Czech Pattern Recognition Workshop 2000, Tomáš Svoboda (Ed.)
Peršlák, Czech Republic, February 2–4, 2000
Czech Pattern Recognition Society

Practice of 3D Reconstruction from Multiple Uncalibrated Unorganized Images

Tomáš Werner, Tomáš Pajdla, and Martin Urban

Czech Technical University
Center for Machine Perception
Karlovo nám. 13, CZ 121 35
{werner,pajdla,urbanm}@cmp.felk.cvut.cz

Abstract *The paper presents a system for 3D reconstruction from multiple uncalibrated unorganized images. A number of alternative reconstruction techniques can be combined together by the means of a simple language. A complete reconstruction procedure is defined by a program in this language. Such an approach allows to choose a suitable reconstruction method for the situation at hand and thus obtain useful results even if a one-step reconstruction would not be possible due to occlusion and errors in image data.*

1 Introduction

This paper[1] presents a system for 3D reconstruction from multiple uncalibrated and unorganized images. By unorganized, we mean that neighborhood relations between views are not known as e.g. in a linear image sequence, but the input is a set of images taken from arbitrary locations about which nothing is known.

The system for has the following features:

1. Graphic User Interface allows manual marking of (i) correspondences and (ii) planar polygons from which texture is taken [1]. Automatic search for correspondences is not considered.
2. The reconstruction procedure is defined as a program in a simple language. The program describes how the basic building block algorithms are combined together.
3. The reconstruction computation consists of a number of basic building blocks which include algorithms for estimation of multifocal tensors from image correspondences, estimation of projection matrices from the multifocal tensors, computation of scene points by triangulation, gluing another view or reconstruction to an existing reconstruction, factorization, bundle adjustment, stratification of projective reconstruction to quasi-affine, affine, or Euclidean reconstruction, etc.
4. A VRML model of the 3D reconstruction with correctly mapped texture can be build.

[1] This research was supported by the Czech Ministry of Education under the grant VS96049, by grant OCAMS No. 4/11/AIP CR, and by the Research Programme J04/98:212300013 Decision and control for industry.

2 Used Notation

Projective space of dimension d is denoted by $\mathcal{P}_d$. Let a scene consist of N 3D scene points. They are indexed by index $n = 1, \ldots, N$ in subscript. The n-th scene point lies in $\mathcal{P}_3$ and is represented by 4-vector of homogeneous coordinates $\mathbf{X}_n$.

Let the scene be observed by K cameras. The cameras/views are indexed by index $k = 1, \ldots, K$ in superscript. The k-th camera is represented by its 3×4 camera projection matrix $\mathbf{P}^k$.

Let us measure image points, which are represented by 3-vectors $\mathbf{u}_n^k$ of homogeneous coordinates, and that are projections of n-th scene point in k-th image. The image points $\mathbf{u}_n^k$ may have not been measured for each pair $[n, k]$ due to e.g. occlusions. The measurement presence variable π_n^k is non-zero iff image point $\mathbf{u}_n^k$ has been measured. However, if the extension to missing data case is straightforward, the formulae will be mostly presented as if $\mathbf{u}_n^k$ were available for all pairs $[n, k]$.

Projection equation describes the process of projecting a scene point by a camera:

$$\rho_n^k \mathbf{u}_n^k = \mathbf{P}^k \mathbf{X}_n \tag{1}$$

where $\rho_n^k \neq 0$ is a scale factor.

3 Language Describing Reconstruction Procedure

To compute a projective reconstruction from all K views, the building block algorithms are combined together by the means of a language. The exact reconstruction procedure is specified by the user as a sentence in this language. Here are some generic examples of the reconstruction commands:

- '$[1;3]$' means projective reconstruction from views 1,3 via fundamental matrix,
- '$[1;3,5]$' means projective reconstruction from views 1,3,5 via trifocal tensor,
- '$[1;2,3]+4$' means gluing view 4 to a reconstruction,
- '$[1;2,3]+[2;4]$' means fusing two projective reconstructions, and
- '$\mathrm{euclids}([1;2,3])$' means Euclidean stratification from diagonal $\mathbf{K}$.

