A Unified View on Deformable Shape Factorizations

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. Multiple-view geometry and structure-from-motion are well established techniques to compute the structure of a moving rigid object. These techniques are all based on strong algebraic constraints imposed by the rigidity of the object. Unfortunately, many scenes of interest, e.g. faces or cloths, are dynamic and the rigidity constraint no longer holds. Hence, there is a need for non-rigid structure-from-motion (NRSfM) methods which can deal with dynamic scenes. A prominent framework to model deforming and moving non-rigid objects is the factorization technique where the measurements are assumed to lie in a low-dimensional subspace. Many different formulations and variations for factorization-based NRSfM have been proposed in recent years. However, due to the complex interactions between several subspaces, the distinguishing properties between two seemingly related approaches are often unclear. For example, do two approaches just vary in the optimization method used or is really a different model beneath?

In this paper, we show that these NRSfM factorization approaches are most naturally modeled with tensor algebra. This results in a clear presentation which subsumes many previous techniques. In this regard, this paper brings several strings of research together and provides a unified point of view. Moreover, the tensor formulation can be extended to the case of a camera network where multiple static affine cameras observe the same deforming and moving non-rigid object. Thanks to the insights gained through this tensor notation, a closed-form and an efficient iterative algorithm can be derived which provide a reconstruction even if there are no feature point correspondences at all between different cameras. An evaluation of the theory and algorithms on motion capture data show promising results.

1 Introduction and Related Work

Factorization-based methods for structure-from-motion problems are the focus of this paper. Since this paper builds heavily upon previous factorization formulations, references to prior work in this area will often be given at the appropriate places throughout the text. Here, only a short overview of the most important developments in the area of factorizations for rigid and non-rigid structure-from-motion problems will be presented.

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part VI, LNCS 7577, pp. 682–695, 2012.

Numerous extensions of the classical low-rank factorization approach for rigid structure-from-motion have been proposed ever since Tomasi and Kanade's seminal work [1]. In their work, 2D feature point tracks of a single rigidly moving object observed by an affine camera have been shown to be restricted to a 4-dimensional subspace. In a similar way, trajectories of multiple independently moving objects also give rise to a low-dimensional data matrix [2]. This time however, the trajectories originate from multiple independent subspaces and the SfM-problem gets combined with a motion segmentation problem [3]. Articulated objects can also be modeled with low-rank factorizations and multiple subspaces. In contrast to multiple independent rigid objects, the subspaces of articulated objects are not independent anymore and can intersect each other [4,5]. In this paper, we focus on non-articulated deformable objects. In contrast to the previously mentioned approaches, non-rigid structure-from-motion (NRSfM) enjoys less strict algebraic constraints: the low-rank assumption only holds approximatively. The classical way to model these kind of deformations is the basis-shape model [6] which will be presented in detail in Sec. 3.2. Basis-shape factorization approaches suffer from the fact that the initial factorization must be corrected with a so-called corrective transformation in order to account for the algebraic Kronecker-structure prescribed by the basis-shape model. Later work addressed this issue in detail [7] and presented a closed-form linear solution [8] or a more robust non-linear solution [9]. Another line of research for NRSfM are piecewise approaches which depart from the classical factorization framework. With piecewise NRSfM we mean non-factorization based approaches which build a patch-based representation of non-rigid deformable shapes and glue these patches together using heuristics such as smoothness assumptions of motion and shape. Even though piecewise approaches for NRSfM present an interesting line of future research, in this paper we solely focus on factorization approaches for NRSfM and refer to [10] and references therein for piece-wise NRSfM.

As already mentioned, this paper presents a unified view of low-rank models for 3D point trajectories of non-rigidly moving 3D point clouds, such as basis-shape approaches [11], implicit low-rank shape models [12], or such as the more recent representations using a Discrete Cosine Transform (DCT) basis [13,14]. A tensor-based formulation is derived which seamlessly handles the case of multiple cameras, subsuming earlier models for binocular NRSfM [15]. Factorizations for camera networks have gained renewed interests in the last couple of years [16,17]. Our approach builds heavily upon our previous work [16] and the resulting formulation allows to give a clear and intuitive description of the algebraic constraints encapsulated in 2D feature point trajectories seen in different cameras. Equipped with this deeper understanding, a factorization-based algorithm which provides the solution in closed form can be derived. Alternatively, an iterative multi-linear optimization can be used which can handle partial feature tracks. In contrast to recently presented NRSfM algorithms for multiple cameras [18,19,17], our algorithms do not require feature point correspondences between different cameras. This is analogous to [16] where similar results have been

Table 1. Formulations used throughout this paper

Symbol	Meaning
$\mathbf{a}$	Vectors.
$\mathbf{A}$	Matrices.
$\mathcal{A}$	Tensors.
$\mathrm{vec}(\mathbf{A})$	Column-wise vectorization, in Matlab notation `A(:)`.
$\mathcal{A} = \mathcal{S} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$	Three-mode Tucker-tensor decomposition with core tensor $\mathcal{S}$.
$\mathcal{A}_{(i)}$	Flattening of tensor $\mathcal{A}$ along mode i.
$[\Downarrow_i \mathbf{A}_i]$	Vertical stacking of matrices $\mathbf{A}_i \in \mathbb{R}^{m_i \times n}$ below each other. Sometimes the range of the index i will be indicated for clarity reasons in the following way $\left[\Downarrow_{i=1}^{I} \mathbf{A}_i\right]$.
$[\Rightarrow_i \mathbf{A}_i]$	Horizontal stacking of matrices $\mathbf{A}_i \in \mathbb{R}^{m \times n_i}$ next to each other.
$[\searrow_i \mathbf{A}_i]$	Block-diagonal stacking of matrices $\mathbf{A}_i \in \mathbb{R}^{m_i \times n_i}$.
$\mathbf{A} \otimes \mathbf{B}$	Kronecker product.
$\mathbf{A}_{m \times n}$	The size of a matrix is sometimes indicated in subscripts.
$\mathbf{I}_m$	Identity matrix of dimension m
$k \in \{1, \ldots, K\}$	Index of camera.
$f \in \{1, \ldots, F\}$	Index of frame.
$n \in \{1, \ldots, N\}$	Index of point.
$b \in \{1, \ldots, B\}$	Index of basis shape.

derived for rigidly moving objects. In summary, the main contributions of this paper are:

i) A unified formulation for low-rank non-rigid deforming shapes which clearly reveals the interactions of all the involved subspaces and enables an intuitive reasoning about these subspaces thereby avoiding getting lost in shuffling around indices. As we will see, this also facilitates the development of algorithms.
ii) A closed-form factorization algorithm or a non-linear iterative algorithm which compute the 3D reconstruction given 2D feature tracks in multiple cameras. No feature point correspondences between different cameras need to be known.

