A Unified Approach to Moving Object Detection in 2D and 3D Scenes

Michal Irani and P. Anandan

Abstract—The detection of moving objects is important in many tasks. Previous approaches to this problem can be broadly divided into two classes: 2D algorithms which apply when the scene can be approximated by a flat surface and/or when the camera is only undergoing rotations and zooms, and 3D algorithms which work well only when significant depth variations are present in the scene and the camera is translating. In this paper, we describe a unified approach to handling moving-object detection in both 2D and 3D scenes, with a strategy to gracefully bridge the gap between those two extremes. Our approach is based on a stratification of the moving object-detection problem into scenarios which gradually increase in their complexity. We present a set of techniques that match the above stratification. These techniques progressively increase in their complexity, ranging from 2D techniques to more complex 3D techniques. Moreover, the computations required for the solution to the problem at one complexity level become the initial processing step for the solution at the next complexity level. We illustrate these techniques using examples from real-image sequences.

Index Terms—Moving object detection, rigidity constraints, multiframe analysis, planar-parallax, parallax geometry, layers.

1 Introduction

Moving object detection is an important problem in image sequence analysis. It is necessary for surveillance applications, for guidance of autonomous vehicles, for efficient video compression, for smart tracking of moving objects, and many other applications.

The 2D motion observed in an image sequence is caused by 3D camera motion (the egomotion), by the changes in internal camera parameters (e.g., camera zoom), and by 3D motions of independently moving objects. The key step in moving-object detection is accounting for (or compensating for) the camera-induced image motion. After compensation for camera-induced image motion, the remaining residual motions must be due to moving objects.

The camera-induced image motion depends both on the egomotion parameters and the depth of each point in the scene. Estimating all of these physical parameters (namely, egomotion and depth) to account for the camera-induced motion is, in general, an inherently ambiguous problem [3]. When the scene contains large depth variations, these parameters may be recovered. We refer to these scenes as *3D scenes*. However, in *2D scenes*, namely, when the depth variations are not significant, the recovery of the camera and scene parameters is usually not robust or reliable [3]. Sample publications that treat the problem of moving objects in 3D scenes are [4], [21], [30], [31], [9]. A careful treatment of the issues and problems associated with moving-object detection in 3D scenes is given in [29].

- *M. Irani is with the Department of Applied Math and Computer Science, The Weizmann Institute of Science, 76100 Rehovot, Israel. E-mail: irani@wisdom.weizmann.ac.il.*
- *P. Anandan is with Microsoft Corporation, One Microsoft Way, Redmond, WA 98052. E-mail: anandan@microsoft.com.*

Manuscript received 29 Jan. 1996; revised 25 Feb. 1998. Recommended for acceptance by B.C. Vemuri.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 106431.

An effective approach to accounting for camera-induced motion in 2D scenes is to model the image motion in terms of a global 2D parametric transformation. This approach is robust and reliable when applied to flat (planar) scenes, distant scenes, or when the camera is undergoing only rotations and zooms. However, the 2D approach cannot be applied to the 3D scenes. Examples of methods that handle moving objects in 2D scenes are [14], [7], [8], [10], [28], [24], [33], [5].

Therefore, 2D algorithms and 3D algorithms address the moving object-detection problem in very different types of scenarios. These are two extremes in a continuum of scenarios: flat 2D scenes (i.e., *no 3D parallax*) vs. 3D scenes with dense depth variations (i.e., *dense 3D parallax*). Both classes fail on the other extreme case or even on the intermediate case (when 3D parallax is *sparse* relative to amount of independent motion).

In real-image sequences, it is not always possible to predict in advance which situation (2D or 3D) will occur. Moreover, both types of scenarios can occur within the same sequence, with gradual transitions between them. Unfortunately, no single class of algorithms (2D or 3D) can address the general moving object-detection problem. It is not practical to constantly switch from one set of algorithms to another, especially since neither class treats well the intermediate case.

In this paper, we present a unified approach to handling moving-object detection in both 2D and 3D scenes, with a strategy to gracefully bridge the gap between those two extremes. Our approach is based on a stratification of the moving object-detection problem into scenarios which gradually increase in their complexity:

1) scenarios in which the camera-induced motion can be modeled by a single 2D parametric transformation,

0162-8828/98/$10.00 © 1998 IEEE

2) those in which the camera-induced motion can be modeled in terms of a small number of *layers* of parametric transformations, and
3) general 3D scenes, in which a more complete parallax motion analysis is required.

We present a set of techniques that match the above stratification. These techniques progressively increase in their complexity. Moreover, the computations required for the solution to the problem at one complexity level become the initial processing step for the solution at the next complexity level. In our approach, we always apply first the 2D analysis. When that is all the information that is contained in the video sequence, that is where the analysis should stop (to avoid encountering singularities). Our 3D analysis gradually builds *on top* of the 2D analysis, with the gradual increase in 3D information, as detected in the image sequence. After 2D alignment, there can be two sources for residual motions: *3D parallax* and *independent motions*. To distinguish between these two types of motions, we develop a new rigidity constraint based on the residual parallax displacements. This constraint is based on an analysis of the parallax displacements of a few points over multiple frames, as opposed to the epipolar constraint, which is based on many points over a pair of frames. As such, they are applicable even in cases where 3D parallax is very sparse and in the presence of independent motions.

The goal in taking this approach is to develop a strategy for moving object detection, so that the analysis performed is tuned to match the complexity of the problem and the availability of information at any time. This paper describes the core elements of such a strategy. The integration of these elements into a single algorithm remains a task for our future research. A shorter version of this paper appeared in [13].

2 2D Scenes

The instantaneous image motion of a general 3D scene can be expressed as in [22], [2]:

$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \begin{bmatrix} -\left(\frac{T_X}{Z} + \Omega_Y\right) + x\frac{T_Z}{Z} + y\Omega_Z - x^2\Omega_Y + xy\Omega_X \\ -\left(\frac{T_Y}{Z} + \Omega_X\right) - x\Omega_Z + y\frac{T_Z}{Z} - xy\Omega_Y + y^2\Omega_X \end{bmatrix} \quad (1)$$

where $(u(x, y), v(x, y))$ denotes the image velocity at image location, (x, y), $T = (T_X, T_Y, T_Z)^t$ denotes the translational motion of the camera, $R = (\Omega_X, \Omega_Y, \Omega_Z)^t$ denotes the camera rotation, and Z denotes the depth of the scene point corresponding to (x, y).

The instantaneous image motion (1) can often be approximated by a single 2D parametric transformation. Below, we review the conditions associated with the scene geometry and/or camera motion when such an approximation is valid.

1) *A planar surface:* When the scene can be modeled as a single planar surface, i.e., when $Z = A \cdot X + B \cdot Y + C$, where (X, Y, Z) are 3D scene coordinates and (A, B, C) denote the parameters that describe the plane, Eq. (1) can be reduced to:

$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \begin{bmatrix} a + b \cdot x + c \cdot y + g \cdot x^2 + h \cdot xy \\ d + e \cdot x + f \cdot y + g \cdot xy + h \cdot y^2 \end{bmatrix} \quad (2)$$

where the parameters (a, b, c, d, e, f, g, h) are functions of the camera motion (R, T) and the planar surface parameters (A, B, C). Thus, the image motion is described by an eight-parameter quadratic transformation in 2D [15].
2) *Distant Scene:* When the scene is very distant from the camera, namely, when the deviations from a planar surface are small relative to the overall distance of the scene from the camera, the planar surface model is still a very good approximation. In this case the 2D quadratic transformation describes the image motion field to subpixel accuracy. Moreover, as the overall distance $Z \to \infty$, then $\frac{T_x}{Z}, \frac{T_y}{Z}, \frac{T_z}{Z} \to 0$, i.e., the translational component of image motion is negligible. (This is similar to the case of pure rotation described below.) The "distant scene" conditions are often satisfied in remote surveillance applications, where narrow field-of-view (FOV) cameras (typically 5° or less) are used to detect moving objects in a distant scene (typically at least 1 km away).
3) *Camera Rotation:* When the camera undergoes a pure rotational motion (i.e., $T = 0$) or when the camera translation is negligible ($|T| \ll Z$), then Eq. (1) becomes

$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \begin{bmatrix} -\Omega_Y + y\Omega_Z - x^2\Omega_Y + xy\Omega_X \\ \Omega_X - x\Omega_Z - xy\Omega_Y + y^2\Omega_X \end{bmatrix}. \quad (3)$$

Thus the 2D image motion field is described by a quadratic transformation in this situation as well.
4) *Camera Zoom:* Finally, on top of its motion, when the camera zooms in, the image undergoes an additional dilation. The resulting image motion field can still be modeled as a quadratic transformation of the form of Eq. (2); the zoom will influence the parameters b and f.

We refer to scenes that satisfy one or more of the abovementioned conditions (and, hence, Eq. (2) is applicable) as *2D scenes*.

Under these conditions, we can use a previously developed method [6], [14] in order to compute the 2D parametric motion. This technique "locks" onto a "dominant" parametric motion between an image pair, even in the presence of independently moving objects. It does not require prior knowledge of their regions of support in the image plane [14]. This computation provides only the 2D motion parameters of the camera-induced motion, but no explicit 3D shape or motion information. To make the paper self-contained, we briefly review these steps for estimating these 2D motion parameters in the next few paragraphs. Note that this 2D estimation process is also used later as an initial step in the layered and the 3D analysis methods.

2.1 The 2D Parametric Estimation

A number of techniques have been described in the computer vision literature for the estimation of 2D parametric motion (e.g., [14], [7], [24], [33], [5], [16], [32]). In this paper, we follow the approach described in [14]. To make this presentation self-contained, we briefly outline this technique below.