Here is an example of a more complex command:

$$\text{bundle}(\text{euclids}(\text{fact}([1;3,4]+5)) + ([2;6,7,8]+9))$$

In the next two sections, the used algorithms will be described in more detail.

4 Projective Reconstruction

The goal of projective reconstruction is to compute $\mathbf{P}^k$ and $\mathbf{X}_n$ from the measured image points $\mathbf{u}_n^k$ so that (1) holds.

It is an important result (e.g., [4]) that $\mathbf{P}^k$ and $\mathbf{X}_n$ can be determined only up to a 3D-to-3D projectivity. It is because we can rewrite the right-hand side of (1) to

$$\mathbf{P}^k\mathbf{X}_n = (\mathbf{P}^k\mathbf{H}^{-1})(\mathbf{H}\mathbf{X}_n) = \mathbf{P}'^k\mathbf{X}'_n \qquad (2)$$

where $\mathbf{H}$ is a 4×4 matrix with rank four, describing the 3D-to-3D projective transformation. The factors in parentheses on the right-hand side can be considered as a new and valid projective reconstruction. Matrix $\mathbf{H}$ cannot be determined without further knowledge about the cameras or the scene. The estimation of $\mathbf{P}^k$ and $\mathbf{X}_n$ up to an unknown $\mathbf{H}$ from $\mathbf{u}_n^k$ is called projective reconstruction.

A good algorithm for automatically computing projective reconstruction should (i) cope with any configuration of scene/cameras that theoretically enables computing a projective reconstruction, (ii) recognize configurations that do not enable it, (iii) treat all points and cameras equally, (iv) tollerate sufficiently large amount of noise in image measurements, and (v) cope with missing image measurements (zero π_n^k). As the evidence why designing such an algorithm is difficult let us notice that the algorithm has to cope equally well with three different scene/cameras configurations shown in Figure 1.

Thus far, no algorithm with the above properties is known. However, practical and efficient algorithms are known for computing (i) $\mathbf{P}^k$ from $\mathbf{u}_n^k$ for 2, 3, and 4 views, (ii) $\mathbf{X}$ from $\mathbf{P}^k$ and $\mathbf{u}^k$, and (iii) $\mathbf{P}$ from $\mathbf{X}_n$ and $\mathbf{u}_n$. We describe them in the rest of this section. We propose semi-automatic way for computing the projective reconstruction: these basic algorithms works automatically but the way in which they are combined so that the complete projective reconstruction is achieved is specified by the user.

4.1 Reconstruction from 2 views

Projective reconstruction from views 1 and 2 is done via decomposing fundamental matrix $\mathbf{F}^{12}$, estimated by an 8-point algorithm [2], to $\mathbf{P}^1$ and $\mathbf{P}^2$. In 8-point algorithm, as well as in solving other linear equation systems, the suitable normalization of $\mathbf{u}_n^k$ is done [3].

A simple and unusual algorithm is used for decomposing $\mathbf{F}^{12}$. For any two points $\mathbf{u}_n^1, \mathbf{u}_n^2$ we have $\mathbf{u}_n^{1\top}\mathbf{F}^{12}\mathbf{u}_n^2 = 0$, hence for any scene point $\mathbf{X}_n$ it is $\mathbf{X}_n^\top\mathbf{P}^{1\top}\mathbf{F}^{12}\mathbf{P}^2\mathbf{X}_n = 0$. $\mathbf{P}^1, \mathbf{P}^2$ can always be transformed so that $\mathbf{P}^1 = [\mathbf{I}_3|\mathbf{0}_3]$. Then the constraint for $\mathbf{P}^2$ is simply that

$$\begin{bmatrix} \mathbf{F}^{12}\mathbf{P}^2 \\ \mathbf{0}_4^\top \end{bmatrix}$$

