# A Unified View on Deformable Shape Factorizations: Supplemental Material


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.** This supplemental material provides additional details and derivations related to the ECCV submission with paper ID 1560 [1]. Sec. 1 presents additional insights related to the basis shape model. These sections are not required for the understanding of the paper. However, additional insights related to the tensor formulation of basis-shape models are given. Sec. 2 describes the steps for the closed-form algorithm in detail. This section does not add any new insight but should facilitate an implementation of this algorithm. The iterative algorithm mentioned in the paper requires the Jacobian of a matrix-valued objective function w.r.t. matrix-valued variables. Sec. 3 provides these Jacobians, rendering an implementation of the iterative algorithm straight-forward. And lastly, Sec. 4 provides a detailed description of the relation between Zaheer et.al.'s previous work [2] and our method.


## 1 Basis-Shapes Representation

In the upcoming subsections, previously established results are recompiled in tensor algebraic form. These sections will therefore not present new results per se, but rather a new way of deriving and reasoning about them. We think that such a unified presentation of previously dispersed results in a consistent formulation will ultimately lead to a better and clearer understanding.

In Sec. 1.1, the non-uniqueness of factorization approaches will be recapitulated shortly, while Sec. 1.2 presents a different view on the results of [3]. Finally, Sec. 1.3 gives a generalized view on the widely-used orthogonality constraints.

### 1.1 Ambiguities

In this section, we will look at the ambiguities inherently contained in the low-rank non-rigid trajectories. To start, let us define the two affine transformations for the camera and the structure

$$\mathbf{Q}_C = \begin{bmatrix} \mathbf{T}_C & \mathbf{t}_C \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \quad \text{and} \quad \mathbf{Q}_S \quad = \begin{bmatrix} \mathbf{T}_S & \mathbf{t}_S \\ \mathbf{0}_{1\times d_S} & 1 \end{bmatrix} \in \mathbb{R}^{d_S+1\times d_S+1}. \qquad (1)$$

These affine transformation are the source for the ambiguities. Applying these transformations to the camera and structure matrices should not modify the core tensor. Hence, solving

$$\mathcal{S} = \mathcal{S} \times_f \mathbf{Q}_M \times_k \mathbf{Q}_C \times_n \mathbf{Q}_S^T \tag{2}$$

for $\mathbf{Q}_M$ gives the transformation for the motion space $\mathbf{M}\mathbf{Q}_M^{-1}$ in order to compensate for the affine transformation of the cameras and the structure. As an explicit form for $\mathbf{Q}_M$ reveals all the ambiguities in the NRSfM formulation, in particular for the basis shape approach, we will go through the necessary algebra. From the Kronecker product properties (see Eq. (1) and (2) in the paper) and Eq. (6) in the paper it follows

$$\begin{aligned}
\mathcal{W}_{:,f,n} =& \mathbf{C}\mathbf{Q}_C^{-1}\mathbf{Q}_C \begin{bmatrix} \mathbf{R}_f\mathbf{Y}_f & \mathbf{t}_f \\ \mathbf{0}_{1\times d_S} & 1 \end{bmatrix} \mathbf{Q}_S\mathbf{Q}_S^{-1} \begin{pmatrix} \mathbf{s}_n \\ 1 \end{pmatrix} && (3)\\
=& \left[\operatorname{vec}\left(\mathbf{T}_C\mathbf{R}_f\mathbf{Y}_f\mathbf{T}_S\right)^T, \left(\mathbf{T}_C\mathbf{R}_f\mathbf{Y}_f\mathbf{t}_S + \mathbf{T}_C\mathbf{t}_f + \mathbf{t}_C\right)^T, 1\right] \mathcal{S}_{(f)} && (4)\\
& \left[\mathbf{Q}_S^{-1}\begin{bmatrix} \mathbf{s}_n \\ 1 \end{bmatrix} \otimes \mathbf{Q}_C^{-T}\mathbf{C}^T\right] && (5)\\
=& \left[\operatorname{vec}\left(\mathbf{R}_f\mathbf{Y}_f\right)^T, \mathbf{t}_f^T, 1\right] \underbrace{\begin{bmatrix} \mathbf{T}_S \otimes \mathbf{T}_C^T & \mathbf{t}_S \otimes \mathbf{T}_C^T & \mathbf{0}_{3d_S\times 1} \\ \mathbf{0}_{3\times 3d_S} & \mathbf{T}_C^T & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{1\times 3d_S} & \mathbf{t}_C^T & 1 \end{bmatrix}}_{=\mathbf{Q}_M^{-1}} \mathcal{S}_{(f)} && (6)\\
& \left[\mathbf{Q}_S^{-1} \otimes \mathbf{Q}_C^{-T}\right] \left[\begin{bmatrix} \mathbf{s}_n \\ 1 \end{bmatrix} \otimes \mathbf{C}^T\right]. && (7)
\end{aligned}$$

We recall how the basis shape approach models the motion, namely as $\mathbf{R}_f\mathbf{Y}_f = \mathbf{\Omega}_{f,:} \otimes \mathbf{R}_f$. As shown in Eq. (3) this motion is right-multiplied with $\mathbf{T}_S$, the non-translational component of the affine transformation of the structure. However, the algebraic structure in the motion must be preserved, hence $[\mathbf{\Omega}_{f,:} \otimes \mathbf{R}_f]\,\mathbf{T}_S$ must have Kronecker-structure as well. If the affine transformation looks like $\mathbf{T}_S = \mathbf{Q}_\Omega \otimes \mathbf{T}_B$ with $\mathbf{Q}_\Omega \in \mathbb{R}^{B\times B}$ and $\mathbf{T}_B \in \mathbb{R}^{3\times 3}$, then again due to the Kronecker product property it follows

$$\mathbf{M}_f\mathbf{T}_S = [\mathbf{\Omega}_{f,:} \otimes \mathbf{R}_f]\,[\mathbf{Q}_\Omega \otimes \mathbf{T}_B] = [\mathbf{\Omega}_{f,:}\mathbf{Q}_\Omega] \otimes [\mathbf{R}_f\mathbf{T}_B], \tag{8}$$

and the algebraic structure gets preserved. Hence, the allowable affine transformations for the basis shape structure belong to a restricted class, specifically the inverse $\mathbf{T}_S^{-1} = \mathbf{Q}_\Omega^{-1} \otimes \mathbf{T}_B^{-1}$ is applied to the structure $[\Downarrow_b \mathbf{S}_b]$

