Affine Structure from Line Correspondences with Uncalibrated Affine Cameras

Long QUAN and Takeo KANADE

Abstract— **This paper presents a linear algorithm for recovering 3D affine shape and motion from line correspondences with uncalibrated affine cameras. The algorithm requires a minimum of seven line correspondences over three views. The key idea is the introduction of a one-dimensional projective camera. This converts 3D affine reconstruction of "line directions" into 2D projective reconstruction of "points". In addition, a line-based factorisation method is also proposed to handle redundant views. Experimental results both on simulated and real image sequences validate the robustness and the accuracy of the algorithm.**

Key-words: structure from motion, affine structure, factorisation method, line correspondence, affine camera, one-dimensional camera, uncalibrated image.

I. Introduction

Using line segments instead of points as features has attracted the attention of many researchers [1], [2], [3], [4], [5], [6], [7], [8], [9] for various tasks such as pose estimation, stereo and structure from motion. In this paper, we are interested in structure from motion using line correspondences across mutiple images. Line-based algorithms are generally more difficult than point-based ones for the following two reasons. The parameter space of lines is non linear, though lines themselves are linear subspaces, and a line-to-line correspondence contains less information than a point-to-point one as it provides only one component of the image plane displacement instead of two for a point correspondence. A minimum of three views is essential for line correspondences, whereas two views suffice for point ones. In the case of calibrated perspective cameras, the main results on structure from line correspondences were established in [4], [10], [5]: With at least six line correspondences over three views, nonlinear algorithms are possible. With at least thirteen lines over three views, a linear algorithm is possible. The basic idea of the thirteen-line linear algorithm is similar to the "eight-point" one [11] in that it is based on the introduction of a redundant set of intermediate parameters. This significant over-parametrization of the problem leads to the instability of the algorithm reported in [4]. The thirteen-line algorithm was extended to uncalibrated camera case in [12], [9]. The situation here might be expected to be better, as more free parameters are introduced. However, the 27 tensor components that are introduced as intermediate parameters are still subject to 8 complicated algebraic constraints. The algorithm can hardly be stable. A subsequent nonlinear optimization step is almost unavoidable to refine the solution [5], [4], [10], [12].

Long QUAN is with CNRS-GRAVIR-INRIA, ZIRST 655, avenue de l'Europe, 38330 Montbonnot, France. E-mail: Long.Quan@imag.fr
Takeo KANADE is with The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, U.S.A. E-mail: tk@cs.cmu.edu

In parallel, there has been a lot of work [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [14], [16], [23], [17], [24], [25] on structure from motion with simplified camera models varing from orthographic projections via weak and para-perspective to affine cameras, almost exclusively for point features. These simplified camera models provide a good approximation to perpsective projection when the width and depth of the object are small compared to the viewing distance. More importantly, they expose the ambiguities that arise when perspective effects diminish. In such cases, it is not only easier to use these simplified models but also advisable to do so, as by explicitly eliminating the ambiguities from the algorithm, one avoids computing parameters that are inherently ill-conditioned. Another important advantage of working with uncalibrated affine cameras is that the reconstruction is affine, rather than projective as with uncalibrated projective cameras.

Motivated on the one hand by the lack of satisfactory line-based algorithms for projective cameras and on the other by the fact that the affine camera is a good model for many practical cases, we investigate the properties of projection of lines by affine cameras and propose a linear algorithm for affine structure from line correspondences. The key idea is the introduction of a one-dimensional projective camera. This converts the 3D affine reconstruction of "line directions" into 2D projective reconstruction of "points". The linear algorithm requires a minimum of seven lines over three images. We also prove that seven lines over three images is the strict minimum data needed for affine structure from uncalibrated affine cameras and that there are always two possible solutions. This result extends the previous results of Koenderink and Van Doorn [14] for affine structure with a minimum of two views and five points. To deal with redundant views, we also present a line-based factorisation algorithm which extends the previous point-based factorisation methods [18], [21], [22]. A preliminary version of this work was presented in [26].

The paper is organized as follows. In Section II, the affine camera model is briefly reviewed. Then, we investigate the properties of projection of lines with the affine camera and introduce the one-dimensional projective camera in Section III. Section IV is focused on the study of the uncalibrated one-dimensional camera, and in this section we present also a linear algorithm for 2D projective reconstruction which is equivalent to the 3D affine reconstruction of line directions. Later, the linear estimation of the translational component of the uncalibrated affine camera is given in Section V and the affine shape recovery is described in Section VI. To handle redundant views, a line-based factorisation method is proposed in Section IX. The passage to metric structure from the affine structure using known camera parameters will be described in Section XI. Finally in Section XIII, discussions and some concluding remarks are given.

Throughout the paper, tensors and matrices are denoted in upper case boldface, vectors in lower case boldface and scalars in either plain letters or lower case Greek.

II. Review of the affine camera model

For a projective (pin-hole) camera, the projection of a point $\mathbf{x} = (x, y, z, t)^T$ of $\mathcal{P}^3$ to a point $\mathbf{w} = (u, v, w)^T$ of $\mathcal{P}^2$ can be described by a 3×4 homogeneous projection matrix $\mathbf{P}_{3\times 4}$:

$$\lambda \mathbf{w} = \mathbf{P}_{3\times 4}\mathbf{x}. \tag{1}$$

For a restricted class of camera models, by setting the third row of the perspective camera $\mathbf{P}_{3\times 4}$ to $(0, 0, 0, \lambda)$, we obtain the affine camera initially introduced by Mundy and Zisserman [27],

$$\begin{aligned}\mathbf{A}_{3\times 4} &= \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & 0 & 0 & p_{34} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{M}_{2\times 3} & \\ \mathbf{0}_{1\times 3} & \mathbf{t}_{3\times 1} \end{pmatrix}. \end{aligned} \tag{2}$$

The affine camera $\mathbf{A}_{3\times 4}$ encompasses the uncalibrated versions of the orthographic, weak perspective and paraperspective projection models. These reduced camera models provide a good approximation to the perspective projection model when the depth of the object is small compared to the viewing distance. For more detailed relations and applications, one can refer to [20], [22], [?], [28], [13].

For points in the affine spaces $I\!\!R^3$ and $I\!\!R^2$, they are naturally embedded into $\mathcal{P}^3$ and $\mathcal{P}^2$ by the mappings $\mathbf{w}_a \mapsto \mathbf{w} = (\mathbf{w}_a, 1)^T$ and $\mathbf{x}_a \mapsto \mathbf{x} = (\mathbf{x}_a, 1)^T$. We have thus

$$\mathbf{w}_a = \mathbf{M}_{2\times 3}\mathbf{x}_a + \mathbf{t}_0,$$

where $\mathbf{t}_0 = (t_1/t_3, t_2/t_3)^T = (p_{14}/p_{34}, p_{24}/p_{34})^T$. If we further use relative coordinates of the points with respect to a given reference point (for instance, the centroid of the set of points), the vector $\mathbf{t}_0$ is cancelled and we obtain the following linear mapping between space points and image points:

$$\Delta \mathbf{w}_a = \mathbf{M}_{2\times 3}\Delta \mathbf{x}_a. \tag{3}$$

This is the basic equation of the affine camera for points.

III. The affine camera for lines

Now consider a line in $I\!\!R^3$ through a point $\mathbf{x}_0$, with direction $\mathbf{d}_x$:

$$\mathbf{x}_a = \mathbf{x}_0 + \lambda \mathbf{d}_x, \text{ for } \lambda \in I\!\!R.$$

The affine camera $\mathbf{A}_{3\times 4}$ projects this to an image line

$$\begin{aligned}\mathbf{A}_{3\times 4}\begin{pmatrix}\mathbf{x}_a \\ 1\end{pmatrix} &= (\mathbf{M}_{2\times 3}\mathbf{x}_0 + \mathbf{t}_0) + \lambda \mathbf{M}_{2\times 3}\mathbf{d}_x \\ &\equiv \mathbf{w}_0 + \lambda \mathbf{M}_{2\times 3}\mathbf{d}_x\end{aligned}$$

passing through the image point

$$\mathbf{w}_0 = \mathbf{M}_{2\times 3}\mathbf{x}_0 + \mathbf{t}_0,$$

with direction

$$\mathbf{d}_w = \lambda \mathbf{M}_{2\times 3}\mathbf{d}_x. \tag{4}$$

This equation describes a linear mapping between direction vectors of 3D lines and those of 2D lines, and reflects a key property of the affine camera: lines parallel in 3D remain parallel in the image. It can be derived even more directly using projective geometry by considering that the line direction $\mathbf{d}_x$ is the point at infinity $\mathbf{x}_\infty = (\mathbf{d}_x^T, 0)^T$ of the projective line in $\mathcal{P}^3$ and the line direction $\mathbf{d}_w$ is the point at infinity $\mathbf{w}_\infty = (\mathbf{d}_w^T, 0)^T$ of the projective line in $\mathcal{P}^2$. Equation (4) immediately follows as the affine camera preserves the points at infinity by its very definition.