We will refer to the two image frames (whose image motion is being estimated) by the names "inspection" im-

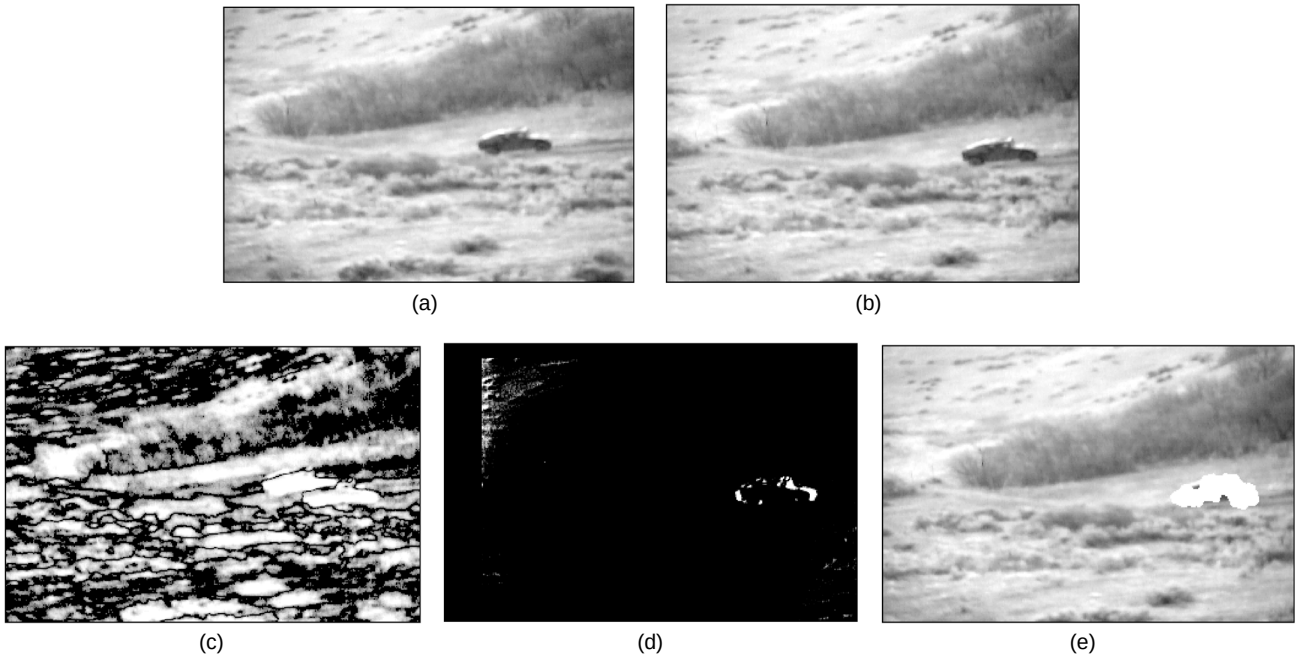


Fig. 1. Small 2D moving object detection. (a), (b) Two frames in a sequence obtained by a translating and rotating camera. The scene itself was not planar, but was distant enough (about 1 km away from the camera) so that effects of 3D parallax were negligible. The scene contained a car driving on a road. (c) Intensity differences before dominant (background) 2D alignment. (d) Intensity differences after dominant (background) 2D alignment. Nonoverlapping image boundaries were not processed. The 2D alignment compensates for the camera-induced motion, but not for the car's independent motion. (e) The detected moving object based on *local misalignment* analysis. The white region signifies the detected moving object.

age and "reference" image, respectively. A Laplacian pyramid is first constructed from each of the two input images and then estimates the motion parameters in a coarse-fine manner. Within each level the sum of squared difference (SSD) measure integrated over regions of interest (which is *initially* the entire image region) is used as a match measure. This measure is minimized with respect to the 2D image motion parameters.

The SSD error measure for estimating the image motion within a region is:

$$E(\vec{\alpha}) = \sum_{\mathbf{x}} \left(I(x, y, t) - I\left(x - u(x, y; \vec{\alpha}), y - v(x, y; \vec{\alpha}), t - 1\right)\right)^2 \quad (4)$$

where I is the (Laplacian pyramid) image intensity, $\vec{\alpha} = (a, b, c, d, e, f, g, h)$ denotes the parameters of the quadratic transformation, and $\left(u(x, y; \vec{\alpha}), v(x, y; \vec{\alpha})\right)$ denotes the image velocity at the location (x, y) induced by the quadratic transformation with parameters $\vec{\alpha}$ as defined in (2). The sum is computed over all the points within a region of interest, often the entire image.

The objective function E given in (4) is minimized w.r.t. the unknown parameters $\vec{\alpha} = (a, b, c, d, e, f, g, h)$ via the Gauss-Newton optimization technique. Let $\vec{\alpha}_i = (a_i, b_i, c_i, d_i, e_i, f_i, g_i, h_i)$ denote the current estimate of the quadratic parameters. After warping the inspection image (towards the reference image) by applying the quadratic transformation based on these parameters, an incremental estimate $\vec{\delta\alpha} = (\delta a, \delta b, \delta c, \delta d, \delta e, \delta f, \delta g, \delta h)$ can be determined. After iterating a certain number of times within a pyramid level, the process continues at the next finer level.

With the above technique, the reference and inspection images are registered so that the desired image region is aligned, and the quadratic transformation (2) is estimated. The above estimation technique is a least-squares-based approach and hence possibly sensitive to outliers. However, as reported in [7], this sensitivity is minimized by doing the least-squares estimation over a pyramid. The pyramid-based approach locks on to the dominant image motion in the scene.

A robust version of the above method [14] handles scenes with multiple moving objects. It incorporates a gradual refinement of the complexity of the motion model (ranging from pure translation at low resolution levels, to a 2D affine model at intermediate levels, to the 2D quadratic model at the highest resolution level). Outlier rejection is performed before each refinement step within the multiscale analysis. This robust analysis further enhances the locking property of the above-mentioned algorithm onto a single *dominant* motion.

Once the dominant 2D parametric motion has been estimated, it is used for warping one image towards the other. When the dominant motion is that of the camera, all regions corresponding to static portions of the scene are completely aligned as a result of the 2D registration (except for nonoverlapping image boundaries), while independently moving objects are not. Detection of moving objects is therefore performed by determining local misalignments [14] after the global 2D parametric registration.

Fig. 1 shows an example of moving-object detection in a "2D scene." This sequence was obtained by a video camera with an FOV of four degrees. The camera was mounted on a vehicle moving on a bumpy dirt road at about 15 km/hr and was looking sideways. Therefore, the camera was both translating and rotating (camera jitter). The scene itself was

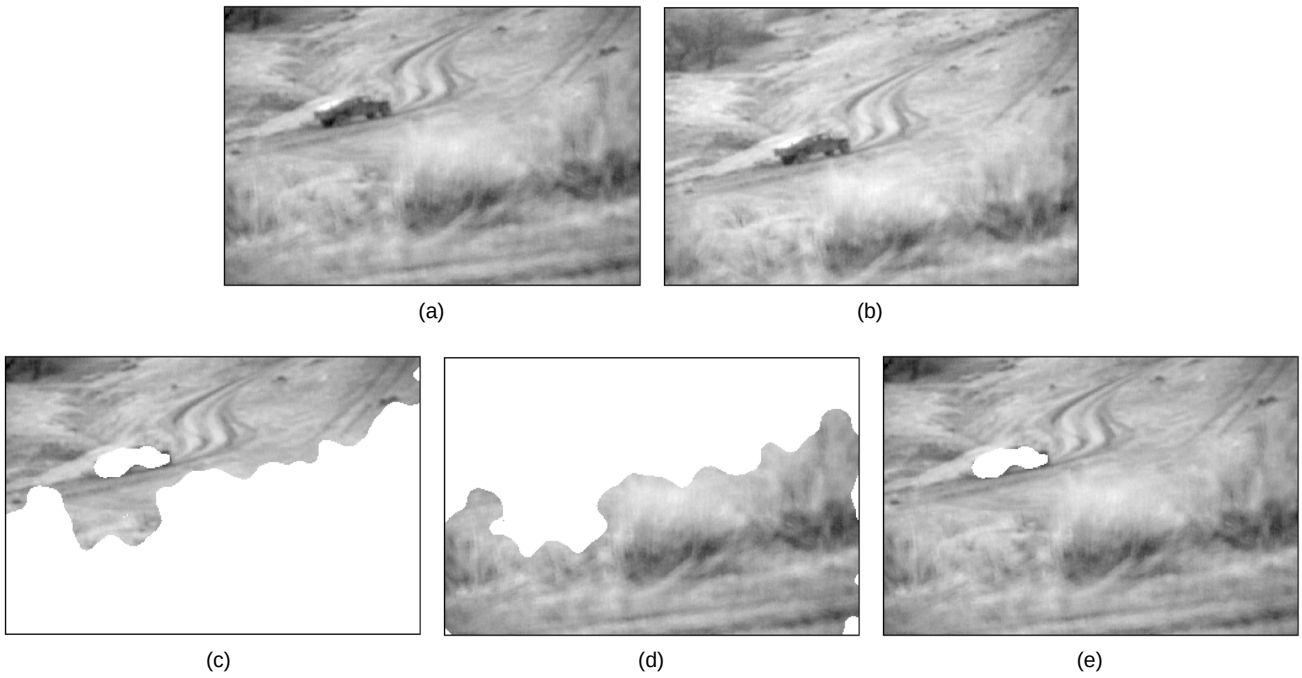


Fig. 2. Layered moving object detection. (a), (b) Two frames in a sequence obtained by a translating and rotating camera. The FOV captures a distant portion of the scene (hills and road) as well as a frontal portion of the scene (bushes). The scene contains a car driving on a road. (c) The image region which corresponds to the *dominant* 2D parametric transformation. This region corresponds to the remote part of the scene. White regions signify image regions which were misaligned after performing global image registration according to the computed dominant 2D parametric transformation. These regions correspond to the car and the frontal part of the scene (the bushes). (d) The image region which corresponds to the *next* detected dominant 2D parametric transformation. This region corresponds to the frontal bushes. The 2D transformation was computed by applying the 2D estimation algorithm again, but this time only to the image regions highlighted in white in Fig. 2c (i.e., only to image regions inconsistent in their image motion with the *first* dominant 2D parametric transformation). White regions in this figure signify regions inconsistent with the bushes' 2D transformation. These correspond to the car and to the remote parts of the scene. (e) The detected moving object (the car) highlighted in white.

not planar, but was distant enough (about 1 km away from the camera) so that 2D parametric transformations were sufficient to account for the camera-induced motion between successive frames. The scene contained a car moving independently on a road. Fig. 1a and Fig. 1b show two frames out of the sequence. Fig. 1c and Fig. 1d show intensity differences before and after dominant (background) 2D alignment, respectively. Fig. 1e shows the detected moving object based on local misalignment analysis [14].

