

The Analysis of Ambiguous Solutions in Linear Systems and its Application to Computer Vision

Marta Wilczkowiak * Peter Sturm Edmond Boyer
Movi–Gravir–Inria Rhône-Alpes, 655 Avenue de l’Europe
38330 Montbonnot, France
Firstname.Surname@inrialpes.fr

Abstract

The success of practical systems based on computer vision algorithms may critically depend on the correct treatment of degenerate configurations and missing data. However, even if theoretical basis exist, it appears that few efforts have been made to solve these problems in practice. In this paper, we address estimation problems based on linear constraints and for which degenerate cases are in principle easy to detect. Many algorithms, in particular in computer vision, either do not detect them or simply stop and produce no output. In many cases however, degenerate situations nevertheless allow a reliable estimation of a subset of the unknowns. We present a practical approach for splitting the variable set of a degenerate linear system into underconstrained and well defined variables. It means that even if the system as a whole is underconstrained, we can still extract useful information and give the correct solution for a subset of the unknowns. Our method is based on singular value decompositions (SVD). Using a very simple analysis of the matrix nullspace, it becomes easy and fast to split the variables into uniquely defined and ambiguous ones. To illustrate our approach, we present its applications to a novel iterative 3D reconstruction algorithm as well as to plane-based camera calibration.

1 Introduction

Recovering 3D scenes from 2D images is a non-trivial task and a lot of consideration has to be given to the issues of degenerate configurations and missing data. Theoretical work has been done to study degenerate configurations for e.g. camera calibration and autocalibration [14, 9]. Even if the theoretical basis is known, it is not always obvious how to detect and deal with these problems in practice.

In this paper, we focus on algorithms based on linear constraints, which are very common in computer vision. Especially, we refer to linear approaches for camera (auto-) calibration [1, 15, 17] 3D scene reconstruction [6, 8, 10, 13, 11, 3]. The problem of detecting and dealing with critical configurations and underconstrained systems was addressed in some of these papers [6, 10, 13, 15]. However, the proposed approaches will at best produce no output when degenerate cases occur.

We present a practical and flexible method which allows to extract, from underconstrained systems, the variables that are sufficiently well defined, without making any assumptions on the data. This results in the optimal use of the input information and also limits the risk of accepting erroneous solutions. Although our method is based on basic properties of the matrix Singular Value Decomposition (SVD), we have not found references to similar approaches in the literature of numerical analysis [5, 2, 12].

*This work was supported by the project IST-1999-10756, VISIRE.

The paper is organized as follows. Section 2 introduces definitions that are used in the rest of the paper. Section 3 explains how to detect well defined unknowns. In section 4, we present the results of the application of the proposed analysis to plane-based calibration [15, 17]. In section 5, we present a novel constraint-based iterative reconstruction system based on the proposed splitting method. The principle of this reconstruction system is similar to some existing approaches [8, 11] but, as will be shown, it increases both flexibility and efficiency of the reconstruction process.

2 Preliminaries

We make no distinction between 3D objects and their vector or matrix representations, which are written in bold letters, e.g. \mathbf{X} for a point \mathbf{X} . The columns respectively rows of a matrix \mathbf{A} are denoted by \mathbf{A}^i respectively \mathbf{A}_i . \mathbf{A}^\dagger stands for the pseudoinverse of matrix \mathbf{A} .

We model the camera using perspective projection. The projection transformation is represented by a matrix $\mathbf{P}_{3 \times 4} \sim \mathbf{KR}(\mathbf{I}_{3 \times 3} \mid -\mathbf{t})$, such that the image \mathbf{x} of a 3D point \mathbf{X} is $\mathbf{x} \sim \mathbf{PX}$ (\sim stands for equality of homogeneous vectors or matrices). \mathbf{R}, \mathbf{t} are the camera rotation and translation. The intrinsic camera parameters matrix \mathbf{K} and the image of the absolute conic (IAC) $\omega \sim \mathbf{K}^{-T}\mathbf{K}^{-1}$ are:

$$\mathbf{K} = \begin{pmatrix} \tau f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \omega \sim \begin{pmatrix} 1 & 0 & -u_0 \\ 0 & \tau^2 & -\tau^2 v_0 \\ -u_0 - \tau^2 v_0 & \tau^2 f^2 + u_0^2 + \tau^2 v_0^2 & \end{pmatrix}. \quad (1)$$

We consider 4 intrinsic parameters: the focal length f , the aspect ratio τ , and the principal point (u_0, v_0) . The skew is considered to be 0.

