Linear Multi View Reconstruction with Missing Data

Carsten Rother and Stefan Carlsson *

Computational Vision and Active Perception Laboratory (CVAP)
Dept. of Numerical Analysis and Computer Science
KTH, SE-100 44 Stockholm, Sweden
{carstenr, stefanc}@nada.kth.se

Abstract. General multi view reconstruction from affine or projective cameras has so far been solved most efficiently using methods of factorizing image data matrices into camera and scene parameters. This can be done directly for affine cameras [18] and after computing epipolar geometry for projective cameras [17]. A notorious problem has been the fact that these factorization methods require all points to be visible in all views. This paper presents alternative algorithms for general affine and projective views of multiple points where a) points and camera centers are computed as the nullspace of one linear system constructed from all the image data b) only three points have to be visible in all views. The latter requirement increases the flexibility and usefulness of 3D reconstruction from multiple views. In the case of projective views and unknown epipolar geometry, an additional algorithm is presented which initially assumes affine views and compensates iteratively for the perspective effects. In this paper affine cameras are represented in a projective framework which is novel and leads to a unified treatment of parallel and perspective projection in a single framework. The experiments cover a wide range of different camera motions and compare the presented algorithms to factorization methods, including approaches which handle missing data.

Keywords: Structure from Motion, Linear Multiple View Reconstruction, Missing Data, Affine and Projective Cameras.

1 Introduction

Efficient 3D reconstruction from multiple camera views is a problem of great importance in computer vision with far reaching applications. It has also received considerable attention over the years as seen from the number of publications [2, 5,6,7,9,10,11,12,13,14,15,16,17,18,19,20,21] and books [4,1] devoted to the topic. It is generally accepted that for parallel projection the factorization method of

* This work was supported by the Swedish Foundation for Strategic Research in the VISIT program.

A. Heyden et al. (Eds.): ECCV 2002, LNCS 2351, pp. 309–324, 2002.

Tomasi-Kanade [18] is numerically the most satisfying. It is optimal under the assumption of isotropic Gaussian noise in the image data. For perspective projection, the projective factorization of Sturm-Triggs [17,19] has been demonstrated to be one of the most numerically efficient methods, see e.g. [5]. This method is similar to affine factorization but requires known epipolar geometry. Both these methods have a major disadvantage, however, in the fact that they require all points to be visible in all views. This of course limits their usefulness in most common multiple view situations where points eventually will be occluded as the camera viewpoint changes. Some suggestions to overcome this problem has been made [9,18], but they require careful analysis and selection of image data in order to be used. Alternative methods for handling missing data for affine [10] and projective views [20] use the so-called closure constraints. The idea is to obtain the camera's motion linearly and simultaneously from a series of bi- or tri-focal tensors.

Ideally, an algorithm for multiple view reconstruction should utilize *all* available image data directly in an efficient manner. Obviously, a minimum overlap of views is necessary for the computation of relative camera positions [14].

In [15] a linear algorithm for multi view reconstruction was presented which requires four coplanar reference points to be visible in all views. All image data, except for image data from points on this reference plane, is used directly to recover points and camera centers simultaneously. The key idea is to map the reference plane to infinity which transforms the projective multi camera situation to the case of purely translating calibrated cameras. In this paper we will demonstrate that there are more multiple view situations which can be transformed to this mathematically simpler structure of translating calibrated cameras. Namely, these are exactly the cases for which affine and projective factorization can be applied:

- general affine cameras
- general projective cameras with known relative epipolar geometry.

No assumption about the scene structure is needed. The main differences of our approach to the bilinear factorization methods are:

- the selection of a finite plane as the plane at infinity
- the allowance of arbitrary missing data, with three points visible in all views
- the computation of the null space of one image data matrix.

The fact that a finite plane will be mapped to infinity is a potential problem for numerical calculations. However, we will demonstrate practically and experimentally that this problem can be handled. Additionally, we present an iterative algorithm for the case of projective cameras and unknown epipolar geometry that initially assumes affine views and compensates iteratively for the perspective effects. A similar idea has been suggested by [4,6,19] to circumvent the pre-estimation of the epipolar geometry for projective factorization. In this paper affine cameras are represented in a projective framework. This is novel and leads to a unified treatment of parallel and perspective projection in a single framework.

2 Structure, Motion and the Infinite Homography

General perspective projection of a 3D point P_i onto the 2D image point p_{ij} can be described in homogeneous coordinates as:

$$p_{ij} \sim H_j \left(I \mid -\bar{Q}_j \right) P_i \sim H_j \left(\bar{P}_i - \bar{Q}_j \right), \tag{1}$$

where $H_j \left(I \mid -\bar{Q}_j \right)$ represents the 3×4 projection matrix of camera j. Non-homogeneous coordinates are denoted with a bar, e.g. $\bar{Q}_j$, and homogeneous coordinates without a bar, e.g. p_{ij}. The symbol "$\sim$" means equality up to scale. Let us consider the homography H_j in more detail. A point $P = (X, Y, Z, 0)^T$, which lies on the plane at infinity π_∞, is mapped by eqn. (1) onto the image plane π_j as:

$$p_{ij} \sim H_j \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}. \tag{2}$$

Therefore, H_j can be considered as the *infinite homography*[1] between the plane at infinity π_∞ and the image plane π_j. From eqn. (1) we see that if H_j is known, we are left with a linear and symmetric relationship between non-homogeneous points and camera centers:

$$p^*_{ij} \sim H_j^{-1} p_{ij} \sim \bar{P}_i - \bar{Q}_j. \tag{3}$$

This suggests the following approach for structure and motion recovery:

1. *Determine the infinite homographies H_j*
2. *Reconstruct points and camera centers.*

Section 3 will discuss several ways to determine H_j for affine views, projective views and scenes containing a reference plane.

