# Dynamic-Scene and Motion Analysis Using Passive Sensors

## Part II: Displacement-Field and Feature-Based Approaches

**Bir Bhanu, University of California at Riverside**
**Ramakant Nevatia, University of Southern California**
**Edward M. Riseman, University of Massachusetts**

*SEVERAL MOTION-UNDERSTANDING TECHNIQUES USING DISPLACEMENT FIELDS, 3D MOTION AND STRUCTURES, AND FEATURE CORRESPONDENCE SHOW PROMISE. BUT THERE ARE MANY ISSUES TO ADDRESS AND PROBLEMS TO SOLVE BEFORE WE ACHIEVE ROBUST DYNAMIC-SCENE AND MOTION ANALYSIS.*

DYNAMIC-SCENE AND MOTION analysis remains one of the more difficult and challenging areas of vision research. Many practical outdoor applications require algorithms that can withstand complex vehicle maneuvers and responses by other independent entities, detect and track objects moving in the face of occlusions, and operate in high-clutter and low-contrast situations when objects are far away and when environmental conditions and the terrain are changing. There are no practical motion systems that can robustly and accurately determine dense depth in real scenes.

However, researchers are actively exploring several promising directions. In a companion article (see p. 45), we present the basic elements of dynamic-scene and motion analysis and the qualitative motion-understanding approach, including computing the fuzzy focus of expansion and performing qualitative reasoning to interpet image changes. Here we describe other dynamic-scene and motion analysis techniques, developed at the University of Massachusetts and the University of Southern California, to support the DARPA Strategic Computing program's Autonomous Land Vehicle effort. We also discuss the issues and technical advances to be made in this important area.

## Displacement-field approaches

We used the term "optical flow" in the first article to mean the two-dimensional velocity of image pixels. When measured on discrete image frames, optical flow is often called a displacement field. We have used these terms interchangeably. Other terms are defined in a glossary on p. 61.

**Reliable computation of optical flow.** Researchers at the University of Massachusetts have developed a unified hierarchical computational framework that uses correlation matching to determine dense displacement fields from a pair of images (see Figure 1).[1] A key idea is the separation of computations according to scale: Large-scale (or low spatial-frequency) intensity variations can provide approximate measurements over a large range of magnitudes of motion, while small-scale (or high spatial-frequency) variations can provide more accurate measurements over a smaller range. This leads to the first three components of the framework: spatial-frequency decomposition (filtering) to separate intensity variations according to scale, a local parallel match-criterion within each scale, and a control strategy for combining measurements from different scales.

The fourth component is a directionally dependent confidence measure, which associates different confidences with each of the displacement vector's directional components. Since image displacement is a

0885/9000/92/0200-0053 $3.00 © 1992 IEEE

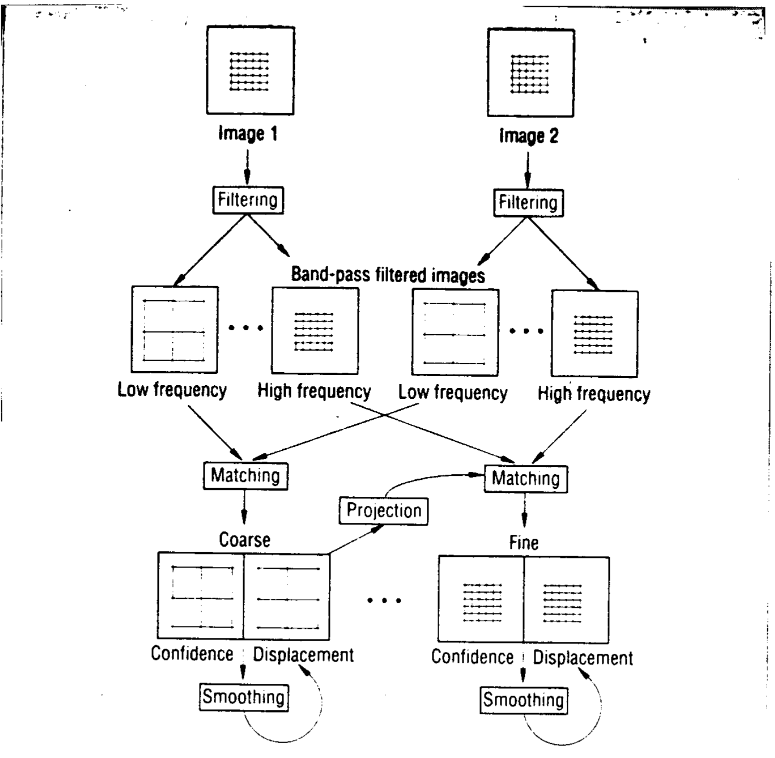

**Figure 1. The hierarchical computational framework.**

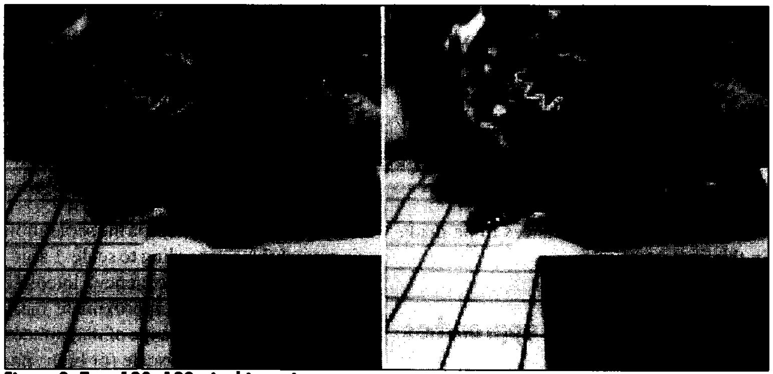

**Figure 2. Two 128x128-pixel input images.**

vector quantity, its reliability can vary according to direction. An image feature such as a high-contrast boundary might have a reliable match in one direction (for example, perpendicular to the boundary), but not in the other direction (parallel to the boundary). This suggests that a directionally dependent confidence measure would be useful. Also, while an area might be homogeneous at one scale, it could have information useful for reliable matching at a different scale. Therefore, confidence measures should be separately computed with each spatial-frequency channel. To obtain a dense displacement field given unreliable displacement vectors, we might have to propagate reliable displacements to their less reliable neighbors (for example, regional boundaries are more likely to have reliable matches than interior points if the region is relatively homogeneous). This leads to the last essential component of the framework, a smoothness constraint that specifies the criterion for propagating reliable displacements as a function of confidence.

This integrated system (see Figure 1) uses the minimization of the sum-of-squared-differences (SSD) of gray values as the local criterion for determining subwindow matches between frames. The system computes confidence matches based on the shape of the SSD surface, and formulates the smoothness assumption as the minimization of an error function.

The confidence measure is a two-dimensional vector. It is convenient to describe it in terms of two orthogonal basis vectors $e_{max}$ and $e_{min}$, which vary from pixel to pixel in an image.[2] The displacement vector $D$ can be decomposed in terms of its components along these basis vectors, and confidence measures $C_{max}$ and $C_{min}$ are associated with these components. We can easily understand basis vectors and confidence vectors by their behavior at the image's high-curvature points, edge points, and homogeneous areas. At a high-curvature point, both $C_{max}$ and $C_{min}$ are high, indicating that all the components of a displacement vector are reliable. In this case, the exact directions of $e_{max}$ and $e_{min}$ are not crucial; they depend on the precise shape of the contour. At an edge point, $C_{max}$ is high and $C_{min}$ low, and $e_{max}$ and $e_{min}$ are perpendicular and parallel to the edge, respectively. At a homogeneous area, both the confidences are low, and the directions of the basis vectors depend on the details of the image intensity variations at that point.