3 Detecting the Well-Defined Subset of Unknowns

Consider the following linear equation system:

$$\mathbf{AX} = \mathbf{B},$$

and assume that the solution for \mathbf{X} is not unique. We want then to determine if a subset of coefficients of \mathbf{X} can nevertheless be unambiguously estimated. Our approach is based on the analysis of the matrix \mathbf{A} nullspace. To this purpose, we use the Singular Value Decomposition of \mathbf{A} [5, 2, 12]:

$$\mathbf{A} = \mathbf{U}\mathbf{W}\mathbf{V}^\top,$$

where the matrices \mathbf{U} and \mathbf{V} are orthogonal and \mathbf{W} is diagonal, with the singular values w_i of \mathbf{A} on its diagonal, in decreasing order. The nullspace of \mathbf{A} , $\Phi(\mathbf{A})$, is spanned by the columns $\mathbf{v}^i, i \in \{r+1, \dots, n\}$ of the matrix \mathbf{V} where i is such that $w_i = 0$. Let \mathbf{X}_0 be any vector satisfying the equation system. From the definition of the nullspace, it follows that all vectors \mathbf{X} satisfying the equation system can be written as:

$$\mathbf{X} = \mathbf{X}_0 + \sum_{i=r+1}^n \lambda_i \mathbf{v}^i, \lambda_i \in \mathbb{R}.$$

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:

$$\forall i \in \{r+1, \dots, n\} : \mathbf{v}^i(k) = 0. \quad (2)$$

Hence, all variables $x_k = \mathbf{X}(k)$ corresponding to rows \mathbf{r}_k of $\Phi(\mathbf{A})$ with $\|\mathbf{r}_k\| = 0$, are computed unambiguously, and that their value is $x_k = \mathbf{X}_0(k)$. Geometrically, this means that the axis of the solution space \mathbb{R}^n associated to such well defined variable is orthogonal the nullspace.

Thus, using equation (2), it is in principle straightforward to split the unknowns of the system into well-defined and ambiguous ones. Note that this relies on 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 [5, 2, 12] where singular values w_i are set to 0 when they satisfy the following condition: $w_i < tol = \varepsilon \max(\text{size}(\mathbf{A})) w_1$, for a given 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 given threshold ε_1 .

Of course, the results of the method depends on the choice of the thresholds ε and ε_1 , which may depend themselves on the underlying application¹. If a threshold is too large, then there is a possibility that some variables which are well defined 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, disadvantageously influencing the overall results of the underlying algorithm. We are thus currently investigating ways of incorporating an analysis based on uncertainty estimates for the input data.

4 Application to Plane-Based Calibration

We applied our method to the plane-based calibration algorithm of [15, 17]. In the following, we briefly describe the algorithm's principles and then show some results.

4.1 Outline of the algorithm

The perspective projection to the image plane, of points on some plane in 3D, is described by a 3×3 homography H that depends on the relative positions of the camera and the plane and the camera's intrinsic parameters. The homography can be estimated from at least four point correspondences. Every known plane-to-image homography leads to 2 equations on the elements of the matrix ω (cf. (1) in section 2):

$$\omega_{11}\mathbf{a}_1 + \omega_{22}\mathbf{a}_2 + \omega_{13}\mathbf{a}_3 + \omega_{23}\mathbf{a}_4 + \omega_{33}\mathbf{a}_5 = \mathbf{0} \quad , \quad (3)$$

where elements a_i are functions of the elements of the plane-to-image homography (see [15] for details). We only review the algorithm's properties that are important for the understanding of our experiments:

- all equations of type (3) (i.e. for any images of any plane) can be grouped into one linear system and solved simultaneously;
- known camera parameters can be used to reduce the number of unknowns, and thus to reduce the number of columns in the design matrix of the linear equation system;
- if some of the camera parameters are varying across different images, it is in most cases possible to simply add the columns and unknowns corresponding to the varying parameters to the global system. For example, if between ω^0 and ω^i only

¹In particular, we observed in our experiments that the choice for ε_1 was not very critical.

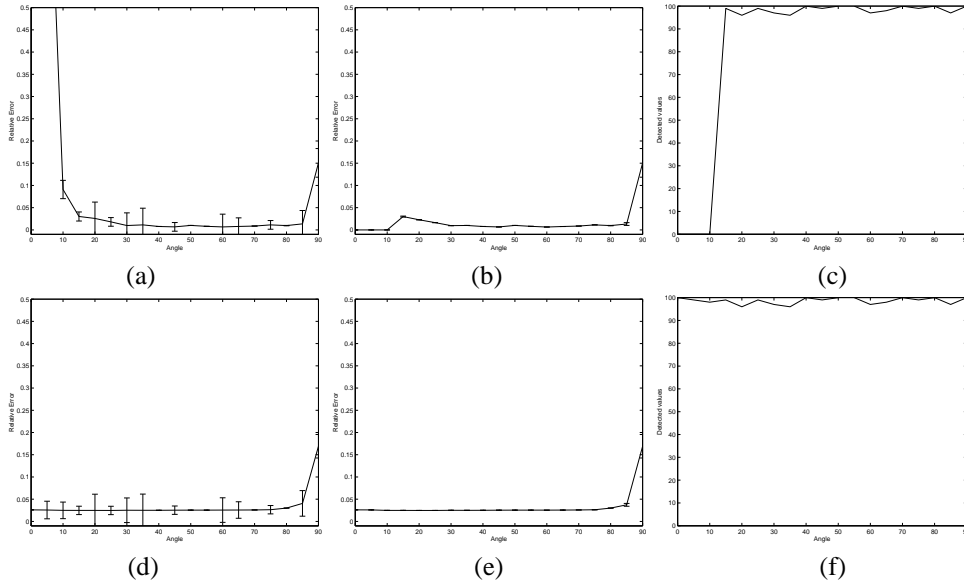


Figure 1: Estimations of $\omega_{3,3}^1$ (first row) and $\omega_{3,3}^2$ (second row) as functions of the first view’s rotation angle. 1st and 2nd column: median and variance of the relative calibration error without and with rejection of underconstrained values, respectively. The error variances are considerably smaller in the second case. 3rd column: number of accepted variable values.

the focal length is assumed to have changed, only one additional unknown ω_{33}^i is needed.

4.2 Results

With the following experiments, we want to exhibit our method’s performance in configurations that are singular or near-singular for plane-based calibration.

For our simulated experiments, we used a diagonal calibration matrix (the principal point was assumed known) with $f = 1000$ and $\tau = 1$. Calibration is performed using the projections of the 4 corner points of squares of size 40cm. The projections of the corner points are perturbed by centered Gaussian noise of 0.8 pixel variance. In the performed experience two images of the plane were taken. The constraint that they had the same focal length, was not used. Thus, two matrices ω^1 and ω^2 were estimated, with a total of four unknowns (two in common for both views, $\omega_{1,1}$ and $\omega_{2,2}$, and two for $\omega_{3,3}^1$ and $\omega_{3,3}^2$).

The second view is always taken from an angle of 45° between the optical axis and the plane normal, while the first one (i.e. corresponding to ω^1) is subject to rotations ranging from 0° (image plane parallel to 3D plane) to 90° (perpendicular to 3D plane).