If H_j is known, eqn. (3) can be transformed into three projection relations, where the unknown scale is eliminated by taking ratios:

$$\begin{aligned} -y^*_{ij}\, \bar{X}_i + x^*_{ij}\, \bar{Y}_i + y^*_{ij}\, \bar{A}_j - x^*_{ij}\, \bar{B}_j &= 0 \\ -w^*_{ij}\, \bar{X}_i + x^*_{ij}\, \bar{Z}_i + w^*_{ij}\, \bar{A}_j - x^*_{ij}\, \bar{C}_j &= 0 \\ -w^*_{ij}\, \bar{Y}_i + y^*_{ij}\, \bar{Z}_i + w^*_{ij}\, \bar{B}_j - y^*_{ij}\, \bar{C}_j &= 0 \end{aligned} \tag{4}$$

with $p^*_{ij} = (x^*_{ij}, y^*_{ij}, w^*_{ij})^T$, $\bar{P}_i = (\bar{X}_i, \bar{Y}_i, \bar{Z}_i)^T$ and $\bar{Q}_j = (\bar{A}_j, \bar{B}_j, \bar{C}_j)^T$. Therefore, each scene point $\bar{P}_i$ visible in view j provides three linear relations of the form (4) which can be put into a set of linear equations (SLE). For n points and m views the SLE has the form (explicit in [15]):

$$\begin{aligned} &L\ h = 0 \text{ with} \\ &h = (\bar{X}_1, \bar{Y}_1, \bar{Z}_1, \ldots, \bar{X}_n, \bar{Y}_n, \bar{Z}_n, \bar{A}_1, \bar{B}_1, \bar{C}_1, \ldots, \bar{A}_m, \bar{B}_m, \bar{C}_m)^T. \end{aligned} \tag{5}$$

[1] Note, the definition of the infinite homography is slightly different to [4,1].

The Singular Value Decomposition (SVD) of L shows that L has a four dimensional null-space. However, three of the four singular vectors of the null-space have the trivial form: $\bar{P}_i = \bar{Q}_j = (1,0,0)^T$, $\bar{P}_i = \bar{Q}_j = (0,1,0)^T$ and $\bar{P}_i = \bar{Q}_j = (0,0,1)^T$. Therefore, the summation of all four singular vectors of the null-space gives the non-trivial solution for all camera centers and points. However, points on the plane at infinity π_∞ increase the dimensionality of the null-space of L (see [15]). Therefore, the projection relation of those points have to be excluded from the SLE. Since the infinite homography H_j is known, those points can be reconstructed directly with eqn. (2). How such points are detected automatically and how the SLE is formulated in an optimal way will be discussed in section 4. Let us summarize the main advantages of this approach:

- *One linear system containing all image data*
- *Missing data can be handled*
- *Points and cameras are determined simultaneously.*

3 Determine the Infinite Homographies

It was shown in [15] that the infinite homographies H_j can be determined if the scene contains a reference plane visible in all views. However, in this section we will show that H_j can be determined for affine or projective cameras and *general scenes* without constraints on the scene structure. Additionally, for projective cameras the epipolar geometry has to be known. Let us begin with the reference plane case.

3.1 Reference Plane

In order to determine H_j, it was assumed in [15] that four coplanar scene points P_1, P_2, P_3, P_4 are visible in all views. The coordinates of these reference points were chosen in a canonical way in the projective space P^3. Furthermore, the image basis p_{ij} was mapped to the normalized projective image basis p^*_{ij} with $p_{1j}, p_{2j}, p_{3j}, p_{4j}$ as basis points. This corresponds to the following mapping of the reference points onto the image plane j:

$$\begin{array}{cccc} P_1 & P_2 & P_3 & P_4 \\ - & - & - & - \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{array} \longrightarrow \begin{array}{cccc} p^*_{1j} & p^*_{2j} & p^*_{3j} & p^*_{4j} \\ - & - & - & - \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{array} . \tag{6}$$

We see that all the reference points lie on the plane at infinity. This means that in the particular chosen projective space, the reference plane is the plane at infinity. This transformation was noted in [21] as a fundamental requirement for obtaining the linear relationship between points and camera centers (see eqn. (3)). The infinite homography of each image j can now be derived from eqn. (2) as $H_j = I$. Alternatively, H_j could be derived from the inter-image homographies induced by the reference plane.

3.2 Affine Cameras

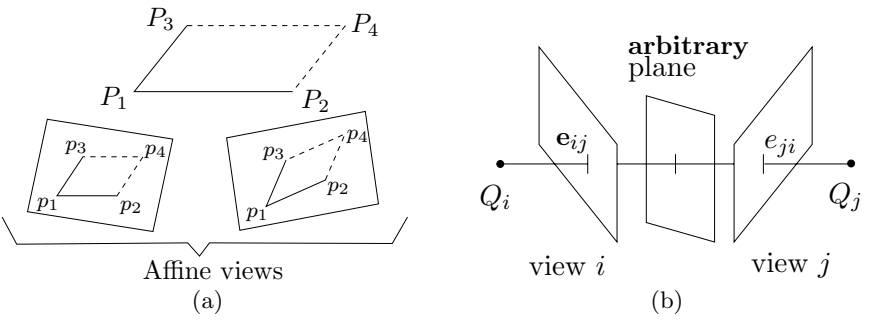

Fig. 1. Determine a fourth coplanar point for affine (a) and projective (b) views.

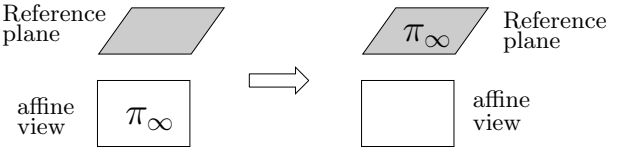

Fig. 2. Moving the plane at infinity π_∞ from its "true" location to the reference plane.

