International Journal of Computer Vision 48(3), 173–194, 2002

Multi-Frame Correspondence Estimation Using Subspace Constraints*

MICHAL IRANI
Department of Computer Science and Applied Math, The Weizmann Institute of Science, 76100 Rehovot, Israel

Received October 18, 2000; Revised September 26, 2001; Accepted October 11, 2001

Abstract. When a rigid scene is imaged by a moving camera, the set of all displacements of *all points* across *multiple frames* often resides in a low-dimensional linear subspace. Linear subspace constraints have been used successfully in the past for recovering 3D structure and 3D motion information from multiple frames (e.g., by using the factorization method of Tomasi and Kanade (1992, *International Journal of Computer Vision*, 9:137–154)). These methods assume that the 2D correspondences have been precomputed. However, correspondence estimation is a fundamental problem in motion analysis. In this paper we show how the multi-frame subspace constraints can be used for *constraining* the 2D correspondence estimation process itself.

We show that the multi-frame subspace constraints are valid not only for affine cameras, but also for a variety of imaging models, scene models, and motion models. The multi-frame subspace constraints are first translated from constraints on correspondences to constraints directly on *image measurements* (e.g., image brightness quantities). These brightness-based subspace constraints are then used for estimating the correspondences, by requiring that all corresponding points across all video frames reside in the appropriate low-dimensional linear subspace.

The multi-frame subspace constraints are geometrically meaningful, and are not violated at depth discontinuities, nor when the camera-motion changes abruptly. These constraints can therefore replace heuristic constraints commonly used in optical-flow estimation, such as spatial or temporal smoothness.

Keywords: correspondence estimation, optical-flow, direct (gradient-based) methods, subspace constraints, factorization

1. Introduction

This paper presents an approach for simultaneous estimation of dense correspondences across multiple video frames. Optical flow (or "correspondence") estimation is usually applied to local image patches. Small image regions, however, carry very little information (this is known as the "aperture problem"), and the optical flow estimates obtained are hence noisy and/or partial. To overcome this problem, spatial smoothness constraints have been employed (e.g., Horn and Schunck (1981) and Anandan (1989)). However, these smoothness constraints are heuristic, and are violated especially at depth discontinuities. For a review and comparison of several of these optical flow techniques see Barron et al. (1992). Temporal smoothness constraints have also been introduced (e.g., Black and Anandan, 1991). These, however, are violated when the camera motion changes abruptly.

Other methods overcome the aperture problem by applying global model constraints (Hanna, 1991; Hanna and Okamoto, 1993; Bergen et al., 1992; Irani et al., 1994; Stein and Shashua, 1997; Black and Anandan, 1996; Bergen et al., 1992). This allows the use of large analysis windows (often the entire image), which do not suffer from lack of image information. These techniques, however, assume an a-priori restricted model of the world or of the camera motion. For example (Irani et al., 1994; Black and Anandan, 1996; Bergen et al., 1992) assume a planar (or very distant) world, resulting in constrained 2D parametric

*This research was supported by the Israel Science Foundation grant no. 153/99.

motion. While these methods are quite robust, they are restricted to 2D scenarios. The methods of Hanna (1991), Hanna and Okamoto (1993), Stein and Shashua (1997), Szeliski and Kang (1995) and Irani et al. (1999) assume a 3D world with dense 3D parallax, and use the epipolar constraints to constrain the correspondence estimation process. The 3D methods, however, fail when they are applied to "2D scenes",[1] because 2D scenarios form singular cases for the 3D-based algorithms. A hierarchy of such global 2D and 3D motion models is reviewed in Bergen et al. (1992). While these methods perform well when the restricted model assumptions are applicable, they fail when these are violated.

Also, most methods for correspondence/flow estimation have been restricted to *pairs* of frames. Those methods that use information from *multiple frames*, usually rely on temporal smoothness. The resulting estimates are hence noisy and are "over-smoothed". A few exceptions (e.g., Hanna and Okamoto (1993), Szeliski and Kang (1995) and Irani et al. (1999)) exploit geometric consistency across multiple frames already in the correspondence estimation process itself, but these methods rely on prior knowledge that the underlying model is a 3D world with dense 3D parallax, and will not be able to handle 2D scenarios (flat scenes, distant scenes, or camera on a tripod).

In this paper we develop a unified approach for simultaneously estimating correspondences across *multiple* frames by using information from all the frames, without assuming the prior choice of a model. Our approach is based on the observation that the set of all flow-fields across multiple frames (that image the same rigid scene) reside in a *low-dimensional linear subspace*. This is true despite the fact that different frames in the image sequence are obtained with different camera motions. The subspace constraints provide the additional constraints needed to resolve the ambiguity in image regions that suffer from the aperture problem. This is done *without* resorting to spatial or temporal smoothness. As opposed to smoothness constraints, the subspace constraints are geometrically meaningful, and are not violated at depth discontinuities or when camera-motion changes abruptly.

Linear subspace constraints have been used successfully in the past for recovering 3D information from *known* 2D correspondences (e.g., Tomasi and Kanade (1992) and Heeger and Jepson (1992)). In contrast, we use multi-frame linear subspace constraints to *constrain* the 2D correspondence estimation process itself, and not for recovering 3D information. Furthermore, we show that for a variety of world models (e.g., 2D planar scenes vs. general 3D scenes) and a variety of camera models (e.g, orthographic cameras vs. perspective cameras undergoing instantaneous motion) give rise to subspaces of very similar low dimensionalities. Because we employ subspace constraints based on the subspace *dimensionality* alone, these constraints can be used without prior knowledge of the underlying world or camera model.

This paper has four main contributions:

(i) We show that the set of all flow-fields across multiple frames (that image the same rigid scene) reside in a low-dimensional linear subspace. This is shown for a *variety of motion models, scene models, and imaging models*. Section 2 reviews the general idea, and Appendix A provides the detailed derivations for the different models.

(ii) We extend the notion of multi-frame subspace constraints on motion fields to *subspace constraints directly on image brightness quantities*. These brightness-based subspace constraints are derived in in Section 3.

(iii) We describe an algorithm which uses these multi-frame brightness subspace constraints to constrain the correspondence estimation process itself. In particular, we show how the two-frame Lucas and Kanade algorithm (Lucas and Kanade, 1981) can be extended to a multi-frame multi-point algorithm, which simultaneously uses all available spatio-temporal information in a short video sequence (Section 4).

(iv) We propose an approach to extend the applicability of the brightness-based subspace constraints to some non-linear varieties by employing the Plane + Parallax model (Section 6).

This work is a generalization of the approach presented by Zelnik and Irani (2000), where parametric transformations of planar surfaces (homographies) were simultaneously estimated across multiple frames using subspace constraints on the homography parameters. In contrast, the work here estimates general flow fields, and is not restricted to planar worlds. A preliminary version of this paper appeared in Irani (1999).

2. Subspace Constraints on Displacement Fields

Let $I_1, \ldots, I_{\mathcal{F}}$ denote a sequence of $\mathcal{F}$ frames taken by a moving camera with *arbitrary* 3D motions. All

frames are of the same size, and contain $\mathcal{N}$ pixels. Let I denote the *reference frame* in the sequence, i.e., the frame with respect to which all displacement fields will be estimated (e.g., the middle frame of the sequence). Let (u_{ij}, v_{ij}) denote the displacement of pixel (x_i, y_i) from the reference frame I to frame I_j $(i = 1..\mathcal{N},\ j = 1..\mathcal{F})$. Let $\mathbf{U}$ and $\mathbf{V}$ denote two $\mathcal{F} \times \mathcal{N}$ matrices constructed from the displacements of all the image points across all frames:

$$\mathbf{U} = \begin{bmatrix} u_{11}, u_{21}, \ldots, u_{\mathcal{N}1} \\ u_{12}, u_{22}, \ldots, u_{\mathcal{N}2} \\ \vdots \\ u_{1\mathcal{F}}, u_{2\mathcal{F}}, \ldots, u_{\mathcal{N}\mathcal{F}} \end{bmatrix}$$
$$\mathbf{V} = \begin{bmatrix} v_{11}, v_{21}, \ldots, v_{\mathcal{N}1} \\ v_{12}, v_{22}, \ldots, v_{\mathcal{N}2} \\ \vdots \\ v_{1\mathcal{F}}, v_{2\mathcal{F}}, \ldots, v_{\mathcal{N}\mathcal{F}} \end{bmatrix} \quad (1)$$

Each row in these matrices corresponds to a single frame, and each column corresponds to a single point.

We next argue that although the matrices $\mathbf{U}$ and $\mathbf{V}$ are large, *their ranks are very small*. This low-rank constraint will be exploited to constrain the correspondence estimation. In particular, we identify the ranks of the following two large matrices: $[\frac{\mathbf{U}}{\mathbf{V}}]_{2\mathcal{F}\times\mathcal{N}}$ (i.e., U and V are stacked vertically), and $[\mathbf{U} \mid \mathbf{V}]_{\mathcal{F}\times 2\mathcal{N}}$ (i.e., U and V are stacked horizontally). Note that each *column* in the matrix $[\frac{\mathbf{U}}{\mathbf{V}}]_{2\mathcal{F}\times\mathcal{N}}$ corresponds to the "*trajectory*" of a single pixel across all frames, while each *row* in the matrix $[\mathbf{U} \mid \mathbf{V}]_{\mathcal{F}\times 2\mathcal{N}}$ corresponds to the "*displacement field*" of all pixels between two frames (the reference frame and another frame).

These low-rank constraints are referred to as "*subspace constraints*", as only a few columns (or rows) suffice to span all other columns (or rows) of these large matrices. Linear subspace constraints are not new constraints, and have been successfully used in the past (e.g, by Tomasi and Kanade (1992)) for factoring known 2D correspondences into the unknown 3D information, namely, the 3D camera motion and the 3D shape. Tomasi and Kanade, however, assumed that the correspondences were known (i.e., that $\mathbf{U}$ and $\mathbf{V}$ are known). In contrast, in our case, we use the knowledge that the correspondences reside in a low-dimensional linear subspace (i.e., that the ranks of these matrices are low) in order to constrain the 2D correspondence estimation process itself, and not for recovering any 3D information. In Sections 3 and 4 we explain how to use these low-rank constraints in order to *constrain* the estimated correspondences (displacements).

We next derive the upper bounds on the ranks of these two large matrices under many different imaging and scene conditions, and show that these ranks are indeed low for many real-world scenarios (not only for affine cameras). It can be shown that the collection of all points across all views lie in a low-dimensional *variety* (Torr, 1998). Under full perspective projection and discrete views, this variety is non-linear (Anandan and Avidan, 2000). However, there are two cases in which this variety is linear: (i) when an "affine" camera model (Shapiro, 1995) is used (i.e., weak-perspective, or orthographic projection). This approximation is valid when the field of view is very small, and the depth fluctuations in the scene are small relative to the overall depth. (ii) when an "instantaneous motion model" is used (e.g., Longuet-Higgins and Prazdny (1980)). This approximation is valid when the camera rotation is small and the forward translation is small relative to the depth. The instantaneous model is a good approximation of the image motion over *short video segments*.[2] In some cases, such as in airborne video, this approximation is good also for very long sequences. The affine model (as opposed to the instantaneous motion model) is *not* restricted to short sequences, and applies as long as the field-of-view remains small throughout the entire sequence (e.g., as in the case of camera zoomed in on an object rotating on a turn-table).

We have derived the linear subspace (rank) constraints for these two types of models. In each case we have considered both a general 3D scene as well as a planar 2D scene, and calibrated as well as uncalibrated cameras. The resulting ranks for the various cases are summarized in Table 1. The derivations of these rank constraints can be found in Appendix A.

The results summarized in the Table 1 indicate that regardless of the camera projection model, and regardless of whether the scene is 2D (planar) or 3D, the ranks of the matrices $[\frac{\mathbf{U}}{\mathbf{V}}]$ and $[\mathbf{U} \mid \mathbf{V}]$ for short video sequences are guaranteed to be no more than 9. The ranks of these matrices are therefore significantly lower than their actual sizes ($2\mathcal{F} \times \mathcal{N}$ and $\mathcal{F} \times 2\mathcal{N}$, respectively). We will use these rank constraints alone to constrain the correspondence estimation process itself. No 3D information needs to be recovered in this process. (The 3D analysis in Appendix A is used only for deriving the upper bounds on the ranks of these matrices.)

Table 1. Upper bounds on the ranks of the matrices $[\frac{\mathbf{U}}{\mathbf{V}}]$ and $[\mathbf{U} \mid \mathbf{V}]$ for various camera projection models, world models, and motion models.

Camera	Scene	Calibration	Other restrictions	rank $[\frac{\mathbf{U}}{\mathbf{V}}]$	rank $[\mathbf{U} \mid \mathbf{V}]$
Orthographic/Affine	3D	Fixed/Varying	None	4	8
Orthographic/Affine	2D	Fixed/Varying	None	3	6
Perspective	3D	Fixed	Instantaneous motion	8	6
Perspective	3D	Varying	Instantaneous motion	9	9
Perspective	2D	Fixed	Instantaneous motion	6	6
Perspective	2D	Varying	Instantaneous motion	6	8

In all cases, the rank of these two large matrices never exceeds 9 (and in practice is usually much smaller—see text).

The *actual ranks* of these matrices may be even lower than the derived theoretical upper bounds. This happens, for example, when the camera motion is *spatially degenerate* (e.g., in the case of pure translation, or pure rotation), or when it is *temporally degenerate* (e.g., in the case of uniform motion across the sequence). These cases are very common in real video sequences. Our algorithm automatically detects the actual underlying ranks directly from image brightness quantities, prior to computing the correspondences (Section 4.3). In other words, the rank constraint is applied to a sequence of frames without the need to a-priori determine the underlying model, or its degeneracies. Thus, our algorithm exploits spatial or temporal degeneracies (or smoothness) when these do exist in the data, without making any prior assumptions about their existence.

3. Subspace Constraints on Image Brightness

The most straightforward way to take advantage of the subspace constraints in the correspondence estimation process is to take the following two steps: (a) First compute inter-frame displacements using any existing two-frame correspondence or flow estimation technique, and (b) then project the collection of all these flow fields into the appropriate lower dimensional linear subspace (i.e., project the large matrices $\mathbf{U}$ and $\mathbf{V}$ and their compounds onto the closest low-rank matrices). However, there are two problems with this two-stage approach: (i) The first step, namely, the unconstrained two-frame displacement estimation, is notoriously noisy. It will typically include noisy displacements that cannot be corrected even by subspace projection. Moreover, if a significant number of displacement (flow) vectors are severely corrupted, the estimated subspace itself will be incorrect, thus damaging all other displacements. (ii) All displacements are treated equally in the subspace projection, without any regard to their reliability. Yet, different points are tracked throughout the sequence with different reliability, depending on their local underlying image structure. For example, corner points can be tracked much more reliably than points on lines. Their estimated displacements should therefore not be treated equally in the subspace projection.