2 Notation

The notational conventions used in this paper are summarized in Tab. 1. The paper makes use of the following Kronecker product properties between matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ of appropriate sizes

$$\mathbf{ACB} = \mathbf{D} \Leftrightarrow \left[\mathbf{B}^T \otimes \mathbf{A}\right] \mathrm{vec}(\mathbf{C}) = \mathrm{vec}(\mathbf{D}) \tag{1}$$

$$[\mathbf{A} \otimes \mathbf{B}][\mathbf{C} \otimes \mathbf{D}] = [\mathbf{AC} \otimes \mathbf{BD}]. \tag{2}$$

Tensor algebra and especially the Tucker tensor decomposition will be used in later sections. Due to space limitations, the interested reader is referred to the tensor tutorial [20] or to our previous work [16].

3 Low-Rank Non-rigid Deformations

3.1 Redundancy in Trajectories

Wolf and Zomet's work [15] considers the case of two cameras observing a non-rigid object. They assume that every 3D point tracked in the second camera can be expressed as a linear combination of some of the 3D points tracked by the first camera. This approach can be generalized by assuming that any point $\mathbf{x}_{n,f} \in \mathbb{R}^3$ in 3D-space can be expressed as a linear combination $\mathbf{x}_{n,f} = \mathbf{Y}_f \mathbf{s}_n$ of d_S time-varying basis points $\mathbf{Y}_f \in \mathbb{R}^{3\times d_S}$. Stacking the data from multiple points over multiple frames into one matrix gives

$$\mathbf{X} = \left[\Downarrow_f \Rightarrow_n \mathbf{x}_{f,n}\right] = \left[\Downarrow_f \mathbf{Y}_f\right]\left[\Rightarrow_n \mathbf{s}_n\right] = \mathbf{Y}\mathbf{S} \in \mathbb{R}^{3F\times N}. \tag{3}$$

This representation reveals two important facts: Firstly, the matrix $\mathbf{X}$ is highly redundant as it factorizes into two lower-rank matrices (given $d_S < \min(3F, N)$) and secondly, the temporally varying part $\mathbf{Y}$ is split from the temporally static part $\mathbf{S}$. This low-rank factorization due to redundancies in trajectories lies at the heart of all bilinear non-rigid shape models. This representation has also been suggested in [12] where this low-rank model for 3D trajectories is called 3D-implicit low-rank shape model. As will be seen in Sec. 4, this low-rank model leads to severe ambiguities in the 3D structure for monocular image sequences: For any regular 3-by-3 matrices $\mathbf{H}_f$, $\mathbf{X}$ and $\left[\searrow_f \mathbf{H}_f\right]\mathbf{X}$ will fulfill the same low-rank constraint leaving the dynamic 3D structure ambiguous. Hence, in monocular sequences there is a need for additional constraints, such as a Kronecker structure due to a basis-shape model (see Sec. 3.2) or smoothness priors on $\mathbf{Y}_f$ and as-rigid-as-possible assumption on $\mathbf{S}$ as done in [12]. Note for multiple cameras however, the low-rank assumption itself is sufficient and no additional constraints are necessary (see also Sec. 5.2). Related to the above formulation is the implicit model of [21] where the low-rank model has not been applied directly to the 3D trajectory matrix $\mathbf{X}$ but rather to the observed 2D image trajectories. Specifically, the observed image trajectories were given by $\mathbf{W} = \mathbf{A}\mathbf{S}$ where $\mathbf{A} = \left[\searrow_f \mathbf{C}_f\right]_{2F\times 3F}\mathbf{Y}$ is a combination between the time-varying basis points $\mathbf{Y}$ and the affine camera matrices $\mathbf{C}_f$ of a single moving camera. However, [21] did not enforce the correct algebraic structure on $\mathbf{A}$ and therefore this low-rank model regularizes the 2D feature tracks but does not provide any 3D reconstruction of the moving points.

3.2 Basis-Shapes

Traditionally, the 3D shape $\mathbf{X}_f \in \mathbb{R}^{3\times N}$ of a deformable object at frame f is modeled as a linear combination $\mathbf{X}_f = \sum_{b=1}^{B} \mathbf{\Omega}_{f,b}\mathbf{S}_b = \left[\mathbf{\Omega}_{f,:} \otimes \mathbf{I}_3\right]\left[\Downarrow_b \mathbf{S}_b\right]$ of B temporally static basis shapes $\mathbf{S}_b \in \mathbb{R}^{3\times N}$ with $b = 1, \cdots, B$, weighted by time-varying weights $\mathbf{\Omega}_{f,b} \in \mathbb{R}$ [6]. Collecting the data over all frames leads to

$$\mathbf{X} = \left[\Downarrow_f \mathbf{X}_f\right] = \left[\mathbf{\Omega} \otimes \mathbf{I}_3\right]\left[\Downarrow_b \mathbf{S}_b\right]. \tag{4}$$

Hence, the basis shape approach follows from Eq. (3) by choosing $\mathbf{Y} = \mathbf{\Omega} \otimes \mathbf{I}_3$.

