Structure and Motion for Dynamic Scenes – The Case of Points Moving in Planes

Peter Sturm

INRIA Rhône-Alpes, 38330 Montbonnot, France,
Peter.Sturm@inrialpes.fr,
http://www.inrialpes.fr/movi/people/Sturm

Abstract. We consider dynamic scenes consisting of moving points whose motion is constrained to happen in one of a pencil of planes. This is for example the case when rigid objects move independently, but on a common ground plane (each point moves in one of a pencil of planes parallel to the ground plane). We consider stereo pairs of the dynamic scene, taken by a moving stereo system, that allow to obtain 3D reconstructions of the scene, for different time instants. We derive matching constraints for pairs of such 3D reconstructions, especially we introduce a simple tensor, that encapsulates parts of the motion of the stereo system and parts of the scene structure. This tensor allows to partially recover the dynamic structure of the scene. Complete recovery of structure and motion can be performed in a number of ways, e.g. using the information of static points or linear trajectories. We also develop a special self-calibration method for the considered scenario.

1 Introduction

Most existing works on structure and motion from images concentrate on the case of rigid scenes. The rigidity constraint allows to derive matching relations among two or more images, represented by e.g. the fundamental matrix or trifocal tensors. These matching tensors encapsulate the geometry/motion of the cameras which took the underlying images, and thus all the geometric information needed to perform 3D reconstruction. Matching tensors for rigid scenes can also be employed for scenes composed of multiple, independently moving objects [1,2], which requires however that enough features be extracted for each object, making segmentation, at least implicitly, possible.

Shashua and Wolf introduced the so-called homography tensors [9] – matching constraints that exist between three views of a planar scene consisting of independently moving points (each point being static or moving on a straight line). Basically, given correspondences of projections of such points in three images, the plane homographies between all pairs of images can be computed from the homography tensor, which would not be possible with only two images of the scene. It is important to note that this does not make any assumption about the camera's motion, i.e. the camera is indeed allowed to move freely between image takings. So, this work is maybe the first that considers scenarios where *everything* is moving independently: the camera as well as any point in the scene [1].

[1] Of course, if the camera were not moving, two images would be enough to do the job (the plane homography between them, for any plane, is intrinsically known – it is the identity).

A. Heyden et al. (Eds.): ECCV 2002, LNCS 2351, pp. 867–882, 2002.

Naturally, the question arises if there are other dynamic scenarios that might be interesting to examine. Wolf et al. considered the case of a rigid stereo system taking stereo pairs of a *threedimensional* scene consisting of points moving on straight lines, but independently from each other [11]. From each stereo pair, a 3D reconstruction of the current state of the scene can be obtained (a projective reconstruction if the cameras are not calibrated). Similarly to the above mentioned work on 2D homography tensors, the aim is now to determine 3D homographies between pairs of 3D reconstructions, that would allow to align them. If the stereo system were static, this would again be no problem: the searched for 3D homography is simply the identity transformation. In case of a *moving* stereo system however, Wolf et al. showed that there exist matching tensors, between three 3D reconstructions, representing the state of the scene at three different time instants. From these so-called join tensors, the 3D homographies between all pairs of 3D reconstructions can be recovered, and the reconstructions can be aligned. These 3D homographies represent in fact the stereo system's motions.

Other works along similar lines include that of Han and Kanade [3,4], who consider points moving with constant velocities (thus on linear trajectories), for the case of affine or perspective cameras. Wolf and Shashua [12] consider several dynamic scenarios, and derive matching constraints by embedding the problem representations in higher-dimensional spaces than e.g. the usual projective 3-space for rigid scenes.

The work presented in this paper is inspired by these works. We consider the following scenario: a moving stereo system taking 2D views of a 3D scene consisting of moving points, each point moving arbitrarily in what we call its *motion plane*. In addition, all motion planes are constrained to belong to the same pencil of planes. The most practical instance of this kind of scenario is the case where all motion planes are parallel to each other and, say, horizontal. This scenario covers for example all scenes where objects move on a common ground plane.

For each time instant considered, the stereo system gives a 3D view of the current state of the scene, which would be a projective reconstruction for example, if the system is uncalibrated. In this paper, we derive matching constraints that exist between such 3D views, and examine which amount of 3D motion and structure information can be recovered from the associated matching tensors. We show that there already exists a matching tensor between two 3D views, for two different time instants. This tensor is more or less the analogue to the fundamental matrix between pairs of 2D views. However, it does not allow full recovery of the stereo system's and the 3D points' structure and motion. Full recovery of these requires additional information, e.g. the knowledge that certain points are static, or that certain points move on linear trajectories (if three or more 3D views are available). In the latter case, the join tensors [11] may be applied, but in our more constrained scenario (pencil of motion planes), a simpler matching constraint exists, that can be estimated with fewer correspondences.

For the special case of parallel motion planes, we present a simple self-calibration method that overcomes singularities that exist without the knowledge of parallelism.

2 Background and Notation

We will both use standard matrix-vector notations, and tensor notation. In tensor notation, points are specified by superscripts, e.g. P^i. Transformations mapping points onto points,

have one superscript and one subscript, e.g. T_i^m. Mapping the point P by T gives a point Q with $Q^m = T_i^m P^i$. Transformations mapping points to hyperplanes, are denoted as e.g. $\mathcal{L}_{ij}$. Let ϵ denote the $3 \times 3 \times 3$ "cross-product tensor", which is defined as $\epsilon_{ijk} a^i b^j c^k = \det A$ where a, b and c are the three columns of matrix A. Among the 27 coefficients of ϵ, 21 are zero (all coefficients with repeated indices), the others are equal to $+1$ or -1.

