Multi-Frame Estimation of Planar Motion

Lihi Zelnik-Manor and Michal Irani, *Member*, *IEEE*

Abstract—Traditional plane alignment techniques are typically performed between *pairs* of frames. In this paper, we present a method for extending existing two-frame planar motion estimation techniques into a *simultaneous multi-frame* estimation, by exploiting multi-frame subspace constraints of planar surfaces. The paper has three main contributions: 1) we show that when the camera calibration does not change, the collection of all parametric image motions of a planar surface in the scene across multiple frames is embedded in a low dimensional linear subspace; 2) we show that the *relative* image motion of multiple planar surfaces across multiple frames is embedded in a yet *lower* dimensional linear subspace, even with varying camera calibration; and 3) we show how these multi-frame constraints can be incorporated into simultaneous multi-frame estimation of planar motion, without explicitly recovering any 3D information, or camera calibration. The resulting multi-frame estimation process is more constrained than the individual two-frame estimations, leading to more accurate alignment, even when applied to small image regions.

Index Terms—Motion estimation, plane alignment, multi-frame analysis, gradient-based methods.

1 Introduction

Plane stabilization ("2D parametric alignment") is essential for many video-related applications: It is used for video stabilization and visualization, for 3D analysis (e.g., using the Plane+Parallax approach [10], [14]), for moving object detection, mosaicing, etc.

Many techniques have been proposed for estimating the 2D parametric motion of a planar surface between *two* frames. Some examples are [12], [4], [21], [3], [13]. While these techniques are very robust and perform well when the planar surface captures a large image region, they tend to be highly inaccurate when applied to small image regions. Moreover, errors can accumulate over a sequence of frames when the motion estimation is performed between *successive* pairs of frames (as is often done in mosaic construction).

An elegant approach was presented in [17] for automatically estimating an optimal (usually virtual) reference frame for a sequence of images with the corresponding motion parameters that relate each frame to the virtual reference frame. This overcomes the problem associated with error accumulation in sequential frame alignment. However, the alignment method used for estimating the motion between the virtual reference frame and all other frames remains a *two*-frame alignment method.

Other multi-frame estimation techniques (e.g., [5], [7], [11], [12]) *incrementally* apply a two-frame motion estimation technique, while relying on temporal smoothness of the motion. This assumption is a heuristic, which is violated when the camera motion changes abruptly. Sequential two-frame parametric alignment methods do not exploit the fact that all frames imaging the same planar surface share the same plane geometry (but not necessarily the same camera motion).

In this paper, we present a method for extending traditional two-frame planar-motion estimation techniques into a *simultaneous multi-frame* estimation method, by exploiting multi-frame linear subspace constraints of planar motions (Section 4). These multi-frame constraints are geometrically meaningful and do not rely on heuristics such as temporal smoothness. However, when such smoothness does exist in the video data, our method can detect it and take advantage of it. The use of linear subspace constraints, for motion analysis, has been introduced by Tomasi and Kanade [20]. They used these constraints for factoring $2D$ correspondences into $3D$ motion and shape information. In contrast, here we use linear subspace constraints for *constraining* our $2D$ planar motion estimation process and *not* for factoring out any $3D$ information. This results in a multi-frame estimation technique which is more constrained than the individual two-frame estimation processes, leading to more accurate alignment, even when applied to small image regions. Furthermore, multi-frame rigidity constraints relating multiple planar surfaces are applied to further enhance parametric motion estimation in scenes with *multiple* planar surfaces (Section 5).

2 Basic Model and Notations

The *instantaneous* image motion of a $3D$ planar surface π, between two image frames can be expressed as a $2D$ quadratic transformation (see Appendix A).

$$\vec{u}(\vec{x};\vec{p}) = X(\vec{x})\vec{p}, \tag{1}$$

where $X(\vec{x})$ is a matrix which depends only on the pixel coordinates $(\vec{x}) = (x, y)$:

$$X(\vec{x}) = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 & x^2 & xy \\ 0 & 0 & 0 & 1 & x & y & xy & y^2 \end{bmatrix}$$

and $\vec{p} = (p_1, p_2, \ldots, p_8)^T$ is a parameter vector:

• *The authors are with the Department of Computer Science and Applied Mathematics, The Weizmann Institute of Science, Rehovot, 76100, Israel. E-mail: {lihi, irani}@wisdom.weizmann.ac.il.*

Manuscript received 15 June 1999; revised 24 May 2000; accepted 1 June 2000.
Recommended for acceptance by M.J. Black.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 110057.

$$\begin{array}{ll} p_1 = f'(\gamma t_X + \Omega_Y) & p_2 = \frac{f'}{f}(1 + \alpha t_X) - \gamma t_Z - 1 \\ p_3 = -\frac{f'}{f}(\Omega_Z - \beta t_X) & p_4 = f'(\gamma t_Y - \Omega_X) \\ p_5 = \frac{f'}{f}(\Omega_Z + \alpha t_Y) & p_6 = \frac{f'}{f}(1 + \beta t_Y) - \gamma t_Z - 1 \\ p_7 = \frac{1}{f}(\Omega_Y - \alpha t_Z) & p_8 = -\frac{1}{f}(\Omega_X + \beta t_Z), \end{array} \tag{2}$$

where $\vec{n} = (\alpha, \beta, \gamma)^T$ is the normal of the plane π (i.e., $\vec{Q}^T\vec{n} = 1$, $\forall \vec{Q} \in \pi$), $\vec{\Omega} = (\Omega_X, \Omega_Y, \Omega_Z)^T$, and $\vec{t} = (t_X, t_Y, t_Z)^T$ are the camera rotation and translation, respectively, and f and f' are the camera focal lengths used for obtaining the two images.

The instantaneous motion model is valid when the camera rotation is small and the forward translation is small relative to the depth.

3 Two-Frame Parametric Alignment

In this paper, we extend the *direct* two-frame motion estimation approach of [4], [12] to multiple frames. To make the paper self-contained, we first briefly outline the basic two-frame technique below.

Two image frames (whose parametric image motion is being estimated) are referred to by the names "reference" image J and "inspection" image K. A Gaussian pyramid [1] is constructed for J and K and the motion parameters from J to K are estimated in a coarse-to-fine manner. Within each pyramid level, the sum of squared *linearized* differences (i.e., the linearized brightness constancy measure) is used as a match measure. This measure (Err) is minimized with respect to the unknown 2D motion parameters $\vec{p}$ of (1):

$$\begin{aligned} Err(\vec{p}) &= \sum_{(\vec{x})} \Big((K(\vec{x}) - J(\vec{x})) + \nabla J(\vec{x})^T \vec{u}(\vec{x}; \vec{p}) \Big)^2 \\ &= \sum_{(\vec{x})} \Big((K(\vec{x}) - J(\vec{x})) + \nabla J(\vec{x})^T X(\vec{x}) \vec{p} \Big)^2, \end{aligned} \tag{3}$$

where $\vec{u}(\vec{x}; \vec{p}) = X(\vec{x})\vec{p}$ is as defined in (1), $J(\vec{x})$ and $K(\vec{x})$ denote the brightness value of image J and K at pixel $\vec{x}$, respectively, and $\nabla J(\vec{x})$ denotes the spatial gradient of J at $\vec{x}$: $\nabla J(\vec{x}) = \left(\frac{\partial J}{\partial x}(\vec{x}), \frac{\partial J}{\partial y}(\vec{x})\right)^T$. The sum is computed over all the points within a region of interest (often the entire image). Deriving Err with respect to the unknown parameters $\vec{p}$ and setting to zero, yields eight linear equations in the eight unknowns:

$$\mathcal{C}\vec{p} = \vec{b}, \tag{4}$$

where $\mathcal{C}$ is an 8×8 matrix:

$$\mathcal{C} = \sum_{(\vec{x})} \Big[X(\vec{x})^T \nabla J(\vec{x}) \nabla J(\vec{x})^T X(\vec{x}) \Big], \tag{5}$$

and $\vec{b}$ is an 8×1 vector:

$$\vec{b} = \sum_{(\vec{x})} \Big[X(\vec{x})^T \nabla J(\vec{x}) (J(\vec{x}) - K(\vec{x})) \Big]. \tag{6}$$

This leads to the linear solution $\vec{p} = \mathcal{C}^{-1}\vec{b}$. Note that $\mathcal{C}$ and $\vec{b}$ are constructed of *measurable image quantities*, hence, the word *direct* estimation. This process does not require recovery of any $3D$ information.

To allow for large displacements $\vec{u}(\vec{x}; \vec{p})$, the estimation process of $\vec{p}$ is performed iteratively, within a coarse-to-fine computational framework. Multiscale analysis provides three main benefits: 1) larger misalignments can be handled, 2) the convergence rate is faster, and 3) it avoids getting trapped in local minima. These three benefits are discussed in detail in [4]. The iterative process starts at the coarsest resolution level and after a few iterations of parameter refinement (typically 5), the result is propagated to the next resolution level, where the process is repeated. For a detailed description of the iterative process, see Appendix D.

The two-frame method performs well when applied to large image regions (i.e., the region of interest over which the summation is performed). Figs. 1f and 1i show an example of applying this parametric alignment method. This is an airborne sequence taken from a large distance, hence, the camera induced motion can be described by a single $2D$ parametric transformation of (1). Fig. 1c was the reference image and Fig. 1b was the inspection image. Fig. 1e shows the amount of initial misalignment between the two input images. When the method is applied to the *entire image region*, it yields accurate alignment (at subpixel accuracy), as can be seen in Figs. 1f and 1i. However, once the same method is applied to a small image region (such as the rectangular region marked in Figs. 1g and 1j), its accuracy degrades significantly. The farther a pixel is from the region of analysis, the more misaligned it is. In the next section, we show how information from *multiple* frames (as opposed to two) can be used to increase accuracy of image alignment even when applied to small image regions.

