Trackability as a Cue for Potential Obstacle Identification and 3D Description

Harpreet Sawhney
Allen Hanson

COINS TR92-15

February 1992

This work has been supported in part by the Defense Advanced Research Projects Agency under Contract Number DACA76-89-C-0017, and by the National Science Foundation under Grant Number CDA-8922572.

TRACKABILITY AS A CUE FOR POTENTIAL OBSTACLE IDENTIFICATION AND 3D DESCRIPTION[1]

Harpreet S. Sawhney
Allen R. Hanson

Department of Computer Science
University of Massachusetts
Amherst, MA 01003
Phone : (413)545-1519
NetAd : sawhney@cs.umass.EDU
February 17, 1992

[1]This work was supported in part by the NSF under grant number DCR-8500332 and by DARPA under contract number ARMY ETL DACA76-89-C-0017.

Abstract

In many man–made environments, obstacles in the path of a mobile robot can be characterized as *shallow*, i.e. they have relatively small extent in depth compared to the distance from the camera. We present a framework for segmenting shallow structures from their background over a sequence of images. Shallowness is first quantified as *affine describability*. This is embedded in a tracking system within which hypothesized model structures undergo a cycle of prediction and model–matching. Structures emerge either as shallow or non–shallow based on their *affine trackability*. This work rejects continuity heuristics for purely image motion in favor of temporal continuity defined as the consistency of generic 3D models, namely shallow structures. Further, shallow structures are reconstructed in 3D using the change of scale of the aggregate structure over time. Temporal integration of many 3D estimates is shown to produce promising results on real sequences.

1 Introduction

Detecting obstacles in the path of a mobile robot is an important, and yet generally unsolved, problem in the area of autonomous visual navigation. Autonomous navigation would be greatly benefited if 3D representations of surfaces could be derived using vision. This work deals with 3D reconstruction from monocular vision. Much of the work in recovering scene structure from monocular vision has concentrated on deriving depths of points, lines or pixels but has, unfortunately, achieved only limited success. Both the motion and structure computations suffer from inherent ambiguities [2] in many realistic scenarios and also are very sensitive to noise in correspondences or flow extraction [22]. The recovery of aggregate 3D structures is generally left to some later stage in which features are grouped into surface patches. In this paper, we demonstrate that useful quantitative inferences about the scene structure can be derived if descriptions are based on generic assumptions about the world over and above rigidity of motion. The motion could be due either to camera motion and/or to object motion.

A major line of research has been to track lines or points over two or more frames, followed by the application of a structure from motion technique to the resulting correspondences [11, 13, 23]. The tracking of image tokens over time has been largely based on heuristics about the motion of these tokens in the image plane. For instance, locally constant acceleration in time and similarity measures over image tokens have been employed [8, 9]. As the 3D structure of the scene and the motion of the camera are both confounded in the motion of the image tokens, heuristics about image motion can easily break down. The approach described here employs generic assumptions about the 3D motion and structure to compute descriptions of aggregate structures in the imaged scene. This is a significant difference from the tracking algorithms developed by Crowley et al. [8] and Deriche and Faugeras [9] who employ only partially valid heuristics involving 2D motion. In our work, generic model–matching, common fate grouping and prediction–based tracking are integrated in a single dynamic framework for describing scene structure over time.

Furthermore, an advantage of the approach described here is that 3D structure information is derived reliably without the intermediate step of explicit computation of the 3D motion parameters. The well–known inherent ambiguities ([2, 23]) in the process of decomposing the image motion into a 3D rotation and a translation can lead to large errors in the 3D structure estimation.

The goal here is to discover aggregate structures in the imaged scene which can be characterized as *shallow structures*. Shallow structures are 3D structures with the property that the difference in depth within the whole structure is small compared to its distance from the camera. Figure 1 shows an image of a hallway. This scene consists of compact structures like the cones and the trash can, and extended structures like the walls, the floor and the ceiling. When viewed from distances at which it might be desirable for a mobile robot to represent these internally, the variation in depth within these structures is small compared to their average distances from the camera. That is, the structures can be characterized as shallow at distances where the path planner for the robot might

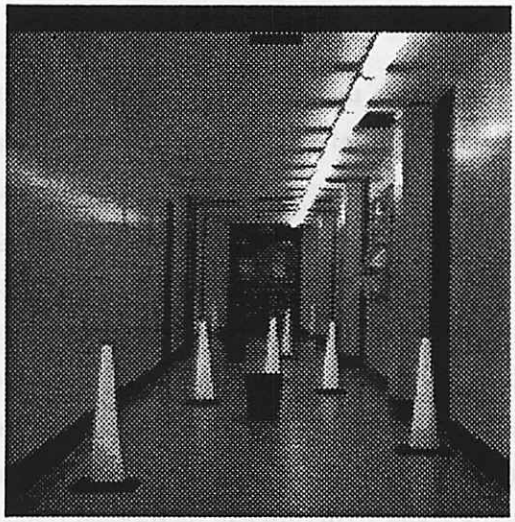


Figure 1: **A Hallway scene with shallow and non–shallow structures.**

need an internal representation of the structures.

In this work, constraints derived from the shallowness property are employed to identify shallow structures among the larger scale structures in the background. A general formulation is developed and a dynamic algorithm is presented which works over a sequence of images captured by a camera undergoing smooth motion. Hypothesized shallow structures are dynamically tracked under the shallowness assumption. Within a temporal window of a few frames, true shallow structures are extracted from the set of hypothesized aggregate structures on the basis of both the consistency of predictions in tracking and the depth of the structure. In other words, temporal evolution of a hypothesized structure is used to verify its consistency within the constraints of spatial shallowness.

Shallow structures are shown to be *affine describable* over time. Instead of clustering image features into shallow structures on the basis of this property applied over only two frames, the idea of *affine trackability* is applied dynamically to each hypothesized shallow structure. The key idea in this work is that affine trackability can be used to segment shallow structures in a scene and to reconstruct these in 3D. Two important insights have been developed within an estimation theoretic framework for the problem of robust shallow structure tracking. First, it is observed that matching of an aggregate structure as a whole is generally unambiguous in comparison with independent matching of features within the structure. Representation of a structure as a state vector along with the associated covariance matrix that allows for uncertainties in modeling and measurements, provides a natural representation for the aggregate structure as a whole that is suitable for model matching. Second, in order to circumvent the high dimensionality of this representation in matching, a nice decoupling of the structure parameters is shown to lead to a matching problem of less complexity.

The 3D location and the dynamics of the entire aggregate structure are directly represented instead of the depth of more primitive tokens like points and individual lines. The derived descrip-

tion of the scene can be viewed as a set of fronto–parallel planes (*cardboard cut–out* surfaces) of constant depth, one for each shallow object in the scene.

2 Relationship to Previous Work

The approach developed in this work has been inspired by the work of Crowley et al. [8] and Deriche and Faugeras [9] on multi–frame tracking of line segments in images, and by the structure from looming idea of Williams and Hanson [24], but goes beyond the framework developed by either of them. In [8] and [9], a locally–constant acceleration model is used for tracking of individual 2D line segments over a sequence of frames. Each model represents the location and dynamics of a single line segment and is kept current using a prediction and matching technique within the Kalman filtering framework. This model can lead to tracking errors even for simple cases of motion, such as a uniform translation in depth, especially in images where more than one line of similar orientation appears proximally in the image plane[2]. In contrast, the model defined here assumes both a property of 3D structure (shallowness) and smooth motion of the camera. The affine motion of a shallow structure provides a more exact trackability constraint. A shallow structure, being a collection of primitive tokens (lines and points), provides implicit figural context for more robust matching than just the primitive tokens. In addition, tracking in our work is used for segmentation and 3D reconstruction.

Williams and Hanson [24], in their work on flow–predicted line correspondences, have demonstrated that for translations in depth, reliable depth can be computed by measuring the temporal magnification (looming) of lengths and regions at approximately constant depth. Their method was demonstrated on manually selected virtual line segments and regions in the image. Automatic segmentation and temporal–persistence in tracking was not addressed. For instance, their system had limited ability to recover from undergrouping/overgrouping errors, and no ability to handle occlusions. Further, it did not assume or utilize motion continuity over time for tracking. Since it was based directly on the computed image displacements, it handles fairly arbitrary kinds of motions if the displacement fields are sound. The work presented here differs in that the notion of shallowness of depth of a structure has been formalized into a constraint which is utilized to automatically identify shallow structures in the scene. For motion with a significant component in depth, reliable depth of the structure can be computed from its scale parameter, which is related to looming and divergence.

A different approach for representing the scene as image regions corresponding to surfaces at different depths has been developed by Nelson and Aloimonos [19]. The divergence of the flow field between a pair of frames is used to divide various regions in the image into surfaces at different depths with respect to the camera. Reliable computation of flow and its divergence requires textured surfaces. In many real–world navigation scenarios, like a robot moving down

[2] A situation which is not uncommon in buildings and hallways.

a hallway, most surfaces are smooth and featureless with only a few reflectance edges. In such situations, occluding contours and significant reflectance edges are a reliable source of geometric cues. Our work uses the temporal evolution of such geometric image tokens. Furthermore, figural cues can be very naturally integrated within the framework of tracking of aggregate structures consisting of line and/or point tokens.

One of the earliest attempts at describing the scene as planar patches and its subsequent segmentation into multiple object motions was that of Adiv [1]. His approach employed the constraints on image flow from the rigid motion of a planar patch to group image regions, each region corresponding to such a motion. The input used was sparse or dense image flow and the associated confidence measures between a pair of images [4]. Again, since the method is based on image flow, it is not very reliable when the scene is composed primarily of textureless surfaces. Furthermore, Adiv's approach was limited to descriptions based on only two image frames and extensions to multiple frames has not been proposed.

Faugeras and Lustman [10] also suggest an approach for reconstructing the scene as planar patches based on line tokens. The relationship between a pair of image projections of a set of lines on a plane is derived as a projective transformation involving the plane and motion parameters. However, no clear algorithm is given for using this constraint to obtain the desired segmentation.

3 Shallowness as Affine Describability

In this and the next two sections, we show how projections of shallow structures are affine transformable over time, and present the solution for their affine parameters. Furthermore, a match measure is developed for matching predictions against the data while accounting for measurement and prediction errors.

Given a set of 3D points whose extent in depth δZ about a nominal point Z_0 is small compared to Z_0 and assuming that the rotations between two image frames are small, then the transformation of the projections of the point sets between the two time instants can be accurately approximated by a four–parameter affine transformation. Subscript i for the *ith* point is dropped in the derivation for notational convenience. A camera–centered coordinate system is chosen in which the XY–axes are in the image plane and the Z–axis points into the scene along the optical axis of the camera. The origin is the center of projection and lies on the optical axis with the image plane a focal length away from it along the positive Z–axis. The following notation is adopted:

$\boldsymbol{P}$, $\boldsymbol{p}$: *3D vector* $[X, Y, Z]$ *of an imaged point at* t *and the corresponding 2D image vector* (x, y).

$\boldsymbol{P'}$, $\boldsymbol{p'}$: *The 3D vector* $[X', Y', Z']$ *at* $t+1$ *and the corresponding 2D image vector* $[x', y']$.

Z_0, Z_0' : *The depths of the 3D centroids of the point set at* t *and* $t+1$.

δZ : *Extent in depth around the centroid.*

s : *Scale defined as* Z_0/Z_0'.

R : *The small angle approximation to the* 3×3 *rotation matrix formed out of* $[\omega_x, \omega_y, \omega_z]$.

R_z : *The* 2×2 *rotation matrix for rotations around the* $z-$*axis.*

T : *The 3D translation vector* $[T_x, T_y, T_z]$.

Ω, T_{2D} : *The 2D vectors* $[\omega_y, -\omega_x]$ *and* $[T_x, T_y]$, *respectively.*

f : *The effective focal length of the camera given a square image.*

The *weak perspective projection* equation for a shallow structure, approximated to the first order, can be written as,

$$\frac{1}{f}\boldsymbol{p} \approx \frac{1}{Z_0}(1 - \frac{\delta Z}{Z_0})\boldsymbol{P} \tag{1}$$

The rigid body transformation between the two 3D vectors is:

$$\boldsymbol{P'} = R\boldsymbol{P} + \boldsymbol{T} \tag{2}$$

Using these two equations, the relationship between the projections at the two instants can be written as:

$$\frac{1}{f}\boldsymbol{p'} = \frac{Z_0}{Z_0'}(1 - \frac{\delta Z}{Z_0'} + \frac{\delta Z}{Z_0})\frac{1}{f}R_z\boldsymbol{p} + \frac{Z_0}{Z_0'}(1 - \frac{\delta Z}{Z_0'} + \frac{\delta Z}{Z_0})\boldsymbol{\Omega} + (1 - \frac{\delta Z}{Z_0'})\frac{1}{Z_0'}\boldsymbol{T}_{2D} \tag{3}$$

Since our assumption is that the rotations, field–of–view and $\frac{1}{Z_0'}[T_x, T_y]^T$ are small, the second and higher order terms can be ignored and this transformation can be approximated as follows:

$$\frac{1}{f}\boldsymbol{p'} \approx \frac{1}{f}sR_z\boldsymbol{p} + \boldsymbol{t}, \quad \boldsymbol{t} = s\boldsymbol{\Omega} + \frac{1}{Z_0'}\boldsymbol{T}_{2D} \tag{4}$$

which is a four–parameter affine transformation (also called a *similarity transformation*). We emphasize that these assumptions are easily satisfied in most visual motion scenarios using commonly available CCD cameras. For instance, rotations up to 0.1 radians (about 5 degrees), FOVs of up to 25 degrees (maximum $\frac{X}{f}$ of about 0.2) and translations in the X and Y directions of up to 1 unit for objects as close as 10 units, satisfy these assumptions. Similarly, structures possessing a $\frac{\delta Z}{Z_0}$ ratio of 0.1 or less can be reasonably characterized shallow and therefore affine describable over time.

4 Does Affine Describability Imply Shallowness ?

The above formulation shows that if, for a structure in 3D, a fronto–parallel plane (parallel to the image plane) is a good approximation, and if the motion between two image frames is small, then

its motion in the image plane can be approximated by a four–parameter affine transformation. The question in the context of 3D reconstruction is whether this transformation is a sufficient condition too, that is, whether affine transformable patterns in the image plane correspond to shallow structures.

For the four–parameter transformation derived above, the answer to the question of existence and uniqueness is straightforward. There is always a unique fronto–parallel plane at a distance given by the scale parameter whose projections are the image patterns. However, there is no unique rigid motion which can be derived because the 2D translation parameters are a combination of the 3D rotation and translation parameters.

In addition to the issue of uniqueness, we have to say how well the reconstructed structure corresponds to some real structure. Unfortunately, there is one configuration of points for which the reconstruction can have an arbitrarily large error. For a purely translational motion, consider the point at the focus of expansion/contraction (FOE/FOC) and any other set of points which are projections of 3D points lying at an arbitrary constant depth. For this total configuration of points, the transformation is affine even though there may be an arbitrarily large difference between the depth of the point at the FOE/FOC and the remaining points. However, this is a degenerate case and can generally be avoided.

4.1 General 2D Affine Transformation

In the above formulation, we have chosen to approximate a 3D shallow structure by a fronto–parallel plane. What is the resulting description if a plane of arbitrary orientation is chosen to approximate a shallow structure? It can be shown that the four–parameter 2D affine description generalizes to a six–parameter 2D transformation for the approximation with an arbitrarily oriented plane. It is well known that the object–plane–to–image–plane transformation for a planar object under weak perspective projection is a six–parameter 2D affine transformation [16]. In other words, if a shallow structure is approximated by an arbitrary plane and not by a fronto–parallel plane, then the transformation from the object plane coordinate system to the image coordinate system is a general 2D affine transformation. Further, under rigid motion, projections of this structure over two time instants are also related through an affine transformation. Thus, projections of planar approximations of arbitrary shallow structures can be related through a general 2D affine transformation.

Given a general 2D affine transformation, what can be said about the corresponding 3D motion and planar parameters? In the context of object recognition, Huttenlocher [14] has shown that given a 2D affine transformation between a model plane and the image plane, a 3D similarity transformation (up to a reflection) that relates the model plane and its image can be recovered. In other words, the relative orientation (up to a reflection), translation (parallel to the image plane) and distance along the optical axis (inverse scale) of the model plane with respect to the image

plane can be recovered. However, this result is not directly useful for the case of motion where the model plane is not available and the affine tranformation relates two image projections of an unknown plane. For shape from textons, Kanade and Kender [15] showed that given the affine transformation between the image projections of two planar patches, the relative orientation can be recovered only if the absolute orientation of one of the patches in the camera coordinate system is known. The scale can be recovered only if the slant of one patch is known or if the slants for both the patches are equal. Extending this to the case of motion, it is evident that the relative orientation and scale cannot be recovered in general from the six–parameter affine transformation.