Let us assume that three reference points P_1, P_2, P_3 are visible in all views. These three points uniquely define a reference plane. The basic idea is to deduce a fourth "virtual" reference point which lies on the reference plane as well. Let us define the coplanar 3D point P_4 as $P_4 = P_3 + P_2 - P_1$ (see fig. 1 (a)). Since affine cameras perform a parallel projection on scene points, the affine image of P_4 in view j is $p_{4j} = p_{3j} + p_{2j} - p_{1j}$. Alternatively, the fourth point could be chosen as the centroid of the three reference points.

However, how are affine cameras embedded in the projective framework derived in the previous section? Let us reconsider the mapping of a general projective camera as in eqn. (1):

$$p_{ij} \sim \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ v_1 & v_2 & v_3 & v_4 \end{pmatrix} P_i. \tag{7}$$

The last row of the camera matrix is the *principle plane* $\pi_{prin} = (v_1, v_2, v_3, v_4)^T$ of the camera which contains the camera center and is parallel to the image plane. In a projective space where the plane at infinity is at its true location, the principle plane of an affine camera is the plane at infinity, i.e. $\pi_{prin} = \pi_\infty = (0, 0, 0, 1)^T$ (fig. 2 left). However, we have seen in the previous section that in order to determine H_j, the reference plane has to be the plane at infinity in the particular chosen projective space (fig. 2 right). This means that in this particular projective space all camera centers lie on a plane π_{prin} which is different to π_∞. Eqn.

(7) can now be transformed into eqn. (1) with non-homogeneous coordinates for the camera centers and scene points. From the four coplanar reference points, the infinite homographies H_j can be derived with eqn. (2) and (6). The reconstructed cameras provide the principle plane π_{prin}, which contains all camera centers. Finally, by mapping π_{prin} to π_∞ the projective reconstruction transforms into an affine reconstruction.

How does this approach compare to other affine reconstruction methods? In our approach 6 parameters of each affine camera are determined directly by the infinite homographies. The remaining 2 unknown parameters, which represent the direction of an affine camera, are reconstructed simultaneously with the scene points. In contrast to this, affine factorization [18] determines 2 parameters of each affine camera in forehand. The remaining 6 parameters of each camera are determined simultaneously with the scene points. However, this method does not allow missing data. It has been shown [6,12] that all 8 unknown parameters of an affine camera could be determined directly by choosing a special affine basis in the scene and in the image. However, from an numerical point of view this is less favourable.

3.3 Projective Cameras

Let us assume that the three reference points P_1, P_2, P_3 have canonical coordinates in the projective space and in the image as in eqn. (6). The infinite homography for each view j is then described as:

$$H_j = \begin{pmatrix} a_j & 0 & 0 \\ 0 & b_j & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{8}$$

The arbitrary scale of the matrix is fixed by setting $H_j(3,3) = 1$. The variables a_j and b_j are unknown in each view j and can be considered as the mapping of point $(1,1,1,0)^T$ into view j: $H_j\ (1,1,1)^T = (a_j, b_j, 1)^T$. Let us assume that the epipolar geometry is known, i.e we have the fundamental matrices between each pair of views which have at least seven points in common. We denote the epipole $e_{ij} = (e_{ijx}, e_{ijy}, e_{ijw})^T$ as the projection of camera center j into view i (see fig. 1(b)). The inter-image homography from view i to view j via a certain plane is defined as $H_{ij} = H_i^{-1}\ H_j$ where H_i, H_j denote the respective homographies as defined in eqn. (1). Since the epipols between two views are in correspondence via any plane in the scene (see fig. 1(b)), we may write:

$$e_{ji} \ \sim\ H_{ij}\ e_{ij} \quad \Leftrightarrow \quad H_i\ e_{ji} \ \sim\ H_j\ e_{ij}. \tag{9}$$

Taking equation (8) and (9) we obtain two constraints between views i and j:

$$\begin{aligned} a_i\ e_{jix}\ e_{ijw} \ -\ a_j\ e_{ijx}\ e_{jiw} &= 0 \\ b_i\ e_{jiy}\ e_{ijw} \ -\ b_j\ e_{ijy}\ e_{jiw} &= 0. \end{aligned} \tag{10}$$

All the a_i's and b_i's may now be determined separately but simultaneously. Each pair of images i and j, which are linked by a fundamental matrix, gives an linear

equation in a_i, a_j and b_i, b_j respectively. With m images we obtain two sets of linear equations:

$$\begin{aligned} L_a\ h_a = 0 \text{ with } h_a = (a_1, \ldots, a_m)^T \text{ and} \\ L_b\ h_b = 0 \text{ with } h_b = (b_1, \ldots, b_m)^T. \end{aligned} \tag{11}$$

The last singular vector of the SVD of L_a and L_b gives the solution for h_a and h_b respectively. The vector h_a and h_b have an arbitrary scale which corresponds to the fact that the fourth unknown reference point on the reference plane has two degrees of freedom.

The advantage of deriving the infinite homographies in this way is that all homographies are determined in one step which implies that the complete information given by the geometry is used simultaneously.

3.4 Known Structure and Cameras

For an iterative structure and motion algorithm, we would like to update the infinite homography H_j on the basis of known 3D scene points and cameras. This means that P_i and $\bar{Q}_j$ are known and we obtain:

$$p_{ij} \ \sim\ H_j\ (\ I\ |\ -\bar{Q}_j)\ P_i \ \sim\ H_j\ p'_{ij} \tag{12}$$

where p'_{ij} is the projection of point P_i by camera $(\ I\ |\ -\bar{Q}_j)$. Since p_{ij} and p'_{ij} are known, the infinite homography H_j can be determined for each image j individually with e. g. the normalized 8-point algorithm (see [3]).

3.5 Choice of Reference Points

In practice more than three points might be visible in all views of a multiple view situation. Naturally the question arises of how to find the optimal three reference points. Let us consider the criteria for good reference points. Firstly, a camera center must not lie on the reference plane. This means that the three reference points must not be collinear in any view. Secondly, in the presence of noise the infinite homography is determined more accurately if the projected reference points are far apart in the image. Since the two criteria are not contradictive, we choose as reference points those three points which are "least collinear". This is done by considering the distance between one reference point to the line defined by the other two reference points.