4 Multiframe Parametric Alignment

In this section, we present a method for extending the *two*-frame technique reviewed in Section 3 into a *multi*-frame technique, which exploits *multi-frame constraints* on the image motion of a planar surface. In Section 4.1, we derive such a multi-frame constraint and in Section 4.2, we show how it can be incorporated into the $2D$ parametric estimation of planar motion, without requiring any recovery of $3D$ information, or camera calibration. In Section 4.3, we present the idea of frame reliability measure, to further enhance multi-frame motion estimation.

4.1 Single Plane Subspace Constraint

Let J be a reference frame, and let $K^1, \ldots, K^F$ be a sequence of F inspection frames imaging the *same* planar surface with the *same* focal length f. Let $\vec{p}^1, \ldots, \vec{p}^F$ be the corresponding quadratic parameter vectors of the planar motion (see (1)). The instantaneous motion model of (1) is a good approximation of the motion over *short video segments*, as the camera does not gain large motions in short periods of time. In some cases, such as airborne video, this approximation is good also for very long sequences. Choosing the reference frame as the *middle* frame extends the applicability of the model to twice as many frames.

We arrange $\vec{p}^1, \ldots, \vec{p}^F$ in an $8 \times F$ matrix P, where each column corresponds to one frame. From (2):

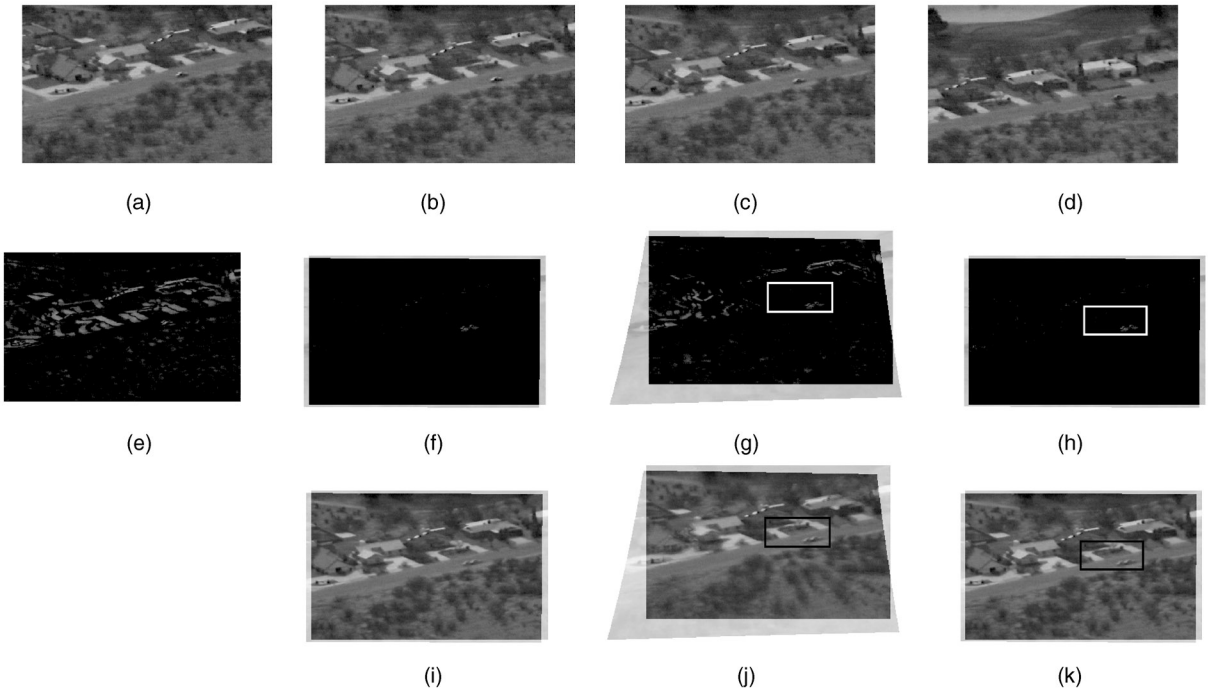

Fig. 1. Single Plane Motion Estimation. (a), (b), (c), and (d) are sample frames from a 17-frame airborne video clip. Apart from camera motion, there is also a small moving car. Image (c) was the reference frame. (e) shows the absolute differences between two input frames (b) and (c) which indicates initial misalignment. (f) and (i) display high quality alignment (at subpixel accuracy) from applying the two-frame technique to the entire image region. (f) shows absolute differences after alignment, while (i) shows the average of the two aligned images. Only the independently moving car is misaligned. (g) and (j) display poor alignment from applying the two-frame alignment to a small image region, marked by a rectangle. Although the rectangular region is well-aligned (apart from the moving car), large misalignments can be detected in image pixels which are distant from the analysis region. (h) and (k) show high quality alignment from applying the constrained multi-frame alignment to the same small rectangular region. It was applied simultaneously to all 17 frames. Even pixels distant from the analysis window appear well-aligned.

$$P = \left[\vec{p}^1 \cdots \vec{p}^F\right]_{8\times F} = S_{8\times 6}\begin{bmatrix} \vec{t}^1 & \cdots & \vec{t}^F \\ \vec{\Omega}^1 & & \vec{\Omega}^F \end{bmatrix}_{6\times F,} \tag{7}$$

where

$$S = \begin{bmatrix} f\gamma & 0 & 0 & 0 & f & 0 \\ \alpha & 0 & -\gamma & 0 & 0 & 0 \\ \beta & 0 & 0 & 0 & 0 & -1 \\ 0 & f\gamma & 0 & -f & 0 & 0 \\ 0 & \alpha & 0 & 0 & 0 & 1 \\ 0 & \beta & -\gamma & 0 & 0 & 0 \\ 0 & 0 & -\frac{\alpha}{f} & 0 & \frac{1}{f} & 0 \\ 0 & 0 & -\frac{\beta}{f} & -\frac{1}{f} & 0 & 0 \end{bmatrix} \tag{8}$$

and $\vec{t}^j, \vec{\Omega}^j$, are the camera translation and rotation, between the reference frame J and frame K^j $(j = 1..F)$. Note that the shape matrix S is common to all frames, because they all share the same plane normal $\vec{n} = (\alpha, \beta, \gamma)^T$ and focal length f. The dimensionality of the matrices on the right hand side of (7) implies that, without noise, the parameter matrix P is of rank 6 at most. This implies that the collection of all the $\vec{p}^j$s $(j = 1..F)$ resides in a low dimensional linear subspace. The actual rank of P may be even lower than six, depending on the complexity of the camera motion over the sequence (e.g., in case of uniform motion it will be 1).

Note that the quadratic motion model defined in (1) and (2) must be valid between the reference frame and each of the other frames. As this model is only an approximation to the full motion equations, it is applicable to a limited number of frames. This number is sequence-dependent (e.g., will be large for airborne videos and smaller for indoor videos). In our experiments, the number of frames varied from 10 to 50 frames. Automatic determination of the number of frames to which the validity of the model extends is a topic for future work.

4.2 Incorporating Subspace Constraint into Multiframe Estimation

In this section, we show how the low-rank constraint on P can be incorporated into the estimation of $\vec{p}^1, \ldots, \vec{p}^F$, *without* explicitly solving for any $3D$ information, nor for camera calibration.

It is *not* advisable to first solve for P and then project its columns onto a lower dimensional subspace, because then the individual $\vec{p}^j$s will already be very erroneous. Instead, we would like to use the low dimensionality constraint to *constrain* the estimation of the individual $\vec{p}^j$s a priori. We next show how we can apply this constraint *directly* to *measurable image quantities* prior to solving for the individual $\vec{p}^j$s. This method is presented below. For a review and comparison of the other possible approaches which we have tried (and rejected), see Appendix C.

Since all inspection frames $K^1, \ldots, K^F$ share the same reference frame J, (4) can be extended to multiple frames as:

$$\mathcal{C}_{8\times 8}\left[\vec{p}^1 \cdots \vec{p}^F\right]_{8\times F} = \left[\vec{b}^1 \cdots \vec{b}^F\right]_{8\times F} \tag{9}$$

or, in short: $\mathcal{C}P = B$. Equation (9) implies that $rank(B) \leq rank(P) \leq 6$. B contains only measurable image quantities. Therefore, instead of applying the low-rank

constraint to P, we apply it directly to B, and only then solve for P. Namely, at each iteration i of the algorithm, first compute $B_i = [\vec{b}_i^1 \cdots \vec{b}_i^F]$ and then project its columns onto a lower-dimensional linear subspace by seeking a matrix $\hat{B}_i$ of rank r ($r \leq 6$), which is closest to B_i (in the Frobenius norm). Then solve for $P_i = \mathcal{C}_i^{-1}\hat{B}_i$, which yields the desired $\vec{p}^j$s.

The advantage of applying the constraint to B instead of P can also be explained as follows: Note that the matrix $\mathcal{C}$ in (4) is the posterior inverse covariance matrix[1] of the parameter vector $\vec{p}$. Therefore, applying the constraint to B is equivalent to applying it to the matrix P, but after weighting its columns by the inverse covariance matrix $\mathcal{C}$ (Note that all $\vec{p}^j$s share the same $\mathcal{C}$.)