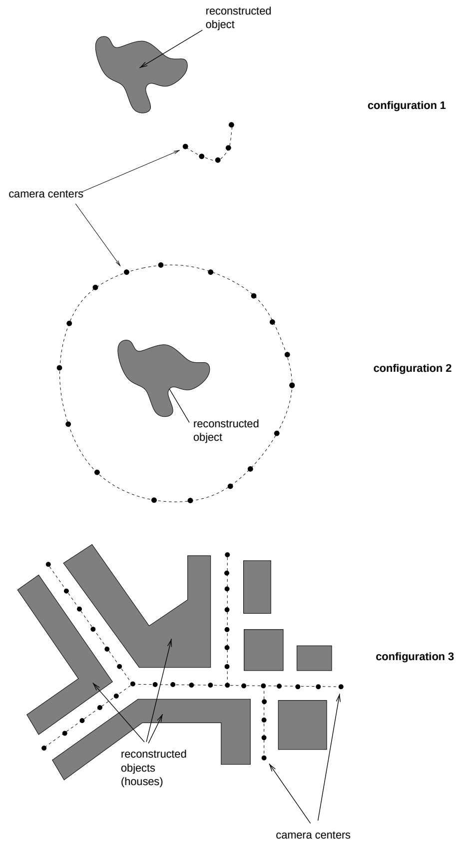

Figure 1: Qualitatively different configurations of scene/cameras for 3D reconstruction. In configuration 1, approximately the same part of the scene is visible from all the viewpoints. In configuration 2, the viewpoints are all around the reconstructed object. Different parts are visible from different viewpoints, however, the whole geometry can be estimated accurately because different scene parts are connected thanks to the fact that the viewing curve is closed. This would be even more obvious in a 2-D extension of configuration 2. Configuration 3 demonstrates walking on the streets, where the distant parts of the scene are connected only very loosely. Accumulation of errors in geometry is inevitable.

be a skew-symmetric matrix. This yields a linear underdetermined equation system for $\mathbf{P}^2$. From the solution space, some $\mathbf{P}^2$ with a full rank is chosen.

4.2 Reconstruction from 3 views

It is done via estimation and decomposition of trifocal tensor [4, 5].

4.3 Reconstruction from more than 3 views with one common view

If the views $1,\ldots,K, K > 3$, can be divided in $K-1$ triplets $(1,2,3), (1,3,4), \ldots, (1,K,2)$ such that trifocal tensor can be estimated from each triplet, projective reconstruction from these views is computed by the following method [9]: (i) Estimate trifocal tensor for each triplet independently, (ii) recover the epipoles from adjoining tensors, (iii) estimate $\mathbf{P}^1,\ldots,\mathbf{P}^K$ from $\mathbf{u}_n^1,\ldots,\mathbf{u}_n^K$ simultaneously using the recovered epipoles.

4.4 Reconstructing a scene point from image points and camera matrices

Let us have image points $\mathbf{u}^k$ in $k = 1, 2, \ldots, K$ images, and projection matrices $\mathbf{P}^k$. We want to estimate scene point $\mathbf{X}$ satisfying $\rho^k \mathbf{u}^k = \mathbf{P}^k \mathbf{X}$. By eliminating ρ^k, we can write

$$[\mathbf{u}^k]_\times \mathbf{P}^k \mathbf{X} = \mathbf{0}_3$$

where $[\mathbf{u}]_\times$ is a 3×3 skew-symmetric matrix such that for each $\mathbf{u}, \mathbf{v}$ it is $[u]_\times \mathbf{v} = \mathbf{u} \times \mathbf{v}$ where $\times$ denotes vector product. This is an overdetermined linear system for $\mathbf{X}$.

4.5 Estimating camera matrix from corresponding scene and image points

Let us have corresponding pairs $[\mathbf{X}_n, \mathbf{u}_n]$ for $n = 1, \ldots, N$. We want to estimate camera matrix $\mathbf{P}$ such that $\rho_n \mathbf{u}_n = \mathbf{P}\mathbf{X}_n$ holds. After some rearranging it is

$$[\mathbf{u}_n]_\times \begin{bmatrix} \mathbf{X}_n^\top & \mathbf{0}_4^\top & \mathbf{0}_4^\top \\ \mathbf{0}_4^\top & \mathbf{X}_n^\top & \mathbf{0}_4^\top \\ \mathbf{0}_4^\top & \mathbf{0}_4^\top & \mathbf{X}_n^\top \end{bmatrix} \begin{bmatrix} \mathbf{p}^\top \\ \mathbf{q}^\top \\ \mathbf{r}^\top \end{bmatrix} = \mathbf{0}_3 \qquad (3)$$

where $\mathbf{p}^\top, \mathbf{q}^\top, \mathbf{r}^\top$ are rows of $\mathbf{P}$. This is an over-determined linear system for $\mathbf{P}$.

