5D Motion Subspaces for Planar Motions

Roland Angst and Marc Pollefeys

Computer Vision and Geometry Lab, Department of Computer Science
ETH Zürich, Universitätstrasse 6, 8092 Zürich, Switzerland
{rangst,marc.pollefeys}@inf.ethz.ch http://www.cvg.ethz.ch/

Abstract. In practice, rigid objects often move on a plane. The object then rotates around a fixed axis and translates in a plane orthogonal to this axis. For a concrete example, think of a car moving on a street. Given multiple static affine cameras which observe such a rigidly moving object and track feature points located on this object, what can be said about the resulting feature point trajectories in the camera views? Are there any useful algebraic constraints hidden in the data? Is a 3D reconstruction of the scene possible even if there are no feature point correspondences between the different cameras? And if so, how many points are sufficient? Does a closed-form solution to this shape from motion reconstruction problem exist?
This paper addresses these questions and thereby introduces the concept of 5 dimensional planar motion subspaces: the trajectory of a feature point seen by any camera is restricted to lie in a 5D subspace. The constraints provided by these motion subspaces enable a closed-form solution for the reconstruction. The solution is based on multilinear analysis, matrix and tensor factorizations. As a key insight, the paper shows that already two points are sufficient to derive a closed-form solution. Hence, even two cameras where each of them is just tracking one single point can be handled. Promising results of a real data sequence act as a proof of concept of the presented insights.

Key words: 3D reconstruction, shape from motion, matrix and tensor factorizations, feature point trajectories, affine cameras, planar rigid motion

1 Introduction and Related Work

Setting and Objective: Assume a rigid object is moving on a plane. The object is therefore rotating around a fixed axis orthogonal to this plane and translations are restricted to shifts inside that plane. Multiple stationary affine cameras observe the moving object and track feature points located on this object. Computing correspondences across a wide baseline is a difficult problem in itself and sometimes even impossible to solve (think of two cameras which point at two completely different sides of the rigid object). In our setting, each camera therefore tracks its own set of feature points. There are no feature point correspondences between the different cameras. The only available correspondence

between the cameras is the *motion correspondence*: all the cameras observe the same planar motion. This paper presents a thorough analysis of the geometric and algebraic structure contained in 2D feature point trajectories in the camera image planes. A closed-form solution for the reconstruction problem based on the motion correspondence is derived.

Motivation: The reasons why an analysis of planar motions is important are at least three-fold. Firstly, rigid planar motions are an important special case of rigid motions. Vehicles moving on the street, traffic surveillance and analysis represent prominent examples. Even data from a camera rig mounted on a moving car behaves according to the above described setting: the camera rig can be considered as stationary and the whole surrounding world as a moving rigid object. Because the car is moving on the ground plane, the motion is restricted to a planar motion. Secondly, in a fully practical system, we have to deal with missing data, i.e. lost feature tracks. It is unreasonable to assume in a practical scenario having feature tracks over a long temporal sequence. Thus in practice, we are limited to trajectories over a short period of time. However, continuous motions over a short period can often be well approximated by a rotation and translation in a plane. The third reason is theoretical curiosity. What can be gained by using an affine rather than a projective camera model? What multiple-view insights are hidden in $2D$ feature trajectories obtained under the given setting? The elegance of a theoretical exact derivation of a closed-form solution under the given assumptions should not be despised either.

Main Contributions: A thorough theoretical analysis of the important special case of planar rigid motions observed by multiple stationary affine cameras is presented. Specifically, any feature point trajectory seen by any camera is restricted to a 5 dimensional subspace which is common amongst all the cameras. A general framework for planar motions is proposed. This framework together with the theoretical insights enables a reconstruction algorithm which provides a closed-form solution as long as the total number of tracked points is larger or equal than two. Hence, the two minimal cases of one single camera tracking two points or two cameras where each of them is tracking only one point can be handled by the algorithm. No correspondences between different camera views are required. Moreover, the algorithm fuses the data of all the cameras in order to compute a robust reconstruction.

Related Work: There is a long history in computer vision about factorizations for the structure from motion problem under affine cameras. Due to lack of space, the interested reader is also referred to references contained in the mentioned related work. The initial work by Tomasi and Kanade [1] about monocular rigid factorizations initiated many variations and extensions, such as deformable [2] and articulated objects [3,4]. The concept of motion subspaces has also widely been used for feature trajectory motion segmentation [5]. Factorization based approaches with a projective camera model have been proposed in [6]. Some methods have been suggested to handle missing data in the feature trajectories due to occlusions or outliers [7,8]. The monocular structure from planar motion problem has previously attracted some interest [9,10]. However, these

approaches either resort to iterative solutions or require additional information, like e.g. the relative position of the plane of rotation w.r.t. the camera.

Extensions of the factorization approach to the case of multiple cameras observing the same scene have also been proposed, even though less numerous. Most of them [11,12] require feature point correspondences between the cameras to be known. Methods which deal with non-overlapping camera views are generally not based on factorization approaches (e.g. hand-eye-calibration [13]). However, a separate reconstruction for each camera is usually computed and thus strong assumption about the captured data are implicitly assumed. The classical factorization approach [1] has recently been extended to the multi-camera case [14]. This extensions considers the same setting, except the rigid object is assumed to move fully general in 3D space whereas we assume the object to move on a plane. This minor distinction has far reaching consequences. For example, we will see in Sec. 2 that this requires the object to rotate around at least 6 different axes of rotation, otherwise the 13 dimensional motion space is only spanned partially. The 13 dimensional factorization will thus fail miserably if applied to planar motions.

2 Rigid Planar Motions as Vectors in 5D Subspaces

This section presents how rigid planar motions can be embedded in linear subspaces. The general case of non-planar rigid motions has already been investigated [14]. In contrast to that work, where 13-dimensional subspaces were required, planar motions only ask for 5D subspaces.

Some notational conventions have to be defined first. The orthogonal projection matrix onto the column space of a matrix $\mathbf{A}$ is denoted as $\mathbb{P}_{\mathbf{A}}$. The projection matrix onto the orthogonal complement of the columns space of $\mathbf{A}$ is $\mathbb{P}_{\mathbf{A}}^{\perp} = \mathbf{I} - \mathbb{P}_{\mathbf{A}}$. A matrix whose columns span the orthogonal complement of the columns of matrix $\mathbf{A}$ is denoted as $\mathbf{A}_{\perp}$. Concatenation of multiple matrices indexed with a sub- or supscript i is represented with arrows. For example, $[\Downarrow_i \mathbf{A}_i]$ concatenates all the matrices $\mathbf{A}_i$ below each other, implicitly assuming that each of them consists of the same number of columns. The Matlab® standard indexing notation is used for the slicing operation (cutting out certain rows and columns of a matrix). Multiplication of a tensor $\mathcal{T}$ along its i-th mode with the matrix $\mathbf{A}$ is denoted as $\mathcal{T} \times_i \mathbf{A}$. The matrix which results by flattening a tensor along mode i is written as $\mathcal{T}_{(i)}$. We refer to [15] for an introductory text on multilinear algebra, tensor operations and decomposition.