The subspace projection of the columns of B_i is done using SVD (see Appendix C). To equalize the effect of subspace projection on all matrix entries and to further condition the numerical process, we use the coordinate normalization technique suggested by Hartley [9] (also used in the two-frame method).

In the results presented throughout this paper, we compare our multi-frame subspace constrained method against the unconstrained two-frame method described in Section 3. The reason for this is that this specific two-frame method and our multi-frame method differ *only* in the use of the subspace constraints. Other than that, they are *identical*. This allows us to isolate the effects of the subspace constraints on the estimation process.

Fig. 1 shows a comparison of applying the *two*-frame and and *multi*-frame alignment techniques to a small image region (marked by a rectangle). Figs. 1g and 1j are the result of the *two*-frame alignment. The region of interest is indeed aligned, but the rest of the image is completely distorted. In contrast, the *multi*-frame constrained alignment (applied to 17 frames), successfully aligned the entire image even though applied only to the same small region. This can be seen in Figs. 1h and 1k.

Fig. 2 shows a *quantitative* comparison of the *two*-frame and *multi*-frame alignment techniques. When applying two-frame motion estimation to the small region, the farther the pixel is from the center of the region the larger the error is. However, when applying multi-frame motion estimation to the same small region, the errors everywhere are at subpixel level.

Fig. 3 shows another comparison of applying *two*-frame alignment and *multi*-frame alignment to small image regions. The sequence contains 34 frames taken by a moving camera. Because the camera is imaging the scene from a short distance and because its motion also contains a translation, therefore different planar surfaces (e.g., the house, the road sign, etc.) induce different $2D$ parametric motions. As long as the house was not occluded, the *two*-frame alignment, when applied only to the house region, stabilized the house reasonably well. However, once the house was partially occluded by the road sign, and was not fully in the camera's field of view, the quality of the

1. The error term Err in (3) can be viewed as an χ^2 merit function, assuming Gaussian noise on the temporal brightness differences $(K(\vec{x}) - J(\vec{x}))$ (or for the multi-frame case $(K^j(\vec{x}) - J(\vec{x}))$), with the same standard deviation for all image points. The eight linear equations of (4) are the *normal equations* of the least-squares minimization of Err. Hence, under the above Gaussian noise assumptions, the matrix $\mathcal{C}$, defined by (5), is the inverse covariance matrix of the parameter vector $\vec{p}$ (see [16], the chapter on "General Linear Least-Squares").

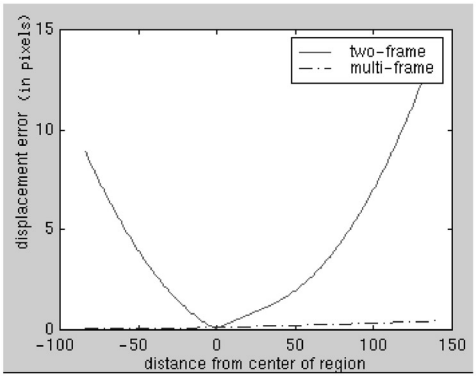

Fig. 2. Quantitative Comparison. A quantitative comparison of *two*-frame and *multi*-frame alignment. The values in the graph correspond to misalignments in Figs. 1g and 1h. These errors are displayed as a function of the distance from the center of the rectangular region in Fig. 1. The results of two-frame alignment applied to the entire image region (Fig. 1f) were used as ground truth.

two-frame alignment degraded drastically (see Figs. 3f and 3j). The *multi*-frame constrained alignment, on the other hand, successfully aligned the house, even in frames where only a small portion of the house was visible (see Figs. 3g and 3k). In this case, the actual rank used was much smaller than six (it was two), and was detected from studying the rate of decay of the eigenvalues of the matrix B (Currently, this detection was done manually; however, this process could be automated, see [8]). Applying a robust two-frame estimation technique might slightly improve the quality of the two-frame alignment as it will ignore outliers. However, because of the very small region of interest it will still not provide accurate alignment (see [19]). While our multi-frame method does not include any explicit outlier rejection process, it gave good alignment results even for frames with more outliers than inliers (e.g., see Fig. 3g). This indicates the built-in robustness of the multi-frame method.

4.3 Frame Reliability

We further enhance the multi-frame estimation process, introducing confidence measures on frames. We associate a weight w^j with each frame K^j according to the accuracy of alignment obtained at the previous iteration. We can use these weights to obtain confidence-weighted subspace projection as follows: Instead of projecting the columns of the matrix B on to a lower-dimensional subspace, we will project the columns of:

$$\tilde{B} = B \begin{bmatrix} w^1 & & 0 \\ & \ddots & \\ 0 & & w^F \end{bmatrix}.$$

Beacause the weights matrix is regular ($w^j \neq 0$, $\forall j$), the rank of $\tilde{B}$ is the same as that of B. After projecting $\tilde{B}$ to obtain $\hat{\tilde{B}}$, we multiply $\hat{\tilde{B}}$ by the inverse weight matrix to get $\hat{B}$, i.e.:

$$\hat{B} = \hat{\tilde{B}} \begin{bmatrix} \frac{1}{w^1} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{w^F} \end{bmatrix}$$

which is then used to solve for $P : P = \mathcal{C}^{-1}\hat{B}$.

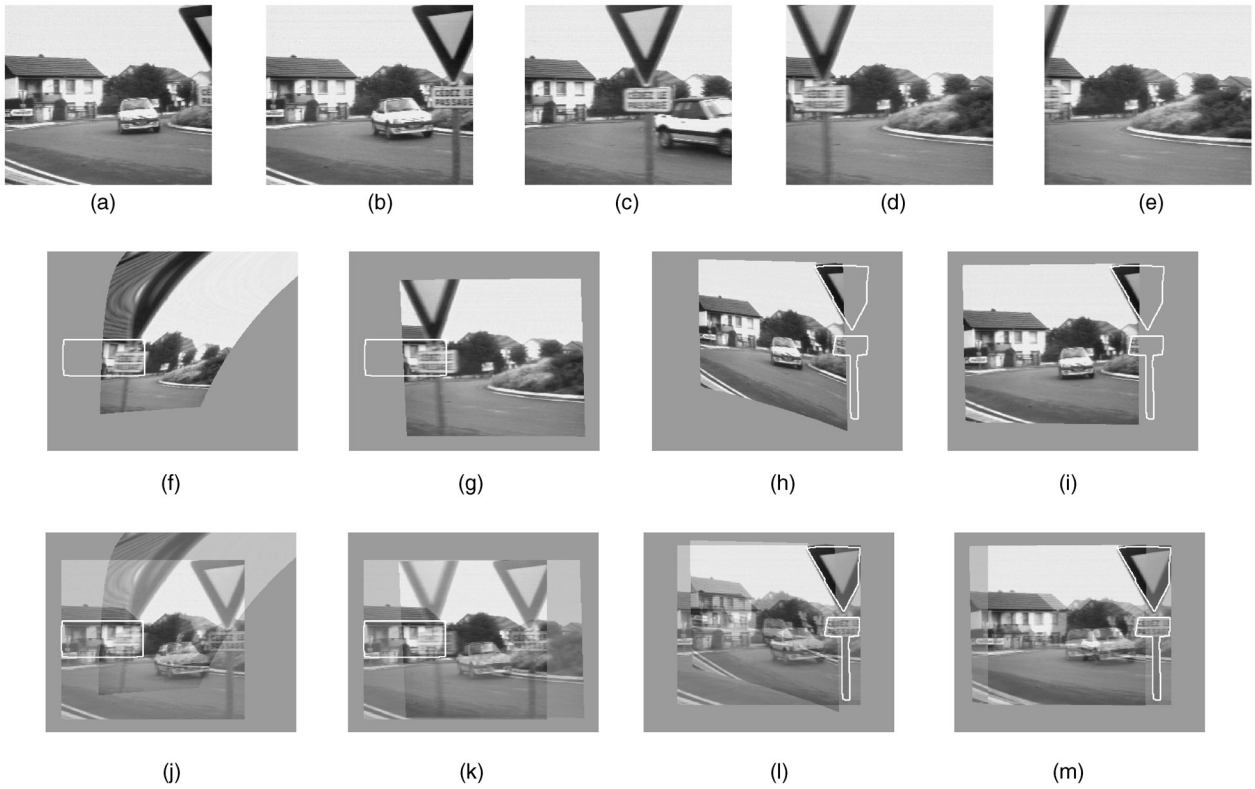

Fig. 3. Two-frame vs. multi-frame. (a), (b), (c), (d), and (e) are sample images from a sequence of 34 frames. Image (b) was used as the reference frame. (f) shows poor two-frame alignment of the house region between the reference frame (b) and frame (d). The frame was completely distorted because the house region was significantly occluded by the road sign. The region of interest, marked in white, was manually selected only in the reference frame (No need to specify the region in any of the other frames, as all other frames are matched against the reference frame). For clarity of the results, we overlayed the reference frame region-of-interest on all the result figures. (j) shows the same result overlayed on top of the reference image (b). (g) is the corresponding result from applying the constrained multi-frame alignment. The house is now well-aligned even though only a small portion of the house is visible (see overlay image (k)), while the rest of the image is not distorted. The road sign is not aligned because it is at a different depth and displays accurate $3D$ parallax. (h) and (l) shows badly distorted two-frame alignment applied to the road sign (region marked in white) between frames (b) and (a) (where the sign is barely visible). Although the sign appears aligned, the rest of the image is distorted, and the house displays wrong parallax. (i) and (m) show the corresponding result (to (h) and (l)) from applying the constrained multiplane (multi-frame) alignment to the sign (see text). The misalignment of the house is due to $3D$ parallax. For full video sequences of the results, see http://www.wisdom.weizmann.ac.il/~lihi/Demos/multiframe-align.html.