Comparing Equation (4) with Equation (1)—a projection from $\mathcal{P}^3$ to $\mathcal{P}^2$, we see that Equation (4) is nothing but a projective projection from $\mathcal{P}^2$ to $\mathcal{P}^1$ if we consider the 3D and 2D "line directions" as 2D and 1D projective "points". This key observation allows us to establish the following.

The affine reconstruction of line directions with a two-dimensional affine camera is equivalent to the projective reconstruction of points with a one-dimensional projective camera.

One of the major remaining efforts will be concerned with 2D projective reconstruction from the points in $\mathcal{P}^1$. There have been many recent works [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [10], [?], [41] on projective reconstruction and the geometry of multi-views of two dimensional uncalibrated projective cameras. Particularly, the tensorial formalism developed by Triggs [35] is very interesting and powerful. We now extend this study to the case of the one-dimensional camera. It turns out that there are some nice properties which were absent in the 2D case.

IV. Uncalibrated one-dimensional camera

A. Trilinear tensor of the three views

First, rewrite Equation (4) in the following form:

$$\lambda \mathbf{u} = \mathbf{M}_{2\times 3}\mathbf{x} \tag{5}$$

in which we use $\mathbf{u} = (u_1, u_2)^T$ and $\mathbf{x} = (x_1, x_2, x_3)^T$ instead of $\mathbf{d}_w$ and $\mathbf{d}_x$ to stress that we are dealing with "*points*" in the projective spaces $\mathcal{P}^2$ and $\mathcal{P}^1$ rather than "*line directions*" in the vector spaces $I\!R^3$ and $I\!R^2$.

We now examine the matching constraints between multiple views of the same point. Since two viewing lines in the projective plane always intersect in a point, no constraint is possible for less than three views. There is one constraint only for the case of 3 views. Let the three views of the same point $\mathbf{x}$ be given as follows:

$$\left\{ \begin{array}{rcl} \lambda \mathbf{u} & = & \mathbf{M}\mathbf{x}, \\ \lambda' \mathbf{u}' & = & \mathbf{M}'\mathbf{x}, \\ \lambda'' \mathbf{u}'' & = & \mathbf{M}''\mathbf{x}. \end{array} \right. \quad (6)$$

These can be rewritten in matrix form as

$$\begin{pmatrix} \mathbf{M} & \mathbf{u} & 0 & 0 \\ \mathbf{M}' & 0 & \mathbf{u}' & 0 \\ \mathbf{M}'' & 0 & 0 & \mathbf{u}'' \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ -\lambda \\ -\lambda' \\ -\lambda'' \end{pmatrix} = 0, \quad (7)$$

which is the basic reconstruction equation for a one-dimensional camera. The vector $(\mathbf{x}, -\lambda, -\lambda', -\lambda'')^T$ cannot be zero, so

$$\begin{vmatrix} \mathbf{M} & \mathbf{u} & 0 & 0 \\ \mathbf{M}' & 0 & \mathbf{u}' & 0 \\ \mathbf{M}'' & 0 & 0 & \mathbf{u}'' \end{vmatrix} = 0. \quad (8)$$

The expansion of this determinant produces a trilinear constraint of three views

$$\sum_{i,j,k=1}^{2} T_{ijk} u_i u'_j u''_k = 0, \quad (9)$$

or in short

$$\mathbf{T}_{2\times2\times2}\mathbf{u}\mathbf{u}'\mathbf{u}'' = 0,$$

where $\mathbf{T}_{2\times2\times2} = (T_{ijk})$ is a $2 \times 2 \times 2$ homogeneous tensor whose components T_{ijk} are 3×3 minors of the following 6×3 joint projection matrix:

$$\begin{pmatrix} \mathbf{M} \\ \mathbf{M}' \\ \mathbf{M}'' \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1' \\ 2' \\ 1'' \\ 2'' \end{pmatrix}. \quad (10)$$

The components of the tensor can be made explicit as

$$T_{ijk} = [\bar{i}\bar{j}'\bar{k}''], \text{ for } i, j', k'' = 1, 2. \quad (11)$$

where the bracket $[ij'k'']$ denotes the 3×3 minor of i-th,j'-th and k''-th row vector of the above joint projection matrix and bar "$^-$" in $\bar{i}$, $\bar{j}$ and $\bar{k}$ denotes the dualization

$$(1, 2) \mapsto (2, -1). \quad (12)$$

It can easily be seen that any constraint obtained by adding further views reduces to a trilinearity. This proves the uniqueness of the trilinear constraint. Moreover, the $2 \times 2 \times 2$ homogeneous tensor $\mathbf{T}_{2\times2\times2}$ has $7 = 2 \times 2 \times 2 - 1$ d.o.f., so it is a minimal parametrization of three views since three views have exactly

$$3 \times (2 \times 3 - 1) - (3 \times 3 - 1) = 7$$

d.o.f. up to a projective transformation in $\mathcal{P}^2$.

Each point correspondence over three views gives one linear constraint on the tensor components T_{ijk}. We can establish the following.

The tensor components T_{ijk} can be estimated linearly with at least 7 points in $\mathcal{P}^1$.

At this point, we have obtained a remarkable result that for a one-dimensional projective camera, the trilinear tensor encapsulates exactly the information needed for projective reconstruction in $\mathcal{P}^2$. Namely, it is the unique matching constraint, it minimally parametrizes the three views and it can be estimated linearly. Contrast this to the 2D projective camera case in which the multilinear constraints are algebraically redundant and the linear estimation is only an approximation based on over-parametrization.

B. Retrieving normal forms for projection matrices

The geometry of the three views is most conveniently, and completely represented by the projection matrices associated with each view. In the previous section, the trilinear tensor was expressed in terms of the projection matrices. Now we seek a map from the trilinear tensor representation back to the projection matrix representation of the three views.

Without loss of generality, we can always take the following normal forms for the three projection matrices

$$\begin{array}{rclcl} \mathbf{M} & = & \begin{pmatrix} \mathbf{I}_{2\times2} & \mathbf{0} \end{pmatrix}, & & \\ \mathbf{M}' & = & \begin{pmatrix} \mathbf{A}_{2\times2} & \mathbf{c} \end{pmatrix} & = & \begin{pmatrix} \mathbf{a} & \mathbf{b} & \mathbf{c} \end{pmatrix}, \\ \mathbf{M}'' & = & \begin{pmatrix} \mathbf{D}_{2\times2} & \mathbf{f} \end{pmatrix} & = & \begin{pmatrix} \mathbf{d} & \mathbf{e} & \mathbf{f} \end{pmatrix}. \end{array} \quad (13)$$

Actually, the set of projection matrices $\{\mathbf{M}, \mathbf{M}', \mathbf{M}''\}$ parametrized this way has 10 d.o.f.—still 3 more than the minimum of 7. Further constraints can be imposed. We can observe that any projective transformation in $\mathcal{P}^2$ of the form

$$\mathbf{H} = \begin{pmatrix} \mathbf{I}_{2\times2} & 0 \\ \mathbf{v}^T & 1 \end{pmatrix}$$

for an arbitrary 2-vector $\mathbf{v}$ leaves $\mathbf{M}$ invariant and transforms $\mathbf{M}'$ into

$$\tilde{\mathbf{M}}' = \mathbf{M}'\mathbf{H} = \begin{pmatrix} \mathbf{A} + \mathbf{c}\mathbf{v}^T & \mathbf{c} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{A}} & \mathbf{c} \end{pmatrix}.$$

As $\mathbf{c}$ cannot be a zero vector, it can be normalized such that $\mathbf{c}^T\mathbf{c} = 1$. If we further choose an arbitrary vector $\mathbf{v}$ to be $-\mathbf{A}^T\mathbf{c}$, then $\tilde{\mathbf{A}} = \mathbf{A} - \mathbf{c}\mathbf{c}^T\mathbf{A}$. It can now be easily verified that $\tilde{\mathbf{A}}^T\mathbf{c} = 0$. This amounts to saying that $\tilde{\mathbf{A}}$ in

$\tilde{\mathbf{M}}'$ can be taken to be a rank 1 matrix up to a projective transformation, *i.e.*

$$\tilde{\mathbf{A}} = \begin{pmatrix} a_1 & \rho a_1 \\ a_2 & \rho a_2 \end{pmatrix},$$

for a non-zero scalar ρ. The 2-vector $\mathbf{c}$ is then $(-a_2, a_1)^T$. Hence $\mathbf{M}'$ can be represented as

$$\mathbf{M}' = \begin{pmatrix} a_1 & \rho a_1 & -a_2 \\ a_2 & \rho a_2 & a_1 \end{pmatrix} \quad (14)$$

by two parameters, the ratio $a_1 : a_2$ and ρ. Therefore, a minimal 7 parameter representation for the set of projection matrices $\{\mathbf{M}, \mathbf{M}', \mathbf{M}''\}$ has been obtained.