The rotation around an axis $\mathbf{a}$ by an angle α can be expressed as a rotation matrix $\mathbf{R}_{\mathbf{a},\alpha} = \cos\alpha\mathbf{I}_3 + (1-\cos\alpha)\mathbf{a}\mathbf{a}^T + \sin\alpha\,[\mathbf{a}]_\times$, where $[\mathbf{a}]_\times$ denotes the skew-symmetric cross-product matrix. Rotation matrices $\mathbf{R}_{\mathbf{a},\alpha}$ around a fixed axis $\mathbf{a}$ are thus restricted to a three dimensional subspace in nine dimensional Euclidean ambient space $\mathrm{vec}(\mathbf{R}) = \left[\mathrm{vec}(\mathbf{I}_3)\ \mathrm{vec}\left(\mathbf{a}\mathbf{a}^T\right)\ \mathrm{vec}\left([\mathbf{a}]_\times\right)\right]\left(\cos\alpha\ 1-\cos\alpha\ \sin\alpha\right)^T$ where $\mathrm{vec}()$ vectorizes a matrix by stacking its columns below each other in a column vector. Let the columns of $\mathbf{V} \in \mathbb{R}^{3\times 2}$ denote an orthonormal basis for the orthogonal complement of the rotation axis $\mathbf{a}$, i.e. these columns span the plane

orthogonal to the rotation axis. A rigid motion in this plane (i.e. the rotation is around the plane normal and the translations are restricted to shifts inside the plane) is then given by

$$\begin{bmatrix} \mathbf{R}_{\mathbf{a},\alpha} & \mathbf{V}\mathbf{t} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \Leftrightarrow \begin{pmatrix} \mathrm{vec}\left(\mathbf{R}_{\mathbf{a},\alpha}\right) \\ \mathrm{vec}\left(\mathbf{V}\mathbf{t}\right) \\ 1 \end{pmatrix} = \begin{bmatrix} \mathrm{vec}\left(\mathbf{I}_3\right) & \mathrm{vec}\left(\mathbf{a}\mathbf{a}^T\right) & \mathrm{vec}\left([\mathbf{a}]_\times\right) & \mathbf{0}_{9\times 2} \\ \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 1} & \mathbf{V} \\ 1 & 1 & 0 & \mathbf{0}_{1\times 2} \end{bmatrix} \begin{pmatrix} \cos\alpha \\ 1-\cos\alpha \\ \sin\alpha \\ \mathbf{t} \end{pmatrix}, \quad (1)$$

which shows that *any* rigid motion in this plane is restricted to a five dimensional subspace of 13-dimensional (or 16 if zero-entries are not disregarded) Euclidean space. Interestingly, by noting that the space of symmetric rank-1 matrices $\mathrm{vec}\left(\mathbf{a}\mathbf{a}^T\right)$ considered as a linear space is 6 dimensional, we see that rotations around at least *six different axes of rotation* are required to span the full 13-dimensional space (the vector space of skew-symmetric matrices $[\mathbf{a}]_\times$ is 3 dimensional and thus rotations around 3 different axes already span this space).

3 Tensor Notation

Feature trajectories of points undergoing a planar rigid motion seen by different cameras can be arranged as a 3rd-order tensor. Such a representation clearly reveals the interplay between the three involved subspaces, namely the subspace of the cameras, the points, and the planar rigid motion. The structure (homogeneous coordinates of the N feature points) is given by $\mathbf{S} \in \mathbb{R}^{4\times N}$, the K affine cameras (each of them consisting of two camera axes) are described by $\mathbf{P} \in \mathbb{R}^{2K\times 4}$ and the motion over F frames will be described by the motion matrix $\mathbf{M} \in \mathbb{R}^{F\times 5}$. The projection matrix of camera k is denoted as $\mathbf{P}^k \in \mathbb{R}^{2\times 4}$, the points tracked by this camera as $\mathbf{S}^k \in \mathbb{R}^{4\times N_k}$. The combined camera matrix is thus $\mathbf{P} = [\Downarrow_k \mathbf{P}^k]$, and the combined point matrix $\mathbf{S} = [\Rightarrow_k \mathbf{S}^k]$. The axis of rotation is denoted with the unit vector $\mathbf{a}$ and the two columns of $\mathbf{V} \in \mathbb{R}^{3\times 2}$ are an orthonormal basis for the space orthogonal to the rotation axis. The image coordinate $\mathcal{W}_{[k,f,n]}$ of feature point n, at frame f, seen by camera axis k is thus

$$\mathcal{W}_{[k,f,n]} = \mathbf{P}_{[k,:]} \begin{bmatrix} \mathbf{R}_{\mathbf{a},\alpha_f} & \mathbf{V}\mathbf{t}_f \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \mathbf{S}_{[:,n]} = \mathrm{vec}\left(\begin{bmatrix} \mathbf{R}_{\mathbf{a},\alpha_f} & \mathbf{V}\mathbf{t}_f \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \right)^T \left[\mathbf{S}_{[:,n]}^T \otimes \mathbf{P}_{[k,:]}\right]^T, \quad (2)$$

where the Kronecker product property $\mathrm{vec}\left(\mathbf{A}\mathbf{X}\mathbf{B}\right) = \left[\mathbf{B}^T \otimes \mathbf{A}\right] \mathrm{vec}\left(\mathbf{X}\right)$ has been used in the second step. The values $\mathcal{W}_{[k,f,n]}$ are interpreted as a third order tensor. In contrast to [14], planar rigid motions are restricted to a five rather than a 13-dimensional space (as we have seen in Sec. 2). Thus, the core tensor $\mathcal{C} \in \mathbb{R}^{5\times 4\times 4}$, which captures the interactions between the three subspaces, becomes in its flattened representation along the temporal mode

$$\mathcal{C}_{(f)} = \begin{bmatrix} \mathrm{vec}\left(\mathbf{I}_3\right)^T & \mathbf{0}_{1\times 3} & 1 \\ \mathrm{vec}\left(\mathbf{a}\mathbf{a}^T\right)^T & \mathbf{0}_{1\times 3} & 1 \\ \mathrm{vec}\left([\mathbf{a}]_\times\right)^T & \mathbf{0}_{1\times 3} & 0 \\ \mathbf{0}_{2\times 9} & \mathbf{V}^T & \mathbf{0}_{2\times 1} \end{bmatrix} \begin{bmatrix} \mathbf{I}_3 \otimes \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times 1} \end{bmatrix} & \mathbf{0}_{9\times 4} \\ \mathbf{0}_{4\times 12} & \mathbf{I}_4 \end{bmatrix} \in \mathbb{R}^{5\times 16} \quad (3)$$

and the data tensor is described as a Tucker tensor [15] decomposition[1] $\mathcal{W} = \mathcal{C} \times_k \mathbf{P} \times_f \mathbf{M} \times_n \mathbf{S}^T \in \mathbb{R}^{F \times 2K \times N}$. These equations can be derived by arranging the values of Eq. (2) in matrix form $\mathbf{W} = \left[\Downarrow_f \Rightarrow_{k,n} \mathcal{W}_{[f,k,n]}\right]$, plugging in Eq. (1) for the planar rigid motions, using Eq. (3) to properly combine the rigid motion matrix with the Kronecker product of the points and camera matrices, and defining the motion matrix as

$$\mathbf{M} = [\Downarrow_f (\cos\alpha_f, (1 - \cos\alpha_f), \sin\alpha_f, \mathbf{t}_f^T)]. \tag{4}$$