4.6 Gluing separate views to an existing reconstruction

We use a simple algorithm to compute a reconstruction from $k+1$ views, having already a reconstruction from k views. E.g., let us have image correspondences $\mathbf{u}_n^1, \ldots, \mathbf{u}_n^K$ from which $\mathbf{P}^1, \mathbf{P}^2, \mathbf{P}^3$ have been computed via trifocal tensor. Then $\mathbf{P}^4$ can be estimated in two steps: (i) From $\mathbf{P}^k$ and $\mathbf{u}_n^k$ where $k = 1, 2, 3$, reconstruct those scene points $\mathbf{X}_n$ that are visible in at least two views of the views 1, 2, 3. (ii) Estimate $\mathbf{P}^4$ from $\mathbf{u}_n^4$.

Having $\mathbf{P}^4$, we again reconstruct those $\mathbf{X}_n$ that are visible in at least two views of the views 1, 2, 3, 4. Iterating the process for $k = 5, \ldots, K$, the reconstruction from all K views is computed.

4.7 Fusing other reconstructions with existing reconstruction

Another way how to compute a reconstruction from more than 4 views is done in two steps: (i) compute reconstructions from different sets of views, (ii) transform them to a common projective coordinate system. E.g., assume $\mathbf{P}^1, \mathbf{P}^2, \mathbf{P}^3$ computed from views $1, 2, 3$ via trifocal tensor, and $\mathbf{P}'^4, \mathbf{P}'^5$ computed from views $4, 5$ via fundamental matrix. These matrices are related via a projectivity $\mathbf{H}$ (4×4 matrix) as follows:

$$\mathbf{u}_n^k \simeq \mathbf{P}^k \mathbf{X}_n, \quad k = 1, 2, 3 \qquad (4)$$

$$\mathbf{u}_n^k \simeq \mathbf{P}'^k \mathbf{X}'_n, \quad k = 4, 5 \qquad (5)$$

where $\mathbf{X}'_n \simeq \mathbf{H}\mathbf{X}_n, \mathbf{P}'^k \simeq \mathbf{P}^k \mathbf{H}^{-1}$. Transforming $\mathbf{P}^k$ and $\mathbf{P}'^k$ to a common system is done in four steps: (i) Reconstruct $\mathbf{X}_n$ from $\mathbf{u}_n^k$ and $\mathbf{P}^k$ for $k = 1, 2, 3$, (ii) reconstruct $\mathbf{X}'_n$ from $\mathbf{u}_n^k$ and $\mathbf{P}'^k$ for $k = 4, 5$, (iii) estimate $\mathbf{H}$ from the system $\mathbf{X}'_n \simeq \mathbf{H}\mathbf{X}_n$, and (iv) transform $\mathbf{P}'^k$ to $\mathbf{P}^k$ as $\mathbf{P}'^k \simeq \mathbf{P}^k \mathbf{H}^{-1}$.

In this example, the two sets of views $\{1, 2, 3\}$ and $\{4, 5\}$ have no view in common. What if the two sets have views in common? E.g., in case when there are two reconstructions from views $1, 2, 3$ and $3, 4$, we have one more condition $\mathbf{P}'^3 \simeq \mathbf{P}^3 \mathbf{H}^{-1}$, which is added to the linear system $\mathbf{X}'_n \simeq \mathbf{H}\mathbf{X}_n$.

4.8 Enhancing an existing reconstruction by factorization

(1) can be written for all n and k in matrix form as

$$[\rho_n^k \mathbf{u}_n^k] = [\mathbf{P}^k][\mathbf{X}_n] \qquad (6)$$

where $[\rho_n^k \mathbf{u}_n^k]$ is a $3K \times N$ joint image matrix, $[\mathbf{P}^k]$ is a $3K \times 4$ joint camera matrix [8], and $[\mathbf{X}_n]$ is a $4 \times N$ matrix. If $\mathbf{u}_n^k$ are perturbed by noise, the equality does not hold accurately. Then, the matrix $[\mathbf{P}^k][\mathbf{X}_n]$ has rank four, yet the matrix $[\rho_n^k \mathbf{u}_n^k]$ generally does not. The point of the factorization algorithm is to force the matrix $[\rho_n^k \mathbf{u}_n^k]$ to have rank four by, e.g., SVD.