The error function consists of two terms. Approximation errors measure how well a given displacement field approximates the local match estimate, while smoothness errors measure the global spatial variation of a given displacement field. The system uses the finite-element method to solve the minimization problem. The functional-minimization problem formulated in the matching technique converges to the minimization problem used in gradient-based techniques.[3] In particular, by relating an approximation error function used in the matching approach to the intensity constraint used in the gradient-based approach, we can identify confidence measures explicitly that have been used only implicitly in the gradient-based approach.

Figure 2 shows two 128×128-pixel input

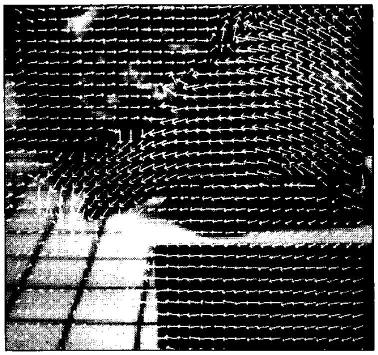

Figure 3. The smoothed displacement vector field superimposed on the first frame of Figure 2.

images of toy animals. The scene consists of a toy dinosaur, a toy chicken in the background, and a tea box in the foreground, all of which rest on a tabletop with a grid pattern on it. The 3D motion between the two frames consists of a translation of the camera to the right along its $x$ axis, and with a leftward rotation (1.5 degrees) about the vertical $y$ axis, as well as an independent movement of the toy dinosaur.

Figure 3 shows the smoothed displacement field superimposed on the first frame of Figure 2. Only a 32x32-pixel sample of the displacements are shown to improve their visibility. The algorithm works well, and the smoothness constraint has "filled in" several areas of the image; for example, the lower right portion of the dinosaur, part of the chicken, and the floor. To make the behaviors of the confidence measures more explicit at corners, edges, and homogeneous areas, image pixels are classified according to the values of $C_{max}$ and $C_{min}$, as we described earlier.

Figure 4 presents the results of this classification. Two interesting areas in the image are the boundary between the chicken and the dinosaur, and the boundary between the tea box and the floor. The first area has been maintained correctly during smoothing, primarily because the confidence values are rather large for the vectors on either side of this boundary, preventing those vectors from changing during smoothing. However, the area of the floor just left of the tea box has incorrect displacements, because the reliable displacements at the edges of the tea box have influenced their less reliable neighbors. Thus, although errors due to occlusion boundaries can be reduced, they still occur. The problem of detecting depth and motion discontinuities remains an important unsolved problem.

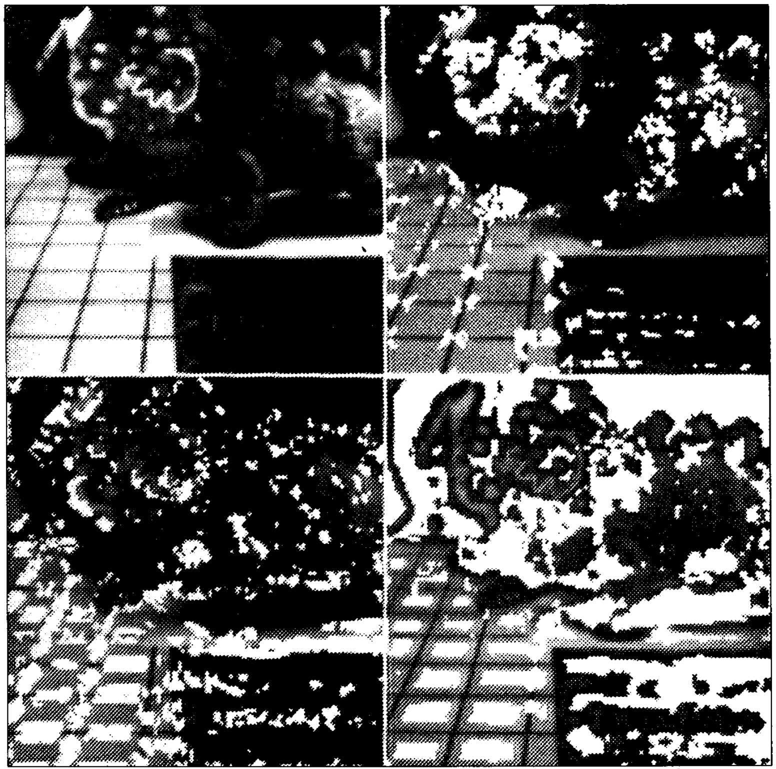

Figure 4. The input image (top-left quadrant) and classifications of the image's pixels as corners (top-right quadrant), edges (bottom-left quadrant), and homogeneous areas (bottom-right quadrant) as a function of the confidences $C_{max}$ and $C_{min}$ at each pixel.

**Optical flow for detecting motion and determining structure.** To interpret optical flow, a system must recover the three-dimensional motion parameters of the sensor and any visible moving objects, as well as the depth of visible points and surfaces. Researchers at the University of Massachusetts have developed a two-step algorithm for determining general sensor motion (five degrees of freedom) in an environment where other objects are movinging independently.[4] Input to this algorithm consists of a flow field and associated confidences. In the first step, the algorithm segments the flow field into connected sets of flow vectors, where each set is consistent with the rigid motion of an approximately planar patch. The segmentation is based on a modified version of the generalized Hough transform, with displacement vectors voting for motion parameters. Each segment is expected to correspond to the motion of a portion of only one rigid object. This approach makes it possible to deal with independently moving objects. This stage of segmenting the flow field is unnecessary when there are no independently moving objects, and the next step of the algorithm can treat the entire static environment as a single rigid object.

In the second step, the algorithm groups together the segments found in the first step under the hypothesis that they have been induced by a single, moving, rigid object (that is, the planar-surface assumption is dropped). The algorithm computes the optimal motion parameters and related error measure for each segment using a least-squares approach, which minimizes the deviation between the measured flow fields and those predicted from the estimated motion and surface structure. This step involves grouping flow-field segments that are consistent with the same motion parameters. After computing the 3D motion parameters, the algorithm can easily compute the depth if it knows the total translation between frames.

For the image pair shown in Figure 2, the algorithm extracted segments corresponding to the main surfaces in the environment (see Figure 5). However, their boundaries

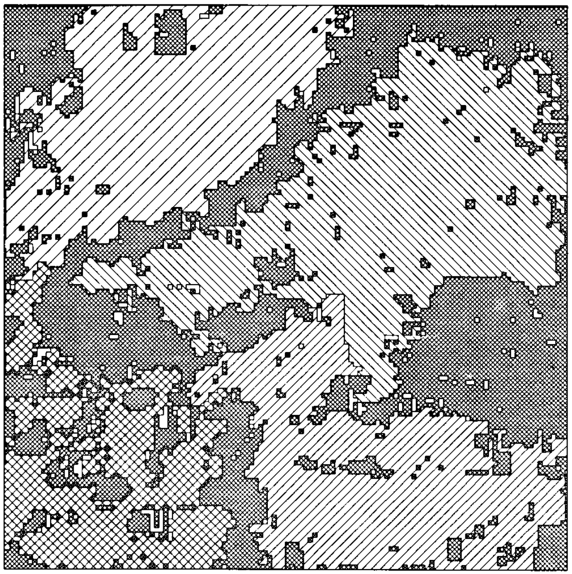

**Figure 5. A segmented flow field. The white areas correspond to flow vectors with zero confidence. The areas with the densest pattern correspond to unsegmented vectors.**

