Using Geometric Constraints Through Parallelepipeds for Calibration and 3D Modelling

Marta Wilczkowiak, Peter Sturm, Edmond Boyer

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Using Geometric Constraints Through Parallelepipeds for Calibration and 3D Modelling

Marta Wilczkowiak — Peter Sturm — Edmond Boyer

N° 5055

Novembre 2003

THÈME 3

ISSN 0249-6399 ISRN INRIA/RR--5055--FR+ENG

INRIA
RHÔNE-ALPES

Using Geometric Constraints Through Parallelepipeds for Calibration and 3D Modelling

Marta Wilczkowiak , Peter Sturm , Edmond Boyer

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet MOVI

Rapport de recherche n° 5055 — Novembre 2003 — 51 pages

Abstract: This paper concerns incorporation of geometric information in the camera calibration and 3D modeling. Using the geometric constraints enables stabler results and allows to perform tasks with fewer images. Our approach is motivated and developed within a framework of semi-automatic 3D modeling, where the user defines geometric primitives and constraints between them. It is based on the observation that constraints such as coplanarity, parallelism or orthogonality, are often embedded intuitively in parallelepipeds. Moreover, parallelepipeds are easy to delineate by a user, and are well adapted to model the main structure of e.g. architectural scenes. In this paper, first a duality that exists between the shape of a parallelepiped and the intrinsic parameters of a camera is described. Then, a factorization-based algorithm exploiting this relation is developed. Using the images of the parallelepipeds, it allows to simultaneously calibrate cameras, to recover shapes of parallelepipeds and to estimate the relative pose of all entities. Dealing with a well constrained three-dimensional structure makes it possible to overcome the common problems of the factorization methods: missing data and unknown scale factors. The reconstruction obtained this way is of the affine character. To remove the affine ambiguity, all the available metric information: constraints on parallelepipeds' edge lengths and angles, as well as the usual self-calibration constraints on cameras can be used simultaneously. The proposed algorithm is completed by a study of the singular cases of the calibration method. Also a method for the reconstruction of scene primitives that are not modeled by parallelepipeds is introduced. The method is validated by various experimental results for real and simulated scenes, for cases where a single or several views are available.

Key-words: camera calibration, geometrical constraints, parallelepipeds, factorization, 3D scene analysis

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier (France)
Téléphone : +33 4 76 61 52 00 —Télécopie +33 4 76 61 52 52

Comment utiliser les parallélépipèdes pour calibrer des images et construire des modèles 3D

Résumé : Ce rapport traite de l'utilisation de contraintes géométriques dans les processus de calibrage de caméras et de modélisation 3D à partir d'images. Ces processus sont en effet sensibles aux bruits et les contraintes géométriques, introduites par l'utilisateur, permettent d'en stabiliser les résultats même lorsque peu d'image sont considérées. Nous nous focalisons, dans ce document, sur les contraintes géométriques associées à un type de primitive particulier, les parallélépipèdes. Ces primitives permettent de caractériser des contraintes de coplanarité, de parallélisme et d'orthogonalité. De plus, les parallélépipèdes sont facilement identifiables par l'utilisateur dans les images et sont bien adaptés à la modélisation de bâtiments ou d'environnements architecturaux. Nous mettons en évidence, dans ce rapport, le fait que l'image d'un parallélépipède de caractéristiques connues caractérise entièrement la transformation de projection dans l'image. Il y a en fait dualité entre les caractéristiques du parallélépipède et celles de la caméra au travers de la transformation de projection. Nous développons ensuite un algorithme de factorisation qui exploite cette relation de dualité et permet de calibrer et de reconstruire simultanément l'ensemble des caméras et des parallélépipèdes définis par l'utilisateur dans un jeu d'images. L'intérêt des parallélépipèdes est de résoudre les difficultés classiques des méthodes de factorisation appliquées aux points : les données manquantes et les facteurs d'échelle inconnus. La reconstruction obtenue de cette manière est de type affine. Pour lever l'ambiguïté, l'ensemble des informations métriques sur les parallélépipèdes et sur les caméras peut être utilisées simultanément. Nous complétons la présentation de l'algorithme par une étude des configurations singulières pour le calibrage. Une méthode de reconstruction des primitives qui ne sont pas modélisées par les parallélépipèdes est aussi présentée. Les performances de l'algorithme sont démontrées au travers d'expériences sur des données synthétiques et réelles, pour des jeux d'une seule ou de plusieurs images.

Mots-clés : calibrage de caméras, contraintes géométriques, parallélépipèdes, factorisation, analyse 3D

Contents

1 Introduction

Efficient 3D modeling from images is one of the most challenging issues in computer vision. The tremendous research effort made to develop feasible methods in this domain has proved that recovering 3D structures from 2D images is a difficult and often under-constrained problem. Several reasons account for that, including the fundamental fact that without any prior information on cameras, or on the scene to recover, a metric reconstruction is not possible at all [13, 19]. This is why knowledge on the acquisition process, or on the scene, is required. A number of approaches have been proposed to exploit various prior information, both on camera and scene parameters. Such prior information do not only solve the projective ambiguity in the reconstruction but do also usually stabilize the sensitive reconstruction process. Furthermore, it often leads to simple and direct solutions for the estimation of both camera and scene parameters, which may eventually be adjusted non-linearly for highest accuracy. In addition, it enables modeling from small sets of images, in particular from single images, thus making possible reconstructions from images not originally taken for that purpose, like archival images or images from the internet for instance.

The literature on using prior information for self-calibration and for metric reconstruction is vast and widely scattered. An exhaustive review of all existing approaches is therefore beyond the scope of this article, and we will concentrate on works which have somehow inspired the method we propose, especially the direct approaches giving a good first estimate of the camera and scene parameters. Among many possible criteria for classifying modeling methods in this context, two appear to be more natural: first, the type of information which is used and second, how this information is used. There is a large variety of information which can be incorporated into a 3D modeling process. It can be simple knowledge on camera intrinsic parameters or poses (stationarity, translation, etc.) or on 3D scene structures (calibration patterns); it can also be the complete knowledge of calibration primitives and scene elements such as points, lines and planes, as well as a partial knowledge of high-level primitives like cubes, prisms, cylinders, etc. Nonetheless, note that whatever the information is, it can often be used at any stage of the 3D modeling process, including the initial intrinsic calibration, the pose estimation, the model reconstruction or even an additional non-linear adjustment of the initial estimate at every step.

The first class of approaches we mention is based on known positions of points in the 3D space, or known calibration patterns [50]. Unfortunately, such information relies on specific acquisition systems and is therefore seldom available in general situations. The use of prior knowledge on some intrinsic camera parameters, i.e. self-calibration, offers the opportunity to build much more flexible systems. In standard self-calibration algorithms [30, 48, 18, 33], the 3D reconstruction is done in 3 steps, recovering, in order, the projective, affine and Euclidean strata, the projective–affine step being considered as the most non-linear and thus most difficult step. One of the main problems are the critical motion sequences, for which the self-calibration does not have a unique solution [40]. This problem has been dealt with by restraining the camera motions [17, 10, 2] or by incorporating prior knowledge on the camera [58] or the scene into the calibration process. To get stable results for self-calibration, an important number of images is usually necessary.

A large variety of geometric constraints can disambiguate projective reconstructions from metric ones and allow to decrease the number of images required to obtain a satisfying reconstruction. Many of them can easily be incorporated into a self-calibration framework. A very common constraint in this category is given by vanishing points of mutually orthogonal directions, as defined by known cuboidal structures [7, 9, 8] or by dominating scene directions [24]. Also, knowing the metric structure of scene planes appears to be useful in this context, through rectified planes [26], maps [5] or known 3D plane–image homographies [42, 57]. It is also possible to use multiple images of unknown planes, however more images in general positions are needed in that case [49, 27].