A *linear line complex* in 3D is a set of lines that are all incident with a line A, the *axis* of the linear line complex [8].

3D lines may be represented via 6-vectors of *Plücker coordinates*. Plücker coordinate vectors are defined up to scale and they must satisfy one bilinear constraint on their coefficients. The Plücker coordinates of a line A can be determined from any two different points on A, as follows. Let B and C be two points on A (represented by 4-vectors of homogeneous coordinates). Define $([A]_\times)^{ij} = B^i C^j - C^i B^j$. This is a skew-symmetric 4×4 matrix and has thus only six different non zero coefficients – these are the Plücker coordinates of A. There are several possibilities of ordering the coefficients to get a 6-vector, we choose the following:

$$[A]_\times = \begin{pmatrix} 0 & -A_4 & A_6 & -A_2 \\ A_4 & 0 & -A_5 & -A_3 \\ -A_6 & A_5 & 0 & -A_1 \\ A_2 & A_3 & A_1 & 0 \end{pmatrix} .$$

3 Problem Statement

We consider a dynamic scene of the type described above. Any point of the scene may move freely[2] inside what we call its *motion plane*. All motion planes form a pencil of planes, whose axis is a 3D line A (see figure 1). For ease of expression, we also call A the *horizon* or *horizon line* of the motion (although A need not be a line at infinity in general). Let the positions of some point at three different time instants be represented by the 4-vectors of homogeneous coordinates, P, P' and P''. We call *point motion* the "displacement" of an individual point between different time instants.

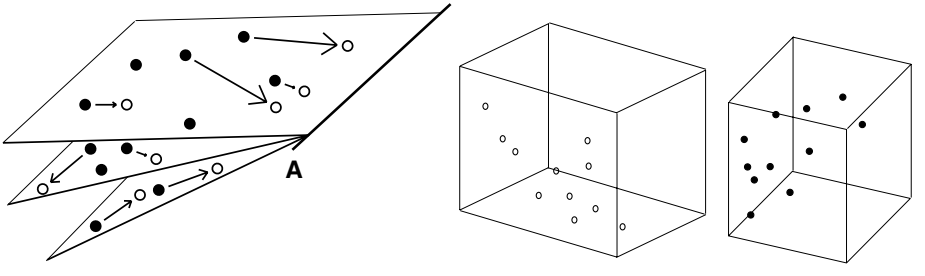

Fig. 1. Left: the considered scenario – points moving in a pencil of motion planes. Right: 3D views at two time instants.

[2] This includes that a point may actually be static.

We consider that the scene is observed my a *moving* stereo system (consisting of two or more cameras). We suppose that at each time instant, a *3D view* of the current state of the scene can be obtained. In the most general case, this will be a projective reconstruction, based on a weak calibration of the stereo system, for the images taken at the considered instant. The stereo system is considered to be moving[3], so different 3D views are represented in different coordinate frames (see figure 1). We call *stereo motion* the transformation between these coordinate frames. Let T' respectively T'' be the transformations mapping points from the second respectively third 3D view, into the frame of the first one. Let Q, Q' and Q'' be the coordinates inside the 3D views, of a moving point P at three time instants, i.e. of the points P, P' and P''. The basic question dealt with in this paper is, which amount of stereo and point motion (i.e. scene structure) can be reconstructed, given the input of matching 3D views.

We first study this question for the case of two 3D views, by deriving the associated matching tensor and showing what information on stereo and point motion can be extracted from it. We show that in general, i.e. for unconstrained motion of individual points inside their motion planes, a complete reconstruction is not possible, even if arbitrarily many views (for arbitrarily many time instants) are available. Several ways of obtaining a complete reconstruction, are then described. These are based on additional knowledge about point motion, e.g. knowledge that individual points are actually static or that points are moving on linear trajectories.

4 Two 3D Views – The Projective Case

4.1 The Matching Tensor – A Kind of 3D Epipolar Geometry

The structure of all points, observed at two time instants, may be represented as a linear line complex: the lines spanned by pairs of corresponding points P and P', are all bound to lie in the pencil of motion planes, thus they all intersect the pencil's axis A.

Let us now consider two 3D views of the dynamic scene, taken at two different time instants, by an uncalibrated stereo system. Hence, the 3D views are projective reconstructions of the scene, at the respective time instants. Let point positions in the first 3D view be denoted as Q and in the second one, as Q'. If we knew the stereo motion T' and A, the point motion's horizon line in the first 3D view, then, after mapping all Q' by T', the lines spanned by corresponding points Q and $T'Q'$, would form a linear line complex, with A as axis, as observed above. Let B and C be any two points on A. We must have coplanarity of $Q, T'Q', B$ and C, thus:

$$\det \begin{pmatrix} | & | & | & | \\ B & C & Q & T'Q' \\ | & | & | & | \end{pmatrix} = 0 \ . \tag{1}$$

This equation is bilinear in the coefficients of the reconstructed 3D points Q and Q' and we may rewrite it in the following form:

$$Q^i Q'^j \mathcal{L}_{ij} = 0 \ , \tag{2}$$

[3] Note that it is nowhere required that the stereo system be moving rigidly or the individual cameras have constant intrinsic parameters or the like.

where $\mathcal{L}$ is a 4×4 matrix, that depends on the stereo motion and the point motion's horizon line. We might call $\mathcal{L}$ a "Linear Line Complex Tensor", or, L-tensor for short, for the reasons given above. The coefficients of the two points B and C, that appear in $\mathcal{L}$, can all be contracted to the Plücker coordinates of A. It is then easy to derive the following decomposition of $\mathcal{L}$:

$$\mathcal{L} \sim T'^{\mathsf{T}} [A]_\times \quad . \tag{3}$$

In the following, we describe several properties of the tensor and in §4.2 we explain, what information can be extracted from it.

