

A Linear Solution to 1-Dimensional Subspace Fitting under Incomplete Data

Hanno Ackermann, Bodo Rosenhahn

Leibniz University Hannover
Appelstr. 9a, 30167 Hannover, Germany
{ackermann, rosenhahn}@tnt.uni-hannover.de

Abstract. Computing a 1-dimensional linear subspace is an important problem in many computer vision algorithms. Its importance stems from the fact that maximizing a linear homogeneous equation system can be interpreted as subspace fitting problem. It is trivial to compute the solution if all coefficients of the equation system are known, yet for the case of incomplete data, only approximation methods based on variations of gradient descent have been developed.

In this work, an algorithm is presented in which the data is embedded in projective spaces. We prove that the intersection of these projective spaces is identical to the desired subspace. Whereas other algorithms approximate this subspace iteratively, computing the intersection of projective spaces defines a *linear* problem. This solution is therefore not an approximation but *exact* in the absence of noise. We derive an upper boundary on the number of missing entries the algorithm can handle. Experiments with synthetic data confirm that the proposed algorithm successfully fits subspaces to data even if more than 90% of the data is missing. We demonstrate an example application with real image sequences.

1 Introduction

In this work, we consider the problem of estimating the linear subspace of a matrix of rank-1 if not all entries of this matrix are known. Since the maximization of a linear homogeneous equation system amounts to fitting a 1-dimensional subspace to the coefficient matrix, the mathematical community already considered the additional difficulty of unknown coefficients in the late 60s and 70s [23, 17, 10, 22, 8, 18]¹.

In computer vision, subspace fitting with missing data received considerable attention after the introduction of the *factorization* algorithm by Tomasi and Kanade [20]. In the same work, the authors also proposed an algorithm which computes left and right subspaces starting with some part of the matrix where all entries are known. This solution is iteratively propagated until all unknown entries are estimated. However, it is likely that the cameras which gave rise to

¹ Please notice that for the Wiberg-algorithm, [22] is usually cited. However, this publication is missing, at least in the issue available online.

the initial submatrix are in degenerate configuration, hence subsequent estimates would be incorrect.

Jacobs proposed to first estimate the intersections of all triplets of affine subspaces induced by points having unknown coordinates [13]. The subspace is taken to be identical with the space which contains all intersections, i.e. the intersection of intersections. In general, however, even in the absence of any noise, this intersection is empty.

Many other approaches in the computer vision community are variants of the approach of Tomasi and Kanade, for instance by integrating statistical reliability measures [7] or optimizing a different cost function [3]. Modified EM-schemes [23, 8] were also proposed, e.g. considering some statistical properties [19], or a different cost [2]. Hartley and Schaffalitzky alternately estimate left and right subspaces [11]. Buchanon and Fitzgibbon [4] proposed to improve convergence speed by combining a Gauss-Newton approach [17, 18] with the method of [11]. Recently, it was proposed to integrate problem-specific constraints into EM-schemes [16, 1].

A new type of algorithms was recently proposed: heuristics which minimize the l_0 -, the l_1 -norm, or the nuclear norm allow for the recovery of low-rank matrices with partially missing entries, or even matrices with some entries corrupted [6, 9]. Minimizing a spectral norm implies that the number of non-zero singular values is not kept fixed. Consequently, such algorithms can converge to a solution where the number of non-zero singular values is different as required by the physical model. According to our experience this frequently happens indeed. Relying on convex optimization, they require certain parameters specified a priori, and they are known to be slow if the problem size becomes large. Lastly, their gradient-descent nature makes them susceptible to local minima.

Summarizing, Gauss-Newton approaches [17, 18, 4] are susceptible to local minima hence they require good initial solutions. Conversely, EM-schemes [23, 22, 8, 19, 16, 1] or EM-like schemes [11] seem to converge more robustly, yet can require tremendously much time [4]. Approaches generalizing some estimate obtained from some part of the data to the complete data are highly susceptible to noise, larger amounts of missing data and degenerate camera configurations [20, 13].

In this work we consider the special case of subspaces of rank-1. We will prove that for this case a *linear* solution exists and is unique. Subspace estimation in the presence of missing matrix entries reduces to solving a single linear equation system, hence there is no necessity to iterate. No parameters need be chosen a priori, and the algorithm is immune to local minima. The proposed approach is significantly faster than iterative approaches even for large problems. Experimental evaluation will show that our algorithm estimates accurate subspace even if less than 10% of the data is known. Its usefulness is further demonstrated with several challenging real image sequences.

In Section 2 we will prove existence and uniqueness of a linear solution for the estimation of a rank-1 subspace if not all matrix entries are known. Section 3 continues with an experimental evaluation of accuracy using simulated data. A

model application with real-image data is presented in Section 4. The paper ends with conclusions in Section 5.

2 A Linear Solution

Capitalized letters M are used for scalar constants, bold lower-case letters \mathbf{x} for vectors, bold capital letters \mathbf{X} for matrices, and calligraphic capital letters \mathcal{A} for spaces. The symbol \mathbf{x}^\top denotes the transpose of a vector or matrix, and $\|\cdot\|$ the L_2 -norm.