5 Solving for Affine Parameters and Their Covariances

Given a set of line correspondences in two frames, we wish to compute their affine motion parameters. Although the following derivation is for lines, it is easy to generalize it for a combined set of lines and points. The error measure is general enough to support a range of image measurement models — from strict line segments with absolutely reliable endpoints (equivalent to point tokens) to lines with infinite extent (absolute uncertainty in the longitudinal location of endpoints). As shown in Figure 2, the error measure is a sum of the parallel and perpendicular components of the vectors joining the corresponding endpoints of the line in frame $t+1$ and the affine transformed line in frame t [3]. The parallel and perpendicular directions are defined with respect to the line in frame $t+1$.

Equation 4 can be rewritten in pixel coordinates as follows:

$$\boldsymbol{p}' = D\boldsymbol{r}_s + \boldsymbol{t} \tag{5}$$

where the matrix $D = \begin{bmatrix} x & -y \\ y & x \end{bmatrix}$ is the data matrix which is constructed using the point $\boldsymbol{p} = [x \;\; y]^T$ in frame t. Vector $\boldsymbol{r}_s = [s\cos\omega_z \;\; s\sin\omega_z]^T$ is the product of scale s and rotation, ω_z, around the optical axis. With this simplification, the error measure, for a pair of corresponding lines i, is,

$$E_i = \sum_{j=1}^{2} w_{\perp_i}[(D_{ij}\boldsymbol{r}_s + \boldsymbol{t} - \boldsymbol{p}'_{ij}) \cdot \boldsymbol{n}'_i]^2 + w_{\|_i}[(D_{ij}\boldsymbol{r}_s + \boldsymbol{t} - \boldsymbol{p}'_{ij}) \cdot \boldsymbol{l}'_i]^2 \tag{6}$$

Here j refers to endpoint 1 or 2, $w_{\perp_i}$ and $w_{\|_i}$ are the weights for the perpendicular and parallel error components, respectively, and $\boldsymbol{n}'_i$ and $\boldsymbol{l}'_i$ are the unit normal and direction, respectively, of the line in frame $t+1$. It is clear from Figure 2 that the first term in the above equation is the weighted perpendicular distance between the affine transformed endpoint of a line at t to the corresponding line in the next frame. The second term is the weighted longitudinal distance. The weights associated with each of the error components can be chosen appropriately for both points and lines extracted from the image data. In order to model a circular uncertainty region associated with an extracted point token, $w_{\perp_i}$ can be set equal to $w_{\|_i}$. If $w_{\|_i}$ is set to 0, then the error

measure captures the error model for lines represented as infinite lines. Similarly, measurement errors for line segments can be represented by appropriate choices of the two weights. For example, for lines typically $w_{\perp_i}$ is much larger than $w_{\|_i}$, reflecting the known noise characteristics of most line extraction algorithms. In general, the weights can be suitably chosen depending on the type of token used and the associated noise model of the extraction process.

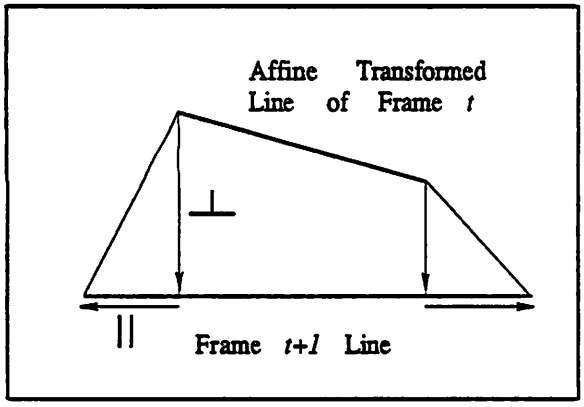


Figure 2: **The parallel and perpendicular error components.**

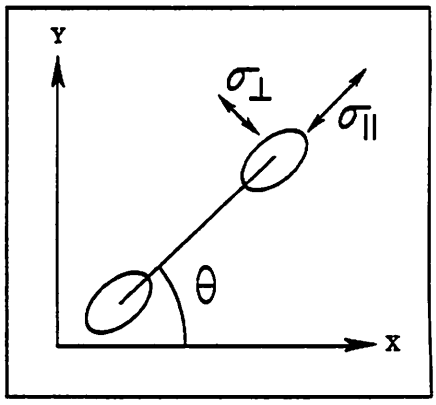


Figure 3: **The model for noise in lines.** Parallel and perpendicular endpoint uncertainties.

For a set of token correspondences, the unknown parameters r_s and t can be found by minimizing $\sum_i E_i$. Through a series of simple algebraic manipulations it can be shown that the following linear system gives the solution:

$$\begin{bmatrix} M_{24} & M_{13} \\ M_{13}^T & M_{56} \end{bmatrix} \begin{bmatrix} t \\ r_s \end{bmatrix} = \begin{bmatrix} v_{12} \\ v_{34} \end{bmatrix} \tag{7}$$

where[3]

$$\begin{array}{lllllllllll}
M_{13} & = & M_1 & + & M_3\,, & M_1 & = & \sum_i w_{\perp_i} n_i' n_i'^T D_i\,, & M_3 & = & \sum_i w_{\|_i} l_i' l_i'^T D_i\,; \\
M_{24} & = & M_2 & + & M_4\,, & M_2 & = & \sum_i w_{\perp_i} n_i' n_i'^T\,, & M_4 & = & \sum_i w_{\|_i} l_i' l_i'^T\,; \\
M_{56} & = & M_5 & + & M_6\,, & M_5 & = & \sum_i w_{\perp_i} D_i^T n_i' n_i'^T D_i\,, & M_6 & = & \sum_i w_{\|_i} D_i^T l_i' l_i'^T D_i\,; \\
v_{12} & = & v_1 & + & v_2\,, & v_1 & = & \sum_i w_{\perp_i} n_i' n_i'^T p_i'\,, & v_2 & = & \sum_i w_{\|_i} l_i' l_i'^T p_i'\,; \\
v_{34} & = & v_3 & + & v_4\,, & v_3 & = & \sum_i w_{\perp_i} D_i^T n_i' n_i'^T p_i'\,, & v_4 & = & \sum_i w_{\|_i} D_i^T l_i' l_i'^T p_i'\,.
\end{array}$$

The vector r_s computed from Equation 7 can be further decomposed into the two parameters s and ω_z.

5.1 Modeling Uncertainties in Image Lines

Lines extracted in images are more reliable in their lateral than in their longitudinal locations. The unreliability of their endpoints is, in general, due to overgrouping/undergrouping, occlusions/deocclusions and corner effects of the intensity surface. The uncertainties in the endpoints

[3] In the following we drop the subscript j for the endpoints and assume the error term for each line includes both endpoints.

of a line can be modeled as variances, $\sigma_{\parallel}^2$ and $\sigma_{\perp}^2$ which are the parallel and perpendicular uncertainties respectively in a coordinate system aligned with the line as shown in Figure 3. If the orientation of the line in the image coordinate system xy is θ, then the corresponding uncertainties in any endpoint can be expressed as [9]:

$$\Lambda_{xy} = \begin{bmatrix} \sigma_{\parallel}^2 \cos^2\theta + \sigma_{\perp}^2 \sin^2\theta & (\sigma_{\parallel}^2 - \sigma_{\perp}^2)\cos\theta\sin\theta \\ (\sigma_{\parallel}^2 - \sigma_{\perp}^2)\cos\theta\sin\theta & \sigma_{\parallel}^2 \sin^2\theta + \sigma_{\perp}^2 \cos^2\theta \end{bmatrix} \tag{8}$$

5.2 Covariances of the Affine Parameters

If $w_{\perp_i}$ and $w_{\parallel_i}$ are chosen to be the reciprocal of the perpendicular and parallel variances ($\sigma_{\perp}^2$ and $\sigma_{\parallel}^2$), then Equation 6 represents a standard weighted linear least squares problem. Its solution, given in Equation 7, can be written concisely as $M_{tot}v_{aff} = v_{tot}$, where the new symbols have the obvious correspondence with their expansions in the equation. Using a standard result for the covariances of the output parameters of a least squares problem [20], the covariances of the affine parameters can be written as

$$\Lambda_{r_s t} = M_{tot}^{-1} \tag{9}$$

where $\Lambda_{r_s t}$ is the 4×4 covariance matrix of the affine parameters r_s and t.

This completes the discussion of the estimation of affine parameters and their covariances, given correspondences between noisy measurements of a set of line pairs in two frames. In the next section, a representation for aggregate structures is developed and a match measure for comparing two such structures is derived.

6 Aggregate Structure Representation and Matching

It is emphasized that an aggregate structure refers here to any set of lines (and/or points), shallow or non–shallow, and hence is called a hypothesized structure. The constituent lines of such a structure are used in two distinct ways by the algorithm — as infinite lines for motion computation and as line segments for prediction and matching. Each use imposes its own requirements on the representation of a line and consequently, on the representation of a hypothesized aggregate structure.

Given a set of correspondences obtained by matching the prediction of an aggregate structure with its appearance in a newly acquired frame, the affine motion parameters are solved for by treating the lines as infinite lines, that is, $w_{\perp_i} = 1$ and $w_{\parallel_i} = 0$ in Equations 6 and 7. This leads to the most accurate affine parameters possible for a set of lines even when lines break or grow, or become partially occluded, since only the transverse position of the lines needs to be accurate.

For prediction and matching, this is not sufficient, especially when a model includes only a small number of lines. In particular, it can be shown that for small line sets, the longitudinal image

location of the affine projected lines in frame $t+1$ can be quite erroneous with respect to the data lines even when the residual error for the affine solution based on the perpendicular error is small (Section 8). Thus, if the idea of shallowness is to be fully exploited for matching, then lines with infinite extent cannot be used. Moreover, although the extent and location of a line is relatively unreliable, this information still imposes a strong constraint on its motion when the uncertainties are modeled correctly.

Thus, we employ both requirements in different phases of the algorithm to achieve a representation appropriate for both aggregate structures and their constituent lines. It is easily shown that if lines are treated as infinite when solving for the affine parameters, then a minimum of three lines not intersecting at a single point overconstrain the solution. In fact, any set of parallel lines or any set of lines all intersecting in a single point lead to an infinite number of solutions. Consequently, a primitive aggregate structure is defined as a set of three (or four) lines. The degenerate configurations are automatically detected when solving for the affine parameters. We emphasize here that the algorithm and its implementation are not restricted to sets of only three lines. However, this is the minimum number sufficient for simple shallow structure segmentation while keeping the complexity of matching to a minimum.

6.1 Representing Lines and Aggregate Structures

Each line is represented as a finite line segment with the four–tuple

$$l_s = [\, x_m \;\; y_m \;\; \theta \;\; l \,]^T$$

where $(x_m,\ y_m)$ is its midpoint, θ the orientation and l its length. Given the model of perpendicular and parallel uncertainties, the covariance matrix for the model of a segment is:

$$\Lambda_{l_s} = \begin{bmatrix} \frac{1}{2}\Lambda_{xy} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} \\ \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 2\sigma_\perp^2/l^2 & 0 \\ 0 & 2\sigma_\parallel^2 \end{matrix} \end{bmatrix} \tag{10}$$

where Λ_{xy} is the 2×2 endpoint covariance matrix of Equation (8). It was shown by Deriche and Faugeras [9] that in this representation the covariance matrix for a given line is independent of its position in the image plane. Also, the midpoint uncertainties are uncorrelated with the orientation and length; this will be used to advantage in the matching.

Each aggregate structure of three lines is represented as a hypothesized 3D structure in two parts. Its image projection is a 12×1 vector,

$$M_{Iloc} = [x_{m1} \;\; y_{m1} \ldots x_{m3} \;\; y_{m3} \;\; \theta_1 \;\; l_1 \ldots \theta_3 \;\; l_3]^T \;=\; [{M_m}^T \;\; {M_{\theta l}}^T]^T \tag{11}$$

composed of the three line segments. All the midpoints have been concatenated. Its 3D location is represented as its currently estimated depth $\hat{Z}$. Note that this 13×1 representation (image location

and depth) completely defines an aggregate shallow structure in 3D; it is called the *location state* of the structure in the following.

The dynamics of the structure are represented by the current four affine motion parameters (Equation 7), their covariance matrix, the total residual error (Equation 6) and a *projection error*. Recall that the residual error measures only the error in the transverse positions between lines in frame $t+1$ and the affine transformed lines in frame t. In addition, to measure the total image location error for the affine projected aggregate structure, a projection error is defined which measures the sum of the *Mahalanobis distances* [9, 17] between the affine projections of *line segments* in frame t and the corresponding lines in frame $t+1$. That is,

$$merr_{proj} = \sum_{i=1}^{3} (l'_{s_i} - l_{s_{aff-proj-i}})^T (\Lambda_{l'_{s_i}} + \Lambda_{l_{s_{aff-proj-i}}})^{-1} (l'_{s_i} - l_{s_{aff-proj-i}}) \qquad (12)$$

where l'_{s_i} is the vector for the *ith* line segment in frame $t+1$, and $l_{s_{aff-proj-i}}$ is the vector for the affine projected corresponding line of frame t. The Mahalanobis distance between two state vectors is the covariance normalized Euclidean distance between them.

These location and motion state vectors completely describe the current location and the current affine motion parameters of a given structure along with their associated covariances.

6.2 Model Matching with Measurement and Prediction Errors

For the purposes of the development in this section, it is assumed that the predicted affine parameters and their covariances for a hypothesized aggregate structure at time $t+1$ are available from its past history. The specific prediction model used is discussed in Section 7.

6.2.1 Sources of Error

The process of matching the predicted model structure with potential aggregate matches must account for three sources of error:

1. Measurement uncertainty in the image data on which the prediction is based.
2. Departures from modeled predictions of motion, (e.g. non–uniform motion).
3. Error in affine description due to departures from a fronto–parallel plane for the real shallow structure.

First, each of these sources of error is discussed independently from the point of view of how they affect the location and affine motion models of an aggregate structure. In the next section, it is then shown how all these sources of uncertainty are incorporated into a unified error model through the covariances of the predicted model.

It is possible to account for measurement uncertainties by propagating the covariances of a line in the previous frame and those of the predicted affine parameters into the covariances of the predicted line. The problem with this approach is that if each line is matched individually to its potential match set, in effect each line is allowed a perturbation within the limits of its variance independent of the other lines in the model. This is not desirable since beyond the individual line measurement uncertainties, model matching should incorporate the perturbation of the model as a whole when searching for the best match.

In order to model deviations from uniformity of motion in the prediction process for the motion of aggregate structures, we now analyze the typical imaging scenario for possible non–uniformities. Assume that the camera is mounted on a mobile platform[4] and the sequence of frames is sampled uniformly in time under smooth motion. The most significant source of error in this scenario is excessive rotation around either of the three axes. These errors typically occur due to two major causes — (i) the rotations induced due to non-uniformity of torque on the wheels or differential slipping and (ii) the small differential slants and tilts (small bumps, shallow depressions and ramps) on an otherwise planar ground plane. Out of the three rotations, those in depth (ω_x and ω_y) are the dominant source of error in prediction for small FOV cameras [1]. Consequently, errors in rotation in depth are modeled as uncertainties in certain of the affine motion parameters. In addition, it is to be emphasized that uncertainties due to errors in the other motion parameters also can be handled within the framework of dynamic prediction and matching with uncertainties.

It was shown in Equation 4 that the 3D rotations ω_x and ω_y lead to translations in the image plane under the shallowness constraint. The non–uniformity of motion is primarily accounted for by adding a diagonal covariance 2×2 matrix, $\Lambda_{t_{err}}$, to the already computed covariance, $\Lambda_{t_{pred}}$, for the predicted affine translation vector. This is equivalent to adding plant noise to the dynamic model in a Kalman–filter [12]. Similarly, uncertainties in the prediction of the other two parameters (s and ω_z) due to motion uncertainties can be modeled by increasing their measurement covariances. An advantage of handling non–uniformity in this way is that it provides a principled method for model matching while allowing for modeling uncertainties.

The third source of error, approximation of a structure by a fronto–parallel plane, can also be modeled by allowing for uncertainties in the predicted parameters. In this case, however, the constituent lines in the structure will be affected independently and not as a whole model. Each line can be the projection of a real 3D line that lies in front or behind the reconstructed plane with equal probability. The parameters of a predicted line that are most likely to be affected by this are the scale and the orientation.

[4]Like the Denning vehicle used in all our experiments.

6.2.2 A Model Match Measure

If we consider the complete specification of the model as the 12×1 image location vector of Equation 11 and match this model as a whole using the Mahalanobis distance as the match measure, it requires the inversion of a 12×12 covariance matrix for every match to be checked. This is not very practical. However, from the discussion in the previous section, the major uncertainty is expected to be in the prediction of the translation parameters. The translation parameters affect only the location of the midpoints for each of the lines and not their orientation or lengths. Thus, the 12×1 vector was separated in Equation 11 into a 6×1 sub–vector of midpoints and a 6×1 sub–vector of orientations and lengths.

