CVPR - INT. CONF. ON COMPUTER VISION AND PATTERN RECOGNITION, HILTON HEAD, PP. 706-711, JUNE 2000. EXTENDED VERSION.

Algorithms for Plane-Based Pose Estimation

Peter Sturm
INRIA Rhône-Alpes
655 Avenue de l'Europe
38330 Montbonnot St Martin, France
Peter.Sturm@inrialpes.fr

Abstract

We present several methods for the estimation of relative pose between planes and cameras, based on projections of sets of coplanar features in images. While such methods exist for simple cases, especially one plane seen in one or several views, the aim of this paper is to propose solutions for multi-plane multi-view situations, possibly with little overlap. We propose a factorization-based method for the general case of n planes seen in m views. A mechanism for computing missing data, i.e. when one or several of the planes are not visible in one or several of the images, is described. Finally, a bundle adjustment procedure is developed that optimizes camera and plane pose as well as camera calibration.

1 Introduction

The work presented in this paper is part of a project on multi-view 3D modeling based on scene regularities. Scene regularities like coplanarity of points or lines, perpendicularity and parallelism of lines or planes etc. are increasingly used for interactive 3D modeling of man-made scenes [1, 2, 3, 12, 14, 16]. Typically, the entire scene (very often a building) is depicted by hand in one or several images; the geometric constraints representing scene regularities allow then to obtain a 3D reconstruction of the scene. This approach is feasible and gives very good results if the scene consists of a limited number of "primitives" and if its geometry can be described well enough by geometric constraints like the ones mentioned.

If the environment to be modeled is large and cluttered, it is usually not feasible to depict all the primitives needed for a 3D model. Also, useful geometric constraints might very often only be provided for a fraction of the environment. In such circumstances, one natural solution for 3D modeling is triangulation, based on feature correspondences obtained by image matching. Beside the matching, camera calibration and relative camera pose have to be obtained in some way (if a metric model of the scene is desired). A complete automatization of this process is of course desirable, but it is questionable if current systems are performing well enough for cluttered large scenes. Also, there will always persist a certain failure rate; so, a user might prefer to trade a limited amount of interaction for a higher reliability of the results.

We follow a different approach, as described in the following. Given a set (or a sequence) of images of an environment, we first want to use scene regularities (and associated features depicted in images by a user) to calibrate the views and estimate their relative pose. Once this is achieved, the calibration and pose information give us multi-view constraints for automatically matching and triangulating other features than those used to capture the scene regularities.

Attractive "primitives" for calibration and pose estimation are planar objects with known metric structure: each image of such an object provides two constraints on calibration and, if calibration can be fully determined, relative pose up to two solutions in general [6]. Especially rectangles are very useful since determining their metric structure is done by simply measuring their edge lengths and since they abound in man-made environments.

Pose estimation from planar objects turns out to be rather harder than camera calibration: calibration constraints based on the projection of planar objects with known metric structure [8, 15, 18, 19, 20] can be accumulated over many images. As for pose estimation however, the goal is to obtain relative camera (and plane) pose in a *global* reference frame: estimation of relative pose of two cameras seeing the same plane is rather easy (see e.g. [6] and references therein for algorithms), but estimating simultaneously the pose of m cameras, each one seeing one or only a few of n planes, is not trivial. We are not aware of general methods for this task in the literature, although it is quite probable that developments have been made in the photogrammetric community. However, photogrammetric techniques are often designed for strong camera network geometries or for situations where at least approximate pose information is already available.

The paper is organized as follows. The problem of multi-view multi-plane pose estimation is formulated in §2. A method for the basic one-view one-plane case is given in §3. A factorization-based method for the multi-view multi-plane situation is presented in §4. Global optimization of pose and calibration is briefly described in §5. Experimental results are shown in §6, followed by conclusions.

2 Problem Formulation

The problem at hand is to estimate the relative pose of m cameras and n planes, based on projections of the planes (i.e. features on the planes) in (some of) the cameras. In the following, we only deal with point features, but our ideas may be extended to other features, in particular line segments. We suppose that the cameras are calibrated, e.g. using one of the algorithms described in [15, 20] and that the metric structure of the planes is known, i.e. that the coordinates of points on a plane are known in some Euclidean reference frame attached to the plane.

In the following, we describe the coordinate transformations that lead from 2D point coordinates of points on a plane to the coordinates of their projections in an image. Let $\mathbf{Q}_{jk}$ be the kth point on the jth plane, given by coordinates (X_{jk}, Y_{jk}). Let the position and orientation of the jth plane (in the sequel simply called the plane's *pose*) be given by a rotation matrix S_j and a translation vector $\mathbf{v}_j$ with respect to some global 3D world reference frame, such that the coordinates of $\mathbf{Q}_{jk}$ in that global frame are:

$$\mathbf{Q}^w_{jk} = \begin{pmatrix} \mathsf{S}_j & \mathbf{v}_j \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \begin{pmatrix} X_{jk} \\ Y_{jk} \\ 0 \\ 1 \end{pmatrix}$$