The frame-to-frame motion of the background in remote surveillance applications can typically be modeled by a 2D parametric transformation. However, when a frontal portion of the scene enters the FOV, effects of 3D parallax motion are encountered. The simple 2D algorithm cannot account for camera-induced motion in scenes with 3D parallax. In the next two sections, we address the problem of moving-object detection in 3D scenes *with* parallax.

3 Multiplanar Scenes

When the camera is translating, and the scene is not planar or is not sufficiently distant, then a *single* 2D parametric motion (Section 2) is insufficient for modeling the camera-induced motion. Aligning two images with respect to a *dominant* 2D parametric transformation may bring into alignment a large portion of the scene, which corresponds to a planar (or a remote) part of the scene. However, any other (e.g., near) portions of the scene that enter the FOV cannot be aligned by the dominant 2D parametric transformation. These out-of-plane scene points, although they have the same 3D motion as the planar points, have substantially different induced 2D motions. The *differences* in 2D motions are called *3D parallax motion* [23], [25]. Effects of parallax are only due to camera translation and 3D scene variations. Camera rotation or zoom does not cause parallax (see Section 4.1).

Fig. 2 shows an example of a sequence where the effects of 3D parallax are evident. Figs. 2a and 2b show two frames from a sequence with the same setting and scenario described in Fig. 1, only in this case a frontal hill with bushes (which was much closer to the camera than the background scene) entered the FOV.

Fig. 2c displays the image region which was found to be aligned after *dominant* 2D parametric registration (see Section 2). Clearly the global 2D alignment accounts for the camera-induced motion of the distant portion of the scene, but does *not* account for the camera-induced motion of the closer portion of the scene (the bushes).

Thus, simple 2D techniques, when applied to these types of scenarios, will not be able to distinguish between the independent car motion and the 3D parallax motion of the bush. There is therefore a need to model 3D parallax as well. In this section, we describe one approach to modeling parallax motion, which builds on top of the 2D approach to modeling camera-induced motion. This approach is based on fitting multiple planar surfaces (i.e., multiple 2D "layers" [1], [33]) to the scene. In Section 4, approaches to han-

dling more *complex* types of scenes with (sparse and dense) 3D parallax will be described. They too build on top of the 2D (or layered) approach.

When the scene is piecewise planar, or is constructed of a few distinct portions at different depths, then the camera-induced motion can be accounted for by a few *layers* of 2D parametric transformations. This case is very typical of outdoor surveillance scenarios, especially when the camera FOV is narrow. The multilayered approach is an extension of the simple 2D approach and is implemented using a method similar to the sequential method presented in [14]: First, the dominant 2D parametric transformation between two frames is detected (Section 2). The two images are aligned accordingly, and the misaligned image regions are detected and segmented out (Fig. 2c). Next, the *same* 2D motion estimation technique is reapplied, but this time only to the segmented (misaligned) regions of the image, to detect the *next* dominant 2D transformation and its region of support within the image, and so on. For each additional layer, the two images are aligned according to the 2D parametric transformation of that layer, and the misaligned image regions are detected and segmented out (Fig. 2d).

Each "2D layer" is continuously tracked in time by using the obtained segmentation masks. Moving objects are detected as image regions that are inconsistent with the image motion of any of the 2D layers. Such an example is shown in Fig. 2e.

A moving object is not detected as a layer by this algorithm if it is small. However, if the object is large, it may itself be detected as a 2D layer. A few cues can be used to distinguish between moving objects and static scene layers:

1) Moving objects produce discontinuities in 2D motion everywhere on their boundary, as opposed to static 2D layers. Therefore, if a moving object is detected as a layer, it can be distinguished from real scene layers due to the fact that it appears "floating" in the air (i.e., has depth discontinuities all around it). A real scene layer, on the other hand, is always connected to another part of the scene (layer). On the connecting boundary, the 2D motion is continuous. If the connection to other scene portions is outside the FOV, then that layer is adjacent to the *image* boundary. Therefore, a 2D layer which is fully contained in the FOV, and exhibits 2D motion discontinuities all around it, is necessarily a moving object.
2) The 3D consistency over time of two 2D layers can be checked. In Section 4.2 we present a method for checking 3D consistency of two scene points over time based on their parallax displacements alone. If two layers belong to a single rigid scene, the parallax displacement of one layer with respect to the other is yet another 2D parametric transformation (which is obtained by taking the difference between the two 2D parametric layer transformations). Therefore, for example, consistency of two layers can be verified over time by applying the 3D-consistency check to parallax displacements of one layer with respect to the other (see Section 4.2).
3) Other cues, such as detecting negative depth, can also be used.

In the sequence shown in Figs. 1 and 2, we used the first cue (i.e., eliminated "floating" layers) to ensure moving objects were not interpreted as scene layers. The moving car was successfully and continuously detected over the entire two-minute video sequence, which alternated between the single-layered case (i.e., no 3D parallax; frontal scene part was not visible in the FOV) and the two-layered case (i.e., existence of 3D parallax).

4 Scenes With General 3D Parallax

While the single and multilayered parametric registration methods are adequate to handle a large number of situations, there are cases when the parallax cannot be modeled in terms of layers. An example of such a situation is a cluttered scene which contains many small objects at multiple depths (these could be urban scenes or indoor scenes). In this section, we develop an approach to handling these more complex 3D scenes.

4.1 3D Scenes With Dense Parallax

The key observation that enables us to extend the 2D parametric registration approach to general 3D scenes is the following: the plane registration process (using the dominant 2D parametric transformation) removes all effects of camera rotation, zoom, and calibration, *without explicitly computing them* [15], [18], [26], [27]. The residual image motion after the plane registration is due only to the *translational* motion of the camera and to the *deviations* of the scene structure from the planar surface. Hence, the residual motion is an *epipolar flow field*. This observation has led to the so-called "plane + parallax" approach to 3D scene analysis [17], [15], [18], [26], [27].

4.1.1 The Plane + Parallax Decomposition

Fig. 3 provides a geometric interpretation of the planar parallax. Let $\vec{P} = (X, Y, Z)^T$ and $\vec{P}' = (X', Y', Z')^T$ denote the Cartesian coordinates of a scene point with respect to two different camera views, respectively. Let the 3×3 matrix R and the 3×1 vector T denote the rotation and translation between the two camera systems, respectively.

Let (x, y) and (x', y') denote the image coordinates of the scene point P, and $\vec{p} = (x, y, 1)^T = \frac{1}{Z} K\vec{P}$ and $\vec{p}' = (x', y', 1)^T = \frac{1}{Z'} K'\vec{P}'$ denote the same points in homogeneous coordinates. K and K' are 3×3 matrices representing the internal calibration parameters of the two cameras (see Appendix A). Also, define $\vec{t} = (t_x, t_y, t_z)^T = K\vec{T}$. Note that $(K\vec{P})_z = Z$, $(K'\vec{P}')_z = Z'$, and $t_z = T_Z$. Note that when $T_z \neq 0$, $\vec{e} = \frac{t}{t_z}$ denotes the epipole (or the *focus-of-expansion*, FOE) in homogeneous coordinates.

Let Π be an arbitrary planar surface and A' denote the homography that aligns the planar surface Π between the second and first frame (i.e., for all points $\vec{P} \in S$, $\vec{P} = A'\vec{P}'$).

