	.
7	1	1	1	positive change in all three colors

Table 1: Symbolic mapping of three color difference images. 1 in column R, G, and B indicates a positive change.



(a) *(b)*

Figure 21: Color Difference Picture Motion Detection results. *(a)* Magnitude of the gradient image calculated with the DiZenzo Color Edge Detection Operator for frame 40. *(b)* Moving color edges detected in frame 40.



(a) *(b)*

Figure 22: Color Difference Picture Motion Detection results. (a) Magnitude of the gradient image calculated with the DiZenzo Color Edge Detection Operator for frame 44. *(b)* Moving color edges detected in frame 44.

7.3 MOVING REGION DETECTION

For ALV imagery, motion may be detected as a result of either true object motion or apparent (sensor-induced) motion because the imaging system is constantly translating and rotating due to the undulating road surface. The performance of the Moving Edge Detection operator and other traditional techniques will deteriorate for this variety of imagery unless a mechanism is used to compensate for the motion of the imaging system.

Conventional background motion compensation techniques match significant image features to obtain an estimate of the translation and rotation of the imaging system from the previous frame to the current frame. The transformation must be estimated based on a minimum mean-square error criterion between the observed location of the feature points and the predicted location, on the basis of the estimated transformation. The calculation of this transformation can be very expensive. Therefore, motion detection algorithms that are designed for a moving camera and that don't compensate for sensor-induced motion must be very robust.

We derived such an algorithm by modifying Jain's approach for greater resistance to the detection of changes caused by sensor motion and not by target motion. The Moving Boundary Detection algorithm detects those regions of the image whose location is changing rapidly with the following principle: Targets seen in an image will demonstrate a significantly higher rate of translation in proportion to their size than an arbitrary background region will, due to sensor motion.

We define a rate measure using this principle. First, the input color image is partitioned into a set of connected regions employing a region-based segmentation algorithm such as the Ohlander-Price-Reddy algorithm.[19] The regions where significant motion occurs are obtained by masking the regions detected with the Color Difference Picture (CDP). The criterion function for each region is calculated as the ratio of the number of boundary points which changed location to the total number of boundary points:

$$rate = \frac{\#\ pixels\ of\ boundary\ that\ changed\ location}{total\ boundary\ length} \tag{17}$$

This technique will exhibit degraded performance if any one of the following conditions apply: 1) the distance at which moving objects must be detected is extremely large, in which case the objects appear as part of the background; 2) the moving objects are small; or 3) the objects are moving at a sufficiently slow rate of speed, so that little or no change in the object's location is detectable at 30 Hz. Because none of the preceding degenerate cases occurs for the ALV scenario, the Moving Boundary Detection algorithm is a good algorithm for change detection.

This criterion function is useful for discriminating moving targets from stationary targets and/or the background for the following reasons: If image disparities for a pair of frames are the result of imaging system rotation only, then the apparent motion imparted to distant portions of the background will be greater than the apparent motion of stationary objects in the near-field. The spatial resolution of the image of an object is inversely proportional to its range from the imaging system. Thus, because the resolution of the background is low, its segmentation is poorer than the segmentation of nearer portions of the scene. Therefore, these regions of the image will be eliminated by the region size criteria in the formation of the CDP. The regions of the background that aren't discarded by the preceding step are normally segmented out by the rate measure, because they translate at a relatively slow rate. The only regions of the image that remain after these two segmentation phases are the moving objects. The range of interest for object motion detection will dictate the thresholds for this process. Results obtained with this approach for four frames of the Collage I database are presented in Figure 23.

A Moving Region Detection algorithm was also implemented. This algorithm is derived from the same concepts that applied for the derivation of the Moving Boundary Detection algorithm. Its "rate" measure is:

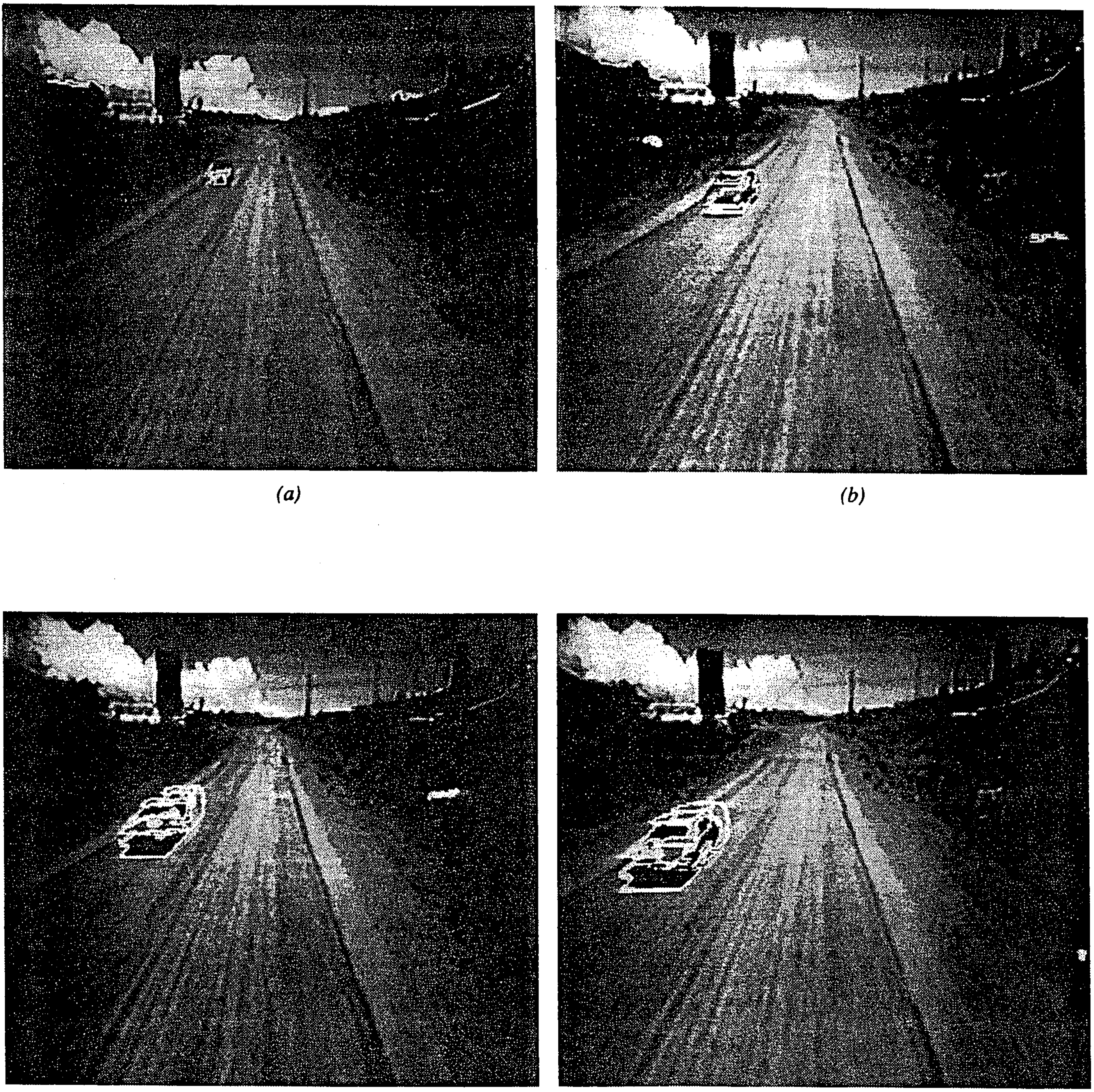
$$rate = \frac{\#\ pixels\ that\ moved}{total\ area\ of\ the\ region} \tag{18}$$

The performance of this algorithm was comparable to the performance of the Moving Boundary Detection algorithm. The regions which resulted after application of this algorithm to the images are shown in Figure 24.

8. CONCLUSIONS

We have presented our qualitative approach to scene understanding for mobile robots in dynamic environments. The challenge of understanding unstructured outdoor image sequences is that stationary objects do not appear as stationary in the image and mobile objects do not necessarily appear to be in motion. Consequently, the detection of 3-D motion often requires reasoning far beyond simple 2-D change analysis.