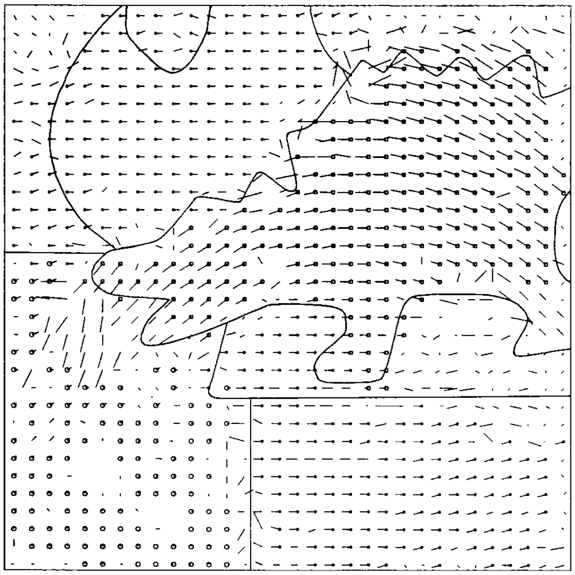

**Figure 6. Grouping flow vectors into segments, and showing the "correct" boundaries.**

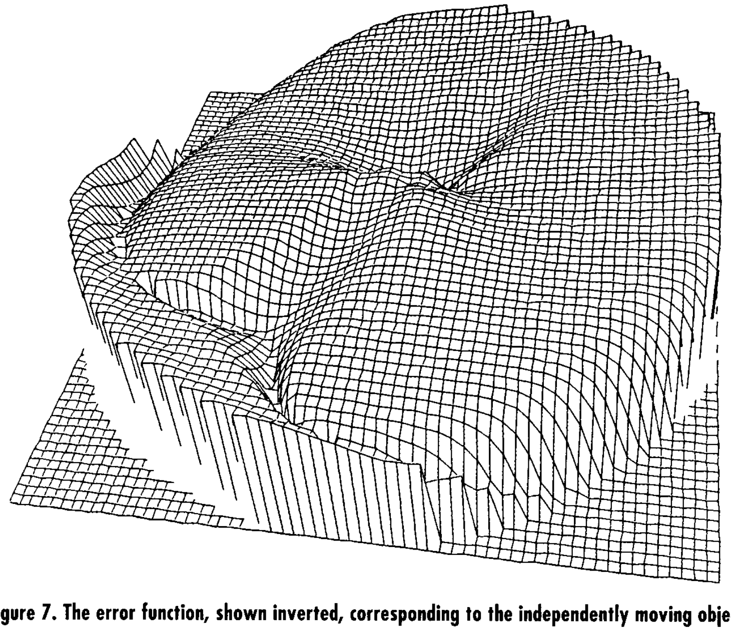

**Figure 7. The error function, shown inverted, corresponding to the independently moving object.**

are inaccurate because of errors in the computed flow field, and because of the continuity of flow fields that cross boundaries between regions. These regions correspond to surfaces at similar depths relative to their distance from the sensor. Since segmentation is based only on the flow field, the algorithm does not use information from intensity images (other than that used to compute the flow vectors). To evaluate the segmentation results, we must examine Figure 6, which shows not only the flow field used as input to the segmentation process, but also how the flow vectors have been grouped into segments using various shapes of the vector tails. The "correct" boundaries are also drawn. In general, we should combine flow-field segmentation and interpretation with intensity-data analysis.

The algorithm then determined a correct and unique grouping of segments into objects, and found three segments corresponding to stationary parts of the environment. The translation axis and rotation parameters of the camera were determined, and were in reasonable agreement with the actual values. The error function corresponding to the independently moving object is shown inverted in Figure 7. The translation axis (manifested by the peaks in the figure) cannot be determined reliably.

This demonstrates the potential inability to recover the motion parameters of independently moving objects due to certain inherent ambiguities in all algorithms based on optical-flow analysis.[5] The first ambiguity is in recovering motion parameters from a noisy flow field generated by rigid motion, since there might be a large set of incorrect solutions that induce flow fields similar to the correct one. If the field of view corresponding to the region containing the interpreted flow field is small, and the depth variation and translation

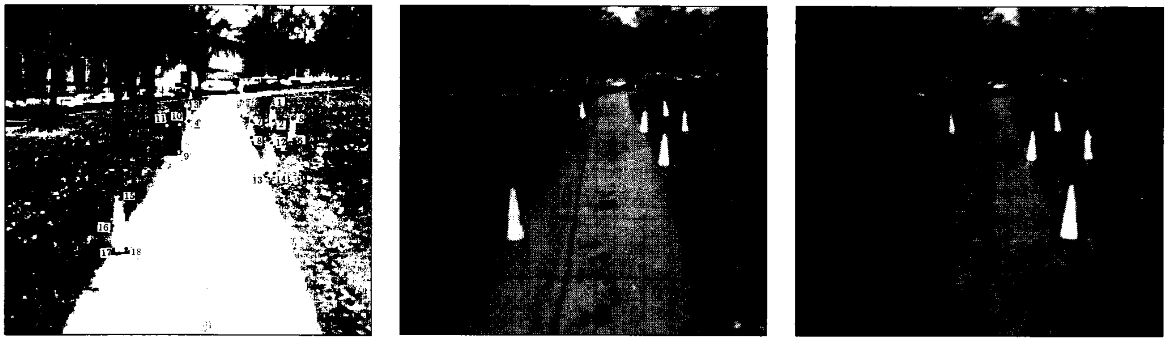

Figure 8. The CMU Navlab image sequence (from left to right): interest points marked on frame 1; frame 1 with displacement vectors for frames 1-3; frame 9 with displacement vectors for frames 9-11.

magnitude are small relative to the object's distance from the camera, then the determination of 3D motion and structure will be very sensitive to noise — and practically impossible in the presence of a realistic noise level.

The second ambiguity is in the decomposition of the flow field into sets of vectors corresponding to independently moving objects. Two independently moving objects can induce optical flows that are compatible with the same motion parameters; therefore, there is no way to refute the hypothesis that one rigid object generated those flows. The rigidity assumption has been found to be inappropriate for noisy flow fields.[6] That is, the consistency of flow vectors with the same motion parameters, up to the estimated noise level, does not guarantee that they are really induced by the motion of one rigid object. In this work, we assume[4] that a connected set of flow vectors consistent with a planar surface's rigid motion is induced by a rigid motion. This assumption is weaker than the first version of the rigidity assumption in the sense that it can only be applied in more restricted situations and, therefore, it is more likely to be correct.

**Passive ranging using optical flow.** Figure 8a shows a typical image from a set of data, collected from the Carnegie Mellon University Navlab, that was used to evaluate the performance of the algorithms discussed earlier.[7] To speed up computation, flow fields were restricted to 100 distinctive points (labeled in Figure 8a) that were obtained using the Moravec operator on the images.[8] The flow fields are shown in Figures 8b and 8c for image pairs 1-3 and 9-11, respectively, taken at four-foot intervals. The segmentation part of the algorithm was unnecessary because there is only one moving object. The algorithm obtained the motion parameters shown in Table 1. The translation results are all the same. Since the vehicle was trying to execute pure translation over this sequence, the algorithm found the translation vector quite well. The results also indicate an approximate constant rotation of 0.4-0.5 degrees about the $y$ axis, a small and varying component about the $x$ axis, and a random rotation about the $z$ axis. This is consistent with

- a road surface that deviates slightly from being planar, either because of bumps or because the surface itself is nonplanar ($x$ axis);
- a small drift of the vehicle to the right ($y$ axis); and
- some random motion, probably due to vehicle roll ($z$ axis).

However, we did not have any measurements of the rotation components to verify how accurate the derived rotational parameters are. Using the same set of algorithms, we measured the depth of the road obstacles for selected frame pairs (shown in Table 2) using 100 points (see Figure 8a). The average depth error is about 15 percent, which is promising for applications on real vehicles.