We use factorization for enhancing an existing reconstruction rather than for computing it directly from $\mathbf{u}_n^k$. The existing reconstruction is used to find the initial estimate of ρ_n^k. The missing $\mathbf{u}_n^k$ are either re-projected as $\mathbf{P}^k \mathbf{X}_n$ or the factorization is done only for a subset of points visible in all images.

4.9 Bundle adjustment

Optimal estimate of $\mathbf{P}^k$ and $\mathbf{X}_n$ from $\mathbf{u}_n^k$ is the maximum likelihood estimate. Let us denote by $\xi(\mathbf{u})$ the 2-vector of coordinates that was measured in actually captured image – i.e., for usual cameras it is $\xi(\mathbf{u}) = [u, v]^\top / w$ where $\mathbf{u} = [u, v, w]^\top$. If the noise enters the measurement in images and if $\xi(\mathbf{u}_n^k)$ can be considered as stochastic variables with isotropic variances, the maximum likelihood estimate is the solution of the following optimization problem:

$$\arg \min_{\mathbf{P}^k, \mathbf{X}_n} \sum_{\pi_n^k \neq 0} ||\xi(\mathbf{u}_n^k) - \xi(\mathbf{P}^k \mathbf{X}_n)||_2 \qquad (7)$$

where $||.||_2$ denotes Euclidean norm. The solution can be found by non-linear search (we use Levenberg-Marquardt), taking $\mathbf{P}^k, \mathbf{X}_n$ obtained by some other method as initial estimate. In actual implementation, we minimize only over $\mathbf{P}^k$. Each time when the residual function is to be evaluated, $\mathbf{X}_n$ is computed from $\mathbf{P}^k$ and $\mathbf{u}_n^k$.

5 Stratifying Projective Reconstruction

We assume that image measurements $\mathbf{u}_n^k$ have been obtained by projecting some real existing scene points $\bar{\mathbf{X}}_n$ by some real existing cameras with matrices $\bar{\mathbf{P}}^k$. When applying a 3D reconstruction algorithm to the image measurements, we would like to compute the original scene structure $\bar{\mathbf{X}}_n$ and cameras $\bar{\mathbf{P}}^k$. However, without further knowledge we can reconstruct the scene and the cameras only up to a projectivity. That is, we obtain scene points $\mathbf{X}_n$ and camera matrices $\mathbf{P}^k$ which differ from $\bar{\mathbf{X}}_n$ and $\bar{\mathbf{P}}^k$ by an unknown projective transformation:

$$\sigma^k \mathbf{P}^k = \bar{\mathbf{P}}^k \mathbf{H}^{-1} \qquad (8)$$

$$\lambda_n \mathbf{X}_n = \mathbf{H}\bar{\mathbf{X}}_n \tag{9}$$

Once having $\mathbf{X}_n, \mathbf{P}^k$, we can use further knowledge about scene structure or cameras to stratify the projective reconstruction to something closer to reality – quasi-affine, affine, or Euclidean reconstruction. In other words, stratifying a projective reconstruction means using an additional knowledge to restrict the space of possible $\mathbf{H}$.

This knowledge can come in various forms: 3D coordinates of some scene points, perpendicularity or parallelity of some scene lines or planes, knowing all or some intrinsic calibration parameters of the cameras, etc. We will describe some stratification algorithms in this section.

5.1 Decomposing projective transformation

A general 3D-to-3D projective transformation $\mathbf{H}$ can be uniquely decomposed as follows:

$$\mathbf{H} = \mathbf{H}_P \mathbf{H}_A \mathbf{H}_E = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \\ \omega^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{K} & \mathbf{0}_3 \\ \mathbf{0}_3^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & -\mathbf{R}\mathbf{t} \\ \mathbf{0}_3^\top & 1 \end{bmatrix} \tag{10}$$

$\mathbf{H}_P$ represents an element of a 3D projective group factorized by 3D affine group. Vector $[-\omega^\top \ 1]^\top$ represents the plane that is at infinity in the original scene. $\mathbf{H}_P$ has 3 DOF. Once we know $\mathbf{H}_P$, we know the scene up to an affine transformation $\mathbf{H}_A\mathbf{H}_E$, that is, we have obtained affine scene reconstruction.