Let the pose of camera i be given by R_i and $\mathbf{t}_i$, such that the coordinates of $\mathbf{Q}_{jk}$ in the local camera frame are:

$$\mathbf{Q}^c_{ijk} = \begin{pmatrix} \mathsf{R}_i & \mathbf{t}_i \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \mathbf{Q}^w_{jk}$$

The camera model used throughout the paper is that of perspective projection, i.e. the coordinates of the projected point are:

$$\begin{aligned} \mathbf{q}_{ijk} &\sim \begin{pmatrix} \mathsf{K}_i & \mathbf{0} \end{pmatrix} \mathbf{Q}^c_{ijk} \\ &\sim \mathsf{K}_i \begin{pmatrix} \mathsf{R}_i\mathsf{S}_j & \mathsf{R}_i\mathbf{v}_j + \mathbf{t}_i \end{pmatrix} \begin{pmatrix} X_{jk} \\ Y_{jk} \\ 0 \\ 1 \end{pmatrix} \end{aligned} \tag{1}$$

where K_i is the *calibration matrix* of view i, which contains its intrinsic parameters.

The aim of the algorithms presented in this paper is to determine the camera and plane pose, i.e. the $\mathsf{R}_i, \mathsf{S}_j, \mathbf{t}_i$ and $\mathbf{v}_j$, from the calibration matrices K_i, the metric structure of the planes, represented by the (X_{jk}, Y_{jk}), and the projections $\mathbf{q}_{ijk}$ of the points. The computations are based on homographies for camera–plane pairs that represent the perspective projections of the planes onto the image planes. The homography H_{ij} for camera i and plane j can be written as[1]:

$$\mathsf{H}_{ij} \sim \mathsf{K}_i \begin{pmatrix} (\mathsf{R}_i\bar{\mathsf{S}}_j)_{3\times 2} & (\mathsf{R}_i\mathbf{v}_j + \mathbf{t}_i)_{3\times 1} \end{pmatrix}$$

where $\bar{\mathsf{S}}_j$ is the 3×2 submatrix of S_j consisting of its first two columns. Since we suppose that calibration is known, we may compute

$$\mathsf{M}_{ij} \sim {\mathsf{K}_i}^{-1}\mathsf{H}_{ij} \sim \begin{pmatrix} (\mathsf{R}_i\bar{\mathsf{S}}_j)_{3\times 2} & (\mathsf{R}_i\mathbf{v}_j + \mathbf{t}_i)_{3\times 1} \end{pmatrix}$$

The algorithms described in the following determine pose using these homographies M_{ij}. The basic constraint used is that the first two columns of any M_{ij} are the first two columns of a rotation matrix, up to scale.

The homographies used are computed from point matches between planes and the images (in our experiments the matches are mainly obtained by hand); we use a linear method analogous to the 8-point method for the fundamental matrix [4].

[1] This is simply the matrix of equation (1), the third column being dropped.

Notations. As already mentioned above, for a 3×3 matrix A, $\bar{\mathsf{A}}$ is the 3×2 submatrix consisting of its first two columns. The sign $\sim$ means equality up to scale (for vectors or matrices). The matrix $\mathtt{I}$ denotes the identity matrix (sometimes with a subscript indicating its size).

3 The Basic Case: One Plane Seen in One View

Let us consider the case of one plane seen in one view. Suppose the view is calibrated and the homography H (we omit the subscripts in this section) has been computed. As shown above, we can compute the matrix

$$\mathsf{M} \sim \left(\left(\mathsf{R}\bar{\mathsf{S}}\right)_{3\times 2} \quad (\mathsf{R}\mathbf{v} + \mathbf{t})_{3\times 1} \right) \quad .$$

Of course, we can only compute *relative* pose, i.e. a rotation matrix T and a vector $\mathbf{w}$ such that:

$$\begin{aligned} \mathsf{T} &= \mathsf{R}\mathsf{S} && (2) \\ \mathbf{w} &= \mathsf{S}^\mathsf{T}\mathbf{v} + \mathsf{S}^\mathsf{T}\mathsf{R}^\mathsf{T}\mathbf{t} && (3) \end{aligned}$$

in which case we have:

$$\mathsf{M} \sim \mathsf{T} \begin{pmatrix} 1 & 0 & \\ 0 & 1 & \mathbf{w} \\ 0 & 0 & \end{pmatrix} \quad .$$

In the absence of noise, the solution for T and $\mathbf{w}$, given M, is simple: first, scale M such that its first two columns have unit norm (there are two solutions for the scale factor, one being the negative value of the other). The first two columns of M are then adopted as the first two columns of T. The third column of T is easily computed as the cross product of the first two columns (plus possibly a scaling by -1 to ensure that $\det \mathsf{T} = +1$). Having solved for T, $\mathbf{w}$ is given as the third column of $\left(\mathsf{T}^\mathsf{T}\right) \mathsf{M}$ (where M is scaled as shown above).

There are two solutions in general (due to the existence of two scale factors for M), one being obtained from the other by:

$$\begin{aligned} \mathsf{T}' &= \mathsf{T} \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ \mathbf{w}' &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \mathbf{w} \end{aligned}$$

. In practice, it is easy to disambiguate between these two, as follows. The two solutions for $\mathbf{w}$ correspond to optical centers on both sides of the plane. Thus, it is sufficient to know which side of a plane is visible (in practice, an individual plane will usually be seen from the same side in all images). We achieve this for example by giving the coordinates of points on the plane (the (X_{jk}, Y_{jk}) introduced above) in a reference frame whose *positive* Z axis (the axis perpendicular to the plane) shows toward the "visibility half-space", and then choosing the pose whose $\mathbf{w}$ has a negative third coefficient, or vice versa.