The approach taken here clearly departs from related work by following a strategy of qualitative, rather than quantitative, reasoning and modeling. All the numerical efforts are packed into the computation of the Focus of Expansion (FOE), which is accomplished entirely in 2-D. To cope with the problems of noise and errors in the



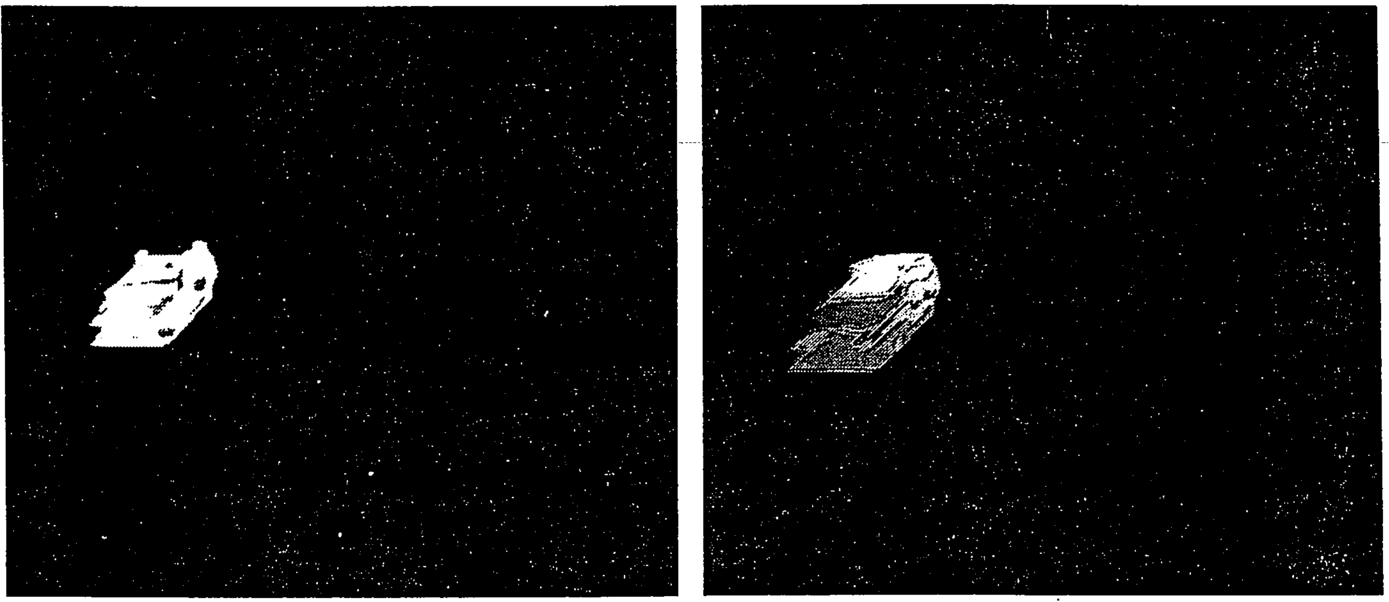
(a)

(b)

(c)

(d)

Figure 23: Moving Boundary Detection algorithm results. The boundary of the detected moving target is shown with a white line. *(a)* Frame 20. *(b)* Frame 40. *(c)* Frame 44. *(d)* Frame 48.



(a) *(b)*

Figure 24: Moving Region Detection algorithm results. The detected moving regions are displayed as uniform intensity regions, where the intensity of each region is arbitrary. Non-moving regions are displayed as black. *(a)* Frame 48. *(b)* Frame 50.

displacement field, we determine a region of possible FOE-locations, known as the *Fuzzy FOE*, instead of a single FOE.

We have shown that even without knowing the exact location of the FOE, conclusions about motion and 3-D scene structure can be drawn. From these clues, we construct and maintain an internal 3-D representation, termed the *Qualitative Scene Model*, in a generate-and-test cycle over extended image sequences. This model also serves as a platform for other visual processes, such as occlusion analysis, perceptual grouping, and object recognition. To overcome the ambiguities inherent to dynamic scene analysis, multiple interpretations of the scene are pursued simultaneously.

The examples given in this paper show the fundamental operation of our approach on real images produced by the Autonomous Land Vehicle (ALV). Since the exclusive use of displacement vectors from point features is a limiting factor, we showed our initial experiments on edge and region-based feature tracking. We plan to integrate wavefront approaches also for region motion detection.[4] Also, to exploit a larger part of the information contained in the image and to demonstrate the full potential of our approach, lines, regions, and map information need to be fully integrated within the Qualitative Scene Model.

ACKNOWLEDGEMENTS

We would like to thank George Lukes of the U.S. Army Engineer Topographic Laboratories for his input on the map-related experiments. We are also thankful to Dave Robohm and Henry Lai for their help in this work. This research is supported by DARPA/U.S. Army ETL Contract DACA 76-86-C-0017.