The weights should reflect the following: 1) the amount of residual misalignment in the region of interest after registration, 2) the amount of distortions introduced by the quadratic parameters (e.g., bending of straight lines), and 3) the degree of overlap within the region of interest (after alignment) between the reference frame and the inspection frame.

For our experiments, we used a heuristic measure reflecting only the amount of distortions in the image. The instantaneous motion assumptions, leading to the parametric quadratic model, imply small values for the parameters p_7, p_8 of (2). Typical values of p_7 and p_8 (without the coordinate normalization of Hartley [9]) are smaller than $1E{-}04$. When this is not maintained, the resulting image after alignment is highly distorted. Based on this, we assigned each frame a weight w^j according to:

$$w^j = \frac{1}{\max(\mid p_7 \mid, \mid p_8 \mid, \epsilon),}$$

where ϵ is used to avoid division by 0, and in our experiments was set to be $\epsilon = 1E{+}06$.

As discussed in Section 4.2, applying the low rank constraint to B instead of P implicitly associates confidence weights with the different *parameter components* (i.e., $p_1, \ldots, p_8$). Here, we associate confidence weights with the different *frames* ($j = 1 \ldots F$). Therefore, using both types of weights, we obtain a confidence-weighted subspace projection both in the *parameter space* and in the *frame space*. We found this approach to significantly improve the results, although no convergence analysis was performed.

5 Extending to Multiple Planes

In this section, we show that in scenes containing multiple planar surfaces, even stronger subspace constraints can be derived and used to improve the parametric motion estimation. We present two different multi-frame subspace constraints for sequences with multiple planes. The second constraint does not require constant camera calibration.

5.1 The Multiplane Rank-6 Constraint

Let $\pi_1, \ldots, \pi_m$ be m planar surfaces with normals $\vec{n}_{\pi_1} = [\alpha_1, \beta_1, \gamma_1]^T, \ldots, \vec{n}_{\pi_m} = [\alpha_m, \beta_m, \gamma_m]^T$, respectively. All planes share the same $3D$ motion. Let $P_{\pi_1}, \ldots, P_{\pi_m}$ be the corresponding quadratic motion parameter *matrices* and let

$S_{\pi_1}, \ldots, S_{\pi_m}$ be the corresponding shape matrices, as defined by (7) and (8). We can stack the $P_{\pi_\eta}(\eta = 1 \ldots m)$ matrices to form an $8m \times F$ matrix $\mathcal{P}$, where each column corresponds to one frame. Since all planar surfaces π_η share the same $3D$ camera motion between a *pair of frames*, we get from (7):

$$\mathcal{P} = \begin{bmatrix} P_{\pi_1} \\ \hline \vdots \\ \hline P_{\pi_m} \end{bmatrix}_{8m \times F} = \begin{bmatrix} S_{\pi_1} \\ \hline \vdots \\ \hline S_{\pi_m} \end{bmatrix}_{8m \times 6} \begin{bmatrix} \vec{t}^1 & \cdots & \vec{t}^F \\ \vec{\Omega}^1 & & \vec{\Omega}^F \end{bmatrix}_{6 \times F.} \quad (10)$$

The dimensionality of the matrices on the right hand side of (10) implies that, without noise, the parameter matrix $\mathcal{P}$ is also of rank 6 at most. (As before, the actual rank of $\mathcal{P}$ may be even lower than six, depending on the complexity and variability of the camera motion over the sequence.)

5.2 Incorporating Multiplane Rank-6 Constraint into Estimation

Again, we would like to apply the constraint to measurable image quantities. We show next how this can be done. Let $\mathcal{C}_{\pi_1}, \ldots, \mathcal{C}_{\pi_m}$ be the matrices corresponding to planes $\pi_1, \ldots, \pi_m$. Note that the matrices $\mathcal{C}_{\pi_\eta}$s are different from each other due to the difference in the region of summation, which is the region of each planar surface in the reference frame. We can write:

$$\begin{bmatrix} \mathcal{C}_{\pi_1} & 0 & \cdots & 0 \\ 0 & \mathcal{C}_{\pi_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathcal{C}_{\pi_m} \end{bmatrix} \begin{bmatrix} P_{\pi_1} \\ \vdots \\ P_{\pi_m} \end{bmatrix}_{8m \times F} = \begin{bmatrix} B_{\pi_1} \\ \vdots \\ B_{\pi_m} \end{bmatrix}_{8m \times F}. \quad (11)$$

or, in short: $\mathcal{C}\mathcal{P} = \mathcal{B}$. Note that here, as opposed to (9), $\mathcal{P}, \mathcal{B}$, and $\mathcal{C}$ contain information from all planar surfaces. Equation (11) implies that $rank(\mathcal{B}) \leq rank(\mathcal{P}) \leq 6$. We thus project the columns of matrix $\mathcal{B}$ onto a lower-dimensional subspace, at each iteration, resulting in $\hat{\mathcal{B}}$ (which is closest to $\mathcal{B}$ in the Frobenius norm), and then solve for $\mathcal{P} = \mathcal{C}^{-1}\hat{\mathcal{B}}$. In other words, we solve for all parametric motions of all planar surfaces, across all frames, simultaneously. The low-rank constraint is stronger here than in the single plane case, because the matrix $\mathcal{B}$ is of a larger dimension $(8m \times F)$.

5.3 Relative Motion Rank-3 Constraint

Moreover, by looking at the *relative* motion of planar surfaces we can get an even *stronger* subspace constraint, which is true *even for the case of varying camera calibration*. For readability purpose, we show it here only for the case of varying focal length. However, this is true for the general uncalibrated case (see Appendix B).

Let f and f^j be the focal length of the frame J and frame K^j, respectively. Let π_r be an arbitrary planar surface (π_r could be one of $\pi_1, \ldots, \pi_m$). Denote $\Delta\vec{p}^j_\eta = \vec{p}^j_{\pi_\eta} - \vec{p}^j_{\pi_r}$ $(\eta = 1 \ldots m)$. It is easy to see from (2) that taking the difference $\Delta\vec{p}^j_\eta$ eliminates all effects of camera rotation, leaving only effects of camera translation and the focal length:

$$[\Delta\vec{p}^j_\eta]_{8\times 1} = [\Delta\tilde{S}_\eta]_{8\times 3}[\vec{\tau}^j]_{3\times 1} \quad (12)$$

where $\vec{\tau}^j = [f^j t^j_X, f^j t^j_Y, t^j_Z]^T$, and:

$$\Delta\tilde{S}_\eta = \begin{bmatrix} (\gamma_\eta - \gamma_r) & 0 & 0 \\ \frac{1}{f}(\alpha_\eta - \alpha_r) & 0 & -(\gamma_\eta - \gamma_r) \\ \frac{1}{f}(\beta_\eta - \beta_r) & 0 & 0 \\ 0 & (\gamma_\eta - \gamma_r) & 0 \\ 0 & \frac{1}{f}(\alpha_\eta - \alpha_r) & 0 \\ 0 & \frac{1}{f}(\beta_\eta - \beta_r) & -(\gamma_\eta - \gamma_r) \\ 0 & 0 & -\frac{1}{f}(\alpha_\eta - \alpha_r) \\ 0 & 0 & -\frac{1}{f}(\beta_\eta - \beta_r) \end{bmatrix}.$$

$\Delta\tilde{S}_\eta$ is common to all *frames*. The camera translation $\vec{t}^j$ and focal length f^j are common to all *planes* (between the reference frame J and frame K^j). We can therefore extend (12) to *multiple planes* and *multiple frames* as follows:

$$\Delta\mathcal{P} = \begin{bmatrix} \Delta P_{\pi_1} \\ \hline \vdots \\ \hline \Delta P_{\pi_m} \end{bmatrix}_{8m \times F} = \begin{bmatrix} \Delta\tilde{S}_{\pi_1} \\ \hline \vdots \\ \hline \Delta\tilde{S}_{\pi_m} \end{bmatrix}_{8m \times 3} [\vec{\tau}^1 \ldots \vec{\tau}^F]_{3\times F.} \quad (13)$$

The dimensionality of the matrices on the right hand side of (13) implies that, without noise, the difference parameter matrix $\Delta\mathcal{P}$ is of rank 3 at most.

It is possible to obtain a similar constraint (with $rank \leq 4$), for *general homographies* case [23] (as opposed to the instantaneous case). The rank-4 constraint is an extension to the constraint shown by [18]. Shashua and Avidan presented a rank-4 constraint on the collection of homographies of *multiple planes* between a *pair of frames*. In our case [23], the constraints are on *multiple planes* across *multiple frames*. We refer the reader to [23] for more details.

In Section 5.4, we show how the multiplane rank-3 constraint can be incorporated into the multi-frame estimation process to further enhance planar-motion estimation.