Define $\vec{u} = \vec{p}' - \vec{p} = (u, v, 0)^T$, where $(u, v)^T$ is the measurable 2D image displacement vector of the image point $\vec{p}$

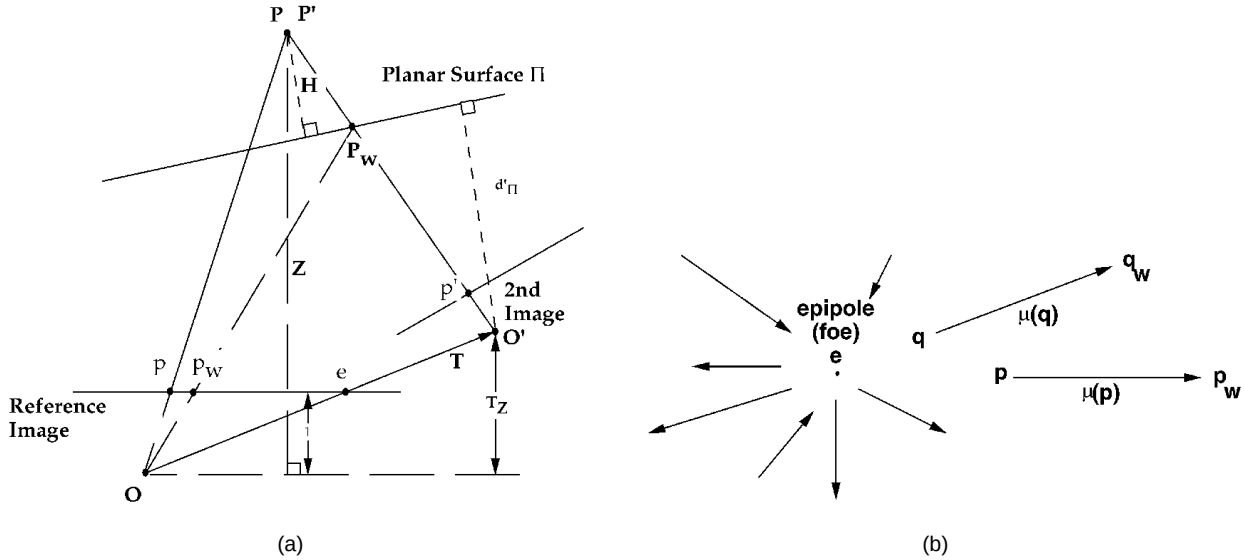


Fig. 3. The plane + parallax decomposition. (a) The geometric interpretation. (b) The epipolar field of the residual parallax displacements.

between the two frames. It can be shown (see Appendix A), as well as [19], [15], [26], [27], that

$$\vec{u} = \vec{u_\pi} + \vec{\mu} \tag{5}$$

where $\vec{\mu_\pi}$ denotes the *planar* part of the 2D image motion (the homography due to Π), and $\vec{\mu}$ denotes the residual *planar parallax* 2D motion. The homography due to Π results in an image motion field that can be modeled as a 2D parametric transformation. In general, this transformation is a *projective transformation*, however, in the case of instantaneous camera motion, it can be well approximated by the quadratic transformation shown in (2).

When $T_z \neq 0$:

$$\vec{\mu_\pi} = \left(\vec{p}' - \vec{p_w}\right); \quad \vec{\mu} = \gamma \frac{T_z}{d'_\pi}\left(\vec{e} - \vec{p_w}\right) \tag{6}$$

where $\vec{p_w}$ denotes the image point (in homogeneous coordinates) in the first frame which results from warping the corresponding point $\vec{p}'$ in the second image, by the 2D parametric transformation of the reference plane Π. We will refer to the first frame as the *reference frame*. Also, d'_π is the perpendicular distance from the second camera center to the reference plane Π (see Fig. 3), and as noted earlier $\vec{e}$ denotes the epipole (or FOE). γ is a measure of the 3D shape of the point $\vec{P}$. In particular, $\gamma = \frac{H}{Z}$, where H is the perpendicular distance from the $\vec{P}$ to the reference plane Π, and Z is the "range" (or "depth") of the point $\vec{P}$ with respect to the first camera. We refer to γ as the projective 3D structure of point $\vec{P}$. In the case when $T_z = 0$, the parallax motion $\vec{\mu}$ has a slightly different form: $\vec{\mu} = \frac{\gamma}{d'_\pi}\vec{t}$, where t is as defined earlier.

The use of the plane + parallax decomposition for egomotion estimation is described in [15], and for 3D shape recovery is described in [18], [26]. The *plane + parallax* decomposition is more general than the traditional decomposition in terms of *rotational* and *translational* motion (and includes the traditional decomposition as a special case). In addition,

1) the planar homography (i.e., the 2D parametric planar transformation) compensates for camera rotation, zoom, and other changes in the internal parameters of the camera,
2) this approach does not require any prior knowledge of the camera internal parameters (in other words, no prior camera calibration is needed), and
3) the planar homography being a 2D parametric transformation can be estimated in a more stable fashion than the rotation and translation parameters. In particular, it can be estimated even when the camera FOV is limited, the depth variations in the scene are small, and in the presence of independently moving objects (see Section 2).

Since the residual parallax displacements after the 2D alignment of the dominant planar surface are due to translational component alone, they form a radial field centered at the epipole / FOE (see Fig. 3b). If the epipole is recovered, all that is required for detecting moving objects is the verification whether the residual 2D displacement associated with a given point is directed towards / away from the epipole. This is known as the *epipolar constraint* [29]. Residual 2D motion that violates this requirement can only be due to an independently moving object. Fig. 4a graphically illustrates this situation. An algorithm for detecting moving objects based on the plane + parallax decomposition is described in [20]. This technique, however, requires the estimation of the 3D shape and the epipole.

4.1.2 Difficulty of Epipole Recovery

While the plane + parallax strategy for moving object detection works generally well when the epipole (FOE) recovery is possible, its performance depends critically on the ability to accurately estimate the epipole. Since the epipole

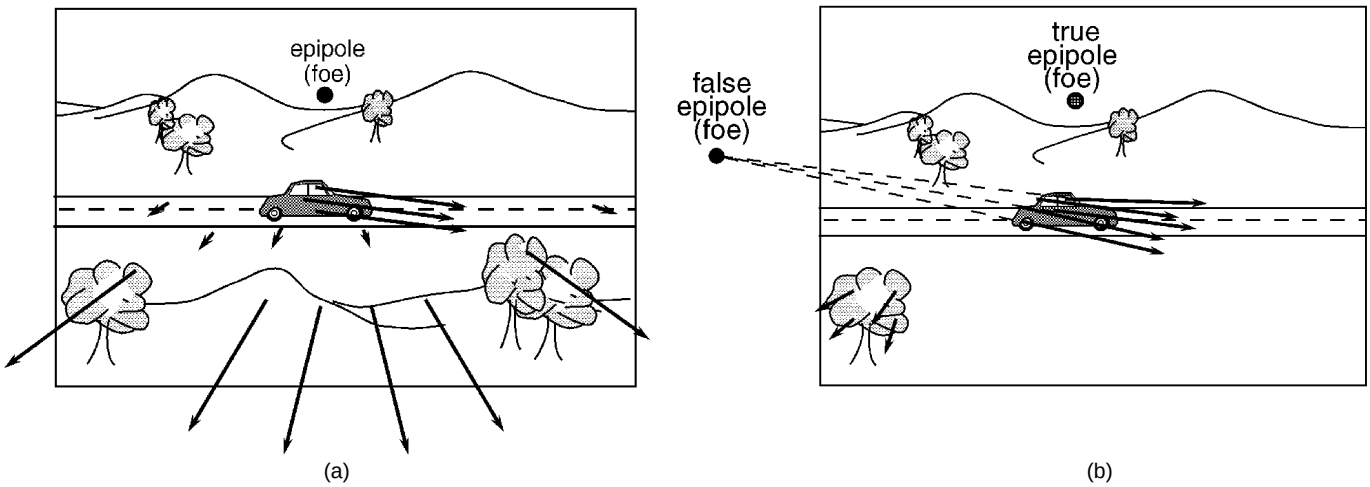


Fig. 4. Moving object detection based on violation of epipolar motion. (a) Moving object detection based on inconsistency of parallax motion with radial epipolar motion field. (b) False epipole estimation when 3D parallax is sparse relative to independent motion.

recovery is based on the residual motion vectors, those vectors that are due to the moving object are likely to bias the estimated epipole away from the true epipole. (Note that this is true even of the "direct" methods that do not explicitly recover the residual motion vectors, but instead rely on spatiotemporal image gradients [18], since the information provided by the points on moving objects will influence the estimate.)

The problem of estimating the epipole is acute when the scene contains sparse parallax information and the residual motion vectors due to independently moving object are significant (either in magnitude or in number). A graphic illustration of such a situation is provided in Fig. 4b. In the situation depicted in this figure, the magnitude and number of parallax vectors on the tree are considerably smaller than the residual motion vectors on the independently moving car. As a result, the estimated epipole is likely to be consistent with the motion of the car (in the figure, this would be somewhere outside the FOV on the left side of the image), and the tree will be detected as an independently moving object.

There are two obvious ways to overcome the difficulties in estimating the epipole. The first is to use prior knowledge regarding the camera/vehicle motion to reject potential outliers (namely, the moving objects) during the estimation. However, if only limited parallax information is available, any attempt to refine this prior information will be unstable. A more general approach would be to defer, or even completely eliminate, the computation of the epipole. In the next section, we develop an approach to moving-object detection by directly comparing the parallax motion of pairs of points *without estimating the epipole*.

4.2 3D Scenes With Sparse Parallax

In this section we present a method we have developed for moving-object detection in the difficult "intermediate" cases, when 3D parallax information is sparse relative to independent motion information. This approach can be used to bridge the gap between the 2D cases and the dense 3D cases.

4.2.1 The Parallax-Based Shape Constraint

THEOREM 1. *Given the planar-parallax displacement vectors* $\vec{\mu}_1$ *and* $\vec{\mu}_2$ *of two points that belong to the static background scene, their* relative 3D projective structure $\frac{\gamma_2}{\gamma_1}$ *is given by*:

$$\frac{\gamma_2}{\gamma_1} = \frac{\vec{\mu}_2^T \left(\Delta \vec{p}_w \right)_\perp}{\vec{\mu}_1^T \left(\Delta \vec{p}_w \right)_\perp}. \tag{7}$$

where, as shown in Fig. 5a, $\vec{p}_1$ *and* $\vec{p}_2$ *are the image locations (in the reference frame) of two points that are part of the static scene,* $\Delta \vec{p}_w = \vec{p}_{w_2} - \vec{p}_{w_1}$, *the vector connecting the "warped" locations of the corresponding second frame points* (*as in* (6)), *and* $\vec{v}_\perp$ *signifies a vector perpendicular to* $\vec{v}$.