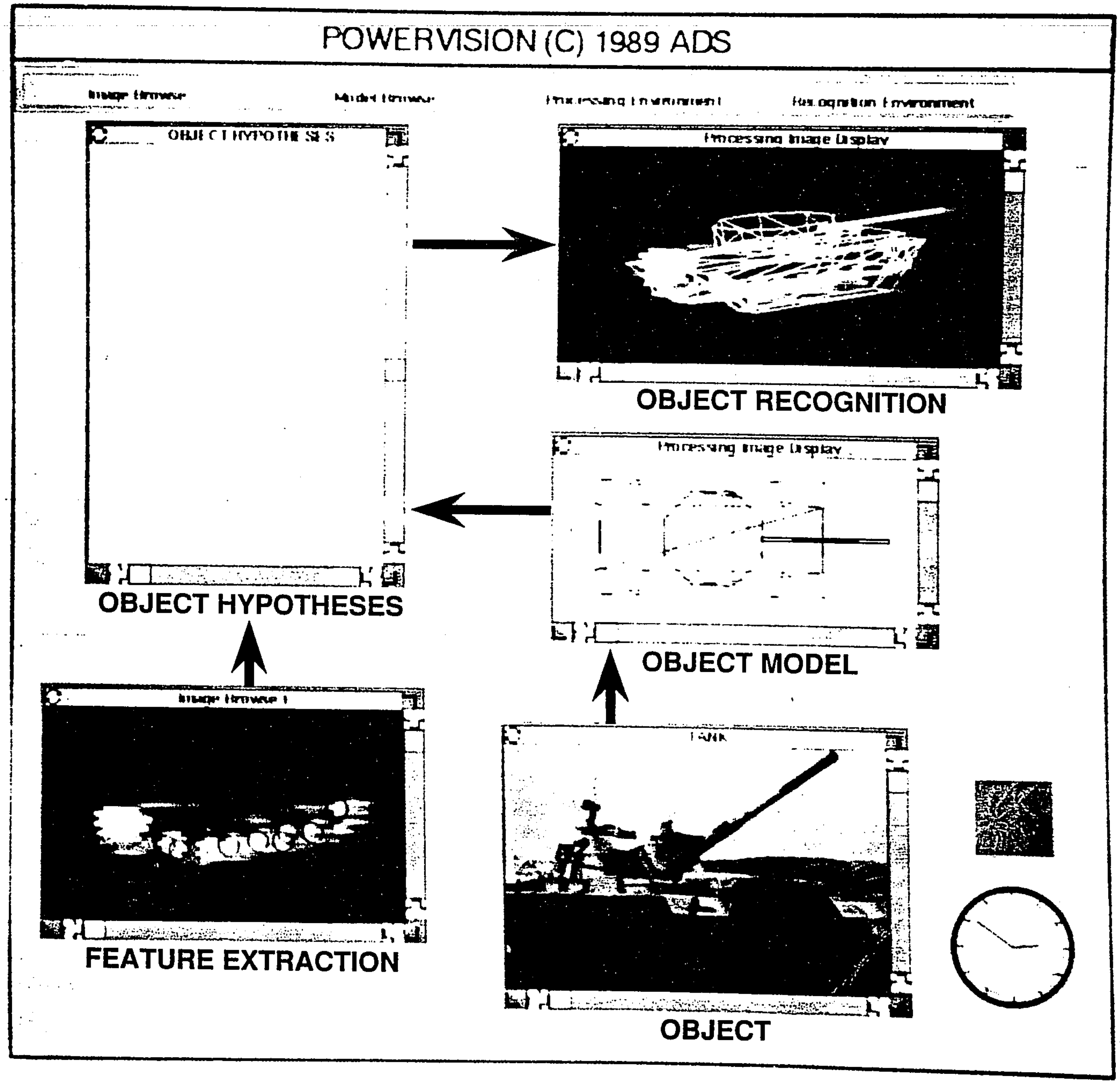
REFERENCES

1. S.T. Barnard and W.B. Thompson, "Disparity Analysis of Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **PAMI-2**(4) pp. 333-340 (July 1980).
2. B. Bhanu, "Automatic Target Recognition: State of the Art Survey," *IEEE Trans. on Aerospace & Electronic Systems* **AES-22**(4) pp. 364-379 (July 1986).
3. B. Bhanu and W. Burger, "Scene Dynamics Technical Report - DRIVE: Dynamic Reasoning from Integrated Visual Evidence," Defense Advanced Research Projects Agency, Contract No. DACA76-86-C-0017, Honeywell Systems and Research Center (June 1987).
4. B. Bhanu and W. Burger, "Approximation of Displacement Field Using Wavefront Region Growing," *Computer Vision, Graphics and Image Processing*, (March 1988).
5. B. Bhanu and W. Burger, "Qualitative Motion Detection and Tracking of Targets from a Mobile Platform," *Proc. DARPA Image Understanding Workshop*, pp. 289-318 (April, 1988).
6. B. Bhanu and D. Panda, "Qualitative Reasoning and Modeling for Robust Target Tracking and Recognition from a Mobile Platform," *Proc. DARPA Image Understanding Workshop*, pp. 96-102 Morgan Kaufmann, (1988).
7. W. Burger and B. Bhanu, "Qualitative Motion Understanding," *Proc. Tenth International Joint Conference on Artificial Intelligence, IJCAI-87, Milan, Italy*, Morgan Kaufmann Publishers, (August 1987).
8. W. Burger and B. Bhanu, "Qualitative Understanding of Scene Dynamics for Autonomous Mobile Robots," *Submitted to International Journal of Robotics Research*, (1987).
9. W. Burger and B. Bhanu, "Dynamic Scene Understanding for Autonomous Mobile Robots," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (June 1988).
10. W. Burger and B. Bhanu, "On Computing a 'Fuzzy' Focus of Expansion for Autonomous Navigation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (June 1989).
11. S. DiZenzo, "A Note on the Gradient of a Multi-Image," *Computer Vision, Graphics and Image Processing* **33**(1) pp. 116-125 (January, 1986).
12. E.C. Hildreth, *The Measurement of Visual Motion*, MIT Press, Cambridge, Mass. (1984).
13. B.K.P. Horn and B.G. Schunck, "Determining Optical Flow," *Artificial Intelligence* **17** pp. 185-203 (1981).
14. R. Jain, "Extraction of Motion Information from Peripheral Processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-3**(5) pp. 489-503 (1981).
