Using Geometric Constraints for Camera Calibration and Positioning and 3D Scene Modelling

Marta Wilczkowiak, Peter Sturm, Edmond Boyer

USING GEOMETRIC CONSTRAINTS FOR CAMERA CALIBRATION AND POSITIONING AND 3D SCENE MODELLING

Marta Wilczkowiak, Peter Sturm, Edmond Boyer

INRIA Rhône-Alpes, 655 Avenue de l'Europe, 38330 Montbonnot, France – Firstname.Lastname@inrialpes.fr

KEY WORDS: 3D, Calibration, Object Reconstruction, Orientation, Reconstruction

ABSTRACT

This work concerns the incorporation of geometric information in camera calibration and 3D modelling. Using geometric constraints enables stabler results and allows to perform tasks with fewer images. Our approach is interactive; the user defines geometric primitives and constraints between them. It is based on the observation that constraints such as coplanarity, parallelism or orthogonality, are easy to delineate by a user, and are well adapted to model the main structure of e.g. architectural scenes. We propose methods for camera calibration, camera position estimation and 3D scene reconstruction, all based on such geometric constraints. Various approaches exist for calibration and positioning from constraints, often based on vanishing points. We generalize this by considering composite primitives based on triplets of vanishing points. These are frequent in architectural scenes and considering composites of vanishing points makes computations more stable. They are defined by depicting in the images points belonging to parallelepipedic structures (e.g. appropriate points on two connected walls). Constraints on angles or length ratios on these structures can then be easily imposed. A method is proposed that "collects" all these data for all considered images, and computes simultaneously the calibration and pose of all cameras via matrix factorization. 3D scene reconstruction is then performed using many more geometric constraints, i.e. not only those encapsulated by parallelepipedic structures. A method is proposed that reconstructs the whole scene in iterations, solving a linear equation system at each iteration, and which includes an analysis of the parts of the scene that can/cannot be reconstructed at the current stage. The complete approach is validated by various experimental results, for cases where a single or several views are available.

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 has proven that recovering 3D structures from 2D images is a difficult and often under-constrained problem. Several reasons account for that, including the fact that without any prior information on cameras, or on the scene to recover, a euclidean reconstruction is not possible at all (Faugeras, 1992). This is why knowledge on the acquisition process, or on the scene, is required. A number of approaches have been proposed to exploit prior information, both on camera and scene parameters. Such prior information does not only solve the projective ambiguity in the reconstruction but do also usually stabilize the 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 higher accuracy. The method proposed in this paper 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. Using parallelepipeds to constrain the calibration and reconstruction process 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.

An exhaustive review of literature on using prior information for self-calibration and euclidean reconstruction is beyond the scope of this paper. We will concentrate on works which have somehow inspired the method we propose, especially direct approaches giving a good first estimate of camera and scene parameters. There is a large variety of information which can be incorporated into a 3D modeling process. This can be simple knowledge on camera intrinsic parameters or pose (stationarity, pure translation, etc.) or on global 3D scene structure (calibration patterns); it can also be information on scene elements such as points, lines and planes, as well as on high-level primitives like cubes, prisms, cylinders, etc. Nonetheless, whatever the information is, it can be used at any stage of the 3D modeling process, including the initial calibration, pose estimation, model reconstruction or an additional non-linear adjustment of the initial estimate at each step.

Approaches based on calibration patterns. Classical calibration approaches are based on known positions of points in 3D space, or known calibration patterns (Tsai, 1986). Unfortunately, such information relies on specific acquisition systems and is thus seldom available in general situations. The use of prior knowledge on some intrinsic parameters, i.e. self-calibration, offers the opportunity to build more flexible systems.

Self-calibration. In standard self-calibration algorithms (Maybank, 1992; Triggs, 1997; Hartley, 1993; Pollefeys, 1997), 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 critical motion sequences, for which self-calibration does not have a unique solution (Sturm, 1997). This problem has been dealt with by restraining the camera motions (Hartley, 1997; de Agapito, 1999; Armstrong, 1994) or by incorporating prior knowledge on the camera (Zisserman, 1998) or on the scene. But to get stable results for self-calibration, a large number of images is usually necessary.

Structure and motion. The basic constraint is that backprojection lines (planes) associated with corresponding image points (lines) intersect in a single space point (line). This observation allows to formulate the matching tensors, which compactly describe two, three and four view geometry. When more views are accessible, it is necessary to combine results computed from small subsets of images, which decreases the accuracy of results. An overview of tensor-based structure&motion methods can be found in (Hartley, 2000).

Another category of approaches allows the simultaneous recovery of cameras and 3D models via the factorization of a measurement matrix of image points (Tomasi, 1992; Sturm, 1996), lines (Triggs, 1996; Martinec, 2002) or similar methods using planes in the scene (Rother, 2002; Sturm, 2000). Factorization methods

suffer from missing data, i.e. when a primitive is not seen in all images, although some ways of dealing with this problem have been proposed (Tomasi, 1992; Martinec, 2002). Using only the above backprojection constraints, it is only possible to recover the scene up to a projective or affine transformation.

Incorporating euclidean scene constraints. A variety of geometric constraints can disambiguate the projective reconstruction to a euclidean one, 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 common constraint is given by vanishing points of mutually orthogonal directions, as defined by known cubical structures (Caprile, 1990; Cipolla, 1998; Chen, 1999) or by dominating scene directions (Kosecka, 2002). Also, knowing the euclidean structure of scene planes is useful in this context, through rectified planes (Liebowitz, 1999), maps (Bondyfalat, 2001) or known plane-to-image homographies (Sturm, 1999a; Zhang, 1999). It is also possible to use multiple images of unknown planes, but more images in general position are needed here (Triggs, 1998; Malis, 2002).

When cameras are calibrated, it is relatively easy to reconstruct 3D structure. However, and as mentioned previously, using geometric constraints may improve dramatically the reconstruction quality, especially when a single or only few images are considered (Boufama, 1993). Even simple constraints can be very efficient, e.g. in (Criminisi, 2000; Sturm, 1999b), vanishing lines of planes and coplanarity constraints are used for single image reconstruction. However, in general, dealing with different types of scene objects and constraints is a complicated problem. Some authors prefer to model the scene by simple primitives like points, lines and planes and constraints between them such as incidence, parallelism, orthogonality, etc. Some direct approaches using the bilinear character of many useful constraints were proposed in (Shum, 1998; Grossman, 2002; Wilczkowiak, 2003a). The results can be improved using non-linear methods applying penalty terms corresponding to the constraints (McGlone, 1995), constrained optimization techniques (Szeliski, 1998; McLauchlan, 2000; Grossmann, 2000), or a minimal scene parameterization (Bondyfalat, 1998; Wilczkowiak, 2003b). Yet a different approach consists in high-level scene descriptions using complex primitives like cubes, prisms, cylinders, etc. (Debevec, 1996; Jelinek, 2000). Recently, some effort has been devoted to the automatic detection of such primitives (Dick, 2001). All these methods ensure, by the strong inherent geometric constraints, that the final models are visually correct.

The proposed approach. In this paper, we address the intrinsic and extrinsic calibration (pose/motion estimation) as well as 3D reconstruction, using geometric constraints. As for calibration, we study the use of a specific calibration primitive: the parallelepipeds. Parallelepipeds are frequently present in man-made environments and they naturally encode the scene's affine structure. Any information about their euclidean structure (angles or ratios of edge lengths), possibly combined with information about camera parameters, may allow to recover the entire scene's euclidean structure. 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. Hence, the image of a known parallelepiped defines the camera parameters, and reciprocally, a calibrated image of a parallelepiped defines its euclidean shape (up to size). In this paper, we synthesize previous work on parallelepipeds (Wilczkowiak, 2001; Wilczkowiak, 2002) and propose more elegant and efficient approaches.

Camera and parallelepiped parameters are recovered in two steps. First, a factorization-based approach is used to compute their intrinsic and orientation (rotation) parameters. The usual problems of factorization methods – missing data and unknown scale factors – are dealt with rather easily. Then, position and size parameters are recovered simultaneously using linear least squares. The use of well-constrained calibration primitives allows to obtain good calibration results even from as little as one image.

Our calibration approach is conceptually close to self-calibration, especially to methods that upgrade an affine structure to euclidean (Hartley, 1993; Pollefeys, 1997) or methods considering special camera motions (Hartley, 1997; de Agapito, 1999; Armstrong, 1994). The way euclidean information on a parallelepiped is used is also similar to vanishing point based methods (Caprile, 1990; Cipolla, 1998; Chen, 1999; Kosecka, 2002). Some properties of our algorithm are also common with plane-based approaches (Sturm, 1999a; Zhang, 1999; Triggs, 1998; Malis, 2002; Rother, 2002; Sturm, 2000). While more flexible than standard calibration techniques, plane-based approaches still require either euclidean information or, for self-calibration, many images in general position (Triggs, 1998), or at least one plane visible in all images (Rother, 2002). In this sense, our approach is a generalization of plane-based methods with euclidean information, to three-dimensional parallelepipedic patterns. Finally, our approach can be compared to methods using complex primitives for scene representation. However, unlike most such methods, we use the parallelepiped parameters directly to solve the calibration problem, without requiring non-linear optimization.