5.4 Incorporating the Rank-3 Constraint into Multiframe Estimation

Assume that for one planar surface, π_r, we know the collection of all its parametric motions, P_{π_r} (This is either given to us, or estimated at previous iteration). We would like to use the $(rank \leq 3)$ constraint to refine the estimation of the collection of parametric motions $P_{\pi_1}, \ldots, P_{\pi_m}$, of all other planes. Using (11), we derive:

$$\mathcal{C}\Delta\mathcal{P} = \begin{bmatrix} B_{\pi_1} - \mathcal{C}_{\pi_1} P_{\pi_r} \\ \hline \vdots \\ \hline B_{\pi_m} - \mathcal{C}_{\pi_m} P_{\pi_r} \end{bmatrix}_{8m \times F} = \mathcal{B}^* \quad (14)$$

Therefore, $rank(\mathcal{B}^*) \leq rank(\Delta\mathcal{P}) \leq 3$. To incorporate the constraint into the estimation of the individual P_{π_η}s, we project the columns of the matrix $\mathcal{B}^*$ onto a lower-dimensional (≤ 3) subspace at each iteration, resulting in $\hat{\mathcal{B}}^*$ (which is closest to $\mathcal{B}^*$ in Frobenius norm). Therefore (from (14)), we can estimate a new matrix B^{**}

$$\mathcal{B}^{**} = \mathcal{C}\mathcal{P} = \begin{bmatrix} \mathcal{C}_{\pi_1} \\ \hline \vdots \\ \hline \mathcal{C}_{\pi_m} \end{bmatrix} P_{\pi_r} + \hat{\mathcal{B}}^* \quad (15)$$

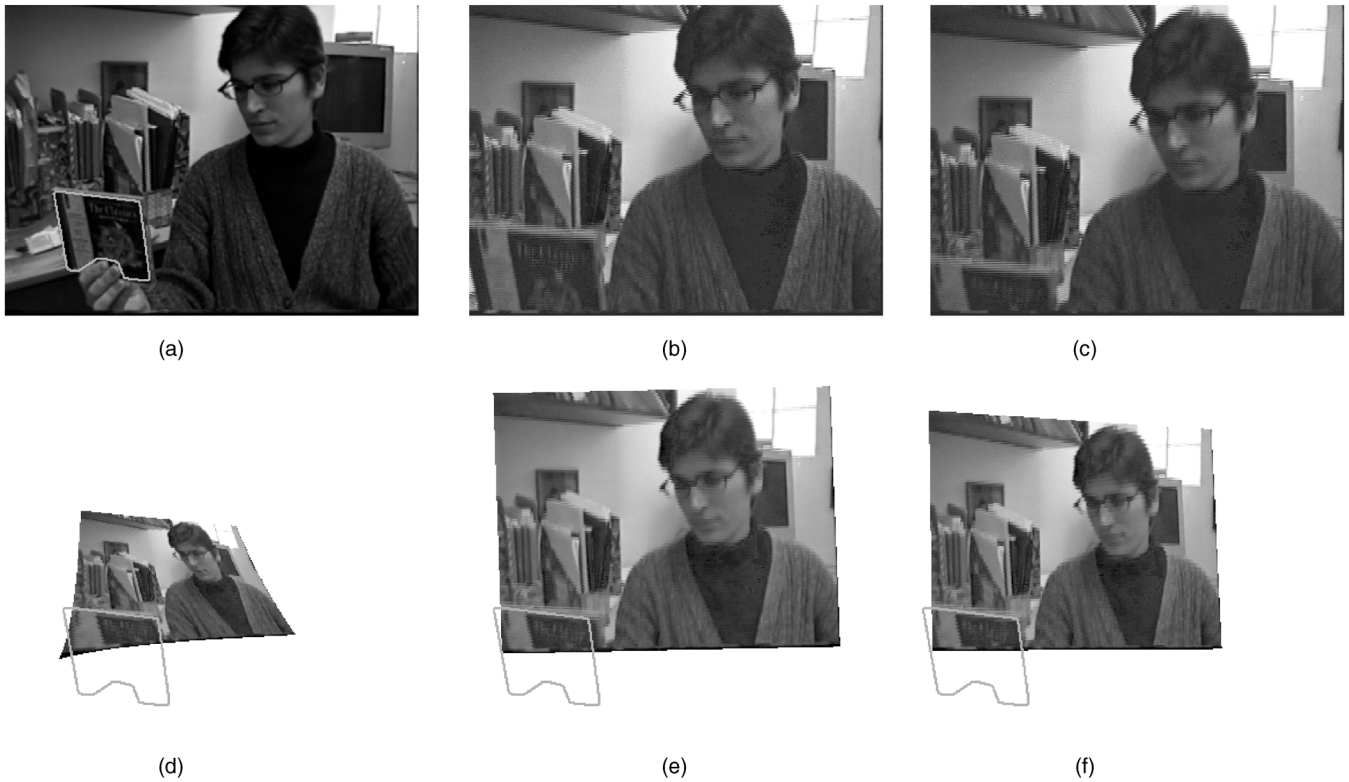

Fig. 4. Single-Plane vs. Multiplane. (a), (b), and (c) are sample images from a sequence of 26 frames obtained by a moving camera. Images (b) and (c) show how as the camera moves, the disk held by the person in the image moves out of the field of view. Image (a) (with the region of interest marked) was used as the reference frame. The region of interest (the disk) was manually selected only in the reference frame (no need to specify the region in any of the other frames as all other frames are matched against the reference frame). (d) shows an example of bad two-frame alignment of the disk region between the reference frame (a) and frame (c). This image shows frame (c) warped towards frame (a) according to the computed alignment parameters using the two-frame method. The frame was completely distorted because the disk region is very small. (e) is an example of *single-plane multi-frame* alignment. This image shows frame (c) warped towards frame (a) according to the computed parameters using the multi-frame method. Since the disk region is very small in all frames, and is partially out of the frame in many of them (e.g., (b) and (c)), the *single-plane multi-frame* alignment was insufficient to give accurate results. Though the image is no longer distorted, it can be seen in (e) that the disk region is not perfectly aligned (the boundaries of the disk are not perfectly aligned with the marked region). (f) displays high quality alignment resulting from applying the constrained *relative motion, multiplane, multi-frame* alignment (see text). The person body (which is larger) was used as the reference plane. More accurate alignment results of the person body region led to good alignment of the disk region as well (although they reside on different planes and have different parameters).

and then solve for $P_{\pi_\eta} = \mathcal{C}_{\pi_\eta}^{-1} B_\eta^{**}$. Note that here $\mathcal{B}^*$ and $\mathcal{B}^{**}$ are constructed from measurable image quantities ($\mathcal{B}$ and $\mathcal{C}$), as well as from the parameters P_{π_r}, which are either known or else estimated at previous iteration. The process is repeated at each iteration. Note: 1) If we know that the focal length does not change along the sequence, we can also apply the constraint $rank(\mathcal{B}^{**}) \leq rank(\mathcal{P}) \leq 6$ prior to solving for P_{π_η}; 2) π_r can alternate between the planes, but we found it to work best when π_r was chosen to be a "dominant" plane (i.e., one whose matrix $\mathcal{C}_{\pi_r}$ is best conditioned).

Fig. 3 also presents a comparison of regular (unconstrained) two-frame alignment with the *multiplane* constrained alignment, applied to the road sign. The motion parameters of the house region were first estimated using the single-plane multi-frame constrained alignment (see Section 4). These were then used as inputs for constraining the estimation of the sequence of 2D motion parameters of the road sign. The two-frame alignment technique did not perform well in cases when the sign was only partially visible (see Figs. 3h and 3l). The multiplane (multi-frame) constrained alignment, on the other hand, stabilized the sign well even in cases when the sign was only partially visible (see Figs. 3i and 3m).

Fig. 4 shows a comparison of *single-plane* constrained alignment with the *multiplane* constrained alignment, applied to the disk region. The motion parameters of the person body region were first estimated using the single-plane multi-frame constrained alignment (see Section 4). Though not being a real planar surface, the body region can be approximated by one. These results were then used as inputs for constraining the estimation of the sequence of 2D motion parameters of the disk region. The single-plane multi-frame alignment technique, although significantly better then the two-frame unconstrained alignment (see Fig. 4d), did not achieve perfect alignment in some cases (see Fig. 4e). This is because the disk region was very small in all frames and in some frames was hardly visible at all (e.g., Fig. 4c). The multiplane (multi-frame) constrained alignment, on the other hand, stabilized the disk even in these cases (see Fig. 4f). The differences are more evident in video mode, where even small imperfections in the stabilization are easily noticed.

In the current implementation, we first estimate the parameters of one plane and then use these to constrain the

estimation of the parameters of a second plane. This could be extended to an iterative process, where at each iteration a different plane is used as a reference plane to constrain the estimation of all the other planes. However, we have not implemented or experimented with this iterative approach.

6 Concluding Remarks

In this paper, a method was introduced for extending existing two-frame planar-motion estimation techniques into a simultaneous multi-frame planar-motion estimation by exploiting multi-frame subsapce constraints on the motion parameters. It was shown how these constraints can be incorporated into the $2D$ parametric estimation of planar motion, without solving for any $3D$ information, nor for camera calibration. The subspace constraints were applied directly to measurable image quantities, which were then used to solve for the $2D$ motion parameters.

The advantage of the presented method, as was shown, is that by simultaneously using information from multiple frames, these frames which have more information content can constrain the estimation in frames where the information is sparse or noisy, as in the case of small image regions or partial occlusion. The multiplane multi-frame motion estimation leads to even more constrained estimation and allows for varying camera calibration.

Currently, the suggested method is a batch process which estimates the motion of the entire sequence simultaneously. Such a process can theoretically be applied to a long sequence by repeatedly applying it to a shifting window in time. An interesting question is how to efficiently and incrementally modify the matrices associated with the shifting window, without reestimating all the necessary values again. This is a topic for future research.

Appendix A

Instantaneous Motion Model

The notations describing the 3D motion of the camera and the corresponding 2D motion of the planar surfaces in the image plane, for the fully uncalibrated camera model, are introduced in this section. These are similar to the ones first suggested by [15], only there the camera calibration was assumed to be fixed and, here, we allow for varying camera calibration.