The matrix $[A]_\times$ is of rank two at the most, since its coefficients are Plücker coordinates (they satisfy the constraint $A_1 A_4 + A_2 A_5 + A_3 A_6 = 0$). Hence, $\mathcal{L}$ too is of rank two at the most. The right and left null spaces of $\mathcal{L}$ represent nothing else than the horizon line A: the right null space consists of the 3D points that lie on A, in the first 3D view, whereas the left null space contains the 3D points lying on the reconstruction of the horizon line A' in the second 3D view.

In the following, we give some geometric interpretation (cf. figure 2) of the L-tensor, and actually show that there are some analogies to the epipolar geometry between two 2D views of a rigid scene. Let us first consider the action of $\mathcal{L}$ on a point Q in the first 3D reconstruction. The product $([A]_\times)_{ij} Q^i$ gives the motion plane Π_j, that is spanned by the horizon line A and the point Q [8]. The transformation T'^{T} maps planes from the first 3D view, onto planes in the second one: $\Pi'_k \sim \left(T'^{\mathsf{T}}\right)^j_k \Pi_j$. The plane Π' contains the horizon line A'. The correspondence of a point Q' with Q is then expressed as Q' lying on Π', or: $Q'^k \Pi'_k = 0$.

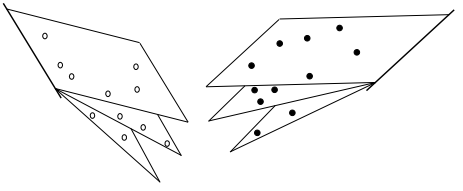

Fig. 2. 3D epipolar geometry.

The analogy to the 2D epipolar geometry is straightforward. The horizon lines A and A' (they represent the same "physical" line, but in 3D views taken at different stereo positions) play the role of the epipoles. In each 3D view, there is a pencil of "epipolar motion planes" containing the horizon line, which is analogous to pencils of epipolar lines in 2D views. Concerning the transformation T', there is an analogous expression to $\mathcal{L} \sim T'^{\mathsf{T}} [A]_\times$ for the 2D epipolar geometry: any plane homography, multiplied by the skew-symmetrix matrix of an epipole, gives the fundamental matrix. Plane homographies are those 2D homographies that map one epipole onto the other and that map corresponding epipolar lines onto each other. Hence, plane homographies are defined up to three parameters. Here, T' is a 3D homography. It is constrained to map

epipoles A and A' onto each other and to map corresponding motion planes onto each other. A difference compared to the 2D epipolar geometry is that here, the epipoles (the horizon lines) do represent a part of the dynamic scene structure, and not only the camera geometry. Also, for a given 3D view, the epipole with respect to any other 3D view of the same dynamic scene, is always the same, whereas in the 2D case, the epipoles of one view with respect to several other views, are in general different from each other.

4.2 What Can Be Extracted from the L-Tensor?

It would be desirable to extract, from the tensor $\mathcal{L}$, the stereo system's motion T' and the point motion's horizon line A. Unhappily, this is not entirely possible, which is clear when counting parameters: $\mathcal{L}$ offers at most 11 constraints (it is a rank-2 4×4 matrix, defined up to scale), which is not sufficient to cover the 15+4 parameters for T' and A.

From the decomposition of $\mathcal{L}$ in (3), it is clear that the horizon line A can be extracted via the right nullspace of $\mathcal{L}$ (the horizon line A' in the second 3D view is the left nullspace). The question left is, how much information can be gained on the stereo motion T' ? Let H be any non-singular 3D homography that maps A to the line at infinity consisting of all points $(X, Y, 0, 0)^\mathsf{T}$. It can be shown that multiplying equation (3) from the right with the inverse of H leads to:

$$\mathcal{L}H^{-1} \sim {T'}^\mathsf{T} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix} \sim \begin{pmatrix} 0 & 0 & -T'_{41} & T'_{31} \\ 0 & 0 & -T'_{42} & T'_{32} \\ 0 & 0 & -T'_{43} & T'_{33} \\ 0 & 0 & -T'_{44} & T'_{34} \end{pmatrix} \quad .$$

Hence, $\mathcal{L}$ gives us 7 coefficients of T' (discarding the scale ambiguity).

Let M' be any 4×4 matrix whose third and fourth rows are the same as that of T', but with arbitrary coefficients in the first two rows. Any such M' maps the horizon line A' of the second 3D view onto A in the first 3D view (to be precise, it maps all points on A' onto points on A) and the motion planes of the second 3D view (planes spanned by the Q' and the line A'), onto the corresponding motion planes in the first view. What remains unknown however, is the motion *inside* the individual motion planes.