The resulting matrix is exactly the same as the data tensor flattened along the temporal mode $\mathbf{W} = \mathcal{W}_{(f)} = \mathbf{M}\mathcal{C}_{(f)} \left[\mathbf{S} \otimes \mathbf{P}^T\right]$. The interested reader is referred to related work [14,15] for more details on tensorial representations.

4 Ambiguities

Let $\mathbf{Q}_P = \begin{bmatrix} \mathbf{R}_P & \mathbf{t}_P \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}$ and $\mathbf{Q}_S = \begin{bmatrix} \mathbf{R}_S & \mathbf{t}_S \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}$ denote two affine transformations of the global camera reference frame and the global point reference frame, respectively. The factorization is obviously ambiguous

$$\mathcal{W}_{[k,f,n]} = \mathbf{P}_{[k,:]}\mathbf{Q}_P^{-1}\mathbf{Q}_P \begin{bmatrix} \mathbf{R}_{\mathbf{a},\alpha_f} & \mathbf{V}\mathbf{t}_f \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \mathbf{Q}_S\mathbf{Q}_S^{-1}\mathbf{S}_{[:,n]}. \tag{5}$$

In tensor notation, this equation looks like

$$\mathcal{W} = \left(\mathcal{C} \times_k \mathbf{Q}_P \times_f \mathbf{Q}_M \times_n \mathbf{Q}_S^T\right) \times_k \mathbf{P}\mathbf{Q}_P^{-1} \times_f \mathbf{M}\mathbf{Q}_M^{-1} \times_n \left(\mathbf{S}^T\mathbf{Q}_S^{-T}\right), \tag{6}$$

where transformations $\mathbf{Q}_P$ and $\mathbf{Q}_S$ which are restricted to similarity transformations inside the plane of motion can be compensated by a corresponding transformation $\mathbf{Q}_M$ of the reference frame of the motion. In mathematical terms, the overconstrained system $\mathcal{C} \times_k \mathbf{Q}_P \times_f \mathbf{Q}_M \times_n \mathbf{Q}_S^T = \mathcal{C}$ can be solved exactly for $\mathbf{Q}_M$, i.e. $\mathbf{Q}_M = \mathcal{C}_{(f)} \left[\mathbf{Q}_S^{-1} \otimes \mathbf{Q}_P^{-T}\right] \mathcal{C}^*_{(f)}$ where $\mathbf{A}^*$ denotes the Moore-Penrose pseudo-inverse. Since the first three columns of $\mathbf{M}\mathbf{Q}_M^{-1}$ should still lead to proper rotations, the scaling factor of the similarity transformations of the cameras and points must cancel each other. A reconstruction inside the plane of rotation is thus unique up to two similarity transformations with reciprocal scaling (one for the cameras and one for the points). Similarity transformations with reciprocal scalings seem to be the only transformations which allow a solution to $\mathcal{C} \times_k \mathbf{Q}_P \times_f \mathbf{Q}_M \times_n \mathbf{Q}_S^T = \mathcal{C}$. This fact will be important later on in our algorithm: Given a reconstruction inside the plane of rotation with proper algebraic structure, we are guaranteed that such a reconstruction is unique up to a similarity transformation.

Transformations of the points or cameras outside the plane of rotation can not be compensated by a transformation of the motion. A out-of-plane transformation of the cameras has to be compensated directly by a suitable transformation

[1] $\times_k$, $\times_f$, and $\times_n$ indicate the mode-i product along the mode corresponding to the camera matrix, the motion matrix, and the point matrix, respectively.

of the points. Let $\mathbf{Z}_{\mathbf{a},\lambda} = \begin{bmatrix}\mathbf{V}\ \mathbf{a}\end{bmatrix} \mathrm{diag}\left(\mathbf{I}_2, \lambda\right) \begin{bmatrix}\mathbf{V}\ \mathbf{a}\end{bmatrix}^T$ be a scaling along the rotation axis, $\mathbf{R}$ an arbitrary rotation matrix, and $\mathbf{t}_{\|} = \mathbf{a}\beta$ a translation along the rotation axis. With the camera and point transformations

$$\mathbf{Q}_P = \begin{bmatrix}\mathbf{R}\mathbf{Z}_{\mathbf{a},\lambda} & -\mathbf{R}\mathbf{Z}_{\mathbf{a},\lambda}\mathbf{t}_{\|} \\ \mathbf{0}_{1\times 3} & 1\end{bmatrix} \quad \text{and} \quad \mathbf{Q}_S = \begin{bmatrix}\mathbf{Z}_{\mathbf{a},\lambda}^{-1}\mathbf{R}^T & \mathbf{t}_{\|} \\ \mathbf{0}_{1\times 3} & 1\end{bmatrix} \tag{7}$$

it can be shown that $\mathcal{C}_{\mathbf{a},\mathbf{V}} \times_k \mathbf{Q}_P \times_n \mathbf{Q}^T = \mathcal{C}_{\mathbf{Ra},\mathbf{RV}}$ where $\mathcal{C}_{\mathbf{a},\mathbf{V}}$ denotes the core tensor with rotation axis $\mathbf{a}$ and orthogonal complement $\mathbf{V}$. Note that neither the scaling nor the translation along the rotation axis influences the core tensor or the motion matrix. Hence, there is a scaling and translation ambiguity along the axis of rotation.

In the problem we are targeting, there are no point correspondences between different cameras. In this situation there is a *per camera* scale and translation ambiguity along the rotation axis. There is still only one global out-of-plane rotation ambiguity: the transformation of the rotation plane is still linked to the other cameras through the commonly observed planar motion, even in the presence of missing correspondences. Fortunately, as we will see later, the scale ambiguity along the rotation axis can be resolved by using orthogonality and equality of norm constraints on the camera axes. The translation ambiguities along the rotation axis however can not be resolved without correspondences between different camera views. Nevertheless, by registering the centroids of the points observed by each camera to the same height along the rotation axis, a solution close to the ground truth can usually be recovered.