Another category of approaches allows for the simultaneous recovery of cameras and 3D models. Factorization methods have been successfully applied to points [46, 44, 28, 39], lines [47, 29] and planes [38, 41]. Even though accurate, factorization algorithms suffer from missing data, i.e. when one primitive is not present in all the concerned images. That is why various systems based on image tensors have been proposed [3, 14]. These systems reconstruct the scene structure using 2, 3 or 4 images, and eventually sequentially or hierarchically join these reconstructed parts into consistent models. An overview of this can be found in [20].

When cameras are calibrated, it is relatively easy to reconstruct 3D structure. However, and as mentioned previously, using geometric constraints improves dramatically the reconstruction quality, especially when a single or only few images are considered. Even simple constraints can be very efficient for this purpose. In [1, 43], vanishing lines of planes and coplanarity constraints are used for single image reconstruction. More complex systems, dealing with multiple images, use constraints such as coincidence, parallelism, orthogonality, etc., between points, lines, and planes [35, 21, 16, 53, 55].

Most of the mentioned methods give solutions for both reconstruction and calibration. However, it is usually advantageous to optimize the obtained parameters using non-linear methods in an additional step. One way to incorporate geometrical information in this process is to augment the cost function by the penalty terms corresponding to the model constraints [31] or use the constrained optimization techniques [45, 32, 51]. The first type of methods however do not always ensure that the final structure will respect the given constraints exactly, and the constrained optimization techniques are in general designed for small equation systems, sometimes not sufficient to describe all the 3D scene dependencies. In answer to this, some systems rather try to find a minimal subset of parameters such that the associated scene models conform to the geometric constraints, and then, optimize over that subset. This includes methods which use simple primitives and constraints to compute the minimal scene parameterization [4, 55], as well as methods which are based on high-level scene descriptions through complex primitives like cubes, prisms, cylinders, etc. [11]. Recently, some effort has been devoted to the automatic detection of such primitives [12, 52]. The methods in this category ensure, by the strong inherent geometric constraints, that the final models are visually correct. On the other hand, they require scenes to be composed of basic primitives only, which is not always the case, see for example most archaeological environments.

In this paper, we address the first part of the 3D model acquisition process, the intrinsic and extrinsic camera calibration. In particular, we study the use of a specific calibration primitive: the parallelepipeds. Parallelepipeds are frequently present in man-made environments and they naturally encode the affine structure of the scene. Any information about their Euclidean structure (angles or ratios of edge lengths), possibly combined with information about camera parameters allows for Euclidean reconstruction. We propose an elegant formalism to incorporate such information, in which camera parameters are dual to parallelepiped parameters, i.e. any knowledge about one entity provides constraints on the parameters of the others. Consequently, the image of a known parallelepiped defines the camera parameters, and reciprocally, a calibrated image of a parallelepiped defines its Euclidean shape (up to its size). In this paper, we synthesize our work on parallelepipeds [53, 54] and propose more elegant and efficient approaches.

The cameras and parallelepipeds parameters are recovered in two steps. Firstly, the factorization based approach is used to compute their intrinsic and orientation parameters. Interestingly, the use of the three-dimensional structure allows to deal very easily with the classical problems of the factorization methods: the unknown scale factors and the missing data. Then the least square optimization is used to recover simultaneously the scale and position parameters in the common euclidean frame. The use of the well-constrained calibration primitives allows to obtain good calibration results even from very small number of images, however, depending on the information about the scene, the singularities might occur. They are collected into a detailed dictionary, accompanied by the sketch of the methodology used for its construction.

Our calibration approach is conceptually close to self-calibration methods, especially to methods for upgrading affine to Euclidean structure [18, 33] or methods that consider special camera motions [17, 10, 2]. The way the metric information of a parallelepiped is used, is also similar to the vanishing point based methods [7, 9, 8, 24]. Some properties of our algorithm are also common with plane-based approaches [42, 57, 49, 27, 38, 41]. While more flexible than standard calibration techniques, homography-based approaches still require either Euclidean information or, for self-calibration, many images in general position [48], or at least one plane visible in all images [37]. In this sense, our approach is a generalization of plane-based methods with metric information, to three-dimensional parallelepipedic patterns. This allows to handle missing data and unknown scale factors and simplifies the formulation of calibration constraints.

While the main contributions of the paper concern the estimation of camera and parallelepiped parameters, we also propose a method for enhancing reconstructions with primitives other than parallelepipeds. The complete system allows for both calibration and 3D model acquisition from a small number of arbitrary images, and this with a reasonable amount of user interaction.

The paper is organized as follows. Section 2 gives definitions and some background. Section 3 introduces the concept of camera–parallelepiped duality. Calibration using parallelepipeds and a study on the singular configurations are described in sections 4 and 5.

Sections 6 and 7 describe our approaches for pose estimation and 3D reconstruction. Experimental results are then presented in section 8 before concluding.

2 Preliminaries

2.1 Camera parameterization

We represent cameras using the pinhole model. The projection from a 3D point $\mathbf{P}$ to its 2D image point $\mathbf{p}$ is expressed by: $\mathbf{p} \sim \mathsf{M}\mathbf{P}$, where M is a 3×4 matrix, which can be decomposed as:

$$\mathsf{M} = \mathsf{K}\begin{pmatrix}\mathsf{R} & \mathbf{t}\end{pmatrix}$$

The 3×4 matrix $\begin{pmatrix}\mathsf{R} & \mathbf{t}\end{pmatrix}$ encapsulates the camera's pose in the world coordinate system, or its extrinsic parameters: the rotation matrix R represents its orientation and the vector $\mathbf{t}$ its position. The matrix K is the 3×3 calibration matrix containing the camera's intrinsic parameters:

$$\mathsf{K} = \begin{pmatrix}\alpha_u & s & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1\end{pmatrix}, \tag{1}$$

where α_u and α_v stand for the focal length, expressed in horizontal and vertical pixel dimensions, s is a skew parameter considered here as equal to zero, and (u_0, v_0) are the pixel coordinates of the principal point. The notation $\tau = \frac{\alpha_u}{\alpha_v}$ will be used for the camera aspect ratio. The term camera axes will be used for axes of the coordinate system attached to the camera projection center, where two axes are parallel to pixel edges and the third one is orthogonal to the image plane (the viewing axe). In the following, we will also use the IAC (image of the absolute conic) representation of the intrinsic parameters, namely the matrix $\omega \sim \mathsf{K}^{-\mathsf{T}}\mathsf{K}^{-1}$:

$$\omega \sim \begin{pmatrix}1 & 0 & -u_0 \\ 0 & \tau^2 & -\tau^2 v_0 \\ -u_0 & -\tau^2 v_0 & \tau^2\alpha_v^2 + u_0^2 + \tau^2 v_0^2\end{pmatrix} . \tag{2}$$

2.2 Parallelepiped parameterization

A parallelepiped is defined by twelve parameters: six extrinsic parameters describing its orientation and position, and six intrinsic parameters describing its Euclidean shape: three dimension parameters (edge lengths l_1, l_2 and l_3) and three angles between edges $(\theta_{12}, \theta_{23}, \theta_{13})$. These intrinsic parameters are illustrated in figure 1. The parallelepiped may be represented compactly in matrix form by a 4×4 matrix N:

$$\mathsf{N} = \begin{pmatrix}\mathsf{S} & \mathbf{v} \\ \mathbf{0}^{\mathsf{T}} & 1\end{pmatrix}\tilde{\mathsf{L}}$$