Definition 1. *Let $\mathcal{S} \in \mathbb{R}^M$ be a linear space of dimensionality 1, and let \mathbf{s} be a vector of length 1 spanning \mathcal{S} .*

If some point $\mathbf{x} = [x_1 \cdots x_M]^\top$ lies in the subspace \mathcal{S} spanned by \mathbf{s} we have

$$\frac{1}{\|\mathbf{x}\|} \mathbf{x}^\top \mathbf{s} = 1. \tag{1}$$

Let \mathbf{X} be a $M \times N$ matrix whose column vectors consist of N points $\mathbf{x}_1, \dots, \mathbf{x}_N$. If \mathbf{X} is completely known, the vector \mathbf{s} spanning the subspace is given by the left singular vector corresponding to the largest singular value of \mathbf{X} [14]. In the absence of noise, all singular values except the largest one are identically zero.

If not all coordinates of a particular vector $\mathbf{x}_i, i = 1, \dots, n$, are known, the set of possible vectors gives rise to an affine space \mathcal{A}_i

$$\mathbf{x}_i = \mathbf{A}_i \mathbf{y}_i + \mathbf{t}_i. \tag{2}$$

Here, the matrix \mathbf{A}_i consists of d basis vectors \mathbf{e}_j where d is the number of unknown coordinates of \mathbf{x}_i . Each basis vector \mathbf{e}_j equals 1 at the coordinate corresponding to the missing coordinate of \mathbf{w}_i and is zero elsewhere. The vector \mathbf{t}_i consists of the known coordinates of \mathbf{x}_i and is zero at the coordinates corresponding to the missing coordinates of \mathbf{x}_i . Vector \mathbf{y}_i is chosen so that Eq. (2) holds true. Confer to the left plot of Figure 1.

Lemma 1. *The affine space \mathcal{A}_i and subspace \mathcal{S} intersect.*

Proof. Since \mathbf{x}_i is a point of \mathcal{A}_i , by Eq. (1) it also is a point of \mathcal{S} , hence \mathcal{A}_i and \mathcal{S} intersect.

Let \mathcal{P}_i denote the projective space spanned by \mathcal{A}_i and the origin (shown in the right plot of Figure 1). Given basis vectors $\mathbf{p}_1, \dots, \mathbf{p}_N$ spanning \mathcal{P}_i we can express any point \mathbf{x} on \mathcal{A}_i by

$$\mathbf{x} = c_1 \mathbf{p}_1 + \cdots + c_N \mathbf{p}_N. \tag{3}$$

Theorem 1. *The projective space \mathcal{P}_i and subspace \mathcal{S} intersect.*

Proof. $\mathcal{A}_i \in \mathcal{P}_i$ and $\mathcal{A}_i \cap \mathcal{S}$, hence $\mathcal{P}_i \cap \mathcal{S}$ must hold true.

Theorem 2. *The intersections $\mathcal{S} \cap \mathcal{P}_i$ and $\mathcal{S} \cap \mathcal{P}_j$, $i \neq j$, are identical.*

Proof. By theorem 1 both \mathcal{P}_i and \mathcal{P}_j intersect with \mathcal{S} . Therefore two points $\mathbf{x}_i \in \mathcal{P}_i$ and $\mathbf{x}_j \in \mathcal{P}_j$ exist which are in \mathcal{S} and in \mathcal{P}_i and \mathcal{P}_j , respectively. Assuming that two different lines of intersection \mathbf{s}_i and \mathbf{s}_j of length 1 exist, we have $\mathbf{x}_i^\top / \|\mathbf{x}_i\| \mathbf{s}_i = 1$ and $\mathbf{x}_j^\top / \|\mathbf{x}_j\| \mathbf{s}_j = 1$ by Eq. (1). This implies that \mathcal{S} must be spanned at least by \mathbf{s}_i and \mathbf{s}_j which violates Definition 1. Hence the assumption that $\mathbf{s}_i \neq \mathbf{s}_j$ must be incorrect.

Theorem 2 proves that all projective spaces \mathcal{P}_i and \mathcal{P}_j , $i \neq j$, intersect in \mathbf{s} . It also proves the uniqueness of the solution.

The right of figures 1 depicts the idea of the proof for 3 points \mathbf{x}_i , $i=1,2,3$, having one unknown coordinate each. Here, projective spaces \mathcal{P}_i intersect in a single line \mathbf{s} on which the points \mathbf{x}_i are located.