PROOF. See Appendix B. □

Note that this constraint directly relates the relative projective structure of two points to their parallax displacements alone: No camera parameters, in particular the *epipole* (FOE), are involved. Neither is any additional parallax information required at other image points. *Theoretically*, one could use the two parallax vectors to recover the epipole (the intersection point of the two vectors) and then use the magnitudes and distances of the points from the computed epipole to estimate their relative projective structure. The benefit of the constraint (7) is that it provides this information *directly* from the positions and parallax vectors of the two points, without the need to go through the computation of the epipole, using as much information as one point can give on another. Fig. 5b graphically shows an example of a configuration in which estimating the epipole is very unreliable, whereas estimating the relative structure *directly* from (7) *is* reliable. Application of this constraint to the recovery of 3D structure of the scene is described in [12]. Here we focus on its application to moving object detection.

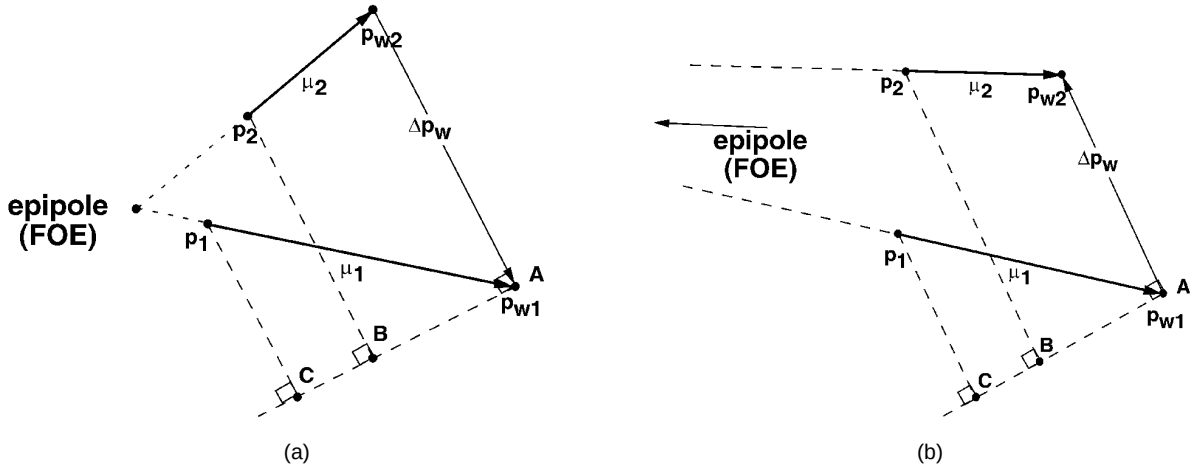


Fig. 5. The pairwise parallax-based shape constraint. (a) This figure geometrically illustrates the relative structure constraint (7): $\frac{\gamma_2}{\gamma_1} = \frac{\vec{\mu}_2^{\,T}(\Delta\vec{p}_w)_\perp}{\vec{\mu}_1^{\,T}(\Delta\vec{p}_w)_\perp} = \frac{\mathbf{AB}}{\mathbf{AC}}$. (b) When the parallax vectors are nearly parallel, the epipole estimation is unreliable. However, the relative structure $\frac{\mathbf{AB}}{\mathbf{AC}}$ can be reliably computed even in this case.

4.2.2 The Parallax-Based Rigidity Constraint

THEOREM 2. *Given the planar-parallax displacement vectors of two points that belong to the background static scene over* three *frames, the following constraint must be satisfied*:

$$\frac{\vec{\mu}_2^{\,j\,T}\left(\Delta\vec{p}_w\right)^j_\perp}{\vec{\mu}_1^{\,j\,T}\left(\Delta\vec{p}_w\right)^j_\perp} - \frac{\vec{\mu}_2^{\,k\,T}\left(\Delta\vec{p}_w\right)^k_\perp}{\vec{\mu}_1^{\,k\,T}\left(\Delta\vec{p}_w\right)^k_\perp} = 0. \tag{8}$$

where $\vec{\mu}_1^{\,j}$, $\vec{\mu}_2^{\,j}$ *are the parallax displacement vectors of the two points between the reference frame and the jth frame,* $\vec{\mu}_1^{\,k}$, $\vec{\mu}_2^{\,k}$ *are the parallax vectors between the reference frame and the kth frame, and* $\left(\Delta\vec{p}_w\right)^j$, $\left(\Delta\vec{p}_w\right)^k$ *are the corresponding distances between the warped points as in* (7) *and* Fig. 5a.

PROOF. The relative projective structure $\frac{\gamma_2}{\gamma_1}$ is invariant to camera motion. Therefore, using (7), for any two frames j and k we get:

$$\frac{\gamma_2}{\gamma_1} = \frac{\vec{\mu}_2^{\,j\,T}\left(\Delta\vec{p}_w\right)^j_\perp}{\vec{\mu}_1^{\,j\,T}\left(\Delta\vec{p}_w\right)^j_\perp} = \frac{\vec{\mu}_2^{\,k\,T}\left(\Delta\vec{p}_w\right)^k_\perp}{\vec{\mu}_1^{\,k\,T}\left(\Delta\vec{p}_w\right)^k_\perp}.$$

□

As in the case of the parallax-based shape constraint (7), the parallax-based rigidity constraint (8) relates the parallax vectors of pairs of points over three frames without referring to the *camera geometry* (especially the epipole/FOE). Furthermore, this constraint does not even explicitly refer to the *structure* parameters of the points in consideration. The rigidity constraint (8) can therefore be applied to detect inconsistencies in the 3D motion of two image points (i.e., say whether the two image points are projections of 3D points belonging to the same or different 3D moving objects) based on their *parallax* motion among three (or more) frames alone, without the need to estimate either *camera geometry*, *camera motion*, or *structure* parameters, and without relying on parallax information at other image points. A consistency measure is defined as the left-hand side of (8), after multiplying by the denominators (to eliminate singularities). The farther this quantity is from zero, the higher is the 3D-inconsistency of the two points.

4.3 Applying the Parallax Rigidity Constraint to Moving Object Detection

Fig. 6a graphically displays an example of a configuration in which estimating the epipole in presence of multiple moving objects can be very erroneous, even when using clustering techniques in the epipole domain as suggested by [21], [30]. Relying on the epipole computation to detect inconsistencies in 3D motion fails in detecting moving objects in such cases.

The parallax rigidity constraint (8) can be applied to detect inconsistencies in the 3D motion of one image point relative to another directly from their "parallax" vectors over multiple (three or more) frames, without the need to estimate either *camera geometry*, *camera motion*, or *shape* parameters. This provides a useful mechanism for clustering (or segmenting) the "parallax" vectors (i.e., the residual motion after planar registration) into consistent groups belonging to consistently 3D moving objects, even in cases such as in Fig. 6a, where the parallax information is minimal, and the independent motion is not negligible. Fig. 6b graphically explains how the rigidity constraint (8) detects the 3D inconsistency of Fig. 6a over three frames.

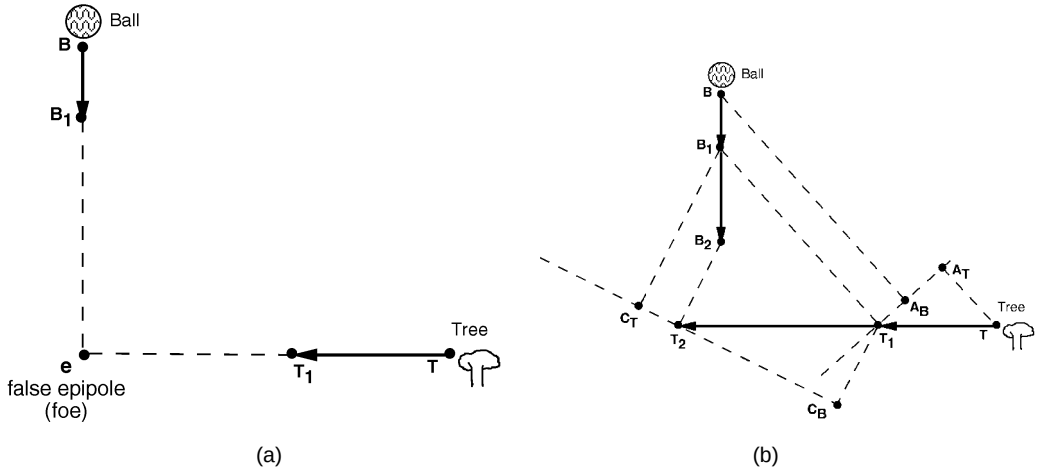


Fig. 6. Reliable detection of 3D motion inconsistency with sparse parallax information. (a) Camera is translating to the right. The only static object with pure parallax motion is that of the tree. Ball is falling independently. The epipole may incorrectly be computed as e. The false epipole e is consistent with both motions. (b) The rigidity constraint applied to this scenario detects 3D inconsistency over three frames, since $\frac{T_1A_B}{T_1A_T} \neq \frac{T_2C_B}{-T_2C_T}$. In this case, even the signs do not match.