Now we show that computing the propagated covariances of the 6×1 vector of midpoints achieves resilience to errors in prediction, due to the non–uniformity of motion, as expected. Assuming that at time instant t, $r_{s_{pred}}$ and t_{pred} are the affine parameters for the predicted motion between t and $t+1$, the predicted vector of midpoints can be written in terms of $r_{s_{pred}}$ and t_{pred}, and the data lines in frame t (using Equation 7) as follows:

$$M'_m = M_D r_{s_{pred}} + I_D t_{pred} \,, \tag{13}$$

$$M'_m = [x'_{m1}\ y'_{m1} \ldots x'_{m3}\ y'_{m3}]^T$$

$$D_i = \begin{bmatrix} x_{mi} & -y_{mi} \\ y_{mi} & x_{mi} \end{bmatrix} \qquad I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$M_D = [D_{m1}^T\ D_{m2}^T\ D_{m3}^T]^T \qquad I_D = [I_2\ I_2\ I_2]^T$$

Using the above equations, it is easy to derive the covariance matrix of the 6×1 vector M'_m.

$$\Lambda_{M'_m} = R_{sbig} \Lambda_D R_{sbig}^T + M_D \Lambda_{r_{s_{pred}}} M_D^T + I_D \Lambda_{t_{pred}} I_D^T \tag{14}$$

where $R_s = Rotation\_Matrix[\omega_{z_{pred}}]$ is the 2×2 rotation matrix for angle $\omega_{z_{pred}}$, $R_{sbig} = diag[R_s\ R_s\ R_s]$ (a matrix with the 2×2 rotation matrices for $\omega_{z_{pred}}$ along its diagonals and zeros elsewhere), $\Lambda_D = diag[\Lambda_{m1}\ \Lambda_{m2}\ \Lambda_{m3}]$ where Λ_{mi} is the 2×2 midpoint covariance matrix for the *ith* model line of the previous frame, and $\Lambda_{r_{s_{pred}}}$ and $\Lambda_{t_{pred}}$ are the 2×2 covariances of the predicted affine parameters. $\Lambda_{r_{s_{pred}}}$ and $\Lambda_{t_{pred}}$ have been considered independent for convenience. A similar form could easily be derived under the assumption that they are correlated.

In order to provide insight into how this combined covariance matrix of midpoints actually encodes the model uncertainties due to motion non–uniformity, consider the last term in Equation 14. It was discussed above that modeling errors are added to $\Lambda_{t_{pred}}$ to account for perturbations. The last term thus transforms $\Lambda_{t_{pred}}$ into the 6×6 matrix

$$\begin{bmatrix} \Lambda_{t_{pred}} & \Lambda_{t_{pred}} & \Lambda_{t_{pred}} \\ \Lambda_{t_{pred}} & \Lambda_{t_{pred}} & \Lambda_{t_{pred}} \\ \Lambda_{t_{pred}} & \Lambda_{t_{pred}} & \Lambda_{t_{pred}} \end{bmatrix}$$

Thus, the modeling errors induce covariances across the lines in the predicted model and achieve the coupling desired for the search for an appropriate match. This has the effect of allowing the model to rigidly translate within a given region of uncertainty and still find a good match if one exists.

The match measure is the Mahalanobis distance between the 6×1 midpoint vectors (M'_m and M_m of Equation 11), and the orientations and lengths of the constituent lines in the predicted model and the potential match structure. That is, *mmeas* is given by,

$$\begin{aligned} mmeas &= (M'_m - M_m)^T (\Lambda_{M'_m} + \Lambda_{M_m})^{-1} (M'_m - M_m) \\ &\quad + \sum_{i=1}^{3} \left[(\Lambda_{\theta'_i} + \Lambda_{\theta_i})^{-1} (\theta'_i - \theta_i)^2 + (\Lambda_{l'_i} + \Lambda_{l_i})^{-1} (l'_i - l_i)^2 \right] \end{aligned} \tag{15}$$

Here Λ_{M_m} is the covariance matrix of midpoints of the potential data matches with the three 2×2 $\frac{1}{2}\Lambda_{xy}$'s (Equation 8) along its diagonals. $\Lambda_{\theta'_i}$, Λ_{θ_i}, $\Lambda_{l'_i}$ and Λ_{l_i} are the variances in orientations and lengths of the constituent lines in the predicted and the potential match structures.

7 Shallowness as Affine Trackability

In this section, we use the formulations of affine describability and model–matching developed earlier to design an algorithm to decide whether or not a hypothesized structure is shallow based on its trackability as an affine structure.

As mentioned earlier, we are interested in applying the system to man–made environments where most surfaces are smooth and largely textureless, and lines provide a fairly complete description of the image in terms of surface boundaries and significant surface markings. In general, shallow structures in the image are composed of only a few lines. Thus we cannot rely on Hough–like clustering techniques over two frames, where every primitive structure votes for a set of affine parameters and sets of structures with similar parameters are clustered as shallow structures [1]. However, the evolution of a hypothesized structure over time is an alternative source of measurement which can be used to check the validity of a shallowness hypothesis, even when it involves a set of only a few lines. The essential idea is that if a hypothesized structure can be consistently tracked

and its 3D depth over time is consistent with a shallow structure model, then the structure is identified as shallow, otherwise it is labeled non–shallow.

7.1 Cycle of Prediction and Matching

A hypothesized aggregate structure, as defined in Section 6, undergoes a cycle of prediction and matching over a sequence of frames, with both the location and dynamic state vectors updated for each frame, before it is declared shallow or non–shallow. The process consists of the following phases:

- Bootstrap Phase
- Tracking Phase consisting of *Prediction*, *Matching* and *Update*.

Bootstrapping occurs only once for every new structure instantiated in any frame. The three parts of the tracking phase are repeated cyclically. Instead of always representing the motion and depth between consecutive frames, a moving window of, say m, frames is considered. The first frame in this window is called the *anchor frame*. The anchor frame for a freshly instantiated structure is its frame of instantiation. For every newly acquired frame in the window the motion parameters are computed and depth represented with respect to the anchor frame. This improves reliability of motion and location computations over time because the magnitude of motion starting from the anchor frame increases successively with every newly acquired frame. The following description assumes that the translation possesses a z–component (translation in depth) along with the x and y components. It can be easily modified for the cases when the z–component is zero. Also, a non–zero z–component of translation is necessary if the scale parameter is to be used for depth computation. It is also expected that the magnitude of the translation, say T_z, is known; otherwise all depth computations are with respect to a scale of unity.[5]

7.1.1 Bootstrap Phase

For a newly instantiated structure (nominally a triple of lines), the motion of the structure is unknown. The line tracking algorithm of Williams and Hanson [24], which matches lines to their displacement field–based predictions, is used to generate correspondences in frame 2. A sample of this matching is shown in Figure 4. In many instances, the flow–based predictions can lead to multiple matches. These are disambiguated by choosing the one with the best match measure of Equation 15. Using the correspondences thus derived, the initial affine motion parameters and their covariances are computed (Equations 7 and 9).

[5]Recall that this scale factor is not recoverable with any monocular motion algorithm.

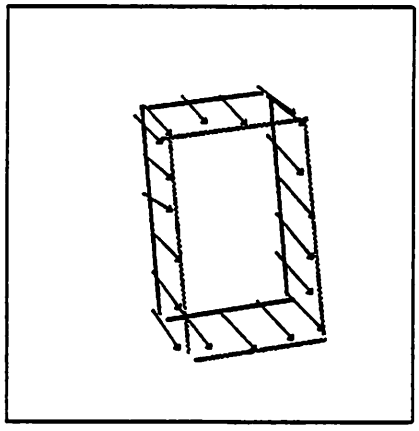


Figure 4: **Bootstrap matching using flow.** Lines in bold are frame 1 lines and those in lighter gray are frame 2 lines. The displacement vectors approximately along the length of the lines are shown as lines with arrows.

7.1.2 Tracking Phase

In the *prediction phase* of tracking, at time t, the motion parameters between the current anchor frame 1 and frame t in the current window are used to predict the motion between frames t and $t+1$. The covariances are propagated into frame $t+1$ as well. The predictions assume uniformity of motion but non–uniformity is dealt with by modeling the uncertainties in the predictions within the framework developed in Section 6.2. So the motion parameters between t and $t+1$ are,

$$\begin{aligned} s_{pred} &= 1/(1 + \frac{1 - s}{t - 1}) \\ \omega_{z_{pred}} &= \omega_z \,/\, (t - 1) \\ t_{pred} &= \frac{1}{t - 1} t \end{aligned} \tag{16}$$

Under the assumption of small rotations, these provide fairly good predictions for uniform motion. From the computed covariances of the affine parameters r_s and t in Equation 9, the covariances of the predictions can be easily derived. These are called $\Lambda_{r_{s_{pred}}}$ and $\Lambda_{t_{pred}}$, as in Section 6.2. It was shown there how these are employed to handle deviations from uniformity.

The predicted affine motion parameters are used to project each line in the aggregate structure at frame t into its position in frame $t+1$ to obtain the predicted structure. Around each predicted line, a window query is performed to obtain potential matches for each line. Let L_1, L_2 and L_3 be the three potential match sets for each line respectively in an aggregate triple of lines. Then all the triples from the product set $L_1 \times L_2 \times L_3$ are matched against the prediction.

In the *matching phase*, the match measure of Equation 15 is computed for each potential data triple against the prediction, and the best triple below a threshold is chosen as the match. This threshold depends on the model of measurement errors in lines, the allowable non–uniformity in motion, and the extent to which the real 3D structure is not a fronto–parallel plane. If all the

errors are assumed to be Gaussian, then the Mahalanobis distance of Equation 15 has a chi–squared distribution with the appropriate degrees of freedom [5]. A threshold on this distance can be chosen by using the chi–squared value corresponding to a desired level of confidence in accepting a match. However, it is not reasonable to assume that all the sources of error are Gaussian. For instance, errors in prediction arising from the departure of a structure from being a fronto–parallel plane cannot, in general, be modeled as Gaussian. This is especially true when the structure consists of a small number of tokens as is the case here. In such situations, the error in modeling the structure dynamics can be systematic. Ideally, an on–line determination of this process noise [18] is desirable so that a threshold for this source of error can be automatically chosen based on the allowable departure from shallowness. In order to accomplish this, more theoretical work along the lines of adaptive filtering needs to be done. In our implementation, the chi–squared values have been used in conjunction with an experimentally determined threshold. It is to be emphasized that in all the experiments, the tracking has been found to be robust for the choice of the same threshold throughout.

Once an acceptable match is found, the model's new motion parameters are computed between the anchor frame and the current frame using the newly found matches. This is called the *update phase*. The covariances of the current location vector and the computed affine parameters are also recomputed. Since depth is a part of the location vector, it is also updated. Additionally, the variance–weighted sample mean and sample dispersion of depth are updated by incorporating the new measurement.

An acceptable match may not be found in the current frame due to failures of line grouping, occlusions and motion discontinuities. The algorithm allows for graceful handling of many of these conditions by upgrading the current prediction to model status whenever a suitable match is not found. That is, the prediction serves as the best current model in the absence of a good match to the data. A counter which keeps track of the number of frames missed is also incremented. In addition, the variances of the model's motion parameters and those of the line segments for its potential matches in the next frame are increased, and consequently, the search window for the next prediction/matching phase is expanded. If a match is re–acquired after a lapse in the previous frame, the motion variances and the window size are reduced, but not below the levels at the start of tracking.

There is an issue of computational complexity versus the temporal persistence of a model when a match is not found. After every frame in which a match is not found, the search windows become larger thus increasing the number of potential matches. This leads to an increase both in the computational expense for matching and the possibility of false matches. In general, there is no theoretically sound mechanism to address this problem because a combination of failures can always be designed to defeat any mechanism. However, any practical system, in which this algorithm is embedded, can place hard computational bounds on the time spent on search. If this maximum

limit is reached then it might be reasonable to abandon the current model being tracked. For instance, consider the case where an object is occluded and either remains occluded for a large number of frames, or undergoes a significant change in motion (say reverses direction) while it is occluded. In general, the actual position of the object could be far away from the predicted location when it is reacquired by the system. In such a case, it seems reasonable to abandon the current model and to reinstantiate a new model for the object when it is again seen in the image.

The last three cycles of the tracking phase discussed above are repeated for every new hypothesized aggregate structure within its window of frames. If 1) it has been tracked for more than half the frames in the window, and 2) its depth dispersion is within an allowed limit, and 3) its projection error (Equation 12) for all matched frames is less than a threshold, then it is declared as a shallow structure, else it is not and is dropped from further consideration.

7.2 The Algorithm

The algorithm presented above can be applied to image data in either an interactive mode or in an automatic mode. In the interactive mode, a set of manually selected lines is presented to the algorithm as a hypothesized shallow structure. The algorithm tracks the structure as described above and declares it shallow or otherwise.

In the automatic mode, triples of lines all over the image are instantiated as hypothesized aggregate structures and the algorithm automatically cycles through them and labels any given structure as shallow or non–shallow. We employ proximity and convexity as generic heuristics to create triples of line tokens as aggregate hypotheses. In most man–made environments, appearances of objects can be described as convex regions or a union of significant convex regions enclosed by boundaries. Each pair of line segments in a triple should be completely contained in the half–space defined by the remaining line extended infinitely. Some amount of tolerance is allowed in testing for the half–space containment in that a small part of the line can straddle the half–space defining line and still qualify. Triples passing this convexity test are represented as hypothesized models and the above algorithm is applied to each one. The result is a labeling of structures in the scene as shallow and non–shallow.

The complexity of extracting triples out of image lines is $O(n^3)$, where n is the number of lines. This can be considerably improved upon by using proximity as a heuristic. Around each endpoint of a line, all lines within a given distance are chosen, and the convexity test is applied to these sets of lines. The complexity of spatial queries based on proximity is $O(1)$ if the image lines are pre–processed and are hashed into a spatial grid defined over the image plane [7]. It is reasonable to assume, and we have found it to be so in our experiments, that the number of lines in the proximal line sets is bounded by a small constant. Thus, the complexity of finding triples is almost always $O(n)$ with a fairly small constant (small compared to n). Consequently, the number of approximately convex triples found is also $O(n)$. We will present specific numbers to illustrate

this in the next section.

The inner core of the algorithm for either mode of application is the same and is presented here.

Given a set of lines constituting a hypothesized shallow structure in frame 1, the following tracking algorithm is applied. The tracking is done for a few frames before the structure is labeled. Also, the first frame in the sequence is the *anchor frame*, that is, the affine parameters are computed between this frame and every new frame. This improves reliability of motion and depth computations over time because the magnitude of motion displacement starting from the anchor frame is expected to increase with every newly acquired frame.

Various steps of the algorithm are:

Step 1: Bootstrap

Compute the line matches for frame 2 using flow–based predictions [24].

Compute the affine motion parameters and their covariances.
(Equations 7 and 9).

Instantiate a model with its location and motion states.

If (less than 3 matching lines found) declare *Non–trackable* and exit.

For every new frame t, **Repeat until** frame m processed:

Step 2: Prediction

Compute the predicted parameters between time t and time $t+1$.
(Equation 16).

Project the current model lines at t into predicted lines at $t+1$.

Compute the covariances of the predicted model using the covariances of motion and data at t, and the model noise covariances accounting for non–uniformity (Equation 15).

Step 3: Find potential match sets, L_1, L_2 and L_3 of data lines for each predicted model line: (a) within the model line's search window, and (b) selecting data lines within a $\delta orientation$ (typically, 15^o) of the model line.

Form the product set $L_1 \times L_2 \times L_3$ from the potential match sets for each line.

For each element of this set, compute the Mahalanobis distance between the element and the predicted model (Equation 15).

Choose the best match below a threshold.

If (none) increment *no-match-count* else decrement *no-match-count.*

Step 4: Update

If match found **then**

Compute the new affine parameters between frame 1 and the matches in frame $t+1$ and the associated covariances.

Update the sample weighted-mean and dispersion for the depth parameter in the model.

else

Promote the prediction to model status with increased modeling noise (Equation 14).

end Repeat.

Step 6: If *no-match-count*, *depth-dispersion* and $merr_{proj}$ (Equation 12) are all less than their thresholds (Section 7.1),

then declare model set as *shallow* else *non-shallow.*

The threshold for *depth-dispersion* is in general chosen to be some percentage (typically, 10 or 20) of the mean depth. Threshold for $merr_{proj}$ is chosen based on considerations which were discussed in Section 7.1. It is reasonable to allow the *no-match-count* to be a fraction (typically 1/3) of the number of frames in a window.

8 Experimental Results

The implementation of our algorithm was greatly facilitated by the use of a Lisp based database called the ISR [7] running on a TI Explorer II. The aggregate structures and their potential matches are instantiated through spatial queries and represented by ISR token sets.