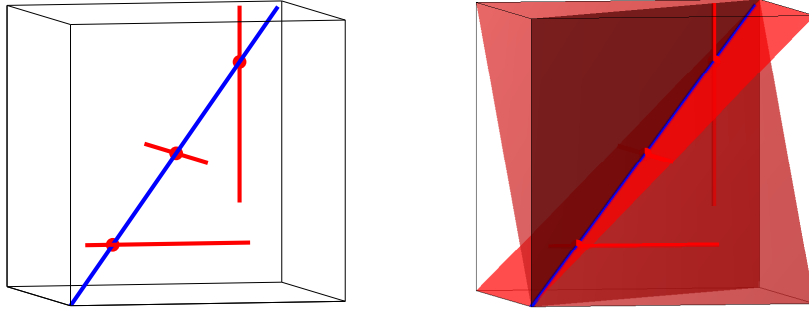


Fig. 1. Left: Three points (red dots) located on a line (blue). Affine spaces induced by unknown coordinates are indicated by red lines. Right: Projective planes (shown in red) through each affine space (red lines) intersect in a single line (blue) which is identical to the subspace on which the data resides.

The remaining question is how to compute \mathbf{s} : Let \mathcal{N}_i be the orthogonal complement of \mathcal{P}_i . A direct consequence of Theorem 2 is that

$$\mathcal{N}_1 \perp \mathbf{s}, \dots, \mathcal{N}_N \perp \mathbf{s}, \quad (4)$$

i.e. \mathbf{s} is orthogonal to all spaces \mathcal{N}_i , $i = 1, \dots, N$. We therefore have

$$\begin{bmatrix} \mathbf{N}_1^\top \\ \vdots \\ \mathbf{N}_N^\top \end{bmatrix} \mathbf{s} = 0. \quad (5)$$

where \mathbf{N}_i denotes the matrix consisting of all normal vectors to \mathcal{P}_i .

Algorithm 1 Linear 1-Dimensional Subspace Estimation

-
- 1: *Input*: Vectors \mathbf{x}_i of which not all coordinates are known.
 - 2: *Output*: The subspace \mathbf{s} on which all *complete* vectors \mathbf{x}_i would reside.
 - 3: Separate each vector \mathbf{x}_i into known coordinates \mathbf{t}_i and unknown coordinates giving rise to an affine space \mathcal{A}_i so Eq. (2) holds true.
 - 4: The basis vectors \mathbf{e}_j of the affine space \mathcal{A}_i and the vector \mathbf{t}_i span the projective space \mathcal{P}_i . Compute the orthogonal complement \mathbf{N}_i of \mathcal{P}_i by SVD.
 - 5: Create matrix \mathbf{N} in Eq. (4).
 - 6: Compute the 1-dimensional nullspace of \mathbf{N} by SVD according to Eq. (5).
-

To determine the M variables of \mathbf{s} , at least M equations are necessary. Each point $\mathbf{x}_i \in R^M$ with u_i unknown coordinates induces a projective space of dimension $u_i + 1$. Hence, $M - (u_i + 1)$ normal vectors exist for each point \mathbf{x}_i . In the absence of noise, if

$$\sum_{i=1}^N (M - (u_i + 1)) \geq M \quad (6)$$

the subspace can be estimated.

The only necessary assumption taken is that the points \mathbf{x}_i are indeed located on a line intersecting the origin, so \mathbf{X} has rank 1 if all its entries are known. If \mathbf{s} does not intersect the origin, i.e. is affine, \mathbf{X} has rank 2.

The algorithm for 1D subspace fitting under incomplete data is summarized in Algorithm 1.

3 Synthetic Evaluation

For a synthetic evaluation of accuracy and required computation time, we randomly created 1000 points in a 100-dimensional space satisfying the rank-1 property. Minimal and maximal values were about -150 and 150 , respectively. We then estimated the subspace using our proposed method and compare it with *PowerFactorization* [11], an EM-algorithm where subspace and missing data are estimated alternately, and an implementation of a nuclear norm minimization algorithm (NNM) [5]². Our implementation of the EM-algorithm is similar to [19] yet without considering statistical reliability of trajectories. All algorithms were implemented using MatLab. The angular difference between the ground truth and the estimated vector was adopted as measure for the error.

We randomly removed entries from the data matrix consisting of all points. We gradually increased the number of entries to be removed in steps of 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, 98%. To evaluate the robustness three different levels of independent, normally distributed noise were added to the data. Standard deviations of noise were set to 0, 0.1, and 1.0. The last level is equivalent to a strong pixel noise of about 3%. For each combination of noise and sampling ratio, all four algorithms were executed ten times and the average

² The code is generously provided at svt.caltech.edu.

