Using Points at Infinity for Parameter Decoupling in Camera Calibration

Jean-Yves Guillemaut, Alberto S. Aguado, and John Illingworth

J.-Y. Guillemaut and J. Illingworth are at the University of Surrey, UK. A. S. Aguado is with Electronic Arts UK.

Abstract

The majority of camera calibration methods, including the Gold Standard algorithm, use point based information and simultaneously estimate all calibration parameters. In contrast, we propose a novel calibration method that exploits line orientation information and decouples the problem into two simpler stages. We formulate the problem as minimisation of the lateral displacement between single projected image lines and their vanishing points. Unlike previous vanishing point methods, parallel line pairs are not required. Additionally, the invariance properties of vanishing points mean that multiple images related by pure translation can be used to increase the calibration dataset size without increasing the number of estimated parameters. We compare the method with vanishing point methods and the Gold Standard algorithm and demonstrate that it has comparable performance.

Index Terms

Computer Vision, Camera Calibration, Invariants.

I. Introduction

Camera calibration is the determination of parameters representing geometric properties of the imaging process. These parameters are constrained by equations linking the coordinates of 3D points and their projections. In general, information about the geometry in the image and the camera motion can be used to simplify camera calibration equations. For example, in [4], [6]–[9], [11], [20], [31], [32], vanishing points (VPs) and vanishing lines have been used to compute intrinsic and rotation parameters in a first stage, while the remaining parameters can be computed by solving a system with reduced degrees of freedom in a second stage. Similarly, specific controlled motions have been used in [9], [27], [30] to decompose the camera calibration process into a series of steps that are simpler than the full camera calibration method.

This paper considers the problem of decoupling translation from the other camera parameters by invariance properties. There are two main motivations. Firstly, decomposition in the parameter space leads to simpler sub-problems. Secondly, if translation is decoupled, data from additional images obtained by translation does not introduce additional parameters to the problem. That is, data size can be increased, and thereby estimation accuracy, without increasing the problem dimensionality. The idea of parameter decomposition, has been used in other areas of computer vision. In [2], it was shown that for two collections of 3D points related by a rotation and translation, the estimation of the motion can be decoupled based on the properties of the centroid. A similar problem for 2D motion projections is described in [19] and [23]. In the case of motion estimation from line correspondences, direction of lines have been used to compute the rotational part of the transformation [3]. Additionally, work on shape matching has considered the decomposition of rotation and translation for a 2D transformation of planar shapes [1].

Our approach presents some similarities with previous methods based on vanishing points [4], [6]–[9], [11], [20], [31], [32]. VPs have strong invariance properties, however, there are usually not many in images and they are difficult to compute. Even if considerable effort has been directed towards their estimation [24], existing VP based calibration methods usually require pairs of parallel scene lines to define VP location. In contrast, our approach computes points at infinity from arbitrary single straight lines in the scene, and formulates a novel constraint which relates single line orientation to the projections of the points at infinity (VPs). Thus, equations linking scene and image data can be expressed independently of translation. Methods that use lines for the estimation of motion and structure have been previously considered in [21], [33]. A discussion of the advantages of the use of lines in terms of accuracy in measurements is given in [33]. If orientation of lines can be more accurately and reliably measured than point location, then this results in more accurate features. In an implementation, straight lines can be defined by edges in the scene or by pairs of points; generally their number exceeds the number of parallel scene line pairs or usable distinguished image points (such as corners) that are necessary for many other calibration methods.

In Section II, we consider the use of points at infinity in the inverse image formation problem and define an invariant for the equations linking the coordinates of 3D points and their projections. Section III formulates the two stage camera calibration procedure based on this invariant. Finally experimental results with synthetic and real data are discussed in Section IV.

II. Inverse Image Formation and Points at Infinity

In the image formation process with a standard pinhole camera model, 3D scene points $\boldsymbol{P}_i = (X, Y, Z, 1)^\top$ are mapped into 2D image points $\boldsymbol{p}_i = (u, v, 1)^\top$ by a perspective projection

$$\boldsymbol{p}_i \sim \mathsf{K}[\mathsf{R}|\boldsymbol{t}]\boldsymbol{P}_i, \quad \text{with} \quad \mathsf{K} = \begin{bmatrix} f & -f\cot\theta & u_0 \\ & fr/\sin\theta & v_0 \\ & & 1 \end{bmatrix}. \tag{1}$$

The symbol $\sim$ denotes equality up to a non-zero scale factor. K represents the intrinsic parameters, which include the coordinates of the principal point (u_0, v_0), the focal length f in pixel units, the aspect ratio r, and the skew angle θ defined by the two axes of the camera. In practice,

θ is significant for CCD and optical cameras only ($\theta = 90^\circ$ for digital cameras). Additionally, r is expected to be close to 1, however should be computed for an accurate calibration. The extrinsic parameters represent the camera orientation (rotation matrix R) and position (translation vector $-\mathsf{R}^\top \boldsymbol{t}$) with respect to the world reference frame.

The inverse image formation problem consists in finding the values that minimises

$$\sum_i d(\boldsymbol{p}_i, \mathsf{K}[\mathsf{R}|\boldsymbol{t}]\boldsymbol{P}_i)^2, \tag{2}$$

where d defines a geometric distance. This can be generalised to a sequence of images by extending the sum to all the images. This cost function is called the *reprojection error*, its minimisation leads to the *Maximum Likelihood* (ML) estimate under some standard hypotheses on the noise distribution (measurement errors are Gaussian, see *e.g.* [18], pp86–87). When camera motion is considered, the solution involves a large number of parameters. However, if invariants are used, it can be simplified such that the minimum depends only on a subset of the parameters. For example, VPs can be used to decouple intrinsic and extrinsic parameters. Contrary to previous methods using VPs [4], [6], [8], [20], the decomposition we propose does not require any specific pattern such as parallel lines, but can be implemented from arbitrary lines or pairs of points in a known scene.