5 Closed-Form Solution

In contrast to a rank-13 motion subspace, one camera is sufficient in order to span the complete 5 dimensional motion subspace of a planar motion. This leads to the following idea: Intuitively, a separate reconstruction can be made for each camera. These separate reconstructions are unique up to the ambiguities mentioned previously. This especially means that the reconstruction of each camera restricted to (or projected onto) the plane of rotation is a *valid* similarity reconstruction, i.e. the individual reconstructions are expressed in varying coordinate reference frames which, however, only differ from each other by similarity transformations. Using knowledge from the 5D-motion subspace, these reconstructions can then be aligned in a consistent world reference frame. If the additional assumption is made that the two camera axes of each camera are orthogonal and have equal norm (the norm can vary between different cameras) then the coordinate frame of the reconstruction can be upgraded to a similarity frame in all three dimensions. We thus end up with a consistent 3D-reconstruction.

There is a major drawback of the above algorithmic sketch. The fact that all the cameras observe the very same rigid motion is only used in the final step to align all the individual reconstructions. It is a desirable property that the information from all the cameras should be fused right at the first stage of the

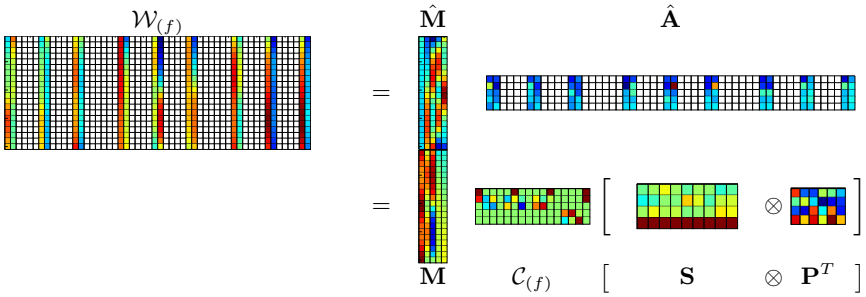

Fig. 1. Visual representation of the rank-5 factorization. Missing data entries due to missing correspondences between different cameras are depicted transparently.

algorithm in order to get a more robust reconstruction. Furthermore, in order to compute the initial reconstruction of a camera, this camera needs to track at least two points. If the camera tracks only one feature point, a reconstruction based solely on this camera is *not* possible: at least two points are necessary to span the 5D-motion subspace. The algorithm which is presented in the upcoming sections on the other hand does not suffer from these shortcomings. The algorithm fuses the information from all the cameras right at the first stage and works even when each camera tracks only one single point. Last but not least, the algorithm provides a closed-form solution.

5.1 Rank-5 Factorization

In a similar spirit to [14], we can fuse the data from all the cameras in order to compute a consistent estimate of the motion matrix. The data tensor $\mathcal{W}^k \in \mathbb{R}^{F\times 2\times N_k}$ of each camera is flattened along the temporal mode and the resulting matrices $\mathbf{W}^k = \mathcal{W}^k_{(f)} = \mathbf{M}\mathcal{C}_{(f)}\mathbf{S}^k \otimes \mathbf{P}^{k^T}$ are concatenated column-wise in a combined data matrix $\mathbf{W} = [\Rightarrow_k \mathbf{W}^k]$. A rank-5 factorization (e.g. with singular value decomposition) of this combined data matrix reveals the correct column span $\text{span}\,(\mathbf{M}) = \text{span}\left(\hat{\mathbf{M}}\right)$ of the motion matrix

$$\mathbf{W} = \hat{\mathbf{M}}\hat{\mathbf{A}} = \underbrace{\left[\Downarrow_f \cos\alpha_f \;\; 1-\cos\alpha_f \;\; \sin\alpha_f \;\; t_{f,1} \;\; t_{f,2}\right]}_{=\hat{\mathbf{M}}\mathbf{Q}} \underbrace{\mathcal{C}_{(f)} \left[\Rightarrow_k \mathbf{S}^k \otimes \mathbf{P}^{k^T}\right]}_{=\mathbf{Q}^{-1}\hat{\mathbf{A}}}, \quad (8)$$

where we have introduced the corrective transformation $\mathbf{Q} \in \mathbb{R}^{5\times 5}$ in order to establish the correct algebraic structure. This factorization separates the temporally varying component (the motion) from temporally static component (the points and the cameras). The factorization is possible since all the cameras share the same temporally varying component as all of them observe the same rigid motion. If all the cameras only track two points in total, the combined data matrix $\mathbf{W}$ will then only consist of four columns and thus a rank-5 factorization is obviously impossible. Luckily, we know that the first two columns of the motion

matrix in Eq. (4) should sum to the constant one vector. Hence, only a rank 4 factorization of the data matrix $\mathbf{W}$ is performed, the resulting motion matrix is augmented with the constant one vector $\hat{\mathbf{M}} \leftarrow [\hat{\mathbf{M}}, \mathbf{1}_{F\times 1}]$ and the second factor is adapted correspondingly $\hat{\mathbf{A}} \leftarrow [\hat{\mathbf{A}}^T, \mathbf{0}_{2N\times 1}]^T$. The rest of the algorithm remains the same.

The corrective transformation is computed in a piecewise (or stratified) way. Specifically, the corrective transformation is split into three separate transformations $\mathbf{Q} = \mathbf{Q}_{trig}\mathbf{Q}_{orient}^{-1}\mathbf{Q}_{transl}^{-1}$ where the transformation $\mathbf{Q}_{trig}$ establishes the correct trigonometric structure on the first three columns of the motion matrix, $\mathbf{Q}_{orient}$ aligns the orientations of the cameras in a consistent similarity reference frame, and $\mathbf{Q}_{transl}$ is related to correctly translate the reconstruction. The individual steps are described in detail in the next sections.

5.2 Trigonometric Structure

The first three columns of $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4, \mathbf{q}_5]$ can be solved for in the following way: since $\hat{\mathbf{M}}_{[f,:]}\mathbf{q}_i\mathbf{q}_i^T\hat{\mathbf{M}}_{[f,:]}^T = \mathbf{M}_{[f,i]}{}^2$ we have

$$\hat{\mathbf{M}}_{[f,:]}((\mathbf{q}_1+\mathbf{q}_2)(\mathbf{q}_1+\mathbf{q}_2)^T)\hat{\mathbf{M}}_{[f,:]}^T = (\cos\alpha_f + (1-\cos\alpha_f))^2 = 1 \tag{9}$$

$$\hat{\mathbf{M}}_{[f,:]}(\mathbf{q}_1\mathbf{q}_1^T + \mathbf{q}_3\mathbf{q}_3^T)\hat{\mathbf{M}}_{[f,:]}^T = \cos^2\alpha_f + \sin^2\alpha_f = 1. \tag{10}$$