3.3 Smooth Trajectories with Discrete Cosine Transform

Now, assume that the 3D shape $\mathbf{X}_f$ deforms smoothly over time, which implies that the x-, y-, and z-coordinates of the time-varying basis points can be represented in a truncated Discrete Cosine Transform (DCT) basis $\mathbf{Y} = [\mathbf{D} \otimes \mathbf{I}_3]$ where $\mathbf{D} \in \mathbb{R}^{F \times B}$ is the truncated DCT basis. We notice that this is of the same algebraic form as the previous basis shape formulation. The semantic connection is that a smooth trajectory motion in the basis shape model implies that the basis shape weights vary smoothly as well and can therefore be represented in a truncated DCT basis $\mathbf{\Omega} = \mathbf{DQ}$ where $\mathbf{Q} \in \mathbb{R}^{B \times B}$ is the change-of-basis matrix. Inserting this into Eq. (4) and using the Kronecker product property of Eq. (1) we get the chain of equations $\mathbf{X} = \big[\mathbf{DQ} \otimes \mathbf{I}_3\big]\big[\Downarrow_b \mathbf{S}_b\big] = \big[\mathbf{D} \otimes \mathbf{I}_3\big]\big[\mathbf{Q} \otimes \mathbf{I}_3\big]\big[\Downarrow_f \mathbf{S}_b\big]$. The change of basis of the basis shapes weights thus lead to new basis shapes $[\mathbf{Q} \otimes \mathbf{I}_3][\Downarrow_b \mathbf{S}_b]$. Each column of this new basis shape matrix corresponds to one smoothly moving 3D point of the deformable object and captures the collection of coefficients for its three separate F-dimensional trajectories in x-, y, and z-direction expressed in a truncated DCT basis. This representation clearly reveals that a linear transformation of the basis shape coefficients implies a change of the basis shapes (and the other way around) due to the bilinearity of the shape representation in Eq. (4). A slightly different derivation of this observation has first appeared in [13,14][1] where it was called duality of the shape and trajectory basis. It is important to highlight that from an algebraic point of view, the trajectory space approach is completely equivalent to the basis shape approach.

4 Projecting Low-Rank Non-rigid Deformations

Having established the low-dimensional structure of deforming 3D points in the previous section, this section presents an analysis of the resulting 2D trajectories observed in affine cameras. We will see that the image observations originate from three interacting subspaces which are most naturally modeled in a multilinear algebra framework.

4.1 General Low-Rank Non-rigid Motion

In preparation for multiple cameras, the bilinear models for non-rigid trajectories of the previous section are slightly reformulated: the rigid component of the non-rigid motion is modeled explicitly with a temporally varying rotation $\mathbf{R}_f$ and translation $\mathbf{t}_f$. This has two advantages: firstly, the non-rigid deformation does not need to explain the rigid component of the motion which is advantageous especially for large rigid transformations, and secondly it facilitates the extension of the model with a camera rig observing the deformable object.

[1] In [13], the basis $\mathbf{I}_3 \otimes \mathbf{D}$ has been used which is a column and row permutation of $\mathbf{D} \otimes \mathbf{I}_3$ (This has also been noted in [22]). As our derivation shows, the latter version is more natural and has also been used in [14].

In the most general case, the non-rigid trajectories are given by

$$\begin{pmatrix}\mathbf{x}_{f,n}\\1\end{pmatrix}=\begin{bmatrix}\mathbf{R}_f & \mathbf{t}_f\\ \mathbf{0}_{1\times 3} & 1\end{bmatrix}\begin{bmatrix}\mathbf{Y}_f & \mathbf{0}_{3\times 1}\\ \mathbf{0}_{1\times d_S} & 1\end{bmatrix}\begin{bmatrix}\mathbf{s}_n\\1\end{bmatrix}=\begin{bmatrix}\mathbf{R}_f\mathbf{Y}_f & \mathbf{t}_f\\ \mathbf{0}_{1\times d_S} & 1\end{bmatrix}\begin{bmatrix}\mathbf{s}_n\\1\end{bmatrix}. \tag{5}$$

The following derivations mostly follow similar steps as done in [16]. By making use of the Kronecker-product property of Eq. (1), the projection of point n into affine camera axis $\mathbf{c}_k^T \in \mathbb{R}^{1\times 4}$ at frame f is given by

$$\mathcal{W}_{k,f,n}=\mathbf{c}_k^T\begin{pmatrix}\mathbf{x}_{f,n}\\1\end{pmatrix}=\mathbf{c}_k^T\begin{bmatrix}\mathbf{R}_f\mathbf{Y}_f & \mathbf{t}_f\\ \mathbf{0}_{1\times d_S} & 1\end{bmatrix}\begin{pmatrix}\mathbf{s}_n\\1\end{pmatrix}=\operatorname{vec}\left(\begin{bmatrix}\mathbf{R}_f\mathbf{Y}_f & \mathbf{t}_f\\ \mathbf{0}_{1\times d_S} & 1\end{bmatrix}\right)^T\left[\begin{pmatrix}\mathbf{s}_n\\1\end{pmatrix}\otimes\mathbf{c}_k^T\right] \tag{6}$$

$$=\left[\operatorname{vec}\left(\mathbf{R}_f\mathbf{Y}_f\right)^T\ \mathbf{t}_f^T\ 1\right]\mathcal{S}_{(f)}\left[\begin{pmatrix}\mathbf{s}_n\\1\end{pmatrix}\otimes\mathbf{c}_k^T\right] \tag{7}$$

with the flattened core tensor

$$\mathcal{S}_{(f)}=\begin{bmatrix}\mathbf{I}_{d_S}\otimes\begin{bmatrix}\mathbf{I}_3 & \mathbf{0}_{3\times 1}\end{bmatrix} & \mathbf{0}_{3d_S\times 4}\\ \mathbf{0}_{4\times 4d_S} & \mathbf{I}_4\end{bmatrix}\in\mathbb{R}^{3d_S+4\times 4d_S+4}. \tag{8}$$