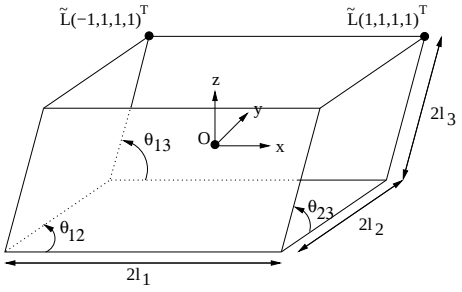

Figure 1: Parameterization of a parallelepiped: $2l_i$ are the edge lengths; θ_{ij} are the angles between non-parallel edges.

where S is a rotation matrix and $\mathbf{v}$ a vector, representing the parallelepiped's pose (extrinsic parameters). The 4×4 matrix $\tilde{\mathsf{L}}$ represents the parallelepiped's shape:

$$\tilde{\mathsf{L}} = \begin{pmatrix} l_1 & l_2 c_{12} & l_3 c_{13} & 0 \\ 0 & l_2 s_{12} & l_3 \frac{c_{23} - c_{13} c_{12}}{s_{12}} & 0 \\ 0 & 0 & l_3 \sqrt{\frac{s_{12}^2 - c_{13}^2 s_{12}^2 - (c_{23} - c_{13} c_{12})^2}{s_{12}^2}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

with: $c_{ij} = \cos\theta_{ij}$, $s_{ij} = \sin\theta_{ij}$, $\theta_{ij} \in \ ]0\ \pi[$, $l_i > 0$.

The matrix $\tilde{\mathsf{L}}$ represents the affine transformation between a canonic cube and a parallelepiped with the given shape. Concretely, a vertex $(\pm 1, \pm 1, \pm 1, 1)^\mathsf{T}$ of the canonic cube is mapped, by $\tilde{\mathsf{L}}$, to a vertex of our parallelepiped's intrinsic shape. Then, the pose part of N maps the vertices into the world coordinate system.

Other parameterizations for $\tilde{\mathsf{L}}$ may be chosen, but the above one is attractive due to its upper triangular form. This underlines the fact that $\tilde{\mathsf{L}}$ plays the same role for the parallelepiped as the calibration matrix K for a camera.

The analogous entity to a camera's IAC ω, is the matrix μ, defined by:

$$\mu \sim \mathsf{L}^\mathsf{T}\mathsf{L} \sim \begin{pmatrix} l_1^2 & l_1 l_2 \cos\theta_{12} & l_1 l_3 \cos\theta_{13} \\ l_1 l_2 \cos\theta_{12} & l_2^2 & l_2 l_3 \cos\theta_{23} \\ l_1 l_3 \cos\theta_{13} & l_2 l_3 \cos\theta_{23} & l_3^2 \end{pmatrix}, \tag{3}$$

where L is the upper left 3×3 matrix of $\tilde{\mathsf{L}}$.

Hence, there is a seemingly perfect symmetry between intrinsic parameters of cameras and parallelepipeds. The only difference is that in some cases, the *size* of a parallelepiped matters, as will be explained in the following. As for cameras, the fact that $K_{33} = 1$ allows to fix the scale factor in the relation $\omega \sim \mathsf{K}^{-\mathsf{T}}\mathsf{K}^{-1}$, and thus to extract K uniquely from the IAC ω, e.g. using Cholesky decomposition. As for parallelepipeds however, we have no such constraint on its "calibration matrix" L, so the relation $\mu \sim \mathsf{L}^\mathsf{T}\mathsf{L}$ gives us a parallelepiped's

Euclidean shape, but not its (absolute) size. This does not matter in general, since we are usually only interested in reconstructing a scene up to some scale. However, when reconstructing several parallelepipeds, one needs to recover at least their *relative* sizes.

There are many possibilities to define the size of parallelepipeds. We choose the following definition, due to its appropriateness in the equations underlying our calibration and reconstruction algorithms below: the **size** of a parallelepiped is defined as

$$s = (\det \mathsf{L})^{1/3} \quad .$$

This definition is actually directly linked to the parallelepiped's volume: $s^3 = \det \mathsf{L} = \mathrm{Vol}/8$ (the factor 8 arises since our canonic cube has an edge length of 2).

3 Projections of Parallelepipeds

3.1 One Parallelepiped in A Single View

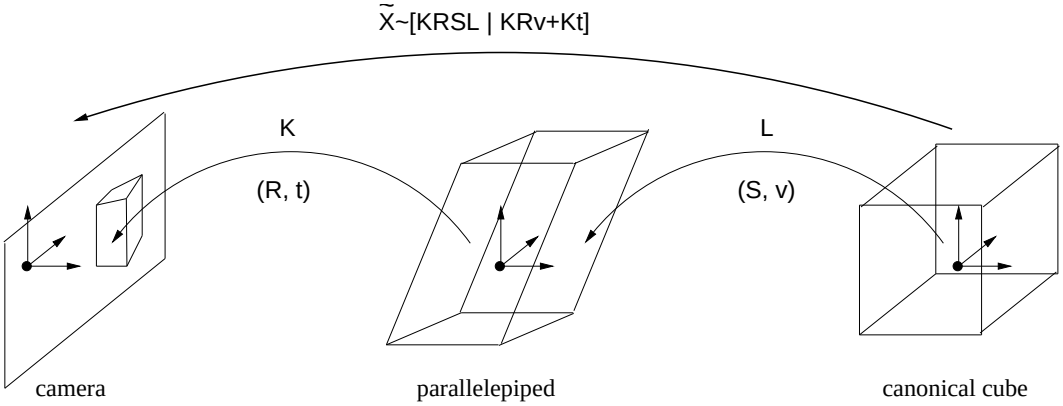

Figure 2: The projection of the canonic parallelepiped (cube) into the image. Matrices K, L correspond to intrinsic parameters of camera and parallelepiped and $(\mathsf{R}, \mathbf{t})$, $(\mathsf{S}, \mathbf{v})$ correspond to extrinsic parameters of camera and parallelepiped, respectively.

In this section, we introduce the concept of duality between the intrinsic characteristics of a camera and those of a parallelepiped.

To do so, consider the projection of the parallelepiped's vertices into the camera. Let $\mathbf{C}_i, i = 1..8$ be the homogeneous coordinates of the canonic cube's vertices. As described in section 2.2, the corresponding vertex of the parallelepiped is given as:

$$\mathbf{P}_i = \mathsf{N}\mathbf{C}_i = \begin{pmatrix} \mathsf{S} & \mathbf{v} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \tilde{\mathsf{L}}\mathbf{C}_i$$

and its image point is:

$$\mathbf{p}_i \sim \mathsf{M}\mathbf{P}_i = \mathsf{K} \begin{pmatrix} \mathsf{R} & \mathbf{t} \end{pmatrix} \begin{pmatrix} \mathsf{S} & \mathbf{v} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \tilde{\mathsf{L}}\mathbf{C}_i \ . \tag{4}$$

In the above equation, we define the *canonic projection matrix*:

$$\tilde{\mathsf{X}} \sim \mathsf{K} \begin{pmatrix} \mathsf{R} & \mathbf{t} \end{pmatrix} \begin{pmatrix} \mathsf{S} & \mathbf{v} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \tilde{\mathsf{L}} \ . \tag{5}$$

This matrix represents a perspective projection that maps the vertices of the canonic cube onto the image points of the parallelepiped's vertices. This is illustrated on the figure 2. Given image points for sufficiently many vertices[1], the canonic projection matrix

[1] Five image points and one image direction are in general sufficient. Additional points make the computation more stable.

can be computed, even in the absence of prior knowledge on intrinsic or extrinsic parameters. Our calibration and pose estimation algorithms are based on the link between the canonic projection matrix (which we suppose given from now on) and the camera's and parallelepiped's intrinsic and extrinsic parameters.