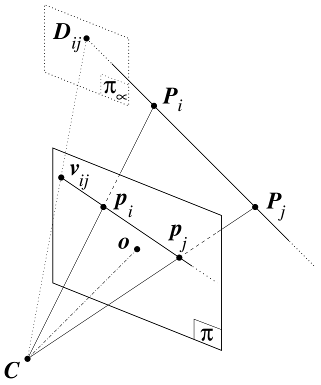

Fig. 1. Projection of a pair of 3D points in an image. The pair of 3D points $\{\boldsymbol{P}_i, \boldsymbol{P}_j\}$ defines a direction which is represented by a point $\boldsymbol{D}_{ij}$ in the plane at infinity $\boldsymbol{\pi}_\infty$. This point projects into a VP $\boldsymbol{v}_{ij}$ which is constrained to lie on the image line $(\boldsymbol{p}_i\boldsymbol{p}_j)$.

A 3-D point with Euclidean coordinates $(X, Y, Z)^\top$ is mapped to a point in the projective space with homogeneous coordinates $\lambda(X, Y, Z, 1)^\top$ ($\lambda \neq 0$). Inversely, a point with homogeneous coordinates $(X, Y, Z, W)^\top$ ($W \neq 0$) is mapped back to the point with Euclidean coordinates $(X/W, Y/W, Z/W)^\top$. If $W = 0$, the point is *at infinity*. These points define the plane at infinity representing the directions of the underlying affine space [12], *e.g.* $(V_X, V_Y, V_Z, 0)^\top$ represents the direction parallel to the vector $(V_X, V_Y, V_Z)^\top$. The projection of a point at infinity in an image is a VP. An important property is that they are independent of camera translation. Intuitively, one can compare them to stars in the sky or points far away on the horizon, which stay fixed as an observer moves in the scene. Points at infinity are defined by the direction of straight lines in the scene. If we consider a pair of points $\boldsymbol{P}_i = (X_i, Y_i, Z_i, W_i)^\top$ ($W_i \neq 0$) and $\boldsymbol{P}_j = (X_j, Y_j, Z_j, W_j)^\top$ ($W_j \neq 0$), the direction of the line $(\boldsymbol{P}_i\boldsymbol{P}_j)$ is represented by the point at infinity $\boldsymbol{D}_{ij} = W_i\boldsymbol{P}_j - W_j\boldsymbol{P}_i$, which can be written in the form $\boldsymbol{D}_{ij} = (\boldsymbol{d}_{ij}^\top, 0)^\top$. The projection of $\boldsymbol{D}_{ij}$ into the image defines a VP $\boldsymbol{v}_{ij} \sim \mathsf{K}[\mathsf{R}|\boldsymbol{t}]\boldsymbol{D}_{ij} = \mathsf{KR}\boldsymbol{d}_{ij}$ which is translation invariant. Since an homography preserves collinearity, $\boldsymbol{v}_{ij}$ must lie on the line $(\boldsymbol{p}_i\boldsymbol{p}_j)$ (see Fig. 1), that is $\boldsymbol{l}_{ij}^\top\boldsymbol{v}_{ij} = 0$, where $\boldsymbol{l}_{ij} \sim \boldsymbol{p}_i \times \boldsymbol{p}_j$ is the homogeneous representation of $(\boldsymbol{p}_i\boldsymbol{p}_j)$. With the notation $\mathsf{H} = \mathsf{KR}$, we can obtain an equation independent of the translation

$$\boldsymbol{l}_{ij}^\top \mathsf{H}\boldsymbol{d}_{ij} = 0. \tag{3}$$

Thus the problem in (2) can be reformulated in terms of the minimisation of $\sum_{i,j} d(\boldsymbol{l}_{ij}, \mathsf{H}\boldsymbol{d}_{ij})^2$. Once the rotation has been determined, the translation can be computed by considering (1) for a known R. As such, the original minimisation problem can be divided into two sub-problems.

III. Application to Camera Calibration

We consider the general problem of computing all the camera parameters from one or several images related by a translation, using a known camera calibration object. The system has $8+3n$ unknowns (where n is the number of images): 5 for K, 3 for R and, for each frame, 3 for $\boldsymbol{t}$. In general, when correspondences between 3D points and their images are known, the camera parameters can be computed simultaneously by solving for a matrix $\mathsf{M} = \mathsf{K}[\mathsf{R}|\boldsymbol{t}]$ satisfying $\boldsymbol{p}_i \sim \mathsf{M}\boldsymbol{P}_i$ for each world to image point correspondence. The procedure is called resectioning and requires $5\frac{1}{2}$ point correspondences in a general position. Once M is estimated, the individual camera parameters can be recovered by matrix decomposition [18] (p150).

Alternatively, the results of Section II can be used to decompose the full parameter space into two smaller sub-systems. The first one contains only the parameters from K and R (8 parameters), whereas the second one contains the remaining parameters from $\boldsymbol{t}$ (there are n independent systems of 3 parameters, one for each image). Thus we define two simpler problems:

1) *Intrinsic parameters and orientation estimation:* Given a set of world directions $\boldsymbol{d}_{ij}$ and the associated image lines $\boldsymbol{l}_{ij}$, compute a 3×3 full rank matrix $\mathsf{H} = \mathsf{KR}$ such that $\boldsymbol{l}_{ij}^\top \mathsf{H}\boldsymbol{d}_{ij}$ is minimised for each (i, j).
2) *Position estimation:* Given a set of world to image point correspondences $\boldsymbol{P}_i \leftrightarrow \boldsymbol{p}_i$, and two known matrices K and R, compute a vector $\boldsymbol{t}$ such that $\boldsymbol{p}_i \sim \mathsf{K}[\mathsf{R}|\boldsymbol{t}]\boldsymbol{P}_i$ for each i.