After discussing calibration, we show that the proposed method can be easily combined with an approach for enhancing reconstructions with primitives other than parallelepipeds (Wilczkowiak, 2003a). The complete system allows for both calibration and 3D model acquisition from a small number of images with a reasonable amount of user interaction.

2 PRELIMINARIES

2.1 Camera parameterization

We represent cameras using the pinhole model. The projection of a 3D point $\mathbf{P}$ to a 2D image point is expressed by $\mathbf{p} \sim \mathsf{M}\mathbf{P}$, where M is a 3×4 *projection 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 (extrinsic parameters) in the world coordinate system: the rotation matrix R represents its orientation and the vector $-\mathsf{R}^\mathsf{T}\mathbf{t}$ its position. The 3×3 calibration matrix K or, equivalently, $\omega \sim \mathsf{K}^{-\mathsf{T}}\mathsf{K}^{-1}$ represents the camera's intrinsic parameters:

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

$$\omega \sim \mathsf{K}^{-\mathsf{T}}\mathsf{K}^{-1} \sim \begin{pmatrix} 1 & 0 & -u_0 \\ 0 & \tau^2 & -\tau^2 v_0 \\ -u_0 & -\tau^2 v_0 & \tau^2\alpha_v^2 + u_0^2 + \tau^2 v_0^2 \end{pmatrix} \quad (1)$$

where α_u and α_v stand for the focal length, expressed in horizontal and vertical pixel dimensions, s is a skew parameter considered as equal to zero in the following, (u_0, v_0) are the pixel coordinates of the principal point and $\tau = \frac{\alpha_u}{\alpha_v}$ is the camera's aspect ratio. ω represents the IAC (image of the absolute conic) and is commonly used to express constraints on the intrinsic parameters. In the following, the term *camera axes* will be used for the axes of the camera coordinate system, i.e. the coordinate system attached to the camera's optical center, two of them being parallel to pixel edges and the third one being orthogonal to the image plane (the optical axis).

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

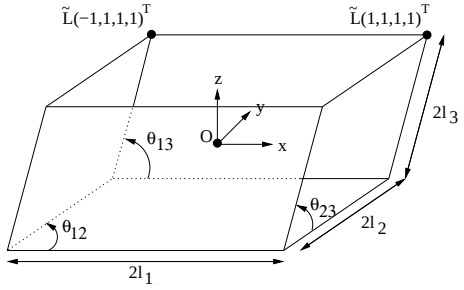


Figure 1: Parameterization of a parallelepiped: $2l_i$ are the edge lengths; θ_{ij} are the angles between non-parallel edges.

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 by a 4×4 matrix N:

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

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 (intrinsic parameters) 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.

Analogous to a camera's IAC ω is the matrix μ, defined by:

$$\mu \sim \mathsf{L}^\mathsf{T}\mathsf{L} \sim \begin{pmatrix} l_1^2 & l_1 l_2 \cos\theta_{12} & l_1 l_3 \cos\theta_{13} \\ l_1 l_2 \cos\theta_{12} & l_2^2 & l_2 l_3 \cos\theta_{23} \\ l_1 l_3 \cos\theta_{13} & l_2 l_3 \cos\theta_{23} & l_3^2 \end{pmatrix} \quad (2)$$

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

Hence, there is a symmetry between the intrinsic parameters of cameras and parallelepipeds (expressions (1) and (2)). 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 a parallelepiped. We choose the following definition, motivated by the equations underlying our calibration and reconstruction algorithms below: the **size** of a parallelepiped is defined as $s = (\det \mathsf{L})^{1/3}$. 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

In this section, we introduce the concept of duality between the intrinsic parameters of cameras and parallelepipeds. Consider the projection of a parallelepiped's vertices into a camera. Let

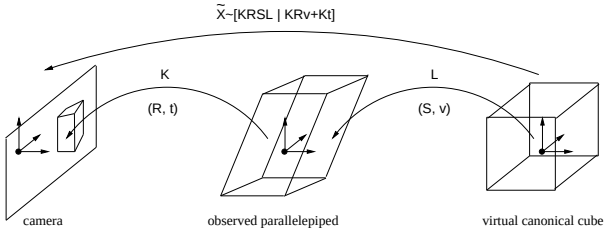


Figure 2: The projection of the canonic parallelepiped (cube) into the image. Matrices K, L correspond to intrinsic parameters of camera and parallelepiped and $(\mathsf{R}, \mathbf{t})$, $(\mathsf{S}, \mathbf{v})$ correspond to extrinsic parameters of camera and parallelepiped, respectively.

$\mathbf{C}_{i,\ i\in[1..8]}$ be the homogeneous coordinates of the canonic cube's vertices. Using results from section 2.2, the projection of the corresponding vertex in the image is:

$$\mathbf{p}_i \sim \mathsf{M}\mathbf{P}_i = \underbrace{\mathsf{K}\begin{pmatrix} \mathsf{R} & \mathbf{t} \end{pmatrix}\begin{pmatrix} \mathsf{S} & \mathbf{v} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix}\tilde{\mathsf{L}}}_{\tilde{\mathsf{X}}}\mathbf{C}_i \quad (3)$$

The matrix $\tilde{\mathsf{X}}$ will be called the *canonic projection matrix*. It represents a perspective projection that maps the vertices of the canonic cube onto the image points of the parallelepiped's vertices. This is illustrated in figure 2. Given image points for at least six vertices, the canonic projection matrix can be computed (Tsai, 1986), even without prior knowledge on intrinsic or extrinsic parameters. Our calibration and pose algorithms are based on the link between the canonic projection matrix $\tilde{\mathsf{X}}$ (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 (3), represented by the following 3×4matrix:

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

Second, let us consider the leading 3×3 sub-matrix X of the canonic projection matrix $\tilde{\mathsf{X}}$, which is given by: $\mathsf{X} \sim \mathsf{K}(\mathsf{RS})\mathsf{L}$.

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:

$$\mathsf{X}^\mathsf{T}\omega\mathsf{X} \sim \mu. \quad (4)$$

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 compute the parallelepiped's euclidean shape, also from a single image. Moreover, even partial information about one set of intrinsic parameters allows to form equations on the other set (Wilczkowiak, 2001).

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 reconstruction of the scene is obtained from image correspondences. Then, this is upgraded to a euclidean reconstruction using the available prior knowledge on intrinsic parameters. Sometimes an intermediate upgrade to an affine reconstruction is performed.

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 ob-

served 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 part of self-calibration. Indeed, our calibration method is somewhat similar to the affine-to-euclidean upgrade of stratified self-calibration approaches, e.g. (Hartley, 1993; Pollefeys, 1997).

Similarities also exist with (self-) calibration approaches based on special camera motions: calibrating a rotating camera (Hartley, 1997; de Agapito, 1999) 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 (Armstrong, 1994; Pollefeys, 1996).

Our approach is similar to all 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

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 and λ_{ik} a scale factor such that equation (3) can be written as a component-wise equality:

$$\lambda_{ik}\tilde{\mathsf{X}}_{ik} = \mathsf{K}_i \begin{pmatrix} \mathsf{R}_i & \mathbf{t}_i \end{pmatrix} \begin{pmatrix} \mathsf{S}_k & \mathbf{v}_k \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \tilde{\mathsf{L}}_k \tag{5}$$

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

$$\underbrace{\begin{bmatrix} \lambda_{11}\tilde{\mathsf{X}}_{11} & \cdots & \lambda_{1n}\tilde{\mathsf{X}}_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{m1}\tilde{\mathsf{X}}_{m1} & \cdots & \lambda_{mn}\tilde{\mathsf{X}}_{mn} \end{bmatrix}}_{\mathcal{X}_{3m\times 4n}} = \underbrace{\begin{bmatrix} \mathsf{K}_1 \begin{pmatrix} \mathsf{R}_1 & \mathbf{t}_1 \end{pmatrix} \\ \vdots \\ \mathsf{K}_m \begin{pmatrix} \mathsf{R}_m & \mathbf{t}_m \end{pmatrix} \end{bmatrix}}_{\mathcal{M}_{3m\times 4}} \underbrace{\begin{bmatrix} \begin{pmatrix} \mathsf{S}_1 & \mathbf{v}_1 \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \tilde{\mathsf{L}}_1 & \cdots & \begin{pmatrix} \mathsf{S}_n & \mathbf{v}_n \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \tilde{\mathsf{L}}_n \end{bmatrix}}_{\mathcal{S}_{4\times 4n}} \tag{6}$$