Let us consider this in more detail. First, we may identify the *relative pose* between camera and parallelepiped in (5), represented by the following 3×4 matrix:

$$(\mathsf{R} \quad \mathbf{t}) \begin{pmatrix} \mathsf{S} & \mathbf{v} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} = (\mathsf{RS} \qquad \mathsf{R}\mathbf{v} + \mathbf{t})$$

Second, let us consider the leading 3×3 sub-matrix X of the canonic projection matrix $\tilde{\mathsf{X}}$, which is given by:

$$\mathsf{X} \sim \mathsf{K}\,(\mathsf{RS})\,\mathsf{L}. \tag{6}$$

Due to the orthogonality of the rotation matrices R and S, it is simple to derive the following relation between the camera's IAC ω and the corresponding entity μ of the parallelepiped:

$$\mathsf{X}^\mathsf{T} \omega \mathsf{X} \sim \mu. \tag{7}$$

This equation establishes an interesting duality between the intrinsic parameters of a camera and those of a parallelepiped. It shows (unsurprisingly) that knowing the parallelepiped's shape μ allows to calibrate the camera. Conversely, knowing the camera's intrinsic parameters allows to directly compute the parallelepiped's Euclidian shape, also from a single image.

In the next sections, we generalize the use of this duality for calibration and pose estimation to the case of multiple parallelepipeds seen in multiple cameras and to the use of *partial* knowledge about the camera's or parallelepiped's intrinsic parameters. Before doing so, let us describe a few interesting links between our and other (self-) calibration scenarios.

Classical self-calibration proceeds usually in two main steps: first, a projective 3D reconstruction of the scene is obtained from correspondences across two or more images. Then, the projective reconstruction is transformed to a Euclidean one using the available prior knowledge on intrinsic parameters. This upgrade is sometimes interlaced by an intermediate upgrade to an affine reconstruction.

In our scenario, we have a 3D reconstruction of the scene already from a *single* rather than multiple images, which is furthermore of *affine* rather than projective nature: we know that the observed parallelepiped's shape is that of a cube, up to some affine transformation. Analogously, our canonic projection matrix is equal to the true one up to an affine transformation. Hence, self-calibration in our scenario does not need to recover the plane at infinity, which is known to be the hardest ("most non-linear") part of classical self-calibration. Indeed, our calibration method is somewhat similar to the affine-to-Euclidean upgrade of stratified self-calibration approaches, e.g. [18, 33].

Similarities also exist with (self-) calibration approaches based on special camera motions: calibrating a rotating camera [17, 10] is more or less equivalent to self-calibrating a camera in general motion once affine structure is known. Other approaches recover the affine structure

by first performing pure translations and then general motions or to approaches that consider special camera motions [2, 34].

Our scenario is similar to these. In the following sections we show how it allows to efficiently combine the usual self-calibration constraints with constraints on scene structure. This enables to perform calibration (and 3D reconstruction) from very few images; one image may actually be sufficient.

3.2 n Parallelepipeds in m Views

The main motivation for the work described in this paper is to generalize the use of the duality introduce in the previous section: we consider the general case where multiple parallelepipeds are seen by multiple cameras (not all parallelepipeds need to be seen by all cameras). Furthermore, we do not in general suppose that some cameras or parallelepipeds are fully calibrated. We rather want to make efficient and complete use of any kind of partial calibration information. As for the cameras, this amounts to partial knowledge on their intrinsic parameters that is routinely used for self-calibration. As for parallelepipeds, we rather consider them as "vehicles" to jointly express simple yet useful geometric scene constraints. Defining orthogonality or parallelism constraints between for example "only" pairs of lines, amounts to providing information about the structure of individual *planes* in the scene. Parallelepipeds however allow to directly express richer couplings of constraints on *3D* scene structure.

Let us now consider the general case where n parallelepipeds are seen by m cameras. Let $\tilde{\mathsf{X}}_{ik}$ be the canonic projection matrix associated with the projection of the kth parallelepiped in the ith camera:

$$\tilde{\mathsf{X}}_{ik} \sim \mathsf{K}_i \begin{pmatrix} \mathsf{R}_i & \mathbf{t}_i \end{pmatrix} \begin{pmatrix} \mathsf{S}_k & \mathbf{v}_k \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \tilde{\mathsf{L}}_k$$

Let us explicitly introduce scale factors λ_{ik} such that the equality up to scale in the above equation can be turned into a component-wise equality:

$$\lambda_{ik}\tilde{\mathsf{X}}_{ik} = \mathsf{K}_i \begin{pmatrix} \mathsf{R}_i & \mathbf{t}_i \end{pmatrix} \begin{pmatrix} \mathsf{S}_k & \mathbf{v}_k \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \tilde{\mathsf{L}}_k \tag{8}$$

We may group together these equations for all m cameras and n parallelepipeds, into the following single matrix equation:

$$\underbrace{\begin{bmatrix} \lambda_{11}\tilde{\mathsf{X}}_{11} & \cdots & \lambda_{1n}\tilde{\mathsf{X}}_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{m1}\tilde{\mathsf{X}}_{m1} & \cdots & \lambda_{mn}\tilde{\mathsf{X}}_{mn} \end{bmatrix}}_{\mathcal{X}_{3m\times 4n}} =$$

$$\underbrace{\begin{bmatrix} \mathsf{K}_1\begin{pmatrix}\mathsf{R}_1 & \mathbf{t}_1\end{pmatrix} \\ \vdots \\ \mathsf{K}_m\begin{pmatrix}\mathsf{R}_m & \mathbf{t}_m\end{pmatrix} \end{bmatrix}}_{\mathcal{M}_{3m\times 4}} \underbrace{\begin{bmatrix} \begin{pmatrix}\mathsf{S}_1 & \mathbf{v}_1 \\ \mathbf{0}^\mathsf{T} & 1\end{pmatrix}\tilde{\mathsf{L}}_1 & \cdots & \begin{pmatrix}\mathsf{S}_n & \mathbf{v}_n \\ \mathbf{0}^\mathsf{T} & 1\end{pmatrix}\tilde{\mathsf{L}}_n \end{bmatrix}}_{\mathcal{S}_{4\times 4n}} \tag{9}$$

This equation naturally leads to the idea of a factorization-based calibration algorithm, which will be developed in section 4. It is based on the following observation. The matrix $\mathcal{X}$ contains all information that can be recovered from the parallelepipeds' image points alone (below, we discuss the issue of computing the scale factors λ_{ik}). In analogy with [46], we call it the *measurement matrix*. Since the measurement matrix is the product of a "motion matrix" $\mathcal{M}$ of 4 columns, with a "shape matrix" $\mathcal{S}$ of 4 rows, its rank can be 4 at most (in the absence of noise).

We might aim at extracting intrinsic and extrinsic parameters of cameras and parallelepipeds directly from a rank-4-factorization of $\mathcal{X}$. One step of many factorization methods for structure and motion recovery is to disambiguate the result of the factorization: in general, for a rank-r-factorization, motion and shape are recovered up to a transformation represented by an $r \times r$ matrix (in our case, this would be a 3D projective transformation). The ambiguity can be reduced using e.g. constraints on intrinsic camera parameters (see more details in section 4). In our case, we observe that the 4×4 sub-blocks of the shape matrix $\mathcal{S}$ are affine transformations (last row consists of three zeroes and a one). We would have to include this constraint into the disambiguation, but nevertheless, the result would not in general exactly satisfy the affine form of these sub-blocks. We thus cut the problem in two steps, which allows to easily guarantee that the sub-blocks of the shape matrix be affine transformations. In the first step (section 4), we consider a "reduced measurement matrix" consisting of the leading 3×3 sub-matrices of the $\tilde{\mathsf{X}}_{ik}$. We extract *intrinsic parameters* and *orientation* of our cameras and parallelepipeds based on a rank-3-factorization and a disambiguation stage using calibration and scene constraints. In the second step (section 6), we then estimate the *position* of cameras and parallelepipeds, as well as the parallelepipeds' *size*.