**Table 1. Results for motion parameters. (*U,V,W*) is the unit translational vector, and *A, B,* and *C* are the rotational components in degrees.**

| | Frame pairs | | | | |
|---|---|---|---|---|---|
| | 1-3 | 3-5 | 5-7 | 7-9 | 9-11 |
| *U* | –0.09 | –0.09 | –0.09 | –0.09 | –0.09 |
| *V* | –0.25 | –0.25 | –0.25 | –0.25 | –0.25 |
| *W* | –0.96 | –0.96 | –0.96 | –0.96 | –0.96 |
| *A* | –0.19 | 0.17 | –0.10 | –0.04 | –0.03 |
| *B* | 0.39 | 0.56 | 0.53 | 0.49 | 0.43 |
| C | –0.30 | 0.01 | 0.07 | 0.06 | 0.28 |

## Feature-based approaches

Feature-based approaches generally involve

- extracting a set of reliable features, such as "interesting" points (like corners), lines, contours, and regions;
- finding the corresponding features in multiple images using matching operations; and
- computing motion estimates based on a series of correspondences.

**Line correspondences: depth-from-looming structure.** Researchers at the University of Massachusetts developed a method to approximate perspective projection using a scaled orthographic projection when the depth to the centroid of an environmental structure is large with respect to the camera's focal length, and the total extent in depth of the structure is small compared to the depth of its centroid. An environmental structure satisfying these two requirements is called a shallow structure.[9] Assuming that an environmental structure of length $L$ satisfies the shallow-structure requirement and lies at a distance $z$ from the image plane, its projected length is $l_0 = Lf/z$, where $f$ is the camera's focal length. If the imaging device is translating into the environment with velocity $T$, then

**Table 2. Depth values for selected frame pairs of some points over a sequence of frames. * indicates that the point was not among the top 100 Moravec points. ** indicates that the point is absent in the image pair.**

| | | Frame pairs | | | |
|---|---|---|---|---|---|
| | | 1-3 | 1-3 | 9-11 | 9-11 |
| Object | Point | Experimental depth (feet) | True depth (feet) | Experimental depth (feet) | True depth (feet) |
| Cone 1 | 1 | 65.7 | 76 | 61.2 | 60 |
| | 2 | 66.9 | 76 | 59.6 | 60 |
| Cone 2 | 3 | 61.4 | 76 | 63.9 | 60 |
| | 4 | 60.8 | 76 | 61.7 | 60 |
| Cone 3 | 5 | 50.2 | 56 | 38.4 | 40 |
| | 6 | 51.1 | 56 | 38.5 | 40 |
| Cone 4 | 7 | 59.3 | 56 | 37.9 | 40 |
| | 8 | 46.3 | 56 | 39.8 | 40 |
| Can | 9 | 44.1 | 46 | 39.8 | 30 |
| | 10 | * | 46 | * | 30 |
| | 11 | * | 46 | • | 30 |
| Cone 5 | 12 | 31.0 | 36 | 20.0 | 20 |
| | 13 | 31.1 | 36 | 20.8 | 20 |
| | 14 | 31.9 | 36 | 20.5 | 20 |
| Cone 6 | 15 | 18.1 | 21 | ** | ** |
| | 16 | 18.4 | 21 | ** | ** |
| | 17 | 18.9 | 21 | ** | ** |
| | 18 | 18.6 | 21 | ** | ** |

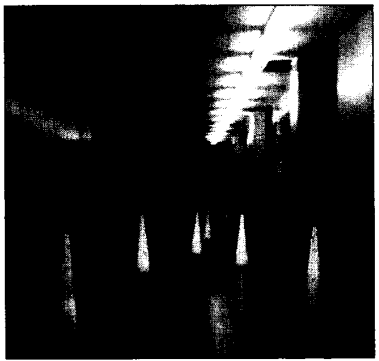

**Figure 9. The first frame of a motion sequence taken by a mobile robot moving down a hallway.**

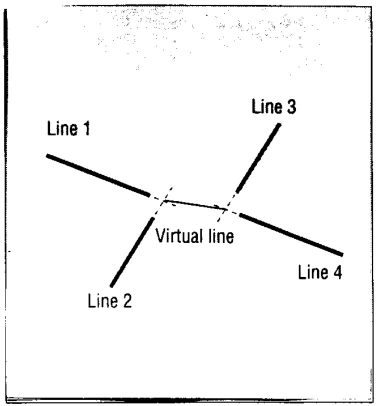

**Figure 10. Defining virtual lines with pairs of line segments.**

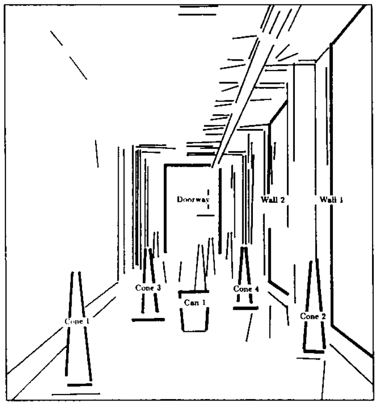

**Figure 11. The line segments used to define virtual lines for depth-from-looming experiments.**

the velocity component in the direction of gaze is $T_z = T \cdot z = T \cos\theta$, where $\theta$ is the angle between the direction of gaze and the focus of expansion. After some time $t$, the projected length is $l_1 = Lf/(z - T_z t)$. Solving for $z$, we get $z = (T_z t) / (1 - l_0/l_1)$, where $l_0$ and $l_1$ are the lengths of the projection of some environmental structure. In effect, the rate at which the structure grows over the image sequence provides the depth of the 3D structure, and therefore is termed "depth from looming." Thus, environmental depth can be determined without recovering motion parameters.

Figure 9 shows the first frame of a motion sequence in which a robot is moving down a hallway. First, line segments are extracted and matched. The lengths of the line segments are not reliable, but their orientation and lateral placement are accurate. This fact is exploited to define a virtual line whose length can be measured accurately over the course of the motion sequence. The endpoints of the virtual lines are defined by the intersections of two pairs of line segments (see Figure 10), and we can use the same technique to obtain virtual regions. Based on its knowledge of the correspondence of the line segments defining the virtual line over time, the system has information about the changing parameters of the virtual line itself. For the results presented here, we manually selected the virtual lines to be tracked in the first image. Organization principles of spacing, parallelness, orthogonality, and symmetry can be used to automatically extract the straight-line configurations used in these experiments. The virtual lines appear with labels in Figure 11. Tables 3 and 4 show the depth-from-looming values we obtained for computed and ground-truth depths to scene entities, the percent errors, and the number of frames contributing to the depth estimates. We used virtual lines to obtain the results in Table 3. The technique was even more accurate when we used virtual regions, as shown in Table 4. This method does not need the precise position of the focus of expansion (defined in the companion article).

**Region correspondences over multiple frames.** Much of the early work on estimating 3D motion from a sequence of matching points suffered from several problems, including

- motion relative to a fixed coordinate system,
- using only small motions and thus high sampling rates,
- restrictive assumptions on motions or feature points,
- noise sensitivity,

**Table 3. Computed and ground-truth depth values to scene entities, using depth from looming via virtual lines.**

| Virtual line | Computed depth (in feet) | Ground-truth depth (in feet) | Percent error | No. of frames used to estimate depth |
|---|---|---|---|---|
| Cone 1 | 19.1 | 20.0 | 4.5 | 1 |
| Cone 2 | 23.6 | 25.0 | 5.6 | 3 |
| Cone 3 | 28.3 | 35.0 | 19.1 | 1 |
| Cone 4 | 42.1 | 40.0 | 5.3 | 7 |
| Can 1 | 29.0 | 30.0 | 3.3 | 7 |
| Wall 1 | 27.7 | 27.1 | 2.2 | 2 |
| Wall 2 | 48.8 | 48.7 | 0.2 | 7 |
| Doorway | 88.8 | 87.1 | 2.0 | 7 |

**Table 4. Computed and ground-truth depth values to scene entities, using depth from looming via virtual regions.**

| Virtual region | Computed depth (in feet) | Ground-truth depth (in feet) | Percent error | No. of frames used to estimate depth |
|---|---|---|---|---|
| Cone 1 | 20.1 | 20.0 | 0.5 | 1 |
| Cone 2 | 25.8 | 25.0 | 3.2 | 3 |
| Cone 3 | 35.5 | 35.0 | 1.4 | 1 |
| Cone 4 | 40.0 | 40.0 | 0.0 | 7 |

- high complexity, and
- using only two frames, an inefficient use of available information.