Mapping the second 3D view by any such M' will in the sequel be called *partial alignment of 3D views*. Methods for obtaining a *full alignment* are described further below. We now describe one method of performing partial alignment. Since everything is defined up to a global projective transformation, we perform the alignment such that the horizon line becomes the line at infinity, consisting of all points $(X, Y, 0, 0)^\mathsf{T}$, which leads to simpler expressions in the sequel. Let the Singular Value Decomposition (SVD) of $\mathcal{L}$ be given as (remember that $\mathcal{L}$ is of rank two):

$$\mathcal{L} = U \begin{pmatrix} a & & & \\ & b & & \\ & & 0 & \\ & & & 0 \end{pmatrix} V^\mathsf{T} \quad .$$

Define the following projective transformations:

$$M = \begin{pmatrix} & & \sqrt{a} \\ & \sqrt{b} & \sqrt{a} \\ -\sqrt{a} & & \end{pmatrix} V^{\mathsf{T}} \qquad M' = \begin{pmatrix} & \sqrt{a} & \\ \sqrt{a} & & \sqrt{a} \\ & \sqrt{b} & \end{pmatrix} U^{\mathsf{T}} \ .$$

These transformations are by construction non-singular (unless $a = 0$ or $b = 0$). Transforming the first 3D view by M and the second one by M' leads to points MQ and $M'Q'$ that satisfy the following constraint:

$$(M'Q')^{\mathsf{T}} \, \mathcal{L}' \, MQ = 0$$

where

$$\mathcal{L}' = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

is the L-tensor of the partially aligned 3D views.

Before describing methods for full alignment, which will be done in §6, we consider the specialization of our scenario to the Euclidean and affine cases.

5 Two 3D Views – The Euclidean Case

We now consider the case where the 3D views are Euclidean reconstructions, obtained using e.g. a calibrated stereo system. In addition, we concentrate on the case of parallel motion planes, which is probably the most interesting one to study. This means that A is a line at infinity, thus $A_4 = A_5 = A_6 = 0$ and

$$\mathcal{L} \sim {T'}^{\mathsf{T}} [A]_{\times} \sim {T'}^{\mathsf{T}} \begin{pmatrix} 0 & 0 & 0 & -A_2 \\ 0 & 0 & 0 & -A_3 \\ 0 & 0 & 0 & -A_1 \\ A_2 & A_3 & A_1 & 0 \end{pmatrix} \ .$$

The vector $a = (A_1, A_2, A_3)^{\mathsf{T}}$ contains the homogeneous coordinates of the line A, on the plane at infinity. Thus, it also represents the homogeneous coordinates of the normal direction of all motion planes.

For Euclidean 3D views, the stereo motion is a similarity transformation, i.e. a rigid motion possibly followed by a global scale change, which is needed since the two 3D views might have different scales. Thus:

$$T' = \begin{pmatrix} & & \\ & sR & t \\ & & \\ 0^{\mathsf{T}} & & 1 \end{pmatrix}$$

for a rotation matrix R, a translation vector t and a scalar s. The tensor is thus given by:

$$\mathcal{L} = \begin{pmatrix} 0_{3\times 3} & -sR^\mathsf{T}a \\ a^\mathsf{T} & -t^\mathsf{T}a \end{pmatrix} .$$

It has a particularly simple structure with only 7 non zero coefficients, and no non-linear constraint on them. However, if the global scale s of the second 3D view, is known in advance, e.g. due to constant stereo calibration in which case $s = 1$, then there is one non-linear constraint: the norm of the leading 3-vectors in the 4th column and the 4th row of $\mathcal{L}$ are the same.

What information on stereo and point motion can be extracted from $\mathcal{L}$? The horizon line can be read off directly, as the leading 3-vector of the 4th row. The scale s is obtained as the ratio of the norms of the two leading 3-vectors in the 4th column and 4th row. As for R, it can be seen that it can be determined, up to a rotation about a, the normal direction of the motion planes (see above). Finally, as for the translation t, only its amount along the direction a, can be determined.

Thus, the L-tensor allows, like in the projective case, only partial alignment of 3D views. Here, the ambiguity has three degrees of freedom: let T' be any similarity transformation doing the partial alignment. Adding any transformation consisting of a rotation about a and a translation perpendicular to a, will also result in a valid alignment transformation. Contrary to the projective case, the ambiguous transformation is the same for all motion planes, i.e. if the ambiguity can be cancelled in one motion plane only, then it can be so for the entire 3D scene alignment (in the projective case, full alignment of at least two planes is necessary).

6 Three 3D Views

As discussed previously, two 3D views are not sufficient for full alignment. We now examine if and how additional 3D views, obtained at other time instants, allow to reduce the ambiguity. Let us first note that even with three or more 3D views of our scenario, without additional information, full alignment is not possible. Every 3D view can be partially aligned with the others, as described above, but it is easy to see that the ambiguity in the alignment can not be reduced without further information. In the following, we outline a few types of additional information, that indeed may contribute to full alignment of 3D views.

First, suppose that every point has a linear trajectory. Wolf and Shashua have derived the matching constraints for three 3D views of this scenario [11]. The so-called join tensors, or J-tensors for short, allow to perform full alignment of the three 3D views. This holds even if the linear trajectories are in general position, i.e. if they are not bound to lie on a pencil of planes. The drawback of this general case is that a linear solution of the J-tensors requires at least 60 corresponding point triplets. In §6.1, we specialize the J-tensors to our scenario, and show how this allows full alignment using fewer correspondences.

Second, we remind that until now we did not assume that there are more than one point per motion plane. Thus, it might be interesting to study the case of one or several motion

planes containing several points. This can actually be detected after partial alignment, see §7.2. In this case, motion planes with enough moving points on them, can be dealt with individually, e.g. by estimating their homography tensor [9]. Since in our scenario, we already know at least one line correspondence per motion plane (the horizon line), we might consider a simplified version of the H-tensor (see §6.2). Each motion plane for which the H-tensor can be estimated, can thus be fully aligned, and it is easy to show that the alignment of two or more motion planes is sufficient to align the rest of the scene.

Third, knowledge of static points helps of course in the alignment of the 3D views. This will be described briefly in §6.3. Other possibly useful types of additional information could be knowledge of conical trajectories, of motion with constant velocity, of linear trajectories going in the same direction, etc.