To avoid these two problems, we propose a one-stage approach for applying the low-dimensionality subspace constraints during the correspondence estimation process itself, and not after the fact. This is done by translating the above subspace constraints, which are currently defined on displacements, to *subspace constraints on image brightness quantities*. These constraints will also be shown to inherently give rise to *confidence-weighted subspace projection*. In particular, we derive two different brightness-based subspace constraints: (i) a *point-based* constraint, which is a generalization of the Brightness Constancy Equation into a multi-point multi-frame constraint (Section 3.1), and (ii) a *region-based* constraint, which is a generalization of the Lucas and Kanade flow constraint into a multi-point multi-frame constraint (Section 3.2). The benefits of using each of these constraints is explained in Section 4.

3.1. The Generalized Brightness Constancy Constraint

Let (x_i, y_i) be a pixel in the reference frame I, whose corresponding pixel in another frame I_j is $(x_i + u_{ij}, y_i + v_{ij})$. The Brightness Constancy Equation, which is defined on a single pixel between two frames, states that: $I(x_i, y_i) = I_j(x_i + u_{ij}, y_i + v_{ij})$. For small (u_{ij}, v_{ij}) we make the approximation: $I(x_i - u_{ij},$

$y_i - v_{ij}) = I_j(x_i, y_i)$. Expanding I to its first-order Taylor series around (x_i, y_i), leads to the linearized brightness constancy equation: $u_{ij} \cdot I_{x_i} + v_{ij} \cdot I_{y_i} + I_{t_{ij}} = 0$, where I_{x_i}, I_{y_i} are the spatial derivative of the *reference* frame I at pixel (x_i, y_i), and $I_{t_{ij}}$ is the temporal derivative: $I_{t_{ij}} = (I_j(x_i, y_i) - I(x_i, y_i))$.

However, in practice, the displacement (u_{ij}, v_{ij}) may not be small, especially when dealing with multiple frames. To increase its range of applicability to larger displacements (u_{ij}, v_{ij}), the linearization can be applied within an iterative (coarse-to-fine) refinement process (Bergen et al., 1992). Let (u^0_{ij}, v^0_{ij}) be the current estimate of the true displacement (u_{ij}, v_{ij}) during an iterative estimation process. Let

$$\begin{aligned} \Delta u_{ij} &= u_{ij} - u^0_{ij} \\ \Delta v_{ij} &= v_{ij} - v^0_{ij} \end{aligned} \tag{2}$$

be the residual displacement. The Brightness Constancy Equation can be rewritten as:

$$\begin{aligned} I(x_i, y_i) &= I_j(x_i + u_{ij}, y_i + v_{ij}) \\ &= I_j\big(x_i + u^0_{ij} + \Delta u_{ij}, y_i + v^0_{ij} + \Delta v_{ij}\big). \end{aligned} \tag{3}$$

For small $(\Delta u_{ij}, \Delta v_{ij})$ we make the approximation:[3]

$$I(x_i - \Delta u_{ij}, y_i - \Delta v_{ij}) = I_j\big(x_i + u^0_{ij}, y_i + v^0_{ij}\big). \tag{4}$$

This approximation allows us to perform the linearization on the reference frame[4] I (assuming small $(\Delta u_{ij}, \Delta v_{ij})$):

$$\begin{aligned} \Delta u_{ij} I_{x_i} + \Delta v_{ij} I_{y_i} + \big(I_j\big(x_i + u^0_{ij}, y_i + v^0_{ij}\big) \\ - I(x_i, y_i)\big) = 0 \end{aligned} \tag{5}$$

Because the multi-frame subspace constraints (see Section 2) are defined on the displacements (u_{ij}, v_{ij}) and not on the increments $(\Delta u_{ij}, \Delta v_{ij})$, we substitute the expression for $(\Delta u_{ij}, \Delta v_{ij})$ from Eq. (2) into Eq. (5), leading to the following form of the brightness constancy equation, which we will use in throughout this paper:

$$u_{ij} \cdot I_{x_i} + v_{ij} \cdot I_{y_i} \;=\; -I^0_{t_{ij}}, \tag{6}$$

where,

$$\begin{aligned} I^0_{t_{ij}} \;=\; \big(I_j\big(x_i + u^0_{ij}, y_i + v^0_{ij}\big) - I(x_i, y_i) - u^0_{ij} I_{x_i} \\ - v^0_{ij} I_{y_i}\big). \end{aligned}$$

Equation (6) provides a single *line constraint* on the two unknowns u_{ij}, v_{ij}, and hence is not sufficient for uniquely determining the unknown displacement of a single pixel between two frames.

Let $I_1, \ldots, I_{\mathcal{F}}$ be a sequence of frames, as defined in Section 2. The collection of all Brightness Constancy Constraints (Eq. (6)) of all image points across all image frames can be compactly written in a single *matrix form* as:

$$[\mathbf{U} \mid \mathbf{V}]_{(\mathcal{F} \times 2\mathcal{N})} \cdot \left[\frac{\mathbf{F_X}}{\mathbf{F_Y}}\right]_{(2\mathcal{N} \times \mathcal{N})} = \mathbf{F_T}_{(\mathcal{F} \times \mathcal{N})} \tag{7}$$

where $\mathbf{F_X}$ and $\mathbf{F_Y}$ are $\mathcal{N} \times \mathcal{N}$ diagonal matrices with the spatial x- and y-derivatives of the reference frame I in their diagonal:

$$\mathbf{F_X} = \begin{bmatrix} I_{x_1} & 0 & \ldots & 0 \\ 0 & I_{x_2} & \ldots & 0 \\ & \vdots & & \\ 0 & 0 & \ldots & I_{x_{\mathcal{N}}} \end{bmatrix}$$

$$\mathbf{F_Y} = \begin{bmatrix} I_{y_1} & 0 & \ldots & 0 \\ 0 & I_{y_2} & \ldots & 0 \\ & \vdots & & \\ 0 & 0 & \ldots & I_{y_{\mathcal{N}}} \end{bmatrix}$$

and $\mathbf{F_T}$ is an $\mathcal{F} \times \mathcal{N}$ matrix of the *temporal* derivatives (of all image points across all frames) estimated at the current stage of the iterative process, namely:

$$\mathbf{F_T} = \begin{bmatrix} -I^0_{t_{11}} & -I^0_{t_{21}} & \ldots & -I^0_{t_{\mathcal{N}1}} \\ -I^0_{t_{12}} & -I^0_{t_{22}} & \ldots & -I^0_{t_{\mathcal{N}2}} \\ & \vdots & & \\ -I^0_{t_{1\mathcal{F}}} & -I^0_{t_{2\mathcal{F}}} & \ldots & -I^0_{t_{\mathcal{N}\mathcal{F}}} \end{bmatrix}$$

The matrices $\mathbf{F_X}$, $\mathbf{F_Y}$, and $\mathbf{F_T}$, contain only *measurable image quantities*. The matrices $\mathbf{U}$ and $\mathbf{V}$ contain all the *unknown displacements*. Note that all flow-vectors corresponding to a single scene point share the same *spatial* derivatives I_{x_i}, I_{y_i} (as these are computed in the reference frame I, and are independent of the other frame j). However, their *temporal* derivatives $I_{t_{ij}}$ *do* vary from frame to frame (and in every iteration). We refer to the multi-point multi-frame Eq. (7) as the *the Generalized Brightness Constancy Equation*.

Note that when no additional information on $[\mathbf{U} \mid \mathbf{V}]$ is used, then Eq. (7) is no more than the collection of all the individual two-frame brightness constancy

equations of Eq. (6). However, this matrix formulation allows us to apply rank constraints directly to measurable image quantities. For example, $rank([\mathbf{U} \mid \mathbf{V}]) \leq r$ implies that $rank(\mathbf{F_T}) \leq r$. We can therefore apply the rank constraint directly to the data matrix $\mathbf{F_T}$ *prior* to solving for the displacements $\mathbf{U}$ and $\mathbf{V}$. This formulation, as well as the one which is next described in Section 3.2, form the basis for our direct multi-point multi-frame algorithm, which is described in Section 4. There are two important observations we would like to stress:

Observation-I: Subspace Constraints on Normal Flow: Note that the matrix $\mathbf{F_T}$ is no more than the collection of all the individual scaled *normal flow* values $\{-I_{t_{ij}}\}$. It is easy to see from Eq. (6) that $-I_{t_{ij}}$ is a scaled normal flow:

$$-I_{t_{ij}} = u_{ij} \cdot I_{x_i} + v_{ij} \cdot I_{y_i} = (u_{ij}, v_{ij}) \cdot \nabla I_i$$

namely, the projection of the displacement vector (u_{ij}, v_{ij}) onto the direction of the gradient ∇I_i (the normal), scaled by the gradient magnitude. This means that the subspace constraints are valid not only on the collection of displacements $\{(u_{ij}, v_{ij})\}$, but also on the collection of normal-flows $\{(u_{ij}, v_{ij}) \cdot \nabla I_i\}$. This, of course, is an advantage, as the normal-flows can be computed much more reliably than the full flow-vectors. One manifestation of the normal-flow information is in the measurement matrix $\mathbf{F_T}$ above. Moreover, the normal-flow is scaled by the gradient magnitude. This leads to the next observation.

Observation-II: Confidence-Weighted Subspace Projection: From Eq. (7) we can see that applying the rank constraint to $\mathbf{F_T}$ is in fact geometrically equivalent to applying the rank constraint directly to the displacements matrix $[\mathbf{U} \mid \mathbf{V}]$, but after first weighting the individual displacements (u_{ij}, v_{ij}) with their corresponding *directional* "*confidences*" (I_{x_i}, I_{y_i}). A larger horizontal derivative I_{x_i} indicates a more confident u_{ij}, and similarly a larger vertical derivative I_{y_i} indicates a more confident v_{ij}. This means that more reliable displacements will have more influence in the subspace projection process, while less reliable displacements will have smaller influence. Therefore, applying the rank constraint to $\mathbf{F_T}$ has the effect of *confidence-weighted subspace projection* of all the displacements, yet this is done *prior* to computing the correspondences, hence avoiding the errors introduced in two-frame flow-estimation methods.

While the matrix $\mathbf{F_T}$ contains point-based measurements, we next show how the above analysis can be extended to confidence-weighted subspace constraints on region-based measurements.

3.2. *The Generalized Lucas & Kanade Constraint*

Lucas and Kanade (1981) extended the pixel-based brightness constancy constraints of Eq. (6) to a local region-based constraint, by assuming a uniform displacement in very small windows (typically 3×3 or 5×5). Then, for each pixel (x_i, y_i), they solve for its displacement vector (u_{ij}, v_{ij}) by minimizing the following local error measure $E(u_{ij}, v_{ij})$ within its neighborhood (window) W_i:

$$E(u_{ij}, v_{ij}) = \sum_{k \in W_i} \left(u_{ij} \cdot I_{x_k} + v_{ij} \cdot I_{y_k} + I^0_{t_{kj}}\right)^2$$

(The Lucas and Kanade equation was slightly modified to fit our iterative notation). Differentiating the error $E(u_{ij}, v_{ij})$ with respect to u_{ij} and v_{ij}, and setting these derivatives to zero, yields a set of two linear equations in the two unknown displacement components (u_{ij}, v_{ij}) for each pixel:

$$[u_{ij} v_{ij}]_{1\times 2} \cdot \begin{bmatrix} a_i & b_i \\ b_i & c_i \end{bmatrix}_{\mathbf{2\times 2}} = [g_{ij} h_{ij}]_{1\times 2} \qquad (8)$$

where $a_i, b_i, c_i, g_{ij}, h_{ij}$ are measurable image quantities:

$$a_i = \sum_k (I_{x_k})^2, \quad b_i = \sum_k (I_{x_k} \cdot I_{y_k}), \quad c_i = \sum_k (I_{y_k})^2,$$
$$g_{ij} = -\sum_k \left(I_{x_k} \cdot I^0_{t_{kj}}\right), \quad h_{ij} = -\sum_k \left(I_{y_k} \cdot I^0_{t_{kj}}\right).$$

a_i, b_i, c_i are point-dependent quantities computed in the reference image I, and are independent of the frame (time) j. g_{ij}, h_{ij} depend on both the point i and the frame j.

Equation (8) provides *two* equations on the two unknowns u_{ij}, v_{ij}, as opposed to Eq. (6), which provides only one equation. This is because of the uniform-displacement assumption within the local windows. While this assumption imposes a type of *local* smoothness constraint, it only affects the accuracy of the flow estimation within the small window, but does not propagate these errors to other image regions (as opposed to *global* smoothness (e.g., Horn and Schunck (1981)).

The vector (u_{ij}, v_{ij}) therefore has a unique solution when the coefficient matrix $\left[\begin{smallmatrix} a_i & b_i \\ b_i & c_i \end{smallmatrix}\right]$ is not singular (e.g., for corners and textured areas). For image regions where the local information is insufficient (e.g., edges), the matrix will be singular. In these regions the flow vector (u_{ij}, v_{ij}) cannot be uniquely determined even by the Lucas and Kanade algorithm. Under Gaussian noise assumptions, the matrix $\left[\begin{smallmatrix} a_i & b_i \\ b_i & c_i \end{smallmatrix}\right]$ in Eq. (8) can be shown to be the *posterior inverse covariance matrix* of the estimated flow vector (u_{ij}, v_{ij}).