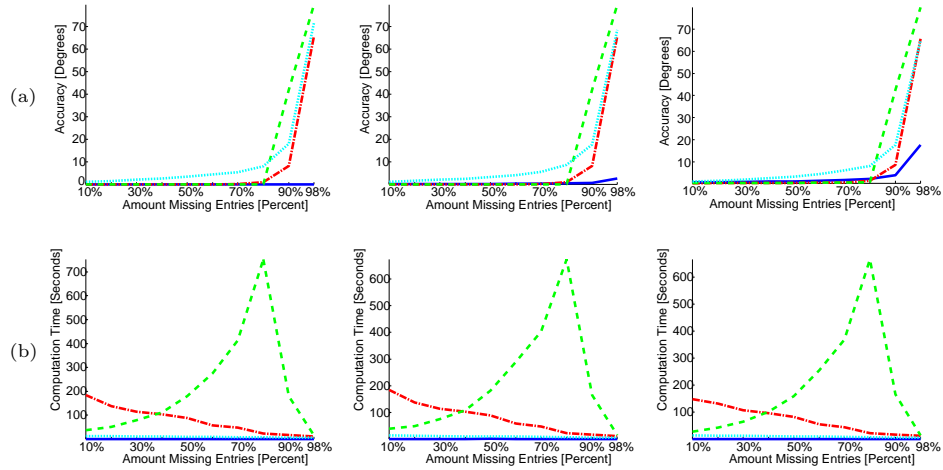


Fig. 2. (a) Average angular difference between ground-truth and estimation for several sampling ratios. The red dash-dotted line indicates PowerFactorization, the green dashed line the EM-algorithm, the cyan dotted line nuclear norm minimization and the blue solid line the proposed method. (b) Average computation time in seconds. The proposed method needs between $0.43sec$ and $0.84sec$. Even for a sampling ratio of 98% where PowerFactorization is fastest, our method is 25 times faster while being much more accurate. Nuclear norm minimization is between 12 and 17 times slower for all sampling ratios and noise levels while being significantly less accurate even for relatively complete data.

angular error and computation time computed. Since PowerFactorization and the EM-approach are susceptible to local minima we randomly initialized them 50 and 5 times, respectively, and took the best result. According to our experience, choosing more than 50 or 5 repetitions does not change the optimal result. All computations were performed on an Intel Core2 Quad CPU with 3.0GHz.

Figure 2(a) shows the average angular errors if there was no noise in the data (left), small pixel noise with standard deviation $\sigma = 0.1$ (middle), and strong noise with $\sigma = 1.0$. The x -axis shows the amount of missing entries in percent, the y -axis the average angular error. The red dash-dotted line indicates the error due to PowerFactorization, the green dashed line the EM-method, the cyan dotted line NNM, and the solid blue line our algorithm. As can be seen, for less than 80% of missing entries, PowerFactorization and the EM-method perform approximately same as good as the proposed method, only NNM is between 1.4 and almost 10000 times less accurate. Beyond 80% missing data, the proposed algorithm is much more accurate than all other algorithms. NNM regularly fails to estimate a subspace of dimension 1, but computes a higher-dimensional solution.

The average computation times are shown in Fig. 2(b). The left plot in Fig. 2(b) shows the computation time in seconds on the y -axis if the data is noise-free, the middle plot if the noise is small ($\sigma = 0.1$), and the right plot shows results for strong noise ($\sigma = 1.0$). The proposed procedure requires less between 0.43sec and 0.84sec of time for all three experiments. Our algorithm is between 12 and 17 times faster than NNM, and between 25 and 240 times faster than PowerFactorization. For all combinations of sampling ratios and noise, the proposed method is at least same as accurate while being magnitudes faster and immune to local minima. Nuclear norm minimization usually fails to estimate the correct subspace from simple data like the one shown in Fig. 1.

4 Real-Image Application

In this section we present a real image application. It is known that 3D-reconstruction from point correspondences can be achieved by means of an iterative version of the so-called factorization algorithm [15, 12]. It involves fitting a 4-dimensional subspace to data. However, this algorithm cannot be used if some data is missing. If the camera is calibrated and does not rotate, the rank-4 constraint can be reduced to rank-1. Under incomplete data, the method of Sec. 2 can be applied then.

In Sec. 4.1 we shortly review the projective factorization algorithm [15, 12]. In Sec. 4.2 we will consider the case when not all feature points are known in all images. We will show that the standard rank-4 constraint reduces to a rank-1 problem if intrinsic and rotation parameters are known. 3D-reconstructions from challenging out-door sequences are shown in Sec. 4.3.

4.1 Projective Factorization

Assuming a perspective camera model, the i th camera is defined by the matrix

$$\mathbf{P}_i = \mathbf{K} [\mathbf{R}_i \mid \mathbf{t}_i]. \quad (7)$$

Here, vector \mathbf{t}_i contains the translation parameters of the i th camera, matrix \mathbf{R}_i the rotation parameters, and matrix \mathbf{K} the intrinsic parameters such as focal length f , principal point $\mathbf{p} = [p_x \ p_y]^\top$ and photo sensor scales a_x and a_y

$$\mathbf{K} = \begin{bmatrix} f a_x & 0 & p_x \\ 0 & f a_y & p_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (8)$$

While we assume here that intrinsic parameters are known and fixed throughout the sequence, varying intrinsics are possible as long as they are known which is the case for many real-world applications.

The j th homogeneous 3D-point $\mathbf{X}_j = [X_j \ Y_j \ Z_j \ 1]^\top$ is projected onto an image point $\mathbf{x}_{ij} = [u_{ij} \ v_{ij} \ 1]^\top$ by

$$\lambda_{ij} \mathbf{x}_{ij} = \mathbf{P}_i \mathbf{X}_j. \quad (9)$$

The projective depth λ_{ij} is a scalar to ensure that the third coordinate of \mathbf{x}_{ij} equals 1.