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 (Tomasi, 1992), we call it *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 directly from a rank-4-factorization of $\mathcal{X}$. One step of factorization-based methods for structure and motion recovery is to disambiguate the factorization's result: in general, for a rank-r-factorization, motion and shape are recovered up to a transformation represented by an $r\times r$ matrix (here, this would be a 3D projective transformation). The ambiguity can be reduced using e.g. constraints on intrinsic camera parameters (see details in section 4). In our case, we observe that the 4×4 sub-blocks of the shape matrix $\mathcal{S}$ are affine transformations. We would have to include this constraint into the disambiguation, but nevertheless, the result would not in general exactly satisfy the affine form for these blocks. We thus cut the problem in two steps, which allows to guarantee that the sub-blocks of the shape matrix be affine transformations. In the first step (sections 4.1–4.5), we consider a "reduced measurement matrix" consisting of the leading 3×3 sub-matrices of the $\tilde{\mathsf{X}}_{ik}$. We extract *intrinsic* and *orientation* parameters 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 4.6), we then estimate the *position* of cameras and parallelepipeds, as well as the parallelepipeds' *size*.

4 CALIBRATION AND POSITIONING

4.1 Problem Formulation

Up to section 4.5, we concentrate on the computation of the cameras' and parallelepipeds' intrinsic parameters and orientation (rotation), based on equation (6) and the observations concerning it, cf. the previous section. As mentioned, we first restrict our attention to the leading 3×3 submatrices of the $\tilde{\mathsf{X}}_{ik}$, like in section 3.1 for the establishment of the duality between intrinsic parameters of cameras and parallelepipeds. We thus deal with the following subpart of equation (6):

$$\underbrace{\begin{bmatrix} \lambda_{11}\mathsf{X}_{11} & \cdots & \lambda_{1n}\mathsf{X}_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{m1}\mathsf{X}_{m1} & \cdots & \lambda_{mn}\mathsf{X}_{mn} \end{bmatrix}}_{\mathcal{X}'_{3m\times 3n}} = \underbrace{\begin{bmatrix} \mathsf{K}_1\mathsf{R}_1 \\ \vdots \\ \mathsf{K}_m\mathsf{R}_m \end{bmatrix}}_{\mathcal{M}'_{3m\times 3}} \underbrace{\begin{bmatrix} \mathsf{S}_1\mathsf{L}_1 & \cdots & \mathsf{S}_n\mathsf{L}_n \end{bmatrix}}_{\mathcal{S}'_{3\times 3n}} \tag{7}$$

In the following, we describe the different steps of our factorization-based method. We first deal with the problem of missing data. Then we describe how to compute the scale factors λ_{ik}, needed to construct the measurement matrix $\mathcal{X}'$. The factorization itself is described in section 4.4, followed by the most important aspect: disambiguating the factorization's result in order to extract intrinsic and orientation parameters.

In section 4.6, we then describe the subsequent computation of position parameters and parallelepiped size The complete calibration and positioning algorithm is summarized in section 4.7.

4.2 Missing Data

As is usual with factorization approaches, 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 is usually not satisfied. However, each missing matrix X_{ik} can be deduced from others if there is one camera j and one parallelepiped l such that X_{jl}, X_{jk} and X_{il} are known. The missing matrix can be computed using:

$$\mathsf{X}_{ik} \sim \mathsf{X}_{il} \left(\mathsf{X}_{jl}\right)^{-1} \mathsf{X}_{jk}. \tag{8}$$

Several equations of this type can be used simultaneously to increase the accuracy. Care has to be taken since (8) 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 (Sturm, 2000) to compute missing matrices X_{ik}: at each iteration, we compute the one for which most equations of type (8) are available. Previously computed matrices X_{ik} can be involved at every successive iteration of this procedure.

4.3 Recovery of Scale Factors

The reduced measurement matrix $\mathcal{X}'$ in (7) 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. (Malis, 2002; Sturm, 1996). 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 a_i and b_k that satisfy:

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

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

Hence, there exist scale factors a_i and b_k with:

$$\forall i,k : \mathsf{X}_{ik} \sim a_i b_k \mathsf{A}_i\mathsf{B}_k \tag{9}$$

$$\forall i,k : \det(a_i b_k \mathsf{A}_i\mathsf{B}_k) = \det(a_i\mathsf{A}_i)\det(b_k\mathsf{B}_k) = 1 \tag{10}$$

As for the sought for scale factors λ_{ik}, we use those that give $\det(\lambda_{ik}\mathsf{X}_{ik}) = 1$. They are computed as:

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

Due to (9), we have $\lambda_{ik}\mathsf{X}_{ik} \sim a_i b_k \mathsf{A}_i\mathsf{B}_k$ and since the determinants of both sides of this equation are equal (they are both equal to 1, cf. the definition of λ_{ik} and (10)), the equation not only holds up to scale, but component-wise (two non-singular 3×3 matrices that are equal up to scale and whose determinants are equal, are also equal component-wise):

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

This means that the measurement matrix in (7), with the scale factors λ_{ik} as described here, is of rank 3: it is the product of one matrix of 3 columns (the $a_i\mathsf{A}_i$ stacked on top of each other) and one of 3 rows (the $b_k\mathsf{B}_k$ side-by-side).

In the following, we assume that the X_{ik} are already scaled to unit determinant, i.e. that $\lambda_{ik} = 1$. Equation (7) becomes:

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

The scale factors a_i and b_k do not matter for now; all that counts is that they exist and 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.

4.4 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 $\mathcal{X}'$: $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{3n}$. 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).

In the following, we consider 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 corresponding to non-zero σ_j contribute. Hence:

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

where U' (resp. V') consists of the first three columns of U (resp. V). Let us define $\mathsf{U}'' = \mathsf{U}'\ \mathrm{diag}(\sqrt{\sigma_1},\sqrt{\sigma_2},\sqrt{\sigma_3})$ and $\mathsf{V}'' = \mathsf{V}'\ \mathrm{diag}(\sqrt{\sigma_1},\sqrt{\sigma_2},\sqrt{\sigma_3})$. Thus we have: $\mathcal{X}' = \mathsf{U}''\mathsf{V}''^\mathsf{T}$. This represents a decomposition of the measurement matrix $\mathcal{X}'$ into a product of a matrix of 3 columns (U'') with a matrix of 3 rows (V''^T). 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 (11), we obtain:

$$\begin{bmatrix} a_1\mathsf{K}_1\mathsf{R}_1 \\ \vdots \\ a_m\mathsf{K}_m\mathsf{R}_m \end{bmatrix}\begin{bmatrix} b_1\mathsf{S}_1\mathsf{L}_1 & \cdots & b_n\mathsf{S}_n\mathsf{L}_n \end{bmatrix} = \left(\mathsf{U}''\mathsf{T}^{-1}\right)\left(\mathsf{T}\mathsf{V}''^\mathsf{T}\right) \tag{12}$$

Let us decompose matrices U'' and V'' in 3×3 submatrices: $\mathsf{U}''^\mathsf{T} = [\mathsf{U}_1^\mathsf{T}\cdots\mathsf{U}_m^\mathsf{T}]$ and $\mathsf{V}''^\mathsf{T} = [\mathsf{V}_1^\mathsf{T}\cdots\mathsf{V}_n^\mathsf{T}]$. Equation (12) thus becomes:

$$\begin{bmatrix} a_1\mathsf{K}_1\mathsf{R}_1 \\ \vdots \\ a_m\mathsf{K}_m\mathsf{R}_m \end{bmatrix}\begin{bmatrix} b_1\mathsf{S}_1\mathsf{L}_1\cdots b_n\mathsf{S}_n\mathsf{L}_n \end{bmatrix} = \begin{bmatrix} \mathsf{U}_1\mathsf{T}^{-1} \\ \vdots \\ \mathsf{U}_m\mathsf{T}^{-1} \end{bmatrix}\begin{bmatrix} \mathsf{T}\mathsf{V}_1^\mathsf{T}\cdots\mathsf{T}\mathsf{V}_n^\mathsf{T} \end{bmatrix} \tag{13}$$

How to estimate T is explained in section 4.5. 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 and calibration matrices can be recovered by Cholesky or QR-decompositions. The Cholesky decomposition of $\mathsf{A}_i\mathsf{A}_i^\mathsf{T}$ e.g., 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 $\mathsf{K}_i = \frac{1}{a_i}\mathsf{M}_i$. Note that in overconstrained situations, the computed calibration matrices will not in general exactly satisfy the constraints used for their computation; the best way of dealing with this may be a constrained non-linear optimization.