Now, considering multiple-points over multiple-frames. As in the case of the Generalized Brightness Constancy Eq. (7), all the flow-vectors (u_{ij}, v_{ij}) from a reference pixel (x_i, y_i) in I to all other frames I_j $(j = 1..\mathcal{F})$ share the *same coefficient (inverse covariance) matrix* $\left[\begin{smallmatrix} a_i & b_i \\ b_i & c_i \end{smallmatrix}\right]$ in their two-frame Lucas and Kanade constraints (Eq. (8)). Hence, all the Lucas and Kanade constraints on *all points* $(i = 1..\mathcal{N})$ across *all frames* $(j = 1..\mathcal{F})$ can be compactly written in a single matrix form as:

$$[\mathbf{U} \mid \mathbf{V}]_{(\mathcal{F}\times 2\mathcal{N})} \cdot \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{B} & \mathbf{C} \end{array}\right]_{(2\mathcal{N}\times 2\mathcal{N})} = [\mathbf{G} \mid \mathbf{H}]_{(\mathcal{F}\times 2\mathcal{N})} \quad (9)$$

where $\mathbf{U}$ and $\mathbf{V}$ are as defined in Eq. (1). The three $\mathcal{N} \times \mathcal{N}$ diagonal matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ are constructed from the coefficient values a_i, b_i, c_i, respectively:

$$\mathbf{A} = \begin{bmatrix} a_1 & 0 & \dots & 0 \\ 0 & a_2 & \dots & 0 \\ & \vdots & & \\ 0 & 0 & \dots & a_{\mathcal{N}} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} b_1 & 0 & \dots & 0 \\ 0 & b_2 & \dots & 0 \\ & \vdots & & \\ 0 & 0 & \dots & b_{\mathcal{N}} \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} c_1 & 0 & \dots & 0 \\ 0 & c_2 & \dots & 0 \\ & \vdots & & \\ 0 & 0 & \dots & c_{\mathcal{N}} \end{bmatrix}$$

The two $\mathcal{F} \times \mathcal{N}$ matrices $\mathbf{G}$ and $\mathbf{H}$ are constructed from the values g_{ij}, h_{ij}:

$$\mathbf{G} = \begin{bmatrix} g_{11} & g_{21} & \dots & g_{\mathcal{N}1} \\ g_{12} & g_{22} & \dots & g_{\mathcal{N}2} \\ & \vdots & & \\ g_{1\mathcal{F}} & g_{2\mathcal{F}} & \dots & g_{\mathcal{N}\mathcal{F}} \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{21} & \dots & h_{\mathcal{N}1} \\ h_{12} & h_{22} & \dots & h_{\mathcal{N}2} \\ & \vdots & & \\ h_{1\mathcal{F}} & h_{2\mathcal{F}} & \dots & h_{\mathcal{N}\mathcal{F}} \end{bmatrix}$$

We refer to the multi-point multi-frame Eq. (9) as the *the Generalized Lucas and Kanade Equation*.

When no additional information on $[\mathbf{U} \mid \mathbf{V}]$ is used, then Eq. (9) is no more than the collection of all the individual two-frame equations of Eq. (8). However, as before, if we know that $rank([\mathbf{U} \mid \mathbf{V}]) \leq r$, it entails that $rank([\mathbf{G} \mid \mathbf{H}]) \leq r$. Since $[\mathbf{G} \mid \mathbf{H}]$ is a matrix constructed from *known measurable image quantities*, applying the rank constraint to it *prior* to solving for $[\mathbf{U} \mid \mathbf{V}]$ will constrain the flow estimation process itself. The geometric interpretation of this operation is explained below.

Observation: Covariance-Weighted Subspace Projection: From Eq. (9) we can see that applying the rank constraint to $[\mathbf{G} \mid \mathbf{H}]$ is in fact equivalent to applying the rank constraint directly to the displacements matrix $[\mathbf{U} \mid \mathbf{V}]$, but after first weighting the individual displacements (u_{ij}, v_{ij}) with their corresponding individual inverse covariance matrices $\left[\begin{smallmatrix} a_i & b_i \\ b_i & c_i \end{smallmatrix}\right]$. Applying the rank constraint to $[\mathbf{G} \mid \mathbf{H}]$ therefore has the effect of *covariance-weighted subspace projection*[5] of all the displacements, yet this is done *prior* to computing the correspondences, hence avoiding the errors introduced in two-frame flow-estimation methods. While the confidence-weighted subspace-projection presented in Section 3.1 is founded on a *point-based* directional-confidence measure (namely, the spatial gradient measured at each pixel), here the confidence-weighted subspace-projection is founded on a *region-based* directional-confidence measure (namely, the inverse covariance matrix measured at each pixel). This means that more reliable local image regions ("features") will have more influence in the subspace projection process, while less reliable image regions will have smaller influence. In general, we found the region-based subspace constraint to be more robust than the point-based constraint. This is used to constrain the correspondence estimation process which is described in Section 4.

4. A Multi-Frame Multi-Point Algorithm

We first explain the significance of using subspace constraints on both matrices $[\mathbf{U} \mid \mathbf{V}]$ and $\left[\frac{\mathbf{U}}{\mathbf{V}}\right]$. The subspace

constraint on $[\mathbf{U} \mid \mathbf{V}]$ is used for cleaning noise in the image-measurements in a confidence-weighted manner (Section 4.1). However, the subspace constraint on $[\mathbf{U} \mid \mathbf{V}]$ alone is not sufficient for resolving the aperture problem. The subspace constraint on $[\frac{\mathbf{U}}{\mathbf{V}}]$ is therefore also employed, in order to resolve the aperture problem (Section 4.2). An example of how both constraints can be integrated into a single multi-point multi-frame correspondence estimation algorithm is described in Section 4.3. The use of these subspace constraints, however, is more general and is not limited to this particular algorithm. The issue of automatic rank detection is discussed in Section 4.4.

The subspace constraints can be applied either to the point-based measurements (matrix $\mathbf{F_T}$) or the region-based measurements (matrix $[\mathbf{G} \mid \mathbf{H}]$), leading to two slightly different algorithms. Both possibilities will be discussed in Sections 4.1 and 4.2 with their advantages and disadvantages. However, we found the region-based algorithm to be more robust, and is therefore the algorithm we have used to produce the results reported in this paper. This is also the algorithm summarized in Section 4.3.

4.1. Noise Reduction in Image Measurements

Let r_1 denote the actual rank of $[\mathbf{U} \mid \mathbf{V}]$. We know that $r_1 \leq 9$ (see Table 1), but in practice, the actual rank of these matrices may be significantly lower than the theoretical upper bound of 9. According to Eqs. (7) and (9), the ranks of the measurement matrices $\mathbf{F_T}$ and $[\mathbf{G} \mid \mathbf{H}]$, should also be at most r_1. In practice, due to noise in image brightness, these matrices usually have higher ranks. However, inspection of the rate of decay of the singular values of these measurement matrices allows us to automatically detect their actual rank (see Section 4.4). The matrices $\mathbf{F_T}$ and $[\mathbf{G} \mid \mathbf{H}]$, are hence projected onto lower-rank matrices $\hat{\mathbf{F}}_\mathbf{T}$ and $[\hat{\mathbf{G}} \mid \hat{\mathbf{H}}]$ of rank r_1, thus cleaning the image measurements from noise, and automatically detecting the actual rank of $[\mathbf{U} \mid \mathbf{V}]$.

The rank-reduction process inhibits noisy measurements in the measurement matrices. It can be directly applied either to the point-based measurements $\mathbf{F_T}$, or to the region-based measurements $[\mathbf{G} \mid \mathbf{H}]$, and corresponds to applying *confidence-weighted subspace projection* on the flow vectors *prior* to computing them (see Sections 3.1 and 3.2).

One source of noise in the brightness-based measurement matrices ($\mathbf{F_T}$ or $[\mathbf{G} \mid \mathbf{H}]$) is the violation of the brightness constancy constraint at occluded pixels. However, boundary pixels are very sparse relative to the total number of pixels in the image, and therefore the resulting increase in the rank of $\mathbf{F_T}$ or $[\mathbf{G} \mid \mathbf{H}]$ can be treated as noise, and is handled by the subspace projection. (Note that the subspace constraints on *displacements fields* are geometrically meaningful also at depth discontinuities, and are therefore useful for handling such violations of the *brightness-based* subspace constraints.)

While inhibiting the noise in the image measurements improves the quality of the recovered correspondences significantly, the rank constraint on the matrix $[\mathbf{U} \mid \mathbf{V}]$ alone does not suffice to resolve the aperture problem, even when applied directly to $\mathbf{U}$ and $\mathbf{V}$. For example, let $(u_{i1}, v_{i1}) \ldots (u_{i\mathcal{F}}, v_{i\mathcal{F}})$ be the displacements of a point i across all frames. Then the i-th column in the matrix $\mathbf{U}$ is $[u_{i1} \ldots u_{i\mathcal{F}}]^t$, and the i-th column in the matrix $\mathbf{V}$ is $[v_{i1} \ldots v_{i\mathcal{F}}]^t$. If we now multiply the i-th column of $\mathbf{U}$ by an arbitrary scale factor, and the i-th column of $\mathbf{V}$ by a *different* scale factor, then the subspace spanned by the columns of $[\mathbf{U} \mid \mathbf{V}]$ remains unchanged (and so does the rank of the matrix). Such scaling, however, changes the proportion between the u components and the v components of the displacements of point i across all frames. In other words, constraining one component of all displacements of a point i (e.g., the v's) does not suffice to uniquely constrain the other component of all displacements of a point i (e.g., the u's). Similarly, constraining all the normal-flow values of the (u, v)'s does not suffice to uniquely constrain the tangent-flow values of these displacements vectors. Therefore, the subspace constraint on the matrix $[\mathbf{U} \mid \mathbf{V}]$ alone does not suffice to resolve the aperture problem.

This, however, is not the case with the subspace constraint on the matrix $[\frac{\mathbf{U}}{\mathbf{V}}]$. For example, a scaling the i-th column of $\mathbf{U}$ will require a scaling of the i-th column of $\mathbf{V}$ by the *same* scale factor, in order to maintain the subspace of $[\frac{\mathbf{U}}{\mathbf{V}}]$. We next show how adding the rank constraint on the matrix $[\frac{\mathbf{U}}{\mathbf{V}}]$ resolves the aperture problem.

4.2. Eliminating the Aperture Problem

We use the rank constraint on $[\frac{\mathbf{U}}{\mathbf{V}}]$ to determine the missing components of flow vectors at pixels with insufficient local image structure. Let r_2 denote the rank of $[\frac{\mathbf{U}}{\mathbf{V}}]$ This implies that there is a decomposition:

$$\left[\frac{\mathbf{U}}{\mathbf{V}}\right]_{(2\mathcal{F}\times\mathcal{N})} = \mathbf{K}_{(2\mathcal{F}\times r_2)} \cdot \mathbf{L}_{(r_2\times\mathcal{N})} \tag{10}$$

The columns of $\mathbf{K}$ form a basis which spans the subspace of all columns of $[\frac{\mathbf{U}}{\mathbf{V}}]$. The columns of $\mathbf{L}$ are the coefficients in the linear combination (i.e., the i-th column of L contains the r_2 coefficients which are needed to generate the i-th column of $[\frac{\mathbf{U}}{\mathbf{V}}]$ from the r_2 basis columns of $\mathbf{K}$). This decomposition is of course not unique, as for any invertible $r_2 \times r_2$ matrix $\mathbf{M}$: $[\frac{\mathbf{U}}{\mathbf{V}}] = (\mathbf{KM^{-1}}) \cdot (\mathbf{ML})$ is also a valid decomposition. Theoretically, if there are more than r_2 pixels whose correspondences across all frames can be reliably computed, and whose trajectories across all frames are linearly independent, then these r_2 trajectories could be used to generate a basis $\mathbf{K}$. In practice, we use all available information from all reliable points to produce the best basis. This is explained next.

Let S_0 be a set of all the highly reliable pixels in the reference frame, namely, those pixels that have enough local image structure (i.e., pixels whose 2×2 inverse covariance matrix $\left[\begin{smallmatrix} a_i & b_i \\ b_i & c_i \end{smallmatrix}\right]$ is non-singular). Let $[\hat{\mathbf{G}}_\mathbf{0} \mid \hat{\mathbf{H}}_\mathbf{0}]$ be matrix containing only the columns in $[\hat{\mathbf{G}} \mid \hat{\mathbf{H}}]$ (of Eq. (9)) that correspond to the pixels in S_0. Similarly, let $[\frac{\mathbf{A_0} \mid \mathbf{B_0}}{\mathbf{B_0} \mid \mathbf{C_0}}]$ be the corresponding submatrix of $[\frac{\mathbf{A} \mid \mathbf{B}}{\mathbf{B} \mid \mathbf{C}}]$. In general, $[\frac{\mathbf{A} \mid \mathbf{B}}{\mathbf{B} \mid \mathbf{C}}]$ is not invertible (due to pixels which suffer from the aperture problem). However, because all the individual 2×2 inverse covariance matrices $\left[\begin{smallmatrix} a_i & b_i \\ b_i & c_i \end{smallmatrix}\right]$ in S_0 are non-singular, the matrix $[\frac{\mathbf{A_0} \mid \mathbf{B_0}}{\mathbf{B_0} \mid \mathbf{C_0}}]$ is now invertible. We can therefore estimate the displacements of all the reliable pixels (i.e., of all pixels in S_0) across all frames by solving:

$$[\mathbf{U_0} \mid \mathbf{V_0}] = [\hat{\mathbf{G}}_\mathbf{0} \mid \hat{\mathbf{H}}_\mathbf{0}] \cdot \left[\frac{\mathbf{A_0} \mid \mathbf{B_0}}{\mathbf{B_0} \mid \mathbf{C_0}}\right]^{-1}$$

(Note that because of the diagonal structure of $\mathbf{A_0}, \mathbf{B_0}, \mathbf{C_0}$, the matrix $[\frac{\mathbf{A_0} \mid \mathbf{B_0}}{\mathbf{B_0} \mid \mathbf{C_0}}]^{-1}$ consists of the components of the individual inverse 2×2 matrices $\left[\begin{smallmatrix} a_i & b_i \\ b_i & c_i \end{smallmatrix}\right]^{-1}$, and can therefore be easily estimated without having to invert the large matrix).

The columns of the reliable trajectories $[\frac{\mathbf{U_0}}{\mathbf{V_0}}]$ are used to generate a basis $\mathbf{K}$. This is done by taking the first r_2 eigenvectors corresponding to the r_2 largest eigenvalues of the smaller $2\mathcal{F} \times 2\mathcal{F}$ matrix $([\frac{\mathbf{U_0}}{\mathbf{V_0}}] \cdot [\frac{\mathbf{U_0}}{V_0}]^T)$. In fact, the actual rank r_2 of $[\frac{\mathbf{U}}{\mathbf{V}}]$ (which is usually lower than the theoretical upper bound of 9) is automatically detected by inspection of the rate of decay of the eigenvalues (see Section 4.4).