According to [15], calibration from the first view alone is not possible in critical configurations (note that the aspect ratio is unknown) that occur for rotation angles 0° and 90° . However, as the second view is all the time in a non-singular position, the aspect ratio is well constrained using the associated two equations. Hence, since the views share the aspect ratio, calibration using all four associated equations is only degenerate for a rotation angle of 0° for the first view. As suggested in [15], we scaled the matrix’ columns to improve the conditioning. We fixed the threshold ε (cf. section 3) to a rather large value ($\varepsilon = 10^{-2}$), in order to disqualify potentially unstable calibration results in near-degenerate configurations. We fixed the value $\omega_{1,1}$ to obtain a non-homogeneous system.

The results are presented on Fig.1. In the first row, the calibration results of $\omega_{3,3}^1$ (first row) and $\omega_{3,3}^2$ (second row) as functions of the first view's rotation angle are shown. The charts in the two first columns represent the median and variance of relative errors from 100 random experiments using the algorithm without and with the degeneracy verification, respectively. The third row represents the number of accepted variable values for $\omega_{3,3}^1$ (first row) and $\omega_{3,3}^2$ (second row). As expected, in most cases the value of $\omega_{3,3}^2$ is estimated, while the value of $\omega_{3,3}^1$ cannot be estimated for rotations of the first view close to 0° . Some values are rejected for both cameras even in theoretically good positions. This happens in cases where the noise applied to the matrices causes degeneracies of the whole system. We can observe however, that the values of $\omega_{3,3}^2$ are not rejected more often when the camera corresponding to $\omega_{3,3}^1$ is in a degenerate position, which confirms the validity of our method.

5 Application to Euclidean Reconstruction Based on Geometrical Constraints

5.1 Outline of the Algorithm

We propose an approach for interactive scene modeling. The scene is modeled using points, lines and planes. Each of these entities can be represented by a homogeneous coordinate vector \mathbf{X} [8, 11, 7]. Contrary to existing approaches, our approach allows geometric constraints which influence several objects at once.

Three general types of constraints between objects are considered:

- **Projections.** Every known projection of a point or a line give linear constraints on the 3D coordinates of the corresponding object.
- **Bilinear constraints.** Using basic results of the Grassman-Cayley algebra, incidence, parallelism and perpendicularity between two objects \mathbf{X} , \mathbf{Y} can be expressed as bilinear forms $F(\mathbf{X}, \mathbf{Y}) = 0$ [8, 11, 7]. Thus, knowing coordinates of one of the objects induces linear constraints on the other one.
- **Linear constraints.** Relations like points lying on a parallelogram [16] or symmetry are useful in practice and are linear in terms of all the involved objects.

Usually, the above constraints do not allow to reconstruct all objects in one go. Let us consider the following example: a point \mathbf{X}_0 is defined as the intersection of two lines \mathbf{L}_0 and \mathbf{L}_1 , each of them being defined by two other points. Without further constraints, \mathbf{X}_0 can only be computed once the lines have been estimated. Similarly, a perpendicularity constraint between two lines will only be useful (in our linear estimation framework) once one of the lines has been reconstructed (see e.g. §5.3). That is why it is necessary to proceed by iterations, each iteration reconstructing as many primitives as possible from previously reconstructed ones and geometric constraints. In the following, the constraints influencing the system at the given iteration will be called *active*. There are two reasons why in such defined system the extraction of the well defined variables 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, so only the projection and symmetry/parallelogram constraints are active.

Also, even when the reconstruction process is advanced, 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 at once 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.