Just as a sidenote, we observe that, for two views i and j, and a parallelepiped k, the infinite homography between the two views is given by the product $\mathsf{X}_{ik}\mathsf{X}_{jk}^{-1}$.

4 Estimating Intrinsic and Orientation Parameters by Factorization

In this section we concentrate on the computation of the cameras' and parallelepipeds' intrinsic parameters and orientation (rotation), based on equation (9) and the observations concerning it, cf. the previous section. As mentioned previously, we first restrict our attention to the leading 3×3 submatrices of the $\tilde{\mathsf{X}}_{ik}$, like we did in section 3.1 for the establishment of the duality between intrinsic parameters of cameras and parallelepipeds. We thus deal with the corresponding subpart of equation (9):

$$\underbrace{\begin{bmatrix} \lambda_{11}\mathsf{X}_{11} & \cdots & \lambda_{1n}\mathsf{X}_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{m1}\mathsf{X}_{m1} & \cdots & \lambda_{mn}\mathsf{X}_{mn} \end{bmatrix}}_{\mathcal{X}'_{3m\times 3n}} = \underbrace{\begin{bmatrix} \mathsf{K}_1\mathsf{R}_1 \\ \vdots \\ \mathsf{K}_m\mathsf{R}_m \end{bmatrix}}_{\mathcal{M}'_{3m\times 3}} \underbrace{\begin{bmatrix} \mathsf{S}_1\mathsf{L}_1 & \cdots & \mathsf{S}_n\mathsf{L}_n \end{bmatrix}}_{\mathcal{S}'_{3\times 3n}} \tag{10}$$

In the following sections, we describe the different steps of our factorization-based method. We first deal with the important problem of missing data. Then it is described how the scale factors λ_{ik} needed to construct the measurement matrix $\mathcal{X}'$, are computed. The factorization itself is described in section 4.3, followed by the most important aspect: how to disambiguate the factorization's result in order to extract intrinsic and orientation parameters. A summary of these steps and a discussion of minimal cases and singularities is provided at the end of this section. The computation of position parameters and parallelepiped size is dealt with in section 6.

4.1 Missing Data

As with all factorization problems, our method might suffer from the problem of missing data, i.e. missing X_{ik}. Indeed, in practice, the condition that all parallelepipeds are seen in all views can not always be satisfied. However, each missing matrix X_{ik} can be deduced from others if there is one camera j and one parallelepiped l such that the transformations X_{jl}, X_{jk} and X_{il} are known. The missing matrix can then be computed using:

$$\mathsf{X}_{ik} \sim \mathsf{X}_{il}\ (\mathsf{X}_{jl})^{-1}\ \mathsf{X}_{jk}. \tag{11}$$

Several equations of this type may be used simultaneously to increase the accuracy. In that case, care has to be taken since equation (11) is defined up to scale only. This problem can be circumvented very simply though, by normalizing all X_{ik} to unit determinant.

These observations motivate a simple recursive method[2] to compute missing matrices X_{ik}: at each iteration, we compute the one for which most equations of type (11) are available. Previously computed matrices X_{ik} can be involved at every successive iteration of this procedure.

[2] Compare with the analogous method in [41].

4.2 Recovery of Scale Factors

The reduced measurement matrix $\mathcal{X}'$ in (10) is, in the absence of noise, of rank 3, being the product of a matrix of 3 columns and a matrix of 3 rows. This however only holds if a correct set of scale factors λ_{ik} is used. For other problems, these are often non trivial to compute, see e.g. [27, 44, 56]. In our case however, this turns out to be rather simple.

Let us first write $\mathsf{A}_i = \mathsf{K}_i\mathsf{R}_i$ and $\mathsf{B}_k = \mathsf{S}_k\mathsf{L}_k$. What we know is that (in the absence of noise), there exist matrices $\mathsf{A}_i, i = 1..m$ and $\mathsf{B}_k, k = 1..n$ such that:

$$\forall i, k : \mathsf{X}_{ik} \sim \mathsf{A}_i\mathsf{B}_k$$

Since this equation is valid up to scale only, we also have:

$$\forall i, k : \mathsf{X}_{ik} \sim (a_i\mathsf{A}_i)\,(b_k\mathsf{B}_k)$$

for any non-zero scale factors $a_i, i = 1..m$ and $b_k, k = 1..n$. Consequently, this is also true for the scale factors with:

$$\begin{aligned} \det(a_i\mathsf{A}_i) &= 1 \\ \det(b_k\mathsf{B}_k) &= 1 \end{aligned}$$

Note that we do not need to know these scale factors; it is sufficient to know they exist. Hence:

$$\forall i, k : \mathsf{X}_{ik} \sim a_i b_k \mathsf{A}_i\mathsf{B}_k,$$

with $\det(a_i b_k \mathsf{A}_i\mathsf{B}_k) = \det(a_i\mathsf{A}_i)\det(b_k\mathsf{B}_k) = 1$.

To achieve a component-wise equality $\lambda_{ik}\mathsf{X}_{ik} = (a_i\mathsf{A}_i)(b_k\mathsf{B}_k)$, we need to use scale factors[3] λ_{ik} such that $\det(\lambda_{ik}\mathsf{X}_{ik}) = 1$. Hence:

$$\lambda_{ik} = (\det \mathsf{X}_{ik})^{-1/3}$$

In the following, we assume that the X_{ik} are already normalized to unit determinant, i.e. that $\lambda_{ik} = 1$. Equation (10) becomes:

$$\underbrace{\begin{bmatrix} \mathsf{X}_{11} & \cdots & \mathsf{X}_{1n} \\ \vdots & \ddots & \vdots \\ \mathsf{X}_{m1} & \cdots & \mathsf{X}_{mn} \end{bmatrix}}_{\mathcal{X}'_{3m\times 3n}} = \underbrace{\begin{bmatrix} a_1\mathsf{K}_1\mathsf{R}_1 \\ \vdots \\ a_m\mathsf{K}_m\mathsf{R}_m \end{bmatrix}}_{\mathcal{M}'_{3m\times 3}} \underbrace{\begin{bmatrix} b_1\mathsf{S}_1\mathsf{L}_1 & \cdots & b_n\mathsf{S}_n\mathsf{L}_n \end{bmatrix}}_{\mathcal{S}'_{3\times 3n}} \tag{12}$$

The scale factors a_i and b_k do not matter for now; all that counts is that the measurement matrix $\mathcal{X}'$ containing the normalized X_{ik}, is of rank 3 at most, and can thus be factorized as shown below.

[3] It is well known that two non-singular 3×3 matrices that are equal up to scale and whose determinants are equal, are also equal component-wise.

4.3 Factorization

As usual, we use the SVD (Singular Value Decomposition) to obtain the low-rank factorization of the measurement matrix. Let the SVD of $\mathcal{X}'$ be given as:

$$\mathcal{X}'_{3m\times 3n} = \mathsf{U}_{3m\times 3n}\Sigma_{3n\times 3n}\mathsf{V}^{\mathsf{T}}_{3n\times 3n}$$

The diagonal matrix Σ contains the singular values of $\mathcal{X}'$. Let them be ordered: $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{3n}$. In the absence of noise, $\mathcal{X}'$ is of rank 3 at most and $\sigma_4 = \cdots = \sigma_{3n} = 0$. If noise is present, $\mathcal{X}'$ is of full rank in general. Setting all singular values to zero, besides the three largest ones, leads to the best rank-3 approximation of $\mathcal{X}'$ (in the sense of the Frobenius norm [36]).