Once a basis $\mathbf{K}$ has been computed, the only remaining unknowns are the components of the coefficient matrix $\mathbf{L}$. Determining $\mathbf{L}$ determines both $\mathbf{U}$ and $\mathbf{V}$ uniquely, and thus resolves the aperture problem. Therefore, once a basis $\mathbf{K}$ has been computed, the number of unknowns shrinks from the original number of $2\mathcal{FN}$ unknowns (i.e., the unconstrained displacements $\{(u_{ij}, v_{ij})\}$) to $\mathcal{N} r_2$ unknowns, which is the size of the unknown coefficient-matrix $\mathbf{L}$ (see Eq. (10)). In other words, once the number of frames $\mathcal{F}$ exceeds $\frac{r_2}{2}$ ($\mathcal{F} > \frac{r_2}{2}$), then there is already a reduction in the number of unknowns. Since $r_2 \leq 9$, therefore for video sequences of five or more frames ($\mathcal{F} \geq 5$) there is already a reduction in the number of unknowns.

In practice, in short video sequences the rank is usually much smaller than 9. This is because the camera motion from frame to frame does not tend to be fully three dimensional. It tends to be dependent (often uniform), and hence usually does not span the full-ranked subspace. Note, however, that we do not make such an assumption a-priori, but automatically detect these cases. Therefore, a reduction in the number of unknowns is usually achieved for fewer frames than five.

We next show how the unknown coefficient matrix $\mathbf{L}$ is computed. Let $\mathbf{K} = [\frac{\mathbf{K_U}}{\mathbf{K_V}}]$, where $\mathbf{K_U}$ and $\mathbf{K_V}$ are the upper and lower halves of the matrix $\mathbf{K}$. Then according to Eq. (10):

$$\mathbf{U} = \mathbf{K_U L}, \quad \mathbf{V} = \mathbf{K_V L} \tag{11}$$

Plugging the decomposition of Eq. (11) into Eq. (7) leads to a set of $\mathcal{FN}$ *linear* equations in the $\mathcal{N} r_2$ unknowns:

$$[\mathbf{K_U L} \mid \mathbf{K_V L}] \cdot \left[\frac{\mathbf{F_X}}{\mathbf{F_Y}}\right] = \hat{\mathbf{F}}_\mathbf{T}. \tag{12}$$

This set of equations is over-constrained if the number of equations (determined by the size of $\hat{\mathbf{F}}_\mathbf{T}$) exceeds the number of unknowns (determined by the size of $\mathbf{L}$), which happens when the number of frames $\mathcal{F}$ exceeds the rank r_2. Once again, since $r_2 \leq 9$, and in practice for short video sequences is usually much smaller than 9, therefore for very few frames the system becomes over-determined.

Note that both $\mathbf{U}$ and $\mathbf{V}$ share the same coefficients $\mathbf{L}$ in the decomposition of the matrix $[\frac{\mathbf{U}}{\mathbf{V}}]$ (which is not the case in the decomposition of the matrix $[\mathbf{U} \mid \mathbf{V}]$). Equation (12) combines the constraints on both matrices into a single matrix equation. This is the key to resolving the aperture problem, as the unknown coefficient matrix $\mathbf{L}$ is computed directly from the matrix $\hat{\mathbf{F}}_\mathbf{T}$ in Eq. (12), i.e., directly from normal flow information (which is the only reliable information for pixels which

suffer from the aperture problem). However, once $\mathbf{L}$ and $\mathbf{K}$ are known, $\mathbf{U}$ and $\mathbf{V}$ can be uniquely determined from Eq. (11).

This bears resemblance to the epipolar line-constraint, which, when combined with the brightness constancy constraint, uniquely resolves the aperture problem (Hanna and Okamoto, 1993; Szeliski and Kang, 1995; Irani et al., 1999; Stein and Shashua, 1997). However, unlike the epipolar constraint, our constraint is *implicit*, does not require the recovery of the epipolar geometry. It applies both to "2D scene" (where the induced motion is a homography and explicit epipolar geometry cannot be recovered, such as in Fig. 2), as well as to "3D scenes" (where 3D parallax is also induced in the video sequence, such as in Fig. 1).

When the underlying motion model is known (e.g., a 2D or a 3D model), *explicit* geometric constraints can be (and have been) used to constrain the correspondence estimation across multiple frames. For example, explicit epipolar constraints have been used in the case of "3D scenes" (e.g., Hanna and Okamoto (1993), Szeliski and Kang (1995), Irani et al. (1999) and Stein and Shashua (1997)), and explicit parametric constraints on homographies have been used in the case of "2D scenes" (e.g., Zelnik-Manor and Irani (2000)). These explicit geometric constraints usually provide tighter constraints on the displacement fields than our implicit linear subspace constraints. The explicit constraints, however, are scene-specific (i.e., are applicable either to 2D or to 3D scenes, but not to both), and thus require prior model selection. The subspace constraints, on the other hand, can handle both types of scenes, and thus provide *a unified approach* to correspondence estimation in 2D and 3D scenes.

Similarly, plugging the decomposition of Eq. (11) into Eq. (9) leads to an alternative set of linear equations, with *twice* as many equations ($2\mathcal{F}N$ equations, determined by the size of $[\hat{\mathbf{G}} \mid \hat{\mathbf{H}}]$) in the same number of unknowns (the $\mathcal{N}r_2$ unknown components of $\mathbf{L}$):

$$[\mathbf{K}_U\mathbf{L} \mid \mathbf{K}_V\mathbf{L}] \cdot \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{B} & \mathbf{C} \end{array}\right] = [\hat{\mathbf{G}} \mid \hat{\mathbf{H}}]. \qquad (13)$$

This set of equations is thus over-constrained if the number of frames $\mathcal{F}$ is larger than $\frac{r_2}{2}$.

Each of the two abovementioned options (the point-based and the region-based) has its advantages: The region-based approach of Eq. (13) is numerically

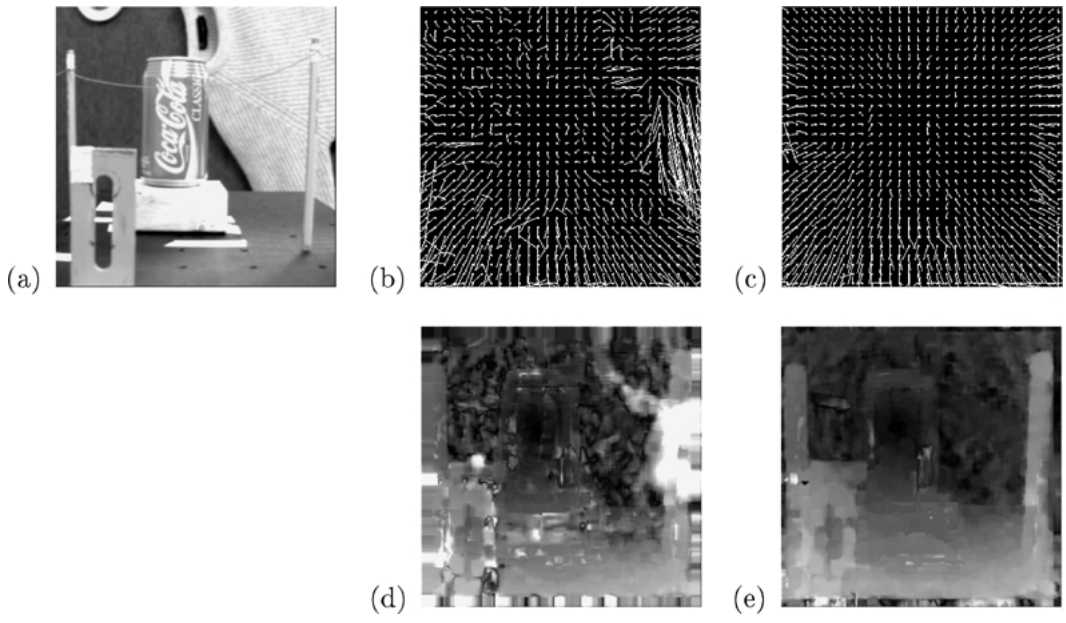

Figure 1. Real image sequence (the NASA coke-can sequence). (a) One frame from a 27-frame sequence of a forward moving camera in a 3D scene. (b) Flow field generated with the two-frame Lucas and Kanade algorithm. Note the errors in the right hand side, where there is depth discontinuity (pole in front of sweater), as well as the aperture problem. (c) The flow field for the corresponding frame generated by the multi-frame constrained algorithm. Note the good recovery of flow in those regions. (d, e) The flow magnitude at every pixel. This display provides a higher resolution display of the error. Note the clear depth discontinuities in the multi-frame flow image. The flow values on the coke can are very small, because the camera FOE is in that area.

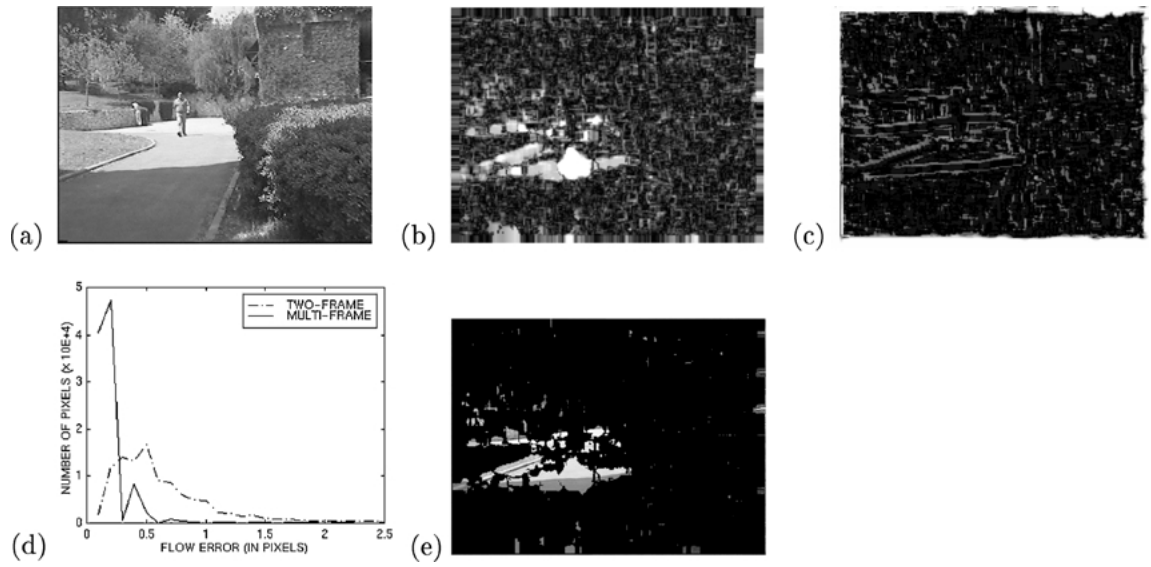

Figure 2. Synthetic sequence with ground truth—a quantitative comparison. (a) One out of a 10-frame sequence. The sequence was synthetically generated by applying a set of 3-D consistent homographies to warp a single image. This provides ground truth on the flow. (b, c) Error maps showing magnitudes of residual errors between the ground truth flow and the computed flow field. (b) Shows errors for the two-frame Lucas and Kanade algorithm. (c) Shows errors for the multi-frame constrained algorithm for the corresponding frame. Brighter values correspond to larger errors. (d) A histogram of the errors in both flow fields. Flow values at image boarders were ignored. In the multi-frame method almost all errors are smaller than 0.2 pixel, and all are smaller than 0.5 pixel. In the two-frame method, most flow vectors have an error of *at least* 0.5 pixel. (e) The image regions for which the errors in the *two-frame* method exceeded 1.0 pixel. These, as expected, correspond to areas which suffer from the aperture problem. The subspace constrained algorithm accurately recovered the flow even in those regions.

more stable, because of the local confidence-weighted averaging over the small (3×3 or 5×5) windows from the Lucas & Kanade algorithm, and because there are twice as many equations. But this benefit comes with the price of lower spatial resolution in the recovery of displacement fields. On the other hand, the point-based approach of Eq. (12) provides half as many equations, but allows for *higher spatial resolution* in the displacement field, since it does not use the small window averaging. In our experiments we found that it was preferable to trade high resolution information for numerical stability.

Note that there is a significant difference between using window averaging for increased numerical stability, and using smoothness constraints as a *necessary* constraint without which the problem is underdetermined (such as in the case of Horn and Schunk (1981), or in the case of the two-frame Lucas and Kanade (1981)). In the former case (our case), the necessary constraints are already provided by the subspace constraints. The window averaging only adds further conditioning to the problem, but is not a necessity.

In the current implementation of our algorithm we used the region-based approach of Eq. (13). The algorithm is summarized next.

4.3. *A Summary of the Multi-Point Multi-Frame Algorithm*

1. Construct a Gaussian pyramid for all image frames.
2. For each iteration in each pyramid level do:
   (a) Compute matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{G}$, $\mathbf{H}$.
   (b) Project $[\mathbf{G} \mid \mathbf{H}]$ onto lower-rank (r_1) matrix $[\hat{\mathbf{G}} \mid \hat{\mathbf{H}}]$.
   (c) Compute reliable displacement estimates only for reliable points:

$$[\mathbf{U_0} \mid \mathbf{V_0}] = [\hat{\mathbf{G}}_\mathbf{0} \mid \hat{\mathbf{H}}_\mathbf{0}] \cdot \left[\begin{array}{c|c} \mathbf{A_0} & \mathbf{B_0} \\ \hline \mathbf{B_0} & \mathbf{C_0} \end{array}\right]^{-1}.$$

   (d) Compute an r_2-dimensional basis $\mathbf{K}$ from the columns of $\left[\frac{\mathbf{U_0}}{\mathbf{V_0}}\right]$
   (e) Linearly solve for the unknown matrix $\mathbf{L}$ using either Eq. (12) (*Generalized Brightness*

Constancy) or Eq. (13) (*Generalized Lucas and Kanade*). [we used Eq. (13)]

(f) Compute the displacements from $\mathbf{K}$ and $\mathbf{L}$ using Eq. (11): $\hat{\mathbf{U}} = \mathbf{K}_U\mathbf{L}$ and $\hat{\mathbf{V}} = \mathbf{K}_V\mathbf{L}$

3. Keep iterating to refine $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$.

Step (b) reduces noise in the measurements using the multi-frame subspace constraints, while steps (d)–(f) eliminate the aperture problem using the multi-frame subspace constraints.

Step (d) bears resemblance to the recovery of the motion matrix in Tomasi and Kanade (1992). However, there are a few significant distinctions between these two approaches: Here we use the subspace constraints to estimate dense flow. This is done even for points which suffer from the aperture problem. Furthermore, our approach is valid also for the singular "2D scenes", where explicit epipolar geometry and 3D information (i.e., 3D motion and 3D shape) cannot be recovered from uncalibrated views. The reader is further referred to Irani and Anandan (2000), which suggests an alternative approach to overcoming the aperture problem within this framework using covariance-weighted subspace projection of only $[\mathbf{U} \mid \mathbf{V}]$, without the need for recovering the intermediate matrix[6] K.