$\mathbf{H}_A$ represents an element of a 3D affine group factorized by 3D Euclidean group. Upper triangular matrix $\mathbf{K}$ describes what remain of a general 3D-to-3D affine transformation when Euclidean transformation is removed from it. Namely, $\mathbf{K}$ describes anisotropic scaling (3 diagonal entries) and a skewing of coordinate axes (3 off-diagonal entries). Algebraically, factoring affine group by Euclidean group is QR-factorization of matrix $\mathbf{KR}$. $\mathbf{H}_A$ has 6 DOF. Once we know $\mathbf{H}_P$ and $\mathbf{H}_A$, we know the scene up to a Euclidean transformation $\mathbf{H}_E$, that is, we have obtained Euclidean scene reconstruction.

$\mathbf{H}_E$ represents an element of a 3D Euclidean group. Rotation matrix $\mathbf{R}$ ($\mathbf{R}\mathbf{R}^\top = \mathbf{I}_3$) represents rotation and vector $\mathbf{t}$ translation. $\mathbf{H}_E$ has 6 DOF – 3 for $\mathbf{R}$ and 3 for $\mathbf{t}$. Once we know $\mathbf{H}_P\mathbf{H}_A\mathbf{H}_E$, we know the scene in absolute world coordinates.

Any projective reconstruction can be transformed by some projective transformation such that one camera projection matrix equals $[\mathbf{I}_3|\mathbf{0}_3]$. Assume further the original camera matrices in the form

$$\bar{\mathbf{P}}^k = \mathbf{K}^k \left[\mathbf{R}^k | -\mathbf{R}^k\mathbf{t}^k\right] \tag{11}$$

It can be easily verified that if e.g. $\mathbf{P}^1 = [\mathbf{I}_3|\mathbf{0}_3]$ and (11) holds, it is

$$\mathbf{K} = \mathbf{K}^1, \ \ \mathbf{R} = \mathbf{R}^1, \ \ \mathbf{t} = \mathbf{t}^1 \tag{12}$$

5.2 Affine stratification from known plane at infinity

Knowing at least 3 scene points that are at infinity in the original scene allows finding plane at infinity in the projective reconstruction, represented by vector $[-\omega^\top \ 1]^\top$, and thus recovering $\mathbf{H}_P$. These points can be found as intersections of planes or lines that are known to be parallel in the original scene.

5.3 Stratification from diagonal matrix of intrinsic calibration parameters

This stratification comes from the algorithm due to Pollefeys and van Gool [7]. From (8), (10) and (11) we have

$$\frac{1}{\sigma^k}\mathbf{K}^k \left[\mathbf{R}^k | -\mathbf{R}^k\mathbf{t}^k\right] = \mathbf{P}^k \begin{bmatrix} \mathbf{KR} & -\mathbf{KRt} \\ \omega^\top & 1 \end{bmatrix} \tag{13}$$

By multiplying the equation by matrix $\bar{\mathbf{\Omega}} = \mathrm{diag}([1\ 1\ 1\ 0]^\top)$ from the right and multiplying each side by the transpose of itself from the right we obtain

$$(\sigma^k)^{-2}\mathbf{K}^k\mathbf{K}^{k\top} = \mathbf{P}^k \begin{bmatrix} \mathbf{K} \\ \omega^\top \end{bmatrix} \begin{bmatrix} \mathbf{K}^\top & \omega \end{bmatrix} \mathbf{P}^{k\top} \tag{14}$$