Given N points in M images, we may arrange all image measurements \mathbf{x}_{ij} into an observation matrix \mathbf{W}

$$\mathbf{W} = \begin{bmatrix} \lambda_{11}\mathbf{x}_{11} & \cdots & \lambda_{1N}\mathbf{x}_{1N} \\ \vdots & \ddots & \vdots \\ \lambda_{M1}\mathbf{x}_{M1} & \cdots & \lambda_{MN}\mathbf{x}_{MN} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_M \end{bmatrix}}_{\mathbf{P}} \underbrace{[\mathbf{X}_1 \cdots \mathbf{X}_N]}_{\mathbf{X}}. \quad (10)$$

With some risk of confusion, denote the stack of all camera matrices by \mathbf{P} and the matrix of all homogeneous 3D-points by \mathbf{X} . Equation (10) implies that the columns of matrix \mathbf{W} span a 4-dimensional subspace if all scalars λ_{ij} are known.

Equation (10) may be written as

$$\left(\mathbf{I} - \mathbf{P}\mathbf{P}^\top\right)\mathbf{W} = 0, \quad (11)$$

if the column vectors of \mathbf{P} have unit length and are mutually orthogonal. The symbol \mathbf{I} denotes the identity matrix. Equation (11) can be used to compute $\left(\mathbf{I} - \mathbf{P}\mathbf{P}^\top\right)$ as left nullspace of \mathbf{W} .

Conversely, if \mathbf{P} is known, the projective depths λ_{ij} may be computed by solving

$$\left(\mathbf{I} - \mathbf{P}\mathbf{P}^\top\right)\mathbf{D}_j\boldsymbol{\lambda}_j = 0 \quad (12)$$

for $\boldsymbol{\lambda}_j$ where \mathbf{D}_j denotes a $3M \times M$ matrix consisting of the measurements \mathbf{x}_{ij} of the j th trajectory on its diagonal, and $\boldsymbol{\lambda}_j$ the vector consisting of the corresponding variables λ_{ij} .

If both the variables λ_{ij} and the camera matrices \mathbf{P} are unknown, a projective reconstruction can be estimated by initializing the λ_{ij} , for instance to 1, and iteratively solving Eqs. (11) and (12). To avoid a trivial solution we impose the constraint that the columns of \mathbf{W} should have unit length. If all observations \mathbf{x}_{ij} are normalized to length 1, this constraint is satisfied if all vectors $\boldsymbol{\lambda}_j$ have length 1 which is automatically enforced by standard tools for computing the singular value decomposition.

4.2 Missing Observations

The procedure in Sec. 4.1 may only be used if all observations \mathbf{x}_{ij} are known across all images. Unfortunately, this is a rather unrealistic requirement. For instance, feature points can disappear due to failure of the tracking software, or due to occlusion in the scene. Conversely, the method introduced in Sec. 2 can fit subspaces to incomplete data yet it requires a rank-1 constraint. In the following, we will show that the rank-4 constraint can be reduced to rank-1 so a projective reconstruction can be computed from data where feature points

are missing: In many real-world problems, the intrinsic camera parameters are known or can be easily estimated. Furthermore, in applications like car or robot navigation, information about the rotation of the vehicle is available. If both \mathbf{K} and all rotation matrices \mathbf{R}_i are known, the only remaining unknown is the vector $\mathbf{t} = [\mathbf{t}_1^\top \cdots \mathbf{t}_M^\top]^\top$. The known measurements \mathbf{x}_{ij} are weighted with \mathbf{K}^{-1} . Unknown observations $\tilde{\mathbf{x}}_{ij}$ are initialized to $\frac{1}{\sqrt{3}} [1 \ 1 \ 1]^\top$. From Eq. (11) we have the requirement that

$$(\mathbf{I} - \mathbf{t}\mathbf{t}^\top) \mathbf{W} = 0. \quad (13)$$

However, Eq. (13) does not sufficiently constrain \mathbf{t} . It has also be orthogonal to the known subspace spanned by

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_M \end{bmatrix}. \quad (14)$$

Since vector \mathbf{t} should be orthogonal to \mathbf{R} , it must reside in the orthogonal complement of \mathbf{R} . Letting

$$\mathbf{N} = \begin{bmatrix} \mathbf{N}_1^\top \\ \vdots \\ \mathbf{N}_N^\top \end{bmatrix}, \quad (15)$$

where the matrices \mathbf{N}_i are the normals to the projective spaces induced by trajectory vectors with missing coordinates. If we combine the orthogonality constraint with Eq. (5) we then obtain

$$(\mathbf{I} - \mathbf{R}\mathbf{R}^\top) \mathbf{N}\mathbf{t} = 0. \quad (16)$$