With the projection matrices given by (13), the trilinear tensor (T_{ijk}) defined by (11) becomes

$$\lambda T_{ijk} = (-1)^{i+1}(d_{\bar{k}i}c_{\bar{j}} - a_{\bar{j}i}f_{\bar{k}}) \text{ for } i,j,k = 1,2, \quad (15)$$

where bar "$^-$" in $\bar{j}$ and $\bar{k}$ represents the dulization (12).

If we consider the tensor (T_{ijk}) as an 8-vector

$(t_1, \ldots, t_l, \ldots, t_8)^T$, for $l = 15-(4k+2j+i)$ and $i,j,k = 1,2$,

the eight homogeneous equations of (15) can be rearranged into 7 non-homogeneous ones by taking the ratios $t_l : t_8$ for $l = 1, \ldots, 7$. By separating the entries of $\mathbf{M}'$ from those of $\mathbf{M}''$, we obtain

$$\mathbf{G}_{7\times 6} \begin{pmatrix} \mathbf{d} \\ \mathbf{e} \\ \mathbf{f} \end{pmatrix} = 0, \quad (16)$$

where the matrix $\mathbf{G}_{7\times 6}$ is given by

$$\begin{pmatrix}
0 & t_8c_2 & -t_1c_1 & 0 & t_1b_1 & -t_8a_2 \\
0 & 0 & -t_2c_1 & t_8c_2 & t_2b_1 & -t_8b_2 \\
0 & -t_8c_1 & -t_3c_1 & 0 & t_3b_1 & t_8a_1 \\
0 & 0 & -t_4c_1 & -t_8c_1 & t_4b_1 & t_8b_1 \\
-t_8c_2 & 0 & -t_5c_1 & 0 & t_8a_2 + t_5b_1 & 0 \\
0 & 0 & -t_6c_1 - t_8c_2 & 0 & t_8b_2 + t_6b_1 & 0 \\
t_8c_1 & 0 & -t_7c_1 & 0 & -t_8a_1 + t_7b_1 & 0
\end{pmatrix}.$$

Since the parameter vector $(\mathbf{d}, \mathbf{e}, \mathbf{f})^T$ of $\mathbf{M}''$ cannot be zero, the 7×6 matrix in Equation (16) has at most rank 5. Thus all of its 6×6 minors must vanish. There are $2 = (6-5)\times(7-5)$ such minors which are algebraically independent, and each of them gives a quadratic polynomial in a_1, a_2 and ρ as follows:

$$\begin{cases} t_3\rho a_1 - t_4a_1 - t_1\rho a_2 + t_2a_2 = 0, \\ t_7\rho a_1 - t_8a_1 - t_5\rho a_2 + t_6a_2 = 0. \end{cases}$$

By eliminating ρ, we obtain a homogeneous quadratic equation in a_1 and a_2:

$$\alpha a_1^2 + \beta a_1a_2 + \gamma a_2^2 = 0, \quad (17)$$

where

$$\begin{aligned} \alpha &= t_3t_8 - t_4t_7, \\ \beta &= t_2t_7 + t_4t_5 - t_1t_8 - t_3t_6 \\ \gamma &= t_1t_6 - t_2t_5. \end{aligned}$$

This quadratic equation may be easily solved for a_1/a_2. Then ρ is given by the following linear equation for each of two solutions of a_1/a_2

$$(t_3a_1 - t_1a_2)\rho + (t_2a_2 - t_4a_1) = 0. \quad (18)$$

Thus, we obtain two possible solutions for the projection matrix $\mathbf{M}'$.

Finally, the 6-vector $(\mathbf{d}, \mathbf{e}, \mathbf{f})^T$ for the projection matrix $\mathbf{M}''$ is linearly solved from Equation (16) (for instance, using SVD) in terms of $\mathbf{M}'$.

C. 2D projective reconstruction—3D affine line direction reconstruction

With the complete determination of the projection matrices $\{\mathbf{M}, \mathbf{M}', \mathbf{M}''\}$ of the three views, the projective reconstruction of "*points*" in $\mathcal{P}^2$, which is equivalent to the affine reconstruction of "*line directions*" in $\mathbb{R}^3$, can be performed.

From the projection equation $\lambda\mathbf{u} = \mathbf{M}\mathbf{x}$, each point of a view $\mathbf{u} = (u_1, u_2)^T$ gives one homogeneous linear equation in the unknown point $\mathbf{x}$ in $\mathcal{P}^2$

$$(u_1\mathbf{m}_2^T - u_2\mathbf{m}_1^T)\mathbf{x} = 0,$$

where $\mathbf{m}_1^T$ and $\mathbf{m}_2^T$ are the first and second row vector of the matrix $\mathbf{M}$. With one point correspondence in three views $\mathbf{u} \leftrightarrow \mathbf{u}' \leftrightarrow \mathbf{u}''$, we have the following homogeneous linear equation system,

$$\begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix}_{3\times 3} \mathbf{x} = 0,$$

where $*$ designates a constant entry. This equation system can be easily solved for $\mathbf{x}$, either considered as a point in $\mathcal{P}^2$ or as an affine line direction in $\mathbb{R}^3$.

V. Uncalibrated translations

To recover the full affine structure of the lines, we still need to find the vectors $\mathbf{t}_{3\times 1}$ of the affine cameras defined in (2). These represent the image translations and magnification components of the camera. Recall that line correspondences from two views—now a 2D view instead of 1D view—do not impose any constraints on camera motion: The minimum number of views required is three. If the interpretation plane of an image line for a given view is defined as the plane going through the line and the projection center, the well-known geometric interpretation of the constraint available for each line correspondence across three views (*cf.* [3], [5]) is that the interpretation planes from different views must intersect in a common line in space.

If the equation of a line in the image is given by

$$\mathbf{l}^T\mathbf{u} = 0,$$

then substituting $\lambda\mathbf{u} = \mathbf{A}_{3\times4}\mathbf{x}$ into it produces the equation of the interpretation plane of $\mathbf{l}$ in space:

$$\mathbf{l}^T\mathbf{A}_{3\times4}\mathbf{x} = 0.$$

The plane is therefore given by the 4-vector $\mathbf{p}^T = \mathbf{l}^T\mathbf{A}_{3\times4}$, which can also be expressed as $\mathbf{p}^T = (\mathbf{n}_x, p)^T$ where $\mathbf{n}_x$ is the normal vector of the plane.

An image line of direction $\mathbf{n}_w$ can be written as $\mathbf{l} = (\mathbf{n}_w, l)^T$, with its interpretation plane being

$$\mathbf{p}^T = \mathbf{l}^T\mathbf{P} = (\mathbf{M}^T\mathbf{n}_w, \mathbf{l}^T\mathbf{t})^T. \tag{19}$$

The 2×3 submatrices $\mathbf{M}_{2\times3}$ representing uncalibrated camera orientations have already been obtained from the two-dimensional projective reconstruction. Now we proceed to recover the uncalibrated translations.

For each interpretation plane $(\mathbf{n}_x, p)^T$ of each image line, its direction component is completely determined by the previously computed $\{\mathbf{M}, \mathbf{M}', \mathbf{M}''\}$ as

$$\mathbf{n}_x = \mathbf{M}^T\mathbf{n}_w.$$

Only its fourth component $p = \mathbf{l}^T\mathbf{t}$ remains undetermined. This depends linearly on $\mathbf{t}$. Notice that as the direction vector can still be arbitrarily and individually rescaled, the interpretation plane should be properly written as

$$\mathbf{p}^T = (\lambda\mathbf{M}^T\mathbf{n}_w, \mu\mathbf{l}^T\mathbf{t})^T.$$

Hence the ratio λ/μ is significant, and this justifies the homogenization of the vector $\mathbf{t}$.

So far we have made explicit the equation of the interpretation planes of lines in terms of the image line and the projection matrix, the geometric constraint of line correspondences on the camera motion gives a 3×4 matrix whose rows are the three interpretation planes

$$\begin{pmatrix}\mathbf{p}^T\\ \mathbf{p}'^T\\ \mathbf{p}''^T\end{pmatrix} = \begin{pmatrix}\mathbf{n}_w^T\mathbf{M} & \mathbf{l}^T\mathbf{t}\\ \mathbf{n}_w'^T\mathbf{M}' & \mathbf{l}'^T\mathbf{t}'\\ \mathbf{n}_w''^T\mathbf{M}'' & \mathbf{l}''^T\mathbf{t}''\end{pmatrix}$$

which has rank at most two. Hence all of its 3×3 minors vanish. Only two of the total of four minors are algebraically independent, as they are connected by the quadratic identities [42].