Stacking the dynamic part row-wise and the temporally non-varying part column-wise leads to a flattened data tensor along the temporal mode

$$\mathcal{W}_{(f)}=\left[\Downarrow_f \Rightarrow_{n,k}\mathcal{W}_{k,f,n}\right]=\mathbf{M}\mathcal{S}_{(f)}(\mathbf{S}^T\otimes\mathbf{C})^T\in\mathbb{R}^{F\times 2KN} \tag{9}$$

with

$$\mathbf{M}=\left[\Downarrow_f\left(\operatorname{vec}\left(\mathbf{R}_f\mathbf{Y}_f\right)^T,\mathbf{t}_f^T,1\right)\right]\in\mathbb{R}^{F\times 3d_S+4} \tag{10}$$

$$\mathbf{S}=\left[\Rightarrow_n\begin{pmatrix}\mathbf{s}_n\\1\end{pmatrix}\right]\in\mathbb{R}^{d_S+1\times N} \tag{11}$$

$$\mathbf{C}=\left[\Downarrow_k\mathbf{c}_k^T\right]\in\mathbb{R}^{2K\times 4}. \tag{12}$$

$\mathbf{M}$, $\mathbf{S}$, and $\mathbf{C}$ capture the motion, structure, and camera subspaces respectively. The flattened data tensor along the temporal mode $\mathcal{W}_{(f)}$ must be of rank $3d_S+4$ due to the revealed factorization. Reshaping this matrix into a third-order data tensor[2] gives $\mathcal{W}=\mathcal{S}\times_k\mathbf{C}\times_f\mathbf{M}\times_n\mathbf{S}$ with core tensor $\mathcal{S}\in\mathbb{R}^{4\times 3d_S+4\times 4d_S+1}$. The rigid motion case presented in [16] results by choosing $d_S=3$ and every $\mathbf{Y}_f$ as the identity matrix.

4.2 Basis-Shape Model

In the more specific basis shape representation (or equivalently in the trajectory basis representation), a property of the Kronecker product (see Eq. (2)) leads to an interesting result if the model of Eq. (4) is again extended with a rigid transformation

$$\begin{bmatrix}\mathbf{X}_f\\ \mathbf{1}_{1\times N}\end{bmatrix}=\begin{bmatrix}\mathbf{R}_f & \mathbf{t}_f\\ \mathbf{0}_{1\times 3} & 1\end{bmatrix}\begin{bmatrix}\mathbf{\Omega}_{f,:}\otimes\mathbf{I}_3 & \mathbf{0}_{3\times 1}\\ \mathbf{0}_{1\times 3B} & 1\end{bmatrix}\begin{bmatrix}\Downarrow_b\mathbf{S}_b\\ \mathbf{1}_{1\times N}\end{bmatrix}=\begin{bmatrix}\mathbf{\Omega}_{f,:}\otimes\mathbf{R}_f & \mathbf{t}_f\\ \mathbf{0}_{1\times 3B} & 1\end{bmatrix}\begin{bmatrix}\Downarrow_b\mathbf{S}_b\\ \mathbf{1}_{1\times N}\end{bmatrix}\in\mathbb{R}^{4\times N}. \tag{13}$$

[2] Such a tensor representation is known as Tucker tensor decomposition [20].

The rigid rotation $\mathbf{R}_f$ interacts with the non-rigid dynamic part $\mathbf{\Omega}$ through a Kronecker product. Inserting Eq. (13) in Eq. (5), we immediately see that the basis shape approach follows from the previous formulation in Eq. (9) by setting $d_S = 3B$ and choosing

$$\mathbf{M} = \left[\Downarrow_f \left(\mathrm{vec}\left(\mathbf{\Omega}_{f,:} \otimes \mathbf{R}_f\right)^T, \mathbf{t}_f^T, 1\right)\right] \in \mathbb{R}^{F \times 9B+4}, \; \mathbf{S} = \begin{bmatrix} \Downarrow_b \mathbf{S}_b \\ \mathbf{1}_{1\times N} \end{bmatrix} \in \mathbb{R}^{3B+1 \times N}. \quad (14)$$

This representation might seem rather unconventional, however it clearly reveals all the multilinear relationships encoded in the data. This representation not only seamlessly handles the case of multiple cameras, but also facilitates the exposure of the unknown matrices for an iterative optimization algorithm. In order to exemplify this fact, we arrange the entries of the tensor in a better-known form revealing a matrix of rank $3B + 1$ which exposes the basis shape matrix $\mathbf{S}$. This arrangement actually corresponds to the transpose of the flattening of the tensor along the mode of the points

$$\mathcal{W}_{(n)}^T = \mathcal{W}_{(f,k)} = \left[\mathbf{C} \otimes \mathbf{M}\right]^T \mathcal{S}_{(n)}^T \mathbf{S} = \left[\Downarrow_f \mathbf{C} \begin{bmatrix} \mathbf{\Omega}_{f,:} \otimes \mathbf{R}_f & \mathbf{t}_f \\ \mathbf{0}_{1\times 3B} & 1 \end{bmatrix}\right] \mathbf{S}. \quad (15)$$

Note that flattening a tensor along a mode is a purely algebraic operation and can easily be done by strictly following some rules. In previous work, usually only one single camera with orthogonal camera axes with non-varying scale has been considered. This corresponds to choosing $\mathbf{C} = [\mathbf{I}_2, \mathbf{0}_{2\times 2}]$ in the above formulation which leads to the standard monocular NRSfM basis shape equation

$$\mathcal{W}_{(n)}^T = \left[\Downarrow_f \left[\mathbf{\Omega}_{f,:} \otimes \mathbf{R}_{f,1:2,:}, \mathbf{t}_{f,1:2}\right]\right] \begin{bmatrix} \Downarrow_b \mathbf{S}_b \\ \mathbf{1}_{1\times N} \end{bmatrix}. \quad (16)$$