The 3×3 matrix H defined in 1) is the homography between the plane at infinity and the image plane. Once H is known, K and R can be recovered by a simple RQ decomposition

[18] (p150). However, 1) is different from a simple homography estimation problem. Namely, we do not have a strict correspondence, but only a constraint that establishes that a VP should lie on the image line. This is fundamentally different to other camera calibration methods that propose computing the VPs from parallel lines before estimating the camera parameters [4], [6]–[9], [11], [20], [31], [32]. It should also be noted that the equations defined in 1) are similar to the estimation of the fundamental matrix via the 8-point algorithm [15]. In the case of the fundamental matrix estimation however, the solution matrix must be degenerate to satisfy the singularity constraint (rank 2), whereas in our case a full rank matrix (rank 3) is sought. Problem 2) can be solved in a straightforward manner using simple least-square techniques, therefore we focus on solving 1) in the rest of the paper.

A. Practicality

Problem 1) uses only the directional information contained in the scene, derived from 3D lines with known direction, or pairs of known 3D points. One practical advantage of the decomposition method is that it can be used in situations where only directional information is present in the scene, this is for example the case for architectural applications [8], [20], wherein direction are defined by building edges. The method is also useful in other situations when one is interested only in the intrinsic parameters or in the orientation of the camera. Another advantage over standard camera resectioning is that when multiple images obtained from a purely translating camera are used, data size is increase, without requiring to compute additional translation parameters. Accurate calibration can therefore be done from directional information only. If all the camera parameters are required, a full camera calibration is performed by solving both problems separately. In that case Problem 1) still uses only directional information, while Problem 2) requires additional information, such as points correspondences.

B. Linear Solution

In comparison to non-linear methods, linear methods are significantly faster and easier to implement. Writing $\boldsymbol{l}_{ij} = (l_{ij_x}, l_{ij_y}, l_{ij_z})^\top$ and $\boldsymbol{d}_{ij} = (d_{ij_x}, d_{ij_y}, d_{ij_z})^\top$ and denoting by $\boldsymbol{h}$ the entries of H in row major order, a straightforward development of (3) leads to

$$(l_{ij_x}\boldsymbol{d}_{ij}^\top, l_{ij_y}\boldsymbol{d}_{ij}^\top, l_{ij_z}\boldsymbol{d}_{ij}^\top)\boldsymbol{h} = \boldsymbol{a}_{ij}\boldsymbol{h} = 0. \tag{4}$$

From a set of n directions, a $n \times 9$ matrix A is obtained by stacking up the terms $\boldsymbol{a}_{ij}$ from (4) for each direction. The vector $\boldsymbol{h}$ is then computed by solving the linear system $A\boldsymbol{h} = 0$ using for example Singular Value Decomposition as described in [18] (pp166–169); the solution obtained minimises $\|A\boldsymbol{h}\|$ subject to the constraint $\|\boldsymbol{h}\| = 1$. A may have a very large number of rows. For a memory efficient implementation, A can be transformed into a smaller dimension matrix $\hat{A}$ as follows. A is written in the form $A = [A_1, \ldots, A_N]^\top$. Denoting by $\hat{A}_i$ the 9×9 reduced measurement matrix of each block matrix A_i (see [18] p175 for a definition), we can define $\hat{A} = [\hat{A}_1, \ldots, \hat{A}_N]^\top$. The matrix $\hat{A}$ has dimension $9N \times 9$ (with $N \ll n$) and it can be shown that it satisfies $\|A\boldsymbol{h}\| = \|\hat{A}\boldsymbol{h}\|$, allowing A to be substituted by $\hat{A}$. The system has 8 unknowns (H has 9 entries, but it is defined up to a non zero scale factor) and each point at infinity leads to one equation, therefore a minimal solution is obtained from 8 directions in a general position. The term general position will be made clearer in Section III-D.

Without normalisation of the input data, the previous algorithm may be ill-conditioned and thereby perform poorly. Here we follow a similar normalisation scheme as in the original work from Hartley in the case of the computation of the fundamental matrix [15] and camera calibration [17]. $\boldsymbol{l}_{ij}$ are normalised such that their centroid is the coordinate origin $(0,0)^\top$ and the average distance from the origin is $\sqrt{2}$. $\boldsymbol{d}_{ij}$ may be at or near infinity, therefore a different normalisation strategy must be used in this case, *e.g.* see [18] (pp113–114). We carried out an isotropic normalisation; $\boldsymbol{d}_{ij}$ are normalised such that their homogeneous coordinates (U_{ij}, V_{ij}, W_{ij}) satisfy $\sum_{ij} U_{ij} = \sum_{ij} V_{ij} = 0$, $\sum_{ij} U_{ij}^2 = \sum_{ij} V_{ij}^2 = \sum_{ij} W_{ij}^2$, and $\forall i,j \quad U_{ij}^2 + V_{ij}^2 + W_{ij}^2 = 1$.