8.1 Tracking Results

We present the tracking results on two image sequences, *cones-seq* and *room-seq-1*, both of which were captured with a SONY B/W AVC-D1 camera, with effective FOV 24 by 23 degrees mounted on a Denning robot, and digitized to 256-by-242 pixels. The camera moved into the scene with a translation magnitude measured to be approximately 1.95 feet for the *cones-seq*, and 0.39 feet for the *room-seq-1* between successive frames. Four image frames for each of these sequences are shown in Figures 5 and 6, respectively. It is emphasized that the effective motion is neither purely translational nor uniform. In each frame lines are extracted using Boldt's [6] line grouping

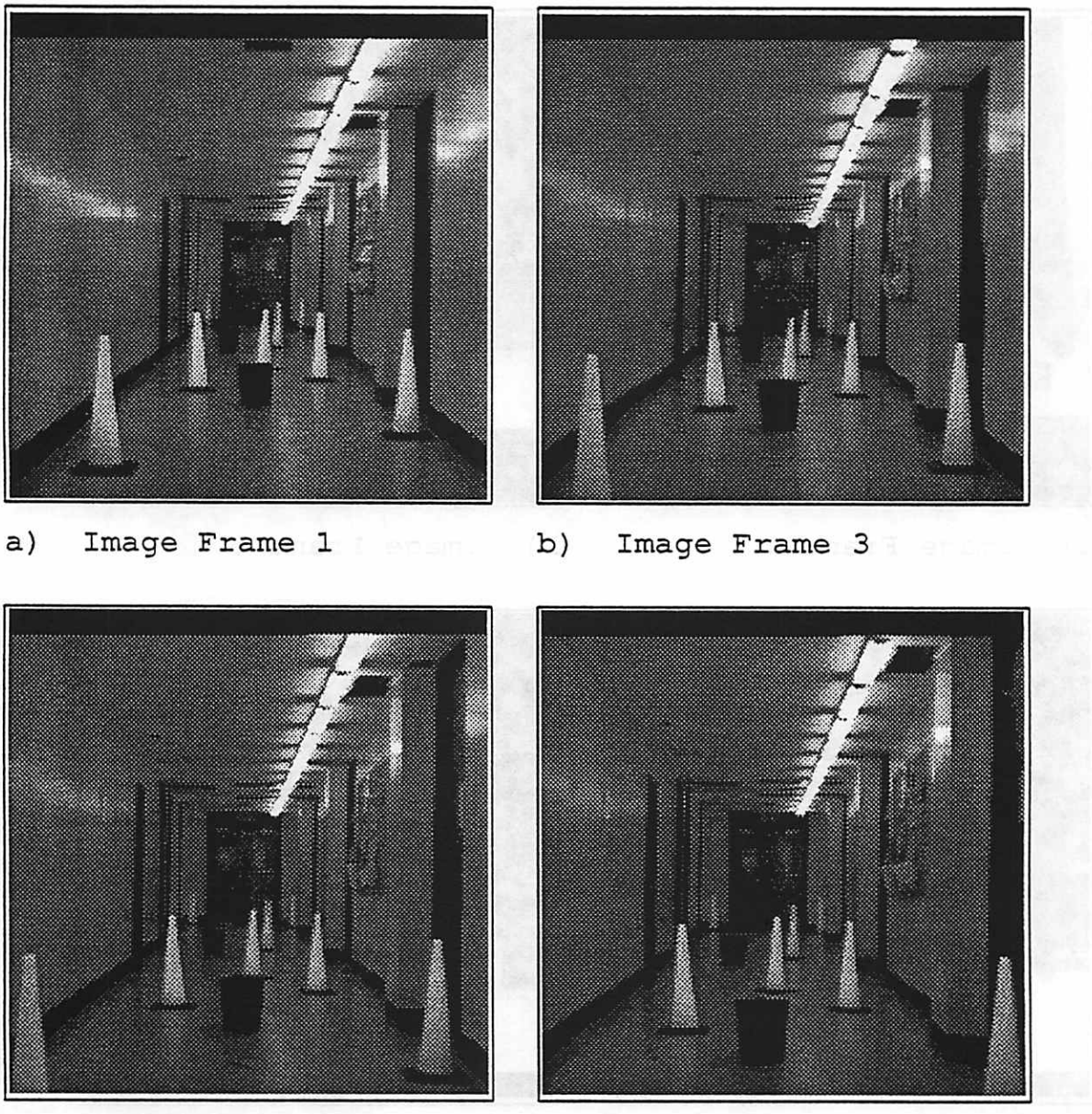


Figure 5: **Four image frames of the** *cones-seq.* Frames 1, 3, 4 and 6.

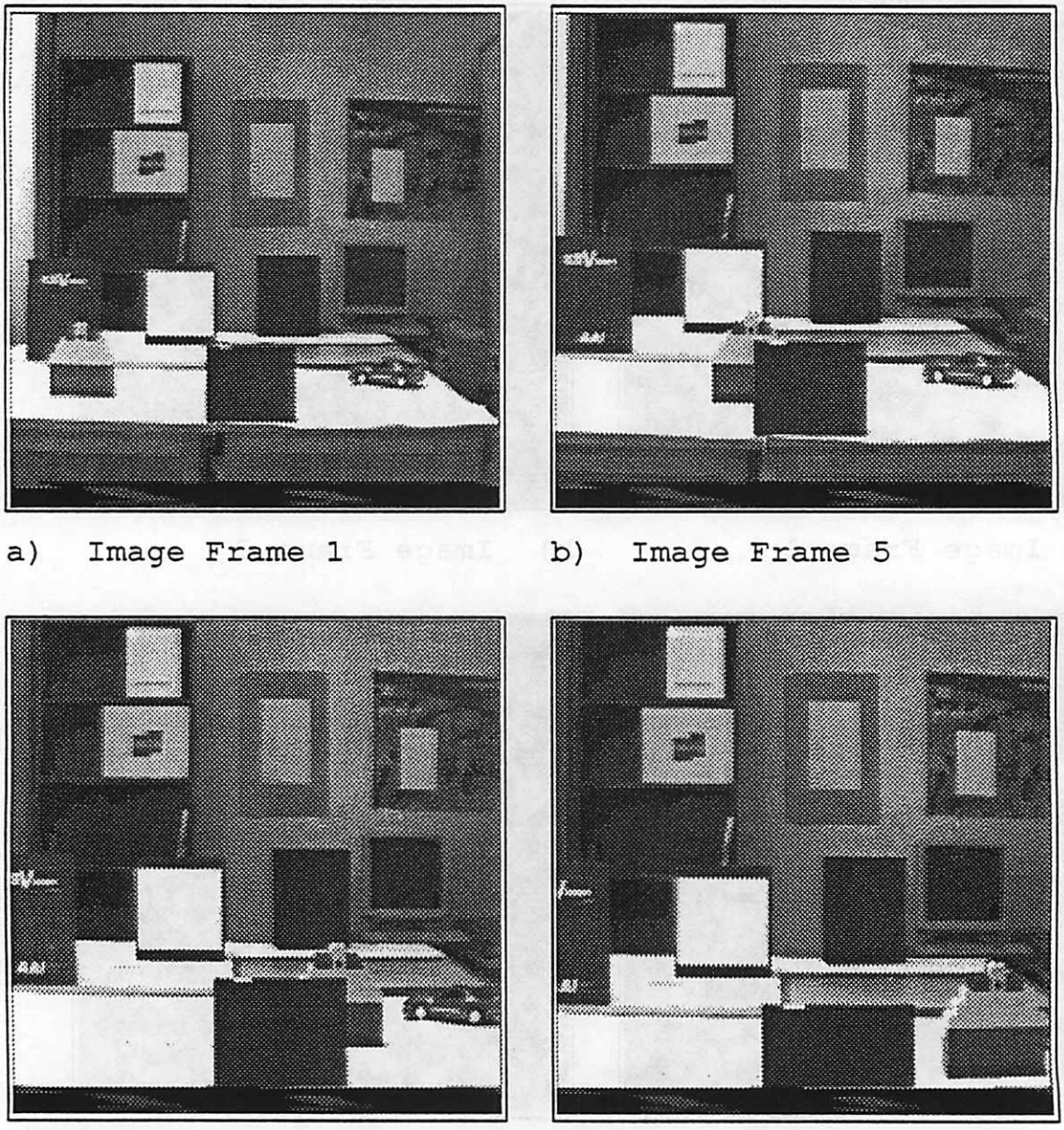


Figure 6: **Four image frames of the *room-seq-1*.** Frames 1, 5, 8 and 10.

system. A window of six frames is used for most of the results here. However, for the *room-seq-1*, some interesting aspects of the algorithm are shown with ten-frame windows.

For both sequences, Figures 8–14 are to be read left–to–right and top–to–bottom. In each figure, panels a) and b) show the hypothesized aggregate of lines overlaid in black or white on the first and the last images, respectively. Panel c) shows the structure highlighted in bold and overlaid on lines in frame 1. Panel d) highlights the corresponding structure in frame 2; the correspondence was derived in the bootstrap phase using flow–based line tracking [24]. Subsequently, corresponding to each frame in the sequence, each panel, starting with panel e) onwards, depicts matching for each successive frame. Only the region around the structure of interest is expanded and shown in detail. The prediction windows for each line are shown as shaded areas. The central spine of these windows is the actual prediction. Thin lines show all the lines in and around the region of interest. Lines of medium thickness show the union of sets of potential matches for each line. If a match is found in a frame, it is drawn using bold lines. Note that the match thresholds and window sizes have been kept the same for all the experiments with the two sequences.

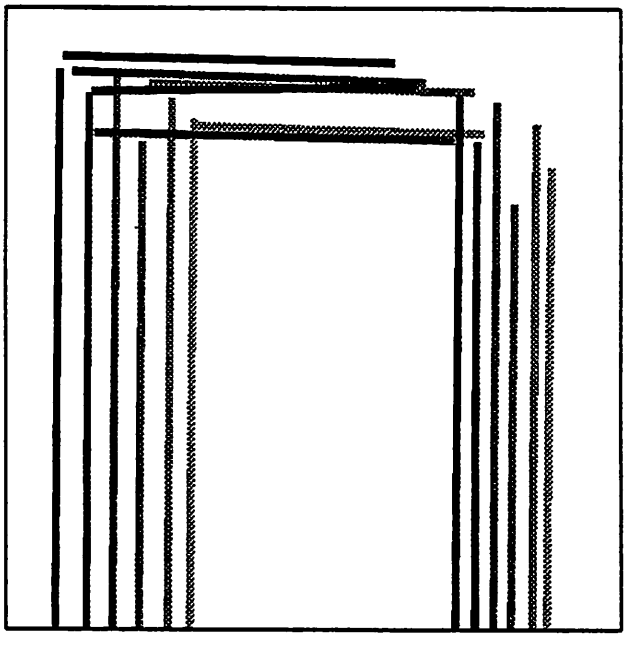


Figure 7: **Motion of the doorway lines in the *cones-seq*.** Image motion of the doorway lines from frames 1 to 6. Frame 1 lines are in the lightest shade and frame 6 in the darkest. The up and down motion in the image plane shows the non–uniformity of motion.

Figure 7 shows the image motion of the lines on the doorway at the far end of the hallway in the *cones-seq* over six frames. It is clear from the up and down motion of the image lines over time that the motion is definitely not uniform. Even on the smooth surface of the floor in the hallway, slight undulations lead to rotations in depth and around the optical axis.

Figure 8 depicts tracking of three lines on a cone. An interesting event happens in frame 4 (panel f); the left line of the cone is merged with a door line in the background by the line grouping

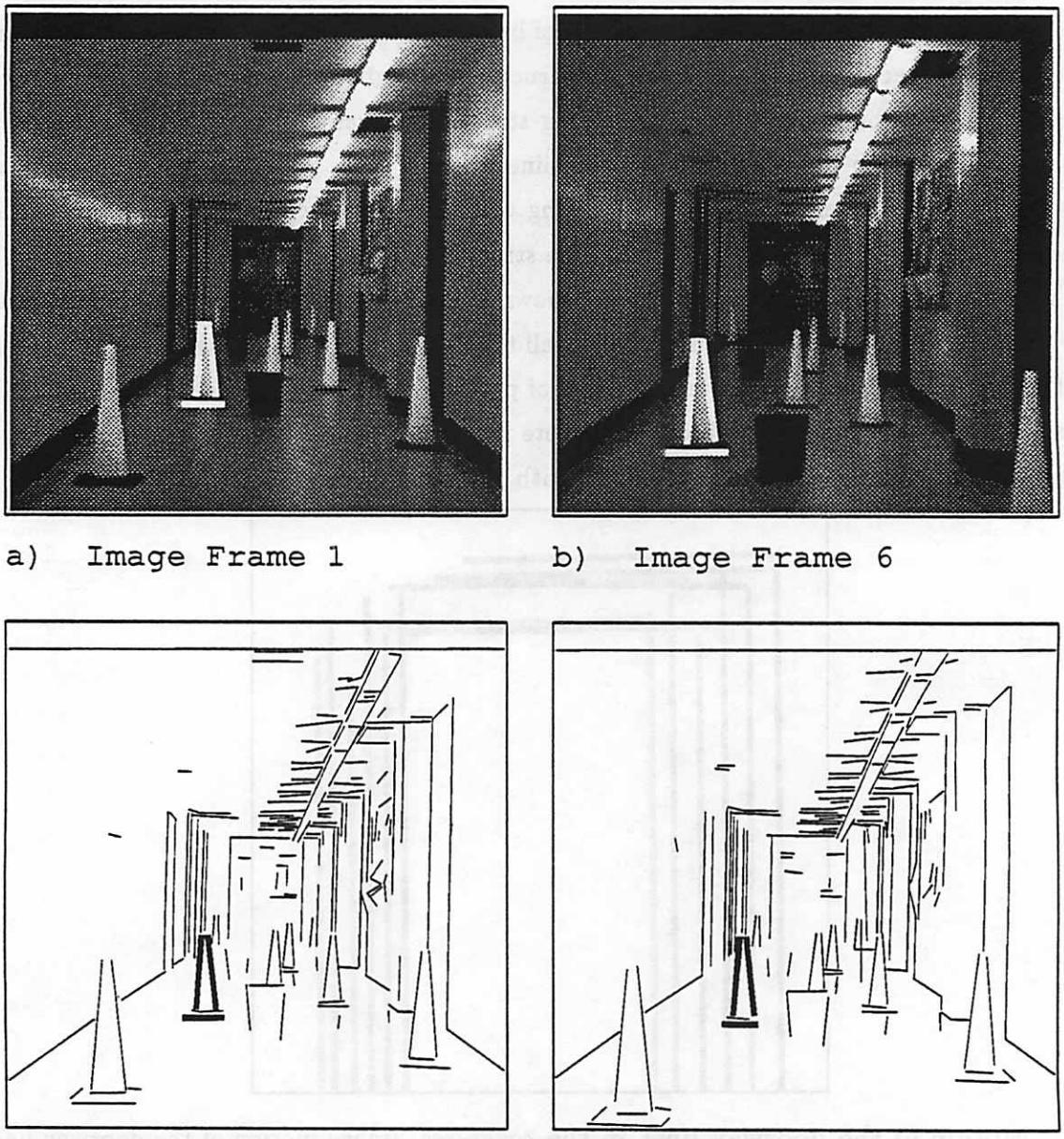


Figure 8: **Tracking of a shallow triple in** *cones–seq.* Tracking over six frames is shown. a), b): First and the last image frames with the triple highlighted; c), d): Highlighted shallow triple overlaid on lines in frames 1 and 2. The correspondence in frame 2 was obtained using flow–based prediction in frame 1.
(*contd. next page*)

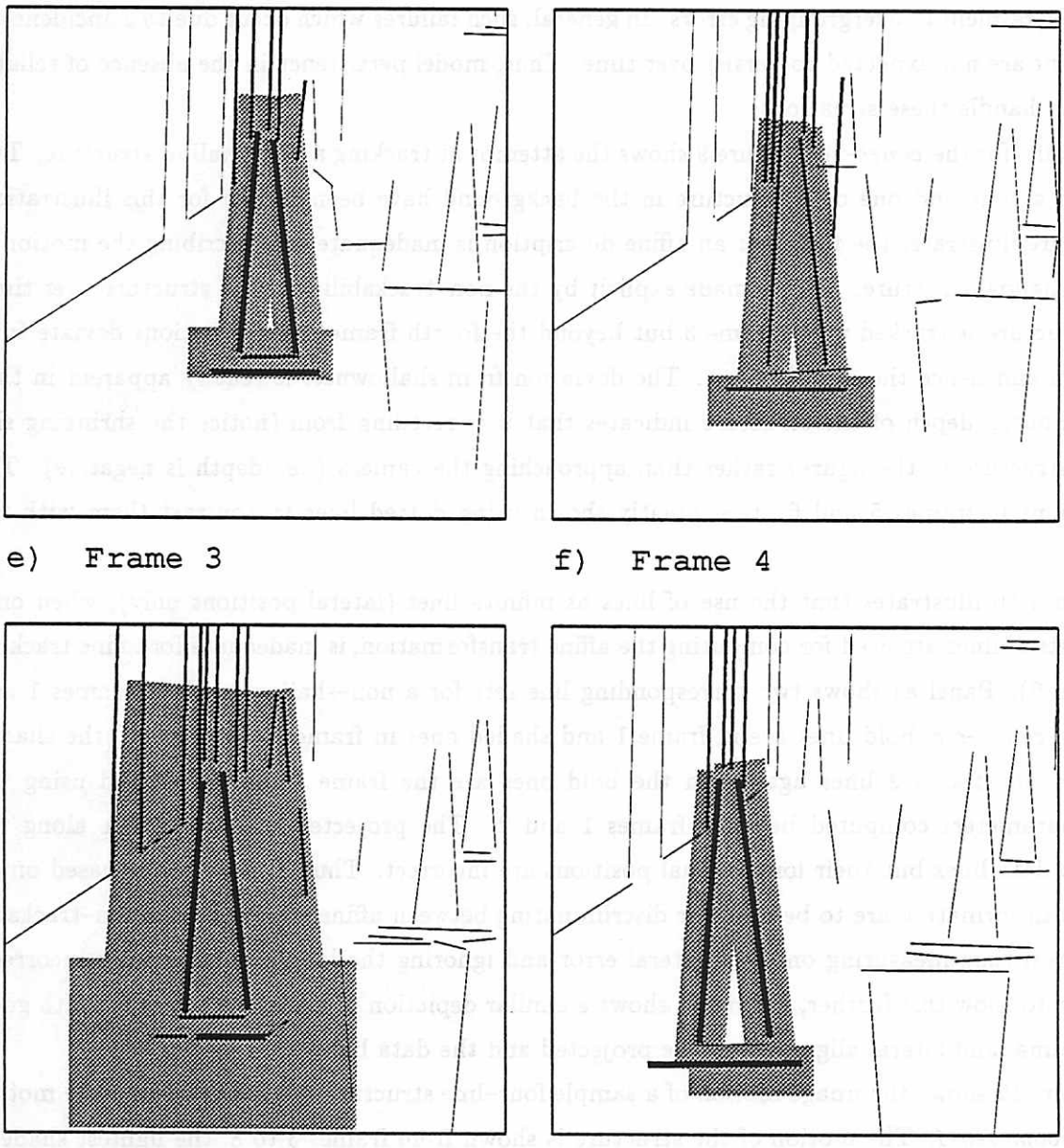


Figure 8: (*contd.*) **e)–h): Matching and tracking in frames 3–6.** The shaded areas are the search windows with the central spine being the position of predicted lines. The thinnest lines are all the lines in the background in the region of interest, medium thickness lines are the candidate matches, and the boldest lines are the matches found. f): No match found due to overgrouping of the left cone line in frame 4; g) Recovery from error by model persistence in frame 5.

system. No match is found for the structure in this frame, but its prediction persists. In the next frame, the lines separate again and the match is successfully found. This is an example of how the system is resilient to overgrouping errors. In general, such failures which occur due to coincidence of viewpoint are not expected to persist over time. Thus, model persistence in the absence of reliable data can handle these situations.