If noise is present, the matrix M will not be exactly of the form shown above, and we have to determine some "best" T and $\mathbf{w}$, according to some criterion. As usual in this case, the criterion used is the Frobenius norm $\|\cdot\|_F$ for matrices, i.e. the root of the sum of squared matrix coefficients. Concretely, the problem may be formulated as:

$$\min_{\mathsf{T},\mathbf{w},\lambda} \|\lambda\mathsf{M} - \mathsf{T} \begin{pmatrix} 1 & 0 & \\ 0 & 1 & \mathbf{w} \\ 0 & 0 & \end{pmatrix} \|_F^2 \quad \text{subject to} \quad \left(\mathsf{T}^\mathsf{T}\right) \mathsf{T} = \mathtt{I}_3 \quad . \tag{4}$$

It is easy to show, along the lines of [5], that the optimal solution for the rotation T can be obtained independently from λ and $\mathbf{w}$, and that the latter may then be obtained from T.

The optimal solution for T is obtained by solving the following subproblem:

$$\min_{\bar{\mathsf{T}}} \|\bar{\mathsf{M}} - \bar{\mathsf{T}}\|_F^2 \quad \text{subject to} \quad \left(\bar{\mathsf{T}}^\mathsf{T}\right) \bar{\mathsf{T}} = \mathtt{I}_2 \quad . \tag{5}$$

In words, we determine the rotation matrix T whose first two columns are closest to those of M, in the sense of the Frobenius norm. Note that this is different from the formulation chosen by Zhang [20], who finds the rotation matrix closest to the 3×3 matrix consisting of $\bar{\mathsf{M}}$ and a third column computed (more or less) as the cross product of these first two columns. It can be shown that this approach does not solve the original problem (4) optimally.

The problem (5) is easily solved using Singular Value Decomposition (SVD, [9]). Let $\left(\bar{\mathsf{M}}\right)_{3\times 2} = \mathsf{U}_{3\times 2}\Sigma_{2\times 2}\mathsf{V}_{2\times 2}^{\mathsf{T}}$ be the SVD of $\bar{\mathsf{M}}$. The optimal "amputated" rotation matrix $\bar{\mathsf{T}}$ is then:

$$\bar{\mathsf{T}} = \mathsf{U}\mathsf{V}^{\mathsf{T}} \quad .$$

The third column of T may then be computed in the same manner as described above for the noise free case.

Having solved for T, the optimal scale factor λ and vector $\mathbf{w}$ are obtained as:

$$\begin{aligned} \lambda &= \frac{\operatorname{trace}(\bar{\mathsf{T}}^{\mathsf{T}}\ \bar{\mathsf{M}})}{\operatorname{trace}(\bar{\mathsf{M}}^{\mathsf{T}}\ \bar{\mathsf{M}})} = \frac{\sum_{i=1}^{3}\sum_{j=1}^{2} T_{ij}M_{ij}}{\sum_{j=1}^{2} M_{ij}^2} \\ \mathbf{w} &= \left(\mathsf{T}^{\mathsf{T}}\right)\mathsf{M}\begin{pmatrix}0\\0\\\lambda\end{pmatrix} \end{aligned}$$

Again, there are two solutions in general which can be disambiguated as discussed for the noise free case.

We do not claim that the method presented in this section is original, but described it here since it is an important part of the method described in the next section.

4 Factorization-Based Multi-View Multi-Plane Pose

The method of the previous section may be used to determine the relative pose for m cameras observing a single plane or n planes being observed by a single camera, just by applying it for the different camera–plane pairs individually and stitching together the results. However, if more than one camera observe more than one plane, the situation becomes more complicated. In the following, we present a method that simultaneously uses the relative pose information obtained for individual camera–plane pairs to determine global relative pose of cameras and planes. We first assume that all planes are visible in all cameras. The case of missing data, i.e. when one or several planes are not visible in one or several views, is dealt with in §4.3.

In the following, we first compute the rotational part of the camera and plane pose, followed in §4.4 by the translational part.

4.1 Rotational Part of Pose

Let T_{ij} represent the rotational part of the relative pose between camera i and plane j, as computed using the method of the previous section. We may group all equations of type (2) for camera–plane pairs in one single equation system:

$$\underbrace{\begin{pmatrix}\mathsf{T}_{11} & \mathsf{T}_{12} & \cdots & \mathsf{T}_{1n}\\ \mathsf{T}_{21} & \mathsf{T}_{22} & \cdots & \mathsf{T}_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ \mathsf{T}_{m1} & \mathsf{T}_{m2} & \cdots & \mathsf{T}_{mn}\end{pmatrix}}_{\mathsf{W}} = \underbrace{\begin{pmatrix}\mathsf{R}_1\\ \mathsf{R}_2\\ \vdots\\ \mathsf{R}_m\end{pmatrix}}_{\mathsf{R}} \underbrace{\begin{pmatrix}\mathsf{S}_1 & \mathsf{S}_2 & \cdots & \mathsf{S}_n\end{pmatrix}}_{\mathsf{S}} \tag{6}$$