C. Minimisation of a geometric distance

The linear solution is computationally attractive. However, it presents some limitations such as the non-invariance to the coordinate reference frame, which required us to introduce normalisation. Another criticism of this method is that it minimises a term called algebraic distance, which has little physical meaning. In this section, a non-linear method which minimises a geometric distance is introduced. The geometric distance d_{geom} from a VP $\boldsymbol{v}$ with homogeneous coordinates $(u,v,w)^\top$ $(w \neq 0)$ to the corresponding line $\boldsymbol{l}$ with homogeneous coordinates $(a,b,c)^\top$ is

$$d_{\text{geom}}(\boldsymbol{v}, \boldsymbol{l}) = \frac{1}{\sqrt{a^2+b^2}} \left| a\frac{u}{w} + b\frac{v}{w} + c \right| . \tag{5}$$

Direct minimisation of the sum of squared geometric distance can lead to inaccurate results because all the measurements do not have the same confidence. For example, shorter image segments lead to larger errors in the computation of line coordinates, or scene directions nearly parallel to the image plane lead to VP far away from the image centre, and therefore larger errors in the computation of d_{geom}. Thus we adopt a cost function d_{mah} based on the Mahalanobis distance with respect to the error covariance associated to each measurement. In this case, the error covariance is the variance σ^2_{geom} associated to d_{geom}, and the cost function is

$$d_{\text{mah}}(\boldsymbol{v}, \boldsymbol{l}) = \frac{1}{\sigma_{\text{geom}}} d_{\text{geom}}(\boldsymbol{v}, \boldsymbol{l}). \tag{6}$$

The exact computation of σ_{geom} is complicated as $\boldsymbol{v} = \boldsymbol{\mathsf{H}d}$ depends on H, whose parameters are unknown. We make the assumption that the error is due only to the uncertainty in the image line extraction. Physically, this means that either the camera calibration object used is engineered with a very good accuracy compared to the accuracy of the extraction of the image data, or that the accuracy is the same for all 3D points, which is a reasonable assumption. Using error propagation [13], [18] (pp123–125), we approximate up to a first-order the variance of d_{geom} by $\sigma^2_{\text{geom}} = \mathsf{J}\mathit{\Sigma}\mathsf{J}^\top$, where $\mathit{\Sigma}$ is the covariance matrix of the image lines $\boldsymbol{l}$ and $\mathsf{J} = [\frac{\partial d_{alg}}{\partial a}, \frac{\partial d_{alg}}{\partial b}, \frac{\partial d_{alg}}{\partial c}]$ is the Jacobian matrix of d_{geom} evaluated at $(\boldsymbol{v}, \boldsymbol{l})$. We assume that each coordinate of the end points $(x_1, y_1, 1)^\top$ and $(x_2, y_2, 1)^\top$ of $\boldsymbol{l}$ are independent variables following Gaussian distributions with mean the exact coordinate value and standard deviation σ. After expansion, we obtain:

$$\mathit{\Sigma} = \sigma^2 \begin{bmatrix} 2 & 0 & -x_1 - x_2 \\ 0 & 2 & -y_1 - y_2 \\ -x_1 - x_2 & -y_1 - y_2 & x_1^2 + x_2^2 + y_1^2 + y_2^2 \end{bmatrix}, \mathsf{J} = \frac{1}{\sqrt{a^2+b^2}}[\frac{u}{w} - \frac{a(a\frac{u}{w}+b\frac{v}{w}+c)}{a^2+b^2}, \frac{v}{w} - \frac{b(a\frac{u}{w}+b\frac{v}{w}+c)}{a^2+b^2}, 1].$$

d_{mah} is non-linear. A solution can be computed by using standard non-linear minimisation algorithms, such as the Levenberg-Marquardt algorithm [22] initialised with the result of the linear method. In the implementation, a suitable parametrisation for the rotation is for example a three-vector using the Rodrigues formula as described in [28]: the three parameters define a vector parallel to the rotation axis whose magnitude is the rotation angle [12].

D. Degenerate Configurations

It was seen in Section III-B that a minimum of eight world directions $\boldsymbol{d}_{ij}$ and their associated image lines $\boldsymbol{l}_{ij}$ in a "general position" are necessary to compute H. We clarify here what is meant by "general position" and compare the degeneracies obtained with the case of camera resectioning [5] [18] (chapter 21). We suppose here that there are at least eight correspondences

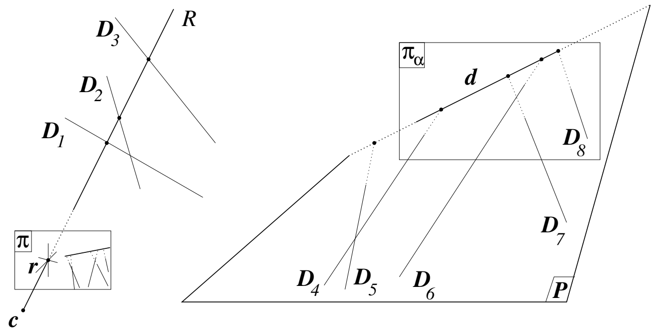

Fig. 2. The "rank 1" degenerate configurations. The 3D lines either form a Linear Line Complex with a ray $\mathcal{R}$ going through the camera centre $\boldsymbol{c}$ (*e.g.* $\boldsymbol{D}_1$, $\boldsymbol{D}_2$, $\boldsymbol{D}_3$), or are parallel to a plane $\mathcal{P}$ (*e.g.* $\boldsymbol{D}_4$, …, $\boldsymbol{D}_8$).