Fig. 7 shows an example of using the rigidity-based inconsistency measure described earlier to detect 3D inconsistencies. In this sequence the camera is in motion (translating from left to right), inducing parallax motion of different magnitudes on the house, road, and road sign. The car moves independently from left to right. The detected 2D planar motion was that of the house. The planar parallax motion was computed after 2D registration of the three images with respect to the house (see Fig. 7d). A single point on the road sign was selected as a point of reference (see Fig. 7e). Fig. 7f displays the measure of *in*consistency of each point in the image with respect to the selected road sign point. Bright regions indicate large values when applying the inconsistency measure, i.e., violations in 3D rigidity detected over three frames with respect to the road sign point. The region which was detected as moving 3D-inconsistently with respect to the road sign point corresponds to the car. Regions close to the image boundary were ignored. All other regions of the image were detected as moving 3D-consistently with the road sign point. Therefore, assuming an *uncalibrated* camera, this method provides a mechanism for segmenting all nonzero residual motion vectors (after 2D planar stabilization) into groups moving *consistently* (in the 3D sense).

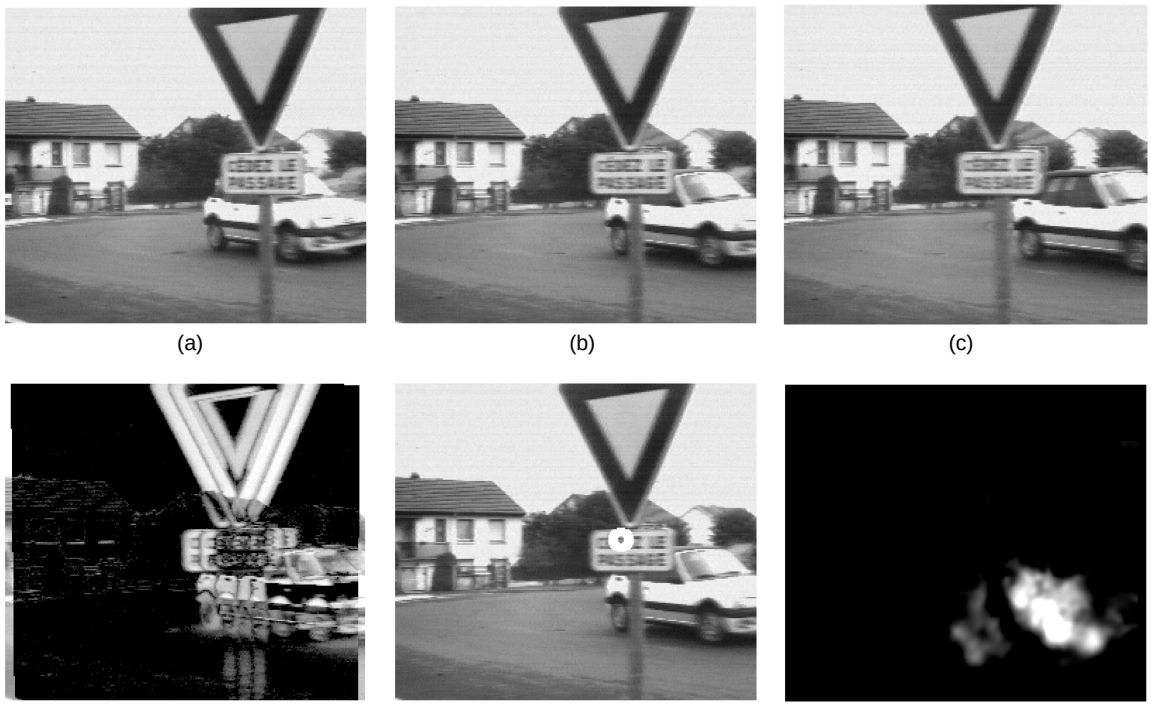
Fig. 8 shows another example of using the rigidity constraint (8) to detect 3D inconsistencies. In this sequence the camera is mounted on a helicopter flying from left to right, inducing some parallax motion (of different magnitudes) on the house roof and trees (bottom of the image) and on the electricity poles (by the road). Three cars move independently on the road. The detected 2D planar motion was that of the ground surface (see Fig. 8d). A single point was selected on a tree as a point of reference (see Fig. 8e). Fig. 8f displays the measure of *in*consistency of each point in the image with respect to the selected reference point. Bright regions indicate 3D-inconsistency detected over three frames. The three cars were detected as moving *in*consistently with the selected tree point. Regions close to the image boundary were ignored. All other image regions were detected as moving consistently with the selected tree point.

The ability of the parallax rigidity constraint (8) to detect 3D-inconsistency with respect to a *single* point provides a natural way to *bridge* between 2D algorithms (which assume that any 2D motion different than the planar motion is an independently moving object), and 3D algorithms (which rely on having prior knowledge of a consistent set of points or, alternatively, dense parallax data).

5 Conclusion

Previous approaches to the problem of moving-object detection can be broadly divided into two classes: 2D algorithms which apply when the scene can be approximated by a flat surface and/or when the camera is only undergoing rotations and zooms, and 3D algorithms which work well only when significant depth variations are present in the scene and the camera is translating. These two classes of algorithms treat two extremes in a continuum of scenarios: *no 3D parallax* (2D algorithms) vs. *dense 3D parallax* (3D algorithms). Both classes fail on the other extreme case or even on the intermediate case (when 3D parallax is *sparse* relative to amount of independent motion).

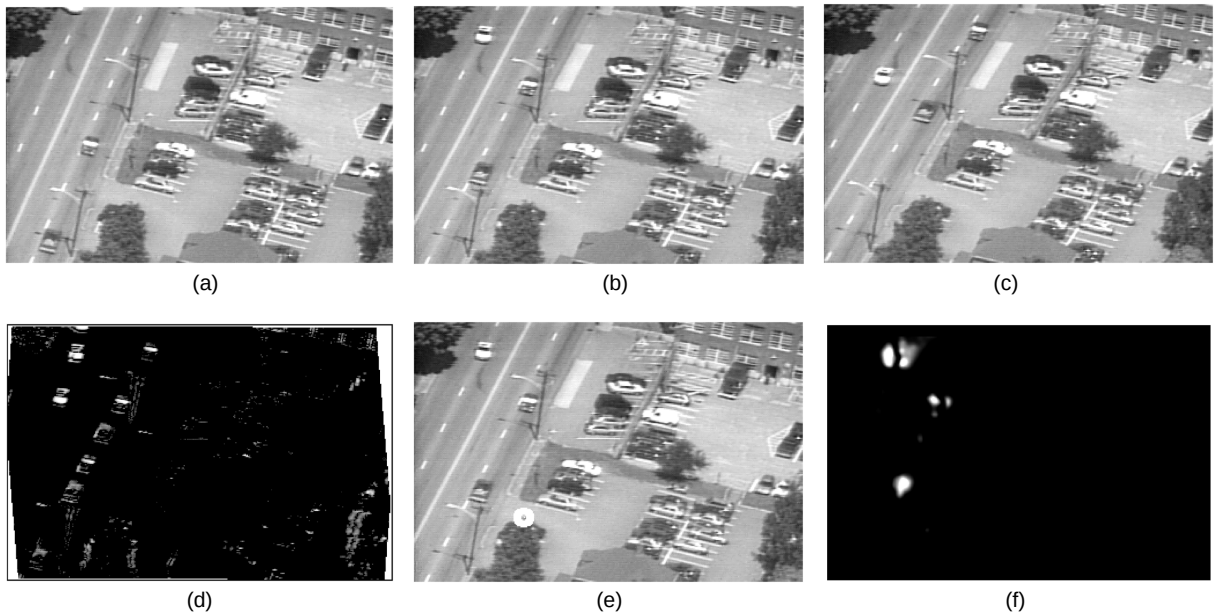
In this paper, we have described a unified approach to handling moving-object detection in both 2D and 3D scenes, with a strategy to gracefully bridge the gap between those two extremes. Our approach is based on a stratification of the moving object-detection problem into scenarios which gradually increase in their complexity. We presented a set of techniques that match the above stratification. These techniques progressively increase in their complexity, ranging from 2D techniques to more complex 3D techniques. Moreover, the computations required for the solution to the problem at one complexity level become the initial processing step for the solution at the next complexity level.



(a) (b) (c)

(d) (e) (f)

Fig. 7. Moving object detection relying on a single parallax vector. (a), (b), (c) Three image frames from a sequence obtained by a camera translating from left to right, inducing parallax motion of different magnitudes on the house, road, and road sign. The car moves independently from left to right. The middle frame (Fig. 7b) was chosen as the frame of reference. (d) Differences taken after 2D image registration. The detected 2D planar motion was that of the house, and is canceled by the 2D registration. All other scene parts that have different 2D motions (i.e., parallax motion or independent motion) are misregistered. (e) The selected point of reference (a point on the road sign) highlighted by a white circle. (f) The measure of 3D-inconsistency of all points in the image with respect to the road sign point. Bright regions indicate violations in 3D rigidity detected over three frames with respect to the selected road sign point. These regions correspond to the car. Regions close to the image boundary were ignored. All other regions of the image appear to move 3D-consistently with the road sign point.



(a) (b) (c)

(d) (e) (f)

Fig. 8. Moving object detection relying on a single parallax vector. (a), (b), (c) Three image frames from a sequence obtained by a camera mounted on a helicopter (flying from left to right while turning), inducing some parallax motion (of different magnitudes) on the house roof and trees (bottom of the image) and on the electricity poles (by the road). Three cars move independently on the road. The middle frame (Fig. 8b) was chosen as the frame of reference. (d) Differences taken after 2D image registration. The detected 2D planar motion was that of the ground surface and is canceled by the 2D registration. All other scene parts that have different 2D motions (i.e., parallax motion or independent motion) are misregistered. (e) The selected point of reference (a point on a tree at the bottom left of the image) highlighted by a white circle. (f) The measure of 3D-inconsistency of each point in the image with the tree point. Bright regions indicate violations in 3D rigidity detected over three frames with respect to the selected tree point. These regions correspond to the three cars (in the reference image). Regions close to the image boundary were ignored. All other regions of the image appear to move 3D-consistently with the tree point.