If the vector \mathbf{t} has been determined, the missing observations can be estimated by minimizing the distance to the subspace $\mathbf{P} = [\mathbf{R} \ \mathbf{t}]$

$$(\mathbf{I} - \mathbf{P}\mathbf{P}^\top) \mathbf{A}_j \mathbf{d}_j = 0 \quad (17)$$

where \mathbf{A}_j denotes a $3M \times 3M$ matrix with vectors $[\lambda_{ij} \ \lambda_{ij} \ \lambda_{ij}]^\top$ on each triple of its diagonal entries, and \mathbf{d}_j the vector of all observations $\mathbf{x}_{1j} \cdots \mathbf{x}_{Mj}$. Similar to [1], \mathbf{d}_j can be separated into known and unknown feature points thus the unknown observations can be estimated by solving a linear equation system.

Since the variables λ_{ij} are not known generally, it is necessary to estimate the vector \mathbf{t} , the projective depths λ_{ij} , and all missing observations $\tilde{\mathbf{x}}_{ij}$. We obtain a projective reconstruction by iterating the following three steps: estimate \mathbf{t} by Eq. (16), estimate all unknown observations $\tilde{\mathbf{x}}_{ij}$ by Eq. (17), and optimize the projective depths λ_{ij} by Eq. (12).

Iterations may be terminated if the difference of reprojection errors

$$\epsilon = \frac{1}{L} \sum_{i=1}^M \sum_{j=1}^N \delta_{ij} \mathbf{K} (\mathbf{x}_{ij} - N [\mathbf{P}_i \mathbf{X}_j]) \quad (18)$$

between two consecutive iterations becomes sufficiently small. Here, δ_{ij} equals one if \mathbf{x}_{ij} was observed and is zero otherwise. The symbol $N[\cdot]$ denotes the scale normalization to make the third coordinate equal to 1. We use the normalization

$$L = \sum_{i=1}^M \sum_{j=1}^N \delta_{ij}. \quad (19)$$

4.3 Real-Image Sequences

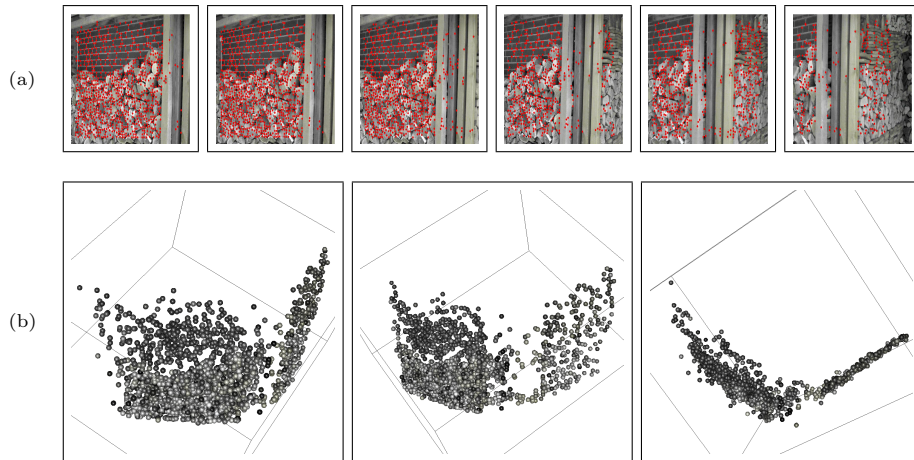


Fig. 3. (a) Six images of a 50-image sequence with 1700 trajectories. A total of 40.5% of all feature points (indicated by the red points) is known. The iterative projective factorization converged to a reprojection error of $3.24[px]$. (b) 3D-reconstruction using the algorithm proposed in Sec. 4.2. The right wall is planar. The angle between floor and the two walls is not exactly perpendicular, but both walls are almost orthogonal. The depth difference between left wall and wood logs is clearly recognizable.

Figures 3(a), 4(a), and 5(a) each show six images of three sequences which were taken with the video function of a consumer-level digital camera (Canon PowerShot A530). The images have size 640×480 pixels and are of low quality.

Correspondences were established using a standard software (*voodoo camera tracker*³). We did not check manually for matching errors, so it is possible that outliers are present in the data. Particular trajectories, e.g. such of short length, were not excluded either. Rotational motion is small but present since we did not carefully arrange the camera. This induces some model-noise which is not independently distributed. Intrinsic calibration parameters were determined using [21]. The 3D-reconstructions are not further optimized by a bundle adjustment.