6.1 Linear Trajectories

The join tensors, introduced for the general case of unconstrained linear trajectories [11], can also be used here of course. However, in our specialized scenario, we can exploit the additional constraint that the trajectories form a linear line complex (they lie in a pencil of motion planes). It is possible to derive tensors that fully encapsulate this constraint, but they are numerous and not very intuitive. Rather, we suppose in the following that partial alignment of the three 3D views has been performed (e.g. the second and third views have been aligned with the first one), as described in §4.2, and derive matching constraints on the already partially aligned 3D views.

We remind that the horizon line A in the aligned 3D views, is the line at infinity consisting of points $(X, Y, 0, 0)^\mathsf{T}$. Hence, the motion planes are given by 4-vectors of homogeneous coordinates of the form $(0, 0, s, -t)^\mathsf{T}$. We are looking for transformations T' and T'' for the second and third 3D views, that leave the horizon line and all motion planes globally fixed. Hence, the transformations are of the following form:

$$T' = \begin{pmatrix} a' & b' & c' & d' \\ e' & f' & g' & h' \\ 0 & 0 & j' & 0 \\ 0 & 0 & 0 & j' \end{pmatrix} \qquad T'' = \begin{pmatrix} a'' & b'' & c'' & d'' \\ e'' & f'' & g'' & h'' \\ 0 & 0 & j'' & 0 \\ 0 & 0 & 0 & j'' \end{pmatrix} . \tag{4}$$

Let Q, Q' and Q'' represent triplets of corresponding points. The matching constraint used here is that $Q, T'Q'$ and $T''Q''$ have to be collinear, which means that the rank of the 4×3 matrix composed of these 3 vectors, is two at the most. This constraint can be expressed by a linear family with four degrees of freedom, of $4 \times 4 \times 4$ join tensors [11]. In our case, due to the special form of T' and T'', the four degrees of freedom remain, but some coefficients are known to be zero for all join tensors, i.e. fewer than the 60 correspondences for the general family of join tensors, are needed here to estimate them.

One problem here, that actually turns out as a benefit, is that the point correspondences available to us in the considered scenario, are constrained – all triplets of points Q, Q', Q'' lie in some "horizontal" plane (in the chosen projective frame). Hence, the

estimation of the join tensors is underconstrained [4]. Hence, if we were to estimate general $4 \times 4 \times 4$ tensors, there would be a family of solutions of degree higher than four.

We now consider matching constraints for triplets of points lying in a motion plane $(0, 0, s, -t)^\mathsf{T}$, thus $Q \sim (X, Y, t, s)^\mathsf{T}, Q' \sim (X', Y', t, s)^\mathsf{T}$ and $Q'' \sim (X'', Y'', t, s)^\mathsf{T}$. These being collinear after alignment, is expressed by:

$$\text{rank} \left(\begin{array}{c|c|c} X & a'X' + b'Y' + c't + d's & a''X'' + b''Y'' + c''t + d''s \\ Y & e'X' + f'Y' + g't + h's & e''X'' + f''Y'' + g''t + h''s \\ t & j't & j''t \\ s & j's & j''s \end{array} \right) \leq 2 \ .$$

In the general case, four join tensors forming a basis for the 4-degree-of-freedom family, might be extracted by expressing that the four possible 3-minor's determinants vanish. With our input, it is clear that the two minors containing both the third and fourth rows of the above matrix, always vanish. Hence, the corresponding join tensors can not be estimated. As for the other two minors, we may write:

$$\epsilon_{lpq} \left(M_i^l Q^i \right) \left(M_m^p T_j'^m Q'^j \right) \left(M_n^q T_k''^n Q''^k \right) = 0$$
$$\epsilon_{lpq} \left(M_i'^l Q^i \right) \left(M_m'^p T_j'^m Q'^j \right) \left(M_n'^q T_k''^n Q''^k \right) = 0$$

where the 3×4 matrices M respectively M' project points $(X, Y, Z, W)^\mathsf{T}$ to $(X, Y, Z)^\mathsf{T}$ respectively $(X, Y, W)^\mathsf{T}$, i.e.

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad M' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \ .$$

We thus obtain the following two join tensors:

$$\mathcal{J}_{ijk} = \epsilon_{lpq} M_i^l \left(M_m^p T_j'^m \right) \left(M_n^q T_k''^n \right) \qquad \mathcal{J}'_{ijk} = \epsilon_{lpq} M_i'^l \left(M_m'^p T_j'^m \right) \left(M_n'^q T_k''^n \right) \ .$$

The slices $\mathcal{J}_{4jk}$ and $\mathcal{J}'_{3jk}$ are zero matrices, and $\mathcal{J}_{3jk}$ and $\mathcal{J}'_{4jk}$ are identical. As for the other two slices, the coefficients with indices lower than 3 inside the slice, are identical in the two tensors. Among the other coefficients, there are several that are the same in both tensors, but that stand at different places. Each one of $\mathcal{J}$ and $\mathcal{J}'$ has only 30 non-zero coefficients. However, again due to the specific type of input correspondences, the tensors can only be estimated up to a 3-degree-of-freedom family of solutions each. Happily, the nature of the ambiguity in the solutions, is known and simple: 24 of the non-zero coefficients for each tensor, can be estimated without ambiguity (up to scale). As for the remaining coefficients, what can be estimated are the following sums: $\mathcal{J}_{134} + \mathcal{J}_{143}, \mathcal{J}_{234} + \mathcal{J}_{243}, \mathcal{J}_{334} + \mathcal{J}_{343}$ and $\mathcal{J}'_{134} + \mathcal{J}'_{143}, \mathcal{J}'_{234} + \mathcal{J}'_{243}, \mathcal{J}'_{434} + \mathcal{J}'_{443}$.