These observations lead to F constraints on symmetric rank-2 matrix $\mathbf{q}_1\mathbf{q}_1^T + \mathbf{q}_3\mathbf{q}_3^T$, symmetric rank-1 matrix $(\mathbf{q}_1+\mathbf{q}_2)(\mathbf{q}_1+\mathbf{q}_2)^T$, or symmetric rank-3 matrix $b(\mathbf{q}_1\mathbf{q}_1^T + \mathbf{q}_3\mathbf{q}_3^T) + (1-b)(\mathbf{q}_1+\mathbf{q}_2)(\mathbf{q}_1+\mathbf{q}_2)^T$ with $b \in \mathbb{R}$:

$$1 = \hat{\mathbf{M}}_{[f,:]}((\mathbf{q}_1+\mathbf{q}_2)(\mathbf{q}_1+\mathbf{q}_2)^T)\hat{\mathbf{M}}_{[f,:]}^T = \hat{\mathbf{M}}_{[f,:]}(\mathbf{q}_1\mathbf{q}_1^T + \mathbf{q}_3\mathbf{q}_3^T)\hat{\mathbf{M}}_{[f,:]}^T \tag{11}$$

$$= \hat{\mathbf{M}}_{[f,:]}(b(\mathbf{q}_1\mathbf{q}_1^T + \mathbf{q}_3\mathbf{q}_3^T) + (1-b)(\mathbf{q}_1\mathbf{q}_1^T + \mathbf{q}_2\mathbf{q}_2^T))\hat{\mathbf{M}}_{[f,:]}^T \tag{12}$$

These F equations are linear in the unknown symmetric matrices and result in a one dimensional solution space (since there is a valid solution for any $b \in \mathbb{R}$). [16] shows how to extract the solution vectors $\mathbf{q}_1$, $\mathbf{q}_2$, and $\mathbf{q}_3$ from this one dimensional solution space. Once this is done, the corrective transformation $\mathbf{Q}_{trig} = \begin{bmatrix}\mathbf{q}_1 & \mathbf{q}_2 & \mathbf{q}_3 & \begin{bmatrix}\mathbf{q}_1 & \mathbf{q}_2 & \mathbf{q}_3\end{bmatrix}_\perp\end{bmatrix}$ is applied to the first factor $\hat{\mathbf{M}}\mathbf{Q}_{trig}$ which establishes the correct trigonometric structure in the first three columns. The inverse of this transformation is applied to the second factor $\tilde{\mathbf{A}} = \mathbf{Q}_{trig}^{-1}\hat{\mathbf{A}}$. Note that the structure of the first three columns of the motion matrix should not get modified anymore and hence any further corrective transformation must have upper block-diagonal structure with an identity matrix of dimension 3 in the upper left corner. The inverse of such an upper block-diagonal matrix has exactly the same non-zero pattern, i.e.

$$\mathbf{Q}_{transl}\mathbf{Q}_{orient} = \begin{bmatrix}\mathbf{I}_3 & \mathbf{Q}_{3\times 2}\\ \mathbf{0}_{2\times 3} & \mathbf{I}_2\end{bmatrix}\begin{bmatrix}\mathbf{I}_3 & \mathbf{0}_{3\times 2}\\ \mathbf{0}_{2\times 3} & \mathbf{Q}_{2\times 2}\end{bmatrix} = \begin{bmatrix}\mathbf{I}_3 & \mathbf{Q}_{3\times 2}\\ \mathbf{0}_{2\times 3} & \mathbf{Q}_{2\times 2}\end{bmatrix}. \tag{13}$$

5.3 Euclidean Camera Reference Frame

No more information can be extracted from the motion matrix and thus, we turn our attention to the second factor $\tilde{\mathbf{A}}$ which after applying a proper transformation should have the following algebraic form

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{Q}_{3\times 2} \\ \mathbf{0}_{2\times 3} & \mathbf{Q}_{2\times 2} \end{bmatrix} \tilde{\mathbf{A}} = \mathcal{C}_{(f)} \left[\Rightarrow_k \mathbf{S}^k \otimes {\mathbf{P}^k}^T \right]. \tag{14}$$

This is a particularly tricky instance of a bilinear system of equations in $\mathbf{Q}_{3\times 2}$, $\mathbf{Q}_{2\times 2}$, $\mathbf{S}^k$, and $\mathbf{P}^k$. Based on our experiences, even algebraic computer software does not succeed in finding a closed-form solution. Nevertheless, we succeeded in deriving manually a solution using geometric intuition and reasoning.

Projection onto Plane of Rotation Eq. (14) together with the known matrix $\mathcal{C}_{(f)}$ in Eq. (3) tells that $\tilde{\mathbf{A}}_{[4:5,:]} = \left[\Rightarrow_k \mathbf{1}_{1\times N_k} \otimes \left(\mathbf{P}^k_{[:,1:3]} \mathbf{V} \mathbf{Q}^{-T}_{2\times 2} \right)^T \right]$, which means that the columns of $\tilde{\mathbf{A}}_{[4:5,:]}$ contain the coordinates (w.r.t. the basis $\mathbf{V}$) of the projection of the rows of the camera matrices onto the plane of rotation. These coordinates however have been distorted with a common, but unknown transformation $\mathbf{Q}_{2\times 2}$. This observation motivates the fact to restrict the reconstruction first to the plane of rotation. Such a step requires a projection of the available data onto the plane of rotation. [16] shows that this can be done by subtracting the second from the first row and keeping the third row of Eq. (14)

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tilde{\mathbf{A}}_{[1:3,:]} + \underbrace{\begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{Q}_{3\times 2}}_{=\mathbf{T}_{2\times 2}} \tilde{\mathbf{A}}_{[4:5,:]} \tag{15}$$

$$= \begin{bmatrix} \mathrm{vec}\left(\mathbb{P}_{\mathbf{V}}\right)^T \\ \mathrm{vec}\left([\mathbf{a}]_\times\right)^T \end{bmatrix} \left[\Rightarrow_k \left(\mathbb{P}_{\mathbf{V}} \mathbf{S}^k_{[1:3,:]} \right) \otimes \left(\mathbb{P}_{\mathbf{V}} {\mathbf{P}^k_{[:,1:3]}}^T \right) \right] \tag{16}$$

$$= \begin{bmatrix} \mathrm{vec}\left(\mathbb{P}_{\mathbf{V}}\right)^T \\ \mathrm{vec}\left([\mathbf{a}]_\times\right)^T \end{bmatrix} \left[\Rightarrow_k \left(\mathbb{P}_{\mathbf{V}} \mathbf{S}^k_{[1:3,:]} \right) \otimes \left(\mathbf{V}\mathbf{Q}_{2\times 2} \right) \left(\mathbf{Q}^{-1}_{2\times 2} \mathbf{V}^T {\mathbf{P}^k_{[:,1:3]}}^T \right) \right]. \tag{17}$$

In the last step we have used $\mathbb{P}_{\mathbf{V}} = \mathbf{V}\mathbf{Q}_{2\times 2}\mathbf{Q}^{-1}_{2\times 2}\mathbf{V}^T$ and the parenthesis in the last term should stress out that for for all the cameras the term $\mathbf{Q}^{-1}_{2\times 2}\mathbf{V}^T {\mathbf{P}^k_{[:,1:3]}}^T$ can be read off from $\tilde{\mathbf{A}}_{[4:5,:]}$. The unknowns of this bilinear equation are the points and the 2-by-2 transformations $\mathbf{T}_{2\times 2}$ and $\mathbf{Q}_{2\times 2}$.