Researchers at the University of Southern California have developed a system to address these problems.[10,11] Using three or more frames, the system estimates a moving object's motion parameters in terms of a natural center of motion. The motion parameters (expressed as rotation about the axis and translation of that center of rotation) are assumed to be constant over the relevant image sequence. The advantage of using more frames lies in the information content over time. The fact that the third frame is captured $\delta t$ after the second frame and $2\delta t$ after the first frame gives more constraints than when the frames are considered only two at a time, or at varying times. The system uses these constraints to derive equations that compute motion parameters when a few points are matched in several frames of the sequence (three points in three frames, two points in four frames, and one point in five frames). The algorithm leads to a set of difference equations across a sequence of images, relating a feature's position in the image plane to the projected point's motion parameters. The solution obtained for one point in five frames consists of a set of fifth-order nonlinear polynomial equations in the unknown motion parameters, whose solution requires a Gauss-Newton nonlinear least-squares method, from numerical analysis, with carefully defined initial-guess schemes.

Figure 12 illustrates the structure of the motion estimation system. Four modules perform the primary computation: the general-motion estimator, the pure-motion estimator, the pure-translation estimator, and the initial-guess generator. The system treats motion that is composed only of rotation and translation as a special case of the general-motion system.

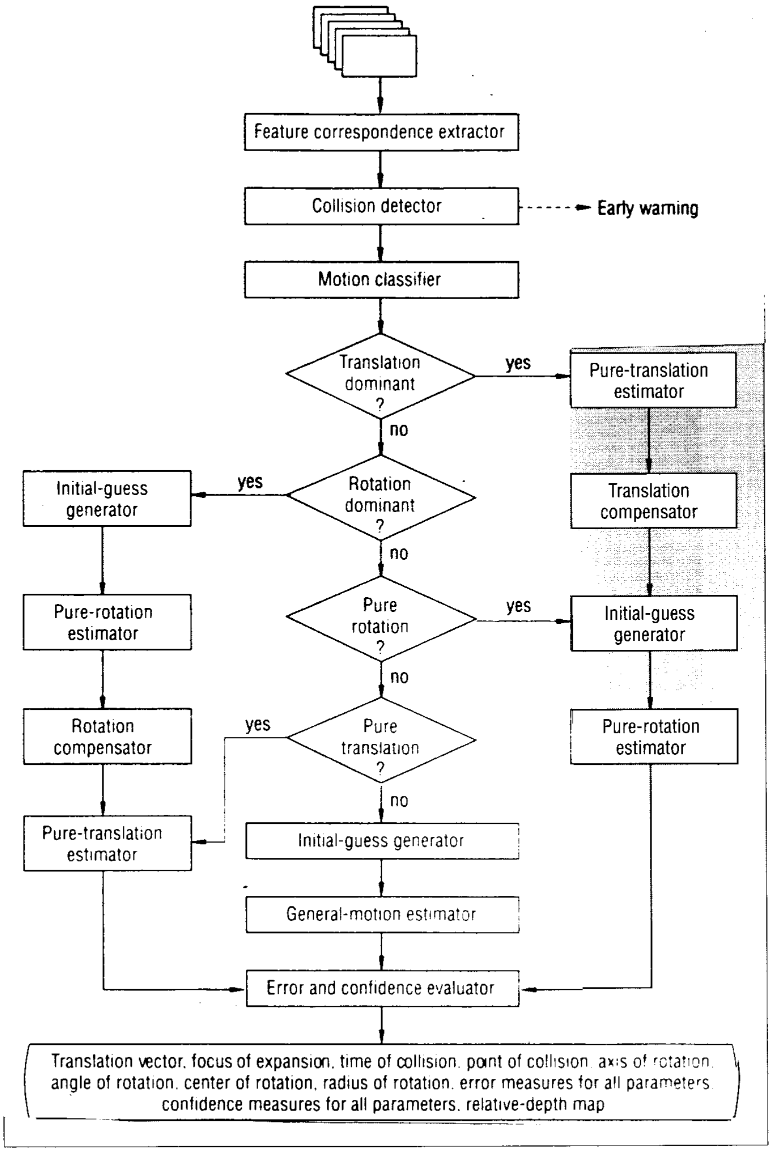
Feature correspondence extractor
Collision detector
Early warning
Motion classifier
Translation dominant ?
yes
Pure-translation estimator
no
Translation compensator
Initial-guess generator
yes
Rotation dominant ?
no
Pure rotation ?
yes
Initial-guess generator
Pure-rotation estimator
no
Rotation compensator
yes
Pure translation ?
Pure-rotation estimator
no
Pure-translation estimator
Initial-guess generator
General-motion estimator
Error and confidence evaluator
Translation vector, focus of expansion, time of collision, point of collision, axis of rotation, angle of rotation, center of rotation, radius of rotation, error measures for all parameters, confidence measures for all parameters, relative-depth map

**Figure 12. Control flow for the region-based motion estimation approach.**

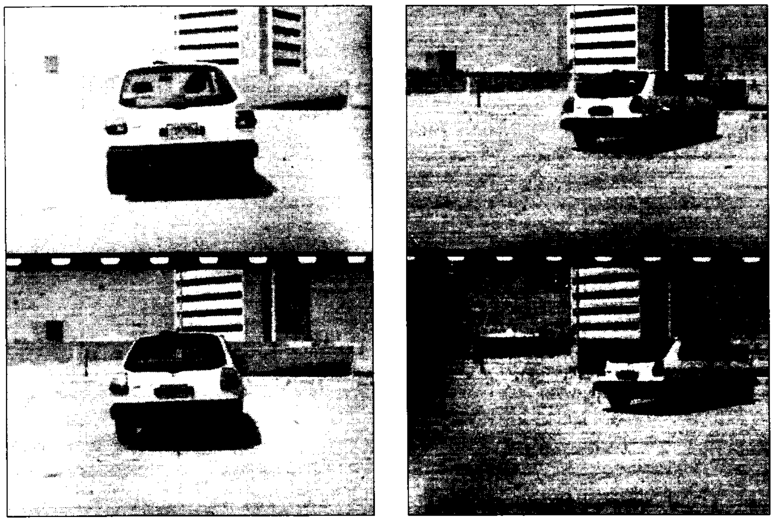

Figure 13. Four frames of the turning car: first and second frames on the left; fifth and sixth frames on the right.

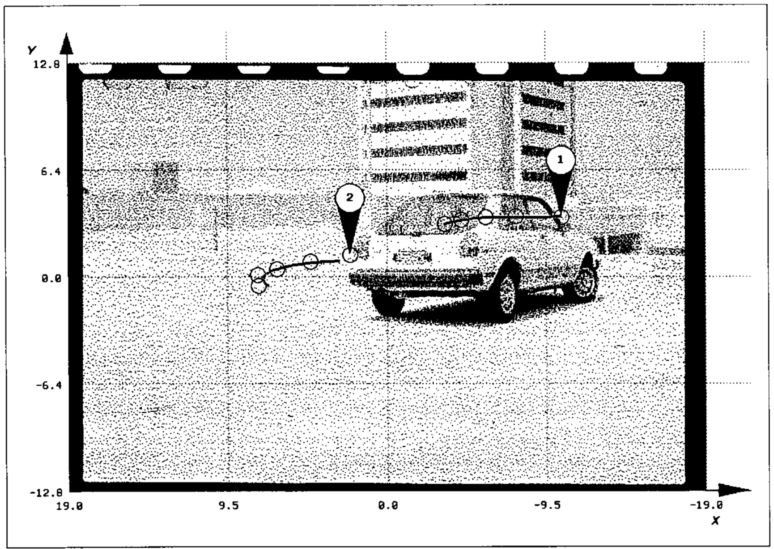

Figure 14. Comparing input data (open circles) to calculated motion (solid curve) for a turning car.