In the following, we consider the decomposition of the rank-3 approximation of $\mathcal{X}'$ (for ease of notation, we denote this also as $\mathcal{X}'$):

$$\mathcal{X}' = \mathsf{U}_{3m\times 3n}\ \ \mathrm{diag}(\sigma_1,\sigma_2,\sigma_3,0,\ldots,0)\ \ \mathsf{V}^{\mathsf{T}}_{3n\times 3n}$$

In the matrix product on the right, only columns of U and rows of V^{T} that correspond to non-zero singular values, contribute. Hence:

$$\mathcal{X}' = \mathsf{U}'_{3m\times 3}\begin{pmatrix}\sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3\end{pmatrix}\left(\mathsf{V}'^{\mathsf{T}}\right)_{3\times 3n}$$

where U' (resp. V') consists of the first three columns of U (resp. V). Let us define

$$\begin{aligned}\mathsf{U}'' &= \mathsf{U}'\begin{pmatrix}\sqrt{\sigma_1} & & \\ & \sqrt{\sigma_2} & \\ & & \sqrt{\sigma_3}\end{pmatrix}\\ \mathsf{V}'' &= \mathsf{V}'\begin{pmatrix}\sqrt{\sigma_1} & & \\ & \sqrt{\sigma_2} & \\ & & \sqrt{\sigma_3}\end{pmatrix}\end{aligned}$$

Thus:

$$\mathcal{X}' = \mathsf{U}''\mathsf{V}''^{\mathsf{T}}$$

This represents a decomposition of the measurement matrix $\mathcal{X}'$ into a product of a matrix of 3 columns with a matrix of 3 rows. Note however, that this decomposition is not unique. For any non-singular 3×3 matrix T, the following is also a valid decomposition:

$$\mathcal{X}' = \left(\mathsf{U}''\mathsf{T}^{-1}\right)\left(\mathsf{T}\mathsf{V}''^{\mathsf{T}}\right)$$

Making the link with equation (12), we obtain:

$$\begin{bmatrix}a_1\mathsf{K}_1\mathsf{R}_1\\ \vdots\\ a_m\mathsf{K}_m\mathsf{R}_m\end{bmatrix}\begin{bmatrix}b_1\mathsf{S}_1\mathsf{L}_1 & \cdots & b_n\mathsf{S}_n\mathsf{L}_n\end{bmatrix} = \left(\mathsf{U}''\mathsf{T}^{-1}\right)\left(\mathsf{T}\mathsf{V}''^{\mathsf{T}}\right) \qquad (13)$$

Let us write U'' and V'' as follows as a composition of 3×3 submatrices:

$$\mathsf{U}'' = \begin{bmatrix} \mathsf{U}_1 \\ \vdots \\ \mathsf{U}_m \end{bmatrix} \qquad \mathsf{V}'' = \begin{bmatrix} \mathsf{V}_1 \\ \vdots \\ \mathsf{V}_n \end{bmatrix}$$

Equation (13) thus becomes:

$$\begin{bmatrix} a_1\mathsf{K}_1\mathsf{R}_1 \\ \vdots \\ a_m\mathsf{K}_m\mathsf{R}_m \end{bmatrix} \begin{bmatrix} b_1\mathsf{S}_1\mathsf{L}_1 \cdots b_n\mathsf{S}_n\mathsf{L}_n \end{bmatrix} = \begin{bmatrix} \mathsf{U}_1\mathsf{T}^{-1} \\ \vdots \\ \mathsf{U}_m\mathsf{T}^{-1} \end{bmatrix} \begin{bmatrix} \mathsf{T}\mathsf{V}_1^\mathsf{T} \cdots \mathsf{T}\mathsf{V}_n^\mathsf{T} \end{bmatrix} \tag{14}$$

How to estimate T is explained in section 4.4. Once a correct estimate is given, we can directly extract the matrices $\mathsf{A}_i = a_i\mathsf{K}_i\mathsf{R}_i$ and $\mathsf{B}_k = b_k\mathsf{S}_k\mathsf{L}_k$, from which in turn the individual rotation matrices and calibration matrices can be recovered using a Cholesky or QR-decomposition. The Cholesky decomposition of $\mathsf{A}_i\mathsf{A}_i^\mathsf{T}$ for example, results in an upper triangular matrix $\mathsf{M}_i = a_i\mathsf{K}_i$. Based on the requirement $K_{i,33} = 1$, we can compute the unknown scale factor a_i as $a_i = M_{i,33}$. The calibration matrix is finally obtained as $\mathsf{K}_i = \frac{1}{a_i}\mathsf{M}_i$.

As for the parallelepipeds, we do not have any constraint on the entries of their calibration matrices L_k. Hence, we can compute them only up to the unknown scale factors b_k. This means that we can compute the *shape* of each parallelepiped, but not (yet) their *size* (or, volume). In section 6, we explain how to compute their (relative) size.

We now briefly discuss the structure and geometric signification of the matrix T. Note that T actually represents the non-translational part of a 3D affine transformation (its upper left 3×3 submarix). This is just another expression of the previously mentioned fact that due to the observation of parallelepipeds, we directly have an affine reconstruction (of scene and cameras).

The matrix T can only be computed up to an arbitrary rotation and scale: for any rotation matrix R and scale factor s, $\mathsf{T}' = s\mathsf{R}\mathsf{T}$ can not be distinguished from T in the factorization since $\mathsf{T}'^{-1}\mathsf{T}' \sim \mathsf{T}^{-1}\mathsf{T}$. This ambiguity is natural and expresses the fact that the global Euclidean reference frame for the reconstruction of our parallelepipeds and cameras can be chosen arbitrarily. Without loss of generality, we may thus assume that T is upper triangular. This highlights the fact that our estimation problem has only 5 degrees of freedom (6 parameters for an upper triangular 3×3 matrix minus one for the arbitrary scale) which can also be explained in more geometric terms: as explained previously, our problem is somewhat equivalent to self-calibration with known affine structure. The 5 degrees of the problem may be interpreted as the coefficients of the absolute conic on the plane at infinity.

4.4 Disambiguating the Factorization

We now deal with the estimation of the unknown transformation T appearing in equation (14). As will be seen below, and as is often the case in self-calibration problems, it is

simpler to not directly estimate T, but the symmetric and positive definite 3×3 matrix Z defined as:

$$\mathsf{Z} = \mathsf{T}^\mathsf{T}\,\mathsf{T} \tag{15}$$

We may observe that this matrix represents the absolute conic on the plane at infinity. Once Z is estimated, T may be extracted from it using Cholesky decomposition. As described above, T is defined up to a rotation and scale, so the upper triangular Cholesky factor of Z can directly be used as the estimate for T.

The matrix Z (and thus T), can be estimated in various ways, using any information about the cameras or the parallelepipeds, e.g. prior knowledge on relative positioning of some entities. Here, we concentrate on exploiting prior information on intrinsic parameters, of both, cameras and parallelepipeds. There are two types of information that we consider:

- knowledge of the actual value of some intrinsic parameter for some camera or parallelepiped.
- knowledge that two or more cameras (or parallelepipeds) have the same value for some intrinsic parameter. We also sometimes speak of "constant" intrinsic parameters.

We first describe the use of prior knowledge on camera intrinsics.

4.4.1 Using Knowledge on Camera Intrinsics

From equation (14), we have:

$$a_i \mathsf{K}_i \mathsf{R}_i = \mathsf{U}_i \mathsf{T}^{-1}$$

Due to the orthogonality of R_i, we get:

$$a_i^2 \underbrace{\mathsf{K}_i \mathsf{K}_i^\mathsf{T}}_{\omega_i^{-1}} = \mathsf{U}_i \underbrace{\mathsf{T}^{-1}\mathsf{T}^{-\mathsf{T}}}_{\mathsf{Z}^{-1}} \mathsf{U}_i^\mathsf{T}$$

Neglecting the unknown scale factor a_i and taking the inverse of both sides of the equation, we obtain (note that the U_i are not orthogonal in general):

$$\omega_i \sim \mathsf{U}_i^{-\mathsf{T}} \mathsf{Z} \mathsf{U}_i^{-1}. \tag{16}$$

We are now ready to formulate constraints on Z based on prior knowledge on the cameras' intrinsics.