4 Structure and Motion with Infinite Homographies

We have seen that with the knowledge of H_j the relationship between known image points p^*_{ij} and unknown points $\bar{P}_i$ and camera centers $\bar{Q}_j$ is linear (see eqn. (3)). Furthermore, eqn. (3) shows that changing the image basis by a homography B and individually scaling the image points p^*_{ij} by s_{ij} does not alter this relationship:

$$p'_{ij}\ \sim s_{ij}\ B\ p^*_{ij}\ \sim\ B\bar{P}_i\ -\ B\bar{Q}_j\ \sim\ \bar{P}'_i\ -\ \bar{Q}'_j. \tag{13}$$

How to choose B and s_{ij} in an optimal way will be discussed in this section.

If B and s_{ij} are known, p'_{ij} can be derived and we obtain a set of linear equations (SLE) as in section (2):

$$L\ h = 0 \text{ with}$$
$$h = (\bar{X}'_1, \bar{Y}'_1, \bar{Z}'_1, \ldots, \bar{X}'_n, \bar{Y}'_n, \bar{Z}'_n, \bar{A}'_1, \bar{B}'_1, \bar{C}'_1, \ldots, \bar{A}'_m, \bar{B}'_m, \bar{C}'_m)^T. \quad (14)$$

Since points which are on or "close" to the reference plane potentially increase the numerical stability of the reconstruction, the projection relations of such points have to be excluded from the SLE. However, how can these points be detected automatically? One idea is to exclude successively points from the SLE which are close to the reference plane. Therefore, a ranking of all points on the basis of their *distance to the reference plane* has to be known.

4.1 Distance between Points and Reference Plane

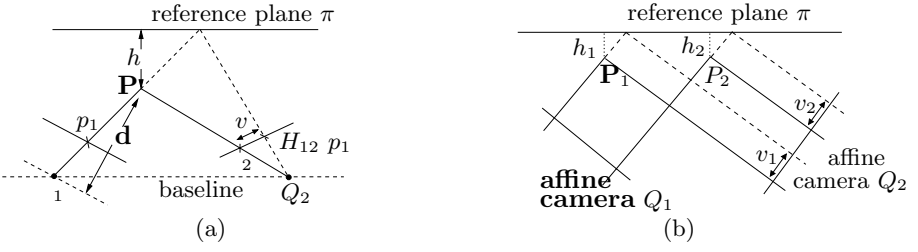

Fig. 3. Parallax geometry for (a) projective and (b) affine cameras.

Let us consider a configuration with two cameras Q_1, Q_2, a 3D point P and a reference plane π where P does not lie on π (fig. 3(a) depicts a top view). The inter-image homography from the first to the second view via the reference is as defined in the previous section: $H_{12} = H_1^{-1}\ H_2$. The residual parallax vector in the second view is given as $v = p_2 - H_{12}\ p_1$. Obviously, v is null if P lies on π. However, v vanishes as well if P lies on the baseline of the two views. Therefore, the distance of a point to the reference plane can not be determined directly from its parallax vector. Let us define $\gamma_i = \frac{h_i}{d_i}$, where h_i is the perpendicular distance of P_i to the reference plane and d_i is the depth of P_i with respect to the first view (see fig. 3(a)). It is known [8] that the relative depth $\frac{\gamma_1}{\gamma_2}$ of two points P_1 and P_2 can be derived directly from their parallax vectors v_1, v_2. This means that the relative distance $\frac{h_1}{h_2}$ of two points depends on both their parallax vectors and their depths. However, if we assume parallel projection, d_i is constant and we obtain the relative distance of two points as:

$$\frac{\gamma_1}{\gamma_2} = \frac{h_1}{h_2} = \frac{v_1}{v_2}. \quad (15)$$

Fig. 3(b) depicts a configuration with affine cameras where $h_1 = h_2$ and therefore $v_1 = v_2$. We will use eqn. (15) as an approximation for projective cameras.

The original task was to determine a unique function $dis(P_i)$ which represents the distance between a point P_i and the reference plane. Eqn. (15) supplies a distance function $dis_{j_1 j_2}(\cdot)$ between each pair of views j_1, j_2, which is unique up to scale. A unique function $dis(\cdot)$ can be obtained by recursively merging the set of functions $dis_{j_1 j_2}(\cdot)$. Finally, $dis(\cdot)$ is scaled so that the maximal distance of a point to the reference plane is equal to one, i.e. $dis(\cdot) \in [0, 1]$.

4.2 The Choice of the Image Basis

It has been shown in [3] that the normalization of image coordinates can dramatically influence the result of a computation based on image coordinates. Normalization means that the centroid of all image coordinates is at the origin and the average distance of an image point to the origin is equal to $\sqrt{2}$. If we consider eqn. (13), normalization would involve to determine for each view j an individual matrix B_j, which represents the normalization. However, such a B_j would destroy the linear relationship between points and camera centers. Therefore, the matrix B has to be determined independently of a certain view j. We define:

$$B \;=\; \frac{1}{m} \sum_{j=1}^{m} B_j \;/\; \left\| B_j \right\|_2 \,, \tag{16}$$

where $\left\| \cdot \right\|_2$ is the Frobenius norm of a matrix and m is the number of views.

4.3 Weighting the Set of Equations

Let us consider a point P_1 which is closer to the reference plane than another point P_2. Since the reference plane is the plane at infinity in the chosen projective space, the coordinates of the reconstructed point $\bar{P}_1$ are larger than the ones of $\bar{P}_2$. This means that in the presence of noise, the point with larger coordinates is reconstructed more accurately. In order to eliminate this favoring of certain points we suggest to choose[2] the scale factors in eqn. (13) as $s_{ij} = dis(P_i)$ where $dis(\cdot) \in [0, 1]$. This means that points which are closer to the reference plane are inhibited. The same applies to the equations in the SLE of such a point.