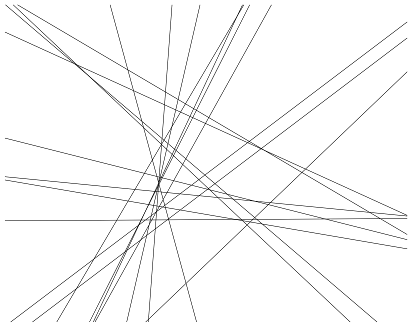

Fig. 3. Camera image of a sample "rank 2" degenerate configuration. Contrary to the "rank 1" case, there is no simple geometric pattern characterising the arrangement of the 3D lines and the pose of the camera.

and study the degeneracies with a single camera; the case of multiple translated cameras is treated analogously to a single camera if we introduce translated 3D lines in the scene.

We begin by reminding that an image is determined up to projectivity by the camera centre alone (see [18], p517), thus all the parameters of the camera, except the camera centre, can be ignored in the characterisation of the degeneracies. Let us now suppose that there is a degeneracy. There exists two distinct rank 3 matrices H and H' satisfying (3) for all (i, j). It results immediately from the bilinearity of (3) that $\mathsf{H}_\theta = \mathsf{H} + \theta\mathsf{H}'$ (θ is a scalar) is also solution. However the determinant of this matrix $\det(\mathsf{H}_\theta)$ is a polynomial of degree 3 in θ, thus it has at least a real solution θ_0 ($\theta_0 \neq 0$, otherwise $\mathsf{H}_\theta = \mathsf{H}$ and it follows that H_θ is rank 3 like H, which contradicts the fact its determinant is zero). H_{θ_0} does not have full rank, *i.e.* $\operatorname{rank}(\mathsf{H}_{\theta_0}) \leq 2$, because by construction $\det(\mathsf{H}_{\theta_0}) = 0$. In addition, it is clear that $\operatorname{rank}(\mathsf{H}_{\theta_0}) \neq 0$, otherwise there would exist a non zero θ such that $\mathsf{H} + \theta\mathsf{H}' = 0$, *i.e.* $\mathsf{H} \sim \mathsf{H}'$, which contradicts the original assumption that the solutions are distinct. We conclude that there exists a matrix H_{θ_0} which is rank 1 or 2. According to this rank, we obtain two types of degeneracies. These configurations can occur for any number of 3D lines (not restricted to the minimum case of eight lines) as long as the features are arranged according to the patterns described below.

A "rank 1" degeneracy occurs if there exists a ray $\mathcal{R}$ going through the camera centre and a plane $\mathcal{P}$ such that all the 3D lines either intersect $\mathcal{R}$ or are parallel to $\mathcal{P}$ (see Fig. 2). The set of 3D lines intersecting at a common line $\mathcal{R}$ forms a pattern called Linear Line Complex (see [26]). In this case, the degenerate solution matrix can be written in the form $\mathsf{H}_\theta = \boldsymbol{r}\boldsymbol{d}^\top$, where $\boldsymbol{d}$

is a 3-vector orthogonal to the null-space of H_θ and $\boldsymbol{r}$ is a 3-vector in the span of H_θ. Replacing in (3), we obtain $\boldsymbol{l}_{ij}^\top \boldsymbol{r}\boldsymbol{d}^\top \boldsymbol{d}_{ij} = 0$, *i.e.* $\boldsymbol{l}_{ij}^\top \boldsymbol{r} = 0$ or $\boldsymbol{d}^\top \boldsymbol{d}_{ij} = 0$. $\boldsymbol{l}_{ij}^\top \boldsymbol{r} = 0$ means that the point $\boldsymbol{r}$ belongs to the image line, *i.e.* the 3D line intersects the ray $\mathcal{R}$ obtained by back projecting $\boldsymbol{r}$. $\boldsymbol{d}^\top \boldsymbol{d}_{ij} = 0$ means that the 3D line intersects the plane at infinity in the line $\boldsymbol{d}$, if we call $\mathcal{P}$ any plane intersecting the plane at infinity in $\boldsymbol{d}$, it results that the 3D line is parallel to $\mathcal{P}$.

It is difficult to give a simple geometric characterisation of the "rank 2" degeneracy but our experience is that this configuration is rather unlikely to happen in human engineered patterns. We show an image of a sample "rank 2" degenerate configuration generated with Matlab$^{\text{TM}}$ in Fig. 3. We now give an interesting analogy with camera resectioning. In that case, the most important degenerate configurations arise when i) the points all lie on the union of a plane and a single straight line containing the camera centre, or ii) the camera and points all lie on a twisted cubic (see [18], chapter 21, for an exhaustive enumeration). It is straightforward that case i) results in our "rank 1" degenerate configuration when pairs of points are taken to form lines. Computer simulations showed that case ii) results in the "rank 2" degenerate configuration.

E. Constrained Camera Calibration

Until now, we have considered a general projective camera. In the case of restricted cameras, for example zero-skew, known aspect ratio, known principal point or known intrinsic parameters, the camera calibration is still possible by minimising either a geometric error or an algebraic error. The geometric error is minimised by optimising the camera parameters to be estimated, while the other constraints are enforced. An algebraic error (which leads to a smaller size minimisation) can be minimised by defining a reduced measurement matrix as described in [17].

IV. Results

In this section, the decomposition method is evaluated with images of a calibration grid. Three different implementations are considered: linear method (using SVD decomposition) with normalisation, linear method without normalisation, and non-linear method (using Levenberg-Marquardt method). In the figures, the different methods are respectively labelled *decomposition (norm. linear)*, *decomposition (linear)* and *decomposition (non-linear)*. In addition, we include the results of two classical camera calibration methods described below.