³ www.digilab.uni-hannover.de/docs/manual.html

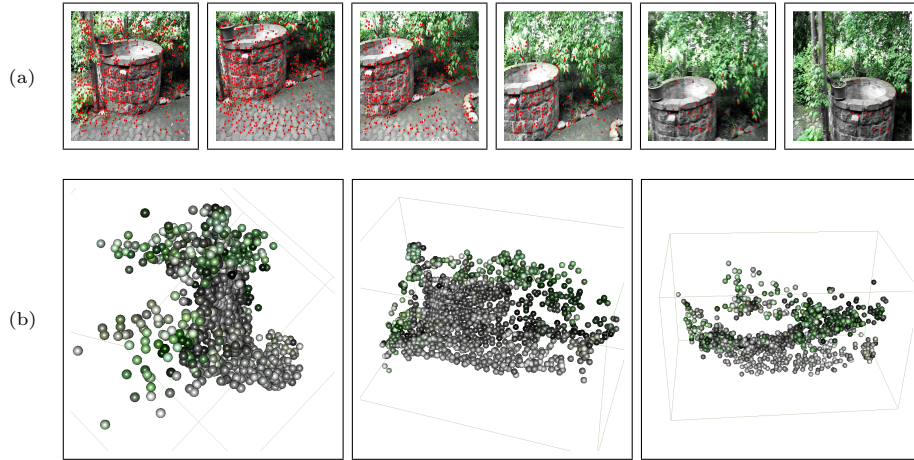


Fig. 4. (a) Six images of a 60-image sequence with 1500 trajectories. A total of 36.7% of all feature points (indicated by the red points) is known. The iterative projective factorization converged to a reprojection error of $3.0[px]$. (b) 3D-reconstruction using the algorithm proposed in Sec. 4.2. The left image shows a side view, the middle one a front view, and the right image a view from top. While the curvature of the well is slightly underestimated, it is well perceivable. Well and floor are perpendicular, and the depth ratios between plants (green points) and well are reasonable.

The sequence corresponding to Fig. 3(a) consists of 1700 trajectories which were tracked over 50 images. A total of 40.5% of all feature points are known. The left plot of Fig. 6 indicates where feature points are known and where unknown. The horizontal axis indicates the number of the trajectory, the vertical axis the image number. Feature points missing in some image are marked by black, present features by white. We stopped the projective factorization algorithm after 30 iterations resulting in a reprojection error of 3.24 (pixels). Figure 3(b) shows three images of the 3D-reconstruction. The outline of the wooden logs in the left part is clearly visible. The angle between ground plane and the two planes is not exactly rectangular, but the angle between both planes is quite orthogonal (shown in the third, top-view image), and the surface of the right wall is almost perfectly planar what can be seen in the first image.

The second sequence, corresponding to the six images shown in Fig. 4(a), consists of 1500 trajectories over 60 images with 36.7% of the feature points being known. Absence and presence of feature points is shown by the middle plot in Fig. 6. The algorithm converged to a reprojection error of $3.0[px]$ after 50 iterations. Three images of the 3D-reconstruction of the sequence are shown in Fig. 4(b). The well can be seen in the center of the images. While the curvature of the well is underestimated (shown in the third, top-view image), the angle between floor and well is almost rectangular. The depth ratios between plants (green points) and well look reasonable.

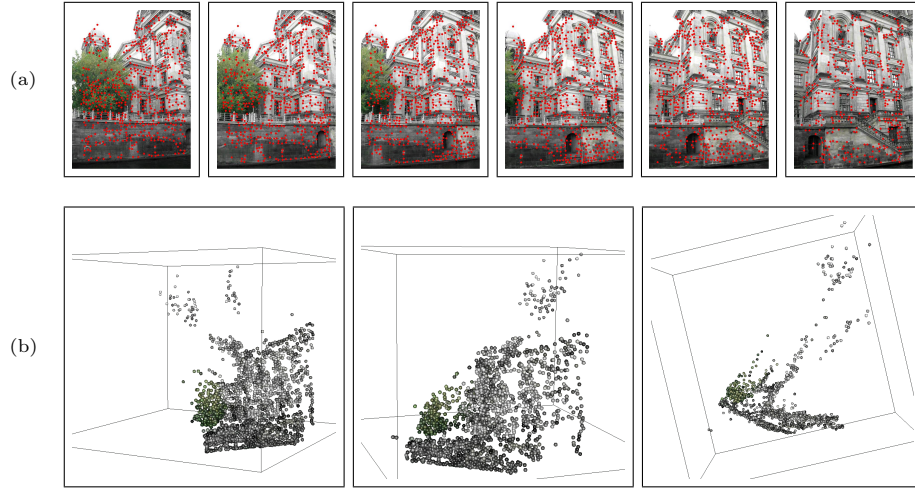


Fig. 5. (a) Six images of a 60-image sequence with 2000 trajectories. A total of 31.38% of all feature points (indicated by the red points) is known. The iterative projective factorization converged to a reprojection error of $2.3[px]$. (b) 3D-reconstruction using the algorithm proposed in Sec. 4.2. The walls of the building, the tree to the left and the stairs in the lower left part of the images can be well perceived. The angle between the two main walls is not exactly rectangular but still looks good.

The last sequence, corresponding to Fig. 5(a), consists of 2000 trajectories over 60 images with 31.38% of the feature points tracked (cf. right plot of Fig. 6). The reprojection error was $2.3[px]$ after 20 iterations. The 3D-reconstruction is shown in Fig. 5(b). The structure of the building is clearly visible. The two main walls, the tree in the left part and the stairs in the lower right part can be seen. The top-view image shows that the angle between the two walls is slightly distorted.