The feature-correspondence module treats regions as the basic feature. Using a recursive region-splitting technique,[12] the module finds compact regions that are relatively uniform in such properties as intensity, color, or texture. The module then uses the properties of these regions and the relations between them to find corresponding regions in successive frames.[13] This technique produces good data for the motion estimation program, where segmentation variations cause most problems. The system runs the segmentation program separately on each frame without any guidance from previous frames, and performs a series of pairwise matches using motion estimates to predict image positions.

The motion classification module determines which motion estimation technique (general, translation, or rotation) to apply. It uses the magnitude of the translation vector, the average length of the disparity vector, and the closeness of fit between the input data and a pure translation estimate. For the simple cases of pure rotation or translation, the system estimates the motion parameters with fewer data points. The system can compensate for dominant translation or rotation before computing the other motion parameters. The iterative Gauss-Newton-based solution of the motion parameters requires good starting points from the initial-guess generator, which automatically generates sets of possible initial solutions for the parameters. The general-motion estimator module starts with initial guesses to iteratively solve the general-motion equations.[14] This module derives error measures and confidence measures (how confident the program is in the error measure) for each parameter. The system produces the final parameter value by averaging the multiple calculations that are possible for each motion parameter, and calculates the final error measure. Larger values of error and confidence measures indicate the unreliability of the calculated parameter.

This motion computation was implemented on a Symbolics 3645. Figure 13 shows four frames from a six-frame sequence of a turning car. This sequence was taken by a motor-driven 50-millimeter camera at about three frames per second while the car was driven at a constant speed in a circle. The calculated motions are scaled to the camera's focal plane and are thus given in millimeters. The dimension of the image is 36x24 millimeters. The object's exact motion is unknown, so we can only qualitatively determine if the answers are correct. We can compare the results of locating points on the object to determine if they are consistent. The example shows that this approach can cope with noisy data and generate motion estimates that explain the actual image-plane motions. The segmentation produces many regions, but most represent stationary background objects. The motion classifier determines that most of the sequences of five or six matching regions (a region tracked through five or six frames) were pure or dominant rotations. The results for two sets of points are presented in Figure 14, which shows the fifth frame of the sequence. Sequence 1 corresponds to the right-hand rear window

Table 5. Results for general motion in a turning-car sequence.

| | Feature 1 | Feature 2 |
|---|---|---|
| Translation vector | (–0.23, –0.60, –10.73) | (–0.33, –0.31, 4.71) |
| Axis of rotation | (–0.42, –0.86, 0.30) | (0.03, –0.96, 0.28) |
| Amount of rotation around the axis of rotation | 69.05° | 23.12° |
| Center of rotation | (–5.66, 3.46, 50.01) | (–0.60, 0.46, 49.98) |
| Radius of rotation | 0.34 | 7.08 |
| Error measure/confidence measure | < .007-13/70-100 | 11-23/56-100 |
| Time (sec.) | 82 | 5,307 |

frame matched in frames 2 through 6, with the last point (at the pointer) being its location in the final, sixth frame. Sequence 2 is the left-hand tail light for frames 1 through 5. The results are summarized in Table 5. The points used are the centers of the regions in each image, and are plotted as open circles on the resultant image. The times in Table 5 include some overhead operations, but not the graphic display of results.

The first feature almost fits the translation model (the points appear almost along a straight line); the small radius of rotation agrees with this (see Table 5). The second feature is very close to a pure rotation, but the general-motion computation yields a better fit to the data than the pure rotational fit. The difference in motion of the two regions is partially explained by the changes in the region segmented from the second to the sixth frame. The initial region is just the narrow strip between the rear window and the side window, but the final region includes the entire window frame. Choosing different points in the region, such as the boundary, could result in a more accurate feature location and should result in a better estimation of the motion. The highest error measures and the lowest confidence measures in Table 5 are for the computation of the axis of rotation. We should expect this since the axis of rotation is calculated from the cross-product of two vectors and hence is sensitive to the noise in each vector.

**Contour correspondences over multiple frames.** Earlier, we described the use of regions as features to match for motion computation. Regions are rather global features and likely to change over large displacements. An alternative is to use features based on the contours, or boundaries, of objects. Contours are more local; their shape is more likely to be preserved, at least in part, over large displacements. Several low-level features related to contours such as edges, linear line segments, and corners have been commonly used in previous work. These features are easier to match than the contours themselves; however, they ignore important information inherent in contours. Edge and corner matching do not use continuity information, and straight-line matching ignores curvature information. Further, detecting corners on smooth curves (for instance, by linear-segment approximation) is often arbitrary, and the resulting corners do not necessarily correspond in the sequences.

Researchers at the University of Southern California have developed a technique to match contours directly rather than matching their derived features.[15] Matching contours presents several problems. Contours in an image sequence can merge or split due to occlusion, noise, errors of the edge linker, or other reasons. Thus, only parts of the detected contours might

## Glossary

**Canny operator:** An optimal filter to detect intensity edges in images.

**Finite-element method:** A computational method to solve differential equations by making discrete approximations to them, for example, by replacing every derivative with a difference quotient.

**Flow vectors:** The vectors representing the apparent motion of the brightness pattern at image pixels.

**Generalized Hough transform:** A transform for detecting entities in images that have no simple analytical form. The basic strategy is to trade off work in parameter space for work in image space.

**Gradient-based techniques:** Computational methods involving evaluation of derivatives of parameters. Typically, they use spatial and temporal gray-level variations to estimate the instantaneous velocity at each pixel.

**Ground truth:** True values of measurable physical quantities.

**Matching technique:** Algorithm to find correspondences between entities.

**Optical flow field:** The apparent motion of the brightness pattern in an image due to the realistic motion between a camera and the scene.

**Orthographic projection:** The special case of perspective transformation where there is no distortion for the spatial coordinates ($x$ and $y$), and the viewpoint is at infinity in the $z$ direction. For this projection, light rays travel parallel to the imaging system's optic axis to impinge on the image plane.

**Perspective projection:** A first-order approximation to the process of taking a picture. In this approximation, a light ray originating at a point in the scene travels through a single point of an imaging system (pinhole lens) to reach the image plane located at the camera's focal length.

**Smoothness assumption:** Spatial variations in measurements are not abrupt but smooth. For example, this assumption implies that the displacement field varies smoothly over the image area covered by a single surface.

**SSD surface:** The surface defined over the space of displacements. Its height is the SSD value corresponding to each displacement.

**Sum of squared differences (SSD):** A function whose value at a point is the sum of the squared differences of image intensities within a window (whose center is represented by the point) placed on two images.