$$\begin{aligned}
\mathbf{T}_S^{-1}\,[\Downarrow_b \mathbf{S}_b] &= \left[\mathbf{Q}_\Omega^{-1} \otimes \mathbf{T}_B^{-1}\right][\Downarrow_b \mathbf{S}_b] && (9)\\
&= \left[\mathbf{Q}_\Omega^{-1} \otimes \mathbf{I}_3\right]\left[\mathbf{I}_B \otimes \mathbf{T}_B^{-1}\right][\Downarrow_b \mathbf{S}_b] && (10)\\
&= \left[\mathbf{Q}_\Omega^{-1} \otimes \mathbf{I}_3\right]\left[\Downarrow_b \mathbf{T}_B^{-1}\mathbf{S}_b\right], && (11)
\end{aligned}$$

which shows that the transformation $\mathbf{T}_B$ transforms all the basis shape basis separately $\mathbf{T}_B^{-1}\mathbf{S}_b$. Since the matrices $\mathbf{R}_f$ represent rotation matrices, the transformations $\mathbf{T}_C$ and $\mathbf{T}_B$ are restricted to scaled rotation matrices, where the scaling factors are reciprocal such that they cancel each other. The transformation $\mathbf{Q}_\Omega$ is an arbitrary regular matrix representing an arbitrary choice for the basis of the basis shape weights $\mathbf{\Omega}$. As previously pointed out, if the basis shape weights vary smoothly, the transformation $\mathbf{Q}_\Omega$ can be chosen such that $\mathbf{\Omega}\mathbf{Q}_\Omega = \mathbf{D}$ represents a truncated DCT basis.

Let us summarize our findings: for the NRSfM basis shape approach, we can freely choose

- a similarity transformation $\mathbf{Q}_C$ for the cameras $\mathbf{P}$
- a scaled rotation matrix $\mathbf{T}_B$ (with inverse scaling w.r.t. the camera similarity transformation) representing the choice of basis for the basis shapes $\mathbf{S}_b$, and a translation vector $\mathbf{t}_S \in \mathbb{R}^{3B\times 1}$
- an arbitrary regular matrix $\mathbf{Q}_\Omega$ representing the arbitrary choice of basis for the basis shape weights.

For the monocular NRSfM basis shape formulation, Brand [4] already verbally mentioned the structure ambiguity in the form of a Kronecker product $\mathbf{T}_S = \mathbf{Q}_\Omega \otimes \mathbf{T}_B$ in a short note. Unfortunately, maybe due to the lack of formally underpinning this insight, the deeper implications of this fact have mostly been overlooked in later work, resulting in lengthy, obscure proofs for the non-uniqueness of the basis shape formulation.

### 1.2 Stability of Monocular Non-Rigid Trajectory Factorization

Restricting ourselves to just one single camera, the original quadrilinear factorization problem in Eq. (15) of the paper reduces to a trilinear problem in the rigid motions $\mathbf{R}_f$ and $\mathbf{t}_f$, the basis shape weights $\mathbf{\Omega}$, and the basis shapes $\mathbf{S}_b$. Furthermore, by precomputing the rigid motion (e.g. from the rigid background), the problem becomes a bilinear factorization problem. Finally, by making use of the smoothly varying basis shape weights prior (which means fixing $\mathbf{\Omega}$ to a truncated DCT basis $\mathbf{D}$), the only unknowns are the $3BN$ DCT coefficients of the $N$ point trajectories which results in a linear system of equations. This is exactly the setup used in [3] where it is shown that a unique solution to this linear problem only exists if the rigid translations are *not* smooth over time, i.e. can not be exactly expressed in the truncated DCT basis. Our representation in Eq. (15) allows to infer further results. Given the remaining unknowns, the uniqueness of the trajectory reconstruction depends solely on the system matrix

$$[\mathbf{C} \otimes \mathbf{M}]^T \mathcal{S}_{(n)}^T = \left[ \Downarrow_f \mathbf{C} \begin{bmatrix} \mathbf{\Omega}_{f,:} \otimes \mathbf{R}_f & \mathbf{t}_f \\ \mathbf{0}_{1\times 3B} & 1 \end{bmatrix} \right] \tag{12}$$

of the linear system in Eq. (15) for the structure $\mathbf{S}$. If the camera matrix $\mathbf{C}$ has full column-rank, i.e. $\operatorname{rank}(\mathbf{C}) = 4$, then the system matrix has full column-rank and there is a unique solution for the structure $\mathbf{S}$ independently whether the rigid translation is smooth or not. However, if the column-rank of the camera matrix is smaller than 4 (this is for example the case for one single extrinsically calibrated camera which has only three rows) then the uniqueness of the solution depends on whether or not the rigid transformation is representable in the basis of the basis shape weights[1]

$$-[\Downarrow_f \mathbf{R}_f \mathbf{t}_f] = [\mathbf{\Omega} \otimes \mathbf{I}_3]\,\mathbf{t}_\Omega + [\mathbf{\Omega} \otimes \mathbf{I}_3]^\perp\,\mathbf{t}_\Omega^\perp, \tag{13}$$

for some coefficients $\mathbf{t}_\Omega \in \mathbb{R}^{3B\times 1}$ and $\mathbf{t}_\Omega^\perp \in \mathbb{R}^{3F-3B\times 1}$. The system matrix has full column-rank only if $\mathbf{t}_\Omega^\perp \neq \mathbf{0}$ and then there is a unique solution. As derived in [3], with noisy data, the robustness of the solution depends on how accurate the point trajectories can be represented in the basis $\mathbf{\Omega}$ and on how strong the component $\mathbf{t}_\Omega^\perp$ of the camera translation is. Only an asymptotic result for the accuracy has been derived, specifically the reconstruction is exact when the ratio between the norm of $\mathbf{t}_\Omega^\perp$ and the reconstruction error of the point trajectory goes to infinity. The linear system in Eq. (15) enables the use of standard tools in linear algebra to investigate the stability of a linear system (e.g. the condition number of the system matrix).

### 1.3 Orthogonality Constraints for Corrective Transformation

**Enforcing the Kronecker-Structure With Corrective Transformation** Low-rank factorization methods, may it be for the monocular, multi-camera, rigid or non-rigid basis shape formulation, generally run into the same problem: The low-rank factorization of a matrix returns two matrix factors, one of which should have a certain bilinear Kronecker-structure with one factor having orthogonal columns or rows. Unfortunately, a general purpose factorization method does not provide factors with the correct algebraic structure and a *corrective transformation* is required to transform the factors into a valid form.