So, 26 point correspondences are in general sufficient to obtain a linear solution for the 24 coefficients and the 3 sums of coefficients (per tensor). The following coefficients of the alignment transformations can be read off directly from the estimated tensor

[4] It is important to note that although T' and T'' conserve motion planes, their join tensors also express the fact that $Q, T'Q'$ and $T''Q''$ may be collinear, for points Q, Q' and Q'' not lying in the same motion plane.

coefficients (after an arbitrary choice for j' and j''): $a', b', e', f', a'', b'', e'', f''$. Having determined them, one can establish, using coefficients of $\mathcal{J}$ and $\mathcal{J}'$, as well as the estimated values of a' etc., a simple linear equation system, to solve for the remaining 8 unknowns, $c', d', g', h', c'', d'', g''$ and h''.

In summary, 26 correspondences are sufficient to determine the alignment transformations T' and T'', and it is nowhere required that there be more than a single moving point per motion plane.

6.2 Using Homography Tensors

We consider the same scenario as in the previous section, i.e. linear trajectories, but now suppose that there are motion planes carrying several points (which can be detected, see §7.2). In this case, we may deal with each motion plane separately.

Consider one motion plane, represented by $(0, 0, s, -t)^\mathsf{T}$. Let Q, Q' and Q'' represent triplets of corresponding points on that plane. Hence, they have the form given in the previous section. The matching constraint for such triplets, corresponds to the homography tensor, or H-tensor for short, introduced in [9]. In that work, the matching constraint was derived for three 2D views of a dynamic planar scene, obtained by a moving 2D camera. Each such 2D view constitutes a projective reconstruction of the planar scene, at the corresponding time instant. Here, we start with three 3D views, which essentially gives us again three projective reconstructions of the motion plane considered.

In order to compute the homography tensor for a motion plane, we first project the three 3D views of the plane by some projection matrix onto some 2D image plane. Let us define the 4×4 projection matrix M:

$$M = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & t & s \end{pmatrix}$$

whose optical center is guaranteed not to lie on the considered motion plane. We project all three 3D views using M. For the resulting 2D views, there must exist 3×3 transformations H' and H'' such that all triplets $MQ, H'MQ'$ and $H''MQ''$ are collinear. In addition, we know a correspondence of a static line, the motion's horizon line. This line is mapped, by M, to the line at infinity of the 2D views. Hence, H' and H'' must be affine transformations of the form:

$$H' = \begin{pmatrix} a' & b' & c' \\ e' & f' & g' \\ 0 & 0 & j' \end{pmatrix} \qquad H'' = \begin{pmatrix} a'' & b'' & c'' \\ e'' & f'' & g'' \\ 0 & 0 & j'' \end{pmatrix} \quad .$$

We obtain the following matching equation, in tensorial notation:

$$\epsilon_{lmn} \left(M_i^l Q^i\right) \left(H_p'^m M_j^p Q'^j\right) \left(H_q''^n M_k^q Q''^k\right) = 0$$

We may rewrite the equation:

$$\left(M_i^l Q^i\right) \left(M_j^p Q'^j\right) \left(M_k^q Q''^k\right) \underbrace{\left(\epsilon_{lmn} H_p'^m H_q''^n\right)}_{\mathcal{H}_{lpq}} = 0 \quad .$$

We may identify $\mathcal{H}_{lpq}$ as the $3 \times 3 \times 3$ homography tensor. It can be shown that, due to the constrained form of H' and H'', the tensor has only 19 non-zero coefficients (compared to 27 for the general H-tensor). Hence, a linear solution is possible with 18 or more correspondences. Extracting the individual transformations H' and H'' from $\mathcal{H}$ can be done analogously to what is described in [9].

The tensor $\mathcal{H}$, for one motion plane, allows to partially determine $\mathcal{J}$ and $\mathcal{J}'$ (valid for all motion planes), dealt with in §6.1. Several coefficients of $\mathcal{H}$ occur identically in $\mathcal{J}$ or $\mathcal{J}'$, and the others give linear equations on coefficients of the join tensors.

It is easy to show that the alignment of two motion planes is sufficient to fully align the entire 3D views: for any 3D point in, say, the second 3D view, which we will call Q', let D' be a line passing through it, but that is not contained in Q's motion plane. Let B' and C' be the intersection points of D' with the two motion planes for which full alignment is possible. We thus may compute the positions B and C of the points B' and C', after alignment. The aligned position Q of Q' is finally given by the intersection of Q's motion plane, with the line joining B and C.

6.3 Using Static Points

Given the special form of the alignment matrices (see (4)), it is clear that one static point (that is known to be static), provides two independent equations on each of them. Hence, correspondences associated to four static points in general position, should be sufficient to achieve full alignment of the 3D views (compared to five correspondences that would be required without the specific nature of our scenario). In the Euclidean case, two point correspondences (actually, one and a half) are sufficient for full alignment (compared to three that would be required without the specific nature of our scenario). It would be interesting to study the general case of mixed static and moving points.