15. R. Jain, "Direct Computation of the Focus of Expansion," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-5**(1) pp. 58-64 (January 1983).
16. R. Jain and S. Haynes, "Time-varying Edge Detection," *Computer Graphics and Image Processing* **21** pp. 345-367 (1982).
17. J. Kim and B. Bhanu, "Motion Disparity Analysis Using Adaptive Windows," Technical Report 87SRC38, Honeywell Systems & Research Center (June 1987).
18. H.P. Moravec, "Towards Automatic Visual Obstacle Avoidance," Proc. 5th International Joint Conference on Artificial Intelligence, pp. 584 (August 1977).
19. R. Ohlander, K. Price, and D.R. Reddy, "Picture Segmentation Using a Recursive Region Splitting Method," *Computer Graphics and Image Processing* **8** pp. 313-333 (1978).
20. K. Prazdny, "Determining the Instantaneous Direction of Motion from Optical Flow Generated by a Curvilinear Moving Observer," *Computer Graphics and Image Processing* **17** pp. 238-248 (1981).
21. D. Regan, K. Beverly, and M. Cynader, "The Visual Perception of Motion in Depth," *Scientific American*, pp. 136-151 (July 1979).
22. J.H. Rieger, "Information in Optical Flows Induced by Curved Paths of Observation," *J. Opt. Soc. Am.* **73**(3) pp. 339-344 (March 1983).
23. S. Ullman, *The Interpretation of Visual Motion*, MIT Press, Cambridge, Mass. (1979).
24. A. B. Watson and A. J. Ahumada, Jr., "Model of Human Visual-Motion Sensing," *Journal of the Optical Society of America A*, 2, pp. 322-342 (1984).

[illegible]

PROCEEDINGS:

Image Understanding Workshop



DARPA

Sponsored by:
Defense Advanced Research Projects Agency
Information Science and Technology Office

SC

May 1989

Image Understanding Workshop

Proceedings of a Workshop
Held at
Palo Alto, California

May 23-26, 1989

Sponsored by:
Defense Advanced Research Projects Agency
Information Science and Technology Office

This document contains copies of reports prepared for the DARPA Image Understanding Workshop. Included are Principal Investigator reports and technical results from both the basic and strategic computing programs within DARPA/ISTO sponsored projects and certain technical reports from selected scientists from other organizations.

APPROVED FOR PUBLIC RELEASE
DISTRIBUTION UNLIMITED

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the United States Government.