The vanishing of any two such minors provides the two constraints on camera motion for a given line correspondence of three views. The minor formed by the first three columns contains only known quantities. It provides the constraint on the directions. It is easy to show that it is equivalent to the tensor by using suitable one-dimensional projective transformations.

By taking any two of the first three columns, say the first two, and the last column, we obtain the following vanishing determinant:

$$\begin{vmatrix} * & * & \mathbf{l}^T\mathbf{t}\\ * & * & \mathbf{l}'^T\mathbf{t}'\\ * & * & \mathbf{l}''^T\mathbf{t}''\end{vmatrix} = 0,$$

where the "$*$" designates a constant entry.

Expanding this minor by cofactors in the last column gives a homogeneous linear equation in $\mathbf{t}$, $\mathbf{t}'$ and $\mathbf{t}''$:

$$\begin{pmatrix}\times & \times & \times\end{pmatrix}_{1\times9}\begin{pmatrix}\mathbf{t}\\ \mathbf{t}'\\ \mathbf{t}''\end{pmatrix} = 0,$$

where the "$\times$" designates a constant 3-vector in a row.

Collecting all these vanishing minors together, we obtain

$$\begin{pmatrix}\times & \times & \times\\ \vdots & \vdots & \vdots\\ \times & \times & \times\end{pmatrix}_{n\times9}\begin{pmatrix}\mathbf{t}\\ \mathbf{t}'\\ \mathbf{t}''\end{pmatrix} = 0$$

for n line correspondences in three views.

At this stage, since the origin of the coordinate frame in space is not yet fixed, we may take $\mathbf{t} = (0,0,1)^T$ up to a scaling factor, say t_0, so the final homogeneous linear equations to solve for $(t_0, \mathbf{t}', \mathbf{t}'')^T$ is

$$\begin{pmatrix}* & \times & \times\\ \vdots & \vdots & \vdots\\ * & \times & \times\end{pmatrix}_{n\times7}\begin{pmatrix}t_0\\ \mathbf{t}'\\ \mathbf{t}''\end{pmatrix} = 0. \tag{20}$$

This system of homogeneous linear equations can be nicely solved by SVD factorisation. The least squares solution for $(t_0, \mathbf{t}', \mathbf{t}'')^T$ subject to $||(t_0, \mathbf{t}', \mathbf{t}'')^T|| = 1$ is the right singular vector corresponding to the smallest singular value.

VI. Affine shape

The projection matrices of the three views are now completely determined up to a common scaling factor. From now on, it is a relatively easy task to compute the affine shape. Two methods to obtain the shape will be described, one based on the projective representation of lines and another on the minimal representation of lines, inspired by [5].

A. Method 1: projective representation

A projective line in space can be defined either by a pencil of planes (a pencil of planes is defined by two projective planes) or by any two of its points.

The matrix

$$\mathbf{W}_P = \begin{pmatrix}\mathbf{p}^T\\ \mathbf{p}'^T\\ \mathbf{p}''^T\end{pmatrix}$$

should have rank 2, so its kernel must also have dimension 2. The range of $\mathbf{W}_P$ defines the pencil of planes and the null space defines the projective line in space.

Once again, using SVD to factorize $\mathbf{W}_P$ gives us everything we want. Let

$$\mathbf{W}_P = \mathbf{U}_P \Sigma_P \mathbf{V}_P^T$$

be the SVD of $\mathbf{W}_P$ with ordered singular values. Two points of the line might be taken to be $\mathbf{v}_3$ and $\mathbf{v}_4$, so the line is given by

$$\lambda \mathbf{v}_3 + \mu \mathbf{v}_4 \quad \text{for} \quad \lambda, \mu \in \mathbb{R}. \tag{21}$$

One advantage of this method is that, using subset selection [43], near singular views can be detected and discarded.

B. Method 2: Minimal representation

As a space line has 4 d.o.f., it can be minimally represented by four parameters. One such possibility is suggested by [5] which uses a 4-vector $\mathbf{l}^T = (a, b, x_0, y_0)^T$ such that the line is defined as the intersection of two planes $(1, 0, -a, -x_0)^T$ and $(0, 1, -b, -y_0)^T$ with equations:

$$\begin{cases} x = az + x_0 \\ y = bz + y_0. \end{cases}$$

Geometrically this minimal representation gives a 3D line with direction $(a, b, 1)^T$ and passing through the point $(x_0, y_0, 0)^T$. This representation excludes, therefore, the lines of direction $(a, b, 0)^T$, parallel to the xy plane. Two other representations are needed, each excluding either the directions $(0, b, c)^T$ or $(a, 0, c)^T$. These 3 representations together completely describe any line in space.

In our case, we have no problem in automatically selecting one of the three representations, as the directions of lines have been obtained in the first step of factorisation, allowing us to switch to one of the three representations. There remain only two unknown parameters x_0 and y_0 for each line.

To get a solution for x_0 and y_0, as the two planes $(1, 0, -a, -x_0)^T$ and $(0, 1, -b, -y_0)^T$ defining the line belong to the pencil of planes defined by $\mathbf{W}_P$, we can still stack these two planes on the top of $\mathbf{W}_P$ to get the matrix $\mathbf{W}'_P$:

$$\mathbf{W}'_P = \begin{pmatrix} * & * & * & x_0 \\ * & * & * & y_0 \\ & \mathbf{W}_P & & \end{pmatrix} = \begin{pmatrix} * & * & * & x_0 \\ * & * & * & y_0 \\ * & * & * & * \\ \vdots & \vdots & \vdots & \vdots \\ * & * & * & * \end{pmatrix}.$$

Since this matrix still has rank 2, all its 3×3 minors vanish. Each minor involving x_0 and y_0 gives a linear equation in x_0 and y_0. With n views, a linear equation system is obtained

$$\mathbf{A}_{n \times 2} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \mathbf{b}. \tag{22}$$

This can be nicely solved using least squares for each line.

VII. Affine-structure-from-lines theorem

Summarizing the results obtained above, we have established the following.

For the recovery of affine shape and affine motion from line correspondences with an uncalibrated affine camera, the minimum number of views needed is three and the minimum number of lines required is seven for a linear solution. There are always two solutions for the recovered affine structure.

This result can be compared with that of Koenderink and Van Doorn [14] for affine structure with a minimum of two views and five points.

We should also note the difference with the well-known results established for both calibrated and uncalibrated projective cameras [3], [4], [5], [38]: A minimum of 13 lines in three views is required to have a linear solution. It is important to note that with the affine camera and the method presented in this paper, the number of line correspondences for achieving a linear solution is reduced from 13 to 7, which is of great practical importance.

VIII. Outline of the 7-line × 3-view algorithm

The linear algorithm to recover 3D affine shape/motion from at least 7 line correspondences over three views with uncalibrated affine cameras may be outlined as follows:

1. If an image line segment is represented by its endpoints $\mathbf{w}_1 = (u_1, v_1)^T$ and $\mathbf{w}_2 = (u_2, v_2)^T$, compute the direction vector of the line $\mathbf{n}_w = \mathbf{u}_2 - \mathbf{u}_1$. View this as the homogeneous coordinates of a point in $\mathcal{P}^1$.
2. Compute the tensor components (T_{ijk}) defined by Equation (9) linearly with at least 7 lines in 3 views.
3. Retrieve the projection matrices $\{\mathbf{M}, \mathbf{M}', \mathbf{M}''\}$ of the one-dimensional camera from the estimated tensor using Equations (17), (18) and (16). There are always two solutions.
4. Perform 2D projective reconstruction using equation (7) which recovers the directions of the affine lines in space and the uncalibrated rotations of the camera motion.
5. Solve the uncalibrated translation vector $(\mathbf{t}, \mathbf{t}', \mathbf{t}'')^T$ using Equation (20) by linear least squares.
6. Compute the final affine lines in space using Equation (21) or (22).

IX. Line-based factorisation method from an image stream

The linear affine reconstruction algorithm described above deals with redundant lines, but is limited to three views. In this section we discuss redundant views, extending the algorithm from the minimum of three to any number $N > 3$ of views.

In the past few years, a family of algorithms for structure from motion using highly redundant image sequences called *factorisation methods* have been extensively studied

[18], [19], [20], [21], [22] for point correspondences for affine cameras. Algorithms of this family directly decompose the feature points of the image stream into object shape and camera motion. More recently, a factorisation based algorithm has been proposed by Triggs and Sturm [35], [36] for 3D projective reconstruction. We will accomodate our line-based algorithm to this projective factorisation schema to handle redundant views.