Finally, for the *cones-seq*, Figure 9 shows the attempt at tracking a non-shallow structure. Two lines on a cone and one on a structure in the background have been chosen for this illustration. The figure illustrates the fact that an affine description is inadequate for describing the motion of a non-shallow structure. This is made explicit by the non-trackability of the structure over time. The structure is tracked up to frame 3 but beyond the fourth frame, the predictions deviate from the data and hence the model is lost. The deviation from shallowness is readily apparent in that the computed depth of the structure indicates that it is receding from (notice the shrinking size of the structure in the figure) rather than approaching the camera (i.e. depth is negative). The predictions in frames 5 and 6 are explicitly shown using dotted lines to contrast them with the data.

Figure 10 illustrates that the use of lines as infinite lines (lateral positions only), when only small sets of lines are used for computing the affine transformation, is inadequate for affine tracking (Section 6). Panel a) shows two corresponding line sets for a non-shallow triple in frames 1 and 2 of the *cones-seq*; bold lines are in frame 1 and shaded ones in frame 2. In panel b) the shaded lines are the frame 2 lines again and the bold ones are the frame 1 lines projected using the affine parameters computed between frames 1 and 2. The projected lines nearly lie along the frame 2 data lines but their longitudinal positions are incorrect. Thus, if predictions based on an affine transformation are to be used for discriminating between affine trackable and non-trackable structures, then measuring only the lateral error and ignoring the longitudinal error is incorrect. In order to show this further, Figure 11 shows a similar depiction of a shallow structure with good longitudinal and lateral alignment of the projected and the data lines.

Figure 12 shows the image motion of a sample four-line structure to give an idea of the motion in the *room-seq-1*. The motion of the structure is shown from frames 3 to 8; the lightest shade is used for frame 3 and the darkest for frame 8. The motion discontinuity between frames 6 and 7 is apparent from the figure.

In Figures 13 and 14, the window is extended to ten frames to show how the algorithm handles a motion discontinuity (Figure 13) and an independently moving object (Figure 14) which is occluded/deoccluded during its course of motion.

In Figure 13, a shallow triple is tracked. Note that in frame 5 (panel g), a break in one of the lines occurs with the result that the matching fails but the correct model is reacquired in the next frame. Also, note that this triple is surrounded by a similar triple throughout the sequence (they are on the same planar surface). In frame 3, the right hand side line of the predicted triple

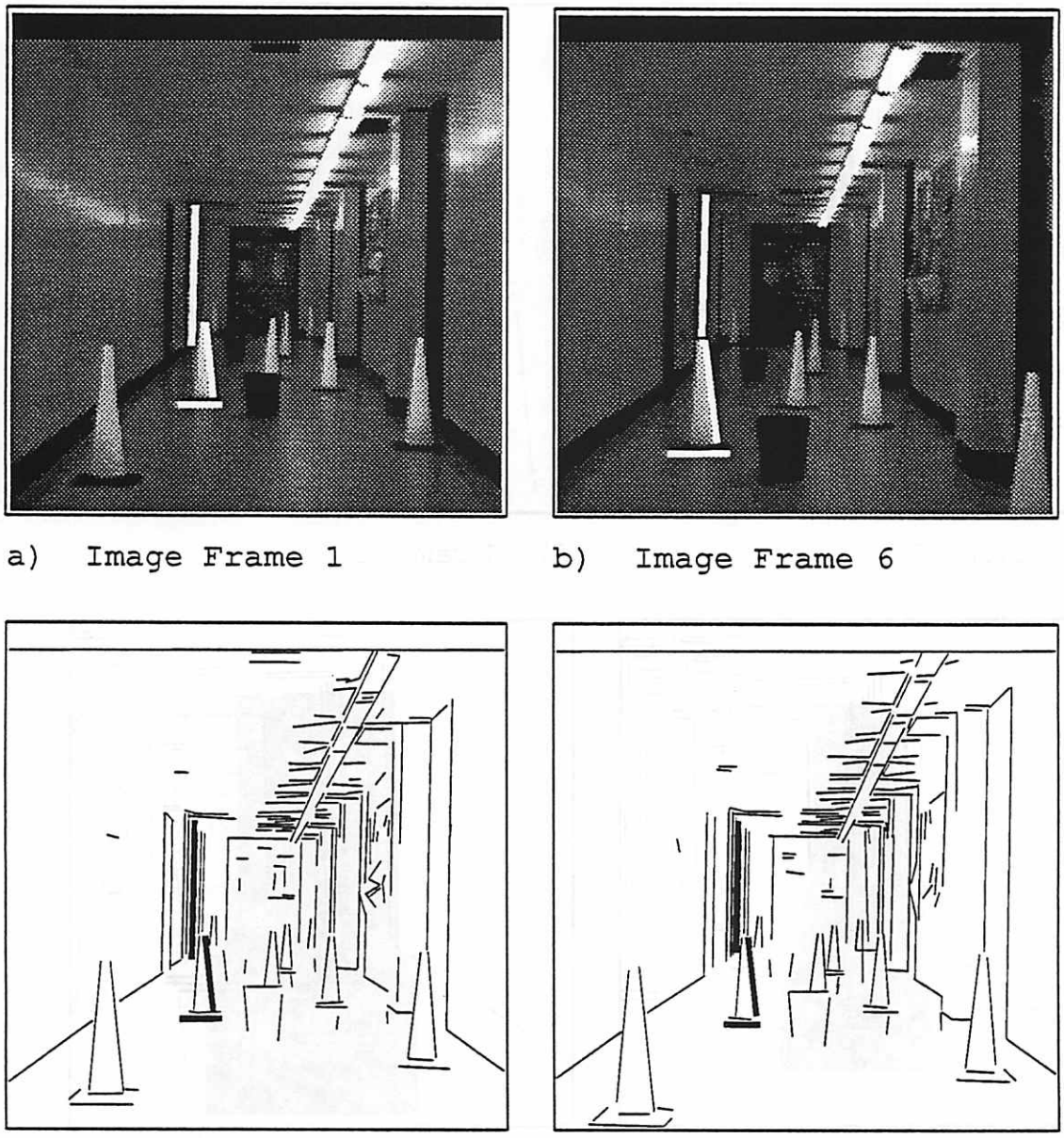
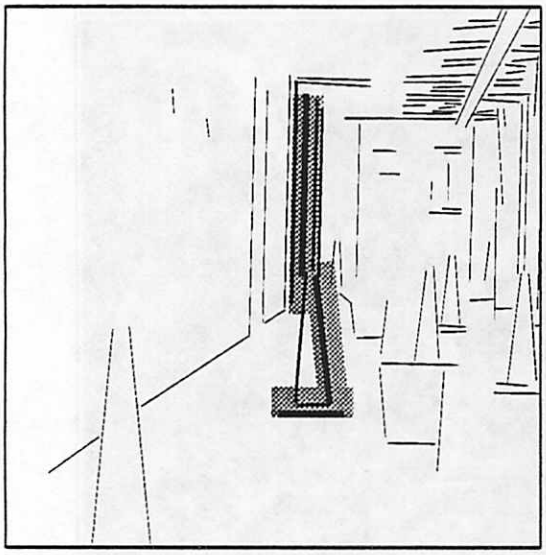
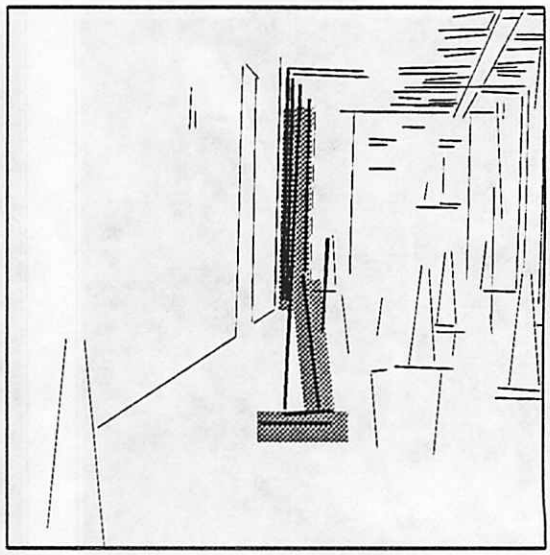


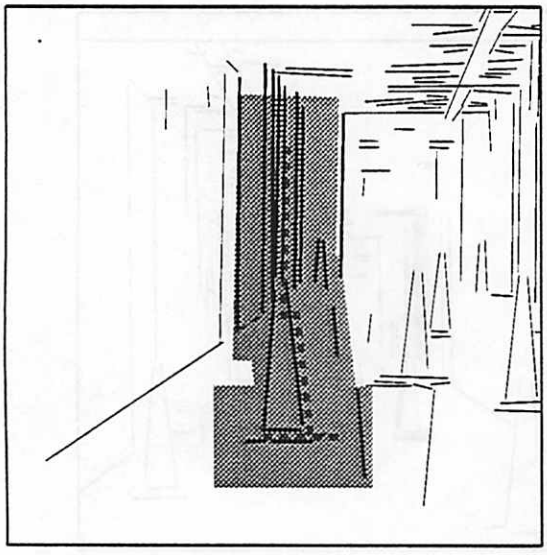
Figure 9: **Non–trackability of a non–shallow triple.** Two lines on a cone and one of the doorway lines in the background. a), b): First and the last image frames with the triple highlighted; c), d): Highlighted shallow triple overlaid on lines in frames 1 and 2. The correspondence in frame 2 was obtained using flow–based prediction in frame 1. (*contd. next page*)



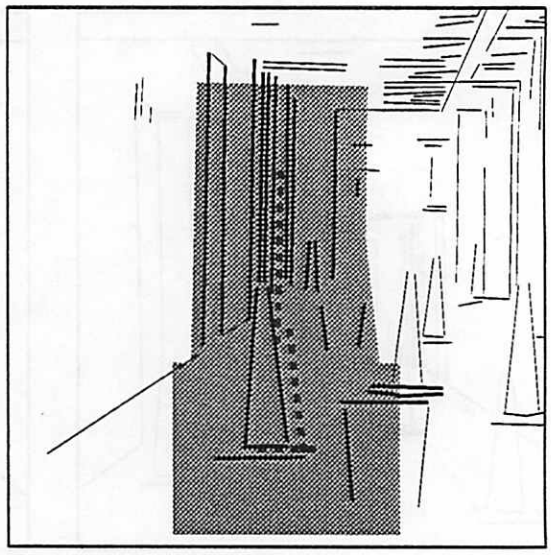
e) Frame 3



f) Frame 4



g) Frame 5



h) Frame 6

Figure 9: (*contd.*) **Non–trackability of a non–shallow triple.** The shaded areas are the search windows with the central spine being the position of predicted lines. The thinnest lines are all the lines in the background in the region of interest, medium thickness lines are the candidate matches, and the boldest lines are the matches found. f)–h): No match found. The prediction is shown as dotted lines in frames 5 and 6.

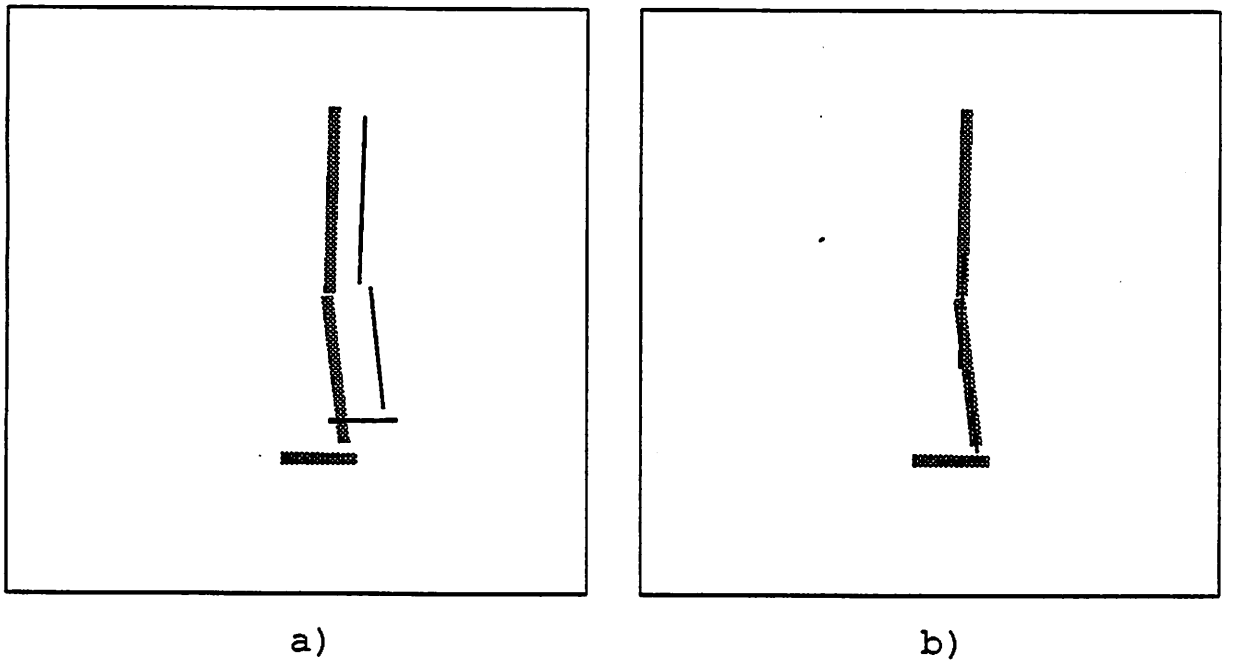


Figure 10: **Longitudinal error in affine projection of a non–shallow structure.** a) Frame 1 triple in bold and Frame 2 shaded. b) Frame 2 triple shaded and the affine projected Frame 1 triple bold.