Per-Camera Reconstruction in the Plane of Rotation Eq. (17) describes a reconstruction problem in a plane which is still bilinear. As with any rigid reconstruction, there are several gauge freedoms. Specifically, the origin and the orientation of the reference frame can be chosen arbitrarily[2]. In the planar case,

[2] The first three columns of the motion matrix have already been fixed and the translation of the cameras has been lost by the projection step. Thus, there is only one planar similarity transformation left from the two mentioned in Sec. 4.

this means a 2D offset and the orientation of one 2D vector can be chosen freely. In the following we will make use of the gauge freedoms in order to render this bilinear problem in multiple sequential linear problems. The reconstruction procedure described in the upcoming paragraphs could be applied to one single camera. This would provide $\mathbf{T}_{2\times2}$ and $\mathbf{Q}_{2\times2}$ which could then be used to solve for the points in the remaining cameras. However, increased robustness can be achieved by solving the sequential linear problems for each camera separately and aligning the results in a final step in a consistent coordinate frame. For each camera, the gauge freedoms will be fixed in a different way which enables the computation of a reconstruction for each camera. The reference frames of the reconstructions then differ only by similarity transformations. This fact will be used in the next section in order to register all the reconstructions in a globally consistent reference frame.

In single camera rigid factorizations, the translational gauge freedoms are usually chosen such that the centroid of the points matches the origin of the coordinate system, i.e. $\frac{1}{N}\mathbf{S}\mathbf{1}_{N\times1} = \mathbf{0}$. We will make the same choice $\frac{1}{N_k}\mathbf{S}^k\mathbf{1}_{N_k\times1} = \mathbf{0}$ on a per-camera basis. Let $\tilde{\mathbf{A}}^k$ denote the columns of $\tilde{\mathbf{A}}$ corresponding to camera k. By closer inspection of Eq. (17) and with the Kronecker product property $[\mathbf{AB}] \otimes [\mathbf{CD}] = [\mathbf{A} \otimes \mathbf{C}]\,[\mathbf{B} \otimes \mathbf{D}]$ we get

$$\begin{aligned}&\left[\begin{bmatrix}1 & -1 & 0\\ 0 & 0 & 1\end{bmatrix}\tilde{\mathbf{A}}^k_{[1:3,:]} + \mathbf{T}_{2\times2}\tilde{\mathbf{A}}^k_{[4:5,:]}\right]\left[\frac{1}{N_k}\mathbf{1}_{N_k\times1}\otimes\mathbf{I}_2\right]\\ =&\begin{bmatrix}\mathrm{vec}\left(\mathbb{P}_\mathbf{V}\right)^T\\ \mathrm{vec}\left([\mathbf{a}]_\times\right)^T\end{bmatrix}\left(\mathbb{P}_\mathbf{V}\mathbf{S}^k_{[1:3,:]}\frac{1}{N_k}\mathbf{1}_{N_k\times1}\right)\otimes\left(\mathbb{P}_\mathbf{V}\mathbf{P}^k_{[:,1:3]}{}^T\right) = \mathbf{0}_{2\times2}.\end{aligned} \tag{18}$$

The last equation followed since the centroid has been chosen as the origin. The above linear system consists of four linearly independent equations which can readily be solved for the four unknowns in $\mathbf{T}_{2\times2}$.

The remaining two gauge freedoms are due to the arbitrary choice of the orientation of the coordinate frame inside the plane of rotation. These gauge freedoms can be chosen s.t. the first row $\begin{pmatrix}1 & 0\end{pmatrix}\mathbf{P}^k_{[:,1:3]}\mathbf{V}$ of the k^{th} camera matrix equals the known row $\begin{pmatrix}1 & 0\end{pmatrix}\mathbf{P}^k_{[:,1:3]}\mathbf{V}\mathbf{Q}^{-T}_{2\times2}$. Such a choice poses two constraints on $\mathbf{Q}_{2\times2}$

$$\begin{pmatrix}1 & 0\end{pmatrix}\mathbf{P}^k_{[:,1:3]}\mathbf{V} = \begin{pmatrix}1 & 0\end{pmatrix}\left(\mathbf{P}^k_{[:,1:3]}\mathbf{V}\mathbf{Q}^{-T}_{2\times2}\right) = \begin{pmatrix}1 & 0\end{pmatrix}\left(\mathbf{P}^k_{[:,1:3]}\mathbf{V}\mathbf{Q}^{-T}_{2\times2}\right)\mathbf{Q}^T_{2\times2}. \tag{19}$$

Knowing $\mathbf{T}_{2\times2}$ as well as the first row of $\mathbf{P}^k_{[:,1:3]}\mathbf{V}$ implies that the remaining unknowns in every second column of $\tilde{\mathbf{A}}^k$ (i.e. the columns which depend on the first row) are only the points. This results in $2N_k$ linear equations in the $2N_k$ unknowns of the projected point coordinates $\mathbb{P}_\mathbf{V}\mathbf{S}^k_{[1:3,:]}$. After solving this system, only the entries of $\mathbf{Q}_{2\times2}$ are not yet known. The two linear constraints of Eq. (19) enable a reparameterization with only two parameters $\mathbf{Q}_{2\times2} = \mathbf{Q}_0 + \lambda_1\mathbf{Q}_1 + \lambda_2\mathbf{Q}_2$. Inserting this parameterization into Eq. (17) and considering only every other second column (i.e. the columns corresponding to the second row of

the camera) leads to a linear system in λ_1 and λ_2 with $2N_k$ linear equations. The linear least squares solution provides the values for λ_1 and λ_2.

The above procedure works fine as long as every camera tracks at least two points. Otherwise the computation of λ_1 and λ_2 in the final step will fail because of our choice to set the mean to the origin. The coordinates of the single point are then equal to the zero vector and hence, this single point does not provide any constraints on the two unknowns. In order to avoid this problem we use the following trick: instead of choosing the origin as the mean of the points which are tracked by the camera currently under investigation, the origin is rather fixed at the mean of the points of *another* camera. Such a choice is perfectly fine as the origin can be chosen arbitrarily. The computation of $\mathbf{T}_{2\times 2}$ for camera k is therefore based on the data of another camera $k' \neq k$. This clever trick allows to compute a reconstruction even for cameras which only track one single point.