In many practical situations we can assume that (i) all cameras have zero skew and aspect ratio equal to 1, and (ii) the principal points are approximately known for all cameras. Then, $\mathbf{K}^k$ can be transformed (transforming image points accordingly) to $\mathbf{K}^k = \mathrm{diag}([f^k\ f^k\ 0]^\top)$. Substituting this to (14) and assuming $\mathbf{P}^1 = [\mathbf{I}_3|\mathbf{0}_3]$ yields

$$(\sigma^k)^{-2}\mathbf{K}^k\mathbf{K}^{k\top} = \mathbf{P}^k \begin{bmatrix} (f^1)^2 & 0 & 0 & f^1\omega_1 \\ 0 & (f^1)^2 & 0 & f^1\omega_2 \\ 0 & 0 & 1 & \omega_3 \\ f^1\omega_1 & f^1\omega_2 & \omega_3 & \omega^\top\omega \end{bmatrix} \mathbf{P}^{k\top} \tag{15}$$

Now, the fact that $c_{12}^k = c_{13}^k = c_{23}^k = 0$ and $c_{11}^k = c_{22}^k$, where $(\sigma^k)^{-2}\mathbf{K}^k\mathbf{K}^{k\top} = [c_{ij}^k]$, yields a linear system for 5 unknowns $(f^1)^2$, $\mathbf{K}^1\omega$, and $\omega^\top\omega$.

These unknowns can be computed uniquelly from $K > 2$ views. For $K = 2$, the non-linear constraints $(f^1)^2$, $\mathbf{K}^1\omega$, and $\omega^\top\omega$ must be used to select the unique solution for f^1 and ω.

5.4 Stratifying to quasi-affine reconstruction

Unlike by $\mathbf{H}_E$ and $\mathbf{H}_A$, transforming the original reconstruction by $\mathbf{H}_P$ does not preserve the relation 'a scene point lies in a convex hull of other scene points'. However, the knowledge of $\mathbf{u}_n^k$ suffices to find a quasi-affine reconstruction which does preserve the relation. We compute projectivity $\mathbf{H}$ that transforms a general projective reconstruction $\mathbf{P}^k, \mathbf{X}_n$ to a quasi-affine reconstruction using the algorithm from [6]. It proceeds in two steps: (i) Multiply some of $\mathbf{P}^k$ and $\mathbf{X}_n$ by -1 so that

$$\mathbf{u}_n^{k\top}\mathbf{P}^k\mathbf{X}_n > 0 \tag{16}$$

for all n, k. (ii) Find $\mathbf{H}$ satisfying the system

$$\mathbf{h}_4^\top \mathbf{X}_n > 0 \tag{17}$$
$$\det(\mathbf{H})\, \mathbf{h}_4^\top C(\mathbf{P}^k) > 0 \tag{18}$$

where $\mathbf{h}_4^\top$ is the 4-th row of $\mathbf{H}$ and $C(\mathbf{P}) = [c_i(\mathbf{P})]$ is a 4-vector such that $c_i(\mathbf{P}) = (-1)^i \det(\mathbf{P}_i)$ where $\mathbf{P}_i$ is a 3×3 matrix obtained by removing the i-th column from $\mathbf{P}$. It is assumed that the true cameras $\bar{\mathbf{P}}^k$ are such that $c_4(\bar{\mathbf{P}}^k) > 0$.

If there are solutions for both positive and negative $\det(\mathbf{H})$, then the plane which is at infinity in the true scene may separate camera centers from the scene. If there is only one solution, the plane is known to be outside the convex hull of scene points and camera centers, and the orientation of the reconstructed scene to be the same as that of the true scene.

5.5 Inflating a reconstruction

It is useful to visualize a 3D projective reconstruction in such a way that the visualization is at least recognized by the user as resembling the original scene. Assuming that an oriented projective or affine reconstruction is available and that the original scene is not too elongated in some direction, the following heuristic can be used: Using principal component analysis (algebraically, SVD), transform $\mathbf{X}_n$ by a 3D-to-3D affine transformation such that singular values of the matrix $[\mathbf{x}_n - \text{mean}(\mathbf{x}_n)]$ are equal. It is $\mathbf{x}_n = [X_n, Y_n, Z_n]^\top / W_n$ where $\mathbf{X}_n = [X_n, Y_n, Z_n, W_n]^\top$.