5 Outline of the Algorithms

The different ideas of the previous sections result in three algorithms: **A-alg.** (for affine cameras), **P-alg.** (for projective cameras) and **AtoP-alg.** (an iterative version for projective cameras). The **AtoP-alg.** assumes initially affine cameras and compensates iteratively for the perspective effects. However, there is no guarantee that the algorithm will converge to a global minimum. The algorithms are composed of the following steps:

[2] This particular choice of the scale factors s_{ij} is motivated by the mapping $(0, 1)^T \rightarrow (1, 0)^T$ and $(1, 1)^T \rightarrow (1, 1)^T$ in the projective space P^1.

1. Detect optimal three reference points (sec. 3.5)
2. Determine H_j (sec. 3)
   A-alg, **AtoP-alg**: assume affine cameras; **P-alg**: assume projective cameras
3. Determine distances between points and reference plane (sec. 4.1)
4. Exclude iteratively points from the SLE
   5. Determine matrix B (sec. 4.2)
   7. Determine scales s_{ij} and image points $p'_{ij} = s_{ij} \left\| B H_j^{-1} p_{ij} \right\|_2$ (sec. 4.3)
   8. Obtain $\bar{P}'_i, \bar{Q}'_j$ by SVD (eqn. (14)) and P_i on the ref. plane (eqn. (2))
9. Only **AtoP-alg.**: Update H_j (sec. 3.4) and go to 3.
   Stop if the RMS-error is either unchanged or increases
10. Take the best result on the basis of RMS-error
11. Undo the basis change: $\bar{P}_i = B^{-1}\ \bar{P}'_i$ and $\bar{Q}_j = B^{-1}\ \bar{Q}'_j$

The quality of the reconstruction is evaluated in terms of the Root-Means-Square (RMS) error. However, other criteria could be used.

6 Experiments

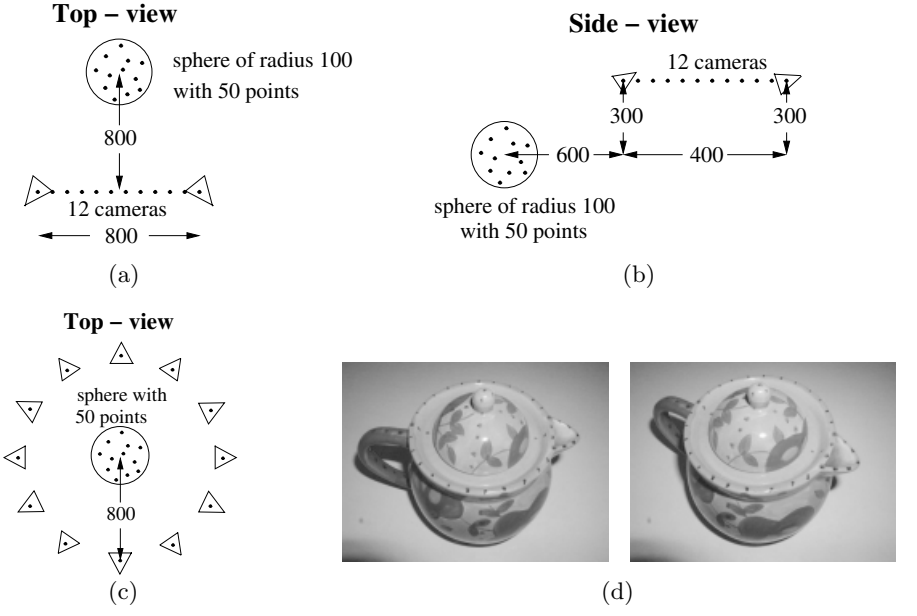

Fig. 4. The four different configurations: lateral movement – LAT (a), translational movement towards the scene – TOW (b), circular movement – CIR (c) and 2 images of a real sequence – TEA (d).

In order to demonstrate the performance of the algorithms, they were applied to a wide range of different camera motions and scene settings. Fig. 4 depicts four of them: three synthetic configurations (a-c) and a real sequence (d). Each synthetic configuration consists of 50 points distributed uniformly in a sphere

and 12 cameras pointing towards the center of the sphere. They differ in the type of camera motion: lateral – LAT (a), circular – CIR (c) and translational towards the scene – TOW (b). The distances (in units) between cameras and points are as in fig. 4. Additionally, a real sequence of 8 images (see fig. 4(d)) was utilized. The camera circled around the teapot, similar to the CIR-configuration. In order to conduct synthetic experiments on this sequence, a reconstruction of the cameras and 96 points of the teapot (see fig. 8) served as the basis for a synthetic configuration (TEA). If not stated explicitly, all the points of the TEA-configuration are visible in all views, i.e. no missing data.

6.1 Synthetic Data

The synthetic experiments were carried out with respect to different levels of Gaussian noise: $\sigma = 0, 0.2, \ldots, 3.0$ (standard deviation). In order to obtain average performance, the following two steps were conducted 20 times for each noise level: a) randomly determine 50 scene points b) add Gaussian noise on the reprojected 3D points. In case of projective cameras, the internal calibration matrix was chosen as diag(500,500,1). Affine cameras were derived from the projective cameras by moving the center of projection to infinity where the image size remained fixed (see [4]).

The computed reconstructions were evaluated in terms of the Root-Mean-Square (RMS) error between reprojected 3D points and 2D image data (potentially corrupted by noise). The performance of the three algorithms presented in this paper: **A-alg., P-alg.** and **AtoP-alg.** is compared to affine factorization of Tomasi-Kanade (**TK-alg.**) [18] and projective factorization of Sturm-Triggs (**ST-alg.**) [17,19]. In [17] it is suggested to derive the initial "projective depths" from epipolar geometry. Other authors, e.g. [6,4,5], have shown that initialising all the "projective depths" to one and reestimating them by reprojection produce good results as well. This more simple approach was used in this paper.