Known values of camera intrinsics Knowing respectively the aspect ratio and principal point coordinates of a camera i gives the following constraints on its IAC ω_i (based on equation (2)):

$$\begin{aligned} \tau_i^2 \omega_{i,11} - \omega_{i,22} &= 0 \\ u_{i,0}\omega_{i,11} + \omega_{i,13} &= 0 \\ v_{i,0}\omega_{i,22} + \omega_{i,23} &= 0 \end{aligned}$$

A known value of the focal length α_v can only be used to formulate linear equations if the other intrinsics are also known [42]. In such a fully calibrated case, other algorithms might be better suited, so we neglect that case in the following.

By substituting ω_i in the above equations according to (16), we get the following *linear* equations on Z:

$$\begin{aligned} \tau_i^2 \left(\mathsf{U}_i^{-\mathsf{T}}\mathsf{Z}\mathsf{U}_i^{-1}\right)_{11} - \left(\mathsf{U}_i^{-\mathsf{T}}\mathsf{Z}\mathsf{U}_i^{-1}\right)_{22} &= 0 \\ u_{i,0} \left(\mathsf{U}_i^{-\mathsf{T}}\mathsf{Z}\mathsf{U}_i^{-1}\right)_{11} + \left(\mathsf{U}_i^{-\mathsf{T}}\mathsf{Z}\mathsf{U}_i^{-1}\right)_{13} &= 0 \\ v_{i,0} \left(\mathsf{U}_i^{-\mathsf{T}}\mathsf{Z}\mathsf{U}_i^{-1}\right)_{22} + \left(\mathsf{U}_i^{-\mathsf{T}}\mathsf{Z}\mathsf{U}_i^{-1}\right)_{23} &= 0 \end{aligned}$$

Constant camera intrinsics In the case that two cameras i and j are known to have the same, yet unknown value for one intrinsic parameter, we in general obtain *quadratic* equations on Z. For example, the assumption of equal aspect ratios leads to the equation:

$$\left(\mathsf{U}_i^{-\mathsf{T}}\mathsf{Z}\mathsf{U}_i^{-1}\right)_{11}\left(\mathsf{U}_j^{-\mathsf{T}}\mathsf{Z}\mathsf{U}_j^{-1}\right)_{22} = \left(\mathsf{U}_j^{-\mathsf{T}}\mathsf{Z}\mathsf{U}_j^{-1}\right)_{11}\left(\mathsf{U}_i^{-\mathsf{T}}\mathsf{Z}\mathsf{U}_i^{-1}\right)_{22}.$$

In practice, we only use available linear equations. In some minimal cases, quadratic equations as above might be useful to find a unique solution or a finite set of solutions, if the available linear constraints are insufficient.

The situation is different if *all* intrinsic parameters of two (or more) views are known to be identical. In that case, we can obtain linear equations instead of quadratic ones, as shown in [17]: the matrices U^i are scaled such as to have unit determinant. Then we can write the following component-wise matrix equality between any pair (i, j) of views:

$$\mathsf{U}_i^{-\mathsf{T}}\mathsf{Z}\mathsf{U}_i^{-1} - \mathsf{U}_j^{-\mathsf{T}}\mathsf{Z}\mathsf{U}_j^{-1} = \mathsf{0}_{3\times 3}$$

This represents 6 *linear* equations on Z for each pair of views, among which 4 are independent.

4.4.2 Prior Information on Parallelepipeds

From equation (14), we have:

$$b_k \mathsf{S}_k \mathsf{L}_k = \mathsf{T}\mathsf{V}_k^{\mathsf{T}}$$

Due to the orthogonality of S_k, we get:

$$b_k^2 \underbrace{\mathsf{L}_k^{\mathsf{T}}\mathsf{L}_k}_{\mu_k} = \mathsf{V}_k \underbrace{\mathsf{T}^{\mathsf{T}}\mathsf{T}}_{\mathsf{Z}} \mathsf{V}_k^{\mathsf{T}}$$

Neglecting the unknown scale factor b_k, we obtain:

$$\mu_k \sim \mathsf{V}_k \mathsf{Z} \mathsf{V}_k^{\mathsf{T}}.$$

Knowledge on parallelepiped intrinsics can be used in an analogous way as knowledge about camera parameters. For example suppose we know the length ratio of two parallelepiped edges $r_{uv} = \frac{l_u}{l_v}$. Referring to (3), we get the following *linear* equation on Z:

$$r_{k,uv}^2 \mu_{k,vv} - \mu_{k,uu} = r_{k,uv}^2 \left(\mathsf{V}_k \mathsf{Z} \mathsf{V}_k^\mathsf{T}\right)_{vv} - \left(\mathsf{V}_k \mathsf{Z} \mathsf{V}_k^\mathsf{T}\right)_{uu} = 0.$$

Similarly, the assumption that the θ_{uv} is a right angle, i.e. $\cos\theta_{uv} = 0$, leads also to a *linear equation*:

$$\mu_{k,uv} = \left(\mathsf{V}_k \mathsf{Z} \mathsf{V}_k^\mathsf{T}\right)_{uv} = 0.$$

As for cameras, *quadratic* equations may be derived from assumptions about two or more parallelepiped having the same, yet unknown value for some intrinsic parameter. Furthermore, two parallelepipeds having the same shape, leads to a set of *linear* equations on Z. This holds even if the parallelepipeds are of different size. Knowing in addition that they are of the same size, gives an additional *linear* equation.

Currently, we only exploit constraints on individual parallelepipeds (right angles and length ratios), since they are easier to provide for the user.

4.5 Complete Algorithm

1. Estimate the canonical projection matrices $\tilde{\mathsf{X}}_{ik}$.
2. Compute missing X_{ik}.
3. Normalize the X_{ik} to unit determinant.
4. Construct the measurement matrix and compute its SVD.
5. From the SVD, extract the matrices U_i and V_k.
6. Establish linear equation system on Z based on prior knowledge of intrinsic parameters of cameras and parallelepipeds.
7. Solve the system to least squares.
8. Extract T from Z using Cholesky decomposition.
9. Extract the $\mathsf{K}_i, \mathsf{R}_i, \mathsf{L}_k, \mathsf{S}_k$ from the $\mathsf{U}_i\mathsf{T}^{-1}$ and the $\mathsf{T}\mathsf{V}_k^\mathsf{T}$ using e.g. QR-decomposition. Note that at this stage the L_k can only be recovered up to scale, i.e. the parallelepipeds' (relative) sizes remain undetermined.

4.6 Minimal cases for the linear calibration

As mentioned in the last section, all constraints provided by knowledge on the cameras and parallelepipeds can be expressed in terms of the 5 independent parameters of the matrix Z. Thus, information about a total of only five intrinsic parameters of cameras or parallelepipeds

parallelepipeds		cameras	
#	constraint(s)	#	constraint(s)
0		5	• K_1: $\{s, \tau, u_0, v_0\}$ known; K_2: s known • 5 cameras with known s
1	• 1 known length ratio • 1 right angle	4	• 2 cameras with known s and τ • 4 cameras with known s
2	• 2 right angles • 1 right angle and 1 known length ratio	3	• 1 camera with $\{s, u_0, v_0\}$ known • 3 cameras with known s
3	• 3 right angles • 2 right angles and 1 known length ratio • 1 right angle and 2 known length ratios	2	• 1 camera with known s and τ • 1 camera with known u_0 and v_0 • 2 cameras with known s
4	• 3 right angles and 1 known length ratio • 2 right angles and 2 known length ratios	1	• 1 camera with known τ • 1 camera with known s
5	• 3 right angles and 2 length ratios • L_0: 2 and L_1: 3 right angles	0	

Table 1: Some minimal cases for the linear calibration algorithm. The problem has five degree of freedom (see text). The table contains a non-exhaustive list of cases where the number of constraints on parallelepipeds and on cameras, sum up to five.

is in general sufficient to calibrate the whole system. In table 1 we give a non-exhaustive list of practical minimal cases. Note however that certain configurations, i.e. relative positioning of cameras and parallelepipeds, represent singularities, depending on the amount of prior information available. Such singularities are discussed in section 5.