As for parallelepipeds, there is no constraint similar to $K_{i,33} = 1$ 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 4.6, we explain how to compute their (relative) size.

We now briefly discuss the structure and geometric signification of matrix T. Note that T actually represents the non-translational part of a 3D affine transformation (its upper left 3×3 submatrix). 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 we have: $\mathsf{T}'^{-1}\mathsf{T}' \sim \mathsf{T}^{-1}\mathsf{T}$. This ambiguity is natural and expresses the fact that the global euclidean reference frame for the reconstruction of parallelepipeds and cameras can be chosen arbitrarily. Without loss of generality, we 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 free scale) which can also be explained in more geometric terms: as explained above, our problem is somewhat equivalent to self-calibration with known affine structure. The 5 degrees of the problem can thus be interpreted as the coefficients of the absolute conic on the plane at infinity.

4.5 Disambiguating the Factorization

We now deal with the estimation of the unknown transformation T appearing in equation (13). 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: $\mathsf{Z} = \mathsf{T}^\mathsf{T}\,\mathsf{T}$. (We may observe that Z 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 cameras and parallelepipeds. In the following, we consider two types of information, first for cameras and then for parallelepipeds:

- 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.

4.5.1 Using Information on Camera Intrinsics. From equation (13), 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):

$$\omega_i \sim \mathsf{U}_i^{-\mathsf{T}}\mathsf{Z}\mathsf{U}_i^{-1}. \tag{14}$$

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

Known values of camera intrinsics Knowing the aspect ratio and principal point coordinates of a camera i and substituting ω_i according to (14) and (1), the following linear constraints on Z can be written:

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

A known value of the focal length α_v can only be used to formulate linear equations if all the other intrinsics are also known. In such a fully calibrated case, other algorithms (Sturm, 1999a) might be better suited, so we neglect that case in the following.

Constant camera intrinsics In the case when 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 quadratic equation:

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

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 (Hartley, 1997): the matrices U^i are first scaled such as to have unit determinant. Then we can write the following component-wise matrix equality between any pair (i, j) of views:

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

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

4.5.2 Information on Parallelepipeds. From equation (13), 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 factor b_k:

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

Knowledge on parallelepiped intrinsics can be used in analogous ways as for camera parameters. For example suppose we know the length ratio of two parallelepiped edges $r_{uv} = \frac{l_u}{l_v}$. Referring to (2), 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 θ_{uv} is a right angle ($\cos\theta_{uv} = 0$) gives also a *linear equation*:

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

A known angle θ_{uv} that is not a right angle does not lead to a linear, but a *bilinear* equation (Wilczkowiak, 2001).

Like for cameras, *quadratic* equations may be derived from assumptions about two or more parallelepiped having the same, yet unknown value for some intrinsic parameter. Also, two parallelepipeds having the same shape give a set of *linear* equations on Z, even if the parallelepipeds are of different size. Equal size of parallelepipeds gives an additional *linear* equation, but which constrains relative pose rather than intrinsic parameters.

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

4.6 Estimating position and size

In this section we propose an algorithm for estimating the (relative) positions of the cameras and parallelepipeds, as well as the (relative) sizes of the parallelepipeds. Consider equation (5):

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

The leading 3×3 sub-part of the two sides of the equation were used above to compute the intrinsic camera parameters K_i and the rotation matrices R_i and S_k. The parallelepipeds' intrinsic parameters L_k were computed up to scale only, i.e. up to the "size" of the parallelepipeds.

Let us consider this in detail. In the following, we suppose that the matrices $\tilde{\mathsf{X}}_{ik}$ are already scaled such that the sub-matrices X_{ik} have unit determinant, as in section 4.3, i.e. $\lambda_{ik} = 1$. Let $\bar{\mathsf{K}}_i$ and $\bar{\mathsf{L}}_k$ be the calibration matrices scaled to unit determinant. We know all matrices in the following equation: $\mathsf{X}_{ik} = \bar{\mathsf{K}}_i\mathsf{R}_i\mathsf{S}_k\bar{\mathsf{L}}_k$.

What we don't know is the size s_k of the parallelepipeds. Let us observe the following:

$$\tilde{\mathsf{L}}_k = \begin{pmatrix}s_k\bar{\mathsf{L}}_k & \mathbf{0}\\ \mathbf{0}^\mathsf{T} & 1\end{pmatrix} \sim \begin{pmatrix}\bar{\mathsf{L}}_k & \mathbf{0}\\ \mathbf{0}^\mathsf{T} & 1/s_k\end{pmatrix}$$

We may now rewrite equation (5):

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

Let $\mathbf{x}_{ik}$ be the fourth column of $\tilde{\mathsf{X}}_{ik}$. We have:

$$\mathbf{x}_{ik} = \bar{\mathsf{K}}_i\begin{pmatrix}\mathsf{R}_i & \mathbf{t}_i\end{pmatrix}\begin{pmatrix}\mathsf{S}_k & \mathbf{v}_k\\ \mathbf{0}^\mathsf{T} & 1\end{pmatrix}\begin{pmatrix}\mathbf{0}\\ 1/s_k\end{pmatrix} = \frac{1}{s_k}\bar{\mathsf{K}}_i(\mathsf{R}_i\mathbf{v}_k + \mathbf{t}_i)$$

From this, we get an equation that is linear in all unknowns (s_k, $\mathbf{t}_i$ and $\mathbf{v}_k$):

$$s_k\mathbf{x}_{ik} - \bar{\mathsf{K}}_i\mathsf{R}_i\mathbf{v}_k - \bar{\mathsf{K}}_i\mathbf{t}_i = \mathbf{0} \tag{15}$$

The unknowns can be computed via linear least squares: minimizing the sum of the squared L_2 norms of the vectors on the left hand side of (15), over all camera–parallelepiped pairs. The estimates for the s_k, $\mathbf{t}_i$ and $\mathbf{v}_k$ are of course defined up to a single global scale. At this stage, missing data are not an issue any more, contrary to the computations in sections 4.2 and 4.4.

4.7 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 a linear equation system on Z based on prior knowledge of intrinsic parameters of cameras and parallelepipeds and solve it to least squares.
7. If Z is positive definite extract T from Z using Cholesky decomposition.
8. 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.
9. Let $\bar{\mathsf{K}}_i \sim \mathsf{K}_i$ with $\det \bar{\mathsf{K}}_i = 1$, and $\mathbf{x}_{ik}$ the fourth column of $\tilde{\mathsf{X}}_{ik}$. Solve via linear least squares for the s_k, $\mathbf{v}_k$ and $\mathbf{t}_i$, over all available equations of type (15).

This algorithm allows to calibrate a set of cameras using very little prior knowledge (see (Wilczkowiak, 2004) for examples of minimal cases). Indeed, as mentioned in this section, all constraints provided by knowledge on cameras and parallelepipeds can be expressed in terms of the 5 independent parameters of the matrix Z. Thus, to calibrate the whole system it is in general sufficient to know values of a total of only five intrinsic parameters of cameras or parallelepipeds. That is why in practice, we only use the associated linear equations. In most cases they are sufficient to find a unique solution. In some minimal cases, when the available linear constraints are insufficient, quadratic equations might be used to find a unique solution or a finite set of solutions.

There exist singular configurations, i.e. relative positions between cameras and parallelepipeds, for which calibration fails. These depend on the type of available constraints; singularities for the cases of one parallelepiped seen in one or many views, are described exhaustively in (Wilczkowiak, 2004).

5 3D RECONSTRUCTION

The calibration approach presented in section 4 is well adapted to interactive 3D modeling from a few images. It has a major advantage over other methods: simplicity. Indeed, only a small amount of user interaction is needed for both calibration and reconstruction: a few points must be picked in the image to define the primitives' image positions. It thus seems to be an efficient and intuitive way to build models from images of any type, in particular from images taken from the Internet for which no information about the camera is known.

To reconstruct scene elements not belonging to parallelepipeds, but being constrained by bilinear relations such as collinearity, coplanarity or parallelism, we have implemented a multi-linear reconstruction method (Wilczkowiak, 2003a). The reconstruction step is independent from the calibration method, although it uses the same input in the first step. Interestingly, it allows 3D models to be computed from non-overlapping photographs (see e.g. figure 6).

In this section we summarize our reconstruction method. First, we propose a method for extraction of uniquely defined variables in linear systems, which is the basis of the reconstruction method. Then we describe shortly the algorithm and certain related practical issues. Results are presented in the next section.