A. 2D projective reconstruction by rescaling

According to [35], [36], 3D projective reconstruction is equivalent to the rescaling of the 2D image points. We have already proven that recovering the directions of affine lines in space is equivalent to 2D projective reconstruction from one-dimensional projective images. Therefore, a reconstruction of the line directions in 3D can be obtained by rescaling the direction vectors, viewed as points of $\mathcal{P}^1$.

For each 1D image point in three views (*cf.* Equation (6)), the scale factors λ, λ' and λ''—taken individually—are arbitrary. However, taken as a whole $(\lambda, \lambda', \lambda'')^T$, they encode the projective structure of the points $\mathbf{x}$ in $\mathcal{P}^2$.

One way to recover the scale factors $(\lambda, \lambda', \lambda'')^T$ is to use the basic reconstruction equation (7) directly or alternatively to observe the following matrix identity:

$$\begin{pmatrix} \mathbf{M} & \lambda \mathbf{u} \\ \mathbf{M}' & \lambda' \mathbf{u}' \\ \mathbf{M}'' & \lambda'' \mathbf{u}'' \end{pmatrix} = \begin{pmatrix} \mathbf{M} \\ \mathbf{M}' \\ \mathbf{M}'' \end{pmatrix} \begin{pmatrix} \mathbf{I}_{3\times 3} & \mathbf{x} \end{pmatrix}.$$

The rank of the left matrix is therefore at most 3. All 4×4 minors vanish, and three $(3 = (4-3) \times (6-3))$ of them are algebraically independent, for instance,

$$\begin{vmatrix} \mathbf{M} & \lambda \mathbf{u} \\ \mathbf{M}' & \lambda' \mathbf{u}' \end{vmatrix} = 0,$$
$$\begin{vmatrix} \mathbf{M}' & \lambda' \mathbf{u}' \\ \mathbf{M}'' & \lambda'' \mathbf{u}'' \end{vmatrix} = 0,$$
$$\begin{vmatrix} \mathbf{M} & \lambda \mathbf{u} \\ \mathbf{m}'_1 & \lambda' u' \\ \mathbf{m}''_1 & \lambda'' u'' \end{vmatrix} = 0.$$

Each of them can be expanded by cofactors in the last column to obtain a linear homogeneous equation in $\lambda, \lambda', \lambda''$. Therefore $(\lambda, \lambda', \lambda'')^T$ can be solved linearly using

$$\begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} \begin{pmatrix} \lambda \\ \lambda' \\ \lambda'' \end{pmatrix} = 0, \qquad (23)$$

where $*$ designate a known constant entry in the matrix.

For each triplet of views, the image points can be consistently rescaled according to Equation (23). For the case of $n > 3$ views, we can take appropriate triplets among n views such that each view is contained in at least two triplets. Then, the rescaling equations of all triplets of views for any given point can be chained together over n views to give a consistent $(\lambda, \lambda', \ldots, \lambda^{(n)})^T$.

B. Direction factorisation–step 1

Suppose we are given m line correspondences in n views. The view number is indexed by a superscript and the line number by a subscript. We can now create the $2n \times m$ measurement matrix $\mathbf{W}_D$ of all lines in all views by stacking all the direction vectors $\mathbf{d}_{w_i}^{(j)}$ properly rescaled by $\lambda_i^{(j)}$ of m lines in n views as follows:

$$\mathbf{W}_D = \begin{pmatrix} \lambda_1 \mathbf{d}_{w_1} & \lambda_2 \mathbf{d}_{w_2} & \cdots & \lambda_m \mathbf{d}_{w_m} \\ \lambda'_1 \mathbf{d}'_{w_1} & \lambda'_2 \mathbf{d}'_{w_2} & \cdots & \lambda'_m \mathbf{d}'_{w_m} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1^{(n)} \mathbf{d}_{w_1}^{(n)} & \lambda_2^{(n)} \mathbf{d}_{w_2}^{(n)} & \cdots & \lambda_n^{(n)} \mathbf{d}_{w_m}^{(n)} \end{pmatrix}.$$

Since the following matrix equation holds for the measurement matrix $\mathbf{W}_D$:

$$\mathbf{W}_D = \mathbf{M}_D \mathbf{D}_D = \begin{pmatrix} \mathbf{M} \\ \mathbf{M}' \\ \vdots \\ \mathbf{M}^{(n)} \end{pmatrix} \begin{pmatrix} \mathbf{d}_{x_1} & \mathbf{d}_{x_2} & \cdots & \mathbf{d}_{x_m} \end{pmatrix},$$

the rank of $\mathbf{W}_D$ is at most of three. The factorisation method can then be applied to $\mathbf{W}_D$.

Let

$$\mathbf{W}_D = \mathbf{U}_D \Sigma_D \mathbf{V}_D^T$$

be the SVD factorisation (*cf.* [43], [44]) of $\mathbf{W}_D$. The 3×3 diagonal matrix Σ_{D3} is obtained by keeping the first three singular values (assuming that singular values are ordered) of Σ and $\mathbf{U}_{D3}$ ($\mathbf{V}_{D3}$) are the first 3 columns (rows) of $\mathbf{U}$ ($\mathbf{V}$).

Then, the product $\mathbf{U}_{D3} \Sigma_{D3} \mathbf{V}_{D3}^T$ gives the best rank 3 approximation to $\mathbf{W}_D$.

One possible solution for $\hat{\mathbf{M}}$ and $\hat{\mathbf{D}}$ may be taken to be

$$\hat{\mathbf{M}} = \mathbf{U}_{D3} \Sigma_{D3}^{1/2} \quad \text{and} \quad \hat{\mathbf{D}} = \Sigma_{D3}^{1/2} \mathbf{V}_{D3}.$$

For any nonsingular 3×3 matrix $\mathbf{A}_{3\times 3}$—either considered as a projective transformation in $\mathcal{P}^2$ or as an affine transformation in $\mathbb{R}^3$, $\hat{\mathbf{M}}' = \hat{\mathbf{M}} \mathbf{A}_{3\times 3}$ and $\hat{\mathbf{D}}' = \mathbf{A}_{3\times 3}^{-1} \hat{\mathbf{D}}$ are also valid solutions, as we have

$$\hat{\mathbf{M}} \mathbf{A} \mathbf{A}^{-1} \hat{\mathbf{D}} = \hat{\mathbf{M}}' \hat{\mathbf{D}}' = \hat{\mathbf{M}} \hat{\mathbf{D}}.$$

This means that the recovered direction matrix $\hat{\mathbf{D}}$ and the rotation matrix $\hat{\mathbf{M}}$ are only defined up to an affine transformation.

C. Translation factorisation—Step 2

We can stack all of the interpretation planes from different views of a given line to form the following $n \times 4$ measurement matrix of planes:

$$\mathbf{W}_P = \begin{pmatrix} * & * & * & \mathbf{l}^T \mathbf{t} \\ * & * & * & \mathbf{l}'^T \mathbf{t}' \\ \vdots & \vdots & \vdots & \vdots \\ * & * & * & \mathbf{l}^{(n)^T} \mathbf{t}^{(n)} \end{pmatrix}.$$

This matrix $\mathbf{W}_P$ geometrically represents a pencil of planes, so it still has rank at most 2. For any three rows i, j and k of $\mathbf{W}_P$, taking any minor involving the $\mathbf{t}^{(i)}$, we obtain

$$\begin{vmatrix} * & * & \mathbf{l}^{(i)^T}\mathbf{t}^{(i)} \\ * & * & \mathbf{l}^{(j)^T}\mathbf{t}^{(j)} \\ * & * & \mathbf{l}^{(k)^T}\mathbf{t}^{(k)} \end{vmatrix} = 0.$$

Expanding this minor by cofactors in the last column gives a homogeneous linear equation in $\mathbf{t}^{(i)}$, $\mathbf{t}^{(j)}$ and $\mathbf{t}^{(k)}$:

$$\begin{pmatrix} \times & \times & \times \end{pmatrix} \begin{pmatrix} \mathbf{t}^{(i)} \\ \mathbf{t}^{(j)} \\ \mathbf{t}^{(k)} \end{pmatrix} = 0,$$

where each "$\times$" designates a constant 3-vector in a row.