At this point it is interesting to put this formulation in relation to Torresani et.al.'s work [23]. Based on the basis-shape model in Eq. (15), [23] proposes a low-rank constraint for optical flow of non-rigid shapes and a 3D reconstruction algorithm for basis-shapes (i.e. merging [6] with Irani's rigid optical flow constraints [24]). Even though presented originally in a completely different way, the optical flow constraint actually is a consequence of considering $\mathcal{W}_{(f)}$ whereas the 3D reconstruction is based on a block-coordinate descent algorithm derived from $\mathcal{W}_{(n)}^T$. [23] even formulated an extension of their 3D reconstruction algorithm for multiple cameras with the simplification that the data is centered, i.e. translations are not modeled. Based on the same formulation, Del Bue and Agapito showed experimentally [18] that a stereo setup indeed improves the reconstruction accuracy of a basis shape model. More recently, Lladó et.al.[19] drew the same conclusion in an iterative Ransac-framework for a binocular stereo setup with perspective cameras. However, similar to [17], all these approaches require feature point correspondences between different cameras to be known because the exact relation between $\mathcal{W}_{(f)}$ and $\mathcal{W}_{(n)}$ has not been established. By making use of this relation, Sec. 5.2 presents an algorithm which can handle cases where no correspondences between different cameras are available.

The interested reader is also referred to Hartley and Vidal's work [25] which presents a solution for the monocular basis-shape model with perspective rather than affine cameras. We leave it as an open question whether there exists a similar solution to the perspective multi-camera basis-shape model[3]. The supplemental material [26] contains additional insights relating to the stability and uniqueness of basis-shape factorizations, the ambiguities of the basis-shape model, and the orthogonality constraints used for finding a corrective transformation after a low-rank factorization of the data matrix $\mathcal{W}_{(n)}^T$ (establishing connections to [27,8,9] amongst others).

5 Results

Since the unified tensor formulation is one of the main contributions, practical as well as theoretical results of this formulation will be presented in the upcoming sections.

5.1 Theoretical Result: Degenerate Low-Rank Non-rigid Motion

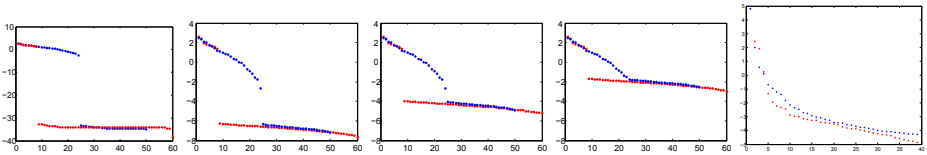
As a theoretical result of the proposed tensor formulation, degenerate motions will be investigated more closely in this section. In the previous derivations we have implicitly assumed that if $\operatorname{rank}\left(\left[\Downarrow_f \mathbf{R}_f\mathbf{Y}_f\right] \in \mathbb{R}^{3F\times N}\right) = d_S$ then $r = \operatorname{rank}\left(\left[\Downarrow_f \operatorname{vec}\left(\mathbf{R}_f\mathbf{Y}_f\right)^T\right] \in \mathbb{R}^{F\times 3N}\right) = 3d_S$. For general matrices, the algebraic rank of such a reshaped matrix indeed fulfills this equality. However, in special cases or in practical cases where the rank is estimated based on a robust rank estimator it might be that $\left[\Downarrow_f \operatorname{vec}\left(\mathbf{R}_f\mathbf{Y}_f\right)^T\right]$ is again redundant and can be factorized even further into two lower-rank matrices $\left[\Downarrow_f \operatorname{vec}\left(\mathbf{R}_f\mathbf{Y}_f\right)^T\right] = \tilde{\mathbf{M}}\mathbf{Q}$ and we get $r < 3d_S$. Such a rank reduction happens for example if the trajectories in x-direction are highly correlated with the trajectories in y- and z-direction which is actually not that uncommon. In such cases, the matrix $\mathbf{Q} = [\mathbf{Q}_1, \cdots, \mathbf{Q}_{d_S}] \in \mathbb{R}^{r\times 3d_S}$ can be absorbed by $\mathcal{S}_{(f)}$

$$\mathbf{M}\mathcal{S}_{(f)} = \left[\tilde{\mathbf{M}}\mathbf{Q}, \left[\Downarrow_f \mathbf{t}_f^T\right], \mathbf{1}_{F\times 1}\right]\mathcal{S}_{(f)} = \left[\tilde{\mathbf{M}}, \left[\Downarrow_f \mathbf{t}_f^T\right], \mathbf{1}_{F\times 1}\right]\underbrace{\begin{bmatrix}\left[\Rightarrow_{b=1}^{d_S}\left[\mathbf{Q}_b, \mathbf{0}_{r\times 1}\right]\right] & \mathbf{0}_{r\times 4}\\ \mathbf{0}_{4\times 4d_S} & \mathbf{I}_4\end{bmatrix}}_{=\tilde{\mathcal{S}}_{(f)}\in\mathbb{R}^{r+4\times 4d_S+4}},$$

which increases the complexity of the factorization since the core tensor is then also (partially) unknown. This complexity on the other hand might lead to greater robustness since the motion subspace can be of any rank r not necessarily equal to $3d_S$. Fig. 1 exemplifies this observation further.

Aware of these degenerate situations, we leave it as future work how to deal with such degenerate motions algorithmically. In the next section, we will assume general non-degenerate motions such as presented in Sec. 4.1.

[3] Note however that even the much simpler extension of the multi-camera rigid model [16] to perspective cameras proves to be rather challenging.