5.1 Extraction of Uniquely Defined Variables in Linear Systems

Consider the following linear equation system:

$$\mathsf{A}_{m\times n}\mathbf{X}_n = \mathbf{B}_m, \tag{16}$$

and assume that the solution for $\mathbf{X}$ is not unique. We then want to determine if there exists a subset of coefficients of $\mathbf{X}$ which can nevertheless be unambiguously estimated. This proves to be very useful in many approaches based on linear constraints, such as intrinsic and extrinsic camera calibration or 3D reconstruction, as described below.

Our approach is based on the analysis of the nullspace of matrix A. Using Singular Value Decomposition (Golub, 1989), matrix A can be decomposed as follows:

$$\mathsf{A}_{m\times n} = \mathsf{U}_{m\times n}\mathsf{W}_{n\times n}\mathsf{V}^{\mathsf{T}}_{n\times n},$$

where the matrices U and V are column-orthogonal and W is diagonal, with the singular values w_i of $\mathbf{A}$ on its diagonal, in decreasing order. Let $\mathbf{X}_0$ be some vector satisfying the equation system. Then, any vector $\mathbf{X}$ satisfying (16) must be of the form:

$$\mathbf{X} = \mathbf{X_0} + \sum_{i=r+1}^{n} \lambda_i \mathbf{v}^i, \lambda_i \in \mathcal{R}. \tag{17}$$

where vectors $\mathbf{v}^i$ are the columns of V corresponding to zero singular values w_i, constituting the nullspace of A and λ_i are arbitrary scalar factors, and r state for rank of matrix A.

The solution for a coefficient of $\mathbf{X}$, say $\mathbf{X}(k)$, is unique, if

$$\forall \lambda_i : \mathbf{X}(k) = \mathbf{X_0}(k),$$

which implies that

$$\forall \lambda_i : \sum_{i=r+1}^{n} \lambda_i \mathbf{v}^i(k) = 0$$

and is equivalent to:

$$\forall i \in \{r+1, \ldots, n\} : \mathbf{v}^i(k) = 0. \tag{18}$$

Hence, all variables $x_k = \mathbf{X}(k)$ corresponding to rows $\mathbf{r}_k$ of matrix $\mathsf{V}[1 \ldots n, r+1 \ldots n]$, such that $\|\mathbf{r}_k\| = 0$, have unique values $x_k = \mathbf{X}_0(k)$. Geometrically, this means that the axis represented by vector $\mathbf{e}^n_k$ of the solution space $\mathcal{R}^n$ corresponding to such a sufficiently constrained variable is orthogonal to the nullspace of matrix A.

Choice of thresholds. Using equation (18), it is in principle straightforward to split the unknowns of the system into well-defined and ambiguous ones. Note that this requires deciding if certain numerical values (singular values and coefficients $\mathbf{v}^i(k)$) are equal to zero. It is well known that due to noise and round-off errors, the numerically computed singular values of a matrix are never exactly zero. We thus use the approach proposed in (Press, 1988; Golub, 1989; Bjorck, 1996) where singular values w_i are set to 0 when they satisfy the following condition: $w_i < \epsilon\, w_1$), for a threshold ϵ. Similarly, the detection of the well-constrained set of variables is based on the comparison of elements $\mathbf{v}^i(k)$ with a threshold ϵ_1. Of course, the results of the method depend on the choice of the thresholds ϵ and ϵ_1, which may depend themselves on the underlying application (in our experiments, the choice for ϵ_1 was not found to be critical). If thresholds are too large, then there is a possibility that some variables which are sufficiently constrained, will be classified as underconstrained. On the other hand, if they are too small, some underconstrained variables will be classified as having been well estimated, negatively influencing the overall results of an underlying algorithm.

Applications and extensions. The application domain of the proposed approach covers all computer vision algorithms based on linear algebra. In particular, it can be useful in any calibration algorithm based on linear equations (see (Wilczkowiak, 2003a) for an example for plane-based calibration), as well as for reconstruction methods, as explained in the next section.

A main advantage of the proposed approach is simplicity. The test for underconstrained variables needs very small additional computation effort. Indeed, the SVD of the constraint matrix, is usually computed anyway to solve the linear problems. A main drawback is the reliance of the approach on the predefined thresholds. It would be advantageous to incorporate to the method a statistical analysis and uncertainty propagation of the data.

5.2 Multi-linear Reconstruction System

In this section we propose an approach for interactive scene modelling. First, in section 5.2.1 a brief overview of available objects and constraints is given, followed by a general study of how thay can be exploited in the system. Then in section 5.2.2 the algorithm implemented in our system is detailed. Finally, an approach for incorporation of soft and hard constraints into the system is given in section 5.2.3.

5.2.1 Overview. The scene is modelled using points, lines and planes. Three general types of constraints between objects are considered:

Projections. Every known projection of a point or a line gives linear constraints on the 3D coordinates of the corresponding object. Reciprocally, known 3D points or lines give linear constraints on the camera projection matrices. Moreover, geometrical constraints or known plane homographies allow to constrain directly the intrinsic camera parameters.

Bilinear constraints. Incidence, parallelism and orthogonality between two objects $\mathbf{X}$, $\mathbf{Y}$ can be expressed as bilinear forms $\mathsf{F}(\mathbf{X}, \mathbf{Y}) = 0$ (Heuel, 2001; Poulin, 1998; Hartley, 2000). Thus, knowing coordinates of one of the objects induces linear constraints on the other one.

Affine point configurations. Relations like points lying on a parallelogram or symmetry are useful in practice and are linear in terms of all the involved objects.

The scene is represented by a graph whose nodes are objects and whose edges are constraints. For example, four coplanar points will be represented by five nodes (4 points and 1 plane) and 4 incidence constraints (each point with the plane). Except for the affine point configurations, all the geometrical relations incorporated into the system involve two objects of different types and can be used to constrain any of the related objects. These relations are bilinear with respect to coordinates of the two related objects. Thus, they can be used in a linear framework only when at least coordinates of one of the involved objects are known. For example, known 3D positions of points can be used simultaneously to constrain the camera projection matrices (Tsai, 1986) as well as calibrated cameras alone or together with some prior scene information can be used simultaneously to compute 3D points positions (Hartley, 2000; Shum, 1998; Robertson, 2000; Grossmann, 2002). However, if a method proceeds in a single step, it is possible that not all the accessible data is used. For example, it is not possible to impose the orthogonality of two scene directions and use them in the same step to constrain the 3D scene points. Similarly, when a 3D line direction is not known, it is not possible to use the collinearity constraint on the associated points.

There are two reasons why the extraction of the sufficiently constrained variables in the system defined above is crucial for the efficiency of the algorithm. Firstly, at each iteration underconstrained variables may exist. Especially at initial iterations, only few constraints are active: the coordinates of 3D lines and planes are still unknown, thus only the projection and symmetry/parallelogram constraints are active. Also, even when the reconstruction process is in an advanced stage, it is common that some objects are underconstrained due to missing or redundant data. By redundant data we mean that the result is very sensitive to noise (e.g. 2 projections available for a 3D point, but for 2 views with a very small baseline; or, a 3D point defined to be the intersection of a line and a plane, but when these two are near parallel).

Secondly, and contrary to existing approaches, our system allows constraints influencing several objects at once, which means that equation systems to be solved may contain simultaneously well constrained and underconstrained unknowns. Without selecting the solvable subset of unknowns, one would either propagate wrong values to subsequent iterations, or would have to stop the whole algorithm. In the following, we give details on the implementation of our algorithm and explain, how the introduced geometrical dependencies can be treated as soft or hard constraints. The experimental evaluation of the method can be found in (Wilczkowiak, 2003a; Wilczkowiak, 2004). All models illustrating our calibration approach as well as initial values used for the reconstruction approach, were computed using the above method.