Camera calibration from point correspondences [17], [18]. The algorithm was implemented with a non-linear minimisation (bundle adjustment) using Levenberg-Marquardt algorithm, and initialised by the linear solution obtained from SVD decomposition with preliminary normalisation of the data. This is a standard camera calibration technique described in [18] (p170) as the Gold Standard Algorithm. The graphs corresponding to this method are labelled *standard*.

Camera calibration from VPs [6], [8], [20]. VPs are computed from the intersection of parallel lines. Three sets of mutually orthogonal parallel lines define three VPs which can be used to compute the intrinsic parameters (assuming zero skew and a known aspect ratio) and the rotation. We use a linear algorithm with normalisation [16]. The translation is recovered by considering additional point correspondences. The graphs corresponding to this method are labelled *vanishing* or *vanishing (aspect ratio = 1)*. In the first case, the aspect ratio is obtained from the standard calibration method, in the second case the aspect ratio is assumed to be 1.

Experiments were performed with synthetic and real data. In both cases, a calibration grid made of two orthogonal planes was used. The directions required for the decomposition method were obtained by considering pairs of points. The set of parallel lines required for calibration from VPs is obtained by least-square fitting a line to each set of aligned points defined by the grid. The common criteria chosen for the evaluation of each algorithm is the Root Mean Squared (RMS) point reprojection error defined by $\epsilon_{\text{rep}} = \sqrt{\frac{1}{N}\sum_i d(\boldsymbol{p}_i, \mathsf{K}[\mathsf{R}|\boldsymbol{t}]\boldsymbol{P}_i)^2}$. In order to obtain less biased results, we use different points for error measure evaluation and camera calibration. The points used to compute the reprojection error are contained in a third plane orthogonal to the two other calibration planes. It should be noted that the reprojection error thus defined is different from the cost function minimised by the non-linear decomposition method, thus it may not necessarily show a reduction in the error when comparing the linear with the non-linear decomposition method. In the case of simulations with synthetic data another criteria called RMS estimation error is used to evaluate the accuracy of the computation of the camera parameters. It is defined by $\epsilon_{\text{est}} = \sqrt{\frac{1}{N}\sum_i (x-\overline{x})^2}$ where $\overline{x}$ is the ground truth parameter and x is the estimated parameter.

A. Synthetic data

Each plane in the synthetic grid contains 36 control points. The virtual camera has the following parameters: $u_0 = 384\,\text{pixels}$, $v_0 = 247\,\text{pixels}$, $f = 714.3\,\text{pixels}$, $r = 1.167$, and $\theta =$

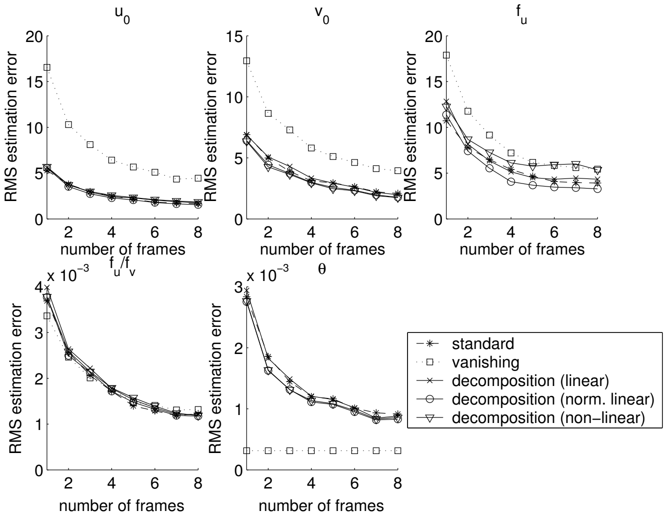

Fig. 4. RMS estimation error for each parameter with respect to the number of frames considered in the case of synthetic experiments of camera calibration with a translating camera (the noise level is $\sigma = 1$ pixel). The results were obtained from 100 experiments.

1.570 rad. We have tested the decomposition method in the context of camera calibration when a pure translation motion is generated. We generate a sequence of 10 images separated by such a motion. To study the influence of the number of frames, subsets of 1 to 8 images are randomly selected, and the camera calibration is performed with the images selected. The experiments are repeated 100 times for each size of the subset. Fig. 4 shows the RMS estimation error for each of the intrinsic parameters and Fig. 5 shows the RMS point reprojection error with respect to the number of frames considered. The noise level of the Gaussian noise was set to $\sigma = 1$ pixel and no information about the translation was used. The RMS point reprojection error for the different methods decreases rapidly when the number of frames increases and seems to converge to some asymptotic values. It can be observed that the normalised method leads to more accurate results than the non-normalised one. The non-linear method seems accurate but it does not give the expected improvement in accuracy over the linear methods. The decomposition method appears to be slightly more accurate than the other methods when the number of frames is increased. It is interesting to compare the asymptotic values observed with the RMS point reprojection error for the ML estimator which represents the minimum error under some standard Gaussian hypotheses on the noise distribution and is given by $\epsilon_{\mathrm{ML}} = \sigma \, [11/(2 \times 36)]^{1/2} \approx 0.391$ (see [18], p121). The asymptotic values observed seem to be close to this limit value, especially in

TABLE I

ESTIMATED INTRINSIC PARAMETERS WITH 8 FRAMES IN THE CASE OF REAL EXPERIMENTS OF CAMERA CALIBRATION WITH A TRANSLATING CAMERA. FOR EACH OF THE THREE METHODS THE MEAN VALUE AND THE STANDARD DEVIATION OBTAINED FROM 100 EXPERIMENTS ARE GIVEN.