As a concrete example, let us look at the low-rank factorization for the monocular basis shape formulation. However a similar reasoning applies for example also to the Kronecker structure of the second factor in the low-rank factorization of $\mathcal{W}_{(f)}$ in the multi-camera rigid and non-rigid case (Eq. (9) in the paper). The flattened data tensor $\mathcal{W}_{(n)}$ along the mode of the structure must have rank $3B+1$ as can be seen in Eq. (16) of the paper. Hence, factoring the flattened tensor $\mathcal{W}_{(n)}^T = \hat{\mathbf{M}}\hat{\mathbf{S}}$ (e.g. with the singular value decomposition) provides two matrix factors $\hat{\mathbf{M}} \in \mathbb{R}^{F\times 3B+1}$ and $\hat{\mathbf{S}} \in \mathbb{R}^{3B+1\times N}$ of the correct dimensions, missing however the correct algebraic Kronecker-structure. A corrective transformation $\mathbf{Q} \in \mathbb{R}^{3B+1\times 3B+1}$ is needed such that

$$\hat{\mathbf{M}}\mathbf{Q} = [\Downarrow_f [\mathbf{\Omega}_{f,:} \otimes \mathbf{R}_{f,1:2,:}, \mathbf{t}_{f,1:2}]] \quad \text{and} \quad \mathbf{Q}^{-1}\hat{\mathbf{S}} = \begin{bmatrix} \Downarrow_b \mathbf{S}_b \\ \mathbf{1}_{1\times N} \end{bmatrix}. \tag{14}$$

The two rows of the rotation matrix $\mathbf{R}_{f,1:2,:}$ should of course be orthonormal.

**Direct Approach** Abstracting from this specific problem instance, the general problem is given $\mathbf{A} \in \mathbb{R}^{mp\times nq}$ to find $\mathbf{Q} \in \mathbb{R}^{nq\times nq}$ such that

$$\mathbf{AQ} = \mathbf{B} \otimes \mathbf{C}, \tag{15}$$

[1] $\mathbf{A}^\perp$ denotes a matrix whose columns span is orthogonal to the column span of $\mathbf{A}$.

with $\mathbf{B} \in \mathbb{R}^{m \times n}$ and $\mathbf{C} \in \mathbb{R}^{p \times q}$ of which some of the row subsets $\mathcal{I}_i$ have to obey orthogonality constraints $\mathbf{C}_{\mathcal{I}_i,:}\mathbf{C}_{\mathcal{I}_j,:}^T = \alpha_{i,j}\mathbf{I}_p$ where the scaling factors $\alpha_{i,j}$ might be known depending on the specific problem. We remark that due to the Kronecker product property there are multiple solutions as

$$[\mathbf{B}\mathbf{Q}_B \otimes \mathbf{C}\mathbf{Q}_C] = [\mathbf{B} \otimes \mathbf{C}]\,[\mathbf{Q}_B \otimes \mathbf{Q}_C] = \mathbf{A}\mathbf{Q}\,[\mathbf{Q}_B \otimes \mathbf{Q}_C]$$

with orthogonal $\mathbf{Q}_C$ and therefore $\mathbf{Q}\,[\mathbf{Q}_B \otimes \mathbf{Q}_C]$ is also a valid solution. Hence the corrective solution is unique up to a right multiplication with $[\mathbf{Q}_B \otimes \mathbf{Q}_C]$.

We can make use of this ambiguity, however. The transformations $\mathbf{Q}_B$ and $\mathbf{Q}_C$ can be chosen arbitrarily, for example we might impose that $\mathbf{B}_{1:n,:}\mathbf{Q}_B = \mathbf{I}_n$ and $\mathbf{C}_{1:q,:}\mathbf{Q}_C = \mathbf{I}_q$ and hence $\mathbf{A}_{1:nq,:}\mathbf{Q} = \mathbf{I}_n \otimes \mathbf{I}_q$ are $nq \times nq$ linear equations in the $nq \times nq$ unknowns of $\mathbf{Q}$. Unfortunately, this approach can not be applied to the basis shape formulation, as only data for $\mathbf{\Omega}_{c,:} \otimes \mathbf{R}_f$ with $c = f$ is observed, the remaining bilinear terms for which $f \neq c$ are missing and therefore there is an insufficient number of equations. In such cases, other constraints must be considered such as the orthonormality constraints.

**Orthonormality Constraints** The *orthonormality constraints* make use of the fact that some tuples of row subsets of the factor $\mathbf{C}$ are constrained by orthogonality constraints: for each such tuple $(\mathcal{I}_i, \mathcal{I}_j)$, a block $\alpha_{i,j}\mathbf{I}_{|\mathcal{I}_i|}$ in the symmetric matrix $\mathbf{C}\mathbf{C}^T$ is known (maybe only up to the scaling $\alpha_{i,j}$). These blocks provide linear constraints for the unknown $\mathbf{Q}\mathbf{Q}^T$

$$\mathbf{A}\mathbf{Q}\mathbf{Q}^T\mathbf{A}^T = [\mathbf{B} \otimes \mathbf{C}]\,[\mathbf{B} \otimes \mathbf{C}]^T = \left[\mathbf{B}\mathbf{B}^T \otimes \mathbf{C}\mathbf{C}^T\right]. \tag{16}$$

Given enough observations, Eq. (16) results in an overdetermined linear system[2] in the $\frac{nq(nq+1)}{2}$ unknowns of the symmetric matrix $\mathbf{Q}\mathbf{Q}^T$. Once this linear system is solved, a final eigenvalue decomposition of $\mathbf{Q}\mathbf{Q}^T$ gives the solution $\mathbf{Q}$. Note that from the previous discussion we know that $\mathbf{Q}$ is unique only up to $[\mathbf{Q}_B \otimes \mathbf{Q}_C]$ which leads to

$$\mathbf{A}\mathbf{Q}\mathbf{Q}^T\mathbf{A}^T = \mathbf{A}\mathbf{Q}\,[\mathbf{Q}_B \otimes \mathbf{Q}_C]\,[\mathbf{Q}_B \otimes \mathbf{Q}_C]^T\,\mathbf{Q}^T\mathbf{A}^T \tag{17}$$

$$= \mathbf{A}\mathbf{Q}\left[\mathbf{Q}_B\mathbf{Q}_B^T \otimes \mathbf{I}_q\right]\mathbf{Q}^T\mathbf{A}^T, \tag{18}$$

and therefore the solution to the orthonormality constraints in Eq. (16) is unique up to $\left[\mathbf{Q}_B\mathbf{Q}_B^T \otimes \mathbf{I}_q\right]$, which has been called a *block-scaled identity matrix* by Xiao et.al.[5]. Each block $(i,j)$ consists of a identity matrix of dimension $q \times q$ scaled by an unknown factor $\mathbf{Q}_{B,i,:}\mathbf{Q}_{B,j,:}^T$. Block-scaled identiy matrices have $\frac{n(n+1)}{2}$ degrees of freedom. Therefore there is a $\frac{n(n+1)}{2}$ dimensional solution space for the orthonormality constraints and $\frac{nq(nq+1)}{2} - \frac{n(n+1)}{2}$ linearly independent equations are necessary. The immediate question is: under which circumstances do the orthonormality constraints result in linearly independent equations?