Registration in a Common Frame Inside the Plane of Motion After the previous per-camera reconstruction a camera matrix is known for each camera. Let $\tilde{\mathbf{P}}^k$ denotes its first three columns whose projection onto the plane of rotation is correct up to a registration with a 2-by-2 scaled rotation matrix $\lambda_k \mathbf{R}_k$. On the other hand, we also know the projections $\mathbf{P}^k_{[:,1:3]}\mathbf{V}\mathbf{Q}^{-T}_{2\times 2}$ of the camera matrices onto the plane of rotation up to an unknown distortion transformation $\mathbf{Q}_{2\times 2}$ which is the same for all the cameras. This implies $\tilde{\mathbf{P}}^k\mathbf{V}\mathbf{R}_k\lambda_k = \mathbf{P}^k_{[:,1:3]}\mathbf{V}$ and thus

$$\tilde{\mathbf{P}}^k\mathbf{V}\mathbf{V}^T\tilde{\mathbf{P}}^{k,T}\lambda_k^2 = \left(\mathbf{P}^k_{[:,1:3]}\mathbf{V}\mathbf{Q}^{-T}_{2\times 2}\right)\mathbf{Q}^T_{2\times 2}\mathbf{Q}_{2\times 2}\left(\mathbf{Q}^{-1}_{2\times 2}\mathbf{V}^T{\mathbf{P}^k_{[:,1:3]}}^T\right). \tag{20}$$

This is a linear system in the three unknowns of symmetric $\mathbf{Q}^T_{2\times 2}\mathbf{Q}_{2\times 2}$ and K scale factors λ_k^2 which is again solved in the least squares sense. Doing so provides a least squares estimate of the three unknowns of $\mathbf{Q}^T_{2\times 2}\mathbf{Q}_{2\times 2}$. An eigenvalue decomposition $\mathbf{E}\mathbf{\Lambda}\mathbf{E}^T = \mathbf{Q}^T_{2\times 2}\mathbf{Q}_{2\times 2}$ provides a mean to recover $\mathbf{Q}_{2\times 2} = \mathbf{E}^T\mathbf{\Lambda}^{\frac{1}{2}}$ which allows to express the projections of the camera matrices $\mathbf{P}^k_{[:,1:3]}\mathbb{P}_{\mathbf{V}} = \left(\mathbf{P}^k_{[:,1:3]}\mathbf{V}\mathbf{Q}^{-T}_{2\times 2}\right)\mathbf{Q}^T_{2\times 2}\mathbf{V}^T$ onto the plane in one single similarity frame.

Orthogonality and Equality of Norm Constraints As has been previously mentioned, the correct scaling along the rotation axis can only be recovered by using additional constraints, like the orthogonality and equal norm constraints on the two camera axes of a camera. These constraints will be used in the following to compute the remaining projection of the camera matrix onto the axis of rotation. Due to $\mathbf{P}^k_{[:,1:3]} = \mathbf{P}^k_{[:,1:3]}(\mathbb{P}_{\mathbf{V}} + \mathbb{P}_{\mathbf{a}})$ and $\mathbb{P}_{\mathbf{V}}\mathbb{P}_{\mathbf{a}} = \mathbf{0}$ we get $\lambda_k^2\mathbf{I}_2 = \mathbf{P}^k_{[:,1:3]}{\mathbf{P}^k}^T_{[:,1:3]} = \mathbf{P}^k_{[:,1:3]}\mathbb{P}_{\mathbf{V}}{\mathbf{P}^k_{[:,1:3]}}^T + \mathbf{P}^k_{[:,1:3]}\mathbb{P}_{\mathbf{a}}{\mathbf{P}^k_{[:,1:3]}}^T$.

Thanks to the previous registration step, the projections $\mathbf{P}^k_{[:,1:3]}\mathbb{P}_{\mathbf{V}}$ are known for all cameras. As $\mathbf{P}^k_{[:,1:3]}\mathbb{P}_{\mathbf{a}}{\mathbf{P}^k_{[:,1:3]}}^T = \mathbf{P}^k_{[:,1:3]}\mathbf{a}\mathbf{a}^T{\mathbf{P}^k_{[:,1:3]}}^T$ and replacing $\mathbf{P}^k_{[:,1:3]}\mathbf{a}$ by $\mathbf{w}^k$, the unknowns of the above equation become λ_k and the two components of the vector $\mathbf{w}^k$. This results in K independent 2^{nd}-order polynomial system

of equations with 3 independent equations in the three unknowns $\mathbf{w}^k$ and λ_k. Straight-forward algebraic manipulation will reveal the closed-form solution to this system (see [16] for details). Once $\mathbf{w}^k$ is recovered, the camera matrix is given by solving the linear system $\mathbf{P}^k_{[:,1:3]}\left[\mathbb{P}_{\mathbf{V}}, \mathbf{a}\right] = \left[\mathbf{P}^k_{[:,1:3]}\mathbb{P}_{\mathbf{V}}, \mathbf{w}^k\right]$. The solution of the polynomial equation is unique up to the sign. This means that there is a per-camera sign ambiguity along the axis of rotation. Note that this is not a shortcoming of our algorithm, but this ambiguity is rather inherent due to the planar motion setting. However, the qualitative orientations of the cameras w.r.t. the rotation axis are often known. For example, the cameras might be known to observe a motion on the ground plane. Then the axis of rotation should point upwards in the camera images, otherwise the camera is mounted upside-down. Using this additional assumption, the sign ambiguity can be resolved.

Using the orthogonality and equality of norm constraints, it is tempting to omit the registration step in the plane of rotation and to directly set up the system of equations

$$\lambda_k^2\mathbf{I}_2 = \mathbf{P}^k_{[:,1:3]}{\mathbf{P}^k_{[:,1:3]}}^T = \mathbf{P}^k_{[:,1:3]}\mathbb{P}_{\mathbf{V}}{\mathbf{P}^k_{[:,1:3]}}^T + \mathbf{P}^k_{[:,1:3]}\mathbb{P}_{\mathbf{a}}{\mathbf{P}^k}^T_{[:,1:3]} \tag{21}$$

$$= \left(\mathbf{P}^k_{[:,1:3]}\mathbf{V}\mathbf{Q}^{-T}_{2\times 2}\right)\mathbf{Q}^T_{2\times 2}\mathbf{Q}_{2\times 2}\left(\mathbf{Q}^{-1}_{2\times 2}\mathbf{V}^T{\mathbf{P}^k_{[:,1:3]}}^T\right) + \mathbf{w}^k{\mathbf{w}^k}^T \tag{22}$$

in the three unknowns of $\mathbf{Q}^T_{2\times 2}\mathbf{Q}_{2\times 2}$, the $2K$ unknowns of $\mathbf{w}^k$, and the K unknowns λ_k^2. Interestingly, these constraints on the camera axes are insufficient to compute a valid matrix $\mathbf{Q}_{2\times 2}$ and valid vectors $\mathbf{w}^k$, even using non-linear local optimization methods (there are solutions with residuum 0 which however turn out to be invalid solutions). Moreover, experiments showed that this nonlinear formulation suffers from many local minima. This observation justifies the need for the registration step in the plane of motion.