5.2.2 Algorithm. In the previous section, we have detailed the constraints used in the system and sketched how they can be exploined for the reconstruction. Let us now consider some practical issues concerning the algorithm's implementation. First, we use precalibrated cameras and do not update their parameters during the reconstruction process, but as suggested in the last section, it is easy to add a re-calibration step to the system. Second, due to the fact that any of the linear constraints used in the system do not involve both points and lines or planes at the same time, point features and line and plane features are computed in two separate steps. Computing line and plane features together allows, if desired, to represent parallel line and plane normal directions by a single vector and use the constraints on lines and planes simultaneously. Finally, we propose this reconstruction algorithm:

Algorithm 1: Iterative Reconstruction Algorithm

while !stop_condition **do**
  for objects=points,lines+planes: **do**
    N:=$\sum_{i=1}^{n}$ nb_of_coordinates(objects[i])
    initialize an empty linear equation system $\mathsf{A}_{0\times N} X_{N\times 1} = \mathsf{B}_{0\times 1}$
    compute the indexing function (bijection) F : idx $\rightarrow (i, j)$; idx $\in [1 \ldots \mathrm{N}]$, where idx is the index in X of the j-th coordinate of the i-th object.
    for all constraint $c[k]$: **do**
      compute $(\mathsf{A}^k_{m_k\times N}, B^k_{m_k\times 1}) :=$ equations($c[k]$.type, $c[k]$.objects)
      add equations to the system: $\mathsf{A} := \begin{bmatrix} \mathsf{A} \\ \mathsf{A}^k \end{bmatrix}$ $\mathsf{B}^k := \begin{bmatrix} \mathsf{B} \\ \mathsf{B}^k \end{bmatrix}$
    end for
    solve $\mathsf{A}X = \mathsf{B}$
    for idx=1 . . . N: **do**
      if variable_computed(idx) **then**
        set $(i, j) := \mathsf{F}$(idx)
        set objects[i].coords[j]:=**X**(idx)
      end if
    end for
  end for
end while

5.2.3 Soft and Hard Constraints. While defining the geometric constraints, the user might wish to enforce some "highly reliable" constraints in an exact manner, instead of incorporating them in a least squares computation. This of course is only possible if these constraints do not contradict one another.

Let us consider a system with m equations, where r of them are to be respected *exactly*. Without loss of generality, we can permute the rows of $\mathbf{A}$ and the coefficients of $\mathbf{B}$ and write:

$$\mathsf{A}_{m\times n} = \begin{bmatrix} \mathsf{A}_e \;\; {}_{r\times n} \\ \mathsf{A}'_{(m-r)\times n} \end{bmatrix};\; \mathsf{B}_{m\times n} = \begin{bmatrix} \mathsf{B}_e \;\; {}_{r\times n} \\ \mathsf{B}'_{(m-r)\times n} \end{bmatrix} \quad (19)$$

where A_e and B_e correspond to equations to be respected exactly and A' B' to equations whose residuals are to be minimised (subject to the exact constraints), i.e.:

> find the vector $\mathbf{X}$ minimizing the function
>
> $$f(\mathbf{X}) = \|\mathsf{A}'\mathbf{X} - \mathbf{B}'\|$$
>
> and satisfying the linear constraints $\mathsf{A}_e\mathbf{X} = \mathbf{B}_e$.

The solution to this problem can be found using constrained optimization techniques, e.g. Langrange multipliers (see e.g. (Gill, 1989)). (Shum, 1998) proposes to use the properties of QR factorisation to solve this problem. We propose another method, based on the SVD (see also (Hartley, 2000), Appendix **5**).

Let us consider the system $\mathsf{A}_e\mathbf{X} = \mathbf{B}_e$. As mentioned above, the set of solutions can be expressed using the SVD of A_e:

$$\begin{array}{c} \mathbf{X}_e = \mathbf{X}_{0e} + \sum_{k=r+1}^{n} \lambda_k \mathbf{v}^k;\; \lambda_k \in \mathcal{R}; \\ \mathbf{X}_{0e} = \mathsf{A}_e^{+}\mathbf{B}_e; \end{array} \quad (20)$$

All vectors $\mathbf{X}_e$ respect the equations $1 \ldots r$ exactly. Now the resolution of the system $\mathsf{A}\mathbf{X} = \mathbf{B}$ is reduced to finding coefficients λ_k such that $\mathbf{X}_e$ is satisfying the equation $\mathsf{A}'\mathbf{X}_e = \mathbf{B}'$ in the optimal way (usually, least squares).

Using equation (20) we can reformulate the problem:

$$\begin{aligned} & \mathsf{A}'\mathbf{X}_e = \mathbf{B}' \\ \Leftrightarrow\; & \mathsf{A}'\mathsf{A}_e^{+}\mathbf{B}_e + \mathsf{A}'\Big(\sum_{k=r+1}^{n} \lambda_k \mathbf{v}^k\Big) = \mathbf{B}' \\ \Leftrightarrow\; & \overbrace{\mathsf{A}'\begin{bmatrix} \mathbf{v}^{r+1} & \cdots & \mathbf{v}^{n} \end{bmatrix}}^{\mathsf{A}''} \begin{bmatrix} \lambda_{r+1} \\ \lambda_{r+2} \\ \vdots \\ \lambda_n \end{bmatrix} = \overbrace{\mathbf{B}' - \mathsf{A}'\mathsf{A}_e^{+}\mathbf{B}_e}^{\mathbf{B}''} \end{aligned}$$

This is again a linear minimisation problem wich can be solved using the SVD decomposition. The undetermined values can be detected like described in the last section. The advantage over using e.g. Lagrange multipliers, is that here, the equation system is of smaller size.

6 EXPERIMENTAL RESULTS

Experiments with synthetic data are presented in (Wilczkowiak, 2004). Their main goal was to study performance of the calibration algorithm in the proximity of singular configurations. In this paper, due to lack of space, we only report on experiments with real images, for indoor and outdoor scenes. These examples correspond to situations where automatic methods are bound to fail: small sets of images are used and occlusions are frequent. Each reconstruction was performed in two steps: first, one or more parallelepipeds were used to calibrate the intrinsic and extrinsic camera parameters; second, scene points and geometric constraints were used for the reconstruction (cf. section 5). Results from single as well as multiple images are shown.

Kio towers. Reconstruction was based on 3 images and 2 calibration primitives. One of the images used for the reconstruction is shown in figure 3 (left). Information used for calibration were: 2 right angles in each tower, zero camera skew, unit aspect ratio and centered principal point. The reconstructed cameras and primitives are shown in figure 3.

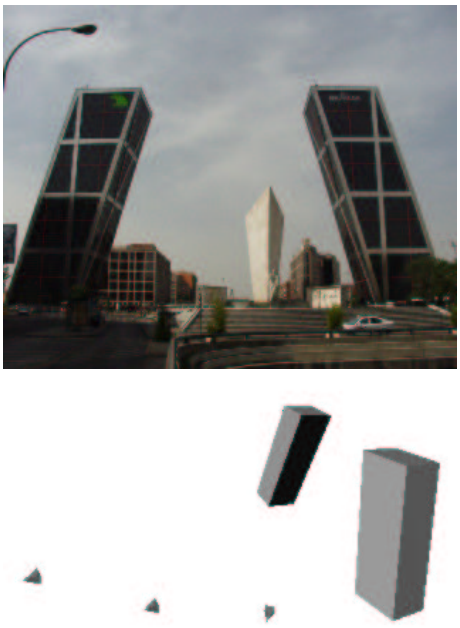
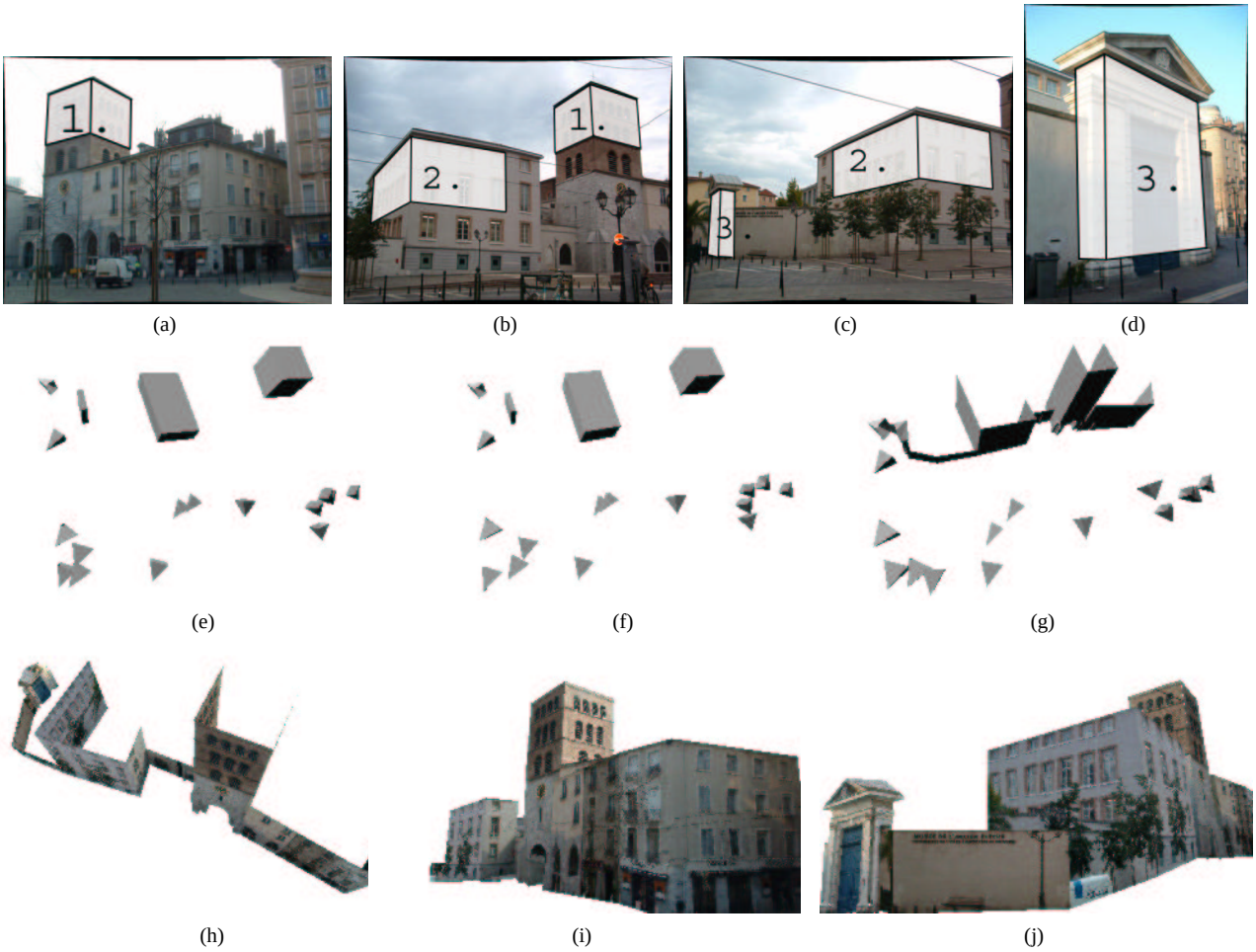