We propose the following reconstruction algorithm:

Algorithm 1 Iterative Reconstruction Algorithm

```

1: while !stop_condition do
2:    $N := \sum_{i=1}^n \text{nb\_coordinates}(\text{objects}[i])$ 
3:   initialize an empty linear equation system  $\mathbf{A}_{0 \times N} \mathbf{X}_{N \times 1} = \mathbf{B}_{0 \times 1}$ 
4:   compute the indexing function (bijection)  $F : \text{idx} \rightarrow (i, j); \text{idx} \in [1 \dots N]$ ,
     where  $\text{idx}$  is the index in  $X_{N \times 1}$  of the  $j$ -th coordinate of the  $i$ -th object.
5:   for all constraint  $c^k$ : do
6:     compute
        $(\mathbf{A}_{m_k \times N}^k, \mathbf{B}_{m_k \times 1}^k) := \text{equations}(c^k.\text{type}, c^k.\text{objects})$ 
7:     add equations to the system:  $\mathbf{A} := \begin{bmatrix} \mathbf{A} \\ \mathbf{A}^k \end{bmatrix} \mathbf{B}^k := \begin{bmatrix} \mathbf{B} \\ \mathbf{B}^k \end{bmatrix}$ 
8:   end for
9:   solve  $\mathbf{A}\mathbf{X} = \mathbf{B}$ 
10:  for  $\text{idx} = 1 \dots N$ : do
11:    if variable_computed( $\text{idx}$ ) then
12:      set  $(i, j) := F(\text{idx})$ 
13:      set  $\text{objects}[i].\text{coords}[j] := \mathbf{X}(\text{idx})$ 
14:    end if
15:  end for
16: end while

```

5.2 Example

The principle of our method is depicted in Fig. 2. The elements in this figure are parts of a much more complex model shown in section 5.3. The nullspace analysis and the associated variable splitting performed at each iteration make all the available information effective. The configuration shown in Fig. 2 consists of four points **P0-P3** related by a parallelogram constraint. The points **P0** and **P2** are visible in several images, while the points **P1** and **P3** are occluded by a fence. Points **P0-P3** are incident with 3 planes and 6 lines (See Fig. 2-(b) – 2-(d)), which are connected with other elements of the model by incidence and parallelism constraints. The columns 2-4 in Fig. 2 show the reconstruction achieved at the 1st, 2nd and 4th iteration. The corresponding numerical results are displayed in Tab. 1. At each reconstruction step (See Sec. 5.1) all the equations resulting from projection and geometrical constraints are added to the common equation system. In this particular case, the system contains 12 variables: the coordinates (x, y, z)

of the points **P0-P3**.

1st iteration The matrix $A_{15 \times 12}$ contains only the equations corresponding to the projection and the parallelogram constraints. The resulting matrix structure is shown in Fig. 2-(b). A straightforward analysis of this matrix shows that the variables associated to the points **P1** and **P3** are underconstrained (only 3 equations are defined for 6 variables). The nullspace analysis confirms this result. The singular values of the constraint matrix are detailed in Fig. 1. The choice of the 3 singular values which should be zeroed is obvious. The vectors spanning the nullspace correspond to the zeroed singular values. The rows of the nullspace basis which should be zeroed are also easily detected. The process result confirms the intuition that only the values corresponding to coordinates of points **P0** and **P2** are well-constrained.

2nd iteration The equations yielded by the planes **q1**, **q2** and the lines **l0**, **l1**, **l2** reconstructed in the first iteration are added to the new system matrix $A_{37 \times 12}$. The structure of this matrix (Fig. 2-(c)) suggests that there are enough constraints to solve the whole system. However, the nullspace analysis (see 4th column of Tab. 1) shows that the system is still degenerate. Indeed, the parallelogram constraint is partially redundant with the constraints coming from the incidences of points **P0-P3** with the plane **q1**.

3rd iteration The plane **q0** and the line **l3** are reconstructed. The resulting constraints are sufficient to reconstruct the whole configuration.