Different Configurations. Let us consider the performance of the A-alg. and TK-alg. for different configurations (fig. 5 (a,b)). In this case, the scene is viewed by affine cameras. The performance of the TK-alg. is equally good for all configurations and close to identical with the theoretical minimum, i.e. Cramer-Rao lower bound (not shown). The differences between the results of the TK-alg. and the A-alg. are not large but noticeable. Furthermore, the A-alg. performed worse for the TOW-configuration than for the other three configurations. Since the TOW-configuration has the shortest baseline relative to the scene (see fig. 4 (b)) this result can be expected.

Fig. 5 (c,d) shows the results of the P-alg. and ST-alg. with respect to different configurations. In contrast to the previous section projective cameras were used. As in the case of affine factorization, the ST-alg. is equally good for all configurations and close to the theoretical minimum. The difference between the results of the ST-alg. and the P-alg. are obvious. However, for practical noise levels, e.g. $\sigma = 1.0$, the results of the P-alg. are still acceptable. A comparison between the A-alg. (a,b) and P-alg. (c,d) shows that the results of the A-alg.

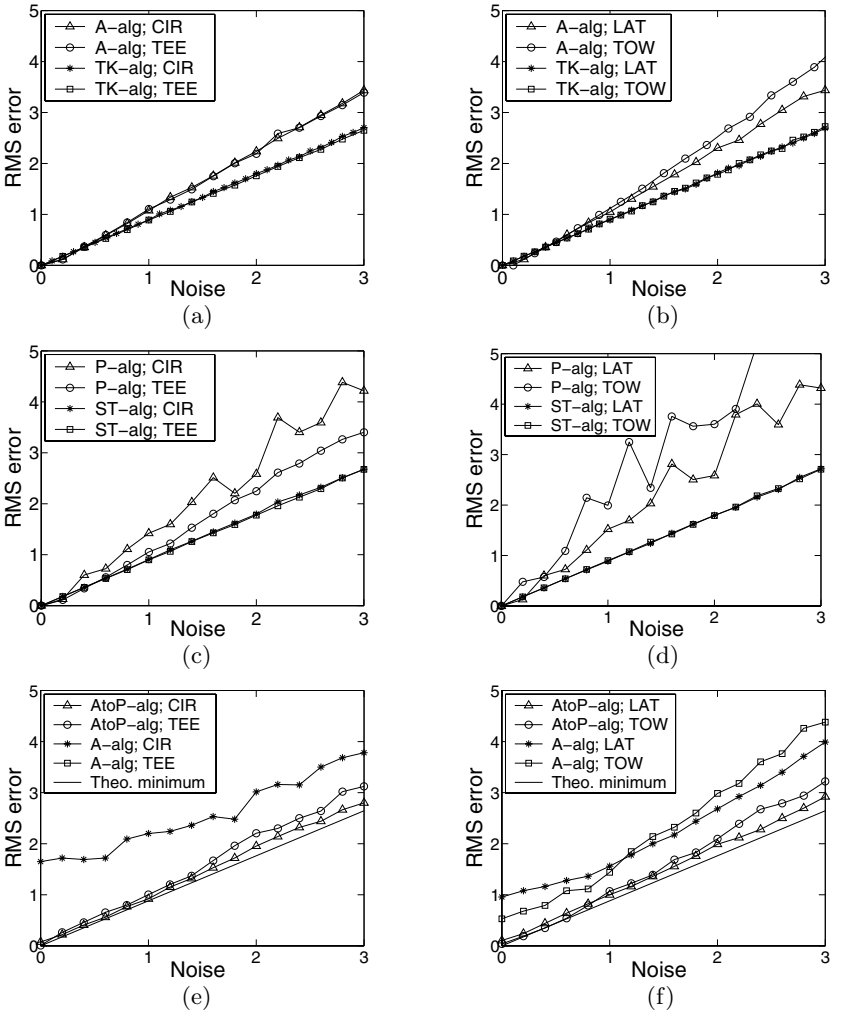

Fig. 5. Graphs in respect to different configurations.

are better and the curves for the A-alg. are more linear. The only difference between the A-alg. and P-alg. is the derivation of the infinite homographies. A more detailed analyses confirmed that this derivation in the case of projective cameras is fairly sensitive to noise in the epipols and reference points.

Finally, fig. 5 (e,f) depicts the performance of the AtoP-alg. for different configurations with projective cameras. Additionally, the results of the A-alg. are shown, which serve as the initialisation for the iterative AtoP-alg. The theoretical minimum is displayed as well. The results of the A-alg. on the TEA-configuration were off the scale (RMS-error between 18.4 and 19.1). It stands out, that for all configurations the initial reconstruction of the A-alg. can be significantly improved by the AtoP-alg. Particularly, in the case of no noise, i.e. $\sigma = 0$, the AtoP-alg. converged for all configuration close to the theoretical minimum.

However, for higher noise levels, e.g. $\sigma = 3.0$, the AtoP-alg. did not always converge close to the theoretical minimum, e.g. TOW-configuration.

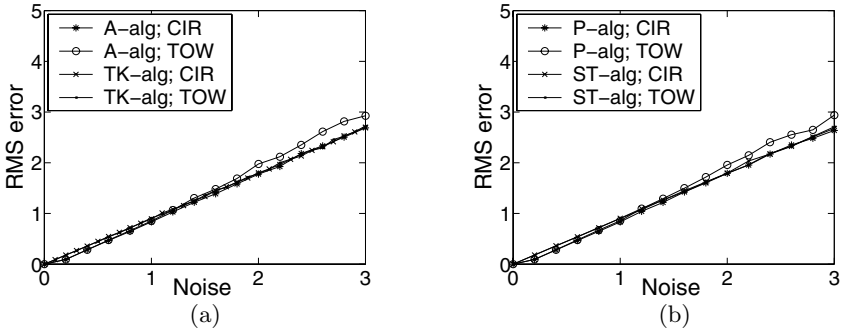

Fig. 6. Graphs for the case of perfect reference points.