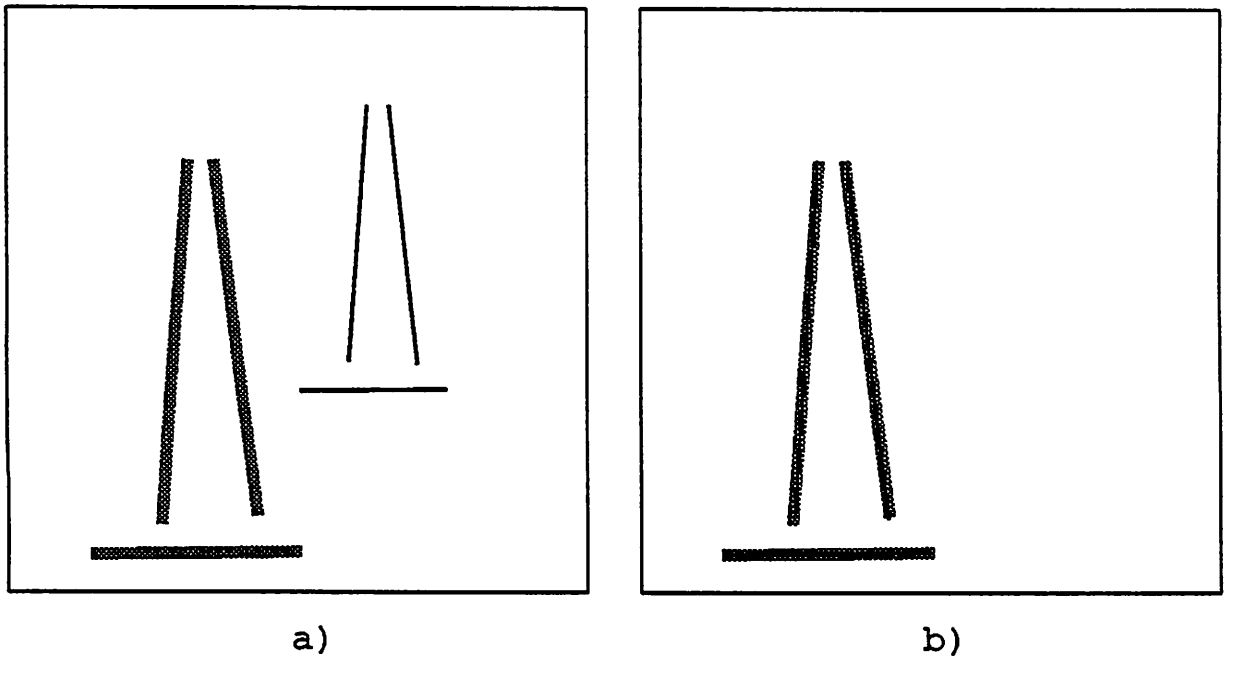


Figure 11: **Longitudinal error in affine projection of a shallow structure.** a) Frame 1 triple in bold and Frame 2 shaded. b) Frame 2 triple shaded and the affine projected Frame 1 triple bold.

(the central spine of the vertical shaded region in the figure) lies almost along the position of the corresponding line in the incorrect triple. A matching algorithm based on individual line matches would have matched to the incorrect data line, but because of covariance based aggregate matching in the algorithm, the tracking system matches to the correct triple. Between frames 6 and 7 (panels h and i), there is a change in the motion; it is as if the robot started going up a gently sloping "hill". Consequently, the prediction and the data move in opposite directions. Although no match is found in frame 7, the prediction persists with expanded windows and variances and the model is reacquired in frame 8.

Finally, for the *room-seq-1*, we demonstrate the algorithm on an independently moving object with occlusions. Figure 14 shows an object constructed from Lego blocks being tracked as it goes behind another surface (in frame 5) and re–emerges (in frame 8) as the camera moves towards it (see also Figure 6). Note that the non–uniformity in motion mentioned above, between frames 6 and 7, further complicates the tracking because during these frames the object remains hidden and the error in prediction increases dramatically. However, the object is still reacquired when it re–emerges in frame 8 (panel j). This example serves as a demonstration of the algorithm's potential use in sequences containing both camera motion and independent object motions.

Note that the mechanisms of model persistence and model uncertainties have been demonstrated to successfully handle all the three types of tracking failures — line grouping errors, motion discontinuity and occlusions. The related issue of computational complexity of matching and the allowable limits on model persistence were discussed in Section 7.1.

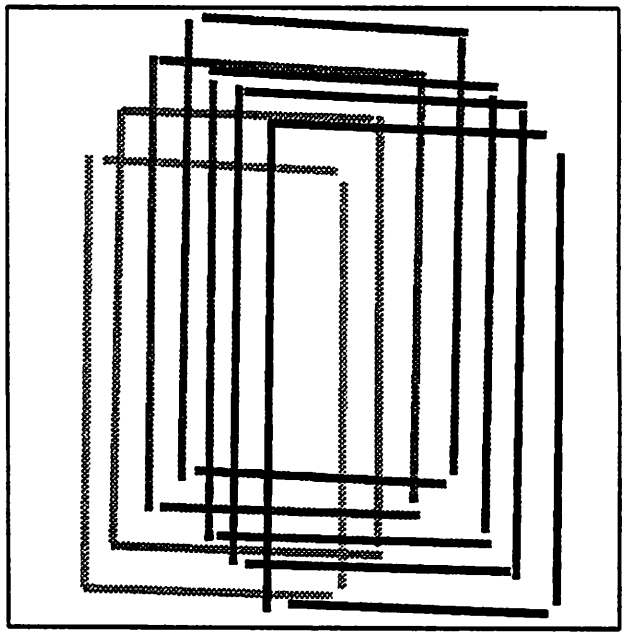


Figure 12: **Motion of a sample structure in the *room-seq-1*.** Image motion of the structure from frames 3 to 9. Frame 3 lines are in the lightest shade and frame 9 in the darkest. A motion discontinuity is shown by the change in direction after frame 6.

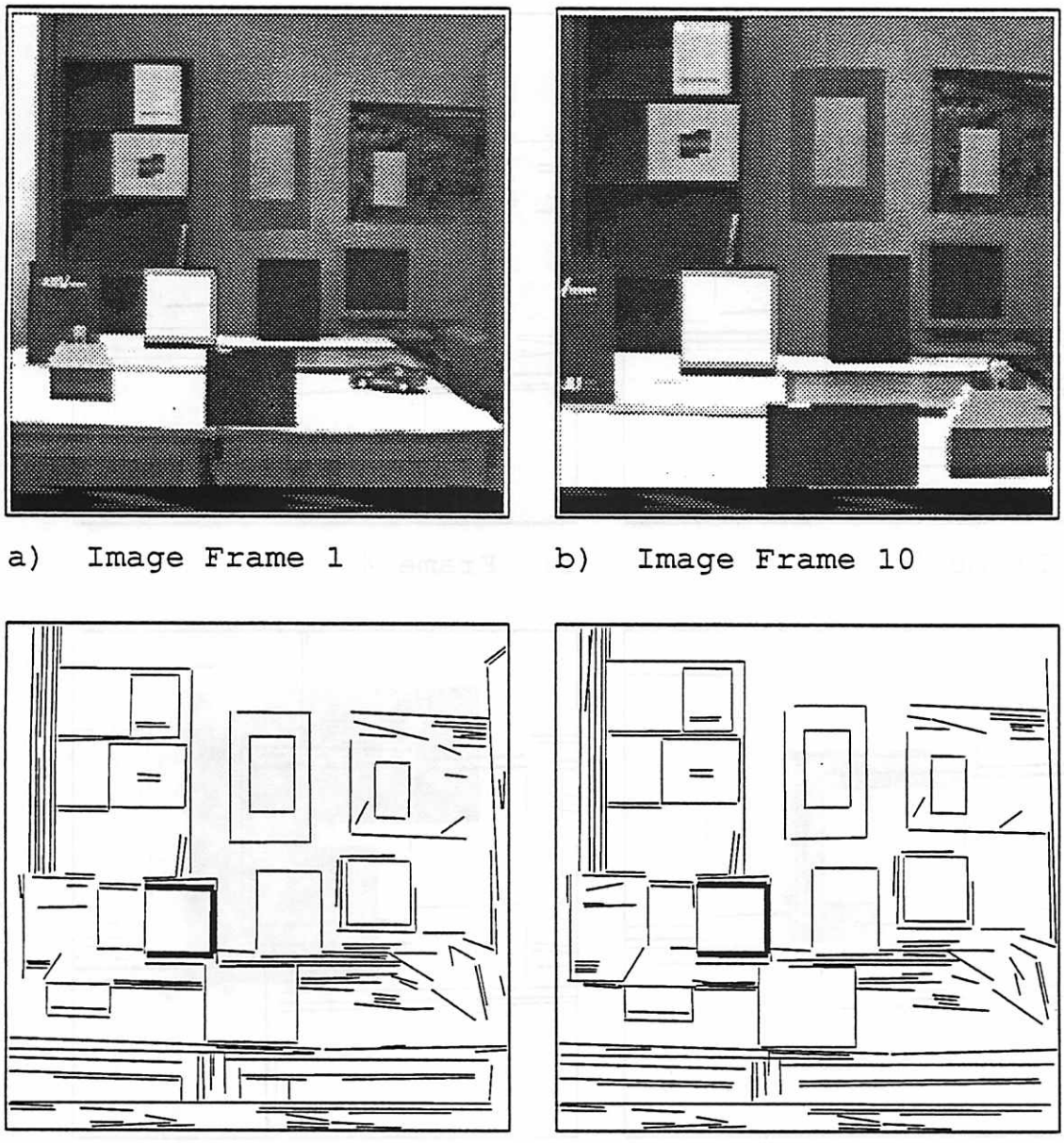
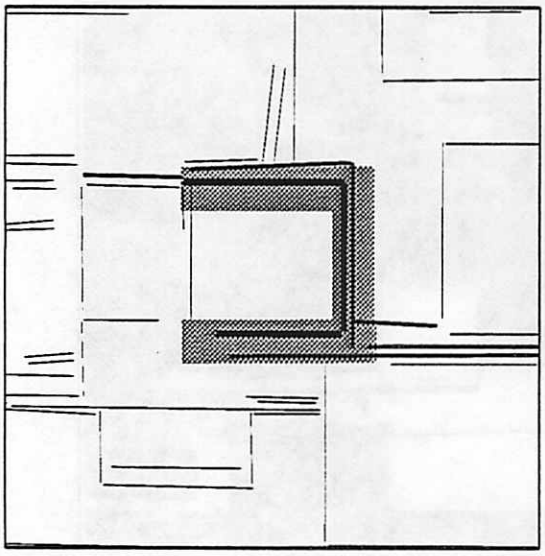
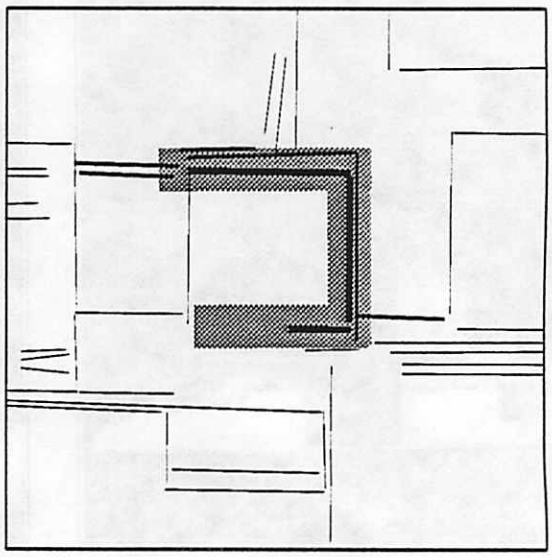


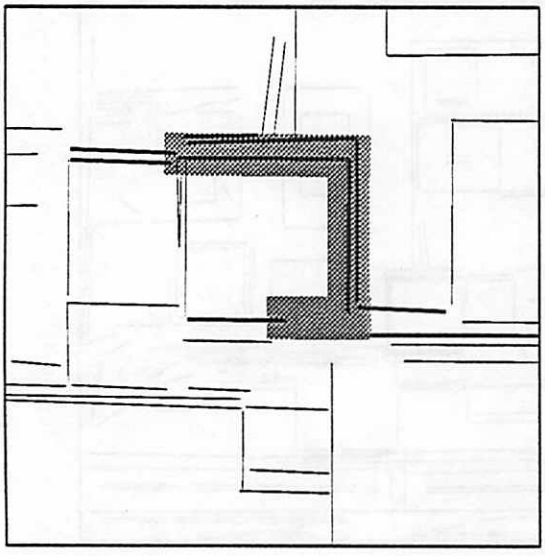
Figure 13: **Tracking of a shallow triple in the** *room-seq-1*. Shown for ten frames. a), b): First and the last image frames with the triple highlighted; c), d): Highlighted shallow triple overlaid on lines in frames 1 and 2. The correspondence in frame 2 was obtained using flow–based prediction in frame 1. (*contd. next page*)



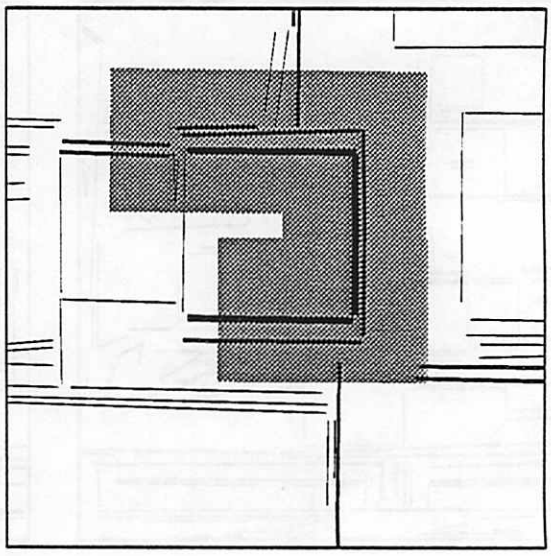
e) Frame 3



f) Frame 4



g) Frame 5



h) Frame 6

Figure 13: (*contd.*) **Tracking of a shallow triple over ten frames in the *room-seq-1*.** The shaded areas are the search windows with the central spine being the position of predicted lines. The thinnest lines are all the lines in the background in the region of interest, medium thickness lines are the candidate matches, and the boldest lines are the matches found. g) No match found due to line breaking. h) Recovery from line break in frame 5. (*contd. next page*)

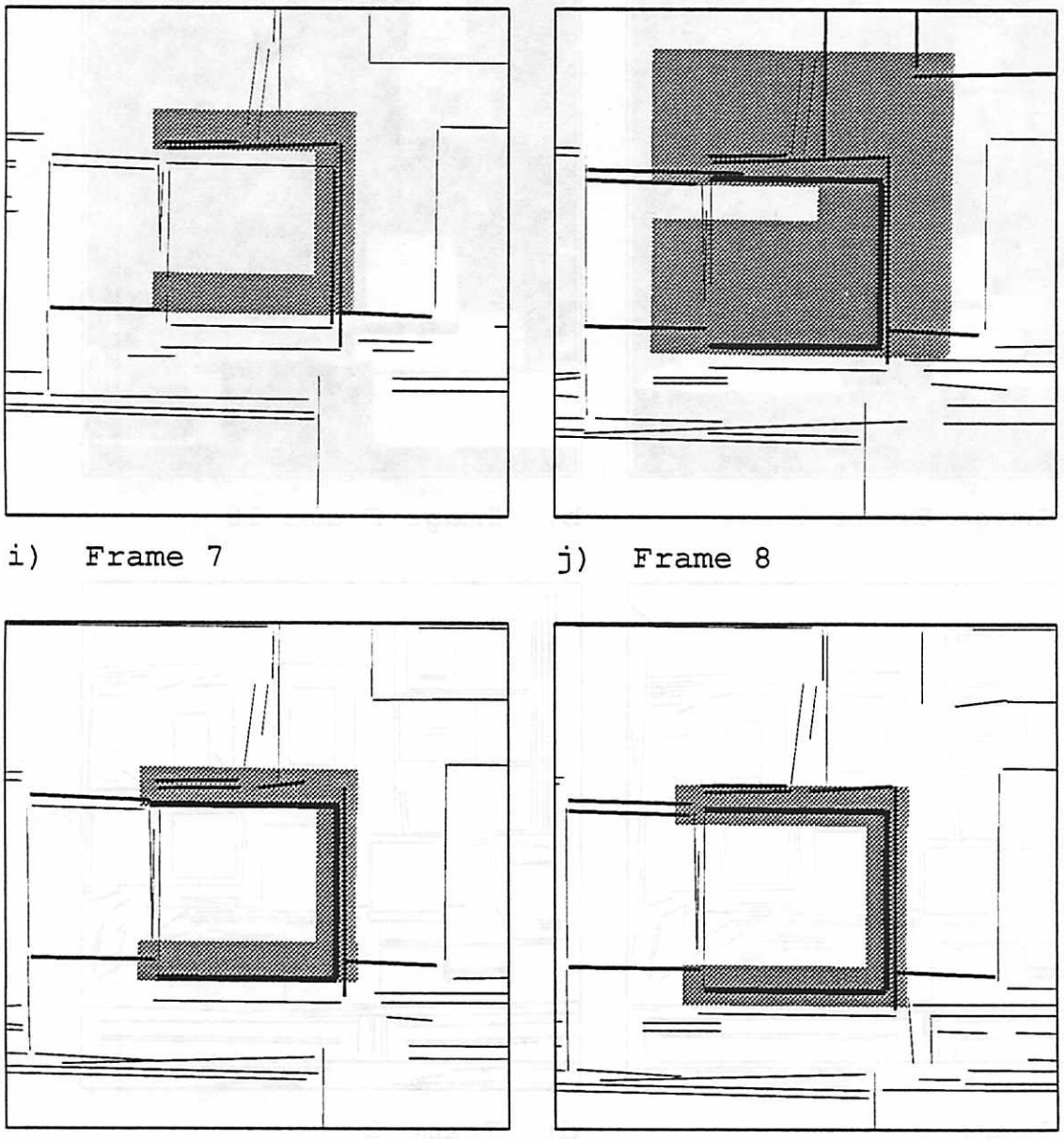


Figure 13: (*contd.*) **Tracking of a shallow triple over ten frames in the *room–seq–1*.** i) No match found due to motion discontinuity. j) Recovery from motion discontinuity between frames 6 and 7.