		u_0 (pixels)	v_0 (pixels)	f (pixels)	r	θ (rad)
standard	mean	327.4	257.0	695.3	1.107	1.574
	std	0.231	0.230	0.821	0.326×10^{-3}	0.252×10^{-3}
vanishing point (aspect ratio from standard method)	mean	329.6	257.5	695.4	1.107	1.571
	std	1.019	0.275	0.838	0.370×10^{-3}	0
vanishing point (aspect ratio enforced to 1)	mean	321.1	325.3	676.9	1.000	1.571
	std	0.812	0.349	0.840	0	0
decomposition (linear)	mean	326.7	258.4	696.0	1.107	1.574
	std	0.282	0.740	0.598	0.365×10^{-3}	0.160×10^{-3}
decomposition (norm. linear)	mean	326.6	257.6	695.2	1.107	1.574
	std	0.272	0.568	0.627	0.341×10^{-3}	0.152×10^{-3}
decomposition (non-linear)	mean	326.6	257.6	695.2	1.107	1.574
	std	0.270	0.568	0.625	0.337×10^{-3}	0.153×10^{-3}

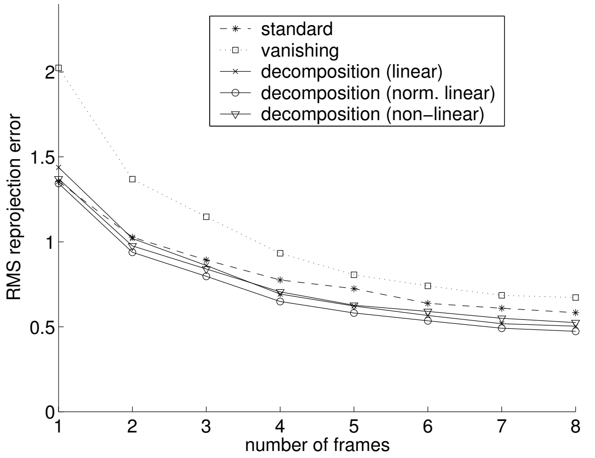

Fig. 5. RMS point reprojection error with respect to the number of frames considered in the case of synthetic experiments of camera calibration with a translating camera (the noise level is $\sigma = 1\,$pixel). The results were obtained from 100 experiments.

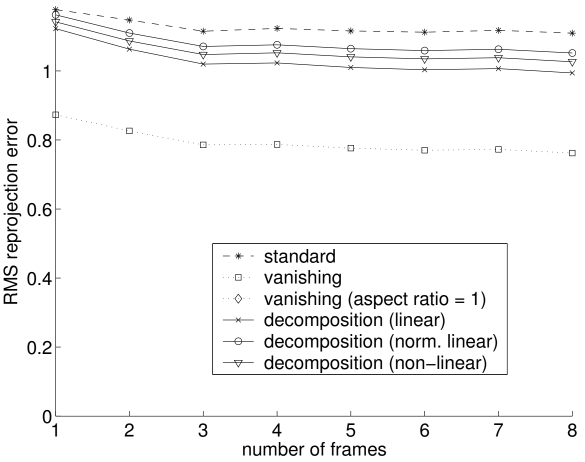

Fig. 6. RMS point reprojection error with respect to the number of frames considered in the case of real experiments of camera calibration with a translating camera. The results were obtained from 100 experiments. The graph corresponding to the vanishing point method with an aspect ratio assumed to be 1 lead to large RMS point reprojection error of the order of 15 which are not visible in the figure.

the case of the decomposition method, which suggest the algorithm is performing well.

B. Real data

A sequence of 20 images of a grid was produced with a Pulnix TMC-7DSP camera equipped with a 6 mm lens. The calibration grid was made of three planar grids, each containing 36 control points generated by a printer, which were positioned on three mutually orthogonal planes. The coordinates of the control points were verified with a measuring tape, the accuracy is estimated to be of the order of one millimetre. The coordinates of the images of the control points have been computed using a Harris corner detector [14]. The camera is mounted on a robot arm that is used to generate a pure translation motion. The images obtained present some lens distortion. In this case, only the radial distortion is corrected, and a first order coefficient appears to be sufficient [29]. Here the lens distortion can be appropriately calibrated from the image of lines, by requiring them to be straight. This technique is known as the plumb-line method in the photogrammetry literature [25]; an implementation for computer vision is presented in [10]. For active camera calibration, random subsets of frames were selected, and for each subset size, 100 experiments were run. The mean and the standard deviation were computed for each number of frames considered. Table I compares the values obtained for each parameter for the different methods considered. Fig. 6 shows the RMS point reprojection error.

It can be seen that the accuracy of the methods increase with the number of frames. The decomposition method and the method computing VPs (with the aspect ratio obtained from the standard method) perform better than the method using point correspondences. With the method using point correspondences, the size of the parameter space increases each time a new frame is added. For example, if n frames are considered, then there are $8 + 3n$ parameters to estimate (5 intrinsic plus 3 for the orientation, and 3 for the position of the camera corresponding to each frame). When the size of the parameter space increases, the risk of being trapped in a local minimum increases and it becomes less likely to converge to the global optimum. In comparison, for the decomposition method and the method computing vanishing points the size of the parameter space remains fixed for the first stage (8 parameters). Similarly to the synthetic data analysis, the linear method with normalisation is more stable than the non-normalised solution. In the particular case where the method using VPs is provided with the aspect ratio obtained from the standard method, it gives more accurate results than the other methods. However, the method computing VPs has some limitations: i) it assumes the camera has zero skew, ii) it requires to provide a value for the aspect ratio. The zero-skew assumption is correct for a digital camera, however does not hold for all CCD cameras. Some inaccuracies are expected if the vanishing point method is provided with an incorrect value for the aspect ratio. For example, it can be seen in Fig. 6 that the method performs poorly when initialised with an aspect ratio of one, which is however a reasonable value for most CCD cameras. It is possible to provide a better value for the aspect ratio, *e.g.* by using the non-linear standard camera calibration method, but it requires to run another preliminary calibration algorithm.