A.1 The Case of the Uncalibrated Camera

Let $\vec{Q} = (X, Y, Z)^T$ and $\vec{Q}' = (X', Y', Z')^T$ denote a scene point with respect to two different camera views, respectively. Let $\vec{q} = (x, y, 1)^T$ and $\vec{q}' = (x', y', 1)^T$ denote the corresponding points in the two images. We can write:

$$\vec{q} \cong V\vec{Q} \quad , \quad \vec{q}' \cong V'\vec{Q}', \tag{16}$$

where $\cong$ denotes equality up to a scale factor. V and V' are projection matrices [6].

Let π be a planar surface with plane normal $\vec{n}$, then $\vec{n}^T\vec{Q} = 1$ for all points $\vec{Q} \in \pi$ ($\vec{n} = \frac{\vec{m}}{d_\pi}$, where $\vec{m}$ is a unit vector in the direction of the plane normal, and d_π is the distance of the plane from the first camera center). The transformation between the $3D$ coordinates of a scene point $Q \in \pi$ in the two views, can be expressed by:

$$\vec{Q}' = G\vec{Q}, \tag{17}$$

where

$$G = R + \vec{t}\vec{n}^T. \tag{18}$$

R is the rotation matrix and $\vec{t}$ is the translation of the camera. Therefore, the induced transformation between the corresponding *image* points is:

$$\vec{q}' \cong H\vec{q}, \tag{19}$$

where

$$H = V'(R + \vec{t}\vec{n}^T)V^{-1} \tag{20}$$

is the induced homography between the two views of the plane π. From (19) it is clear that when H is computed from image point correspondences, it can be estimated only up to a scale factor. Denoting by $[H]_i$ the ith row of the matrix H, we can further derive:

$$\begin{aligned} x' &= \frac{[H]_1\vec{q}}{[H]_3\vec{q}} \\ y' &= \frac{[H]_2\vec{q}}{[H]_3\vec{q}} \end{aligned} \tag{21}$$

and the $2D$ displacement $(u, v) = (x' - x, y' - y)$ in the image plane, can be expressed by:

$$\begin{aligned} u &= \frac{([H]_1 - x[H]_3)\vec{q}}{[H]_3\vec{q}} \\ v &= \frac{([H]_2 - y[H]_3)\vec{q}}{[H]_3\vec{q}}. \end{aligned} \tag{22}$$

We next want to show how under certain assumptions the denominator (which we will denote by $\mathcal{D}$) is ≈ 1. This is required in order to get a quadratic parametric model. In general, the camera internal calibration matrix V has the following form [6]:

$$V = \begin{bmatrix} -f\kappa_u & f\kappa_u \cot(\theta) & u_0 \\ 0 & -\frac{f\kappa_v}{\sin(\theta)} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \equiv \begin{bmatrix} a & b & c \\ 0 & d & e \\ 0 & 0 & 1 \end{bmatrix}, \tag{23}$$

where f is the camera focal length, κ_u, κ_v represent the scaling of retinal coordinates, u_0, v_0 are the principal point coordinates, and θ is the angle between the retinal axes.

Assuming the camera rotation is relatively small, the matrix R can be approximated by:

$$R = \begin{bmatrix} 1 & -\Omega_Z & \Omega_Y \\ \Omega_Z & 1 & -\Omega_X \\ -\Omega_Y & \Omega_X & 1 \end{bmatrix}. \tag{24}$$

Using (24), (23), and (20), we can show that the denominator $\mathcal{D}$ of (22) is the following expression:

$$\begin{aligned} \mathcal{D} =& \Omega_Y \frac{x - u_0}{f\kappa_u} + \Omega_Y \cos(\theta)\frac{y - v_0}{f\kappa_v} \\ & - \Omega_X \sin(\theta)\frac{y - v_0}{f\kappa_v} + \frac{t_Z}{Z} + 1, \end{aligned} \tag{25}$$

therefore, the denominator $\mathcal{D} \approx 1$ if the following assumptions hold: 1) The scaling of retinal coordinates is not very

small (i.e., κ_u, κ_v are not close to 0), 2) small field of view (i.e., f is large), 3) small camera rotation, and 4) $\frac{t_Z}{Z} \ll 1$.

Substituting (24), (23), and (20) in (22), assuming $\mathcal{D} \approx 1$, yields:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} p_1 + p_2 x + p_3 y + p_7 x^2 + p_8 xy \\ p_4 + p_5 x + p_6 y + p_7 xy + p_8 y^2 \end{bmatrix}, \tag{26}$$

where:

$$\begin{aligned}
p_1 &= -\frac{a'c}{d} + \frac{a'be}{ad} - \frac{b'e}{d} + c' - \left(b' + \frac{c'e}{d}\right)\Omega_X \\
&\quad + \left(a' + \frac{cc'}{a} - \frac{bc'e}{ad}\right)\Omega_Y \\
&\quad + \left(\frac{a'e}{d} - \frac{b'c}{a} + \frac{b'be}{ad}\right)\Omega_Z \\
&\quad + a'\Phi_3 t_X + b'\Phi_3 t_Y + c'\Phi_3 t_Z \\
p_2 &= \frac{a'}{a} - 1 + \frac{e}{d}\Omega_X + \left(-\frac{c}{a} - \frac{c'}{a} + \frac{be}{ad}\right)\Omega_Y \\
&\quad + \frac{b'}{a}\Omega_Z + a'\Phi_1 t_X + b'\Phi_1 t_Y \\
&\quad + (c'\Phi_1 - \Phi_3) t_Z \\
p_3 &= -\frac{a'b}{ad} + \frac{b'}{d} + \frac{c'}{d}\Omega_X + \frac{c'b}{ad}\Omega_Y \\
&\quad - \left(\frac{a'}{d} + \frac{b'b}{ad}\right)\Omega_Z \\
&\quad + a'\Phi_2 t_X + b'\Phi_2 t_Y + c'\Phi_2 t_Z \\
p_4 &= -\frac{d'e}{d} + e' - \left(d' + \frac{ee'}{d}\right)\Omega_X \\
&\quad + \left(\frac{ce'}{a} - \frac{bee'}{ad}\right)\Omega_Y \\
&\quad + \left(-\frac{d'c}{a} + \frac{bd'e}{ad}\right)\Omega_Z + d'\Phi_3 t_Y + e'\Phi_3 t_Z \\
p_5 &= -\frac{e'}{a}\Omega_Y + \frac{d'}{a}\Omega_Z + d'\Phi_1 t_Y + e'\Phi_1 t_Z \\
p_6 &= \frac{d'}{d} - 1 + \left(\frac{e}{d} + \frac{e'}{d}\right)\Omega_X \\
&\quad + \left(-\frac{c}{a} + \frac{be}{ad} + \frac{be'}{ad}\right)\Omega_Y \\
&\quad - \frac{bd'}{ad}\Omega_Z + d'\Phi_2 t_Y + (e'\Phi_2 - \Phi_3) t_Z \\
p_7 &= +\frac{1}{a}\Omega_Y - \Phi_1 t_Z \\
p_8 &= -\frac{1}{d}\Omega_X - \frac{b}{ad}\Omega_Y - \Phi_2 t_Z
\end{aligned} \tag{27}$$

and

$$\begin{aligned}
\Phi_1 &= \frac{\alpha}{a} \\
\Phi_2 &= \frac{\beta}{d} - \frac{b\alpha}{ad} \\
\Phi_3 &= \left(\frac{be}{ad} - \frac{c}{a}\right)\alpha - \frac{e}{d}\beta + \gamma.
\end{aligned}$$

This result is equivalent to the ones showed by [15], [2], [4], [10], only they assumed fixed camera calibration and, here, we allow for varying camera calibration.

A.2 The Longuet-Higgins Model

The notations introduced in Appendix A.1 are true for the general case of uncalibrated camera. For readability purposes, we chose to use in the body of the work a limited case, which allows for simpler forms of equations.

Making the following assumptions on the camera calibration: 1) the retinal coordinates are orthogonal (i.e., $\theta = 90°$), 2) the principal point is at the origin of the coordinate axes (i.e, $(u_0, v_0) = (0, 0)$), and 3) the aspect ratio equals 1 (i.e., $\kappa_u = \kappa_v$). Introducing these assumptions into (23), we get a simpler form for the calibration matrix:

$$V = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{28}$$

Substituting (28), (24), and (20) in (22) yields:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} p_1 + p_2 x + p_3 y + p_7 x^2 + p_8 xy \\ p_4 + p_5 x + p_6 y + p_7 xy + p_8 y^2 \end{bmatrix}, \tag{29}$$

where:

$$\begin{aligned}
p_1 &= f'(\gamma t_X + \Omega_Y) & p_2 &= \tfrac{f'}{f}(1 + \alpha t_X) - \gamma t_Z - 1 \\
p_3 &= -\tfrac{f'}{f}(\Omega_Z - \beta t_X) & p_4 &= f'(\gamma t_Y - \Omega_X) \\
p_5 &= \tfrac{f'}{f}(\Omega_Z + \alpha t_Y) & p_6 &= \tfrac{f'}{f}(1 + \beta t_Y) - \gamma t_Z - 1 \\
p_7 &= \tfrac{1}{f}(\Omega_Y - \alpha t_Z) & p_8 &= -\tfrac{1}{f}(\Omega_X + \beta t_Z).
\end{aligned} \tag{30}$$

These equations are the same as (1) and (2), which are used in the body of the paper. Equation (30) is a *generalization* of the Longuet-Higgins equations for planar motion [15], which is the case when $f' = f = 1$.