This equation motivates the idea of solving for the R_i and S_j by factorization: The matrix W is (in absence of noise) of rank 3 and its three non zero singular values are all equal. If noise is present, we may estimate the matrix W' with these properties that is closest to W in the sense of the Frobenius norm, as follows. Let $\mathsf{W} = \mathsf{U}\Sigma\mathsf{V}^{\mathsf{T}}$ be the SVD of W. Let U' (V') be the matrix consisting of the first three columns of U (V). The optimal W' is then given by $\mathsf{W}' = \mathsf{U}'\mathsf{V}'^{\mathsf{T}}$.

Since U' and V' have the same dimensions as R and S in equation (6), we may try to extract the rotation matrices R_i and S_j from them. The factorization does not guarantee that the 3×3 submatrices of U' and V' are valid rotation matrices. Thus, we determine the R_i and S_j as the 3×3 rotation matrices that are closest to the according submatrices in U' and V'. How to determine the rotation matrix that is closest to a general matrix is described in [5, 20].

One issue to discuss is the possibility of ambiguities in the factorization, i.e. the existence of matrices A such that $(\mathsf{U}'\mathsf{A})\left(\mathsf{A}^{-1}\mathsf{V}'^{\mathsf{T}}\right)$ is a valid solution for our problem. Since the two matrices resulting from the factorization have to be collections of rotation matrices, as shown in equation (6), it can be shown that the only possible ambiguities correspond to A being a rotation matrix. This however is not a problem here, since naturally the ensemble of rotation matrices R_i and S_j can only be determined up to a global rotation.

Another important issue for factorization methods is numerical condition. Here, the matrix to be factorized is a collection of rotation matrices, thus the matrix is automatically very well balanced, i.e. its coefficients are in average of the same magnitude. Also, the three non zero singular values (in the noise free case) are equal, meaning that even in the noisy case the condition of the matrix should be good.

4.2 A Variant for Factorization

The factorization described above was performed on a matrix consisting of entire 3×3 rotation matrices T_{ij}. An alternative is to perform the factorization using only the first two columns of the T_{ij}. In this case, we have:

$$\begin{pmatrix} \bar{\mathsf{T}}_{11} & \bar{\mathsf{T}}_{12} & \cdots & \bar{\mathsf{T}}_{1n} \\ \bar{\mathsf{T}}_{21} & \bar{\mathsf{T}}_{22} & \cdots & \bar{\mathsf{T}}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\mathsf{T}}_{m1} & \bar{\mathsf{T}}_{m2} & \cdots & \bar{\mathsf{T}}_{mn} \end{pmatrix} = \begin{pmatrix} \mathsf{R}_1 \\ \mathsf{R}_2 \\ \vdots \\ \mathsf{R}_m \end{pmatrix} \begin{pmatrix} \bar{\mathsf{S}}_1 & \bar{\mathsf{S}}_2 & \cdots & \bar{\mathsf{S}}_n \end{pmatrix}$$

The matrix to be factorized is of size $3m \times 2n$ instead of $3m \times 3n$. After the factorization, the third columns of the S_j may be computed from the $\bar{\mathsf{S}}_j$ as described in §3. Analogously, the factorization could be performed on a $2m \times 3n$ matrix, obtained by dropping a row in each of the T_{ij}.

This way, the SVD could be performed (even) quicker, for the small price of subsequent computation of the third columns of the resulting rotation matrices.

4.3 Computing Missing Data

Our method suffers, as all factorization approaches (e.g. [17]), from the problem of missing data. As mentioned above, in practice we will very often meet the case where several planes are not visible in several cameras, thus the matrix to factorize is not entirely defined. Solutions to this problem have been proposed [7, 11, 17]; these are either of an ad hoc or heuristic nature or rely on an initialization by some means. We propose another ad hoc approach for our problem. Our situation is not too bad, since the missing entries in the matrix to be factorized are 3×3 rotation matrices, thus providing some useful constraints for their determination.

The computation of the missing rotation T_{ij} between a camera i and a plane j is based on the following observation. If we know, for some i' and j', the rotations $\mathsf{T}_{i'j'}$, $\mathsf{T}_{i'j}$ and $\mathsf{T}_{ij'}$, then we can directly compute the missing rotation as:

$$\mathsf{T}_{ij} = \mathsf{T}_{ij'} \left(\mathsf{T}_{i'j'}^{\mathsf{T}}\right) \mathsf{T}_{i'j} \tag{7}$$

as can be seen from equation (2).

If more than one such combination are available, we can think about computing T_{ij} as their "average". To do so, we simply add up the individual estimations of T_{ij}, giving a matrix A. The average solution for T_{ij} may then be determined as the rotation matrix that approximates A best, in the sense of the Frobenius norm (see [5, 20]).

Computation of missing data has usually to be done in a cumulative manner, i.e. some of the T_{ij} can only be computed using other matrices that were missing at the outset but have been computed as shown above. This is illustrated in figure 1.