[2] Alternatively, the problem can also be formulated as a positive-semidefinite programming (SDP) problem. In our experiments, this sometimes proved to be more robust than the linear least squares approach.

**The Whole Story** Let us present a new class of matrices, the so called *block-skew-symmetric matrices*. These matrices have also been introduced in [5] and are defined as symmetric matrices $\mathbf{Y} = [\Downarrow_{i=1,\cdots,n}\Rightarrow_{j=1,\cdots,n} \mathbf{Y}_{i,j}] \in \mathbb{R}^{nq\times nq}$ whose $\mathbf{Y}_{i,j} \in \mathbb{R}^{q\times q}$ subblocks are skew-symmetric. This definition implies $\mathbf{Y}_{i,i} = \mathbf{0}_{q\times q}$ and since $\mathbf{Y}$ is symmetric, it holds that $\mathbf{Y}_{i,j} = \mathbf{Y}_{j,i}^T = -\mathbf{Y}_{j,i}$. Block-skew-symmetric matrices have $\frac{q(q-1)}{2}\frac{n(n-1)}{2}$ degrees of freedom (the diagonal blocks are all zero, hence there are $\frac{1}{2}n(n-1)$ non-zero blocks with $\frac{q(q-1)}{2}$ degrees of freedom each). Block-skew-symmetric matrices $\mathbf{Y} \in \mathbb{R}^{nq\times nq}$ have a nasty property. Consider the quadrilinear form in $\mathbf{a}, \mathbf{c} \in \mathbb{R}^n$ and in $\mathbf{b}, \mathbf{d} \in \mathbb{R}^q$ given by the block-skew-symmetric matrix $\mathbf{Y}$

$$[\mathbf{a}\otimes\mathbf{b}]\,\mathbf{Y}\,[\mathbf{c}\otimes\mathbf{d}]\,.$$

This quadrilinear form equals zero if $\mathbf{a} = \mathbf{c}$ or if $\mathbf{b} = \mathbf{d}$. Hence, we get for *any* block-skew symmetric $\mathbf{Y} \in \mathbb{R}^{nq\times nq}$

$$0 = [\mathbf{B}_{i,:}\otimes\mathbf{C}_{j,:}]\,\mathbf{Y}\,[\mathbf{B}_{k,:}\otimes\mathbf{C}_{l,:}]^T\,, \tag{19}$$

whenever $i = k$ or $j = l$. If there are only constraints for which $i = k$ or $j = l$ then the general solution to the orthonormality constraints is $\mathbf{Q}\left[\mathbf{Y} + \mathbf{Q}_B\mathbf{Q}_B^T \otimes \mathbf{I}_q\right]\mathbf{Q}^T$ with $\frac{n(n+1)}{2} + \frac{q(q-1)}{2}\frac{n(n-1)}{2}$ linear degrees of freedom. Even more importantly, block-skew matrices are indefinite and hence the solution space also contains indefinite solutions which can not be factored into real valued corrective transformation $\mathbf{Q}$. Semi-definite programming (SDP) could be used to find a valid positive definite solution in this solution space which then in turn can be factored into $\mathbf{Q}$ and $\mathbf{Q}^T$. However, SDP does not force the block-skew symmetric matrix $\mathbf{Y}$ to the zero matrix, but rather makes sure that the guaranteed positive definite component $\mathbf{Q}_B\mathbf{Q}_B^T\otimes\mathbf{I}_q$ outweights the indefinite component $\mathbf{Y}$. The indefinite part $\mathbf{Y}$ still spoils the Kronecker-product structure of the solution to the orthonormality constraints.

There are multiple ways to resolve this problem:

i. **Rank-Constraint:** Only consider a slice of $q$ consecutive columns, as proposed in [4]: This results in a non-linear rank-$q$ constraint on $\mathbf{Q}\mathbf{Q}^T$. Formally, let us introduce the matrix $\mathbf{0}_{+r} \in \mathbb{R}^{nq\times nq}$ which is equal to the zero matrix except with $\mathbf{I}_q$ on its $r^{\text{th}}$ diagonal $q\times q$ block. This matrix basically slices $q$ consecutive columns (or rows) out of $\mathbf{Q}$. Then we are looking for a solution to the orthonormality constraints of the form

$$\mathbf{Q}\mathbf{0}_{+r}\left[\mathbf{Y} + \mathbf{Q}_B\mathbf{Q}_B^T\otimes\mathbf{I}_q\right]\mathbf{0}_{+r}^T\mathbf{Q}^T \tag{20}$$

$$=\mathbf{Q}_{:,(r-1)q+1:rq}\left[\mathbf{Y}_{r,r} + \mathbf{Q}_{B,:,r}\mathbf{Q}_{B,:,r}^T\otimes\mathbf{I}_q\right]\mathbf{Q}_{:,(r-1)q+1:rq}^T. \tag{21}$$

Because the diagonal block $\mathbf{Y}_{r,r}$ of the block skew-symmetric matrix is necessarily zero the problem has been resolved. There is a irrelevant ambiguity of a Kronecker-product between a symmetric rank-1 matrix and $\mathbf{I}_q$. This again corresponds to the arbitrary choice of basis for the columns of matrix $\mathbf{B}$. The drawback of this rank constraint is that we have to resort to non-linear iterative optimization methods which might get stuck in a local minima.