(a) No Noise **(b)** 0.01% Noise **(c)** 0.1% Noise **(d)** 1% Noise **(e)** Facial Mocap

Fig. 1. This figure shows the problem of implicitly choosing the dimensionality of the motion too high. For Fig. 1a-Fig. 1d a sequence of a non-rigidly deforming structure $\mathbf{x}_{f,n}$ with $d_S = 7$ was generated. The motion actually corresponds to two realistically independently moving rigid objects and hence the overall structure is perceived as non-rigid. The logarithms of the resulting singular values of $\mathcal{X}_{(f)} = \left[\Downarrow_f \Rightarrow_n \mathbf{x}_{f,n}^T\right] = \mathbf{M}\mathcal{S}_{(f)}\left[\mathbf{S} \otimes \left[\mathbf{I}_3, \mathbf{0}_{3\times 1}\right]^T\right] \in \mathbb{R}^{F\times 3N}$ and $\mathcal{X}_{(n)} = \left[\Downarrow_f \Rightarrow_n \mathbf{x}_{f,n}\right]^T = \mathbf{S}\mathcal{S}_{(n)}\left[\mathbf{M} \otimes \left[\mathbf{I}_3, \mathbf{0}_{3\times 1}\right]\right]^T \in \mathbb{R}^{N\times 3F}$ are visualized in blue resp. red. Fig. 1a with no noise shows the true underlying ranks $\operatorname{rank}\left(\mathcal{X}_{(n)}\right) = d_S + 1 = 8$ and $\operatorname{rank}\left(\mathcal{X}_{(f)}\right) = 3d_S + 3 = 24$. It can be clearly seen that only a minor increase of Gaussian distributed noise already corrupts the rank-24 approximation of $\mathcal{X}_{(f)}$ considerably. However, the numerical rank of $\mathcal{X}_{(n)}$ is considerably more robust w.r.t. noise. This is a clear indication that a tensor-formulation taking the multi-rank of the tensor into account provides increased robustness. Fig. 1e shows the same analysis for the CMU facial motion capture sequence used in [11].

5.2 Algorithms

From a practical point of view, the tensor framework also facilitates the development of algorithms. Here, we present a closed-form and an iterative algorithm.

Closed-from Factorization Algorithm. Let us consider a multi-camera setup with affine cameras where each camera tracks its own set of feature points: there are no feature point correspondences between different cameras. In order to apply matrix factorizations, the trajectories must be known completely, i.e. from the first to the last frame. Furthermore, a non-degenerate motion according to Sec. 4.1 is assumed. In this case, a closed-form factorization algorithm can be derived following along similar lines as in [16]. In that work, we presented a factorization algorithm for multiple cameras observing a rigidly moving object, i.e. exactly the same setup as considered here except that the motion was rigid. The formulation and derivations in Sec. 4 highlighted a close similarity between the rigid and non-rigid case: a rigid motion corresponds to choosing $d_S = 3$ for the dimensionality of the structure coefficients $\mathbf{s}_n \in \mathbb{R}^{d_S\times 1}$ in Eq. (5). It follows that for deformable objects where $d_S > 3$, the algorithm in [16] can be adapted by changing the dimensionality of the motion matrix and structure matrix accordingly. The intuition of the resulting algorithm is that instead of using point correspondences between different cameras, we make use of the motion correspondence: all the cameras observe the same non-rigid deforming object and this motion correspondence enables the registration of all the cameras in one consistent coordinate frame. Self-calibration can be applied to the camera matrices in order to get a representation of the non-rigid deformation in a

similarity coordinate frame. More detailed derivations are provided in [26], and the interested reader is also referred to [16]. According to our knowledge, this is the first closed-form reconstruction algorithm for multiple affine cameras which track feature points on a low-rank non-rigid object without correspondences between different cameras. The existence of such an algorithm is quite surprising since

i) without correspondences between different cameras, no feature points can be triangulated using the rigidity of the object when observed from multiple cameras at the same point in time.
ii) there is no rigidity constraint between successive frames and hence, a feature point tracked in just one camera can not be directly triangulated by standard multiple-view techniques either.
iii) an independent reconstruction per camera is not possible without further assumptions.

Iterative Refinement. The previously described algorithm follows several sequential steps and is thus not optimal in the sense that errors in early steps are propagated and maybe even amplified in subsequent steps. However, this algorithm can serve as an initialization for an iterative optimization for

$$\min_{\mathbf{M},\mathbf{S},\mathbf{C}} \frac{1}{2} \left\| \mathbf{H} \odot \left[\mathbf{M}\mathcal{S}_{(f)} \left[\mathbf{S} \otimes \mathbf{C}^T \right] - \mathcal{W}_{(f)} \right] \right\|_F^2, \tag{17}$$

where $\mathbf{H}$ masks the unobserved entries and $\odot$ denotes the Hadamard (elementwise) product. We implemented an alternating least squares (ALS) method which is a straight-forward and efficient algorithm for optimizing multilinear problems of this form. One has to keep in mind however, that the number of unknowns is quite large and the Kronecker-structure in the Jacobians must be used wisely otherwise performance suffers too much. For the sake of completeness, the Jacobians are provided in the supplemental material [26]. Note that ALS is known to flatline rather quickly when not properly initialized requiring lots of random multiple restarts. Based on our experiments with random intialization, this is indeed also the case in the above trilinear problem of Eq. (17). With the initialization provided by the closed-form algorithm however, ALS converged in very few iteration and we never had to randomly reinitialize ALS. This showcases the quality and accuracy of the solution provided by the closed-form algorithm.

In the presence of incomplete trajectories, this iterative optimization can obviously also be used, even though the initialization is slightly more tricky since the closed-form algorithm can no longer be applied. As a simplification or initialization, the motion subspace can be fixed to a truncated DCT-basis in the case of smooth motions.