7 Other Issues

7.1 Projections $\boldsymbol{P^5 \rightarrow P^3}$

The derivation of the matching tensor for the two-view scenario (see §4.1), could also be performed in the framework of higher dimensional projection matrices used in [12]. Without loss of generality (we deal with projective space), suppose that the horizon A is the line at infinity containing all points $(X, Y, 0, 0)^\mathsf{T}$. Let $P \sim (X, Y, Z, W)^\mathsf{T}$ be a 3D point at the first time instant and let $P' \sim (X + a, Y + b, Z, W)^\mathsf{T}$ be the same point, at the second instant, after having moved in the plane spanned by A and P. We may form the following 6-vector that represents P and its moved version P': $S^\mathsf{T} = \begin{pmatrix} X & Y & a & b & Z & W \end{pmatrix}$. We define two projection matrices $P^5 \rightarrow P^3$:

$$M \sim \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad M' \sim \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} .$$

We may observe that $MS \sim P$ and $M'S \sim P'$. In our scenario, we do not observe P and P' directly, but have projective 3D views of them, i.e.: $\lambda Q = TP$ and $\lambda' Q' = T' P'$

for some 4×4 projective transformations T and T' and scale factors λ and λ'. We may derive the matching constraints for Q and Q' in the way shown e.g. in [10]: due to

$$\underbrace{\left(\begin{array}{c|c|c} TM & Q & 0 \\ T'M' & 0 & Q' \end{array}\right)}_{X_{8\times 8}} \left(\begin{array}{c} S \\ \hline -\lambda \\ \hline -\lambda' \end{array}\right) = 0$$

we know that the matrix X is rank-deficient, i.e. that its determinant is equal to zero. By developing the determinant, one obtains the same 4×4 tensor $\mathcal{L}$ as in §4.1 (if we set T to the identity).

7.2 Segmentation of Points Moving in the Same Motion Plane

After partial alignment (see §4.2), the segmentation of points that move in the same plane, is straightforward and can in principle be done in a single 3D view. This might be done by checking, in 3D, if points are on the same motion plane. An alternative would be to compute plane homographies between the 2D views inside a stereo system, for individual motion planes, and check if corresponding projections of 3D points in the 2D views, are consistent with the plane homographies.

7.3 Self-Calibration

We briefly describe a self-calibration algorithm for the scenario of two projective 3D views, under the assumption that the true motion planes are parallel to each other, i.e. the true horizon line is a line at infinity. Using the L-tensor, the horizon line can be determined in the 3D views. Since the true line is a line at infinity, it has two intersection points with the absolute conic – the circular points of all motion planes. We may perform partial self-calibration by searching for the circular points, on the reconstructed horizon lines in our 3D views.

Consider one of the 3D views, after partial alignment as described in §4.2. We suppose that this 3D view has been obtained using two perspective cameras, with unknown and possibly different focal lengths, but known other intrinsic parameters. The two focal lengths can in general be recovered from the epipolar geometry [5], but this is nearly always singular in practice, due to optical axes passing close to each other [7]. The knowledge of a line at infinity in the projective reconstruction, however, can be used to overcome the singularity, as described in the following.

Let M and M' be the 3×4 projection matrices of the two 2D views. We suppose that the known parts of the calibration matrices (containing aspect ratio and principal point) have been undone, i.e. the unknown calibration matrices of M and M' are $K = \mathrm{diag}(f, f, 1)$ and $K' = \mathrm{diag}(f', f', 1)$. We parameterize the problem in the circular points on the horizon line, which, in the partially aligned 3D view, have coordinates $C_\pm \sim (a \pm I, b, 0, 0)^\mathsf{T}$ for real a, b and $b \neq 0$. Our self-calibration constraints are that the projections of C_+ and C_- lie on the images of the absolute conic in the respective views, which leads to:

$$(am_1 + bm_2 + Im_1)^\mathsf{T} K^{-\mathsf{T}} K^{-1} (am_1 + bm_2 + Im_1) = 0$$
$$(am_1 + bm_2 - Im_1)^\mathsf{T} K^{-\mathsf{T}} K^{-1} (am_1 + bm_2 - Im_1) = 0$$

where m_i is the ith column of M, and similar equations for the second view. Separating the real and imaginary parts of the equations leads to two equations, whose resultant with respect to f^2 is quadratic in a and b. We get a similar equation for the second view. The resultant of these two equations, with respect to a, finally, is the product of the term b^2 and a term that is linear in b^2. Since $b \neq 0$, we thus get a single solution for b^2, which gives us b up to sign (the sign does not matter). From b, unique solutions for a and the squared focal lengths may then be obtained.

We performed simulated experiments with this method. Twenty moving points on each of three planes were simulated. The 3D points were projected in two stereo pairs, and centered Gaussian noise with a standard deviation of 1 pixel was added to the image coordinates. For each stereo pair, the fundamental matrix was computed using the 8-point method [6], projective reconstruction was performed, and the L-tensor between the two resulting point sets estimated. The point sets were then partially aligned. For several stereo configurations (varying vergence angle), 100 simulation runs each were performed. Self-calibration gave focal lengths with an average relative error of about 6% (excepting between 0 and 4 runs were computation failed).

8 Experimental Results

We conducted the following experiment using four stereo pairs of a dynamic scene (see figure 3). About 60 points on the moving objects were manually extracted in all eight images. The experiment was performed for the Euclidean case: the calibration grid visible in the images, was used to obtain full stereo calibration, and thus Euclidean 3D reconstruction of the points. Each such 3D view underwent an arbitrary Euclidean transformation, otherwise they would already have been aligned, since stereo calibration was with respect to the static calibration grid.

From this input, the Euclidean L-tensor between the first and second 3D views was estimated and these 3D views were partially aligned (see §4.2). Then, the other two 3D views were aligned with the two first one, using a simpler variant of the method of §4.2 (the horizon line in the first two views is already known), not described here.