Choice of Reference Points. In this section we will repeat some of the experiments of the previous section. However, Gaussian noise will be added to all image points *except for* the three reference points. Fig. 6 depicts the results for the case of affine (a) and projective cameras (b). In contrast to the previous section, the performance of the A-alg. and TK-alg. (fig. 6 (a)) is close to identical. The same applies to the performance of the P-alg. and ST-alg. (fig. 6 (b)). Further experiments on the AtoP-alg. and other configurations confirmed this performance. This leads to the conclusion that, independent of the configuration, the *choice of reference points is crucial* for the three presented algorithms.

Missing Data. In the previous experiments was assumed that all 96 points of the TEA-configuration are visible in all 8 images, i.e. no missing data. However, in practice some points might be occluded in several views. Fig. 7 (a) shows the "true" *visibility matrix* of the TEA-configuration. An element of this matrix is set (black dot) if the respective point is visible in the respective view. It turns out, that 33% of the entries are not set. If the correspondence between successive frames were obtained by tracking, the final visibility matrix might look like in fig. 7 (b). Each point, except for the three reference points, is only visible in three successive views. The amount of missing data increases to 61%.

If all points are visible in all views, the optimal reference points correspond to points on the body of the teapot (fig. 4 (d)). In the case of missing data (33% or 61%), points on the rim and handle were detected as the best reference points.

Let us consider the performance of the A-alg. (fig. 7(d)) and the P-alg. (fig. 7 (e)) on these three types of visibility matrices: no missing data, 33% missing data (fig. 7(a)) and 61% missing data (fig. 7(b)). The first observation is that the performance of the A-alg. and P-alg. differs only slightly in respect to the different cases of missing data. Further experiments confirmed the conclusion that the performance of the three novel algorithms is "fairly" independent to the amount of missing data. A more detailed analyses shows that both algorithms performed less stable for the case of 61% missing data (zigzag shape of the

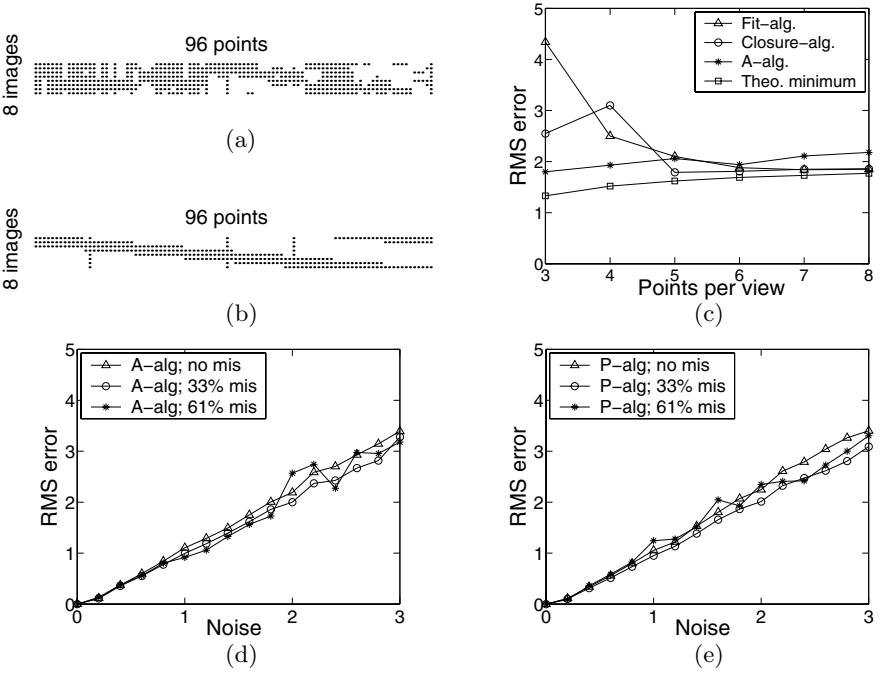

Fig. 7. Visibility matrices (a,b) and graphs (c-e) for the case of missing data.

curves). This performance can be expected since "only" three successive views (fig. 7(b)), i.e. short baseline, provide information about the 3D position of a certain point in the scene.

In the last experiment we compare the A-alg. to two alternative methods which handle missing data for affine views. Jacobs algorithm [9] fits a matrix of rank 3 to the data matrix with missing elements (**Fit-alg.**)[3]. Kahl and Heyden [10] use the centred affine tensors between successive two and three views to obtain all camera matrices simultaneously by using the so-called closure constraints (**Closure-alg.**). The main advantage of these methods is that the image data is used in a uniform manner, i.e no selection of specific reference points. In contrast to our approach, the image data is not used directly to obtain structure and motion simultaneously. Fig. 7(c) shows the result for a noise level of $\sigma = 2.0$. The visibility matrix in fig. 7(b) was used, where the number of points per view varied. If a point is visible in more than 5 views the alternative methods performed slightly better, which might be due to noise in the reference points. However, in the case of more missing data, i.e. only 3 or 4 points per view, the alternative methods performed worse in this experiment. In case of the Closure-alg. an explanation might be that the data is not sufficient to obtain good tensors.

6.2 Real Data

The 8 images of the teapot (see fig. 4 (d)) served as a real image sequence. On the basis of this, 96 corresponding image points were selected manually which

[3] We used the code available at http://www.neci.nj.nec.com/homepages/dwj/.