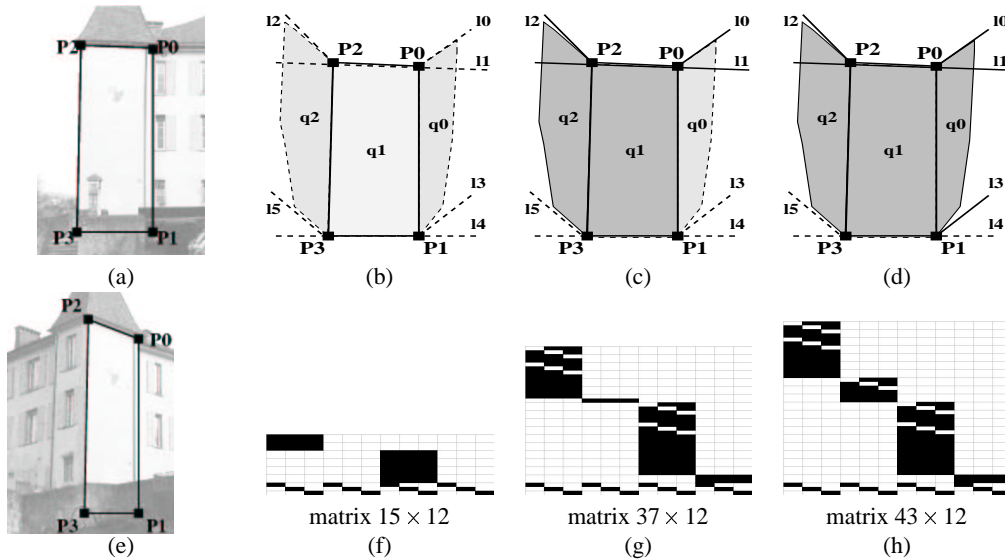


Figure 2: Illustration of the reconstruction process. 1st column: fragments of the original images with marked interest points. P1, P3 are occluded in all images. 2nd, 3rd, 4th column: the constraints influencing the interest points. 1st row: the relations between the considered objects. Lines and planes which can influence the interest points P1 – P3 (their coordinates are reconstructed at this step) are marked with black continuous line (lines) or in dark gray (planes); 2nd row: the corresponding constraint matrices. Black fields correspond to non-zero matrix elements.

5.3 Results

Reconstruction from two images One of the main applications of our approach is to build models of buildings from minimal sets of images. In the following experiment, we took two photos of a building which do not overlap. The cameras' positions and orientations were computed approximatively. A number of constraints were imposed: parallelogram constraints as well as parallelism and perpendicularity constraints between lines oriented in three main scene directions. Fig. 3 shows the original images, the reconstructed textured model and a few screen-shots of the model at different iterations of the process. At first, only the points seen in both images and related by a parallelogram constraint can be computed. The fact that cameras' positions are not perfectly aligned results in inaccurate angles in the model. At the second and third iterations, the points related to already reconstructed points by coplanarity constraints appear in the model. After that, parallelism and perpendicularity constraints start to influence the reconstructed lines and planes. At the fourth iteration, the model gets its final overall shape, and at iteration ten, the model is qualitatively correct and the system stable.

Reconstruction with occluded corners The second experiment is based on 7 images of a castle, one of which is shown in Fig. 4-(a). The cameras used were calibrated using mixed approaches [15, 16]. This reconstruction raises several difficulties:

- the images overlap only slightly, decreasing the quality of the camera calibration;
- 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;
- 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. 4-(b)). Thus geometric constraints are rather used to reconstruct castle elements which are occluded in images (e.g. see Sec. 5.2).

iteration 0				iteration 1	
W	$\Phi(\mathbf{A})$			W	$\Phi(\mathbf{A})$
3.9	<i>-9.4e-17</i>	<i>1.9e-17</i>	<i>3.6e-18</i>	1.6e+02	<i>3.2e-17</i>
3.4	<i>1.5e-16</i>	<i>1.7e-17</i>	<i>-6.3e-17</i>	1.3e+02	<i>3.6e-17</i>
3.2	<i>-7.1e-17</i>	<i>2.5e-18</i>	<i>4.3e-17</i>	77	<i>2.8e-17</i>
2.7	-0.63	0.14	-0.29	77	0.71
2.3	-0.23	0.25	0.62	3.8	-0.046
1.5	-0.23	-0.65	0.18	3.2	-0.0041
1.3	<i>9.2e-17</i>	<i>-1.7e-17</i>	<i>3.8e-17</i>	1.7	<i>-3.2e-17</i>
1.2	<i>-8.3e-17</i>	<i>-2.7e-17</i>	<i>-1.7e-17</i>	1.6	<i>-2.1e-17</i>
0.77	<i>-4.1e-17</i>	<i>-3.0e-18</i>	<i>-3.3e-18</i>	1.4	<i>-7.3e-17</i>
2.1e-16	-0.63	0.14	-0.29	1.0	0.71
8.3e-17	-0.23	0.25	0.62	0.61	-0.046
3.2e-17	-0.23	-0.65	0.18	1.1e-15	-0.0041