The goal in taking this approach is to develop a strategy for moving object detection, so that the analysis performed is tuned to match the complexity of the problem and the availability of information at any time. This paper describes the core elements of such a strategy. The integration of these elements into a single algorithm remains a task for our future research.

APPENDIX A
DERIVATION OF THE PLANE + PARALLAX DECOMPOSITION

In this appendix, we rederive the decomposition of image motion into the image motion of a planar surface (a homography) and residual parallax displacements.

Let $\vec{P} = (X, Y, Z)^T$ and $\vec{P}' = (X', Y', Z')^T$ denote the Cartesian coordinates of a scene point with respect to two different camera views, respectively. An arbitrary 3D rigid coordinate transformation between $\vec{P}$ and $\vec{P}'$ can be expressed by:

$$\vec{P}' = R\vec{P} + \vec{T}', \tag{9}$$

where R represents the rotation between the two camera coordinate systems, $\vec{T}' = (T'_X, T'_Y, T'_Z)$ denotes the 3D translation in between the two views as expressed in the coordinate system of the second camera, and $\vec{T} = (T_X, T_Y, T_Z) = -R^{-1}\vec{T}'$ denotes the same quantity in the coordinate system of the first camera.

Let Π denote an arbitrary 3D planar surface (real or virtual). Let $\vec{N}$ denote its normal as expressed in the coordinate system of the first camera, and $\vec{N}'$ denote the same quantity in the coordinate system of the second camera. Any point $\vec{P} \in \prod$ satisfies the equation $\vec{N}^T\vec{P} = d_\pi$ (and similarly $\vec{N}'^T\vec{P}' = d'_\pi$). For a general scene point $\vec{P}$:

$$\begin{aligned} \vec{N}^T\vec{P} &= d_\pi + H \\ \vec{N}'^T\vec{P}' &= d'_\pi + H \end{aligned} \tag{10}$$

where H denotes the perpendicular distance of $\vec{P}$ from the plane Π. Note that H is invariant with respect to the two camera coordinate systems (see Fig. 3).

By inverting (9), we obtain

$$\begin{aligned} \vec{P} &= R^{-1}\vec{P}' - R^{-1}\vec{T}' \\ &= R^{-1}\vec{P}' + \vec{T} \end{aligned} \tag{11}$$

From (10), we derive

$$\frac{\vec{N}'^T\vec{P}' - H}{d'_\pi} = 1 \tag{12}$$

Substituting this in (11) obtains

$$\vec{P} = R^{-1}\vec{P}' + \vec{T}\frac{\left(\vec{N}'^T\vec{P}' - H\right)}{d'_\pi} \tag{13}$$

$$= \left(R^{-1} + \frac{\vec{T}\vec{N}'^T}{d'_\pi}\right)\vec{P}' - \frac{H}{d'_\pi}\vec{T}. \tag{14}$$

Let $\vec{p} = (x, y, 1)^T = \frac{1}{Z}K\vec{P}$ and $\vec{p}' = (x', y', 1)^T = \frac{1}{Z'}K'\vec{P}'$ denote the images of the scene point P in the two camera views as expressed in homogeneous coordinates. K and K' are 3×3 matrices representing the internal calibration parameters of the two cameras. In general K has the following form [11]:

$$K = \begin{bmatrix} a & b & c \\ 0 & d & e \\ 0 & 0 & 1 \end{bmatrix}.$$

Also, define $\vec{t} = \left(t_x, t_y, t_z\right)^T = K\vec{T}$. (Note that $\left(K\vec{P}\right)_z = Z$, $\left(K'\vec{P}'\right)_z = Z'$, and $t_z = T_Z$.) Multiplying both sides of (14) by $\frac{1}{Z'}K$ gives:

$$\frac{Z}{Z'}\vec{p} = K\left(R^{-1} + \frac{\vec{T}\vec{N}'^T}{d'_\pi}\right)K'^{-1}\vec{p}' - \frac{H}{d'_\pi Z'}\vec{t} \tag{15}$$

Hence,

$$\vec{p} \cong A'\vec{p}' - \frac{H}{d'_\pi Z'}\vec{t}, \tag{16}$$

where $\cong$ denotes equality up to an arbitrary scale. $A' = K\left(R^{-1} + \frac{\vec{T}\vec{N}'^T}{d'_\pi}\right)K'^{-1}$ is a 3×3 matrix which represents the coordinate transformation of the planar surface Π between the two camera views, i.e., the homography between the two views due to the plane Π. Scaling both sides by their third component (i.e., projection) gives the equality:

$$\vec{p} = \frac{A'\vec{p}' - \frac{H}{d'_\pi Z'}\vec{t}}{a'_3\vec{p}' - \frac{HT_z}{d'_\pi Z'}} \tag{17}$$

$$= \frac{A'\vec{p}'}{a'_3\vec{p}'} - \frac{A'\vec{p}'}{a'_3\vec{p}'} + \frac{A'\vec{p}' - \frac{H}{d'_\pi Z'}\vec{t}}{a'_3\vec{p}' - \frac{HT_z}{d'_\pi Z'}} \tag{18}$$

$$= \frac{A'\vec{p}'}{a'_3\vec{p}'} + \frac{\frac{HT_z}{d'_\pi Z'}}{\left(a'_3\vec{p}' - \frac{HT_z}{d'_\pi Z'}\right)}\frac{A'\vec{p}'}{a'_3\vec{p}'} - \frac{\frac{H}{d'_\pi Z'}}{a'_3\vec{p}' - \frac{HT_z}{d'_\pi Z'}}\vec{t} \tag{19}$$

where a'_3 denotes the third row of the matrix A'. Moreover by considering the third component of the vector (15), we obtain

$$\frac{Z}{Z'} = a'_3\vec{p}' - \frac{HT_z}{d'_\pi Z'}. \tag{20}$$

Substituting this into (19), we obtain

$$\vec{p} = \frac{A'\vec{p}'}{a'_3\vec{p}'} + \frac{H}{Z}\frac{T_z}{d'_\pi}\frac{A'\vec{p}'}{a'_3\vec{p}'} - \frac{H}{Zd'_\pi}\vec{t} \tag{21}$$

When $T_z \neq 0$, let $\vec{e} = \frac{1}{T_z}\vec{t}$ denote the epipole in the first image. Then,

$$\vec{p} = \frac{A'\vec{p}'}{a'_3\vec{p}'} + \frac{H}{Z}\frac{T_z}{d'_\pi}\left(\frac{A'\vec{p}'}{a'_3\vec{p}'} - e\right) \tag{22}$$

On the other hand, when $T_z = 0$, we obtain

$$\vec{p} = \frac{A'\vec{p}'}{a'_3\vec{p}'} - \frac{H}{Zd'_\pi}\vec{t}. \tag{23}$$

The point denoted by the vector $\frac{A'\vec{p}'}{a'_3\vec{p}'}$ is of special interest, since it represents the location to which the point $\vec{p}'$ is transformed due to the homography A'. In Fig. 3, this is denoted as the point $\vec{p_w}$. Also, we define $\gamma = \frac{H}{Z}$, which is the *3D projective structure* (γ) of $\vec{P}$ with respect to the planar surface Π. Substituting these into (22) and (23) yields:
when $T_z \neq 0$:

$$\vec{p} = \vec{p_w} + \gamma \frac{T_z}{d'_\pi}\left(\vec{p_w} - \vec{e}\right) \tag{24}$$

and when $T_z = 0$:

$$\vec{p} = \vec{p_w} - \frac{\gamma}{d'_\pi}\vec{t} \tag{25}$$

Rewriting (24) in the form of image *displacements* yields (in homogeneous coordinates):

$$\vec{p}' - \vec{p} = \left(\vec{p}' - \vec{p_w}\right) - \gamma \frac{T_Z}{d'_\pi}\left(\vec{p_w} - \vec{e}\right). \tag{26}$$

Define $\vec{u} = \vec{p}' - \vec{p} = \left(u, v, 0\right)^T$, where $(u, v)^T$ is the measurable 2D image displacement vector of the image point $\vec{p}$ between the two frames. Similarly, define $\vec{u_\pi} = \vec{p}' - \vec{p_w} = \left(u_\pi, v_\pi, 0\right)^T$ and $\vec{\mu} = -\gamma \frac{T_Z}{d'_\pi}\left(\vec{p_w} - \vec{e}\right) = \left(\mu_x, \mu_y, 0\right)^T$. Hence,

$$\vec{u} = \vec{u_\pi} + \vec{\mu}, \tag{27}$$

$\vec{u_\pi}$ denotes the planar part of the 2D image displacement (i.e., the homography due to Π), and $\vec{\mu}$ denotes the residual *parallax* 2D displacement.

When $T_z = 0$ then, from Eq. (25): $\vec{\mu} = \frac{\gamma}{d'_\pi}\vec{t}$.

APPENDIX B
THE PARALLAX-BASED SHAPE CONSTRAINT

In this appendix, we prove Theorem 1, i.e., we derive (7).