When all the subspace-projection related steps are eliminated (i.e., only steps (a) and (c) are kept), then the algorithm reduces to the 2-frame unconstrained Lucas and Kanade algorithm[7] as if it is applied repeatedly and independently to many frames. This is with the exception that in the Lucas and Kanade algorithm, step (c) is applied to *all* pixels (and not only of the reliable ones): $[\hat{\mathbf{U}} \mid \hat{\mathbf{V}}] = [\mathbf{G} \mid \mathbf{H}] \cdot [\frac{\mathbf{A}\mid\mathbf{B}}{\mathbf{B}\mid\mathbf{C}}]^+$, where $[\frac{\mathbf{A}\mid\mathbf{B}}{\mathbf{B}\mid\mathbf{C}}]^+$ is the *pseudo-inverse* of the matrix $[\frac{\mathbf{A}\mid\mathbf{B}}{\mathbf{B}\mid\mathbf{C}}]$ (as in general $[\frac{\mathbf{A}\mid\mathbf{B}}{\mathbf{B}\mid\mathbf{C}}]$ is not invertible; note that for the reliable pixels: $[\frac{\mathbf{A_0}\mid\mathbf{B_0}}{\mathbf{B_0}\mid\mathbf{C_0}}]^+ = [\frac{\mathbf{A_0}\mid\mathbf{B_0}}{\mathbf{B_0}\mid\mathbf{C_0}}]^{-1}$), and $[\mathbf{G} \mid \mathbf{H}]$ is the *noisy* measurement matrix (and not the noise-cleaned matrix $[\hat{\mathbf{G}} \mid \hat{\mathbf{H}}]$, as in Step (c) above).

We examined the utility of the multi-frame subspace constraints by comparing the multi-frame multi-point subspace-constrained algorithm to the 2-frame *un*constrained version of the Lucas and Kanade algorithm (both implemented as explained above). Using the same basic algorithm in the two cases allows us to isolate and evaluate the true effects of subspace projection on the accuracy of the flow estimation, as all other parameters in the two algorithms are the same.

Figures 1 and 2 show such comparisons. The comparison is done both for real data (Fig. 1), as well as for synthetic data with ground truth (Fig. 2). The two examples also show the applicability of the algorithm both to 2D and to 3D scenarios, without prior knowledge of the model: Fig. 1 is an example of a "3D scene", where the camera performs forward translation with induced 3D parallax. Figure 2 is an example of a "2D scene", where the induced image motion is a pure 2D parametric transformation. For further details regarding the results, see figure captions. Note that because the subspace constraints are *global constraints*, they can resolve the aperture problem even when that exceeds the size of a local window, whereas the 2-frame *un*constrained version of the Lucas and Kanade cannot handle such global aperture problems (e.g., see the pole in Fig. 1).

The algorithm described above shows how the 2-frame Lucas and Kanade flow estimation algorithm can be extended into a multi-point multi-frame constrained flow estimation algorithm, by incorporating the brightness-based subspace constraints. However, the brightness-based subspace constraints presented in Section 3 are not necessarily restricted to this particular algorithm. They could similarly be used to extend other 2-frame flow estimation algorithms into corresponding multi-frame constrained flow estimation algorithms.

4.4. *Automatic Rank Detection*

Step (b) of the algorithm of Section 4.3 projects matrices onto lower-rank matrices, according to the ranks defined in Section 2. In practice, however, the actual rank of these matrices (with some allowed noise tolerance) may be much smaller than the theoretical upper bound r_1 (e.g., in cases of degenerate camera motions or degenerate scene structures). We automatically detect the *actual* rank of these matrices: Let $\mathbf{M}$ be a $k \times l$ matrix, with a known upper bound r on its rank, and an *actual* rank r_M ($r_M \leq r$). The rank reduction (i.e., subspace projection) of $\mathbf{M}$ is done by applying Singular Value Decomposition (SVD) (Golub and Van Loan, 1996) to $\mathbf{M}$. Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$ be the singular values of $\mathbf{M}$ (where $m = \min(k, l)$). We check for the existence of a lower rank r' such that $\frac{(\lambda_{r'+1})^2}{(\lambda_1)^2} < \epsilon$, where ϵ allows for some noise tolerance (we usually choose $\epsilon \approx 1\%$). r_M is set to be: $r_M := \min(r, r')$. All singular values other than the r_M largest ones are then set to zero (i.e., $\lambda_{r_M+1} := 0, \lambda_{r_M+2} := 0, \ldots \lambda_m := 0$), and the matrices produced in the SVD step (now with the new singular values) are re-composed, yielding a matrix $\hat{\mathbf{M}}$ of rank r_M (which is closest to $\mathbf{M}$ in the Frobenius norm). Step (d) of the algorithm of Section 4.3 uses

the same SVD procedure to estimate a spanning basis **K**. For more details see Tomasi and Kanade (1992) and Shapiro (1995).

5. The Applicability of the Subspace Approach

The application of subspace constraints to brightness variations for the purpose of correspondence estimation, requires addressing two issues: (i) the variety spanned by the correspondences (motions) of all points across all frames is generally non-linear, and (ii) the image intensities are non-linear functions of image positions. In order to apply our linear subspace projection approach, we must use linear approximations both of the motion variety, as well as of the image brightness function. This section discusses these approximations and the conditions under which they are satisfied.

5.1. The Motion Approximation

The upper bounds on the ranks of the two large displacement matrices $[\mathbf{U} \mid \mathbf{V}]_{\mathcal{F}\times 2\mathcal{N}}$ and $[\frac{\mathbf{U}}{\mathbf{V}}]_{2\mathcal{F}\times\mathcal{N}}$, which we derived in Section 2 (Table 1), range from 3 to 9. In order for the subspace projection process to significantly control the correspondence estimation, these ranks must be much smaller than the matrices dimensions $\mathcal{F}$ and $\mathcal{N}$ (i.e., the number of frames and the number of points, respectively). Since the number of points in an image is typically very large (on the order of a few hundreds of thousands of pixels per image), the bound on the rank of these matrices is determined by the number of frames. This number, however, is restricted by the range of applicability of the underlying motion model. For example, the instantaneous motion model assumes small camera rotations and small forward translation, while the affine model assumes a very narrow field of view and a very shallow scene (see Appendix A). These assumption are often not valid in long sequences.

However, when the camera performs *pure translation* (as was the case in Figs. 1 and 2), but not necessarily uniform or smooth over time, then: (i) the instantaneous motion approximation is valid for longer sequences, (i.e., a large $\mathcal{F}$), and (ii) the upper-bounds on the ranks of the displacement matrices are much smaller. For example, for the case of a fixed focal length and general camera translation (which could change arbitrarily from frame to frame), the upper bound ranks are:[8] $rank([\mathbf{U} \mid \mathbf{V}]) \leq 3$ and $rank([\frac{\mathbf{U}}{\mathbf{V}}]) \leq 3$ (as opposed to 6 and 8, respectively, when *small* camera rotation is also allowed).

There are other typical scenarios where the ranks of the displacement matrices are very small, leading to powerful subspace constraints. One such example is when the camera motion is uniform (i.e., constant over time) or simply temporally smooth. Although *spatially* the motion between successive frames could be quite complex (translation + zoom + small rotation), *temporally* it is degenerate (i.e., the rows of the matrices are linearly dependent), leading to very low ranks (as low as 1 in the case of uniform motion over time).

5.2. The Gradient Approximation

When the observed image brightness of a scene point does not change significantly as a result of (small) camera motion, then:

$$\begin{aligned} I(x_i, y_i) &\approx I_j(x_i + u_{ij}, y_i + v_{ij}) \\ &= I_j\big(x_i + u^0_{ij} + \Delta u_{ij}, y_i + v^0_{ij} + \Delta v_{ij}\big), \end{aligned}$$

where (u^0_{ij}, v^0_{ij}) is an estimate of the displacement (u_{ij}, v_{ij}) (e.g., known from the previous iteration in an iterative estimation process; see Section 3), and $(\Delta u_{ij}, \Delta v_{ij})$ is the residual unknown displacement. This brightness constraint, however, is implicit and non-linear in the unknown residual displacements $(\Delta u_{ij}, \Delta v_{ij})$. It is therefore common to use the linear approximation of this constraint, which is of the form:

$$[\Delta u_{ij} \Delta v_{ij}] \nabla I_{ij} + I_{tij} = 0, \tag{14}$$

where ∇I_{ij} is the spatial gradient of frame I_j at pixel $(x_i + u^0_{ij}, y_i + v^0_{ij})$. This approximation is valid when $(\Delta u_{ij}, \Delta v_{ij})$ is very small. In order to extend the applicability of this constraint to larger motions, it is usually used within a multi-scale coarse-to-fine iterative estimation process (e.g., see Bergen et al. (1992), Irani et al. (1994), Hanna (1991), Stein and Shashua (1997) and Black and Anandan (1996)). This tends to extend the range of recoverable displacements to approximately 10% of the image size.

However, the linearization of Eq. (14) alone does not suffice for deriving Eqs. (7) and (9), in which the critical transition is made from applying subspace constraints on image displacements, to applying them directly on image brightness quantities. The fact that the brightness matrices $\mathbf{F_T}$ or $[\mathbf{G} \mid \mathbf{H}]$ have low ranks, was derived from the low-rank bounds of the displacement matrix

$[\mathbf{U} \mid \mathbf{V}]$, by factoring out the *shared* image derivative matrices $[\frac{\mathbf{F_X}}{\mathbf{F_Y}}]$ or $[\frac{\mathbf{A}|\mathbf{B}}{\mathbf{B}|\mathbf{C}}]$, respectively. (See Eq. (7) or Eq. (9), respectively.) For the derivatives to be *shared* by all frames, we further replaced the approximation of Eq. (14), where ∇I_{ij} could theoretically be different for every frame, with the following approximation:

$$[\Delta u_{ij}\ \Delta v_{ij}]\nabla I_i + I_{tij} \approx 0, \tag{15}$$

where ∇I_i is now shared by all frames. In other words, we have approximated the gradient ∇I_{ij} (the gradient of frame I_j at pixel $(x_i + u^0_{ij},\ y_i + v^0_{ij})$) with the gradient ∇I_i (the gradient of the reference frame I at pixel (x_i, y_i)). The transition from Eq. (14) to Eq. (15) is valid if:

$$\nabla I_j\big(x_i + u^0_{ij},\ y_i + v^0_{ij}\big) \approx \nabla I(x_i, y_i),$$
$$\text{or in short: } \nabla I_{ij} \approx \nabla I_i. \tag{16}$$

Equation (15) is essentially the same as Eq. (6) in Section 3.1, and the gradient approximation of Eq. (16) is equivalent to the assumption made in the transition from Eq. (3) to Eq. (4) in Section 3.1.

We next analyze the conditions under which the gradient approximation of Eq. (16) is valid. Let $\delta\nabla_{ij} = \nabla I_{ij} - \nabla I_i$. Then the term neglected in the transition from Eq. (14) to Eq. (15) is

$$[\Delta u_{ij}\Delta v_{ij}]\ \delta\nabla_{ij}.$$

This inner-product is negligible (i.e., $[\Delta u_{ij}\Delta v_{ij}]\delta\nabla_{ij} \ll [\Delta u_{ij}\Delta v_{ij}]\nabla I_i$) if one of the following conditions holds:

(i) $[\Delta u_{ij}\Delta v_{ij}] \perp \delta\nabla_{ij}$, or,
(ii) $\|\delta\nabla_{ij}\| \ll \|\nabla I_i\|$.

Condition (i) depends on the local orientation of the underlying image structure, and therefore cannot be guaranteed to hold everywhere in the image. Condition (ii), on the other hand, can be guaranteed to hold everywhere in the image for some types of image motions. For example, when the camera only translates, edges do not change their orientations. In such cases $\|\delta\nabla_{ij}\| \ll \|\nabla I_i\|$, and the approximation of Eq. (15) is therefore valid. This is illustrated in Fig. 3(a). The same is also true for the case of forward translation, when T_Z (the forward translation) is small relative to the distance Z to the scene (although large enough to induce 3D parallax), such as in the case of NASA sequence used in Fig. 1. Similarly, the gradient approximation is valid for small image scaling (e.g., due to small changes in the camera focal length), as the orientations of the edges remain the same, and the magnitudes of the gradients do not change significantly. However, the gradient approximation is violated when the camera rotates. In the case of a rotating camera, edges change their orientations, leading to different gradient orientations (although perhaps of the same magnitude) of corresponding points. This is illustrated in Fig. 3(b). In such cases $\|\delta\nabla_{ij}\|$ is no longer negligible, which implies that the approximation used in Eq. (15) is not valid for sequences with non-negligible camera rotation.

Note that this restriction on the camera rotation is even stronger than the small-rotation restriction of the instantaneous motion model (see Appendix A). We have indeed observed that the algorithm performs well for sequences obtained by a translating camera (e.g.,

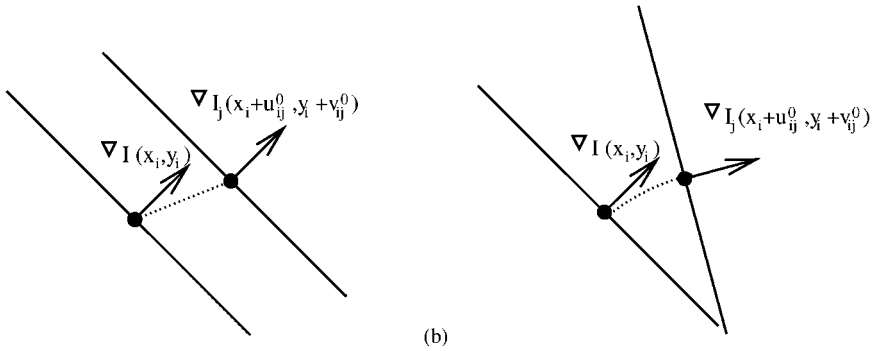

Figure 3. Effects of rotation and translation on the gradient approximation of Eq. (16). Pixel (x_i, y_i) in image I corresponds to pixel $(x_i + u^0_{ij},\ y_i + v^0_{ij})$ in image I_j. (a) When the image translates, the gradient does not change its orientation. Therefore, assuming similar photometric properties between the images (e.g., brightness constancy), $\nabla I_j(x_i + u^0_{ij},\ y_i + v^0_{ij}) \approx \nabla I(x_i, y_i)$. Hence, the gradient approximation of Eq. (16) is valid in this case. (b) When the image rotates, the gradient changes its orientation. Therefore, $\nabla I_j(x_i + u^0_{ij},\ y_i + v^0_{ij}) \neq \nabla I(x_i, y_i)$. Hence, the gradient approximation of Eq. (16) is not valid under rotations.