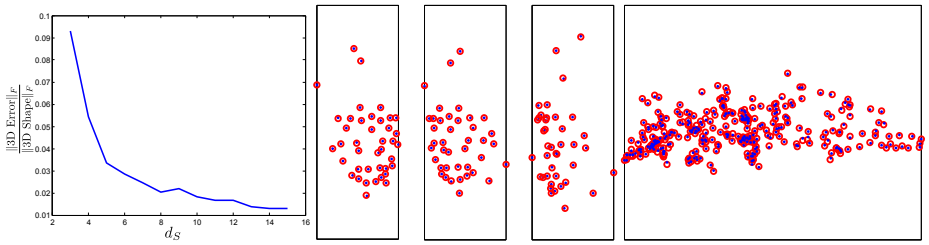
Comparison to [17]. At this point, a comparison with Zaheer et.al.'s recently proposed algorithm [17] is suitable. Their work addresses the same setup of multiple affine cameras observing a non-rigid scene. In a nutshell, the algorithm proposed in [17] reads in our tensor formulation like (we also refer to the supplemental material [26] for further details):

1. Impute missing 2D trajectory entries with a truncated DCT interpolation of the known entries.
2. Compute an orthogonal basis $\mathbf{M}$ for the dominant subspace of the completed trajectories by factorizing $\mathcal{W}_{(f)}$.
3. Factorize $\mathcal{W}_{(k)}\,[\mathbf{I}_N \otimes \mathbf{M}]$ in rank-3 matrices in order to extract the first three columns of the camera matrices.

The last two steps actually correspond to a partial first iteration of the Higher-Order SVD algorithm [20] where the dominant subspaces of a tensor are alternatingly exposed by reshaping the tensor in matrix form and performing PCA to extract the dominant subspaces. Zaheer et.al.'s algorithm assumes *all* the correspondences between cameras to be known, or if some entries are missing, these entries must be interpolated in the first step. Otherwise, the factorization in the third step can not be performed. Furthermore, the input data is assumed to be centered, hence the translations are not modeled and the camera matrices are of rank 3. Subtracting the column means in a centering step factors out the plane at infinity and results automatically in a reconstruction w.r.t. an affine coordinate frame. A standard rank 4 factorization including translations would result in a projective reconstruction asking for more complex auto-calibration techniques. In contrast, even though our formulation explicitly models the translations, the resulting reconstruction is still in an affine frame. Our work is different in many aspects and refines the results in [17] considerably. Using the tensor formulation, we see that the derivation of [17] is missing an important algebraic relation (specifically the algebraic relation between Eq. (2) and (9) in their paper). As seen in the previous sections, thanks to the tensor formulation a closed-form or an iterative algorithm can be derived where no correspondences between different cameras need to be known. This is obviously a considerably stronger result. Of course, if correspondences between cameras are available, then these algorithms derived from the tensor formulation can and should make use of them. Moreover, our presentation assumes no centered data which corrects for the fact that the translational part can *not* be eliminated by subtracting the column mean in the presence of missing data. This has been overseen in [17] and hence the derivation and their algorithm only hold if *all* the data is known in the centering step and entries are only deleted afterward. Our algorithms are therefore applicable whenever [17] is but cover additional settings as well (e.g. missing correspondences between cameras and correct handling of translation in case of incomplete trajectories).

5.3 Experiments

The CMU facial motion capture sequence, which has also been used in [11], is used for practical evaluation because this is a widely used dataset for NRSfM. The facial mocap sequence is projected into $K = 3$ affine cameras spaced 45 degrees apart from each other where the middle one is facing the face directly from the front and is equal to the camera used by [11]. Each camera observed all of the 40 points, however no correspondences between the different cameras



(a) Relative 3D reconstruction error **(b)** Reprojection error at frame f=132 **(c)** Reprojection error in single-point experiment

Fig. 2. Results of facial mocap sequence: Fig. 2a shows the relative 3D error of the reconstructed 3D trajectories as a function of the structure dimensionality. Compared to [11], our approach achieves only slightly better results: an indicator that both Torresani et.al.'s and our approach succeed in an accurate reconstruction, even though using completely different presuppositions ([11] works in a monocular blend-shape setting with strong smoothness prior on shape and motion). Fig. 2b shows the ground truth in blue and reprojections in red for the three cameras at frame $f = 132$ for $d_S = 15$. Fig. 2c shows the reprojection error in camera 3 of the single-point reconstruction experiment. The reprojection error for this camera is roughly 3 pixels for an image of resolution 1200×1200.

have been enforced. The closed-form algorithm has been used for initializing 20 iterations of ALS. The results are summarized in Fig. 2. An interesting observation we made was that the reprojection error directly after the closed-form algorithm was sometimes quite high (up to roughly 10% of the camera resolution). However, after one single iteration of ALS, the error often dropped below 1%. We explain this observation by referring to Fig. 1: the closed-form algorithm is based on several sequential steps. The initial step is a low-rank factorization of the data tensor flattened along the temporal mode $\mathcal{W}_{(f)}$. This corresponds to the blue points in these figures. Since the low-rank model holds only approximatively, a considerable amount of noise might mix in which then propagates to the subsequent steps. This sometimes leads to inaccurate structure estimates $\mathbf{S}$, the cameras and the majority of the motion matrix were usually estimated quite accurately. A few ALS iterations can correct for this error propagation, as shown in the results.

The algorithm can also handle the extreme minimal case where a camera only tracks one single feature point which is not in correspondence with any of the other feature points tracked by the other cameras. As long as all the cameras together provide sufficient data to estimate the motion space $\mathbf{M}$, one trajectory is sufficient to reconstruct the camera pose and the 3D motion of this trajectory. In order to validate this, a single point was selected from camera 3 and all the remaining ones of camera 3 were omitted in the algorithm. With $d_S = 10$, for an almost rigidly moving point an overall relative 3D error of 0.021 resulted and restricting to only the selected point, the relative 3D error of this trajectory was 0.044. For a largely non-rigidly moving point on the mouth, the overall relative

3D error was 0.025 and the relative 3D error of the trajectory was 0.061. We refer to Fig. 2 for the reprojection error of this non-rigid point in camera 3.