Table 1: Numerical results obtained for example in Fig. 2. For each iteration the 1st column contains the singular values of the constraint matrix \mathbf{A} . The horizontal line separates the values which were zeroed. Following columns contain vectors forming the matrix nullspace, with zeroed values in italic. Variables corresponding to rows containing only zero values were classified as reconstructed. These values correspond to coordinates of points P0 and P2 in Fig. 2.

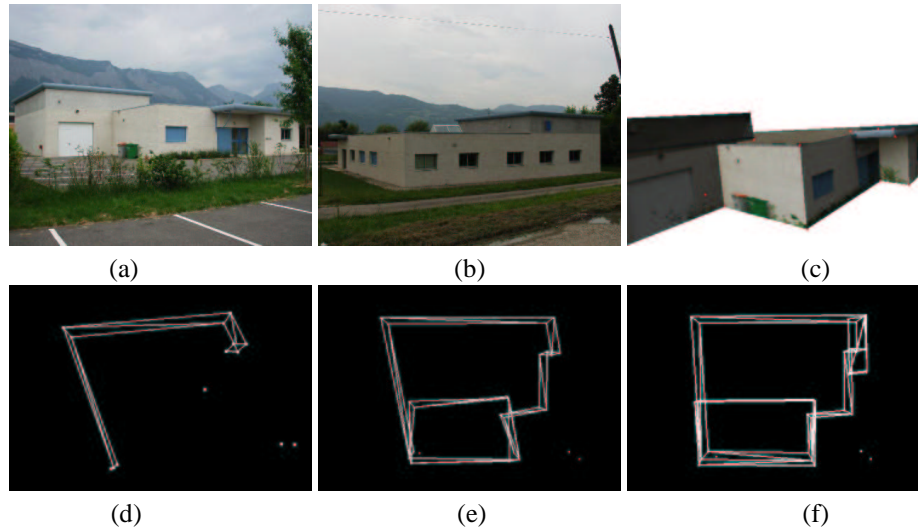


Figure 3: Reconstruction from two images; (a),(b) the original photos; (c) the corresponding textured model; (d)the points reconstructed from projection and parallelogram constraints only ; (e) the objects defined by coplanarity constraints are added and parallelism and perpendicularity affect previously reconstructed objects; (f) iteration 10: the model satisfies all the required constraints.

Fig. 4-(b) displays the 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. 4-(c).

The second row in Fig. 4 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.

6 Summary and Conclusions

We have presented a practical approach for the detection of well-constrained variables in linear equation systems. This approach is based on the SVD of the equation system's matrix and can straightforwardly be applied. Its application domain covers in particular all computer vision algorithms based on linear algebra. The approach efficiency was illustrated with two examples. In the first example of linear plane-based calibration, the approach significantly reduces the number of badly estimated parameters, while keeping well defined variables in the result. We have also proposed a flexible constraint-based 3D reconstruction algorithm. This algorithm exploits nullspace analysis in order to prevent badly estimated object coordinates from affecting subsequent iterations. We are currently investigating uncertainty analysis in order to automatically estimate the two required thresholds in the method.

References

- [1] L. de Agapito, R. Hartley and E. Hayman. *Linear selfcalibration of a rotating and zooming camera*, CVPR, pp. 15-21, 1999.