Figs. 1 and 2), but degrades rapidly for rotations. This restricts the applicability of the current algorithm (of Section 4) to sequences where the camera primarily translates (unconstrained translation, but no rotation). However, we will show next how we can extend the applicability of our algorithm to more general scenarios.

6. Extending the Applicability of Subspace Constraints

In Section 5 we reviewed the conditions under which the basic subspace-constrained correspondence estimation is applicable. We next show how the applicability of subspace constraints can be extended beyond these conditions, to handle cases of complex camera motions (including large rotation and scaling), where the induced displacements span highly *non-linear* varieties, and where the brightness gradients vary in orientation over time.

6.1. The "Plane + Parallax" Approach

Let Π be an arbitrary planar surface in the scene, which is visible in all frames. A homography (a 2D projective transformation) of Π can be estimated[9] between the reference frame I and frame I_j $\forall j$. These homographies are used to stabilize (align) all the frames in the sequence with respect to that planar surface. The only residual motion after plane alignment will be due to residual planar-parallax displacements of scene points which are off the plane Π (see Irani et al. (1998), Irani and Anandan (1996), Irani and Anandan (1999) and Kumar et al. (1994), Sawhney (1994), Shashua and Navab (1994), Irani et al. (1997) and Criminisi et al. (1998)). Let $\{I_j^*\}_{j=1}^F$ be the plane-aligned sequence (the reference frame I remains unchanged after the 2D stabilization). It was shown (e.g., Irani et al. (1999) and Kumar et al. (1994)) that after plane alignment, the residual planar-parallax displacements (i.e., the displacements between I and I_j^*) are:

$$\begin{bmatrix} u_{ij} \\ v_{ij} \end{bmatrix} = -\frac{\gamma_i}{1+\gamma_i \epsilon_{z_j}} \left(\epsilon_{z_j} \begin{bmatrix} x_i \\ y_i \end{bmatrix} - \begin{bmatrix} \epsilon_{x_j} \\ \epsilon_{y_j} \end{bmatrix} \right) \quad (17)$$

where (x_i, y_i) are (as before) the coordinates of a pixel in the reference frame I, $\gamma_i = \frac{H_i}{Z_i}$ represents its 3D structure, where H_i is the perpendicular distance (or "height") of the point i from the reference plane Π, and Z_i is its depth with respect to the reference camera. $(\epsilon_{x_j}, \epsilon_{y_j}, \epsilon_{z_j})$ denotes the *epipole* in projective coordinates. The above formulation is true both for the calibrated as well as for the *uncalibrated* case. (All unknown calibration parameters are folded into the epipole and into the canceled homography. In the normalized calibrated case, $(\epsilon_{x_j}, \epsilon_{y_j}, \epsilon_{z_j})$ would correspond to the Euclidean 3D camera translation $(t_{X\,j}, t_{Y\,j}, t_{Z\,j})$).

After the alignment of the reference plane Π, the residual image motion of Eq. (17) is due only to the *translational* part of the camera motion, and to the *deviations* of the scene structure from the planar surface. All effects of rotations and of changes in calibration within the sequence are captured by the homography (e.g., see Irani and Anandan (1996) and Irani et al. (1999)). The elimination of the homography (via image warping) reduces the problem from the general uncalibrated unconstrained case to the simpler case of pure translation with fixed (unknown) calibration.

Although the original sequence may contain large rotations and strong projective effects, resulting in a highly non-linear variety, this non-linearity is mostly captured by the plane homography. The residual planar-parallax displacements can be approximated well by a linear subspace with very low dimensionality. This is shown next.

The residual planar-parallax displacements in Eq. (17) are *exact* equations (i.e., no approximation was made). In theory, these displacements do not necessarily span a linear subspace. However, when the following relation holds:

$$\gamma_i \epsilon_{z_j} \ll 1 \quad (18)$$

then these displacements do span a linear subspace, since in such cases Eq. (17) reduces to:

$$\begin{bmatrix} u_{ij} \\ v_{ij} \end{bmatrix} = -\gamma_i \left(\epsilon_{z_j} \begin{bmatrix} x_i \\ y_i \end{bmatrix} - \begin{bmatrix} \epsilon_{x_j} \\ \epsilon_{y_j} \end{bmatrix} \right), \quad (19)$$

which has a bilinear form. In Appendix B we show that the planar-parallax displacements of Eq. (19) reside in 3-dimensional linear subspaces, i.e.,:

$$\boxed{rank([\mathbf{U} \mid \mathbf{V}]) \leq 3 \text{ and } rank\left(\begin{bmatrix} \mathbf{U} \\ \mathbf{V} \end{bmatrix} \right) \leq 3.}$$

For complete derivations, see Appendix B.

The condition in Eq. (18) ($\gamma_i \epsilon_{z_j} = \frac{H_i}{Z_i} \epsilon_{z_j} \ll 1$), which gave rise to the bilinear form of Eq. (19), is satisfied if at least one of the following two conditions holds:

Either: (i) $H_i \ll Z_i$, namely, *the scene is shallow* (i.e., the distance H_i of the scene point from the reference plane Π is much smaller than its distance Z_i from the camera. This condition is usually satisfied if the plane lies within the scene, and the camera is not too close to it),

Or: (ii) $\epsilon_{z_j} \ll Z_i$ (or in the calibrated case, $t_{Zj} \ll Z_i$), namely, *the forward motion of the camera is small relative to its distance from the scene*, which is often the case within short temporal segments of real video sequences.

Note that the assumption in Eq. (18) is *significantly less restrictive* than either the assumptions of the affine camera approximation or the instantaneous perspective motion approximation (see Appendix A). Condition (i) above (i.e., $H_i \ll Z_i$) is a subset of the conditions required in the orthographic model approximation, and

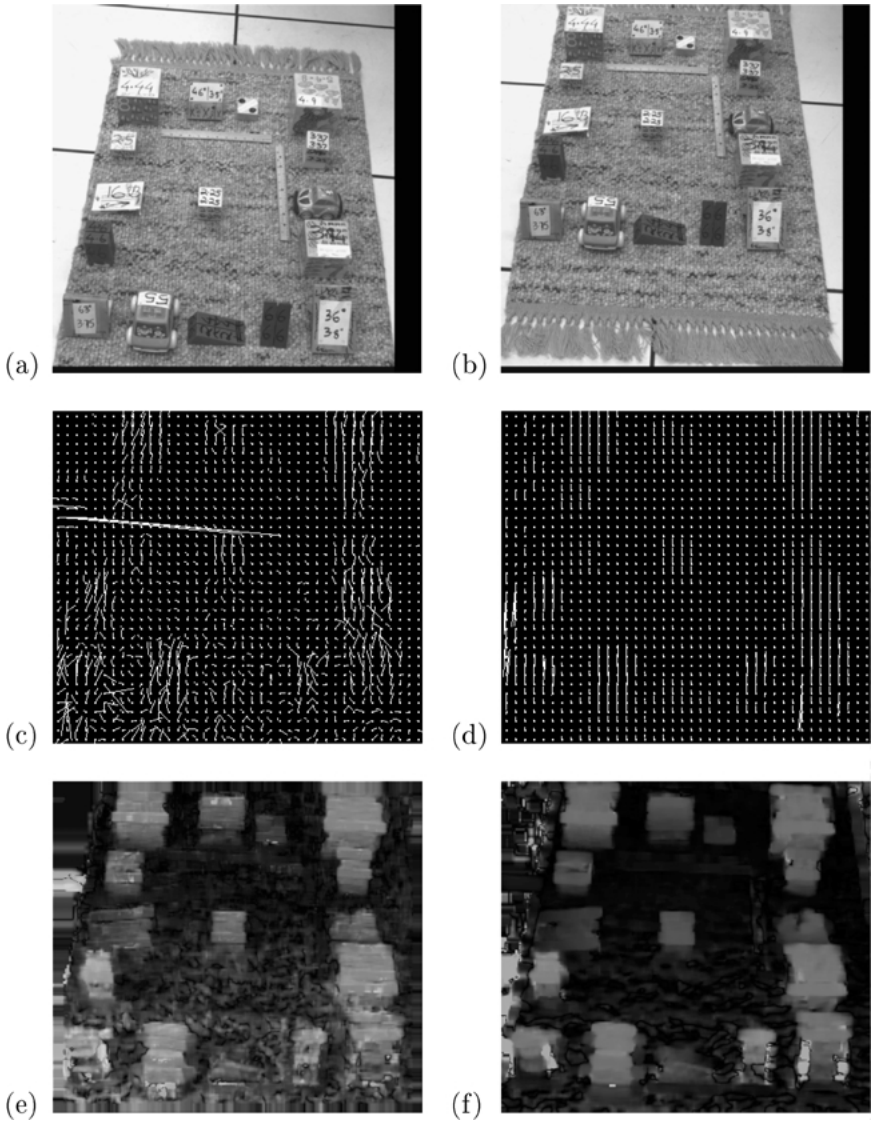

Figure 4. Applying the algorithm to estimate planar-parallax displacements. (a, b) Two images (first and last) from a 6-image sequence obtained by a hand-held still camera moving forward in a 3D scene (this is the "block" sequence from Kumar et al. (1994)). The sequence was then plane-stabilized by aligning the ground plane (the carpet) across all frames. (c) Flow field (planar-parallax displacements) within the plane-stabilized sequence estimated using the unconstrained two-frame Lucas and Kanade algorithm. (d) Flow field (planar-parallax displacements) of the corresponding frame estimated by the multi-frame subspace-constrained algorithm. (e, f) The corresponding flow magnitudes at every pixel.

condition (ii) above (i.e., $\epsilon_{z_j} \ll Z_i$, or $t_{Zj} \ll Z_i$) is a subset of the conditions required in the perspective instantaneous model approximation. Furthermore, the approximation suggested in Eq. (18) does not put any restrictions on the camera rotation. It allows for large camera rotations and for unknown changes in camera calibration, since these effects are fully captured by the homographies of the reference plane Π. This model is therefore valid for much longer video sequences (larger number of frames $\mathcal{F}$) than the affine or the instantaneous perspective models.

The Plane + Parallax decomposition therefore provides a means for extending the applicability of our brightness-based subspace constrained estimation, as it does not suffer from either of the two limitations mentioned in Section 5:

(i) The subspace constraints in the plane-stabilized sequence are powerful constraints, since the ranks of the residual planar-parallax displacement matrices are very small (≤ 3) relative to the number of frames, regardless of the complexity of the camera motion and regardless of the (unknown) changes in camera calibration.
(ii) After plane alignment, the planar-parallax displacements induce *pure translational* motion fields. Therefore, the brightness-based subspace constraints with the gradient approximation of Eq. (16) are applicable.

The limitation of the plane + parallax approach is in the need for good prior alignment of the video frames with respect to a planar surface. This requires that a real physical plane exist in the scene and capture a reasonably-sized image region, in order to guaranty accurate plane alignment across all frames. Such a plane does not exist in every scene. However, indoor scenes often contain many man-made planes, such as walls, windows, floors, etc. In outdoor scenes: a road, a boulevard of trees, or—for all practical purposes—a distant enough portion of the scene, can serve the purpose of a planar surface for estimating the homographies.[10]

Figure 4 shows an example of applying the multi-frame subspace-constrained estimation on an indoor sequence obtained by a hand-held camera (this is the "block" sequence from Kumar et al. (1994)). The reference plane used for alignment was the carpet, which was automatically detected and aligned using the technique of Irani et al. (1994) for dominant 2D parametric motion estimation.

7. Conclusion

In this paper we presented an approach for using multi-frame subspace constraints for *constraining* a 2D correspondence estimation process, while exploiting all available spatio-temporal information in a short video sequence. In particular, the paper has four main contributions:

1. We showed that the set of all flow-fields across multiple video frames (that image the same rigid scene) reside in a low-dimensional linear subspace. This was shown for several motion models, scene models, and imaging models.
2. We extended the notion of multi-frame subspace constraints on motion fields to subspace constraints directly on image brightness quantities.
3. We showed how these brightness-based subspace constraints can be used as *additional constraints* to further constrain the correspondence estimation process. In particular, we showed how the two-frame Lucas and Kanade algorithm can be extended into a multi-frame multi-point algorithm. However, the brightness-based subspace constraints are not necessarily restricted to this particular algorithm. They could similarly be used to extend other 2-frame flow estimation algorithms into corresponding multi-frame constrained flow estimation algorithms.
4. While the brightness-based subspace constraints are powerful when they are applicable, their applicability is restricted. We identify these restrictions, and propose an approach to extend the applicability of the brightness-based subspace constraints beyond these restrictions and to some non-linear varieties by employing the Plane + Parallax model.

Appendix A: Ranks for Various World Models, Motion Models, and Camera Models

In this appendix we derive the ranks (subspace constraints) of the two large matrices $[\frac{\mathbf{U}}{\mathbf{V}}]_{2\mathcal{F}\times\mathcal{N}}$ (the "trajectory" matrix) and $[\mathbf{U} \mid \mathbf{V}]_{\mathcal{F}\times 2\mathcal{N}}$ (the "displacement-field" matrix). We show that these matrices have low ranks under many different conditions. In particular, we derive the rank constraints for two "linear" camera models: (i) an "affine" camera (Shapiro, 1995) which is obtained by linearizing the projection process (i.e., a camera with weak-perspective, para-perspective, or orthographic projection), and (ii) a perspective camera

undergoing small camera rotation and small forward translation (i.e., the "instantaneous motion model" (Longuet-Higgins and Prazdny, 1980)). For each of these two camera models, we examine the ranks for a general 3D scene as well as for a planar 2D scene. We also check the effect of varying the camera calibration (focal length) on these ranks.