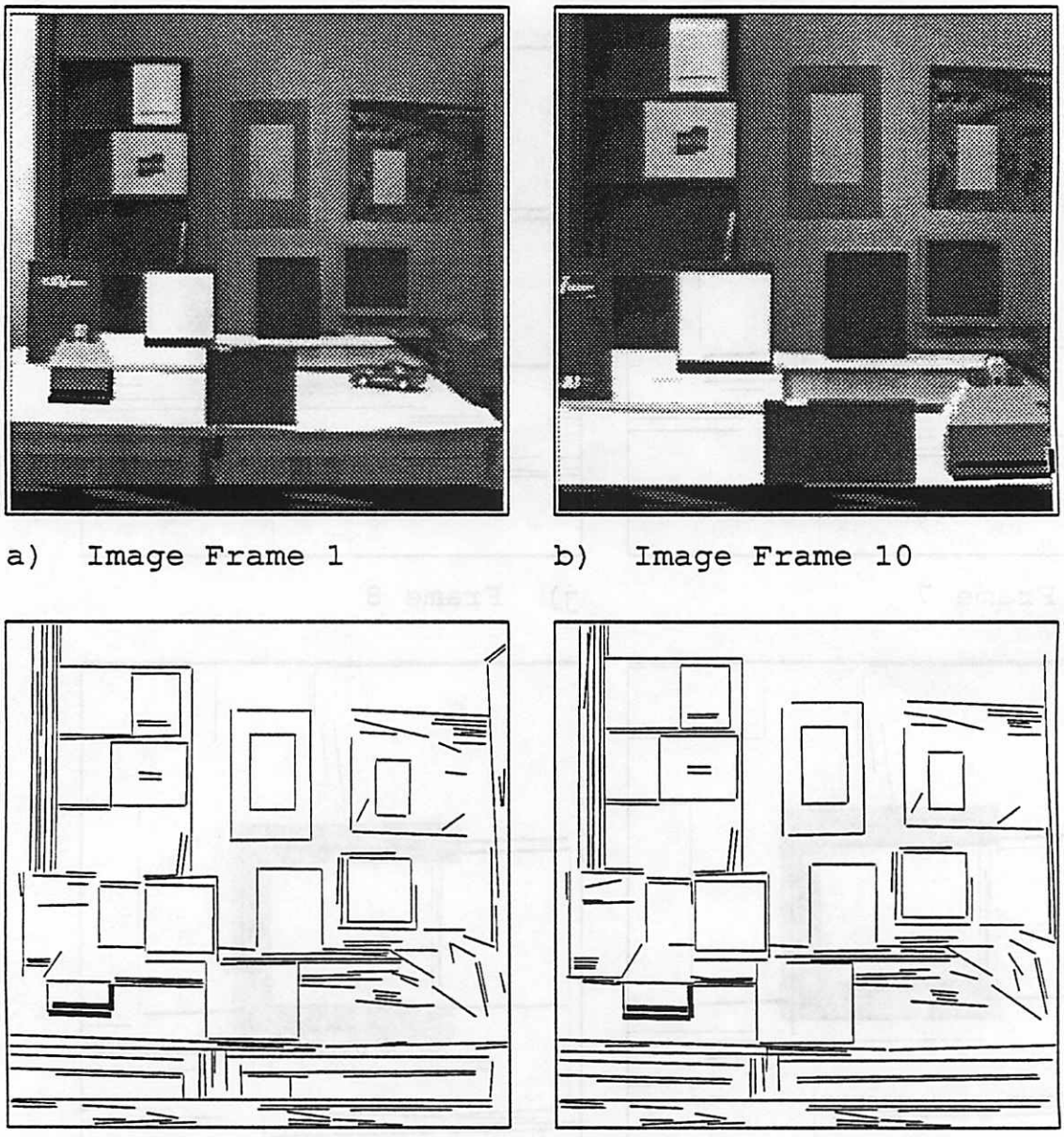
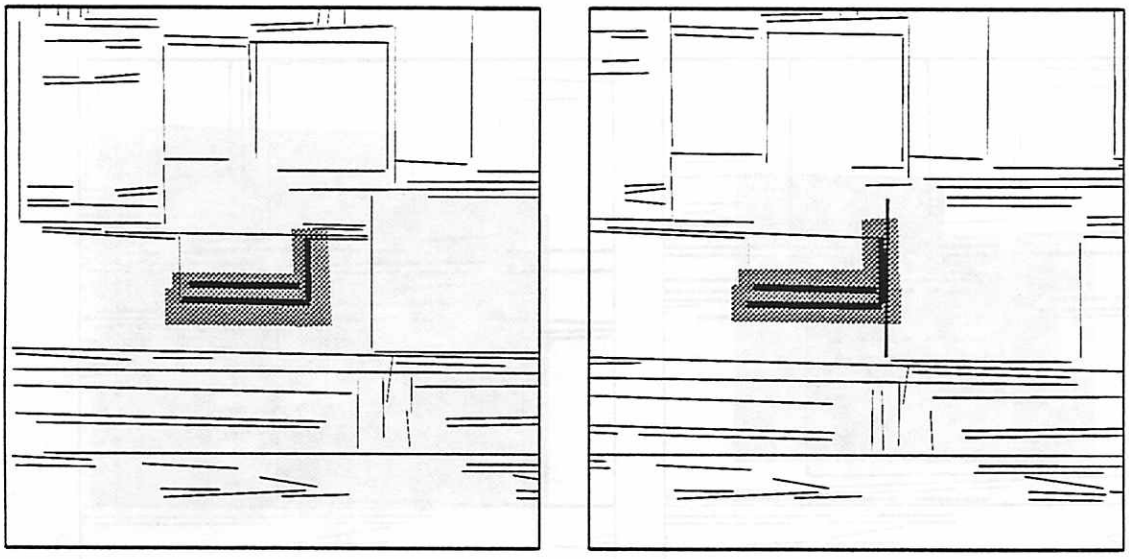
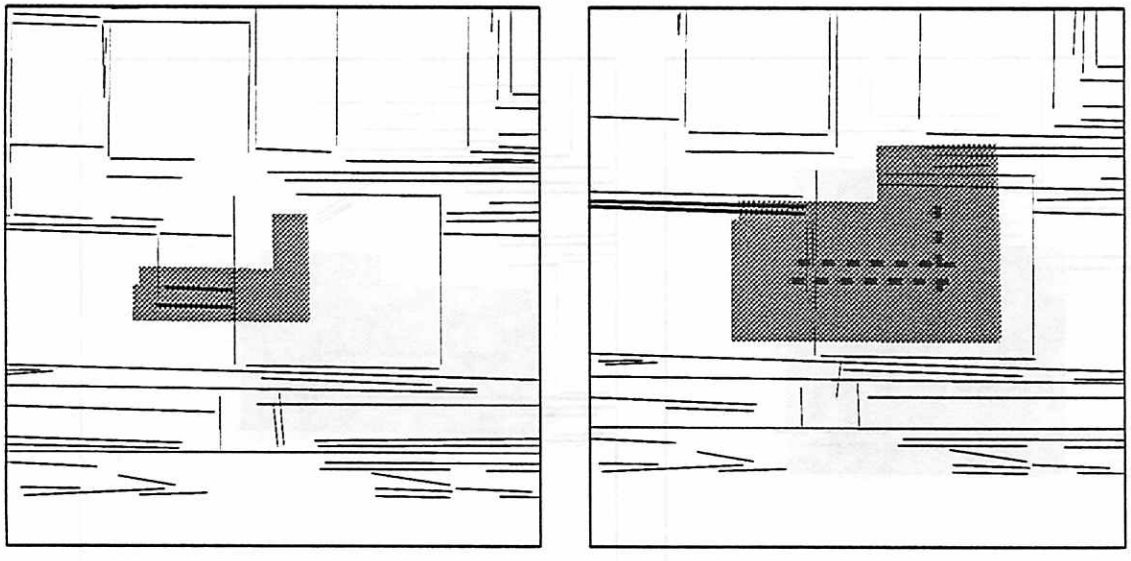


Figure 14: **Tracking of an independently moving object.** Tracking shown over ten frames in the *room-seq-1* with camera motion also present. a), b): First and last image frames; c), d): Highlighted shallow triple overlaid on lines in frames 1 and 2. The correspondence in frame 2 was obtained using flow-based prediction from frame 1. (*contd. next page*)

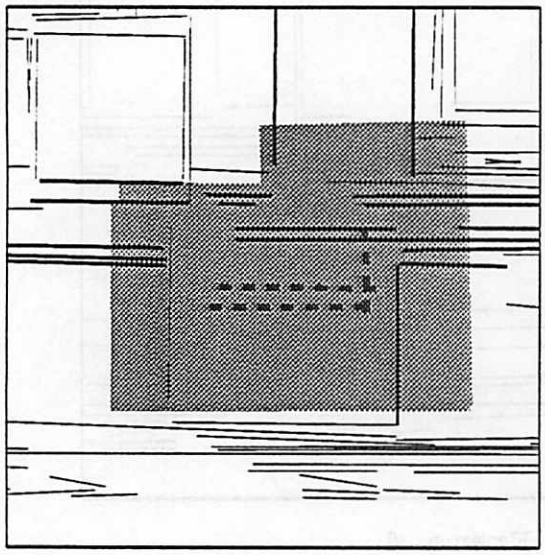


e) Frame 3 f) Frame 4

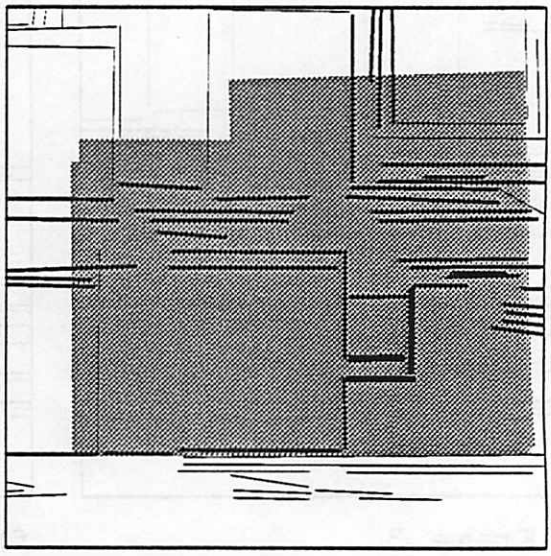


g) Frame 5 h) Frame 6

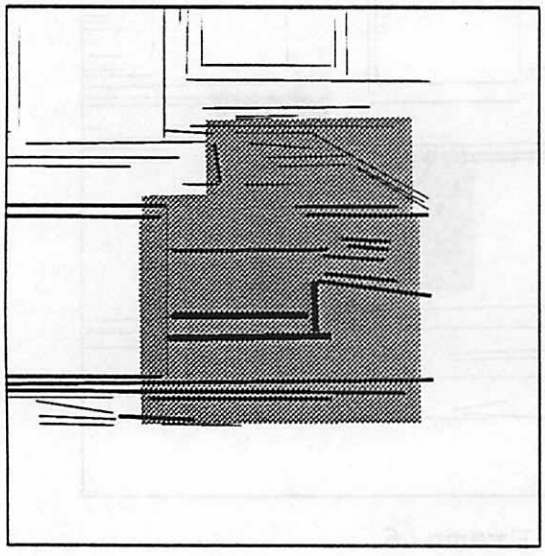
Figure 14: (*contd.*) **Tracking of an independently moving object.** The shaded areas are the search windows with the central spine being the position of predicted lines. The thinnest lines are all the lines in the background in the region of interest, medium thickness lines are the candidate matches, and the boldest lines are the matches found. e), f) Matching in frames 3 and 4. g), h) No match found due to occlusion. (*contd. next page*)



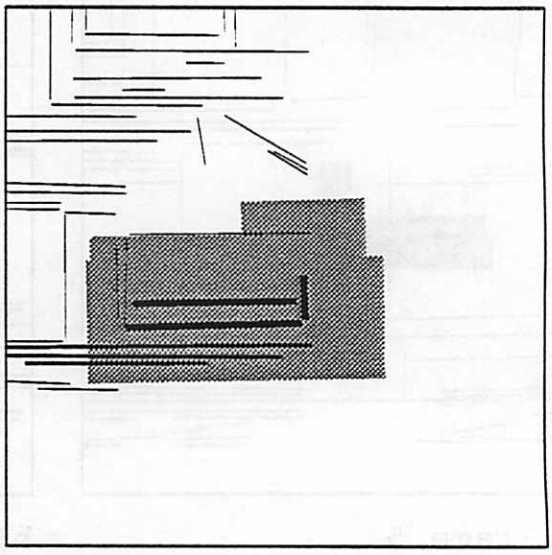
i) Frame 7



j) Frame 8



k) Frame 9



l) Frame 10

Figure 14: (*contd.*) **Tracking of an independently moving object.** i) No match found due to occlusion. j) Reacquisition of the object after occlusion in frames 5, 6 and 7.

8.2 Segmentation and Reconstruction Results

The algorithm in Section 7.2 was applied to the *cones–seq* and the *room–seq–1* to identify the shallow structures in the scene. Line triples were automatically selected to hypothesize aggregate structures. Each of these was tested for affine trackability resulting in its labeling as a shallow or a non–shallow structure. Figures 15 and 16 show the structures identified as shallow by the algorithm in the two sequences. In the *cones–seq* and the *room–seq–1*, 121 and 79 triples were found out of a total number of 167 and 180 lines, respectively. This supports our hypothesis in Section 7.2 that the order of the number of triples found using pruning by proximity is typically closer to linear than cubic in the number of total lines.

In the *cones–seq*, the two cones in the center of the image behind the trash can are merged together as a single shallow structure. This is because a) they are close to the FOE, and b) are far enough away so that their image motion is small.

The depth of some salient structures was measured with a tape measure. Tables 1 and 2 show a comparison of this ground truth with the computed depths for the *cones–seq* and the *room–seq–1*, respectively. The objects referred to in the tables are labelled in Figures 17 and 18. The average percentage depth errors for the *cones–seq* and *room–seq–1* are 3.8% and 3.4%, respectively.

Now we present results of depth computation using the four–parameter affine description for planar objects in a scene which are at a variety of slant angles. The results are illustrated on an image sequence, called the *comp–seq*. Two image frames of this sequence are shown in Figure 19. The approximate translation–in–depth between consecutive frames is 1.4 feet. The depths of some salient structures in the scene were measured from the camera in its position in frame 1. Recall that the affine transformation reconstructs a shallow structure as a fronto–parallel plane. So, for structures that have a large slant, the ground truth depths are the average depths. Figure 20 shows some labelled objects and Table 3 shows the measured and computed depths. The depths are computed over six frames of the sequence. The average absolute percentage error is 2.3%. These results suggest that when rotations are small, the fronto–parallel approximation for highly slanted shallow structures can also be computed robustly by the four–parameter affine approximation.

9 Conclusions

In this paper, we have presented a framework for the integration of spatial constraints on generic object structure and temporal constraints on smooth motion to achieve a semantically useful description of a scene from a sequence of images. A motivation for characterizing many objects as shallow in man–made environments is presented. The motion of shallow structures in the image plane can be described by an affine transformation. Instead of clustering image features, observed over two frames, into an object hypothesis that is consistent with a shallow structure interpretation, we use the temporal evolution of a hypothesized structure to verify its consistency within the constraints of a shallow structure. Temporal evolution is characterized by the trackability

of a structure under the affine constraint. Thus, a scene can be divided into shallow and non–shallow structures through the use of tracking as a verification process.

Tracking and dynamic estimation of the affine parameters of a shallow structure also lead to a reconstruction of the structure from changing scale (depth from looming). The reconstruction of the shallow structure is as a fronto–parallel plane placed at a depth that is equal to the estimated depth. That is, the representation of shallow structures is in terms of cardboard cut–outs facing the camera for each shallow structure. An important advantage of this method is that structure reconstruction is achieved without the intermediate step of explicitly computing the 3D motion parameters (rotation and translation) between successive frames. The reconstructed structure is only an approximation, however, to the average depth of the corresponding true environmental structure. Nevertheless, the robustness of depth of the approximate structure representation might prove to be useful for tasks like obstacle avoidance, where the exact shape of an object may not be of consequence so long as collisions with it can be avoided.

We have also shown that tracking of a structure, which is formed as an aggregate of image features, is resilient to many of the common sources of errors in feature extraction and modeling of motion. Specifically, it is shown that for shallow structures, predictions of their image motion can be based on 3D constraints and not on heuristics about the image motion of features. This leads to a simple method of handling uncertainties in the modeling of 3D motion. Furthermore, matching predictions to newly acquired data of a model as a whole is more reliable than isolated feature matching.

The tracking, identification and reconstruction of shallow structures are demonstrated on real image sequences. Illustrations of how the system handles errors in feature extraction, and motion discontinuity are presented. Furthermore, it is also shown how the algorithm can track independently moving objects imaged with a moving camera. Tracking errors due to feature extraction errors, motion discontinuity and occlusions are handled in a single framework of covariance based prediction and matching.