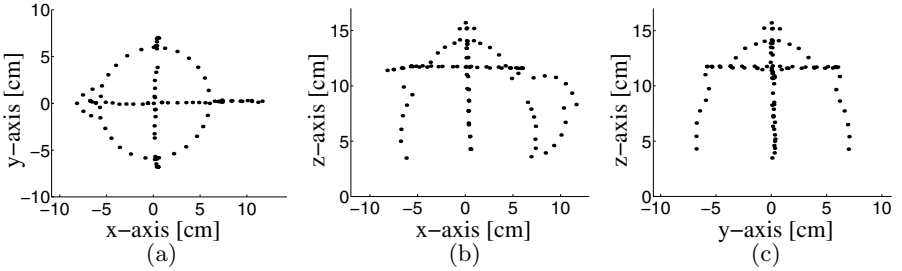

Fig. 8. Top (a), side (b) and front (c) view of the reconstructed teapot (see fig. 4 (d)).

results in the "true" visibility matrix as in fig. 7 (a). The reconstruction obtained with the P-alg. had an RMS error of 2.84 between reprojected 3D points and select image points. Fig. 8 shows the top (a) side (b) and front (c) view of the reconstruction which was metric rectified. Only those scene points which lie on the contour in the top, side or front view of the teapot were reconstructed. The AtoP-alg. performed with an RMS error of 2.06 where the initial reconstruction determined with the A-alg. had a RMS error of 15.82.

7 Summary and Conclusion

We have presented two linear methods for the simultaneous computation of 3D points and camera positions from multiple affine and projective views. This is achieved by computing the nullspace of one linear system constructed from *all image data*. In case of affine views the only requirement is that *three points are visible in all views*. Additionally, for projective views the epipolar geometry has to be known. In case of unknown epipolar geometry, a third iterative algorithm for projective views has been presented. The treatment of affine and projective cameras in a single, unified projective framework is a further, novel contribution.

The only other methods that use all image data directly are the factorization algorithms for affine [18] and projective [17,19] views. However, in contrast to our approach, they require all points to be visible in all views. Since points become inevitably occluded in practice, we consider the presented methods as a major and novel contribution to the problem of structure from motion. Alternative reconstruction methods for handle missing data have been presented [9,10,20], which have the advantage that data is used in a uniform manner, i.e no selection of reference points. However, in contrast to our approach the image data is not used directly to obtain structure and motion simultaneously.

The experiments, which covered a wide range of different camera motions and scene settings, have shown that the presented algorithms perform very well for practical noise levels. If the reference points are chosen carefully the performance of the presented algorithms compared to affine and projective factorization methods is close to identical. Furthermore, the use of all available image data, which is not available for factorization methods, is a most important numerical stabilising factor in this approach.

References

1. Faugeras, O. and Luong, Q.-T. 2001. The Geometry of Multiple Images. The MIT Press.
2. Fitzgibbon, A. W. and Zisserman, A. 1998. Automatic camera recovery for closed or open image sequences. In Europ. Conf. Comp. Vis., Freiburg, Germany, pp. 311-326.
3. Hartley, R. 1997. In defence of the 8-point algorithm. In IEEE Trans. on Pattern Anal. and Machine Intell., 19(6), pp. 580-593.
4. Hartley, R. and Zisserman, A. 2000. Multiple View Geometry in Computer Vision. Cambridge University Press.
5. Hartley, R., Dano, N. and Kaucic, R. 2001. Plane-based Projective Reconstruction. In Int. Conf. Comp. Vis., Vancouver, Canada, pp. 420-427.
6. Heyden, A., Berthilsson, R. and Sparr, G. 1999. An iterative factorization method for projective structure and motion from image sequences. In Image and Vision Computing, 17(13), pp. 981-991.
7. Heyden, A. and Kahl, F. 2000. Direct Affine Reconstruction. Int. Conf. Pattern Recog., Barcelona, Spain, pp. 885-888.
8. Irani, M. and Anandan, P. 1996. Parallax geometry of pairs of points for 3d scene analysis. In Europ. Conf. Comp. Vis., Cambridge, UK, pp. 17-30.
9. Jacobs, D. 1997. Linear Fitting with Missing Data for Structure-from-Motion. In IEEE Conf. Comp. Vis. and Pattern Recog., San Juan, Puerto Rico, pp. 206-212.
10. Kahl, F. and Heyden, A. 1999. Affine structure and motion from points, lines and conics. In Int. J. Computer Vision, 33(3):163-180.
11. Koch, R., Pollefeys, M. and VanGool, L. 1998. Multi viewpoint stereo from uncalibrated video sequences. In Europ. Conf. Comp. Vis., Freiburg, Germany, pp. 55-65.
12. Koenderink, J.J. and van Doorn, A.J. 1991. Affine structure from motion. In J. Opt. Soc. Am. A, 8(2), pp. 377-385.
13. Oliensis, J. 1995. Multiframe Structure from Motion in Perspective. In Workshop on Representations of Visual Scenes, Boston, USA, pp. 77-84.
14. Quan, L., Heyden A. and Kahl, F. 1999. Minimal Projective Reconstruction with Missing Data. In IEEE Conf. Comp. Vis. and Pattern Recog., Fort Collins, Colorado, pp. 210-216.
15. Rother, C. and Carlsson S. 2001. Linear Multi View Reconstruction and Camera Recovery. In Int. Conf. Comp. Vis., Vancouver, Canada, pp. 42-51.
16. Schaffalitzky, F., Zisserman, A., Hartley, R. I. and Torr, P.H.S. 2000. A Six Point Solution for Structure and Motion. In Europ. Conf. Comp. Vis., Dublin, Ireland, pp. 632-648.
17. Sturm, P. and Triggs, B. 1996. A factorization based algorithm for multi-image projective structure and motion. In Europ. Conf. Comp. Vis., Cambridge, U.K., pp. 709-719.
18. Tomasi, C. and Kanade, T. 1992. Shape and Motion from Image Streams under Orthography: a Factorization Method. In Int. J. Computer Vision, 9(2):137-54.
19. Triggs, B. 1996. Factorization methods for projective structure and motion. In IEEE Conf. Comp. Vis. and Pattern Recog., San Francisco, CA, pp. 845-851.
20. Triggs, B. 1997. Linear projective reconstruction from matching tensors. In Image and Vision Computing, 15(8), pp. 617-625.
21. Triggs, B. 2000. Plane + Parallax, Tensors and Factorization. In Europ. Conf. Comp. Vis., Dublin, Ireland, pp. 522-538.