A 3D scene point (X_i, Y_i, Z_i) is observed at pixel (x_i, y_i) in the reference frame I. Let $\vec{t_j} = (t_{Xj}, t_{Yj}, t_{Zj})$ denote the camera translation between frame I and frame I_j, and let $\vec{\Omega}_j = (\Omega_{Xj}, \Omega_{Yj}, \Omega_{Zj})$ denote the camera rotation between the two frames. Let $\mathbf{R_j}$ be the rotation matrix corresponding to $\vec{\Omega}_j$. This scene point is therefore observed in frame I_j at pixel (x_{ij}, y_{ij}) with new world coordinates (X_{ij}, Y_{ij}, Z_{ij}), where,

$$\begin{bmatrix} X_{ij} \\ Y_{ij} \\ Z_{ij} \end{bmatrix} = R_j \cdot \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} + \vec{t_j} \tag{20}$$

We will derive the upper bounds on the ranks of the two large matrices of induced-displacements $[\frac{\mathbf{U}}{\mathbf{V}}]$ and $[\mathbf{U} \mid \mathbf{V}]$, in a manner similar to Tomasi and Kanade (1992). We will show that for various scenarios and under various conditions, each of these two large matrices can be decomposed into a bilinear product of two smaller matrices $\mathbf{M} \cdot \mathbf{P}$, where $\mathbf{M}$ is only frame-dependent (contains the camera motion information), and $\mathbf{P}$ is point-dependent (contains the shape information). The number of columns of $\mathbf{M}$ (which is also the number of rows of $\mathbf{P}$) will be shown to be small. This number will determine the upper bound on the rank of the original decomposed matrix (namely, $[\frac{\mathbf{U}}{\mathbf{V}}]$ or $[\mathbf{U} \mid \mathbf{V}]$). It is important to note that the derivations of the structure of the matrices $\mathbf{M}$ and $\mathbf{P}$ is used only for obtaining the upper bounds on the ranks of $[\frac{\mathbf{U}}{\mathbf{V}}]$ or $[\mathbf{U} \mid \mathbf{V}]$. At no point in the correspondence estimation process (Section 4) is any 3D information required or estimated. Only the knowledge of the upper bounds on the ranks of the two large correspondence matrices (as summarized in Table 1) is used.

1. **Affine Camera—3D Scene:** Tomasi and Kanade (1992) and Shapiro (1995) showed that in the case of an affine camera, corresponding image points across all image frames reside in a 4-dimensional linear subspace. (With some additional manipulation of fixating a point, it can be reduced to 3.) The derivation of the subspace constraints for pixel *displacements* (as opposed to pixel positions) is very similar to the derivation in Tomasi and Kanade (1992) and Shapiro (1995). To make the paper self-contained, we provide this derivation for the simpler orthographic case. The reader is referred to Shapiro (1995) for the weak-perspective and para-perspective cases.

   In the orthographic projection model:

$$\begin{aligned} \begin{bmatrix} u_{ij} \\ v_{ij} \end{bmatrix}_{2\times 1} &= \begin{bmatrix} x_{ij} - x_i \\ y_{ij} - y_i \end{bmatrix} = \begin{bmatrix} X_{ij} - X_i \\ Y_{ij} - Y_i \end{bmatrix} \\ &= \begin{bmatrix} (R_{11\,j} - 1) & R_{12\,j} & R_{13\,j} & t_{X\,j} \\ R_{21\,j} & (R_{22\,j} - 1) & R_{23\,j} & t_{Y\,j} \end{bmatrix}_{2\times 4} \\ &\quad \times \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}_{4\times 1} \end{aligned} \tag{21}$$

   where $R_{kl\,j}$ is the value of rotation matrix R_j at row k and column l. Since the camera motion is common to all points between frame I and frame I_j, then Eq. (21) can be extended to multiple points:

$$\begin{aligned} &\begin{bmatrix} u_{1j}, u_{2j}, \ldots, u_{\mathcal{N}j} \\ v_{1j}, v_{2j}, \ldots, v_{\mathcal{N}j} \end{bmatrix}_{2\times\mathcal{N}} \\ &= \begin{bmatrix} (R_{11\,j} - 1) & R_{12\,j} & R_{13\,j} & t_{X\,j} \\ R_{21\,j} & (R_{22\,j} - 1) & R_{23\,j} & t_{Y\,j} \end{bmatrix}_{2\times 4} \\ &\quad \times \mathbf{P}_{(4\times\mathcal{N})} \end{aligned} \tag{22}$$

   where

$$\mathbf{P} = \begin{bmatrix} X_1, X_2, \ldots, X_{\mathcal{N}} \\ Y_1, Y_2, \ldots, Y_{\mathcal{N}} \\ Z_1, Z_2, \ldots, Z_{\mathcal{N}} \\ 1, 1, \ldots, 1 \end{bmatrix}_{4\times\mathcal{N}}.$$

   Because the matrix $\mathbf{P}$ is invariant to the camera motion, and is only point-dependent, it is common to all frames. Hence, Eq. (22) can be extended to multiple frames:

$$\begin{bmatrix} \mathbf{U} \\ \mathbf{V} \end{bmatrix}_{(2\mathcal{F}\times\mathcal{N})} = \begin{bmatrix} \mathbf{M_U} \\ \mathbf{M_V} \end{bmatrix}_{(2\mathcal{F}\times 4)} \mathbf{P}_{(4\times\mathcal{N})} \tag{23}$$

   where the j-th row of the matrix $\mathbf{M}_{\mathbf{U}(\mathcal{F}\times 4)}$ is

$$(M_U)_j = \begin{bmatrix} (R_{11\,j} - 1) & R_{12\,j} & R_{13\,j} & t_{X\,j} \end{bmatrix}$$

and the j-th row of the matrix $\mathbf{M_V}_{(\mathcal{F}\times 4)}$ is

$$(M_V)_j = \begin{bmatrix} R_{21\,j} & \left(R_{22\,j} - 1\right) & R_{23\,j} & t_{Y\,j} \end{bmatrix}.$$

Equation (23) therefore implies that for the orthographic case, the ranks of the matrices $\mathbf{U}$, $\mathbf{V}$, and $[\frac{\mathbf{U}}{\mathbf{V}}]$ are all at most 4 (could be reduced to an upper-bound of 3 with the proper point fixation; see Tomasi and Kanade (1992) and Shapiro (1995)). A similar constraint can also be derived for the uncalibrated affine camera (see Shapiro (1995)).
Similarly,

$$[\,\mathbf{U} \quad \mathbf{V}\,]_{(\mathcal{F}\times 2\mathcal{N})} = [\,\mathbf{M_U} \quad \mathbf{M_V}\,]_{(\mathcal{F}\times 8)} \times \begin{bmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{P} \end{bmatrix}_{(8\times 2\mathcal{N})} \tag{24}$$

The rank of the matrix $[U \mid V]$ is therefore at most 8 for the orthographic case (could be reduced to an upper-bound of 6 with proper point fixation). A similar constraint can also be derived for the uncalibrated affine camera.
To summarize, in the case of an affine camera, the upper bounds on the ranks of the two large matrices of displacements are:

$$\boxed{rank\left(\left[\frac{\mathbf{U}}{\mathbf{V}}\right]\right) \leq 4 \text{ and } rank([\mathbf{U} \mid \mathbf{V}]) \leq 8.}$$

2. **Affine Camera—Planar (2D) Scene:** When the scene is planar, Z_i can be written as a function of X_i and Y_i: $Z_i = \alpha + \beta \cdot X_i + \gamma \cdot Y_i$. Replacing Z_i in Eqs. (23) and (24) with $(\alpha + \beta \cdot X_i + \gamma \cdot Y_i)$, and regrouping the terms, leads to the following lower rank constraints:

$$\boxed{rank\left(\left[\frac{\mathbf{U}}{\mathbf{V}}\right]\right) \leq 3 \text{ and } rank([\mathbf{U} \mid \mathbf{V}]) \leq 6.}$$

3. **Perspective Camera—Instantaneous Motion, 3D Scene:** In the perspective model: $x_{ij} = f_j \frac{X_{ij}}{Z_{ij}}$, and $y_{ij} = f_j \frac{Y_{ij}}{Z_{ij}}$, where f_j is the focal length of the camera at frame I_j. Longuet-Higgins and Prazdny (1980) showed that in the instantaneous case, when the rotation angle is small and the forward translation is small (i.e., $t_Z \ll Z$), the 2D displacement can be well approximated by:

$$\begin{bmatrix} u_{ij} \\ v_{ij} \end{bmatrix} = \frac{1}{Z_i} \begin{bmatrix} f t_{X_j} - t_{Z_j} x_i \frac{f}{f_j} \\ f t_{Y_j} - t_{Z_j} y_i \frac{f}{f_j} \end{bmatrix} + \begin{bmatrix} -\frac{\Omega_{X_j}}{f_j} x_i y_i + \Omega_{Y_j} f + \frac{\Omega_{Y_j}}{f_j} x_i^2 - \Omega_{Z_j} y_i + x_i \left(1 - \frac{f}{f_j}\right) \\ -\frac{\Omega_{X_j}}{f_j} y_i^2 - \Omega_{X_j} f + \frac{\Omega_{Y_j}}{f_j} x_i y_i + \Omega_{Z_j} x_i + y_i \left(1 - \frac{f}{f_j}\right) \end{bmatrix} \tag{25}$$

where f, f_j are the focal lengths of frames I, I_j, respectively.[11] We will next investigate the ranks for two cases: the case when the focal length varies across the sequence, and the case when it is kept fixed across the sequence (but unknown).

(a) *Varying Focal Length (3D Scene)*: Using Eq. (25), the displacement components (u_{ij}, v_{ij}) of pixel (x_i, y_i) from the reference frame I to frame I_j can be rewritten as a *bilinear* product:

$$\begin{bmatrix} u_{ij} \\ v_{ij} \end{bmatrix}_{2\times 1} = \begin{bmatrix} (M_U)_j \\ (M_V)_j \end{bmatrix}_{2\times 9} P_{i\ (9\times 1)}$$

where

$$P_i = \begin{bmatrix} 1 & x_i & y_i & \frac{1}{Z_i} & \frac{x_i}{Z_i} & \frac{y_i}{Z_i} & x_i^2 & y_i^2 & (x_i y_i) \end{bmatrix}^T$$

is a *point-dependent* column vector ($i = 1..\mathcal{N}$), and

$$(M_U)_j = \begin{bmatrix} -f\Omega_{Y_j} & \left(1 - \frac{f}{f_j}\right) & -\Omega_{Z_j} & f t_{X_j} & -\frac{f}{f_j} t_{Z_j} & 0 & \frac{\Omega_{Y_j}}{f_j} & 0 & -\frac{\Omega_{X_j}}{f_j} \end{bmatrix}$$
$$(M_V)_j = \begin{bmatrix} -f\Omega_{X_j} & \Omega_{Z_j} & \left(1 - \frac{f}{f_j}\right) & f t_{Y_j} & 0 & -\frac{f}{f_j} t_{Z_j} & 0 & -\frac{\Omega_{X_j}}{f_j} & \frac{\Omega_{Y_j}}{f_j} \end{bmatrix}$$

are *frame-dependent* row vectors ($j = 1..\mathcal{F}$). Therefore, all flow vectors of all points across all frames can be expressed as a bilinear product of matrices:

$$\left[\frac{\mathbf{U}}{\mathbf{V}}\right]_{(2\mathcal{F}\times\mathcal{N})} = \left[\frac{\mathbf{M_U}}{\mathbf{M_V}}\right]_{(2\mathcal{F}\times 9)} \mathbf{P}_{(9\times\mathcal{N})} \tag{26}$$

where the i-th column of $\mathbf{P}$ is the vector P_i, and the j-th row of $\mathbf{M_U}$ and $\mathbf{M_V}$ are the vectors $(M_U)_j$ and $(M_V)_j$, respectively. Equation (26) implies that $rank([\frac{\mathbf{U}}{\mathbf{V}}]) \leq 9$.
Similarly, we can analyze the rank of $[\mathbf{U} \mid \mathbf{V}]$:

$$[u_{ij}\ v_{ij}]_{1\times 2} = M_{j\,(1\times 9)}\ [(P_X)_i\ (P_Y)_i]_{9\times 2}$$

where,

$$M_j = \left[\frac{\Omega_{X_j}}{f_j} \quad \frac{\Omega_{Y_j}}{f_j} \quad f\Omega_{X_j} \quad f\Omega_{Y_j} \quad \Omega_{Z_j} \quad ft_{X_j} \quad ft_{Y_j} \quad \frac{f}{f_j}t_{Z_j} \quad \left(1-\frac{f}{f_j}\right)\right]$$

is a *frame-dependent* row vector, and

$$(P_X)_i = \left[-x_i y_i \quad x_i^2 \quad 0 \quad 1 \quad -y_i \quad \frac{1}{Z_i} \quad 0 \quad -\frac{x_i}{Z_i} \quad x_i\right]^T$$

$$(P_Y)_i = \left[-y_i^2 \quad x_i y_i \quad -1 \quad 0 \quad x_i \quad 0 \quad \frac{1}{Z_i} \quad -\frac{y_i}{Z_i} \quad y_i\right]^T$$

are *point-dependent* column vectors. This leads to the following matrix equation for all points and all frames:

$$[\mathbf{U} \mid \mathbf{V}]_{(\mathcal{F}\times 2\mathcal{N})} = \mathbf{M}_{(\mathcal{F}\times 9)}\,[\mathbf{P_X} \mid \mathbf{P_Y}]_{(9\times 2\mathcal{N})} \quad (27)$$

where the i-th column of $\mathbf{P_X}$ and $\mathbf{P_Y}$ are $(P_X)_i$ and $(P_Y)_i$, respectively, and the j-th row of $\mathbf{M}$ is M_j.