Figure 3: Kio towers in Madrid: original image and reconstructed model and camera poses.

Notre-Dame square: Reconstruction from one image. The image and the calibration parallelepiped are shown in figure 4 (left). Prior information used for calibration were: right parallelepiped angles, zero camera skew and principal point in the image center. The final model is composed of 42 points, 3 parallelepipeds, 4 parallelograms and 4 lines and planes. Rendered views of the model are shown in figure 4.

Figure 4: Notre-Dame square scene in Grenoble, France: the single original image (radial image distortion was corrected off-line) and screen-shots of the 3D model.

Notre-Dame square: Reconstruction from multiple images. The sequence used for the reconstruction is composed of 15 images whose sizes vary from 768×1024 to 960×1280 pixels. Calibration was based on 3 parallelepipeds (shown in figures 5(a) to 5(d)). Prior information used were: right angles for parallelepipeds 1 and 2, zero camera skew, unit aspect ratios and centered principal points for all images. Parallelepiped 3 is relatively small in those images where both parallelepipeds 2 and 3 appear. Consequently, the estimation of its vertices is unstable, and thus information about its intrinsic parameters was not used for calibration. Calibration was performed in two steps. First, the proposed linear factorization approach was applied. Second, the parameters of cameras and parallelepipeds obtained from the



(a) (b) (c) (d)

(e) (f) (g)

(h) (i) (j)

Figure 5: Notre-Dame square scene: (a)–(d) 4 images from the 15 used for the reconstruction. Parallelepipeds used for the reconstruction are marked in white; (e) cameras and parallelepipeds as estimated by the proposed linear factorization method; (f) camera and parallelepiped parameters after non-linear optimization; (g) cameras and 194 model points optimized by an unconstrained non-linear method; (h)–(j) synthetic viewpoints of the textured model.

previous step were non-linearly optimized, by minimizing the reprojection error of vertices in a bundle adjustment.

Then, scene elements were added and reconstructed so that the final model is composed of 194 points, 19 planes and 25 lines. The mean reprojection error over all the model points was about 8 pixels (only linear methods were used for reconstruction). As expected, the largest errors occurred in images calibrated using parallelepiped 3.

For comparison, an unconstrained bundle adjustment, using the Levenberg-Marquardt optimization method, was performed over all the model points and the camera focal lengths. This reduced the reprojection error to 2 pixels. It did not reduce, however, the small artifacts occurring in the final model.

The calibration primitives and cameras reconstructed using the factorization method, the parallelepiped-based non-liner optimization and the point-based non-linear optimization are shown, respectively, in figures 5(e)–5(g). Rendered views of the model reconstructed using the parallelepiped-based calibration are shown in images 5(h)–5(j).

Opposite viewpoints scene. Figure 6 shows the reconstruction of a modern building from 2 images taken from completely opposite viewpoints. The parallelepiped used for calibration is shown in figure 6 (top). In the first image, intersections of lines were computed to obtain the six points required to define a parallelepiped. The parallelepiped and the cameras reconstructed by the factorization algorithm are shown in figure 6 (middle). New viewpoints of the whole model, composed of 32 points, 13 parallelograms and 6 planes are shown in figure 6 (middle and bottom).

Castle. Figure 7 shows input and results of the reconstruction of a castle. The 7 input images were calibrated using mixed approaches (Sturm, 1999a; Wilczkowiak, 2001). This scene raises several difficulties: (i) the images overlap only slightly, decreasing the quality of the camera calibration; (ii) some of the model points are either not visible in any image or visible only in image regions where the camera distortion, which is not taken into account, is important; (iii) the geometrical constraints that can improve the reconstruction are not numerous: vertical edges of the castle are slightly pointing to the center, and its faces are not parallel (see Fig. 7–(b)). Thus geometric constraints are rather used to reconstruct castle elements which are occluded in images.

Fig. 7–(b) shows a map of the castle with the reprojected model points. Points marked with circles are those reconstructed from geometrical constraints only. Experiments were also conducted using a ground plane map of the castle as an additional image for the reconstruction. However it did not significantly change the results. The reconstructed model is shown in Fig. 7–(c).

The second row in Fig. 7 shows results for the first three iterations of the reconstruction. Again, at each iteration the model is enriched by new objects computed using the previously reconstructed set and the newly defined constraints.

Figure 6: Opposite viewpoints scene: (top) the original images used for the reconstruction; (middle) the reconstruction scenario with the computed model and the cameras' positions; (bottom) new viewpoints of the model.

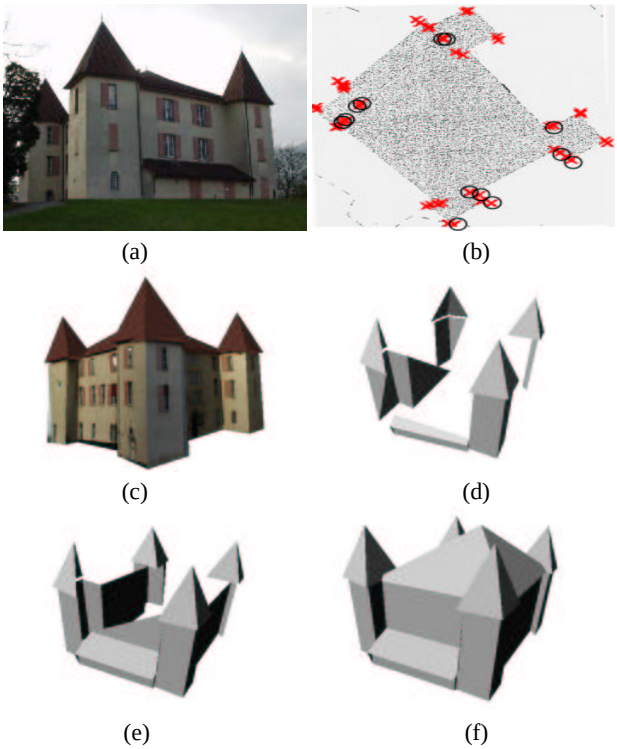


Figure 7: (a) One of the seven images used for the reconstruction; (b) The castle plan; (c) The textured 3D model; (d)-(f) Screenshots of the model at different steps of the reconstruction process.

7 CONCLUSION

We have presented an approach for calibration, pose estimation and 3D model acquisition from several uncalibrated images based on user-provided geometric constraints on the scene. Useful constraints such as parallelism, coplanarity and right angles, can often be nicely modeled via parallelepipeds. Especially, this allows to couple together constraints between several neighboring scene primitives (points, lines, planes), which potentially brings about a higher stability than only using constraints between pairs of primitives. The projections of parallelepipeds already encode the affine structure of the scene. Metric information (length ratios and angles) is then combined with prior information on camera parameters in a self-calibration type approach, performing complete calibration and pose estimation. This is formulated in a factorization framework. The usual problems of missing data and unknown scale factors are dealt with relatively easily, and a satisfying solution can be obtained with already a small number of images and correspondences (starting from 4 correspondences per image pair or 6 per image and parallelepiped). A detailed study on singular cases of this approach is provided in (Wilczkowiak, 2004).