Let $\vec{\mu_1}$ and $\vec{\mu_2}$ be the planar-parallax displacement vectors of two points that belong to the static background. From (6), we know that

$$\vec{\mu_1} = \gamma_1 \frac{T_z}{d'_\pi}\left(\vec{e} - \vec{p}_{w_1}\right) \;;\; \vec{\mu_2} = \gamma_2 \frac{T_z}{d'_\pi}\left(\vec{e} - \vec{p}_{w_2}\right). \tag{28}$$

Therefore,

$$\vec{\mu_1}\gamma_2 - \vec{\mu_2}\gamma_1 = \gamma_1\gamma_2 \frac{T_Z}{d'_\pi}\left(\vec{p_{w_2}} - \vec{p_{w_1}}\right). \tag{29}$$

This last step eliminated the epipole $\vec{e}$. Equation (29) entails that the vectors on both sides of the equation are parallel. Since $\gamma_1\gamma_2 \frac{T_Z}{d'_\pi}$ is a scalar, we get: $\left(\vec{\mu_1}\gamma_2 - \vec{\mu_2}\gamma_1\right) \parallel \Delta\vec{p}_w$, where $\Delta\vec{p}_w = \left(\vec{p_{w_2}} - \vec{p_{w_1}}\right)$. This leads to the *pairwise parallax constraint*

$$\left(\vec{\mu_1}\gamma_2 - \vec{\mu_2}\gamma_1\right)^T \left(\Delta\vec{p}_w\right)_\perp = 0, \tag{30}$$

where $\vec{v}_\perp$ signifies a vector perpendicular to $\vec{v}$. When $T_Z = 0$, a constraint stronger than (30) can be derived: $\left(\vec{\mu_1}\gamma_2 - \vec{\mu_2}\gamma_1\right) = 0$, however, (30), still holds. This is important, as we do not have a priori knowledge of T_Z to distinguish between the two cases.

From (30), we can easily derive:

$$\frac{\gamma_2}{\gamma_1} = \frac{\vec{\mu_2}^T \left(\Delta\vec{p}_w\right)_\perp}{\vec{\mu_1}^T \left(\Delta\vec{p}_w\right)_\perp},$$

which is the same as (7) of Theorem 1.

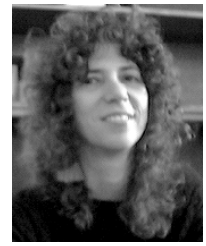
ACKNOWLEDGMENTS

This work was done while the authors were at Sarnoff Corporation, Princeton, N.J. It was supported in part by DARPA under contract DAAA15-93-C-0061.

REFERENCES

[1] E.H. Adelson, "Layered Representations for Image Coding," Technical Report 181, MIT Media Lab, Vision and Modeling Group, Dec. 1991.
[2] G. Adiv, "Determining Three-Dimensional Motion and Structure From Optical Flow Generated by Several Moving Objects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 7, no. 4, pp. 384–401, July 1985.
[3] G. Adiv, "Inherent Ambiguities in Recovering 3D Motion and Structure From a Noisy Flow Field," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, pp. 477-489, May 1989.
[4] Y. Aloimonos, ed. *Active Perception*. Erlbaum, 1993.
[5] S. Ayer and H. Sawhney, "Layered Representation of Motion Video Using Robust Maximum-Likelihood Estimation of Mixture Models and MDL Encoding," *Int'l Conf. Computer Vision*, pp. 777–784, Cambridge, Mass., June 1995.
[6] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani, "Hierarchical Model-Based Motion Estimation," *European Conf. Computer Vision*, pp. 237-252, Santa Margarita Ligure, May 1992.
[7] J.R. Bergen, P.J. Burt, R. Hingorani, and S. Peleg, "A Three Frame Algorithm for Estimating Two-Component Image Motion," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, pp. 886-896, Sept. 1992.
[8] P.J. Burt, R. Hingorani, and R.J. Kolczynski, "Mechanisms for Isolating Component Patterns in the Sequential Analysis of Multiple Motion," *IEEE Workshop Visual Motion*, pp. 187-193, Princeton, N.J., Oct. 1991.
[9] J. Costeira and T. Kanade. "A Multi-Body Factorization Method for Motion Analysis," *Int'l Conf. Computer Vision*, pp. 1,071-1,076, Cambridge, Mass., June 1995.
[10] T. Darrell and A. Pentland, "Robust Estimation of a Multi-Layered Motion Representation," *IEEE Workshop Visual Motion*, pp. 173-178, Princeton, N.J., Oct. 1991.
[11] O. Faugeras, *Three-Dimensional Computer Vision*. Cambridge, Mass.: M.I.T. Press, 1993.
[12] M. Irani and P. Anandan, "Parallax Geometry of Pairs of Points for 3D Scene Analysis," *European Conf. Computer Vision*, Cambridge, UK, Apr. 1996.
[13] M. Irani and P. Anandan, "A Unified Approach to Moving Object Detection in 2D and 3D Scenes," *13th Int'l Conf. Pattern Recognition*, pp. 712–717, Vienna, Austria, Aug. 1996.

[14] M. Irani, B. Rousso, and S. Peleg, "Computing Occluding and Transparent Motions," *Int'l J. Computer Vision*, vol. 12, pp. 5-16, Feb. 1994.
[15] M. Irani, B. Rousso, and S. Peleg, "Recovery of Egomotion Using Region Alignment," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 268-272, Mar. 1997.
[16] S. Ju, M.J. Black, and A.D. Jepson, "Multi-Layer, Locally Affine Optical Flow and Regularization With Transparency," *Proc. of IEEE CVPR96*, pp. 307-314, 1996.
[17] J.J. Koenderink and A.J. van Doorn, "Representation of Local Geometry in the Visual System," *Biol. Cybern.*, vol. 55, pp. 367–375, 1987.
[18] R. Kumar, P. Anandan, and K. Hanna, "Direct Recovery of Shape From Multiple Views: A Parallax Based Approach," *Proc 12th ICPR*, 1994.
[19] R. Kumar, P. Anandan, and K. Hanna, "Shape Recovery From Multiple Views: A Parallax Based Approach," *DARPA IU Workshop*, Monterey, Calif., Nov. 1994.
[20] R. Kumar, P. Anandan, M. Irani, J.R. Bergen, and K.J. Hanna, "Representation of Scenes From Collections of Images," *Workshop on Representations of Visual Scenes*, 1995.
[21] J.M. Lawn and R. Cipolla, "Robust Egomotion Estimation From Affine Motion Parallax," *European Conf. Computer Vision*, pp. 205-210, May 1994.
[22] H.C. Longuet-Higgins, "Visual Ambiguity of a Moving Plane," *Proc. Royal Soc. London, Series B*, vol. 223, pp. 165-175, 1984.
[23] H.C. Longuet-Higgins and K. Prazdny, "The Interpretation of a Moving Retinal Image," *Proc. Royal Soc. London, Series B*, vol. 208, pp. 385-397, 1980.
[24] F. Meyer and P. Bouthemy, "Region-Based Tracking in Image Sequences," *European Conf. Computer Vision*, pp. 476-484, Santa Margarita Ligure, May 1992.
[25] J.H. Rieger and D.T. Lawton, "Processing Differential Image Motion," *J. Optical Soc. Am. A*, vol. A2, no. 2, pp. 354-359, 1985.
[26] H. Sawhney, "3D Geometry From Planar Parallax," *IEEE Conf. Computer Vision and Pattern Recognition*, June 1994.
[27] A. Shashua and N. Navab, "Relative Affine Structure: Theory and Application to 3D Reconstruction From Perspective Views," *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 483-489, Seattle, Wash., June 1994.
[28] M. Shizawa and K. Mase, "Principle of Superposition: A Common Computational Framework for Analysis of Multiple Motion," *IEEE Workshop Visual Motion*, pp. 164-172, Princeton, N.J., Oct. 1991.
[29] W.B. Thompson and T.C. Pong, "Detecting Moving Objects," *Int'l J. Computer Vision*, vol. 4, pp. 29-57, 1990.
[30] P.H.S. Torr and D.W. Murray, "Stochastic Motion Clustering," *European Conf. Computer Vision*, pp. 328-337, May 1994.
[31] P.H.S. Torr, A. Zisserman, and S.J. Maybank, "Robust Detection of Degenerate Configurations for the Fundamental Matrix," *Int'l Conf. Computer Vision*, pp. 1,037-1,042, Cambridge, Mass., June 1995.
[32] B.C. Vemuri, S. Huang, S. Sahni, "A Robust and Efficient Algorithm for Image Registration," *Proc. of 15th Int'l Conf. Information Processing in Medical Imaging*, Poultney, VT, pp. 465-470, 1997.
[33] J. Wang and E. Adelson, "Layered Representation for Motion Analysis," *IEEE Conference Computer Vision and Pattern Recognition*, pp. 361-366, New York, June 1993.



Michal Irani received the BSc degree in mathematics and computer science from the Hebrew University of Jerusalem, Israel, in 1985; the MSc and PhD in computer science from the Hebrew University of Jerusalem in 1989 and 1994, respectively. During 1993-1996, she was a member of the technical staff in the Vision Technologies Laboratory at David Sarnoff Research Center (SRI), Princeton, N.J. Dr. Irani is now a member of the faculty of the Applied Math and Computer Science department at the Weizmann Institute of Science, Israel.

P. Anandan obtained his PhD in computer science from the University of Massachusetts, Amherst, Mass., in 1987. During 1987-1991, he was an assistant professor of computer science at Yale University, New Haven, Conn., and during 1990-1997 he was at the David Sarnoff Research Center, Princeton, N.J. He is currently a senior researcher at Microsoft Corporation, Redmond, Wash. His research interests include computer vision with an emphasis on motion analysis and video processing, vision for robotics, and learning.