APPENDIX B
MULTIFRAME PARAMETRIC ALIGNMENT FOR AN UNCALIBRATED CAMERA

We now show that the rank-3 constraint derived in Section 5.3 applies also in the more general case of completely unconstrained (and unknown) camera calibration.

Let (a, b, c, d, e) and $(a^j, b^j, c^j, d^j, e^j)$ be the camera calibration parameters of the frame J and frame K^j, respectively. Let π_r be an arbitrary planar surface (π_r could be one of $\pi_1, \ldots, \pi_m$). Denote $\Delta\vec{p}^j_\eta = \vec{p}^j_{\pi_\eta} - \vec{p}^j_{\pi_r}$ ($\eta = 1 \ldots m$). It is easy to see from (27) that taking the difference $\Delta\vec{p}^j_\eta$ eliminates all effects of camera rotation, leaving only effects of camera translation and camera calibration:

$$[\Delta\vec{p}^j_\eta]_{8\times 1} = [\Delta\tilde{S}_\eta]_{8\times 3}[\vec{\tau}^j]_{3\times 1}, \tag{31}$$

where $\vec{\tau}^j = [a^j t^j_X + b^j t^j_Y + c^j t^j_Z, d^j t^j_Y + e^j t^j_Z, t^j_Z]^T$, and:

$$\Delta\tilde{S}_\eta = \begin{bmatrix}
(\Phi_{3\eta} - \Phi_{3r}) & 0 & 0 \\
(\Phi_{1\eta} - \Phi_{1r}) & 0 & -(\Phi_{3\eta} - \Phi_{3r}) \\
(\Phi_{2\eta} - \Phi_{2r}) & 0 & 0 \\
0 & (\Phi_{3\eta} - \Phi_{3r}) & 0 \\
0 & (\Phi_{1\eta} - \Phi_{1r}) & 0 \\
0 & (\Phi_{2\eta} - \Phi_{2r}) & -(\Phi_{3\eta} - \Phi_{3r}) \\
0 & 0 & (\Phi_{1\eta} - \Phi_{1r}) \\
0 & 0 & (\Phi_{2\eta} - \Phi_{2r})
\end{bmatrix}. \tag{32}$$

Note that $\Delta\tilde{S}_\eta$ is common to all *frames*. Since the camera translation and calibration parameters are common to all *planes* (but can differ from one frame to another), we can extend (31) to *multiple planes* and *multiple frames* as follows:

$$\Delta\mathcal{P} = \begin{bmatrix} \underline{\Delta P_{\pi_1}} \\ \vdots \\ \underline{\Delta P_{\pi_m}} \end{bmatrix}_{8m\times F} = \begin{bmatrix} \underline{\Delta\tilde{S}_{\pi_1}} \\ \vdots \\ \underline{\Delta\tilde{S}_{\pi_m}} \end{bmatrix}_{8m\times 3} [\vec{\tau}^1 \dots \vec{\tau}^F]_{3\times F}. \quad (33)$$

The dimensionality of the matrices on the right hand side of (33) implies that, without noise, the difference parameter matrix $\Delta\mathcal{P}$ is of rank 3 at most. This is the generalization of the result obtained in (13). Therefore, the algorithm described in Section 5.3 applies to the case of an uncalibrated camera as well.

APPENDIX C
SUBSPACE PROJECTION USING SVD

In the next section, we describe the technique used for subspace projection, and then in (Section C.2), we analyze its applications to our problem.

C.1 Rank Deficiency and the SVD

Let A be a real $m \times n$ matrix. It can be decomposed (see [8]) into $A = U\Sigma V^T$, where U, V are orthogonal matrices, of dimensions $m \times q$ and $q \times n$, respectively. $q = min(m, n)$ i.e., $U^TU = VV^T = I_q$(here I_q is the identity matrix of size $q \times q$), and $\Sigma = diag(\sigma_1, \sigma_2, \dots, \sigma_q)$ with $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_q \geq 0$. If A is known to be of rank $r < q$, we can set all eigenvalues σ_i with $i > r$ to zero, in Σ. This yields: $\Sigma' = diag(\sigma_1, \dots, \sigma_r, 0, \dots, 0)$. We then compose the matrices back together: $\hat{A} = U\Sigma'V^T$. It was shown by Golub and Van-Loan [8] that $\hat{A}$ is the best possible rank-r approximation to the matrix A, in the Frobenius norm (which is defined by: $\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}$). Hence, we use this technique to project matrices onto a lower r-dimensional subspace.

C.2 Analysis

In the following, we analyze three different possible ways for incorporating the low-rank constraint into the estimation process. Our experiments imply that one of them gives better results than the others (which is the method described in the body of this paper).

Method 1. As was shown in (7), P is of rank ≤ 6. Thus, in the process of parametric motion estimation, we could first solve for P and then project its columns on to a lower-dimensional linear subspace.

Method 2. From (9), we get $\mathcal{C}P = B$, which implies that $rank(B) \leq rank(P) \leq 6$, i.e., we can apply the low-rank constraint to the columns of B and then solve for P.

Method 3. Looking at (4), (5), and (6), we can write:

$$\begin{bmatrix} \underline{\nabla J(\vec{x}_1)^T X(\vec{x}_1)} \\ \vdots \\ \underline{\nabla J(\vec{x}_N)^T X(\vec{x}_N)} \end{bmatrix}_{N\times 8} \begin{bmatrix}\vec{p}^1 \cdots \vec{p}^F\end{bmatrix}_{8\times F}$$
$$= \begin{bmatrix} \underline{J(\vec{x}_1) - K^1(\vec{x}_1)} & \cdots & \underline{J(\vec{x}_1) - K^F(\vec{x}_1)} \\ \vdots & \vdots & \vdots \\ \underline{J(\vec{x}_N) - K^1(\vec{x}_N)} & \cdots & \underline{J(\vec{x}_N) - K^F(\vec{x}_N)} \end{bmatrix}_{N\times F} \quad (34)$$
$$\equiv V_{N\times F},$$

where each column in V is:

$$\vec{v}^j_{N\times 1} = \begin{bmatrix} \underline{J(\vec{x}_1) - K^j(\vec{x}_1)} \\ \vdots \\ \underline{J(\vec{x}_N) - K^j(\vec{x}_N)} \end{bmatrix}$$

and N is the number of pixels in the region of interest. From (34), we get $rank(V) \leq rank(P) \leq 6$ and, so, we can apply the low-rank constraint to V and then solve for P.

Note that in Method 1, the subspace projection is performed in the *parameter space* while in Methods 2 and 3 it is performed in spaces of image measurements.

Method 2 is preferable over Method 1 for the following reason: In Method 1, the error minimization is performed in the parameter space (P). However, errors are not equal for different parameters (some parameters in $\vec{p}$ tend to be more reliable then others). On the other hand, note that the matrix $\mathcal{C}$ in (4) is the posterior inverse covariance matrix of the parameter vector $\vec{p}$. Therefore, applying the constraint to B is equivalent to applying it to the matrix P, but after weighting its components by the inverse covariance matrix $\mathcal{C}$ (Note that all $\vec{p}^j$s share the same $\mathcal{C}$.) In other words, Method 2 corresponds to *confidence-weighted* subspace projection of P and, hence, is superior to Method 1. This theoretical observation was also supported by experimental comparisons of the two methods.

We also found Method 2 to be superior to Method 3 as well for two reasons. First, in terms of runtime complexity, Method 2 is significantly faster than Method 3, as the former applies SVD to $8 \times F$ matrices, while the latter applies it to $N \times F$ matrices. The second reason is that SVD-based projection implicitly assumes errors of the same order of magnitude in all matrix entries. While this assumption is true for B (see below), it is not true for the matrix V of Method 3. Errors in the temporal derivatives (which are the entries of V) are due to misalignment errors and due to errors introduced by the subpixel interpolation. These errors are highly dependent on the magnitude of the spatial derivatives at each pixel. On the other hand, errors in B are of the same order of magnitude. This is shown below. From (6), we get:

$$\vec{b} = -\begin{bmatrix} \sum J_x J_t \\ \sum x J_x J_t \\ \sum y J_x J_t \\ \sum J_y J_t \\ \sum x J_y J_t \\ \sum y J_y J_t \\ \sum (x^2 J_x + xy J_y) J_t \\ \sum (xy J_x + y^2 J_y) J_t \end{bmatrix},$$

where $J_x = \frac{\partial J}{\partial x}(\vec{x}), J_y = \frac{\partial J}{\partial y}(\vec{x})$, and $J_t = K(\vec{x}) - J(\vec{x})$. In our algorithm, we use the coordinate normalization technique suggested by Hartley [9], i.e, the pixel coordinates x, y are normalized such that $-\sqrt{2} \le x, y \le \sqrt{2}$ (see [9]). Thus, the magnitude of each component in the vector $\vec{b}$ is bounded by the corresponding entry of the following vector:

$$\begin{bmatrix} \sum \mid J_x J_t \mid \\ \sqrt{2}\sum \mid J_x J_t \mid \\ \sqrt{2}\sum \mid J_x J_t \mid \\ \sum \mid J_y J_t \mid \\ \sqrt{2}\sum \mid J_y J_t \mid \\ \sqrt{2}\sum \mid J_y J_t \mid \\ 2(\sum \mid J_x J_t \mid + \sum \mid J_y J_t \mid) \\ 2(\sum \mid J_x J_t \mid + \sum \mid J_y J_t \mid) \end{bmatrix}. \quad (35)$$

Observing the vector in (35), it can be seen that all entries in $\vec{b}$ will have errors of the same order of magnitude and, hence, applying the SVD-based projection to B is well conditioned and leads to good numerical results.