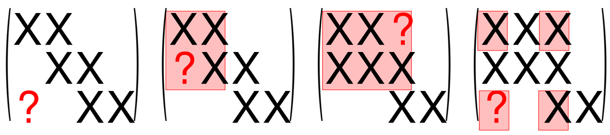

Figure 1: Computation of missing data. (a) There is no possibility to compute the element represented by a question mark from the existing elements using equation (7). (b) The element shown can be computed using those emphasized in red. (c) Two combinations are available to compute the element shown. (d) The element that could not be computed at the beginning, can now be determined using the element computed in the previous step.

4.4 Translational Part of Pose

Having computed the rotational parts of camera and plane pose, the translational parts may be determined as follows. Let $\mathbf{w}_{ij}$ represent the translational part of the relative pose between camera i and plane j, as computed using the method in §3. From equation (3), we have:

$$\mathbf{w}_{ij} = \mathsf{S}_j^\mathsf{T}\mathbf{v}_j + \mathsf{S}_j^\mathsf{T}\mathsf{R}_i^\mathsf{T}\mathbf{t}_i$$

Since we know the S_j, we may compute

$$\mathbf{w}'_{ij} = \mathsf{S}_j\mathbf{w}_{ij} = \mathbf{v}_j + \mathsf{R}_i^\mathsf{T}\mathbf{t}_i$$

Based on this equation, we design a cost function for the estimation of the vectors $\mathbf{v}_j$ and $\mathbf{t}'_i = \mathsf{R}_i^\mathsf{T}\mathbf{t}_i$:

$$c = \sum_{i,j}\left(\mathbf{w}'_{ij} - \mathbf{v}_j - \mathbf{t}'_i\right)^2 \tag{8}$$

where the summation is done over all available camera–plane pairs. In order to minimize the criterion (8), we compute its partial derivatives in the unknowns. The partial derivatives with respect to the kth coefficient of $\mathbf{v}_j$ and the pth coefficient of $\mathbf{t}'_i$ respectively are:

$$\frac{\partial c}{\partial v_{jk}} = 2\sum_i\left(w'_{ijk} - v_{jk} - t'_{ik}\right) \tag{9}$$

$$\frac{\partial c}{\partial t'_{ip}} = 2\sum_j\left(w'_{ijp} - v_{jp} - t'_{ip}\right) \tag{10}$$

The criterion (8) may be minimized by solving for the common roots of the partial derivatives. From equations (9) and (10), we see that this may be done by solving the following simple linear equation system (shown for the case where all planes are seen in all views):

$$\left(\begin{array}{ccc|ccc} m\mathtt{I} & & & \mathtt{I} & \cdots & \mathtt{I} \\ & \ddots & & \vdots & \ddots & \vdots \\ & & m\mathtt{I} & \mathtt{I} & \cdots & \mathtt{I} \\ \hline \mathtt{I} & \cdots & \mathtt{I} & n\mathtt{I} & & \\ \vdots & \ddots & \vdots & & \ddots & \\ \mathtt{I} & \cdots & \mathtt{I} & & & n\mathtt{I} \end{array}\right)\left(\begin{array}{c} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_n \\ \hline \mathbf{t}'_1 \\ \vdots \\ \mathbf{t}'_m \end{array}\right) = \left(\begin{array}{c} \sum_i \mathbf{w}'_{i1} \\ \vdots \\ \sum_i \mathbf{w}'_{in} \\ \hline \sum_j \mathbf{w}'_{1j} \\ \vdots \\ \sum_j \mathbf{w}'_{mj} \end{array}\right)$$

We use a special method to solve this sparse equation system, as described in [13]. The solution is of course only unique up to a translation: adding a 3-vector to all the $\mathbf{v}_j$ and subtracting it from all the $\mathbf{t}'_i$ does not affect the criterion (8).

4.5 Complete Algorithm

1. Compute homographies between planes and images.
2. If the cameras are not calibrated yet, calibrate them using one of the methods in [8, 15, 18, 19, 20].
3. Estimate relative pose between pairs of planes and cameras as described in §3 (take care to reveal the good one among the two possible solutions).
4. Compute missing data as described in §4.3.
5. Estimate the rotational part of global relative pose by factorization as described in §4.1.
6. Estimate the translational part of pose as described in §4.4.
7. Optional, but recommended: optimization of camera and plane pose and simultaneously of camera calibration (see §5).

5 Optimization

Once pose is estimated for all cameras and planes, with respect to a common global reference frame, it may be optimized in a bundle adjustment manner. We have implemented such an algorithm. The optimization is carried out for all pose parameters but also for the cameras' intrinsic parameters, including a coefficient for radial distortion. Our implementation is rather flexible in that it handles varying intrinsic parameters (focal length, principal point and distortion coefficient).

The criterion being minimized is the sum of squared reprojection errors. As it is often the case with bundle adjustment methods, the normal equations being solved during non linear optimization have a special sparse structure, as sketched in figure 2 for the case of 6 cameras and 2 planes. We have adapted the sparse solution method proposed in [13] for basic photogrammetric blocks to our case, which makes an efficient (in time and memory) optimization possible.