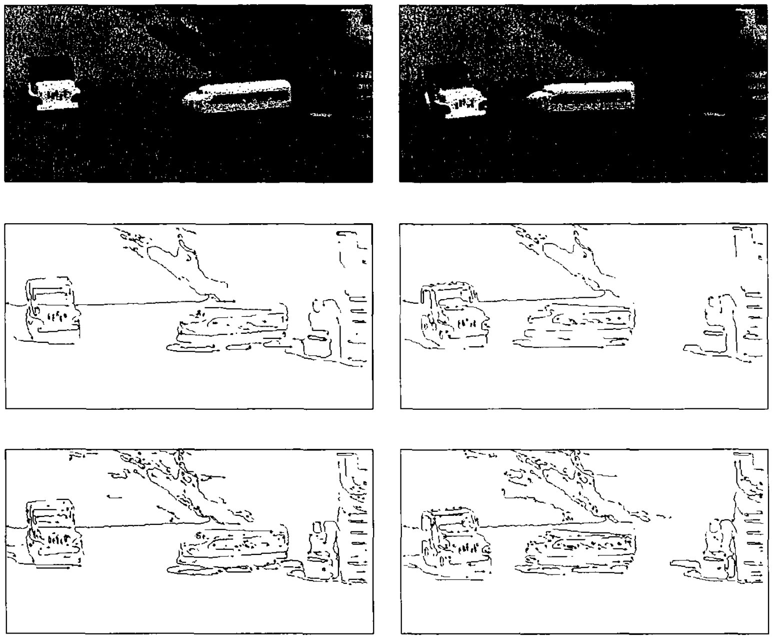

**Figure 15. Two images (size 250x512 pixels) of a sequence containing a toy jeep and train (from left to right and top to bottom): frame 1; frame 2; detected contours in frame 1 at scale parameter 8; detected contours in frame 2 at scale parameter 8; detected contours in frame 1 at scale parameter 4; detected contours in frame 2 at scale parameter 4.**

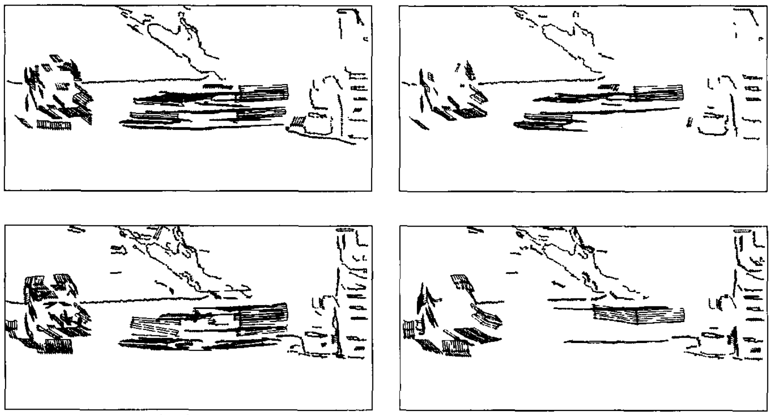

**Figure 16. The results for contour correspondence using three frames (from left to right and top to bottom): matches at scale parameter 8; predictions at scale parameter 4; matches at scale parameter 4; multiple matches at scale parameter 4.**

match. The technique overcomes this difficulty by dividing each contour into several "sections" of equal length (chosen as a function of the contour's total length), each of which is matched independently by sliding it along the contour to be matched and maximizing the similarity. Each match is then extended by adding points until the similarity starts to decrease. This approach has the advantage that a section is less likely to share more than one object and hence its match is more likely to be found. However, this step leaves possible multiple matches for parts of contours.

The system resolves multiple matches through a simple relaxation process. It evaluates each match based on contour similarity and also receives support from other matches if they indicate similar translational motion. Matches not receiving adequate support are eliminated. The system merges overlapping matches, using the section with the highest evaluation first. This process is applied across several frames to give multiple section matches over multiple frames.

The entire matching process is also performed across different scales. The system computes the edges forming contours by first using an adaptive smoothing technique to smooth the image.[16] Different-sized masks give different features; however, the adaptive smoothing technique has the property that feature position does not move by more than a pixel. Thus, the system can use features matched at a coarser scale (through larger masks) to predict feature matches at finer scales. This reduces the computational complexity of the matching process and prevents spurious matches.

Figure 15 shows results using this technique. Figures 15a and 15b are the first two frames of a sequence taken by a stationary camera and consisting primarily of two moving toy vehicles. This is a difficult scene to analyze since the motion is both very large (disparity of at least 70 pixels) and is a general 3D motion. We can easily see that the train is moving rapidly toward the jeep. We detect contours from these images by adaptively smoothing the images, detecting edges using a Canny operator,[17] and finally linking the edges. Figures 15c and 15d show the detected contours in the two frames using adaptive smoothing and a mask size of 8 pixels. Figures 15e and 15f show the contours detected using a mask size of 4 pixels. Figure 16 shows the results of matching the contours in Figure 15, with additional lines drawn between the points on the sections that match. Only those points are included for which a match was found, and an arrow is drawn to the closest point in the other section (after translating to start at the same location). For the sake of clarity, arrows are drawn only for every fifth point in a matching section. Figure 16a shows the contour matches detected at an 8-pixel scale, Figure 16b shows the predictions for expected matches at a 4-pixel scale, and Figure 16c the computed matches at a 4-pixel scale. Figure 16d shows matches obtained across three frames. These results show good performance over large displacements. This technique has also been tested on natural, outdoor scenes and works well if the shape of objects in the scene does not change significantly.

CONSIDERABLE EFFORT WILL be needed to develop working motion-understanding systems that perform robustly in view of the multitude of problems that occur in real-world situations. There are several technical issues and areas where advances must be made so that robust dynamic-scene and motion analysis can be achieved.

*Data availability.* It has been difficult to evaluate algorithm performance quantitatively because of the unavailability of outdoor data with ground-truth information that includes robot motion, depth to scene entities, and the motion of independently moving objects. Recently, attempts have been made to collect such data,[7,18] which will significantly help researchers develop robust algorithms. An effort by the community of motion researchers is now underway.

*Robust algorithms.* We need to develop robust and noise-insensitive techniques for real-time practical applications. To recover depth in practical situations, dense displacement fields must be able to separate rotation from translation with a rotational error of less than half a degree.[19] We need robust techniques for feature correspondences, computation of flow fields, computation of the focus of expansion, decomposition of sensor rotation and translation, detection of moving objects in high-clutter and low-contrast situations, generation of precise 3D descriptions of moving objects (rigid and nonrigid), and techniques for accurate passive ranging.

*Computational throughput.* Dynamic-scene and motion analysis is computationally intensive. Ultimately, the system requirements dictate the desired throughput. To avoid obstacles, for example, helicopters should accomplish all necessary processing (including display and presentation of obstacle information to the pilot) within one second.[20] We need not only efficient algorithms but also much more powerful hardware for real-time motion analysis.

*Integration of motion and binocular stereo.* To obtain accurate depth measurements for the sensor's entire field of view and to detect depth and motion boundaries, we need to do more research on integrating motion analysis with binocular stereo.[21,22]

*Occlusion.* We'd like to be able to robustly detect occlusion and motion boundaries. Otherwise, significant errors in flow field interpretation will probably result.[2,4]

***CONSIDERABLE EFFORT WILL BE NEEDED TO DEVELOP MOTION-UNDERSTANDING SYSTEMS THAT PERFORM ROBUSTLY IN VIEW OF THE PROBLEMS THAT OCCUR IN REAL-WORLD SITUATIONS.***

*Independently moving objects.* We probably cannot get accurate motion parameters for moving objects at the typical camera resolution (512×512 pixels per image) and distance from other moving objects. We need to see whether these motion parameters can be bounded to get approximate or qualitative results.

*Surface reconstruction from motion.* We need algorithms to reliably interpolate sparse range maps so as to provide a rendition of 3D surfaces.

*Integration of motion with other cues.* We would like to be able to plan camera motion to build symbolic maps of the environment. We'd also like to be able to use observer motion for improved recognition, which can be used to further improve estimation for motion parameters. Using an inertial-navigation system and digital map information would also allow for additional constraints in motion analysis, which would facilitate accurate passive ranging[18] and tracking.[23]

*Environment.* For technology transfer, it would be valuable to develop an environment and a system that lets us simulate different algorithms end to end for dynamic-scene and motion analysis.

*Systems.* We need to develop systems that integrate motion analysis, binocular stereo analysis, landmark acquisition and recognition, and clutter rejection algorithms. In such systems, a mobile robot could navigate in practical scenarios using auxiliary information such as digital maps, landmark recognition and acquisition, prediction of object motion, and the tracking of partially occluded objects.