5 Singularities

Many calibration or self-calibration algorithms are subject to more or less severe singularities, i.e. there exist situations, where the algorithm is bound to fail. Furthermore, even in situations that are not exactly singular, but close to a singularity, the results become usually very unstable. In this section, we examine the singularities for the linear calibration algorithm described above. First, we study the singularities in the case of one parallelepiped being seen by one camera. We then study some multi-view cases, where we exploit results on critical motions for classical self-calibration.

5.1 One parallelepiped in a single view

We have studied all possible combinations of *a priori* knowledge, on both camera and parallelepiped intrinsic parameters leading to the linear equations (see sections 4.4.1 and 4.4.2).

In the following we will sketch the methodology followed, give proofs for one sample configuration and provide the results for all configurations studied.

We first formulate the meaning of a singularity in terms of the ingredients of the calibration algorithm. The existence of a singularity in our case means exactly that equation (7) has more than one solution for ω and μ that conform to all available *a priori* information, i.e. that there is at least one solution that is different from the true one. It is easy to show that the existence of a singularity does not depend on the relative *position* of the camera and the parallelepiped, only on the relative *orientation* and the *a priori* knowledge on camera and parallelepiped intrinsic parameters.

Let $\mathsf{K} = \mathsf{K}_k\mathsf{K}_u$ be the true calibration matrix, and $\mathsf{K}' = \mathsf{K}_k\mathsf{K}'_u$ the estimated one (we decompose in known and unknown parts, so K' and K share of course the known part K_k). As for parallelepipeds, a similar decomposition into known and unknown parts is not always possible. If however, we only consider constraints arising from prior knowledge about right angles, then we can decompose as above: $\mathsf{L} = \mathsf{L}_u\mathsf{L}_k$ and $\mathsf{L}' = \mathsf{L}'_u\mathsf{L}_k$ are respectively the true and estimated intrinsic parameters of a parallelepiped.

With these definitions, a singularity exists if there are solutions for (7) with $\mathsf{K}'_u \neq \mathsf{K}_u$ and $\mathsf{L}'_u \neq \mathsf{L}_u$. From (7), it is easy to derive the following equality (using $\mathsf{X} \sim \mathsf{K}_k\mathsf{K}_u\mathsf{R}\mathsf{L}_u\mathsf{L}_k$, $\omega' \sim \mathsf{K}'^{-\mathsf{T}}\mathsf{K}'^{-1}$ and $\mu' \sim \mathsf{L}'^{\mathsf{T}}\mathsf{L}'$):

$$\mathsf{R}^{\mathsf{T}}\mathsf{K}_u^{\mathsf{T}}\mathsf{K}'^{-\mathsf{T}}_u\mathsf{K}'^{-1}_u\mathsf{K}_u\mathsf{R} \sim \mathsf{L}_u^{-\mathsf{T}}\mathsf{L}'^{\mathsf{T}}_u\mathsf{L}'_u\mathsf{L}_u^{-1}.$$

A singularity, as defined above, is then equivalent to the existence of matrices $\omega'' = \mathsf{K}_u^{\mathsf{T}}\mathsf{K}'^{-\mathsf{T}}_u\mathsf{K}'^{-1}_u\mathsf{K}_u$ and $\mu'' = \mathsf{L}_u^{-\mathsf{T}}\mathsf{L}'^{\mathsf{T}}_u\mathsf{L}'_u\mathsf{L}_u^{-1}$, with K'_u and L'_u of the desired form (given by the constraints), but which are different from the identity (otherwise, $\omega' \sim \omega$ and $\mu' \sim \mu$, i.e. we would look at the true solution).

Depending on the *a priori* knowledge, ω'' and μ'' have special forms (as shown in table 2 for ω''), independently of the actual values of the known or unknown intrinsic parameters. Hence, the configuration is singular for calibration if the relative orientation R between parallelepiped and camera is such that there are solutions ω'' and μ'' of the required special form and different from the identity, satisfying:

$$\exists(\omega'' \neq \mathsf{I}_{3\times3},\ \mu'' \neq \mathsf{I}_{3\times3}) : \mathsf{R}^{\mathsf{T}}\omega''\mathsf{R} \sim \mu'' \tag{17}$$

Based on this definition, it is a rather mechanical, though sometimes tricky, task, to derive singular relative orientations. Table 3 shows all singularities for nearly all combinations of the above cases. We explain the singularities in geometrical terms, by describing the relative orientation of the parallelepiped with respect to the camera. In the following paragraphs, we give a few comments on different cases of prior knowledge on the parallelepiped.

Three right angles, two length ratios in this case, the Euclidean structure of the parallelepiped is completely given (up to scale), and it can be used as a classical calibration object. There are singularities proper to the use of a parallelepiped, but of course the generic singularities described in [6] apply here too.

Known camera parameters			
(A) None	(B) τ	(C) u_0, v_0	(D) τ, u_0, v_0
$\begin{pmatrix} a & 0 & d \\ 0 & b & e \\ d & e & c \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & d \\ 0 & 1 & e \\ d & e & c \end{pmatrix}$	$\begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & c \end{pmatrix}$

Table 2: Structure of ω'' depending on prior knowledge on intrinsic camera parameters. Structure of μ'' is similar.

Case	Conditions for singularity
A-3-1	**v** is orthogonal to the **x** or **y** camera axis
B-3-1	**v** is parallel to the optical axis
C-3-1	**v** is parallel to any of the three camera axes
D-3-1	**v** is parallel to the optical axis
A-3-0	always (3 constraints for 4 camera intrinsics)
B-3-0	any edge is parallel to the image plane
C-3-0	any edge is parallel to a camera axis
D-3-0	any edge is parallel to the optical axis
A-2-2	too difficult to describe
B-2-2	**v** $\parallel$ image plane and **w** $\parallel$ optical axis or image plane
C-2-2	**v** $\parallel$ **x** or **y** axis and **w** at 45° angle with image plane
	v $\parallel$ **z** and **w** $\parallel$ image plane and at 45° to both **x** and **y**
D-2-2	never!
A-2-1	always (three constraints for four camera intrinsics)
B-2-1	**v** is parallel to the image plane
C-2-1	**v** parallel to either camera axis
	v and **w** are both orthogonal to the **x** camera axis
	v and **w** are both orthogonal to the **y** camera axis
	v and **w** are parallel to the image plane
D-2-1	**v** and**w** are parallel to the image plane
A-2-0	always (two constraints for four camera intrinsics)
B-2-0	always (two constraints for three camera intrinsics)
C-2-0	**v** orthogonal to the **x** or **y** camera axis or $\parallel$ image plane
D-2-0	**v** parallel to image plane or to optical axis

Table 3: Singular relative orientations for the case of one parallelepiped seen in one camera, for various combinations of prior knowledge. Cases are denoted X-Y-Z, where X $\in$ {A,B,C,D} refers to table 2 and Y and Z are the number of known right angles respectively length ratios. For further explanations, see text.

Three right angles, one length ratio (cases *-3-1 in table 3) in table 3, **v** represents the direction of the parallelepiped's edges which are not "involved" in the known length ratio.