ii. **Mixed Constraints:** As previously mentioned, mixed constraints for pairwise linear independent vectors $\mathbf{a}$ and $\mathbf{c}$ resp. $\mathbf{b}$ and $\mathbf{d}$ in Eq. (19) help to resolve the problem. Hence, observations $0 = [\mathbf{B}_{i,:} \otimes \mathbf{C}_{j,:}]\,\mathbf{Y}\,[\mathbf{B}_{k,:} \otimes \mathbf{C}_{l,:}]$ with $i \neq k$ *and* $j \neq l$ are necessary. However, most often such mixed constraints are not available. And even if, is is not easy to spot that such constraints are actually contained in the availabel observations. For example in rigid and non-rigid multi-camera factorization, such constraints are indeed present since multiple points are usually tracked in a camera: $\mathbf{B}_{i,:}$ and $\mathbf{B}_{k,:}$ correspond to the coordinates of points $i$ and $k$ whereas $\mathbf{C}_{j,:}$ and $\mathbf{C}_{l,:}$ correspond to the x- and y-axes of the same camera. Unfortunately, a large number of points need to be tracked in order to get sufficiently many linear equations which renders this approach less practically useful.

iii. **Basis Constraints:** Due to the basis ambuiguity, we can choose $\mathbf{B} = \begin{bmatrix} \mathbf{I}_n \\ \mathbf{D} \end{bmatrix}$. The orthonormality constraints then look like

$$\mathbf{A}\mathbf{Q}\mathbf{Q}^T\mathbf{A}^T = \begin{bmatrix} \mathbf{I}_n \\ \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{I}_n & \mathbf{D}^T \end{bmatrix} \otimes \mathbf{C}\mathbf{C}^T = \begin{bmatrix} \mathbf{I}_n & \mathbf{D}^T \\ \mathbf{D} & \mathbf{D}\mathbf{D}^T \end{bmatrix} \otimes \mathbf{C}\mathbf{C}^T. \tag{22}$$

This specific choice of basis zeroed out many entries in $\mathbf{B}\mathbf{B}^T \otimes \mathbf{C}\mathbf{C}^T$. Especially, some potentially previously unknown entries $[\mathbf{B}_{i,:} \otimes \mathbf{C}_{j,:}]\,[\mathbf{B}_{k,:} \otimes \mathbf{C}_{l,:}]^T = \mathbf{B}_{i,:}\mathbf{B}_{k,:}^T \otimes \mathbf{C}_{j,:}\mathbf{C}_{l,:}^T$ with $i \neq k$ or $j \neq l$ are forced to zero and hence, now there are mixed constraints which resolve the block skew-symmetric matrix ambiguity. Adding these constraints results in a *unique* full rank solution for $\mathbf{Q}\mathbf{Q}^T$: the block-skew symmetric matrix $\mathbf{Y}$ must be zero and the ambiguity $\mathbf{Q}_B\mathbf{Q}_B^T \otimes \mathbf{I}_q$ has been fixed through our choice of basis. Unfortunately, extracting the corrective transformation $\mathbf{Q}$ from $\mathbf{Q}\mathbf{Q}^T$ proofes to be difficult: an arbitrary decomposition, like for example the eigendecomposition $\mathbf{Q}\mathbf{Q}^T = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ and setting $\mathbf{Q} = \mathbf{V}\mathbf{\Lambda}^{\frac{1}{2}}$, will destroy the Kronecker-structure in $\mathbf{A}\mathbf{Q} = \mathbf{B} \otimes \mathbf{C}$. It is worth to point out again, even though the solution provided by the basis constraints is unique, the extraction of the corrective transformation poses difficulties.

iv. **Basis Constraints with Rank-Constraint:** A closed-form solution can be obtained by combining the basis constraints with the rank constraint [5]. We again choose $\mathbf{B} = \begin{bmatrix} \mathbf{I}_n \\ \mathbf{D} \end{bmatrix}$, but this time only a slice of $q$ consecutive columns are extracted. Using the same notation as before, we get:

$$\mathbf{A}\mathbf{Q}\mathbf{0}_{+r}\mathbf{0}_{+r}^T\mathbf{Q}^T\mathbf{A}^T = \left[\begin{bmatrix} \mathbf{I}_n \\ \mathbf{D} \end{bmatrix} \otimes \mathbf{C}\right] \mathbf{0}_{+r}\mathbf{0}_{+r}^T \left[\begin{bmatrix} \mathbf{I}_n \\ \mathbf{D} \end{bmatrix} \otimes \mathbf{C}\right]^T \tag{23}$$

$$= \left[\begin{pmatrix} \mathbf{e}_r \\ \mathbf{D}_{:,r} \end{pmatrix} \otimes \mathbf{C}\right] \left[\begin{pmatrix} \mathbf{e}_r \\ \mathbf{D}_{:,r} \end{pmatrix} \otimes \mathbf{C}\right]^T = \left[\begin{pmatrix} \mathbf{e}_r \\ \mathbf{D}_{:,r} \end{pmatrix} \begin{pmatrix} \mathbf{e}_r \\ \mathbf{D}_{:,r} \end{pmatrix}^T \otimes \mathbf{C}\mathbf{C}^T\right] \tag{24}$$

$$= \begin{bmatrix} \begin{bmatrix} \mathbf{0}_{r-1\times r-1} & \mathbf{0}_{r-1\times 1} & \mathbf{0}_{r-1\times n-r} \\ \mathbf{0}_{1\times r-1} & 1_{1\times 1} & \mathbf{0}_{1\times n-r} \\ \mathbf{0}_{n-r\times r-1} & \mathbf{0}_{n-r\times 1} & \mathbf{0}_{n-r\times n-r} \end{bmatrix} & \begin{bmatrix} \mathbf{0}_{r-1\times m-n} \\ \mathbf{D}_{:,r}^T \\ \mathbf{0}_{n-r\times m-n} \end{bmatrix} \\ \begin{bmatrix} \mathbf{0}_{m-n\times r-1} & \mathbf{D}_{:,r} & \mathbf{0}_{m-n\times n-r} \end{bmatrix} & \begin{bmatrix} \mathbf{D}_{:,r}\mathbf{D}_{:,r}^T \end{bmatrix} \end{bmatrix} \otimes \mathbf{C}\mathbf{C}^T \quad (25)$$

In comparison with Eq. (22), these low-rank basis constraints drive even more entries to zero. These constraints provide a unique rank-$q$ solution $\mathbf{Q}\mathbf{Q}^T$ from which a slice from the corrective transformation $\mathbf{Q}_r$ can be extracted (with the above mentioned eigendecomposition approach, for example). This then in turn allows to solve for $\mathbf{B}$ and $\mathbf{C}$ in $\mathbf{A}\mathbf{Q}_r = \mathbf{B}_{:,r} \otimes \mathbf{C}$.