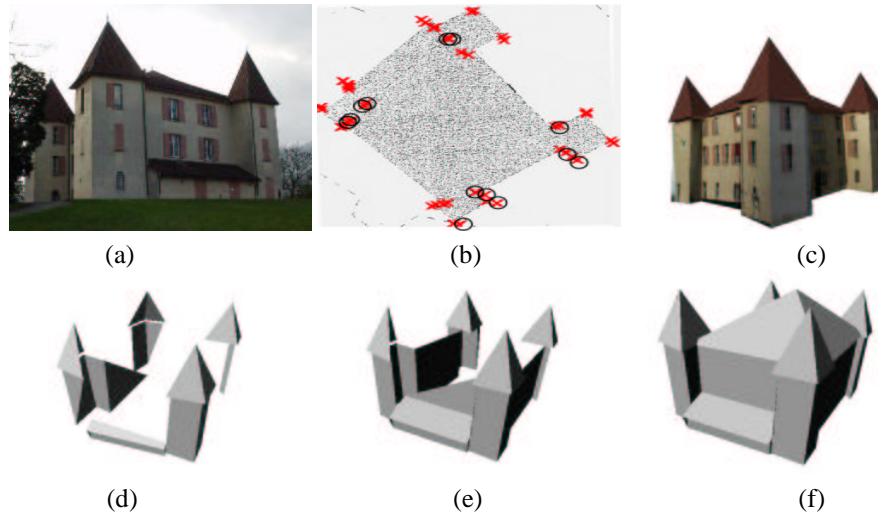


Figure 4: (a) one of the seven images used for the reconstruction; (b) the castle plane; (c) the textured 3D model; (d)-(f) screen-shots of the model at different steps of the reconstruction process.

- [2] A. Bjorck. *Numerical Methods For Least Squares Problems*, SIAM, 1990.
- [3] R. Cipolla and E. Boyer. *3D Model Acquisition from Uncalibrated Images*, IAPR Workshop on Machine Vision Applications, pp. 559-568, 1998.
- [4] P.E. Gill, W. Murray and M.H. Wright. *Practical Optimization*, Academic Press, 1981.
- [5] G. Golub and C. van Loan. *Matrix computations*, The Johns Hopkins University Press Ltd., 1996.
- [6] E. Grossmann and J.S. Victor. *Single and Multi-View Reconstruction of Structured Scenes*, ACCV, 2002.
- [7] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.
- [8] S. Heuel. *Points, Lines and Planes and their Optimal Estimation*, DAGM, 2001.
- [9] F. Kahl. *Geometry and Critical Configurations of Multiple Views*, PhD Thesis, Lund Institute of Technology, Sweden, 2001.
- [10] Y. Ma, J. Kosecka and K. Huang. *Rank Deficiency Condition of the Multiple View Matrix for Mixed Point and Line Features*, ACCV, 2002.
- [11] P. Poulin, M. Ouimet and M.-C. Frasson. *Interactively Modeling with Photogrammetry*, Eurographics Workshop on Rendering, pp. 93-104, 1998.
- [12] H. Press, S. Teukolsky, W. Vetterling and B. Flannery. *Numerical Recipes in C*, Cambridge University Press, 1992.
- [13] J. Oliensis. *Linear Stratified Self-Calibration and Euclidean Reconstruction*, Technical Report, NECI, 2002.
- [14] P. Sturm. *Vision 3D non calibrée – contributions à la reconstruction projective et étude des mouvements critiques pour l'auto-calibrage*, PhD Thesis, INPG, France, 1997.
- [15] P. Sturm and S. Maybank. *On Plane-Based Camera Calibration: A General Algorithm, Singularities, Applications*, CVPR, pp. 432-437, 1999.
- [16] M. Wilczkowiak, E. Boyer and P. Sturm. *Camera Calibration and 3D Reconstruction from Single Images Using Parallelepipeds*, ICCV, pp. 142-148, 2001.
- [17] Z. Zhang. *Flexible Camera Calibration By Viewing a Plane From Unknown Orientations*, ICCV, pp. 666-673, 1999.