Overall if we compare the results of synthetic and real data, it can be seen that the decomposition method usually performs better than the other methods when multiple frames are used. It can be noticed that in some cases the non-linear decomposition method is actually less accurate than the linear one. From a theoretical point of view, there is no reason for the non-linear method to be more accurate since the cost functions used are different (see introduction of this section). Practically, the results observed are very close and the discrepancy are not significant; they could be due to the approximations performed when formulating the non-linear cost function or to different noise conditions in synthetic and real experiments.

V. Conclusions

We have shown that points at infinity can be used to decouple translation and rotation from equations linking the coordinates of 3D points and their perspective projections. Directions in the scene represent points at infinity that project to VPs. Thus, the resulting equations are independent of translation. We have shown that this property can be used to split the parameter search in camera calibration. Additionally, we have shown how translational motion can be used to increase the amount of data for camera calibration without increasing the dimensionality of the problem. Experiments show that, if several frames separated by a translation motion are considered, the decomposition method performs generally better than the standard method or the method computing VPs.

Acknowledgment

This work is supported by EPSRC project grant number GR/R08629/01.

References

[1] A. S. Aguado, E. Montiel, and M. S. Nixon. Invariant characterization of the Hough transform for pose estimation of arbitrary shapes. In *British Machine Vision Conference*, volume 2, pages 785–794, 2000.

[2] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.

[3] A. Bartoli, R. Hartley, and F. Kahl. Motion from 3d line correspondences: Linear and non-linear solutions. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 477–484, 2003.

[4] P. Beardsley and D. Murray. Camera calibration using vanishing points. In *British Machine Vision Conference*, pages 416–425, 1992.

[5] T. Buchanan. The twisted cubic and camera calibration. *Computer Vision, Graphics and Image Processing*, 42:130–132, 1988.

[6] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2):127–140, 1990.

[7] W. Chen and B. C. Jiang. 3-d camera calibration using vanishing point concept,. *Pattern Recognition*, 24(1):57–67, 1991.

[8] R. Cipolla, D. P. Robertson, and E. G. Boyer. Photobuilder - 3D models of architectural scenes from uncalibrated images. In *IEEE International Conference on Multimedia Computing and Systems*, volume 1, pages 25–31, 1999.

[9] K. Daniilidis and J. Ernst. Active intrinsic calibration using vanishing points. *Pattern Recognition Letters*, 17(11):1179–1189, 1996.

[10] F. Devernay and O. D. Faugeras. Straight lines have to be straight. *Machine Vision and Applications*, 13(1):14–24, 2001.

[11] T. Echigo. A camera calibration technique using three sets of parallel lines. *Machine Vision and Applications*, 3(3):159–167, 1990.

[12] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, 1993.

[13] R. M. Haralick. Propagating covariance in computer vision. *International Journal of Pattern Recognition and Artificial Intelligence*, 10(5):561–572, 1996.

[14] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, University of Manchester, 1988.

[15] R. I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997.

[16] R. I. Hartley. Lines and points in three views and the trifocal tensor. *International Journal of Computer Vision*, 22(2):125–140, 1997.

[17] R. I. Hartley. Minimizing algebraic error in geometric estimation problems. In *International Conference on Computer Vision*, pages 469–476, 1998.

[18] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[19] D. J. Heeger and A. D. Jepson. Subspace methods for recovering rigid motion i: Algorithm and implementation. *International Journal of Computer Vision*, 7(2):95–117, 1992.

[20] D. Liebowitz, A. Criminisi, and A. Zisserman. Creating architectural models from images. In *Eurographics*, volume 18, pages 39–50, 1999.

[21] Y. Liu and T. S. Huang. A linear algorithm for determining motion and structure from line correspondences. *Computer Vision, Graphics, and Image Processing*, 44(1):35–57, 1988.

[22] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C, The Art of Scientific Computing*. Cambridge University Press, second edition, 1992.

[23] J. H. Reiger and D. T. Lawton. Processing differential image motion. *Journal of the Optical Society of America A*, 2:354–359, 1985.

[24] J. A. Shufelt. Performance evaluation and analysis of vanishing point detection techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(3):282–288, 1999.

[25] C. C. Slama. *Manual of Photogrammetry*. American Society of Photogrammetry, fourth edition, 1980.

[26] G. P. Stein and A. Shashua. On degeneracy of linear reconstruction from three views: Linear Line Complex and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(3):244–251, 1999.

[27] D. Stevenson and M. Fleck. Robot aerobics: Four easy steps to a more flexible calibration. In *International Conference on Computer Vision*, pages 34–39, 1995.

[28] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – A modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, 2000.

[29] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-self tv cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344, 1987.

[30] T. Viéville. *A Few Steps Towards 3D Active Vision*. Springer Verlag, 1998.

[31] L. L. Wang and W. H. Tsai. Computing camera parameters using vanishing-line information from a rectangular parallelepiped. *Machine Vision and Applications*, 3(3):129–141, 1990.

[32] L. L. Wang and W. H. Tsai. Camera calibration by vanishing lines for 3-d computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):370–376, 1991.

[33] J. Weng and T. Huang. Motion and structure from line correspondences: Closed-form solution, uniqueness, and optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(3):318–335, 1992.