To summarize, when both the focal length and the camera motion change across the sequence, then:

$$\boxed{rank([\mathbf{U} \mid \mathbf{V}]) \le 9 \text{ and } rank\left(\left[\frac{\mathbf{U}}{\mathbf{V}}\right]\right) \le 9.}$$

(b) *Constant Focal Length (3D Scene)*: When the camera motion changes but the focal length remains constant (but unknown) across the sequence, namely, $\forall j \; f_j = f$, then the ranks of these matrices are lower. In that case:

$$\left[\frac{\mathbf{U}}{\mathbf{V}}\right]_{(2\mathcal{F}\times\mathcal{N})} = \left[\frac{\mathbf{M_U}}{\mathbf{M_V}}\right]_{(2\mathcal{F}\times 8)} \mathbf{P}_{(8\times\mathcal{N})} \quad (28)$$

where,

$$P_i = \left[x_i \quad y_i \quad \frac{f}{Z_i} \quad \frac{x_i}{Z_i} \quad \frac{y_i}{Z_i} \quad \frac{x_i y_i}{f} \quad \left(f+\frac{x_i^2}{f}\right) \quad \left(f+\frac{y_i^2}{f}\right)\right]^T$$

$$(M_U)_j = \left[0 \quad -\Omega_{Z_j} \quad t_{X_j} \quad -t_{Z_j} \quad 0 \quad -\Omega_{X_j} \quad \Omega_{Y_j} \quad 0\right]$$

$$(M_V)_j = \left[\Omega_{Z_j} \quad 0 \quad t_{Y_j} \quad 0 \quad -t_{Z_j} \quad \Omega_{Y_j} \quad 0 \quad -\Omega_{X_j}\right]$$

Similarly,

$$[\mathbf{U} \mid \mathbf{V}]_{(\mathcal{F}\times 2\mathcal{N})} = \mathbf{M}_{(\mathcal{F}\times 6)}\,[\mathbf{P_X} \mid \mathbf{P_Y}]_{(6\times 2\mathcal{N})} \quad (29)$$

where,

$$M_j = \left[\Omega_{X_j} \quad \Omega_{Y_j} \quad \Omega_{Z_j} \quad t_{X_j} \quad t_{Y_j} \quad t_{Z_j}\right]$$

$$(P_X)_i = \left[-\frac{x_i y_i}{f} \quad \left(f+\frac{x_i^2}{f}\right) \quad -y_i \quad \frac{f}{Z_i} \quad 0 \quad -\frac{x_i}{Z_i}\right]^T$$

$$(P_Y)_i = \left[-\left(f+\frac{y_i^2}{f}\right) \quad \frac{x_i y_i}{f} \quad x_i \quad 0 \quad \frac{f}{Z_i} \quad -\frac{y_i}{Z_i}\right]^T$$

Therefore, when the focal length of the camera remains constant (*but unknown*) *across the sequence, and only the camera motion varies, then*:

$$\boxed{rank([\mathbf{U} \mid \mathbf{V}]) \le 6 \text{ and } rank\left(\left[\frac{\mathbf{U}}{\mathbf{V}}\right]\right) \le 8.}$$

4. **Perspective Camera—Instantaneous Motion, Planar (2D) Scene:** When the scene is planar, then in the perspective case $\frac{1}{Z_i}$ can be written as in Adiv (1985): $\frac{1}{Z_i} = \alpha' + \beta' \cdot x_i + \gamma' \cdot y_i$. Substituting this expression into Eq. (25) and regrouping the terms leads to simpler bilinear forms with the following rank constraints:

   (a) *Constant Focal Length (Planar Scene)*:

$$\boxed{rank\left(\left[\frac{\mathbf{U}}{\mathbf{V}}\right]\right) \le 6 \text{ and } rank([\mathbf{U} \mid \mathbf{V}]) \le 6}$$

   (b) *Varying Focal Length (Planar Scene)*:

$$\boxed{rank\left(\left[\frac{\mathbf{U}}{\mathbf{V}}\right]\right) \le 6 \text{ and } rank([\mathbf{U} \mid \mathbf{V}]) \le 8.}$$

Appendix B: Ranks of Planar-Parallax Displacements

In Section 6 we derived the linear approximation to the planar-parallax displacements (see Eq. (19)). We will next derive the ranks of the planar-parallax displacement matrices $[\frac{\mathbf{U}}{\mathbf{V}}]$ and $[\mathbf{U} \mid \mathbf{V}]$. Equation (19) can be rewritten as a *bilinear* product:

$$\begin{bmatrix} u_{ij} \\ v_{ij} \end{bmatrix}_{2\times 1} = \begin{bmatrix} (M_U)_j \\ (M_V)_j \end{bmatrix}_{2\times 3} P_{i(3\times 1)}$$

where

$$P_i = [\,\gamma_i \quad -\gamma_i x_i \quad -\gamma_i y_i\,]^T$$

is a *point-dependent* column vector ($i = 1..\mathcal{N}$), and

$$(M_U)_j = [\,\epsilon_{X_j} \quad \epsilon_{Z_j} \quad 0\,]$$
$$(M_V)_j = [\,\epsilon_{Y_j} \quad 0 \quad \epsilon_{Z_j}\,]$$

are *frame-dependent* row vectors ($j = 1..\mathcal{F}$). Therefore, all planar parallax displacements of all points across all (plane-aligned) frames can be expressed as a bilinear product of matrices:

$$\left[\frac{\mathbf{U}}{\mathbf{V}}\right]_{(2\mathcal{F}\times\mathcal{N})} = \left[\frac{\mathbf{M_U}}{\mathbf{M_V}}\right]_{(2\mathcal{F}\times 3)} \mathbf{P}_{(3\times\mathcal{N})} \qquad (30)$$

where the i-th column of $\mathbf{P}$ is the vector P_i, and the j-th row of $\mathbf{M_U}$ and $\mathbf{M_V}$ are the vectors $(M_U)_j$ and $(M_V)_j$, respectively. Equation (30) implies that $rank([\frac{\mathbf{U}}{\mathbf{V}}]) \leq 3$.

Similarly, we can analyze the rank of $[\mathbf{U} \mid \mathbf{V}]$:

$$[u_{ij}\ v_{ij}]_{1\times 2} = M_{j\,(1\times 3)}\,[(P_X)_i\ (P_Y)_i]_{3\times 2}$$

where,

$$M_j = \left[\,\epsilon_{X_j} \quad \epsilon_{Y_j} \quad \epsilon_{Z_j}\,\right]$$

is a *frame-dependent* row vector, and

$$(P_X)_i = [\,\gamma_i \quad 0 \quad -\gamma_i x_i\,]^T$$
$$(P_Y)_i = [\,0 \quad \gamma_i \quad -\gamma_i y_i\,]^T$$

are *point-dependent* column vectors. This leads to the following matrix equation for all points and all frames:

$$[\mathbf{U} \mid \mathbf{V}]_{(\mathcal{F}\times 2\mathcal{N})} = \mathbf{M}_{(\mathcal{F}\times 3)}\,[\mathbf{P_X} \mid \mathbf{P_Y}]_{(3\times 2\mathcal{N})} \qquad (31)$$

where the i-th column of $\mathbf{P_X}$ and $\mathbf{P_Y}$ are $(P_X)_i$ and $(P_Y)_i$, respectively, and the j-th row of $\mathbf{M}$ is M_j.

To summarize, the planar-parallax displacements reside in 3*-dimensional linear subspaces, even for extended sequences* (*large number of frames* $\mathcal{F}$) *and for uncalibrated cameras* (*with unknown changing calibration*):

$$\boxed{rank([\mathbf{U} \mid \mathbf{V}]) \leq 3 \text{ and } rank\left(\left[\frac{\mathbf{U}}{\mathbf{V}}\right]\right) \leq 3.}$$

Acknowledgment

The author would like to thank P. Anandan for the helpful discussions and for his many insightful comments about the paper.

Notes

1. By "2D scenes" we refer to cases when the scene is either very distant from the camera, *or* when the world is planar, *or* when the camera is not translating (i.e., only rotating and zooming, such as with a camera mounted on a tripod). See Irani and Anandan (1998) for a more complete explanation of the distinction made between "2D scenes" and "3D scene".
2. Choosing the reference frame as the middle frame extends the applicability of the model to twice as many frames.
3. The underlying assumption here is that if an image point (x_i, y_i) in frame I corresponds to the image point $(x_i + u^0_{ij} + \Delta u_{ij}, y_i + v^0_{ij} + \Delta v_{ij})$ in frame I_j, then for small $(\Delta u_{ij}, \Delta v_{ij})$ we can assume that the image point $(x_i - \Delta u_{ij}, y_i - \Delta v_{ij})$ in frame I will *approximately* correspond to the image point $(x_i + u^0_{ij}, y_i + v^0_{ij})$ in frame I_j. Although this is an approximation, it becomes more and more accurate in the iterative refinement process, as $(\Delta u_{ij}, \Delta v_{ij})$ becomes smaller and smaller with each iteration.
4. This approximation is often made in direct methods (e.g., see Bergen et al. (1992)), and is usually done for computational efficiency. It allows to linearize I around the point (x_i, y_i), which is kept fixed throughout the iterative-warp refinements, instead of linearizing frame I_j around the point $(x_i + u^0_{ij}, y_i + v^0_{ij})$, which gets updated at every iteration. Therefore, the spatial derivatives need to be estimated only once on the reference image, and not repeatedly at every iteration. In our multi-frame case the computational efficiency issue is even more pronounced, as the reference frame I is shared by all other frames I_j ($j = 1..\mathcal{F}$). Hence, the spatial derivatives need to be estimated only *once* on the reference frame, regardless of the number of frames in the sequence.
5. The observation that the individual matrices $\left[\begin{smallmatrix} a_i & b_i \\ b_i & c_i \end{smallmatrix}\right]$ can be viewed as inverse "covariance matrices", and that the subspace projection on $[\mathbf{G} \mid \mathbf{H}]$ can therefore be interpreted as a "covariance-weighted subspace projection," was pointed out to me by P. Anandan. This point is further elaborated and generalized in Irani and Anandan (2000). For a more complete analysis of Bayesian models in Low-level Vision see Szeliski (1990).
6. Although the problem addressed in Irani and Anandan (2000) was a different one (not the estimation of 2D correspondences, but rather the recovery of 3D motion dense 3D structure and from correspondences with very high degrees of directional uncertainty (including pure normal flow)), the same basic idea used in Irani and Anandan (2000) can also be applied here for constraining the 2D correspondence estimation itself, without recovering K. This would lead to a slightly different algorithm than the one described above in the"Summary of the Algorithm".
7. Or more precisely, the iterative coarse-to-fine version of Lucas and Kanade algorithm as described in Bergen et al. (1992).
8. These ranks are obtained by setting $\vec{\Omega}_j = 0$ in Eqs. (28) and (29) of Appendix A. See also Appendix B for a similar model.
9. Homography estimation is a much simpler problem than general correspondence estimation, as it is described by a few (at most 9)

global parameters. Since the image patch contains many pixels, this is usually a well-posed problem. Bergen et al. (1992), Irani et al. (1994) and Black and Anandan (1996) are some examples of methods for estimating 2D parametric transformations.

10. There are methods for automatically "locking onto" a dominant planar motion in a sequence, without the need to manually specify its image region (e.g., see Irani et al. (1994)).
11. The original equations in Longuet-Higgins and Prazdny (1980) were derived for a fixed focal length. However, the equation for the case of changing focal length (Eq. (25)) is straightforward to derive following the same steps.

References

Adiv, G. 1985. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7(4):384–401.

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

Anandan, P. and Avidan, S. 2000. Integrating local affine into global perspective images in the joint image space. In *European Conference on Computer Vision*, Dublin, pp. 907–921.

Barron, J.L., Fleet, D.J., Beauchemin, S.S., and Burkitt, T.A. 1992. Performance of optical flow techniques. In *IEEE Conference on Computer Vision and Pattern Recognition*, Champaign, pp. 236–242.

Bergen, J.R., Anandan, P., Hanna, K.J., and Hingorani, R. 1992. Hierarchical model-based motion estimation. In *European Conference on Computer Vision*, Santa Margarita Ligure, pp. 237–252.

Bergen, J.R., Burt, P.J., Hingorani, R., and Peleg, S. 1992. A three-frame algorithm for estimating two-component image motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14:886–895.

Black, M.J. and Anandan, P. 1991. Robust dynamic motion estimation over time. In *IEEE Conference on Computer Vision and Pattern Recognition*, Maui, pp. 296–302.

Black, M.J. and Anandan, P. 1996. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63:75–104.

Criminisi, A., Reid, I., and Zisserman, A. 1998. Duality, rigidity and planar parallax. In *European Conference on Computer Vision*, Freiburg.

Golub, G.H. and Van Loan, C.F. 1996. *Matrix Computations*. The John Hopkins University Press: Baltimore, MD.

Hanna, K. 1991. Direct multi-resolution estimation of ego-motion and structure from motion. In *IEEE Workshop on Visual Motion*, Princeton, NJ, pp. 156–162.

Hanna, K.J. and Okamoto, N.E. 1993. Combining stereo and motion for direct estimation of scene structure. In *International Conference on Computer Vision*, Berlin, pp. 357–365.

Heeger, D.J. and Jepson, A.D. 1992. Subspace methods for recovering rigid motion i: Algorithm and implementation. *International Journal of Computer Vision*, 7:95–117.

Horn, B.K.P. and Schunck, B.G. 1981. Determining optical flow. *Artificial Intelligence*, 17(1–3):185–203.

Irani, M. 1999. Multi-frame optical flow estimation using subspace constraints. In *International Conference on Computer Vision*, Corfu, pp. 626–633.

Irani, M. and Anandan, P. 1996. Parallax geometry of pairs of points for 3D scene analysis. In *European Conference on Computer Vision*, Cambridge, UK, pp. 17–30.

Irani, M. and Anandan, P. 1998. A unified approach to moving object detection in 2D and 3D scenes. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20:577–589.

Irani, M. and Anandan, P. 2000. Factorization with uncertainty. In *European Conference on Computer Vision*, Dublin, pp. 539–553.

Irani, M., Anandan, P., and Cohen, M. 1999. Direct recovery of planar-parallax from multiple frames. In *Vision Algorithms 99*, Corfu.

Irani, M., Anandan, P., and Weinshall, D. 1998. From reference frames to reference planes: Multi-view parallax geometry and applications. In *European Conference on Computer Vision*, Freiburg.

Irani, M., Rousso, B., and Peleg, S. 1994. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12:5–16.

Irani, M., Rousso, B., and Peleg, P. 1997. Recovery of ego-motion using region alignment. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(3):268–272.

Kumar, R., Anandan, P., and Hanna, K. 1994. Direct recovery of shape from multiple views: A parallax based approach. In *Proc 12th ICPR*, pp. 685–688.

Longuet-Higgins, H.C. and Prazdny, K. 1980. The interpretation of a moving retinal image. *Proceedings of The Royal Society of London B*, 208:385–397.

Lucas, B.D. and Kanade, T. 1981. An iterative image registration technique with an application to stereo vision. In *Image Understanding Workshop*, pp. 121–130.

Sawhney, H. 1994. 3D geometry from planar parallax. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Shapiro, L.S. 1995. *Affine Analysis of Image Sequences*. Cambridge University Press: Cambridge, UK.

Shashua, A. and Navab, N. 1994. Relative affine structure: Theory and application to 3D reconstruction from perspective views. In *IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, pp. 483–489.

Stein, G.P. and Shashua, A. 1997. Model-based brightness constraints: On direct estimation of structure and motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, San-Juan, pp. 400–406.

Szeliski, R. 1990. Bayesian modeling of uncertainty in low-level vision. *International Journal of Computer Vision*, 5:271–301.

Szeliski, R. and Kang, S.B. 1995. Direct methods for visual scene reconstruction. In *Workshop on Representations of Visual Scenes*.

Tomasi, C. and Kanade, T. 1992. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9:137–154.

Torr, P.H.S. 1998. Geometric motion segmentation and model selection. *Proceedings of The Royal Society of London A*, 356:1321–1340.

Zelnik-Manor, L. and Irani, M. 2000. Multi-frame estimation of planar motion. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22:1105–1116.