Final Step Once the first three columns of the camera matrices are known in an Euclidean reference frame, the first three rows in Eq. (14) become linear in the unknowns $\mathbf{Q}_{3\times 2}$, $\mathbf{S}$, and the camera translations. A least squares approach again provides the solutions to the unknowns of this overdetermined linear system. The linear system has a $4+K$-dimensional nullspace in the noisefree case: 4 degrees of freedom due to the planar translational ambiguities (planar translation of the points or the cameras can be compensated by the planar motion) and K degrees of freedom for the per-camera translation ambiguities along the axis of rotation.

6 Results

If synthetic data is generated with affine cameras and without noise, the algorithm expectedly finds the exact solution in closed-form, even for the case of only two cameras each of them tracking one single point. Based on our experience with synthetic data according to a more realistic setting (i.e. projective camera models with realistic internal parameters, some noise and plausible planar motions) we concluded that the robustness of the algorithm strongly depends

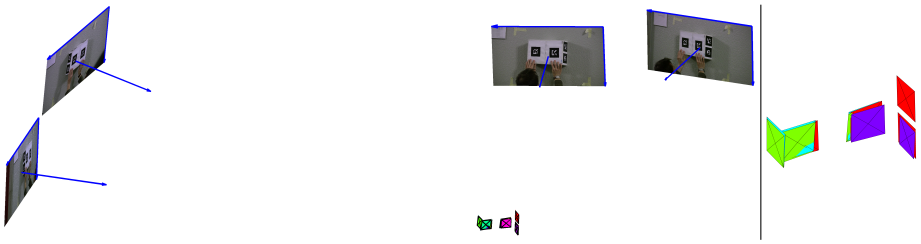

Fig. 2. Reconstruction of a planarly moving box: The right image shows a close-up view of the reconstructed structure (tags tracked by one specific camera share the same color).

on the observed motion. This is actually an expected behavior. If the motion clearly spans the 5D motion subspace, the algorithm works robustly. However, if a dimension of this subspace is not explored sufficiently, noise will overrule this dimension and the reconstruction will deteriorate.

As a proof of concept the algorithm has been applied to a real data sequence. Fig. 2 shows the results of a real sequence with four cameras observing the planar motion of a rigid box. The translation ambiguity along the rotation axis has been resolved s.t. the centroids of the front-facing tags share the same coordinate along the axis of rotation. A template based tracker [17] has been used to generate the feature trajectories. Each camera tracked between 10 to 20 points. Even though some cameras actually tracked the very same points, the algorithm was purposely not aware of these correspondences. Such hidden correspondences allow to evaluate the accuracy of the reconstruction. Based on the overlapping area of the 3D model of the tracked feature tags, we conclude that the algorithm succeeds in computing an accurate reconstruction given the fact that the reconstruction is based on the approximate affine camera model and the solution is given in a non-iterative closed-form. The reprojection error of the closed-form solution is $\frac{1}{\sqrt{F\sum_k N_k}}\|\mathbf{W}-\mathbf{M}\mathcal{C}_{(f)}\left[\Rightarrow_k \mathbf{S}_k\otimes\mathbf{P}_k^T\right]\|_F = 8.95$ pixels (the resolution of the cameras is 1920×1080). A successive nonlinear refinement step still based on the affine camera model did not improve the reprojection error. This provides evidence that most of the error is due to the discrepancy between the employed affine camera approximation and the real projective cameras and not due to the sub-optimal sequential steps of the closed-form solution.

7 Conclusions and Future Work

This paper presented an analysis of a planarly moving rigid object observed by multiple static affine cameras. The theoretical insights gained thereby enabled the development of an algorithm, which provides a closed-form solution to the shape from motion reconstruction problem where no feature point correspondences between the different camera views exist. The motion correspondence, namely that all the cameras observe the same planar motion, was captured by a

5D motion subspace. As future work, we plan to adapt the planar motion subspace constraint to a formulation with projective camera models. This probably asks for iterative solutions for which the closed-form algorithm might provide a good initialization. We also consider trying whether the rank-5 constraint could be used as a means to temporally synchronize multiple camera streams.

Acknowledgments We gratefully acknowledge the support of the 4DVideo ERC Starting Grant Nr. 210806 and a Packard Foundation Fellowship.

References

1. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. International Journal of Computer Vision **9** (1992) 137–154
2. Torresani, L., Hertzmann, A., Bregler, C.: Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. IEEE Trans. Pattern Anal. Mach. Intell. **30** (2008) 878–892
3. Tresadern, P.A., Reid, I.D.: Articulated structure from motion by factorization. In: CVPR (2), IEEE Computer Society (2005) 1110–1115
4. Yan, J., Pollefeys, M.: A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. IEEE Trans. Pattern Anal. Mach. Intell. **30** (2008) 865–877
5. Tron, R., Vidal, R.: A benchmark for the comparison of 3-d motion segmentation algorithms. In: CVPR, IEEE Computer Society (2007)
6. Sturm, P.F., Triggs, B.: A factorization based algorithm for multi-image projective structure and motion. In Buxton, B.F., Cipolla, R., eds.: ECCV (2). Volume 1065 of Lecture Notes in Computer Science., Springer (1996) 709–720
7. Hartley, R., Schaffalitzky, F.: PowerFactorization : 3D reconstruction with missing or uncertain data. In: Japan-Australia Workshop on Computer Vision. (2004)
8. Chen, P.: Optimization algorithms on subspaces: Revisiting missing data problem in low-rank matrix. International Journal of Computer Vision **80** (2008) 125–142
9. Li, J., Chellappa, R.: A factorization method for structure from planar motion. In: WACV/MOTION, IEEE Computer Society (2005) 154–159
10. Vidal, R., Oliensis, J.: Structure from planar motions with small baselines. In Heyden, A., Sparr, G., Nielsen, M., Johansen, P., eds.: ECCV (2). Volume 2351 of Lecture Notes in Computer Science., Springer (2002) 383–398
11. Bue, A.D., de Agapito, L.: Non-rigid stereo factorization. International Journal of Computer Vision **66** (2006) 193–207
12. Svoboda, T., Martinec, D., Pajdla, T.: A convenient multicamera self-calibration for virtual environments. Presence **14** (2005) 407–422
13. Daniilidis, K.: Hand-eye calibration using dual quaternions. I. J. Robotic Res. **18** (1999) 286–298
14. Angst, R., Pollefeys, M.: Static Multi-Camera Factorization Using Rigid Motion. In: Proc. of ICCV '09, Washington, DC, USA, IEEE Computer Society (2009)
15. Kolda, T.G., Bader, B.W.: Tensor Decompositions and Applications. SIAM Review **51** (2009) 455–500
16. Angst, R., Pollefeys, M.: 5d motion subspaces for planar motions: Supplemental material. `www.cvg.ethz.ch/people/phdstudents/rangst/Publications` (2010)
17. D. Wagner and D. Schmalstieg: Artoolkitplus for pose tracking on mobile devices. In: Proc. of 12th Computer Vision Winter Workshop. (2007)