The superiority of Method 2 over Method 3 was also supported by experimental tests. Note, however, that the choice between Method 2 and Method 3 was partially based on the implicit noise assumption in the SVD-based projection. It could very well be that a different subspace projection technique will find Method 3 preferable over Method 2, as the rank-r constraint on V, which is a larger matrix than B, theoretically appears to be a stronger constraint.

APPENDIX D
MOTION ESTIMATION PROCESS

In this appendix, we provide the details of the iterative refinement steps of the two-frame parameter estimation process which was briefly described in Section 3.

Let

$$\vec{u_i} = \vec{u_{i-1}} + \vec{\delta u_i} \quad (36)$$

denote the estimate of the displacement at pixel $\vec{x}$, at iteration i. Assuming small $\vec{\delta u_i}$, the brightness constancy constraint can be rewritten as:

$$\begin{aligned} & K(\vec{x} + \vec{u}_{i-1}(\vec{x}; \vec{p_{i-1}})) \\ & = J(\vec{x} - \vec{\delta u_i}(\vec{x}; \vec{p_i})) \\ & \approx J(\vec{x}) - \nabla J(\vec{x})^T \vec{\delta u_i}(\vec{x}; \vec{\delta p_i}). \end{aligned} \quad (37)$$

Rearranging (37), we get:

$$K(\vec{x} + \vec{u}_{i-1}(\vec{x}; \vec{p_{i-1}})) - J(\vec{x}) + \nabla J(\vec{x})^T \vec{\delta u_i}(\vec{x}; \vec{\delta p_i}) = 0. \quad (38)$$

Substituting $\vec{\delta u_i} = \vec{u_i} - \vec{u_{i-1}}$ into (38) yields:

$$\begin{aligned} & (K(\vec{x} + \vec{u}_{i-1}(\vec{x}; \vec{p_{i-1}})) - J(\vec{x}) \\ & - \nabla J(\vec{x})^T \vec{u}_{i-1}(\vec{x}; \vec{p_{i-1}})) \\ & + \nabla J(\vec{x})^T \vec{u}_i(\vec{x}; \vec{p_i}) = 0. \end{aligned} \quad (39)$$

Therefore, given $\vec{p_{i-1}}$ (from the previous iteration), we can solve for a refined estimate of the parameters $\vec{p_i}$ by minimizing the following error function:

$$Err(\vec{p_i}) = \sum_{(\vec{x})} \Big(S_0(\vec{x}; \vec{p}_{i-1}) \; + \; \nabla J(\vec{x})^T \vec{u_i}(\vec{x}; \vec{p_i}) \Big)^2, \quad (40)$$

where

$$\begin{aligned} S_0(\vec{x}; \vec{p}_{i-1}) = & K(\vec{x} + \vec{u}_{i-1}(x; \vec{p}_{i-1})) \\ & - J(\vec{x}) - \nabla J(\vec{x})^T \vec{u_{i-1}}(\vec{x}; \vec{p_{i-1}}) \end{aligned}$$

is known from the previous iteration and $\vec{u}_i(\vec{x}; \vec{p_i})$ is as defined in (1). Equation (40) is the iterative version of (3). Note that (40) is expressed in terms of the *full* parametric transformation $\vec{p_i}$ and not in terms of the incremental transformation $\vec{\delta p_i}$. This is important because the subspace constraints are valid for $\vec{p_i}$, but not for $\vec{\delta p_i}$. Since the images are discrete, we use bilinear interpolation to approximate image intensity values at nondiscrete positions (e.g., at $K(\vec{x} + \vec{u}_{i-1}(x; \vec{p}_{i-1}))$).

The iterative coarse-to-fine estimation process is summarized below:

1. Construct two Gaussian pyramids, one for each input image: $J_0, J_1, J_2..J_L$ and $K_0, K_1, K_2..K_L$ (where $J_0 = J$ is the highest resolution level and J_L is the lowest level).
2. Initialize $\vec{p}_i := 0$
3. For every resolution level, $l = L..0$, do:
   a) Refine $\vec{p}_i$ according to (40), using images J_l and K_l, and the parameters $\vec{p}_{i-1}$.
   b) Set $\vec{p}_{i-1} := \vec{p}_i$ and repeat Step a for a few iterations (typically 5).
4. Propagate $\vec{p}_i$ to the next pyramid level $l-1$, and repeat Step 3 for J_{l-1} and K_{l-1}.

The resulting $\vec{p}_i$ is the estimated parametric transformation.

In our experiments, we used this method for both the two-frame and the multi-frame motion estimation techniques.

REFERENCES

[1] E.H. Adelson, C.H. Anderson, J.R. Bergen, P.J. Burt, and J.M. Ogden, "Pyramid Methods in Image Processing," *RCA Eng.*, vol. 29, no. 6, pp. 33-41, 1985.

[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, 1985.

[3] S. Ayer and H. Sawhney, "Layered Representation of Motion Video Using Robust Maximum-Likelihood Estimation of Mixture Models and MDL Encoding," *Proc. Int'l Conf. Computer Vision,* pp. 777-784, 1995.

[4] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani, "Hierarchical Model-Based Motion Estimation," *Proc. European Conf. Computer Vision,* pp. 237-252, 1992.

[5] M.J. Black and P. Anandan, "Robust Dynamic Motion Estimation Over Time," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 296-302, 1991.

[6] O. Faugeras, *Three-Dimensional Computer Vision—A Geometric Viewpoint.* Cambridge, Mass.: MIT Press, 1996.

[7] D.J. Fleet and K. Langley, "Recursive Filters for Optical Flow," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 1, pp. 61-67, Jan. 1995.

[8] G. Golub and C. Van Loan, *Matrix Computations.* Baltimore and London: The Johns Hopkins Univ. Press, 1989.

[9] R.I. Hartley, "In Defense of the 8-Point Algorithm," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 6, pp. 580-593, June 1997.

[10] M. Irani, B. Rousso, and P. Peleg, "Recovery of Ego-Motion Using Region Alignment," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 3, pp. 268-272, Mar. 1997.
[11] M. Irani, B. Rousso, and S. Peleg, "Detecting and Tracking Multiple Moving Objects Using Temporal Integration," *Proc. European Conf. Computer Vision,* pp. 282-287, 1992.
[12] M. Irani, B. Rousso, and S. Peleg, "Computing Occluding and Transparent Motions," *Proc. Int'l J. Computer Vision,* vol. 12, pp. 5-16, 1994.
[13] S. Ju, M.J. Black, and A.D. Jepson, "Multilayer, Locally Affine Optical Flow, and Regularization with Transparency," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 307-314, 1996.
[14] R. Kumar, P. Anandan, and K. Hanna, "Direct Recovery of Shape from Multiple Views: A Parallax-Based Approach," *Proc. Int'l Confer. Pattern Recognition,* pp. 685-688, 1994.
[15] H.C. Longuet-Higgins, "Visual Ambiguity of a Moving Plane," *Proc. The Royal Soc. of London B,* vol. 223, pp. 165-175, 1984.
[16] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in C.* Cambridge, Mass.: Cambridge Univ. Press, 1992.
[17] H. Sawhney and R. Kumar, "True Multiimage Alignment and Its Application to Mosaic[k]ing and Lens Distortion Correction" *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 450-456, 1997.
[18] A. Shashua and S. Avidan, "The Rank-4 Constraint in Multiple ($\geq$ 3) View Geometry," *Proc. European Conf. Computer Vision,* 1996.
[19] R. Szeliski and P.H.S. Torr, "Geometrically Constrained Structure from Motion: Points on Planes," *Proc. European Workshop on 3D Structure from Multiple Images of Large-Scale Environments,* pp. 171-186, 1998.
[20] C. Tomasi and T. Kanade, "Shape and Motion from Image Streams Under Orthography: A Factorization Method," *Proc. Int'l J. Computer Vision,* vol. 9, pp. 137-154. 1992.
[21] J. Wang and E. Adelson, "Layered Representation for Motion Analysis," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 361-366, 1993.
[22] L. Zelnik-Manor and M. Irani, "Multiframe Alignment of Planesm," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 151-156, 1999.
[23] L. Zelnik-Manor and M. Irani, "Multiview Subsapce Constraints on Homograhpies," *Proc. Int'l Conf. Computer Vision,* pp. 710-715, 1999.

Lihi Zelnik-Manor received the BSc degree in mechanical engineering from the Technion in 1995, where she graduated summa cum laude. In 1998, she recieved the MSc degree with honors in computer science from the Weizmann Institute of Science. Currently, she is a PhD candidate in the Department of Computer Science and Applied Mathematics in the Weizmann Institute of Science, Rehovot, Israel. Her research focuses on the analysis of video sequences and its applications.

Michal Irani received the BSc degree in mathematics and computer science in 1985, the MSc, and PhD degrees in computer science in 1989 and 1994, respectively, all from the Hebrew University of Jerusalem. From 1993 to 1996, she was a member of the technical staff at the Vision Technologies Laboratory at David Sarnoff Research Center (SRI), Princeton, New Jersey. Currently, she is a member of the faculty in the Computer Science and Applied Mathematics Department at the Weizmann Institute of Science, Israel. Dr. Irani received the David Sarnoff Research Center Technical Achievement Award in 1994, the Yigal Allon three-year fellowship for outstanding young scientists in 1998, the ECCV best-paper prize in 2000. Her research interests are in the area of computer vision and video information processing. These include visual motion analysis, three-dimensional scene analysis, video information analysis, representation, and applications. Dr. Irani is currently an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and a member of the IEEE.