6 Conclusion and Future Work

This paper presented a unified formulation based on tensor algebra for factorization-based NRSfM approaches thereby expressing monocular, binocular, and camera network approaches for NRSfM in a common framework. We have shown that the natural way to capture such multilinear interactions between a motion subspace, camera subspace, and a structure subspace is a Tucker-tensor decomposition. This new understanding of how the subspaces interact with each other enabled us to come up with a closed-form and an iterative algorithm which can handle the case where no feature point correspondences between different cameras are available. Experiments validated the presented algorithms on motion capture data.

As future work, we plan to address the dimensionality selection problem for the motion and structure subspaces. Sec. 5.1 already presented some first theoretical insights but we are currently lacking an algorithm to handle these situations in a principled manner. A first idea might be to impose trace norm regularization terms on the reshaped tensors $\mathcal{W}_{(f)}$ and $\mathcal{W}_{(n)}$ thereby automatically favoring low-dimensional motion and structure subspaces. Another thrust for future research is the application of the low-rank constraints amongst different cameras for optical flow: the flow field of each camera is constrained by one joint motion subspace. From an optimization point of view, the robust handling of outliers in the measurement remains an open issue.

Acknowledgments. Roland Angst is a recipient of the Google Europe Fellowship in Computer Vision, and this research is supported in part by this Google Fellowship. The research has also received funding from the European Research Council under the European Commnunitys Seventh Framework Programme (FP7/2007-2013) / 4DVideo Starting Grant agreement Nr. 210806.

References

1. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. IJCV 9(2), 137–154 (1992)
2. Costeira, J., Kanade, T.: A multi-body factorization method for motion analysis. In: Proc. ICCV, pp. 1071–1076 (June 1995)
3. Tron, R., Vidal, R.: A benchmark for the comparison of 3-D motion segmentation algorithms. In: Proc. CVPR. IEEE Computer Society (2007)
4. Yan, J., Pollefeys, M.: A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. TPAMI 30(5), 865–877 (2008)
5. Tresadern, P.A., Reid, I.D.: Articulated structure from motion by factorization. In: Proc. CVPR, pp. 1110–1115. IEEE Computer Society (2005)

6. Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3D shape from image streams. In: Proc. CVPR, pp. 2690–2696. IEEE Computer Society (2000)
7. Brand, M.: Morphable 3D models from video. In: Proc. CVPR, pp. 456–463. IEEE Computer Society (2001)
8. Xiao, J., Chai, J.-X., Kanade, T.: A Closed-Form Solution to Non-rigid Shape and Motion Recovery. In: Pajdla, T., Matas, J. (eds.) ECCV 2004, Part IV. LNCS, vol. 3024, pp. 573–587. Springer, Heidelberg (2004)
9. Brand, M.: A direct method for 3D factorization of nonrigid motion observed in 2D. In: CVPR (2), pp. 122–128. IEEE Computer Society (2005)
10. Fayad, J., Russell, C., de Agapito, L.: Automated articulated structure and 3D shape recovery from point correspondences. In: Proc. ICCV, pp. 431–438 (2011)
11. Torresani, L., Hertzmann, A., Bregler, C.: Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. TPAMI 30(5), 878–892 (2008)
12. Paladini, M., Bartoli, A., Agapito, L.: Sequential Non-Rigid Structure-from-Motion with the 3D-Implicit Low-Rank Shape Model. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 15–28. Springer, Heidelberg (2010)
13. Akhter, I., Sheikh, Y., Khan, S., Kanade, T.: Nonrigid structure from motion in trajectory space. In: Proc. NIPS, pp. 41–48 (2008)
14. Akhter, I., Sheikh, Y., Khan, S., Kanade, T.: Trajectory space: A dual representation for nonrigid structure from motion. TPAMI 33(7), 1442–1456 (2011)
15. Wolf, L., Zomet, A.: Correspondence-free synchronization and reconstruction in a non-rigid scene. In: Workshop on Vision and Modeling of Dynamic Scenes (2002)
16. Angst, R., Pollefeys, M.: Static multi-camera factorization using rigid motion. In: Proc. ICCV. IEEE Computer Society, Washington, DC (2009)
17. Zaheer, A., Akhter, I., Baig, M.H., Marzban, S., Khan, S.: Multiview structure from motion in trajectory space. In: Proc. ICCV, pp. 2447–2453 (2011)
18. Bue, A.D., de Agapito, L.: Non-rigid stereo factorization. IJCV 66(2), 193–207 (2006)
19. Lladó, X., Del Bue, A., Oliver, A., Salvi, J., de Agapito, L.: Reconstruction of nonrigid 3D shapes from stereo-motion. Pattern Recogn. Lett. 32, 1020–1028 (2011)
20. Kolda, T.G., Bader, B.W.: Tensor Decompositions and Applications. SIAM Review 51(3), 455–500 (2009)
21. Olsen, S.I., Bartoli, A.: Implicit non-rigid structure-from-motion with priors. J. Math. Imag. Vision 31, 233–244 (2008)
22. Gotardo, P.F.U., Martinez, A.M.: Non-rigid structure from motion with complementary rank-3 spaces. In: Proc. CVPR, pp. 3065–3072. IEEE (2011)
23. Torresani, L., Yang, D.B., Alexander, E.J., Bregler, C.: Tracking and modeling non-rigid objects with rank constraints. In: Proc. CVPR, pp. 493–500 (2001)
24. Irani, M.: Multi-frame optical flow estimation using subspace constraints. In: Proc. ICCV, pp. 626–633 (1999)
25. Hartley, R., Vidal, R.: Perspective Nonrigid Shape and Motion Recovery. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 276–289. Springer, Heidelberg (2008)
26. Angst, R., Pollefeys, M.: A unified view on deformable shape factorization: Supplemental material (2012), `http://www.inf.ethz.ch/personal/rangst/publications.php`
27. Park, H.S., Shiratori, T., Matthews, I., Sheikh, Y.: 3D Reconstruction of a Moving Point from a Series of 2D Projections. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part III. LNCS, vol. 6313, pp. 158–171. Springer, Heidelberg (2010)