5 Conclusions

In this work, we presented an algorithm for fitting a 1-dimensional subspace to incomplete data. We proved that for this special case, a linear solution exists and is unique. Using simulated data we showed that for various levels of noise, the proposed algorithm is more accurate than other methods if more than 80% of the data is missing. It is magnitudes faster while being same as accurate or even more precise. We demonstrated that our method is able to compute 3D-reconstructions of low-quality real-image sequences with large amounts of unobserved correspondences.

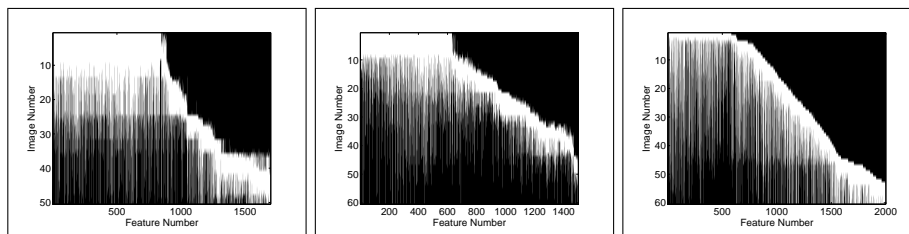


Fig. 6. Presence of the trajectories of the sequences shown in Figs. 3(a) (left), 4(a) (middle) and 5(a) (right). The number of each trajectory is indicated by the horizontal axis, the image number by the vertical axis. White represents known feature points, black absent points.

References

1. H. Ackermann and B. Rosenhahn. Trajectory Reconstruction for Affine Structure-from-Motion by Global and Local Constraints. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, Miami, Florida, USA, June 2009.
2. P. Aguiar, J. Xavier, and M. Stosic. Spectrally Optimal Factorization of Incomplete Matrices. In *IEEE Computer Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.
3. M. Brand. Incremental singular value decomposition of uncertain data with missing values. In *7th European Conference on Computer Vision (ECCV)*, pages 707–720, London, UK, May 2002.
4. A. Buchanan and A. Fitzgibbon. Damped Newton Algorithms for Matrix Factorization with Missing Data. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 316–322, Washington, DC, USA, 2005.
5. J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.
6. E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.
7. P. Chen and D. Suter. Recovering the Missing Components in a Large Noisy Low-Rank Matrix: Application to SFM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1051–1063, August 2004.
8. A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.
9. M. Fazel. *Matrix Rank Minimization with Applications*. PhD thesis, Dept. Electrical Engineering, Stanford University, March 2002.
10. K. Gabriel and S. Zamir. Lower rank approximation of matrices by least squares with any choice of weights. *Technometrics*, 21(4):489–498, 1979.
11. R. Hartley and F. Schaffalitzky. PowerFactorization: 3D Reconstruction with Missing or Uncertain Data. In *Australia-Japan Advanced Workshop on Computer Vision*, June 2002.
12. A. Heyden, R. Berthilsson, and G. Sparr. An Iterative Factorization Method for Projective Structure and Motion from Image Sequences. *Image and Vision Computing*, 17(13):981–991, November 1999.

13. D. W. Jacobs. Linear fitting with missing data for structure-from-motion. *Computer Vision and Image Understanding*, 82(1):57–81, April 2001.
14. K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier Science Inc., New York, NY, USA, 1996.
15. S. Mahamud and M. Hebert. Iterative Projective Reconstruction from Multiple Views. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 430–437, Hilton Head, SC, USA, June 2000.
16. M. Marquez and J. Costeira. Optimal Shape from Motion Estimation with Missing and Degenerate Data. In *IEEE Workshop on Application of Computer Vision (WACV)*, Copper Mountain, CO, USA, January 2008.
17. A. Ruhe. Numerical computation of principal components when several observations are missing. Technical report, Dept. Information Processing, University of Umeda, Umeda, Sweden, April 1974.
18. A. Ruhe and P. Wedin. Algorithms for separable nonlinear least squares problems. *Society for Industrial and Applied Mathematics Review*, 22(3):318–337, 1980.
19. Y. Sugaya and K. Kanatani. Extending Interrupted Feature Point Tracking for 3-D Affine Reconstruction. In *8th European Conference on Computer Vision (ECCV)*, pages 310–321, Prague, Czech Republic, May 2004.
20. C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2):137–154, November 1992.
21. R. Tsai. A versatile camera calibration technique for high-accuracy 3-d machine vision metrology using off-the-shelf cameras and lenses. *IEEE Transaction on Robotics and Automation*, 3(4):323–344, 1987.
22. T. Wiberg. Computation of principal components when data are missing. In *Second Symp. on Computational Statistics*, pages 229–236, Berlin, Germany, 1976.
23. H. Wold. Estimation of principal components and related models by iterative least squares. In Krishnaiah, editor, *Multivariate Analysis*, pages 391–420, 1966.