Collecting all these minors together, we obtain

$$\begin{pmatrix} \times & \times & \times & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \times & \times & \times & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & \times & \times & \times \end{pmatrix} \begin{pmatrix} \mathbf{t} \\ \mathbf{t}' \\ \vdots \\ \mathbf{t}^{(n)} \end{pmatrix} = 0.$$

We may take $\mathbf{t} = (0, 0, 1)^T$ up to a scaling factor, say t_0, so the final homogeneous linear equations to solve for $(t_0, \mathbf{t}', \ldots, \mathbf{t}^{(n)})^T$ are

$$\mathbf{W}_T \begin{pmatrix} t_0 \\ \mathbf{t}' \\ \vdots \\ \mathbf{t}^{(n)} \end{pmatrix} =$$

$$\begin{pmatrix} * & \times & \times & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \times & \times & \times & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & \times & \times & \times \end{pmatrix} \begin{pmatrix} t_0 \\ \mathbf{t}' \\ \vdots \\ \mathbf{t}^{(n)} \end{pmatrix} = 0.$$

Once again, this system of equations can be nicely solved by SVD factorisation of $\mathbf{W}_T$. The least squares solution for $(t_0, \mathbf{t}', \ldots, \mathbf{t}^{(n)})^T$, subject to $||(t_0, \mathbf{t}', \mathbf{t}'', \cdots)^T|| = 1$, is the singular vector corresponding to the smallest singular value of $\mathbf{W}_T$.

Note that the efficiency of the computation can be further improved if the block diagonal structure of $\mathbf{W}_T$ is exploited.

D. *Shape factorisation—Step 3*

The shape reconstruction method developed for three views extends directly to more than 3 views. Given n views, for each line across n views, we just augment the matrix $\mathbf{W}_p$ from a 3×4 to $n \times 4$ matrix, then apply exactly the same method.

X. Outline of the line-based factorisation algorithm

The line-based factorisation algorithm can be outlined as follows:

1. For triplets of views, compute the tensor (T_{ijk}) associated with each triplet, then rescale the directions of lines of the triplet using Equation (23).
2. Chain together all the rescaling factors $(\lambda, \lambda', \ldots, \lambda^{(n)})^T$ for each line across the sequence.
3. Factorise the rescaled measurement matrix of directions
$$\mathbf{W}_D = \mathbf{U}\Sigma\mathbf{V}^T$$
to get the uncalibrated rotations and the directions of the affine lines
$$\begin{aligned} \hat{\mathbf{M}} &= (\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3) diag(\sigma_1^{1/2}, \sigma_2^{1/2}, \sigma_3^{1/2}), \\ \hat{\mathbf{D}} &= diag(\sigma_1^{1/2}, \sigma_2^{1/2}, \sigma_3^{1/2})(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3). \end{aligned}$$
4. Factorise the measurement matrix using the constraints on the motion
$$\mathbf{W}_T = \mathbf{U}\Sigma\mathbf{V}^T$$
to get the uncalibrated translation vector
$$(\hat{t}_0, \hat{\mathbf{t}}, \ldots, \hat{\mathbf{t}}^{(n)})^T = \mathbf{v}_{3(n-1)+1}.$$
5. Factorise the measurement matrix of the interpretation planes for each line correspondence over all views
$$\mathbf{W}_P = \mathbf{U}\Sigma\mathbf{V}^T$$
to get two points of the line
$$\hat{\mathbf{x}}_1 = \mathbf{v}_3 \quad \text{and} \quad \hat{\mathbf{x}}_2 = \mathbf{v}_4.$$

XI. Euclidean structure from the calibrated affine camera

So far we have worked with an uncalibrated affine camera, the recovered shape and motion are defined up to an affine transformation in space. If the cameras are calibrated, then the affine structure can be converted into a Euclidean one up to an unknown global scale factor.

Following the decomposition of the submatrix $\mathbf{M}_{2\times 3}$ of the affine camera $\mathbf{A}_{3\times 4}$ as $\mathbf{M} = \mathbf{KR}$ introduced in [22], the metric information from the calibrated affine camera is completely contained in the affine intrinsic parameters $\mathbf{KK}^T$. Each view with the associated uncalibrated rotation matrix $\mathbf{M} = \begin{pmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \end{pmatrix}$ is subject to

$$\begin{pmatrix} \mathbf{m}_1^T\mathbf{X}\mathbf{X}^T\mathbf{m}_1^T & \mathbf{m}_1^T\mathbf{X}\mathbf{X}^T\mathbf{m}_2^T \\ \mathbf{m}_1^T\mathbf{X}\mathbf{X}^T\mathbf{m}_2^T & \mathbf{m}_2^T\mathbf{X}\mathbf{X}^T\mathbf{m}_2^T \end{pmatrix} = \mathbf{KK}^T$$

for the unknown affine transformation $\mathbf{X}$ which upgrades the affine structure to a Euclidiean one. A linear solution may be expected as soon as we have three views if we solve for the entries of $\mathbf{X}\mathbf{X}^T$. However it may happen that the linear estimate of $\mathbf{X}\mathbf{X}^T$ is not positive-definite due to noise. An alternative non-linear solution using Cholesky parametrization that ensures the positive-definiteness can be found in [22].

Once we obtain the appropriate $\hat{\mathbf{X}}$, then $\hat{\mathbf{M}}\hat{\mathbf{X}}$ and $\hat{\mathbf{X}}^{-1}\hat{\mathbf{D}}$ carry the rotations of the camera and the directions of lines. The remaining steps are the same as the uncalibrated affine camera case.

If we take the weak perspective as a particular affine camera model, with only the aspect ratio of the camera, Euclidean structure is obtained this way.

XII. Experimental results

A. Simulation setup

We first use simulated images to validate the theoretical development of the algorithm. To preserve realism, the simulation is set up as follows. First, a real camera is calibrated by placing a known object of about 50 cm^3 in front of the camera. The camera is moved around the object through different positions. A calibration procedure gives the projection matrices at different positions, and these projection matrices are rounded to affine projection matrices. Three different positions which cover roughly 45^o of the field of view are selected. A set of 3D line segments within a cube of 30 cm^3 is generated synthetically and projected onto the different image planes by the affine projection matrices. All simulated images are of size 512×512. Both 3D and 2D line segments are represented by their endpoints.

The noise-free line segments are then perturbed as follows. To take advantage of the relatively higher accuracy of line position obtained by the line fitting process in practice, each 2D line segment is first re-sampled into a list of evenly spaced points of the line segment. The position of each point is perturbed by varying levels of noise of uniform distribution. The final perturbed line is obtained by a least squares fit to the perturbed point data.

Reconstruction is performed with 21 line segments and two different re-sample rates. The average residual error is defined to be the average distance of the midpoint of the image line segment to the reprojected line in the image plane from the 3D reconstructed line. In Table I, the average residual errors of reconstruction are given with various noise levels. The number of points used to fit the line is the length of the line segment in pixels, this re-sample rate corresponds roughly to the digitization process. Table II shows the results with the number of points used to fit the line equal to only one fourth the length of the line segment. We can notice that the degradation with the increasing noise level is very graceful and the reconstruction results remain acceptable with up to ± 5.5 pixel noise. These good results show that the reconstruction algorithm is numerically stable. While comparing Table I and II, it shows that higher re-sample rate gives better results, this confirms the importance of the line fitting procedure—the key advantage of line features over point features.

Another influential factor for the stability of the algorithm is the number of lines used. Table III confirms that the more lines used, the better the results obtained. In this test, the pixel error is set to ± 1.5.

Lines #	8	13	17	21
Average residual error	1.9	1.6	0.59	0.26

TABLE III

Average residual errors of reconstruction with $\pm$ 1.5 pixel noise and various number of lines.

B. The experiment with real images

A Fujinon/Photometrics CCD camera is used to aquire a sequence of images of a box of size $12 \times 12 \times 12.65 cm$. The image resolution is 576×384. Three of the frames in the sequence used by the experiments are shown in Figure 1.

A Canny-like edge detector is first applied to each image. The contour points are then linked and fitted to line segments by least squares. The line correspondences across three views are selected by hand. There are a total of 46 lines selected, as shown in Figure 2.

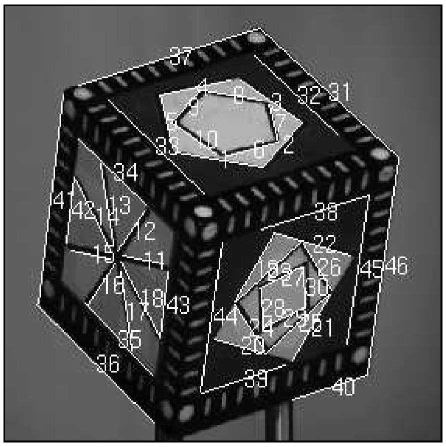

Fig. 2. Line segments selected across the image sequence.

The reconstruction algorithm generates infinite 3D lines, each defined by two arbitrary points on it. 3D line *segments* are obtained as follows. We reproject 3D lines into one image plane. In the image plane selected, the corresponding original image line segments are orthogonally projected onto the reprojected lines to obtain the reprojected line segments. Finally by back-projecting the reprojected line segments to space, we obtain the 3D line segments, each defined by its two endpoints.