Full alignment was done for the first three 3D views, based on the knowledge that individual points moved on linear trajectories (see §6.1), and by estimating join tensors specialized to Euclidean alignment transformations. Finally, the fourth 3D view was fully aligned with the others, again using a simplification of the method of §6.1. Some recovered point trajectories are shown in figure 4. Qualitatively, the result seems to be correct, although a quantitative evaluation should definitely be carried out.

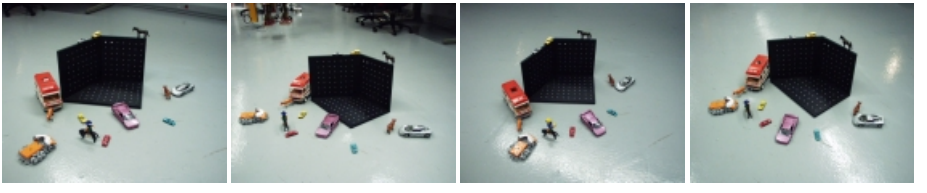

Fig. 3. Two stereo pairs used in the experiments.

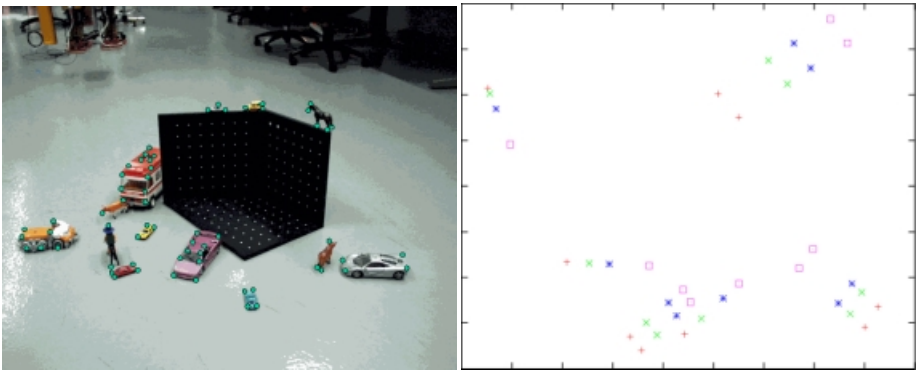

Fig. 4. Left: the moving points used in the experiment. Right: recovered linear trajectories of several points (4 positions each), orthogonally projected onto the ground plane. The point group on the left corresponds to a point on the horse, the group at the bottom to the caravan (3 moving points shown), the group in between, to one of the cars on the grid. The other two point groups belong to the truck and to the sportscar in front of the grid (2 moving points each).

9 Conclusion

We have considered the structure and motion problem for a dynamic scene, consisting of individually moving points, with the restriction that motion happens in a pencil of motion planes. The scene is supposed to be observed by a moving stereo system, resulting in 3D views of the scene, at different time instants. We have derived the matching constraints between two such 3D views, and shown that full alignment of the views is not possible without further information. Information useful to fully recover the motion of the stereo system as well as the motion and structure of the scene, are for example knowledge of static points or linear trajectories. We have especially discussed how to take into account linear trajectories, to achieve full recovery of structure and motion. A preliminary experiment has shown that it may be feasible to solve the problem in practice, at least in the calibrated case.

Among issues for further work on this topic, minimum numbers of correspondences for the mixed case of moving and known/unknown static points, should be established, and a more thorough experimentation is needed.

Acknowledgement. I wish to thank Adrien Bartoli for preparing the experimental data of section 8.

References

1. Costeira, J., Kanade, T.: A Multi-Body Factorization Method for Motion Analysis. ICCV – International Conference on Computer Vision (1995) 1071–1076

2. Fitzgibbon, A.W., Zisserman, A.: Multibody Structure and Motion: 3-D Reconstruction of Independently Moving Objects. ECCV – European Conference on Computer Vision (2000) 891–906
3. Han, M., Kanade, T.: Reconstruction of a Scene with Multiple Linearly Moving Objects. CVPR – International Conference on Computer Vision and Pattern Recognition, Vol. II (2000) 542–549
4. Han, M., Kanade, T.: Multiple Motion Scene Reconstruction from Uncalibrated Views. ICCV – International Conference on Computer Vision, Vol. I (2001) 163–170
5. Hartley, R.I.: Estimation of Relative Camera Positions for Uncalibrated Cameras. ECCV – European Conference on Computer Vision (1992) 579–587
6. Hartley, R.: In Defence of the 8-Point Algorithm. ICCV – International Conference on Computer Vision (1995) 1064–1070
7. Newsam, G.N., Huynh, D.Q., Brooks, M.J., Pan, H.P.: Recovering Unknown Focal Lengths in Self-Calibration: An Essentially Linear Algorithm and Degenerate Configurations. XVIIIth ISPRS Congress, Part B3 (1996) 575–580
8. Semple, J.G., Kneebone, G.T.: Algebraic Projective Geometry. Oxford Science Publications (1952)
9. Shashua, A., Wolf, L.: Homography Tensors: On Algebraic Entities That Represent Three Views of Static or Moving Planar Points. ECCV – European Conference on Computer Vision (2000) 507–521
10. Triggs, B.: Matching Constraints and the Joint Image. ICCV – International Conference on Computer Vision (1995) 338–343
11. Wolf, L, Shashua, A., Wexler, Y.: Join Tensors: On 3D-to-3D Alignment of Dynamic Sets. ICPR – International Conference on Pattern Recognition (2000) 388–391
12. Wolf, L., Shashua, A.: On Projection Matrices $P^k \rightarrow P^2, k = 3, \cdots, 6$, and their Applications in Computer Vision. ICCV – International Conference on Computer Vision, Vol. I (2001) 412–419