## 2 Closed-Form Solution

The following subsections provide additional details about the closed-form algorithm for NRSfM factorizations without correspondences between different cameras. As mentioned in the paper, we follow along similar lines as in [6]. More specifically, after an initial low-rank factorization, a three step stratified approach will be used where the first step singles out the camera translations (and thus corresponds to an affine upgrade in standard multiple-view geometry), the second step computes the camera matrices, and the last step computes the structure. Optionally, a fourth step can subsequently be applied to upgrade the resulting reconstruction w.r.t. an affine coordinate frame to a similarity coordinate frame.

### 2.1 Low-Rank Factorization

The initial factorization is based on the flattened data tensor $\mathcal{W}_{(f)}$ along the temporal mode, see Eq. (9) in the paper. However, due to the non-existence of correspondences between different cameras, many columns of $\mathcal{W}_{(f)}$ will be unknwon: if point $n$ is not observed in camera $k$, then the corresponding columns in $\mathcal{W}_{(f)}$ will be missing. Let us denote with $\mathbf{S}_k \in \mathbb{R}^{d_S+1\times N_k}$ the points tracked by camera $k$ and with $\mathbf{C}_k \in \mathbb{R}^{2\times 4}$ the camera matrix of camera k. Then the known columns of $\mathcal{W}_{(f)}$ can be stacked in a matrix $\mathbf{W}$ which leads to

$$\mathbf{W} = \mathbf{M}\mathcal{S}_{(f)} \left[\Rightarrow_k \mathbf{S}_k \otimes \mathbf{C}_k^T\right] \in \mathbb{R}^{F\times 2\sum_k N_k}, \quad (26)$$

which shows that even though no correspondences between the cameras are known, the motion matrix $\mathbf{M}$ is nonetheless completely defined by the given observations. A low-rank factorization of $\mathbf{W} = \hat{\mathbf{M}}\hat{\mathbf{A}}$ into rank-$(3d_S + 4)$ matrices $\hat{\mathbf{M}}$ and $\hat{\mathbf{A}}$ yields the correct column span $\text{span}\left(\hat{\mathbf{M}}\right) = \text{span}\left(\mathbf{M}\right)$ of the motion matrix. An arbitrary low-rank factorization, like e.g. given by a low-rank singular value decomposition, will not comply with the required Kronecker-structure in the second matrix factor as revealed in Eq. (9) in the paper. Hence, a corrective transformation $\mathbf{Q} \in \mathbb{R}^{3d_S+4\times 3d_S+4}$ is necessary to establish the correct structure in $\mathbf{M} = \hat{\mathbf{M}}\mathbf{Q}$ and in $\mathbf{Q}^{-1}\hat{\mathbf{A}} = \mathcal{S}_{(f)}\left[\Rightarrow_k \mathbf{S}_k \otimes \mathbf{C}_k^T\right]$. As mentioned previously, this corrective transformation will be computed in a stratified way.

### 2.2 Singling out the Camera Translations

Eq. (10) in the paper shows that the last column of the motion matrix equals the constant one vector. This is enforced by solving the overconstrained system $\hat{\mathbf{M}}\mathbf{q} = \mathbf{1}_{F\times 1}$ in the least-squares sense for $\mathbf{q} \in \mathbb{R}^{3d_S+4}$. Proceeding by defining $\mathbf{Q}_1 = \left[[\mathbf{q}]_\perp, \mathbf{q}\right]$ we get $\tilde{\mathbf{M}} = \hat{\mathbf{M}}\mathbf{Q}_1$ and $\tilde{\mathbf{A}} = \mathbf{Q}_1^{-1}\hat{\mathbf{A}}$. Note that since the last column of $\tilde{\mathbf{M}}$ should no longer get modified, any further corrective transformation should not change the last column of $\mathbf{Q}_1$. This implies that for any further corrective transformation $\mathbf{Q}$, the last column of $\mathbf{Q}$ and $\mathbf{Q}^{-1}$ must equal $\mathbf{e}_{3d_S+4} = (\mathbf{0}_{3d_S+3}, 1)^T$.

### 2.3 Extracting Camera Matrix

Combining the structure of the flattened core tensor $\mathcal{S}_{(f)}$ as shown in Eq. (8) in the paper with the homogeneous coordinate of the structure (see Eq. (11) in the paper) shows that the last four rows of $\mathcal{S}_{(f)}\left[\mathbf{S}_k \otimes \mathbf{C}_k^T\right]$ equals a replication $\mathbf{1}_{1\times N_K} \otimes \mathbf{C}_k^T$ of the camera matrix $\mathbf{C}^T$. This observation yields an overconstrained linear system for the camera matrices $\mathbf{C}_k$ and the partial corrective transformation $\mathbf{Q}_2 \in \mathbb{R}^{4\times 3d_S+4}$

$$\mathbf{Q}_2\tilde{\mathbf{A}} = \left[\Rightarrow_k \mathbf{1}_{1\times N_k} \otimes \mathbf{C}_k^T\right]. \tag{27}$$

Note that due to the observation mentioned at the end of the previous subsection, the last column of $\mathbf{Q}_2$ must equal $(0,0,0,1)^T$. After solving this linear system, the camera matrices $\mathbf{C}_k$ are all known w.r.t. a consistent affine coordinate reference frame. Note that due to the gauge freedoms, the resulting system matrix for $\operatorname{vec}(\mathbf{Q}_2)$ and $(\Downarrow_k \operatorname{vec}(\mathbf{C}_k))$ will be rank deficient and will have a 3 dimensional nullspace (the camera translation has already been singled out and hence the nullspace is only 3 dimensional rather than 4 dimensional). This provides three different homogeneous solutions and one inhomogeneous solution (corresponding to last row of $\mathbf{Q}_2$) for the four columns of the camera matrices.

### 2.4 Computing the Structure Matrix

Once the camera matrices are known, the original bilinear problem $\mathbf{Q}_3\tilde{\mathbf{A}} = \mathcal{S}_{(f)}\left[\Rightarrow_k \mathbf{S}_k \otimes \mathbf{C}_k^T\right]$ reduces to a linear problem in $\mathbf{Q}_3$ and in $\mathbf{S}_k$. This overconstrained system can again be solved in the least-squares sense. Note that due to the gauge freedoms, the resulting system matrix for $\operatorname{vec}(\mathbf{Q}_3)$ and $(\Downarrow_k \operatorname{vec}(\mathbf{S}_k))$ will be rank deficient and will have a $d_S + 1$ dimensional nullspace. From this nullspace, $d_S + 1$ linearly independent solutions can be extracted corresponding to the $d_S + 1$ coordinates of the points.