Excellent reconstruction results are obtained. An average residual error of one tenth of a pixel is achieved. Figure 3 shows two views of the reconstructed 3D line segments. We notice that the affine structure of the box is almost perfectly recovered.

Table IV shows the influence of the number of line segments

Noise	±0.5	±1.5	±2.5	±3.5	±4.5	±5.5
Average residual error	0.045	0.061	0.10	0.15	0.20	0.25

TABLE I

Average residual errors with various noise levels for the reconstruction with 21 lines over three views. The number of points to fit the line is the length of the line segment in pixels.

Noise	±0.5	±1.5	±2.5	±3.5	±4.5	±5.5
Average residual error	0.077	0.26	0.31	0.44	0.65	1.1

TABLE II

Average residual errors of reconstruction with various noise levels. The number of points to fit the line segment is one fourth the length of the line segment.

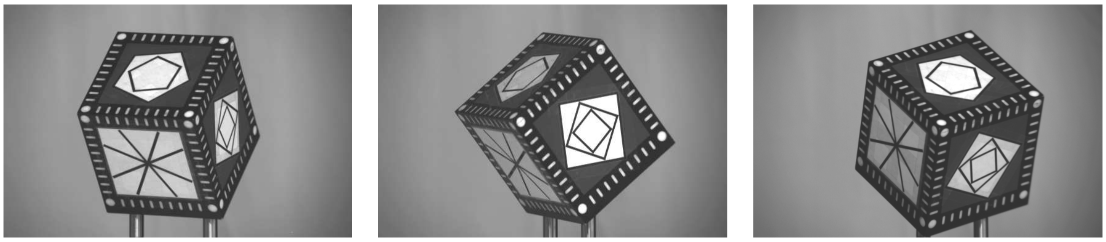

Fig. 1. Three original images of the box used for the experiments.

used by the algorithm. The reconstruction results degrade gracefully with decreasing number of lines.

Lines #	10	20	30	46
Average residual error	1.3	0.88	0.28	0.12

TABLE IV

Table of residual errors of reconstruction with different number of line segments.

Table V shows the influence of the distribution of line segments in space. For instance, one degenerate case for structure from motion is that when all line segments in space lie on the same plane. Actually, in our images, line segments lie on three different planes—pentagon face, star shape face and rectangle face—of the box. We also performed experiments with line segments lying on only two planes. Table V shows the results with various different two-plane configurations. Compared with the three-plane configuration, the reconstruction algorithm does almost equally well.

To illustrate the effect of using affine camera model as an approximation to the perspective camera, we used a bigger cube of size $30 \times 30 \times 30cm$, which is two and a half times the size of the first smaller cube. The affine approximation to the perspective camera is becoming less accurate than it was with the smaller cube. A sequence of images of this cube is aquired in almost the same conditions as for the smaller cube. The perspective effect of the big cube is slightly more pronounced as shown in Figure 4. The configuration of line segments is preserved. A total of 39 line segments of three views is used to perform the reconstruction. Figure 5 illustrates two reprojected views of the reconstructed 3D line segments. Compared with Figure 3, the reconstruction is slightly degraded: in the top view of Figure 5, we notice that one segment falls a little apart from the pentagon face of the cube. Globally, the degradation is quite graceful as the average residual error is only 0.3 pixels, compared with 0.12 pixels for the smaller cube.

The affine structures obtained can be converted to Euclidean ones (up to a global scaling factor) as soon as we know the aspect ratio [22], which is actually 1 for the camera used. Figure 6 shows the rectified affine shape illustrated in Figure 3. The two sides of the box are accurately orthogonal to each other.

XIII. Discussion

A linear line-based structure from motion algorithm for uncalibrated affine cameras has been presented. The algorithm requires a minimum of seven line correspondences over three views. It has also been proven that seven lines over three views are the strict minimum data needed to recover affine structure with uncalibrated affine cameras. In other words, in contrast to projective cameras, the linear algorithm is not based on the over-parametrization. This gives the algorithm intrinsic stability. The previous results of Koenderink and Van Doorn [14] on affine structure from motion using point correspondences are therefore extended to line correspondences. To handle the case of redundant views, a factorisation method was also developed. The experimental results based on real and simulated image sequences demonstrate the accuracy and the stability of the algorithms.

As the algorithms presented in this paper are developed within the same framework as suggested in [22] for points, it is straightforward to integrate both points and lines into the same framework.

Line configuration	star+rect.+pent.	star+rect.	pent.+rect.	star+pent.
Average residual error	0.12	0.078	0.14	0.28

TABLE V

Table of residual errors of reconstruction with different data.

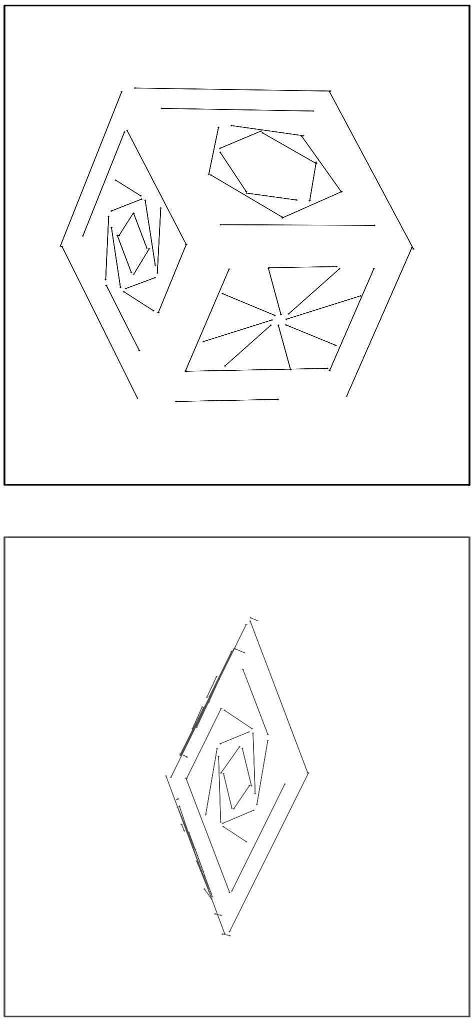

Fig. 3. Reconstructed 3D line segments: a general view and a top view.

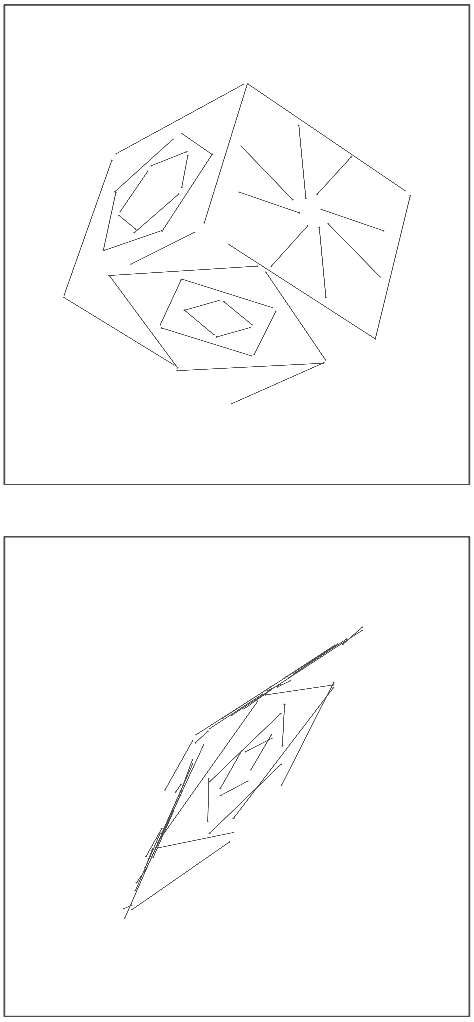

Fig. 5. Two views of the reconstructed line segments for the big box: a general view and a top view.

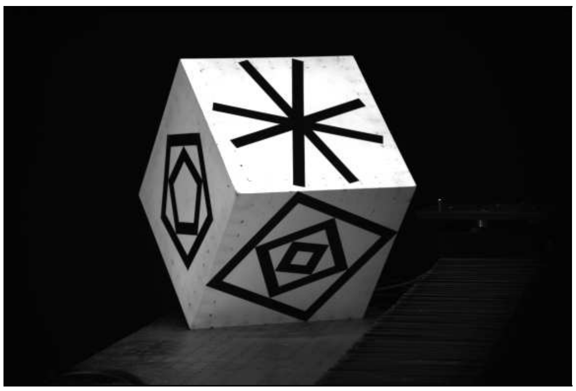

Fig. 4. One original image of the big cube image sequence.