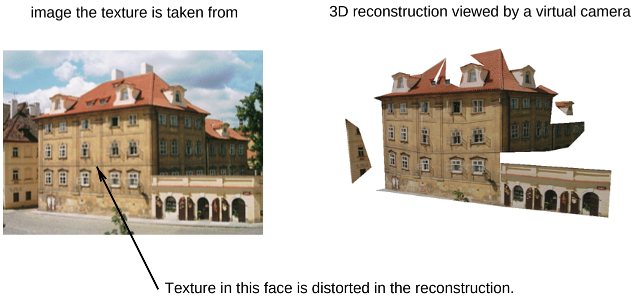

Figure 2: Texture in the reconstruction is distorted.

6 Creating a VRML Model

The reconstructed scene is visualized using a VRML viewer. From $\mathbf{X}_n, \mathbf{P}^k, \mathbf{u}_n^k$, and 3D planar polygons marked in the images, a VRML source code is generated. Visualization of reconstructed points, non-textured or textured faces, cameras, and image residuals is supported.

This is straightforward except the following problem. Transformation that warps texture from an image to a 3D planar polygon is a 2D-to-2D projectivity. In the VRML language, the texture warping is done by the viewer software. The 2D coordinates of vertices of the polygon in the image together with the 3D coordinates of the corresponding vertices of the polygon in the scene are specified in the VRML source code. However, this works correctly only if the image polygon is a front-parallel projection of the corresponding 3D polygon. If this is not satisfied, the texture is distorted as shown in Figure 2. The front face in the image on the right hand side were divided into two triangles by the VRML viewer, and the texture in each triangle was warped by a different transformation. The discontinuity along the line separating the triangles is clearly visible.

This observation leads us to requiring the VRML specification to be extended so that a true projective texture warping is supported. Currently, we removed the distortion by adaptively triangulating polygons to triangles that are small enough to keep the texture distortion under one pixel.

7 Experiments with Reconstructions from Photographs

The described algorithms, the language interpreter, and graphic user interface for marking correspondences (being at the same time a shell for the whole system, see [1]) were implemented as a MATLAB tool box.

Figure 3: Example of resulting 3D model.

We reconstructed (mostly partial) 3D models of several indoor objects and buildings from 2 to 7 views. Images were obtained by scanning photographs taken by a camera with a zoom lens. Example of the resulting model is in Figure 3. Projective reconstruction was stratified to Euclidean by the algorithm from Section 5.3.

8 Summary

We present a system for reconstruction of 3D model from uncalibrated images. The system includes GUI for marking the correspondences, a set of algorithms for computing projective reconstruction, combining different reconstructions, and stratifying projective reconstruction using further knowledge. These algorithms can be combined to a reconstruction procedure using a simple language. The system is open for new algorithms, which rapidly appear due to research progress.

In the paper, the practically useful latest-knowledge algorithms are described in a concise and unified manner. Also several novel ideas are presented.

All algorithms treat all three components of $\mathbf{u}_n^k$ equally – non-homogeneous form $[u/w, v/w, 1]^\top$ is never used. This allows using not only conventional cameras but also e.g. panoramic images in which w can have any value. In near future, we will use the system for reconstruction from both directional and panoramic images.

References

[1] J. Buriánek. Korespondence pro virtualní kameru. Master's thesis, Czech Technical University, FEL ČVUT, Karlovo náměstí 13, Praha, Czech Republic, 1998. In Czech.

[2] O. Faugeras. *Three-Dimensional Computer Vision.* MIT Press, Cambridge, Massachusetts, 1993.

[3] R. Hartley. In defence of the 8-point algorithm. In *International Conference on Computer Vision*, pages 1064–1070, 1995.

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

[5] R. Hartley. Minimizing algebraic error. In *The Sixth International Conference on Computer Vision in Bombay, India*, page Session 3.1, 1998.

[6] R. I. Hartley. Chirality. *Int. Jour. Computer Vision IJCV*, 26(1):41–61, 1998.

[7] M. Pollefeys, R. Koch, and L. VanGool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *The Sixth International Conference on Computer Vision in Bombay, India*, page Session 1.4, 1998.

[8] B. Triggs. The geometry of projective reconstruction 1: Matching constraints and the joint image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1995.

[9] M. Urban, T. Pajdla, and V. Hlaváč. Projective reconstruction from n views having one view in common. In B. Triggs, R. Szeliski, and A. Zisserman, editors, *Vision Algorithms: Theory & Practice*, LNCS, Berlin, Germany, September 2000. Springer.