The optimization is done using a Levenberg-Marquardt type method [9]. The implementation requires procedures for the computation of the cost function to be minimized and its partial derivatives. These are obtained in a very straightforward manner from the projection equations and thus not shown here.

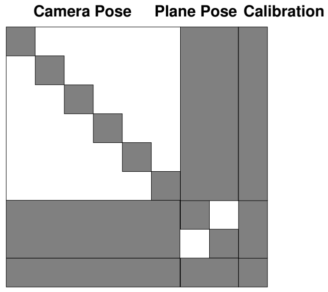

Figure 2: Structure of normal equations to be solved during optimization.

6 Experimental Results

We have tested our methods with several image sequences of different types. First, images of calibration grid were used to test the correctness of the algorithms and evaluate their performance with respect to the number of images used. Second, planar patterns printed on paper were attached to all the walls of a room. This scene is a test for the our methods in the case of a high amount of missing data. The third image sequence is of the same type as the second one, however the planar objects used for calibration and pose estimation were part of the scene (rectangular objects like windows, doors, computer screens etc.).

6.1 Calibration Grid

Images of a calibration grid (see figure 3) were taken by a Canon MV-1 Camcorder. For different zoom positions, 4 images each were taken from different positions. The input to our methods were the coordinates of circular targets in each of the three planes of the calibration grid, and the coordinates of the targets extracted in the images. So, for each zoom setting, a total of 4×3 homographies could be computed. From these, the camera was calibrated and pose was estimated using the methods in §3 and §4. Optimization was performed subsequently.

In figure 4, some results are presented for the zoom position corresponding to shortest focal length (and largest optical distortion). The upper two error curves show the absolute errors (in degrees) of the angles between the three planes of the calibration grid, derived from the estimated pose. With the minimum case of a single view, the error is about $1.4°$ for both the "linear" method (§4) and the subsequent optimization ("Linear+LM" in the graph). Adding more views leads to an error of about $1°$ for the linear method (which seems to be a limit here, maybe due to the neglection of optical distortion) and a linear decrease of the error after optimization, reaching a tenth of a degree when four views are used.

The lower two curves show the average distance errors for the full 3D reconstruction of the calibration grid. Since we know the coordinates of the targets in each of the three planes of the object, and we estimate the pose of the planes, we can obtain a full 3D reconstruction, i.e. full 3D coordinates of the targets. The ground truth for the 3D coordinates is known from the manufacturer. Thus, we can compute how closely the reconstruction matches the ground truth. This is done by computing the "best" rigid transformation between the reconstruction and the ground truth [5] and measuring the residual. The error is practically constant and equal to a tenth of a percent, regardless of the number of images and optimization.

Figure 3: The calibration grid used.

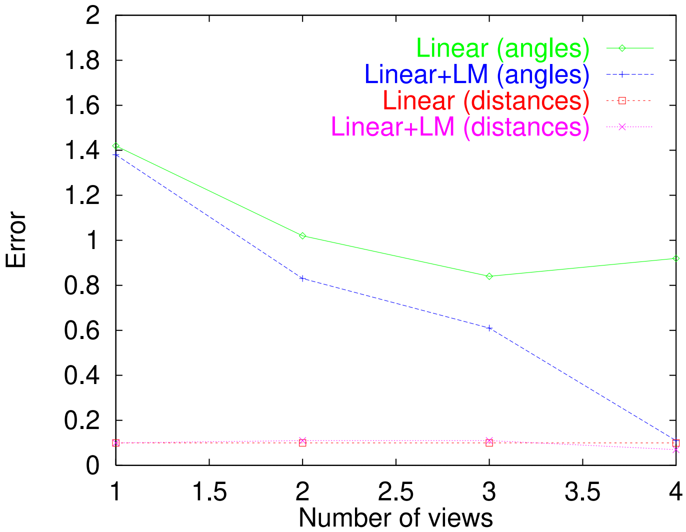

Figure 4: Errors of pose estimation. For the angles (upper curves), absolute errors are shown, given un degrees. For the distances (lower curves), relative errors are shown, given in percent.

6.2 An Indoor Scene

We took a set of about 400 images of an indoor scene (see some examples in figure 5). We measured the edges of 14 rectangular objects in the scene (windows, drawers, a door, blackboard, computer screens, etc., cf. figure 7), thus obtaining their metric structure. In 151 of the images, one or more of the planar objects were visible and in 84 of those, two or more objects. In these images, the 4 corners of the objects seen were marked by hand. This constituted the entire input to our algorithms.

In a first step, the calibration method of [15] was applied to calibrate the 84 views simultaneously. Then, relative pose between each view and the objects seen in it was computed using the method of §3. From the totality of $84 \times 14 = 1176$ image–plane pairs, the relative pose of 218 pairs could be determined from the available images, i.e. the amount of missing data was about 81 %. The missing relative "poses" were computed using the method of §4.3. Global pose was then estimated using the methods of §4 and refined, together with camera calibration (including radial distortion), using the bundle adjustment procedure of §5. The resulting pose of the planar objects was used to obtain a textured VRML model (two renderings are shown in figure 7).