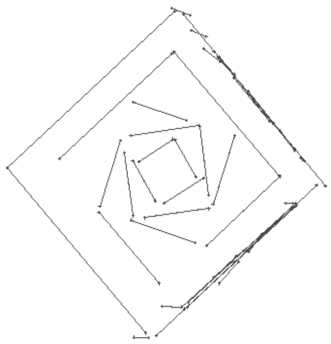

Fig. 6. A side view of the Euclidean shape obtained by using the known aspect ratio of the camera.

Acknowledgement

This work was supported by CNRS and the French Ministère de l'Education which is gratefully acknowledged. I would like to thank D. Morris, N. Chiba and B. Triggs for their help during the development of this work.

References

[1] M. Dhome, M. Richetin, J.T. Lapresté, and G. Rives, "Determination of the attitude of 3D objects from a single perspective view", IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 12, pp. 1265–1278, December 1989.

[2] N. Ayache, *Stereovision and Sensor Fusion*, The MIT Press, Cambridge, MA, USA, 1990.

[3] Y. Liu and T.S. Huang, "Estimation of rigid body motion using straight line correspondences", *Computer Vision, Graphics and Image Processing*, vol. 43, no. 1, pp. 37–52, July 1988.

[4] Y. Liu and T.S. Huang, "A linear algorithm for motion estimation using straight line correspondences", *Computer Vision, Graphics and Image Processing*, vol. 44, no. 1, pp. 35–57, October 1988.

[5] O.D. Faugeras, F. Lustman, and G. Toscani, "Motion and structure from point and line matches", in *Proceedings of the 1st International Conference on Computer Vision, London, England*. June 1987, pp. 25–34, IEEE Computer Society Press.

[6] Z. Zhang and O.D. Faugeras, *3D Dynamique Scene Analysis, a Stereo Based Approach*, vol. 27 of *Springer Series in Information Sciences*, Springer-Verlag, 1992.

[7] J. Weng, T.S. Huang, and N. Ahuja, "Motion and structure from line correspondences: Closed-form solution, uniqueness, and optimization", IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 3, pp. 318–336, 1992.

[8] T.Q. Phong, R. Horaud, A. Yassine, and D.T. Pham, "Optimal estimation of object pose from a single perspective view", in *Proceedings of the 4th International Conference on Computer Vision, Berlin, Germany*, May 1993.

[9] T. Viéville, Q.T. Luong, and O.D. Faugeras, "Motion of points and lines in the uncalibrated case", *International Journal of Computer Vision*, vol. 17, no. 1, 1995.

[10] M. Spetsakis and J. Aloimonos, "A unified theory of structure from motion", in *Proceedings of DARPA Image Understanding Workshop*, 1990, pp. 271–283.

[11] H.C. Longuet-Higgins, "A computer program for reconstructing a scene from two projections", *Nature*, vol. 293, pp. 133–135, September 1981.

[12] R.I. Hartley, "Projective reconstruction from line correspondences", in *Proceedings of the Conference on Computer Vision and Pattern Recognition, Seattle, Washington, USA*, 1994, pp. 903–907.

[13] S. Ullman, *The Interpretation of Visual Motion*, The MIT Press, Cambridge, MA, USA, 1979.

[14] J. Koenderink and A. van Doorn, "Affine structure from motion", *Journal of the Optical Society of America A*, vol. 8, no. 2, pp. 377–385, 1991.

[15] L. Quan and R. Mohr, "Affine shape representation from motion through reference points", *Journal of Mathematical Imaging and Vision*, vol. 1, pp. 145–151, 1992, also in IEEE Workshop on Visual Motion, New Jersey, pages 249–254, 1991.

[16] C.H. Lee and T. Huang, "Finding point correspondences and determining motion of a rigid object from two weak perspective views", *Computer Vision, Graphics and Image Processing*, vol. 52, pp. 309–327, 1990.

[17] P.F. McLauchlan, I.D. Reid, and D.W. Murray, "Recursive affine structure and motion from image sequences", in *Proceedings of the 3rd European Conference on Computer Vision, Stockholm, Sweden*, J.O. Eklundh, Ed. 1994, pp. 217–224, Springer-Verlag.

[18] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method", *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, November 1992.

[19] D. Weinshall and C. Tomasi, "Linear and incremental acquisition of invariant shape models from image sequences", in *Proceedings of the 4th International Conference on Computer Vision, Berlin, Germany*. May 1993, pp. 675–682, IEEE.

[20] L.S. Shapiro, A. Zisserman, and M. Brady, "3D motion recovery via affine epipolar geometry", *International Journal of Computer Vision*, vol. 16, no. 2, pp. 147–182, 1995.

[21] C.J. Poelman and T. Kanade, "A paraperspective factorization method for shape and motion recovery", in *Proceedings of the 3rd European Conference on Computer Vision, Stockholm, Sweden*, J.O. Eklundh, Ed. May 1994, pp. 97–108, Springer-Verlag.

[22] L. Quan, "Self-calibration of an affine camera from multiple views", *International Journal of Computer Vision*, vol. 19, no. 1, pp. 93–105, May 1996.

[23] R. Horaud, S. Christy, and F. Dornaika, "Object pose: the link between weak perspective, para perspective, and full perspective", Tech. Rep., Inria, September 1994.

[24] P.H.S. Torr, *Motion Segmentation and Outlier Detection*, PhD thesis, University of Oxford, England, Department of Engineering Science, Parks Road, Oxford, 1995.

[25] S. Ullman and R. Basri, "Recognition by linear combinations of models", IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 10, pp. 992–1006, 1991.

[26] L. Quan and T. Kanade, "A factorization method for affine structure from line correspondences", in *Proceedings of the Conference on Computer Vision and Pattern Recognition, San Francisco, California, USA*, 1996, pp. 803–808.

[27] J.L. Mundy and A. Zisserman, Eds., *Geometric Invariance in Computer Vision*, The MIT Press, Cambridge, MA, USA, 1992.

[28] J.Y. Aloimonos, "Perspective approximations", *Image and Vision Computing*, vol. 8, no. 3, pp. 179–192, August 1990.

[29] O. Faugeras, "What can be seen in three dimensions with an uncalibrated stereo rig?", in *Proceedings of the 2nd European Conference on Computer Vision, Santa Margherita Ligure, Italy*, G. Sandini, Ed. May 1992, pp. 563–578, Springer-Verlag.

[30] R.I. Hartley, R. Gupta, and T. Chang, "Stereo from uncalibrated cameras", in *Proceedings of the Conference on Computer Vision and Pattern Recognition, Urbana-Champaign, Illinois, USA*, 1992, pp. 761–764.

[31] Q.T. Luong, *Matrice fondamentale et autocalibration en vision par ordinateur*, Thèse de doctorat, Université de Paris-Sud, Orsay, France, December 1992.

[32] Q.T. Luong and T. Vieville, "Canonic representations for the geometries of multiple projective views", Tech. Rep., University of California, Berkeley, EECS, Cory Hall 211-215, University of California, Berkeley, CA 94720, October 1993.

[33] R. Mohr, L. Quan, and F. Veillon, "Relative 3D reconstruction using multiple uncalibrated images", *The International Journal of Robotics Research*, vol. 14, no. 6, pp. 619–632, 1995.

[34] L. Quan, "Invariants of six points and projective reconstruction from three uncalibrated images", IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 34–46, January 1995.

[35] B. Triggs, "Matching constraints and the joint image", in *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, E. Grimson, Ed. IEEE, June 1995, pp. 338–343, IEEE Computer Society Press.

[36] P. Sturm and B. Triggs, "A factorization based algorithm for multi-image projective structure and motion", in *Proceedings of the 4th European Conference on Computer Vision, Cambridge, England*, B. Buxton and R. Cipolla, Eds. April 1996, vol. 1065 of *Lecture Notes in Computer Science*, pp. 709–720, Springer-Verlag.

[37] O. Faugeras and B. Mourrain, "On the geometry and algebra of the point and line correspondences between n images", in *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, June 1995, pp. 951–956.

[38] R. Hartley, "Lines and points in three views - an integrated approach", Tech. Rep., G.E. CRD, 1994.

[39] A. Shashua, "Algebraic functions for recognition", IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 779–789, August 1995.

[40] M. Spetsakis and J. Aloimonos, "Structure from motion using line correspondences", *International Journal of Computer Vision*, vol. 4, pp. 171–183, 1990.

[41] P.A. Beardsley, I.D. Reid, A. Zisserman, and D.W. Murray, "Active visual navigation using non-metric structure", in *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, E. Grimson, Ed. June 1995, pp. 58–64, IEEE Computer Society Press.

[42] B. Sturmfels, *Algorithms in Invariant Theory*, Springer-Verlag, 1993.

[43] G.H. Golub and C.F. van Loan, *Matrix Computation*, The Johns Hopkins University Press, Baltimore, 1989.
[44] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes in C*, Cambridge University Press, 1988.