*Visualization of results.* It is difficult to present the results of dynamic-scene and motion analysis for human perception. We need to develop new visualization techniques for presenting analysis results and displaying computed range values and 3D environmental depth maps. This requires the development of specialized software and hardware so that we can simulate the algorithms in the laboratory and fully understand their strengths and weaknesses.

## Acknowledgments

This article describes the work of several of our colleagues at our respective research laboratories. We particularly wish to thank Salit Gazit, Gérard Medioni, Keith Price, Wilhelm Burger, Barry Roberts, and Harpreet Sawhney for their help in preparing and reading this article.

## References

1. P. Anandan, "A Computational Framework and an Algorithm for the Measurement of Visual Motion," *Int'l J. Computer Vision*, Vol. 2, No. 3, 1989, pp. 283-310.

2. P. Anandan, *Measuring Visual Motion from Image Sequences*, doctoral dissertation, Univ. of Massachusetts, Amherst, Mass., 1987. Also as Computer and Information Sciences Tech. Report 87-21.

3. B.K.P. Horn and B.G. Schunck, "Determining Optical Flow," *Artificial Intelligence*, Vol. 17, Nos. 1-3, Aug. 1981, pp. 185-203.

4. G. Adiv, "Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-7, No. 4, 1985, pp. 384-401.

5. G. Adiv, "Inherent Ambiguities in Recovering 3D Motion and Structure from a Noisy Flow Field," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-11, No. 5, May 1989, pp. 477-489.

6. S. Ullman, *The Interpretation of Visual Motion*, MIT Press, Cambridge, Mass., 1979.

7. R. Dutta et al., "Issues in Extracting Motion Parameters and Depth from Approximate Translation Motion," *Proc. DARPA Image-Understanding Workshop*, Morgan Kaufmann, San Mateo, Calif., 1988, pp. 945-960.

8. H.P. Moravec, "Towards Automatic Visual Obstacle Avoidance," *Proc. Fifth Int'l Joint Conf. Artificial Intelligence*, Morgan Kaufmann, San Mateo, Calif., 1977, p. 584.

9. L.R. Williams and A.R. Hanson, "Depth from Looming Structure," *Proc. DARPA Image-Understanding Workshop*, Morgan Kaufmann, San Mateo, Calif., 1988, pp. 1,047-1,051.

10. H. Shariat and K.E. Price, "Results of Motion Estimation with More Than Two Frames," *Proc. DARPA Image-Understanding Workshop*, Morgan Kaufmann, San Mateo, Calif., 1987, pp. 694-703.

11. H. Shariat and K.E. Price, "Motion Estimation with More Than Two Frames," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-12, No. 5, May 1990, pp. 417-434.

12. R. Ohlander, K. Price, and D.R. Reddy, "Picture Segmentation Using a Recursive Region-Splitting Method," *Computer Graphics and Image Processing*, Vol. 8, No. 3, Dec. 1978, pp. 313-333.

13. O.D. Faugeras and K.E. Price, "Semantic Description of Aerial Images Using Stochastic Labeling," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-3, No. 6, Nov. 1981, pp. 633-642.

14. H. Shariat, "The Motion Problem: How to Use More Than Two Frames?," Tech. Report IRIS 202, Inst. for Robotics and Intelligent Systems, Univ. of Southern California, Los Angeles, 1986.

15. S.L. Gazit and G. Medioni, "Multiscale Contour Matching in a Motion Sequence," *Proc. DARPA Image-Understanding Workshop*, Morgan Kaufmann, San Mateo, Calif., 1989, pp. 934-943.

16. P. Saint-Marc and G. Medioni, "Adaptive Smoothing for Feature Extraction," *Proc. DARPA Image-Understanding Workshop*, Morgan Kaufmann, San Mateo, Calif., 1988, pp. 1,100-1,113.

17. J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, Nov. 1986, pp. 679-698.

18. B. Roberts and B. Bhanu, "Inertial-Navigation Sensor-Integrated Motion Analysis for Autonomous Vehicle Navigation," *Proc. DARPA Image-Understanding Workshop*, Morgan Kaufmann, San Mateo, Calif., 1990, pp. 364-375.

19. R. Dutta and M.A. Snyder, "Robustness of Correspondence-Based Structure from Motion," *Proc. Third Int'l Conf. Computer Vision*, CS Press, Los Alamitos, Calif., 1990, pp. 106-110.

20. B. Bhanu and B. Roberts, "Obstacle Detection During Rotorcraft Low-Altitude Flight and Landing," Second Annual Tech. Report to NASA Ames Research Center by Honeywell Systems and Research Center, Minneapolis, Minn., Aug. 1990.

21. P.F. Symosek et al., "Motion and Binocular Stereo Integrated System for Passive Ranging," *Proc. DARPA Image-Understanding Workshop*, Morgan Kaufmann, San Mateo, Calif., 1990, pp. 356-363.

22. P. Balasubramanyam and M.A. Snyder, "The P-Field: A Computational Model for Binocular Motion Processing," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, CS Press, Los Alamitos, Calif., June 1991, pp. 115-120.

23. B. Bhanu et al., "Qualitative Target Motion Detection and Tracking," *Proc. DARPA Image-Understanding Workshop*, Morgan Kaufmann, San Mateo, Calif., 1989, pp. 370-398.

24. R. Dutta et al., "A Data Set for Quantitative Motion Analysis," *Proc. DARPA Image-Understanding Workshop*, Morgan Kaufmann, San Mateo, Calif., 1989, pp. 714-720.

**Bir Bhanu** is professor of electrical engineering and computer science and director of the Visualization Research Laboratory at the University of California at Riverside. (His photo appears on p. 52.) He began researching qualitative motion understanding while a senior fellow at Honeywell Systems and Research Center. His research interests include machine learning, robot navigation, scientific visualization, target recognition and tracking, object modeling, multisensor integration, parallel algorithms, image-understanding algorithms and systems, and photo interpretation.

Bhanu received his SM and his EE in electrical engineering and computer science from MIT, his PhD in electrical engineering from the University of Southern California Image-Processing Institute, and an MBA from the University of California at Irvine. He also received an ME from the Birla Institute of Technology and Science and a BS from Banaras Hindu University. Bhanu is an associate editor of *Pattern Recognition* and *The Journal of Mathematical Imaging and Vision* and a member of the IEEE Computer Society, IEEE, ACM, AAAI, and the Pattern Recognition Society.

**Ramakant Nevatia** is professor of computer science and electrical engineering at the University of Southern California. (His photo appears on p. 52.) He also directs the Institute for Robotics and Intelligent Systems there. His research interests include computer vision, artificial intelligence, and robotics. He earned a BS and MA from the University of Bombay and a MS and PhD from Stanford University, all in electrical engineering.

Nevatia is an associate editor of the journals *Pattern Recognition* and *Computer Vision, Graphics, and Image Processing*. He is the author of two books, *Machine Perception* and *Computer Analysis of 3D Curved Objects*, and a member of the IEEE Computer Society, IEEE, AAAI, and ACM.

**Edward M. Riseman** is professor of computer and information science at the University of Massachusetts. (His photo appears on p. 52.) He also directs the Computer Vision Laboratory, with projects in knowledge-based scene interpretation, motion analysis, mobile robot navigation, and parallel architectures for real-time vision. He and his colleagues are applying the analysis of static and dynamic images to such domains as natural outdoor scenes, biomedical images, industrial robotic environments, aerial images, and satellite images. His research interests have also included computer vision, artificial intelligence, learning, and pattern recognition.

Riseman received his BS from Clarkson College of Technology and his MS and PhD in electrical engineering from Cornell University. He is a member of the IEEE Computer Society, IEEE, ACM, AAAI, and the Pattern Recognition Society.

Readers can reach the authors in care of Bir Bhanu, College of Engineering, University of California at Riverside, Riverside, CA 92521, or by e-mail to bhanu@shivish.ucr.edu