Qualitatively, the reconstruction captures very well the shape of the room in which the images where taken. We have to admit that the accuracy of the reconstruction is not very high: angles between neighboring planar objects (i.e. the angles between the infinite planes supporting the objects) are in average estimated with an error of about 6° (supposing that the walls form

Figure 5: Images of an indoor scene.

90° angles). This is certainly not enough for photogrammetric standards. However, the global pose is certainly good enough to think of using it for wide-baseline matching using adaptive windows: knowing the relative pose between views, matching windows can be transferred between them via projective mappings computed from the pose (and calibration) and based on the assumption of locally planar object surface. Initial matching experiments are encouraging for the direction we want to follow (cf. §7).

Overall, we consider this experiment as a really hard test: the input data is rather minimal (4 points per plane) and poor (some of the objects were not really planar, extraction of features in the images was quite inaccurate); the imaging geometry is very weak ($\sim$ 80 % of missing data); no special illumination was used, etc. So, the accuracy of our results might be as good as one might expect under these conditions.

A similar experiment was performed using patterns printed on paper and attached to the walls of another room. In each of the 32 images taken, two out of a total of twelve patterns (three per wall) were visible (cf. figure 6), i.e. the amount of missing data was again over 80 %. The average error of angles between neighboring patterns was here about 3°, thus half of the error of the previous experiment, which is certainly owing to the higher accuracy of feature extraction (which although was not perfect since in nearly all the images one of the two patterns is slightly out of focus and there was no special lighting).

Figure 6: Two of the images used in the second experiment.

7 Conclusion and Perspectives

We have presented methods for plane-based pose estimation. Beside a method for the basic one-view one-plane case, a factorization-based method for the multi-view multi-plane case was presented.

Our experimental results suggest that our method may be applied successfully even when the amount of missing data is very high. In "calibration scenarios" the estimated pose can certainly be used as starting point for optimizing calibration and pose. However, the global goal of our work is not calibration but the 3D reconstruction of complicated man-made environments. Our thread of thought is that the process should be initialized by a limited and tolerable amount of user interaction, followed by automatic processes. The type of user interaction described in this paper (depicting some salient objects in the images) allows for a good camera calibration and approximate to good global pose estimation. Especially, the recovered pose is good enough to be used for wide baseline matching by adaptive windows (according to initial experiments). This is what we are currently working on. Our hope (and conviction) is that a few (maybe around ten) additional matches per image (beside the hand picked ones) should be enough to increase the quality of the pose by a sufficient amount in order to make e.g. voxel coloring approaches [10] for 3D reconstruction feasible.

References

[1] R. Cipolla, D.P. Robertson, E.G. Boyer, "Photobuilder – 3D models of architectural scenes from uncalibrated images," *Conference on Multimedia Computing and Systems*, pp. 25-31, June 1999.

[2] A. Criminisi, I. Reid, A. Zisserman, "Duality, Rigidity and Planar Parallax," *ECCV*, pp. 846-861, June 1998.

[3] P.E. Debevec, C.J. Taylor, J. Malik, "Modeling and Rendering Architecture from Photographs: a Hybrid Geometry-and Image-Based Approach," *SIGGRAPH*, August 1996.

[4] R. Hartley, "In Defence of the 8-Point Algorithm," *ICCV*, pp. 1064-1070, June 1995.

[5] B.K.P. Horn, H.M. Hilden, S. Negahdaripour, "Closed-Form Solution of Absolute Orientation Using Orthonormal Matrices," *Journal of the Optical Society of America A*, Vol. 5, pp. 1127-1135, 1988.

[6] R.J. Holt, A.N. Netravali, "Camera Calibration Problem: Some New Results," *CVIU*, Vol. 54, No. 3, pp. 368-383, 1991.

[7] D. Jacobs, "Linear Fitting with Missing Data: Applications to Structure-from-Motion and to Characterizing Intensity Images," *CVPR*, pp. 206-212, June 1997.

[8] R.K. Lenz, R.Y. Tsai, "Techniques for Calibration of the Scale Factor and Image Center for High Accuracy 3-D Machine Vision Metrology," *PAMI*, Vol. 10, No. 5, pp. 713-720, 1988.

[9] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes in C - The Art of Scientific Computing,* 2nd edition, Cambridge University Press, 1992.

[10] S.M. Seitz, C.R. Dyer, "Photorealistic Scene Reconstruction by Voxel Coloring," *CVPR*, pp. 1067-1073, June 1997.

[11] H.Y. Shum, K. Ikeuchi, R. Reddy, "Principal Component Analysis with Missing Data and its Application to Polyhedral Object Modeling," *PAMI*, Vol. 17, No. 9, pp. 854-867, 1995.

[12] H.-Y. Shum, R. Szeliski, S. Baker, M. Han, P. Anandan, "Interactive 3D Modeling from Multiple Images Using Scene Regularities," *SMILE Workshop, Freiburg, Germany*, pp. 236-252, June 1998.

[13] C.C. Slama (Ed.), "Manual of Photogrammetry," Fourth Edition, American Society of Photogrammetry and Remote Sensing, 1980.

[14] P. Sturm, S.J. Maybank, "A Method for Interactive 3D Reconstruction of Piecewise Planar Objects from Single Images," *BMVC*, pp. 265-274, September 1999.