Applying the inverse of this corrective transformation to the motion matrix leads to the corrected motion matrix $\mathbf{M} = \tilde{\mathbf{M}}\mathbf{Q}_3^{-1}$.

### 2.5 Metric Upgrade

The previous steps give a reconstruction w.r.t. an affine coordinate frame: the camera matrices will not have orthogonal axes as is usually enforced in factorization methods. In [6], a similarity upgrade was computed by enforcing rigid motions in the motion matrix: the first 9 entries of each row of $\mathbf{M}$ must equal a vectorized rotation matrix. In the NRSfM case however, this is no longer the case (see Eq. (10) in the paper).

We propose to compute a similarity upgrade based on auto-calibration techniques. The plane at infinity is already known, since the camera matrices are computed w.r.t. an affine reference frame. As mentioned in Sec. 19.5 of [7], the computation of the plane at infitiny is usually the most difficult part of auto-calibration. Using constraints on the internal camera calibration matrices, the affine reconstruction can be upgraded to a similarity reconstruction. We refer to Sec. 19.5 in [7] for details on auto-calibration with known plane at infinity. As an example, assuming square pixels, three cameras are sufficient to compute this upgrade. Note that in the non-rigid case with general low-rank assumption on the trajectories, two cameras can not be sufficient due to the inherent bas-relief ambiguity between two affine cameras.

## 3 Alternating Least Squares: Jacobian Matrices

This section presents the Jacobian matrices required for an Alternating Least Squares (ALS) approach in order to minimize the objective function

$$\min_{\mathbf{M},\mathbf{S},\mathbf{C}} \frac{1}{2} \left\| \mathbf{H} \odot \left[ \mathbf{M} \mathcal{S}_{(f)} \left[ \mathbf{S} \otimes \mathbf{C}^T \right] - \mathcal{W}_{(f)} \right] \right\|_F^2 . \tag{28}$$

This matrix valued least-squares problem can be formulated in a standard vector-valued least squares problem by vectorizing the residual matrix e.g. columnwise. This choice corresponds to a vectorization of the data tensor $\mathcal{W}$ along all three modes. The unknown entries, for which $\mathbf{H}_{i,j} = 0$, can be discarded from the residual vector $\mathbf{r} \in \mathbb{R}^{2FKN\times 1}$ by a left-multiplication $\mathbb{P}_{\mathbf{H}}\mathbf{r}$ with a row-amputed identity matrix $\mathbb{P}_{\mathbf{H}}$ slicing out the known entries from the residual vector. ALS proceeds by alternatingly solving a least-squares problem in each variable while holding the remaining ones fixed. Let $\mathbf{J}_{\mathbf{X}}$ denote the Jacobian of the objective function w.r.t. variable $\mathbf{X}$ and $\mathbf{b}_{\mathbf{X}}$ the corresponding right-hand side of the least-squares system. In each substep of ALS, a system of the form $\min_{\mathbf{x}} \|\mathbf{J}_{\mathbf{X}}\mathbf{x} - \mathbf{b}_{\mathbf{X}}\|_2^2$ with $\mathbf{x} = \mathrm{vec}\left(\mathbf{X}\right)$ must be solved. Such a least-squares problem can for example efficienlty be solved with a QR-decomposition of $\mathbf{J}_{\mathbf{X}}$.

In order to compute the Jacobians, let us introduce

$$\mathbf{M} = \left[\tilde{\mathbf{M}}, \mathbf{T}, \mathbf{1}_{F\times 1}\right] \in \mathbb{R}^{F\times 3d_S+4} \text{ and } \mathbf{S} = \begin{bmatrix} \tilde{\mathbf{S}} \\ \mathbf{1}_{1\times N} \end{bmatrix} \in \mathbb{R}^{d_S+1\times N} \text{ and } \mathbf{C} = [\mathbf{P}, \mathbf{t}] \in \mathbb{R}^{2K\times 4}, \tag{29}$$

where $\tilde{\mathbf{M}} \in \mathbb{R}^{F \times 3d_S}$, $\mathbf{T} \in \mathbb{R}^{F \times 3}$, $\tilde{\mathbf{S}} \in \mathbb{R}^{3d_S \times N}$, $\mathbf{P} \in \mathbb{R}^{2K \times 3}$, and $\mathbf{t} \in \mathbb{R}^{2K \times 1}$. With this notation in place, the objective function can be formulated in a refined way

$$\min_{\mathbf{M},\mathbf{S},\mathbf{C}} \frac{1}{2} \left\| \tilde{\mathbf{M}} \left[ \tilde{\mathbf{S}} \otimes \mathbf{P}^T \right] + \mathbf{T} \left[ \mathbf{1}_{1 \times N} \otimes \mathbf{P}^T \right] + \left[ \mathbf{1}_{F \times N} \otimes \mathbf{t}^T \right] - \mathcal{W}_{(f)} \right\|_F^2 . \tag{30}$$

Computing Jacobian matrices of matrix valued functions w.r.t. matrix valued variables can be a tricky task. We suggest following the rules presented by Magnus and Neudecker in [8]. This results in the following Jacobians:

$$\mathbf{J}_{\tilde{\mathbf{M}}} = \mathbb{P}_{\mathbf{H}} \left[ \tilde{\mathbf{S}}^T \otimes \mathbf{P} \otimes \mathbf{I}_F \right] \tag{31}$$
$$\mathbf{J}_{\mathbf{T}} = \mathbb{P}_{\mathbf{H}} \left[ \mathbf{1}_{N \times 1} \otimes \mathbf{P} \otimes \mathbf{I}_F \right] \tag{32}$$
$$\mathbf{J}_{\tilde{\mathbf{S}}} = \mathbb{P}_{\mathbf{H}} \left[ \mathbf{I}_{2KN} \otimes \tilde{\mathbf{M}} \right] \left[ \mathbf{I}_N \otimes \mathbf{K}_{2K,d_S} \otimes \mathbf{I}_3 \right] \left[ \mathbf{I}_{d_S N} \otimes \operatorname{vec}\left( \mathbf{P}^T \right) \right] \tag{33}$$
$$\mathbf{J}_{\mathbf{P}^T} = \mathbb{P}_{\mathbf{H}} \left[ \mathbf{I}_{2KN} \otimes \tilde{\mathbf{M}} \right] \left[ \mathbf{I}_N \otimes \mathbf{K}_{2K,d_S} \otimes \mathbf{I}_3 \right] \left[ \operatorname{vec}\left( \tilde{\mathbf{S}} \right) \otimes \mathbf{I}_{3 \cdot 2K} \right] \tag{34}$$
$$\mathbf{J}_{\mathbf{t}^T} = \mathbb{P}_{\mathbf{H}} \left[ \mathbf{I}_N \otimes \mathbf{K}_{2K,F} \otimes \mathbf{I}_1 \right] \left[ \operatorname{vec}\left( \mathbf{1}_{F \times N} \right) \otimes \mathbf{I}_{2K} \right] \tag{35}$$