Experiments with real images show that our calibration approach gives excellent initial results for general 3D model reconstruction methods. We believe that an approach such as the one presented here, is a useful tool for easily calibrating cameras using images of unknown though constrained scenes. Also, it allows to efficiently obtain models of the global structure of scenes (including camera pose), which are good starting points for more automatic reconstruction methods.

In (Wilczkowiak, 2003b; Wilczkowiak, 2004), we present an approach for obtaining a minimal parameterization of scenes and cameras, i.e. a parameterization that satisfies all constraints (if this exists). With such a parameterization, initial scene and camera parameters, obtained with the methods presented in this article, can then be optimized via unconstrained bundle adjustment over fewer parameters.

Acknowledgment. This work was partially supported by the European project VISIRE (IST-1999-10756).

REFERENCES

References from Journals:

Buchanan, T., 1988. The twisted cubic and camera calibration. *CVGIP*, 42(1), 130–132.

Caprile, B., Torre, V., 1990. Using vanishing points for camera calibration. *IJCV*, 4, 127–140.

Criminisi, A., Reid, I.D., Zisserman, A., 2000. Single view metrology. *IJCV*, 40(2), 123–148.

Hartley, R., 1997. Self-calibration of stationary cameras. *IJCV*, 22(1), 5–23.

Jelinek, D., Taylor, C., 2000. Reconstruction of linearly parameterized models from single images with a camera of unknown focal length. *IEEE-T-PAMI*, 23(7), 767–774.

Kahl, F., Triggs, B., Åström, K., 2000. Critical motions for auto-calibration when some intrinsic parameters can vary. *Journal of Mathematical Imaging and Vision*, 13(2), 131–146.

Malis, E., Cipolla, R., 2002. Camera self-calibration from unknown planar structures enforcing the multi-view constraints between collineations. *IEEE-T-PAMI*, 4(9), 1268–1272.

Maybank, S., Faugeras, O., 1992. A theory of self calibration of a moving camera *IJCV*, 8(2), 123–151.

Tomasi, C., Kanade, T., 1992. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2), 137–154.

Zisserman, A., Liebowitz, D., Armstrong, M., 1998. Resolving ambiguities in auto-calibration. *Philosophical Transactions of the Royal Society of London, A*, 356(1740), 1193–1211.

References from Books:

Bjorck, A., 1996. *Numerical Methods for Least Squares Problems*. SIAM Publications.

Gill, P.E., Murray, W., Wright, M.H., 1989. *Practical Optimization*. Academic Press.

Golub, G.H., van Loan, C.F., 1989. *Matrix Computation*. The Johns Hopkins University Press.

Hartley, R., Zisserman, A., 2000. *Multiple View Geometry in Computer Vision*. Cambridge University Press.

Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T., 1988. *Numerical Recipes in C*. Cambridge University Press.

References from Conferences:

de Agapito, L., Hartley, R., Hayman, E., 1999. Linear self-calibration of a rotating and zooming camera. *CVPR*, 15–21.

Armstrong, M., Zisserman, A., Beardsley, P., 1994. Euclidean structure from uncalibrated images. *BMVC*, 509–518.

Bondyfalat, D., Bougnoux, S., 1998. Imposing euclidean constraints during self-calibration processes. *SMILE Workshop*, 224–235.

Bondyfalat, D., Papadopoulo, T., Mourrain, B., 2001. Using scene constraints during the calibration procedure. *ICCV*, 124–130.

Boufama, B., Mohr, R., Veillon, F., 1993. Euclidean constraints for uncalibrated reconstruction. *ICCV*, 466–470.

Chen, C., Yu, C., Hung, Y., 1999. New calibration-free approach for augmented reality based on parameterized cuboid structure. *ICCV*, 30–37.

Cipolla, R., Boyer, E., 1998. 3D model acquisition from uncalibrated images. IAPR *Worksh. on Computer Vision*, 559–568.

Debevec, P., Taylor, C., Malik, J., 1996. Modeling and rendering architecture from photographs: a hybrid geometry-and image-based approach. *SIGGRAPH*, 11–20.

Dick, A., Torr, P., Ruffle, S., Cipolla, R., 2001. Combining single view recognition and multiple view stereo for architectural scenes. *ICCV*, 268–274.

Faugeras, O., 1992. What can be seen in three dimensions with an uncalibrated stereo rig? *ECCV*, 563–578.

Grossmann, E., Santos-Victor, J., 2000. Dual representations for vision-based 3D reconstruction. *BMVC*, 516–526.

Grossmann, E., Santos-Victor, J., 2002. Single and multi-view reconstruction of structured scenes. *ACCV*, 93–104.

Hartley, R., 1993. Euclidean reconstruction from uncalibrated views. DARPA–ESPRIT *Workshop on Applications of Invariants in Computer Vision*, 187–202.

Heuel, S., 2001. Points, Lines and Planes and their Optimal Estimation. *DAGM Symposium on Pattern Recognition*, 92–99.

Kosecka, J., Zhang, W., 2002. Video compass. *ECCV*, 476–491.

Liebowitz, D., Zisserman, A., 1998. Metric rectification for perspective images of planes. *CVPR*, 482–488.

Liebowitz, D., Zisserman, A., 1999. Combining scene and autocalibration constraints. *ICCV*, 293–300.

Martinec, D., Pajdla, T., 2002. Structure from many perspective images with occlusions. *ECCV*, 355–369.

McGlone, C., 1995. Bundle adjustment with object space geometric constraints for site modeling. SPIE *Conf. on Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision II*, 25–36.

McLauchlan, P., Shen, X., Manessis, A., Palmer, P., Hilton, A., 2000. Surface-based structure-from-motion using feature groupings. *ACCV*, 699–705.

Pollefeys, M., Van Gool, L., Proesmans, M., 1996. Euclidean 3D reconstruction from image sequences with variable focal lengths. *ECCV*, 31–42.

Pollefeys, M., Van Gool, L., 1997. A stratified approach to metric self-calibration. *CVPR*, 407–412.

Poulin, P., Ouimet, M., Frasson, M.-C., 1998. Interactively Modeling with Photogrammetry. *Eurographics Workshop on Rendering*, 93–104.

Robertson, D.P., Cipolla, R., 2000. An Interactive System for Constraint-Based Modelling. *BMVC*, 536–545.

Rother, C., Carlsson, S., Tell, D., 2002. Projective factorization of planes and cameras in multiple views. *ICPR*, 737–740.

Shum, H.-Y., Han, M. Szeliski, R., 1998. Interactive construction of 3D models from panoramic mosaics. *CVPR*, 427–433.

Sturm, P., Triggs, B., 1996. A factorization based algorithm for multi-image projective structure and motion. *ECCV*, 709–720.

Sturm, P., 1997. Critical motion sequences for monocular self-calibration and uncalibrated euclidean reconstruction. *CVPR*, 1100–1105.

Sturm, P., Maybank, S., 1999a. On plane-based camera calibration: A general algorithm, singularities, applications. *CVPR*, 432–437.

Sturm, P., Maybank, S., 1999b. A method for interactive 3D reconstruction of piecewise planar objects from single images. *BMVC*, 265–274.

Sturm, P., 2000. Algorithms for plane-based pose estimation. *CVPR*, 1010–1017.

Szeliski, R., Torr, P., 1998. Geometrically constrained structure from motion: Points on planes. *SMILE Workshop*, 171–186.

Triggs, B., 1996. Factorization methods for projective structure and motion. *CVPR*, 845–851.

Triggs, B., 1997. Autocalibration and the absolute quadric. *CVPR*, 609–614.

Triggs, B., 1998. Autocalibration from planar scenes. *ECCV*, 89–105.

Tsai, R., 1986. An efficient and accurate camera calibration technique for 3D machine vision. *CVPR*, 364–374.

Wilczkowiak, M., Boyer, E., Sturm, P., 2001. Camera calibration and 3D reconstruction from single images using parallelepipeds. *ICCV*, 142–148.

Wilczkowiak, M., Boyer, E., Sturm, P., 2002. 3D modelling using geometric constraints: A parallelepiped based approach. *ECCV*, 221–236.

Wilczkowiak, M., Sturm, P., Boyer, E., 2003a. The analysis of ambiguous solutions in linear systems and its application to computer vision. *BMVC*, 53–62.

Wilczkowiak, M., Trombettoni, G., Jermann, C., Sturm, P., Boyer, E., 2003b. Scene modeling based on constraint system decomposition techniques. *ICCV*, 1004–1010.

Zhang, Z., 1999. Flexible camera calibration by viewing a plane from unknown orientations. *ICCV*, 666–673.

References from Other Literature:

Wilczkowiak, M., 2004. *3D Modelling From Images Using Geometric Constraints*. PhD thesis, Institut National Polytechnique de Grenoble, France.