[15] P. Sturm, S.J. Maybank, "On Plane-Based Camera Calibration: A General Algorithm, Singularities, Applications," *CVPR, Fort Collins, CO*, pp. 432-437, June 1999.

[16] R. Szeliski, P.H.S. Torr, "Geometrically Constrained Structure from Motion: Points on Planes," *SMILE Workshop, Freiburg, Germany*, pp. 171-186, June 1998.

[17] C. Tomasi, T. Kanade, "Shape and Motion from Image Streams under Orthography: A Factorization Method," *International Journal on Computer Vision*, Vol. 9, No. 2, pp. 137-154, 1992.

[18] R.Y. Tsai, "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," IEEE *Journal of Robotics and Automation*, Vol. 3, No. 4, pp. 323-344, 1987.

[19] G.-Q. Wei, S.D. Ma, "A Complete Two-Plane Camera Calibration Method and Experimental Comparisons," *ICCV*, pp. 439-446, 1993.

[20] Z. Zhang, "Flexible Camera Calibration By Viewing a Plane From Unknown Orientations," *ICCV*, pp. 666-673, 1999.

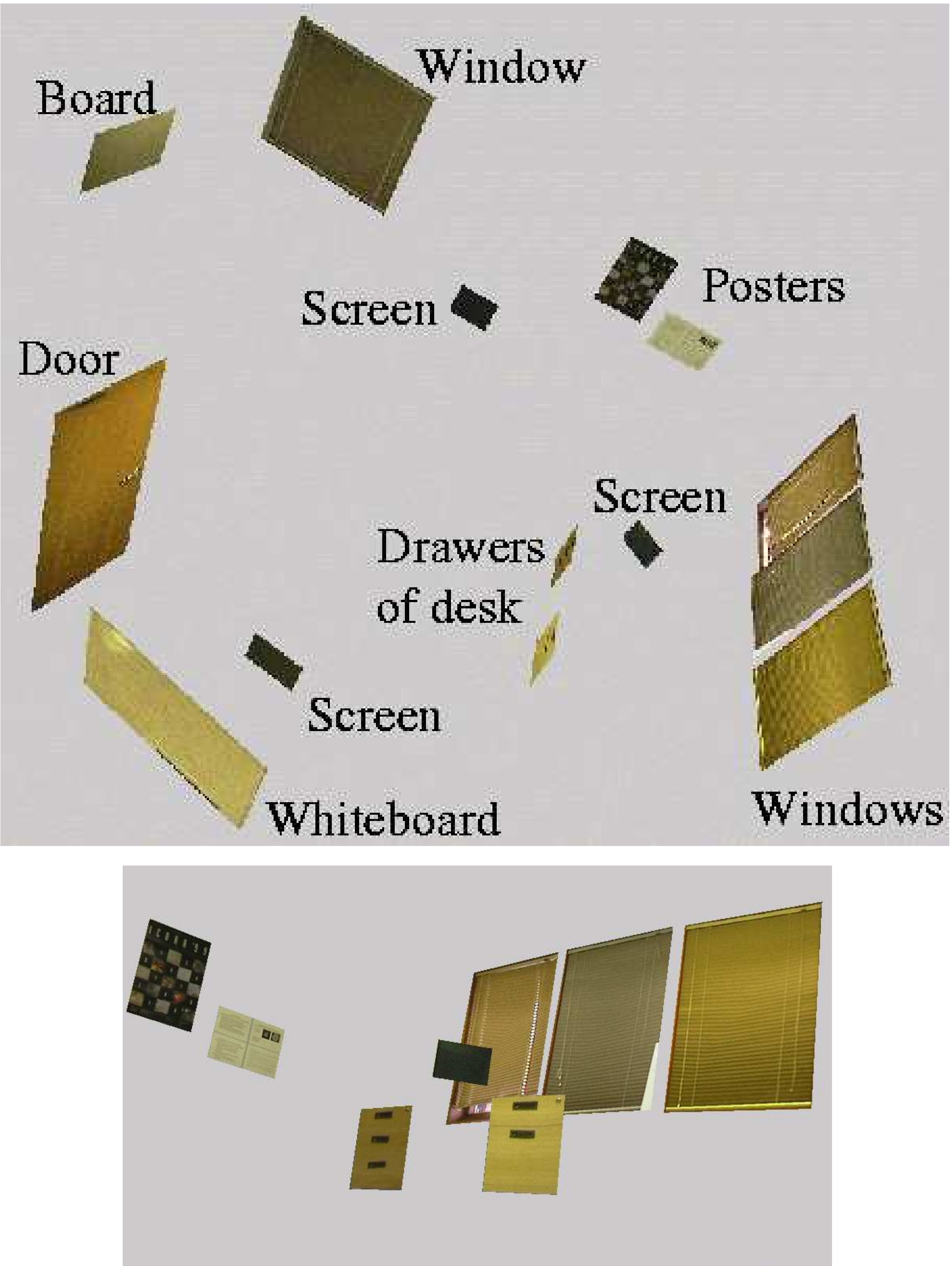

Figure 7: Renderings of a textured 3D model of the planar objects used for calibration and pose estimation.