Figure 15: **Shallow structures identified in the *cones-seq*.**

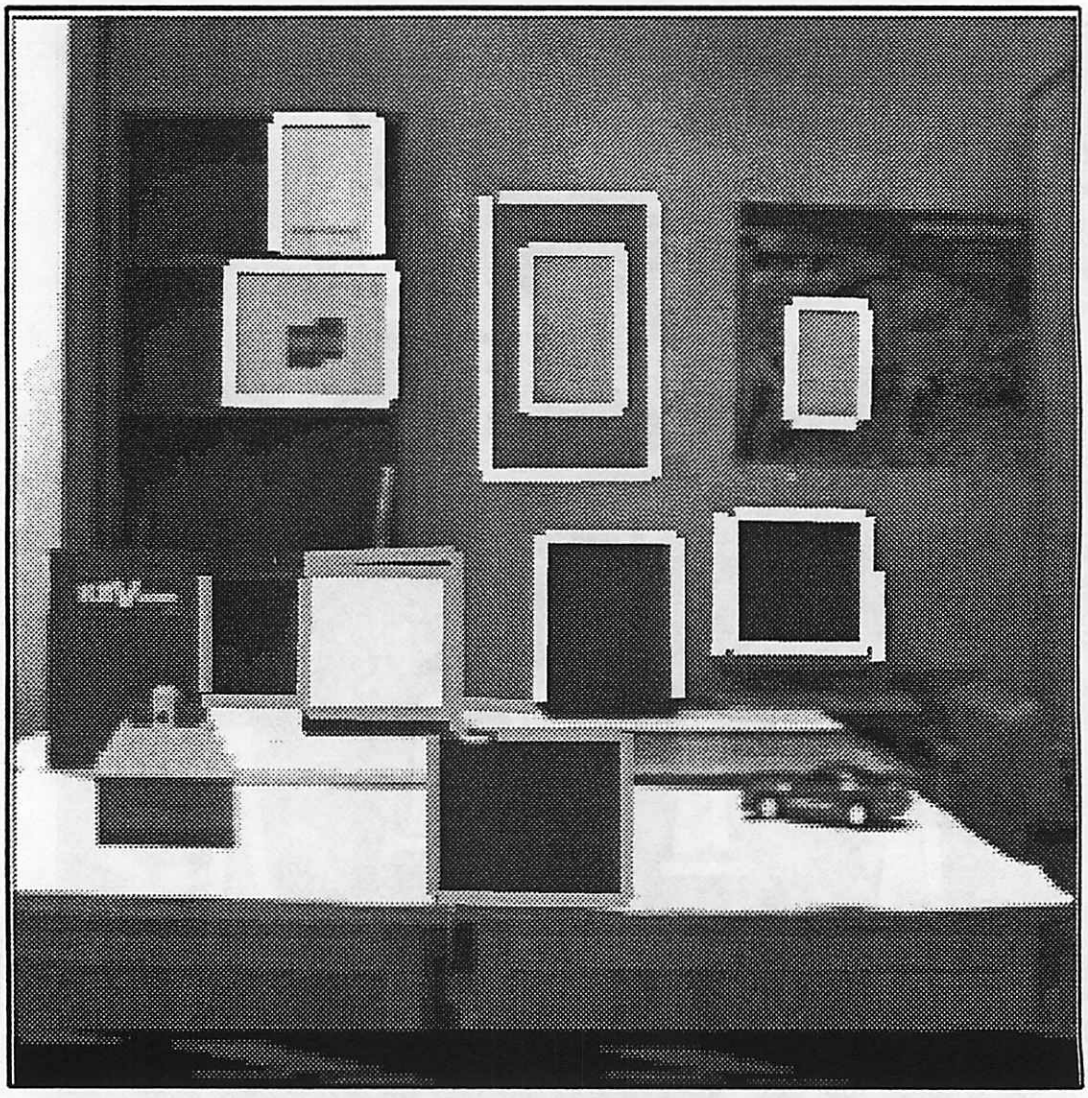


Figure 16: **Shallow structures identified in the *room–seq–1*.** Shown in thick white and light gray outlines.

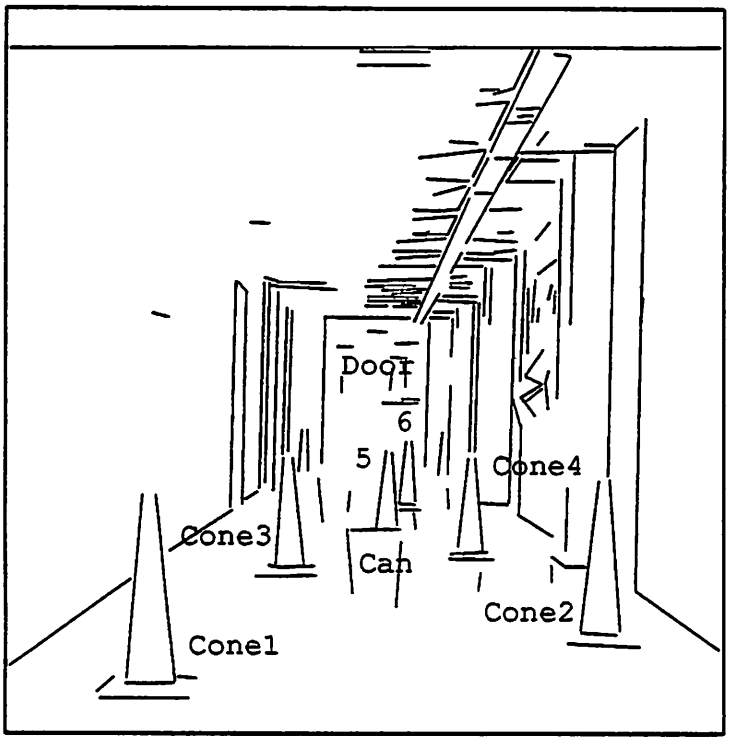


Figure 17: **Labelled objects in the** *cones-seq*. Shown in frame 1 lines.

Table 1: **Depth results for the** *cones-seq*. Computed vs. Measured Depths of some Objects in the *cones-seq* (*in feet*).

Object	Meas. Z	Comp. Z	Error (%)
Cone 1	20.0	20.24	1.2
Cone 2	25.0	25.91	3.6
Can	30.0	31.69	5.6
Cone 3	35.0	36.77	5.1
Cone 4	40.0	40.72	1.8
Cone 5	45.0	47.80	6.2
Cone 6	60.0	63.84	6.4
Door	87.1	87.70	0.7
Average Abs. Error			3.8%

Table 2: **Depth results for the *room-seq-1*.** Computed vs. Measured Depths of some Objects in the *room-seq-1* (*in feet*).

Object	Meas. Z	Comp. Z	Error (%)
1	8.3	8.02	-3.4
2	13.4	12.48	6.9
3	14.57	14.6	0.2
4	18.98	18.78	-1.1
5	11.57	11.78	1.8
6	19.04	18.01	-5.4
7	20.35	19.16	-5.8
8	20.35	19.84	-2.5
Average Abs. Error			3.4%

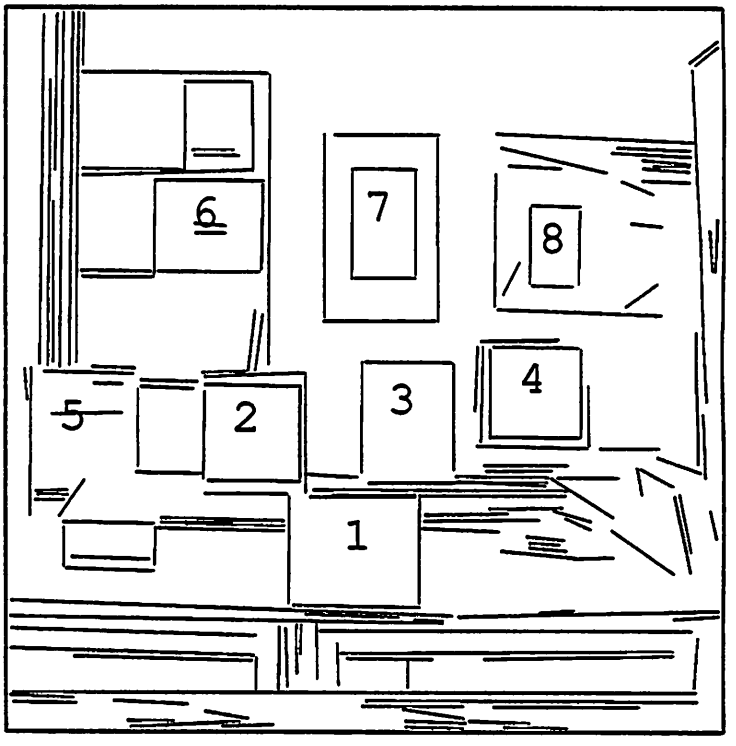
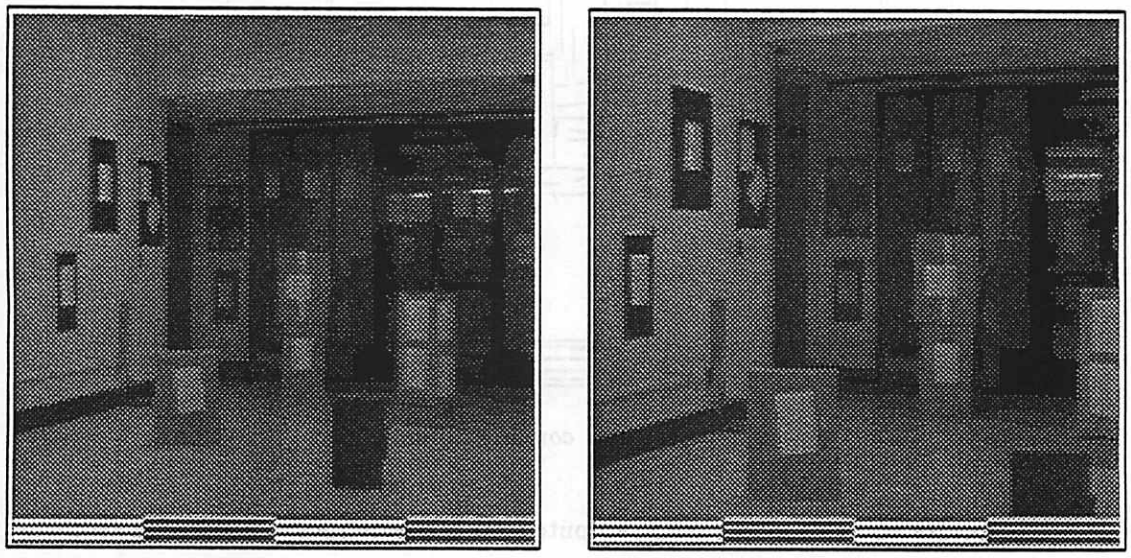


Figure 18: **Labelled objects in the *room-seq-1*.** Shown in frame 1 lines.



a) Image Frame 1 b) Image Frame 6

Figure 19: **Two image frames of the** *comp-seq*. Frames 1 and 6.

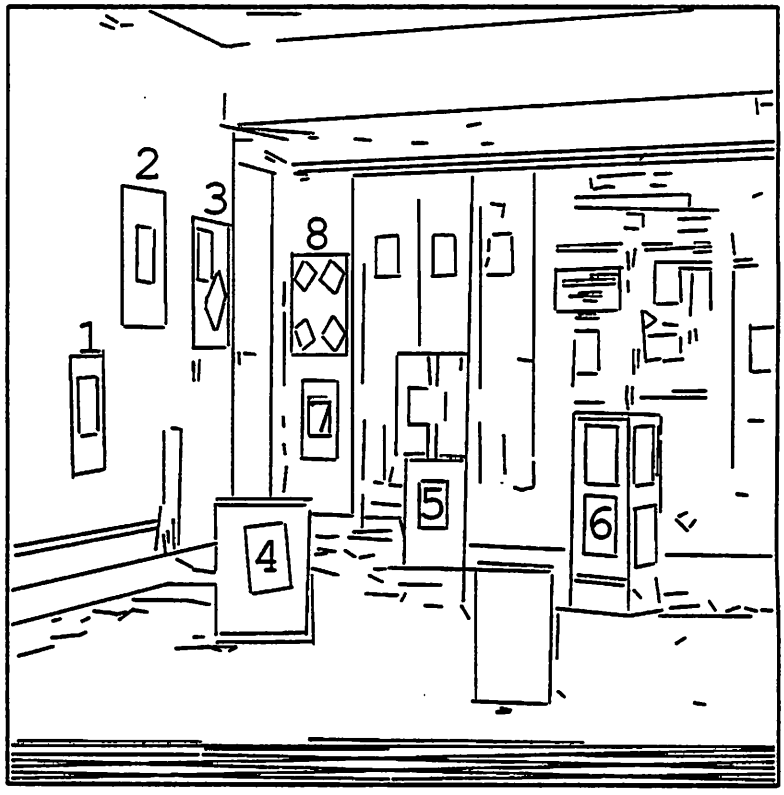


Figure 20: **Labelled objects in the *comp-seq*.** Shown in frame 1 lines.

Table 3: **Depth results for the *comp-seq*.** Computed vs. Measured Depths of some Objects (*in feet*).

Object	Meas. Z	Comp. Z	Error (%)
1	29.28	29.96	2.32
2	31.23	31.04	-0.61
3	33.23	34.37	3.13
4	25.68	25.94	1.01
5	35.83	34.24	-4.44
6	28.18	28.29	0.39
7	43.23	44.88	3.82
8	43.23	42.05	-2.73
Average Abs. Error			2.3%

References

[1] G. Adiv. Determining 3D motion and structure from optical flows generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(4):384–401, 1985.

[2] G. Adiv. Inherent ambiguities in recovering 3D information from a noisy flow field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):477–489, 1989.

[3] Gilad Adiv and Edward Riseman. Recovery of 3D motion and structure from image correspondences using a directional confidence measure. Technical Report COINS TR 88–105, University of Massachusetts

at Amherst, MA, 1988.

[4] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3):283–310, 1989.

[5] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.

[6] M. Boldt, R. Weiss, and E. Riseman. Token–based extraction of straight lines. *IEEE Transactions on Systems Man and Cybernetics*, 19(6):1581–1594, 1989.

[7] J. Brolio, B. Draper, J. R. Beveridge, and A. Hanson. ISR: A database for symbolic processing in computer vision. *IEEE Computer*, 22(12):22–30, 1989.

[8] J. L. Crowley, P. Stelmaszyk, and C. Discours. Measuring image flow by tracking edge–lines. In *Proc. 2nd Intl. Conf. on Computer Vision*, pages 658–664, 1988.

[9] R. Deriche and O. Faugeras. Tracking line segments. In *Proc. 1st European Conference on Computer Vision*, pages 259–268, 1990.

[10] O. D. Faugeras and F. Lustman. Let us suppose the world is piece-wise planar. In *Proc. The Third International Symposium on Robotics Research*, 1987.

[11] O. D. Faugeras, F. Lustman, and G. Toscani. Motion and structure from motion from point and line matches. In *IEEE First International Conference on Computer Vision*, pages 25–34, 1987.

[12] Arthur Gelb. *Applied Optimal Estimation*. The MIT Press, Cambridge, MA, 1986.

[13] B. K. P. Horn. Relative orientation. *International Journal of Computer Vision*, 4(1):59–78, 1990.

[14] Daniel P. Huttenlocher and Shimon Ullman. Recognizing solid objects by alignment. In *Proc. Computer Vision and Pattern Recognition Conference*, pages 1114–1124, 1989.

[15] T. Kanade and J. R. Kender. Mapping image properties into shape constraints: Skewed symmetry, affine–transformable patterns, and the shape–from–texture paradigm. In J. Beck et al, editor, *Human and Machine Vision*, pages 237–257. Academic Press, NY, 1983.

[16] Y. Lamdan, J. T. Schwartz, and H. J. Wolfson. Object recognition by affine invariant matching. In *Proc. Computer Vision and Pattern Recognition Conference*, pages 335–344, 1988.

[17] P. C. Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Science, India*, 12:49–55, 1936.

[18] Raman K. Mehra. On the identification of variances and adaptive Kalman filtering. *IEEE Transactions on Automatic Control*, 15(2):175–184, 1970.

[19] R. C. Nelson and J. Aloimonos. Towards qualitative vision: Using flow field divergence for obstacle avoidance in visual navigation. In *Proc. 2nd Intl. Conf. on Computer Vision*, pages 188–196, 1988.

[20] Gilbert Strang. *Introduction to Applied Mathematics*. Wellesley–Cambridge Press, MA, 1986.

[21] D. Thompson and J. Mundy. Three–dimensional model matching from an unconstrained viewpoint. In *Proc. IEEE Conf. on Robotics and Automation*, pages 208–220, 1987.

[22] R. Y. Tsai and T. S. Huang. Uniqueness and estimation of 3D motion parameters and surface structures of rigid objects. In Whitman Richards and Shimon Ullman, editors, *Image Understanding 1984*, pages 135–171. Ablex Corporation, NJ, 1984.

[23] J. Weng, T. S. Huang, and N. Ahuja. Motion and structure from two perspective views: Algorithms, error analysis, and error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):451–476, 1989.

[24] L. R. Williams and A. R. Hanson. Translating optical flow into token matches and depth from looming. In *Proc. 2nd Intl. Conf. on Computer Vision*, pages 441–448, 1988.