The matrix $\mathbf{K}_{m,n}$ is the so-called commutator matrix which is defined by the property $\operatorname{vec}\left(\mathbf{A}^T\right) = \mathbf{K}_{m,n} \operatorname{vec}(\mathbf{A})$ for any $\mathbf{A} \in \mathbb{R}^{m \times n}$. The formulas above can be slightly simplified which we omit here. Note that the resulting matrices are highly sparse and it is important to use sparse matrices for the computations. The corresponding right-hand sides for the least-squares system are straightforward to compute and we omit explicit formulas here.

## 4 Reinterpreting [2] in Tensor Notation

In the following, we restate the approach of [2] using our tensor notation. Let us summarize again the main steps of [2]:

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

The first step can be considered as a preprocessing step and is not explained further here. The second step starts by exposing the trajectory subspace by flattening the data tensor along the temporal mode

$$\mathcal{W}_{(f)} = \mathbf{M} \mathcal{S}_{(f)} \left[ \mathbf{S} \otimes \mathbf{C}^T \right] = \mathbf{M} \mathcal{S}_{(f)} \left[ \mathbf{S} \otimes \mathbf{I}_4 \right] \left[ \mathbf{I}_N \otimes \mathbf{C}^T \right], \tag{36}$$

where the reformulation in the last equation is not strictly necessary but shows the exact equivalence between Eq. (36) and Eq. (9) in [2]. The trajectory subspace can be computed from the column span of $\mathbf{M}$ which is equal to span $\left(\mathcal{W}_{(f)}\right)$. Let $\mathbf{\Omega} \in \mathbb{R}^{F \times d_M}$ be a matrix with orthonormal columns spanning this subspace.

The final step in [2] is the computation of the camera matrices. In the tensor framework, this is done by flattening the data tensor along the camera mode

$$\mathcal{W}_{(k)} = \mathbf{C}\mathcal{S}_{(k)} \left[\mathbf{S} \otimes \mathbf{M}^T\right], \tag{37}$$

where the last equation corresponds to Eq. (2) in [2]. [2] then continues by a right-multiplication

$$\mathcal{W}_{(k)} \left[\mathbf{I}_N \otimes \mathbf{\Omega}\right] = \mathbf{C}\mathcal{S}_{(k)} \left[\mathbf{S} \otimes \mathbf{M}^T\mathbf{\Omega}\right] = \mathbf{C} \underbrace{\mathcal{S}_{(k)} \left[\mathbf{S} \otimes \left[\mathbf{I}_{d_S}, \mathbf{0}_{d\times d_S}\right]^T\right]}_{=\mathbf{A}\in\mathbb{R}^{4\times N d_S}}, \tag{38}$$

where in the last equation we have assumed w.l.o.g. that the motion matrix $\mathbf{M}$ equals $[\mathbf{\Omega}, \mathbf{\Omega}_\perp]$ for some $\mathbf{\Omega}_\perp \in \mathbb{R}^{F\times F-d_M}$ (since $\mathbf{M}$ can always be orthogonalized by absorbing the required factor in $\mathcal{S}_{(f)}$). Hence, this step amounts to projecting the motion onto a low-rank subspace of dimension $d_M$ spanned by $\mathbf{\Omega}$. For exact low-rank and noise-free data (i.e. then $\mathbf{\Omega}_\perp = \mathbf{0}$), this projection step is not necessary and the same result is given by factorizing the data according to Eq. (37). In such a situation, [2] boils down to i) fitting known entries in the least-squares sense with $\mathbf{\Omega}$ in order to estimate missing entries and ii) factorizing $\mathcal{W}_{(k)}$.

Note that if there are no point correspondences between different cameras, then the pattern of known entries in $\mathcal{W}_{(k)}$ follows a block-diagonal matrix and Eq. (37) can not solely be used to align all the cameras in a consistent coordinate frame. That is the reason why [2] requires all feature point correspondences to be known. Furthermore by comparing Eq. (37) and Eq. (36) with eq. (2) and eq. (9) in [2], we see that [2] misses the exact algebraic interplay between $\mathbf{C}$, $\mathbf{M}$, and $\mathbf{S}$ (resp. $\mathbf{R}$, $\mathbf{\Theta}$, and $\mathbf{A}$ in their formulation) which provides valuable constraints. As described in the paper, by using these constraints it is possible to compute a valid reconstruction in closed-form even if there are no correspondences between different cameras.

## References


1. Angst, R., Pollefeys, M.: A unified view on deformable shape factorizations. In: ECCV, Berlin, Heidelberg, Springer-Verlag (2012)
2. Zaheer, A., Akhter, I., Baig, M.H., Marzban, S., Khan, S.: Multiview structure from motion in trajectory space. In: Proc. ICCV. (2011) 2447–2453
3. Park, H.S., Shiratori, T., Matthews, I., Sheikh, Y.: 3d reconstruction of a moving point from a series of 2d projections. In: Proc. ECCV. (2010) 158–171
4. Brand, M.: A direct method for 3d factorization of nonrigid motion observed in 2d. In: CVPR (2), IEEE Computer Society (2005) 122–128
5. Xiao, J., Chai, J., Kanade, T.: A closed-form solution to non-rigid shape and motion recovery. In: Proc. ECCV. (2004) 573–587
6. Angst, R., Pollefeys, M.: Static multi-camera factorization using rigid motion. In: Proc. ICCV. (2009) 1203 –1210
7. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Second edn. Cambridge University Press, ISBN: 0521540518 (2004)
8. Magnus, J.R., Neudecker, H.: Matrix differential calculus with applications in statistics and econometrics. 2nd edn. John Wiley & Sons (1999)