Supplemental Material

Roland Angst and Marc Pollefeys
Computer Vision and Geometry Lab, Department of Computer Science
ETH Zürich, Switzerland
{rangst, marc.pollefeys}@inf.ethz.ch

Abstract

This technical report provides additional information and explanations to the ICCV09 submission [1]. The derivations make use of the same notation as in the paper. Sec. 1 and Sec. 2 give a short introduction to the required concepts of tensor and matrix calculus. Details on how to set up and solve the linear system for the computation of the affine camera matrix and affine coordinates of the points are given in Sec. 3 and Sec. 4, respectively.

1. Tensors

Tensors express multilinear relationships between variables and are thus a generalization of entities used in linear algebra, i.e., vectors (1st-order tensors) and matrices (2nd-order tensors). A tensor of order n can be thought of as a n-dimensional array of numbers. Varying the i^{th} index of a tensor while keeping the remaining indices fixed defines the mode-i vectors. An important tool for the analysis and usage of tensors is the mode-i product. The mode-i product $\mathcal{B} = \mathcal{A} \times_i \mathbf{M}$ is a tensor-valued bivariate function of a n^{th}-order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times_n I_n}$ and a J_i-by-I_i matrix $\mathbf{M}$. The resulting tensor $\mathcal{B}$ is given by left-multiplying all the mode-i vectors by the matrix $\mathbf{M}$. The tensor $\mathcal{B}$ is still of order n, the dimension of the mode-i vectors however changed from I_i to J_i. An efficient and easy way to compute such a mode-i product is to flatten the tensor $\mathcal{A}$ along its i^{th}-mode (which means stacking all the mode-i vectors of $\mathcal{A}$ into one big matrix $\mathcal{A}_{(i)} \in \mathbb{R}^{I_i \times \prod_{j \neq i}^{n} I_j}$) and to left-multiply by the matrix $\mathbf{M}$. This provides the resulting flattened version of $\mathcal{B}_{(i)} = \mathbf{M}\mathcal{A}_{(i)}$. A straight forward reordering of the elements of this flattened tensor leads to the tensor $\mathcal{B}$. Note that the order in which the mode-i vectors are put next to each other is unimportant as long as the reshaping of $\mathcal{B}_{(i)}$ into a tensor $\mathcal{B}$ is performed consistently.

Given the measured data can be modeled as a tensor, various algorithms exist to analyze the underlying algebraic structure of the process which generated the measured data and also to decompose the data tensor into more meaningful parts. The most prominent two tensor decompositions are the canonical decomposition (aka. parallel factor model) [2, 3] and the Tucker decomposition [5]. A more recent decomposition [4], the higher order singular value decomposition, extends the Tucker decomposition by imposing certain orthogonality constraints. However, the Tucker decomposition without orthogonality constraints is the most suitable tensor decomposition for our purposes because it reveals the underlying mode-i subspaces without imposing unnecessary orthogonality constraints on them. The mode-i subspace is the span of all the mode-i vectors of a tensor. The Tucker decomposition of a n^{th}-order tensor $\mathcal{A}$ expresses the tensor as n mode-i products between a smaller core tensor $\mathcal{S} \in \mathbb{R}^{r_1 \times \cdots r_n}$ and n matrices $\mathbf{M}_i$

$$\mathcal{A} = \mathcal{S} \times_1 \mathbf{M}_1 \times_2 \mathbf{M}_2 \times_3 \cdots \times_n \mathbf{M}_n, \tag{1}$$

where the columns of $\mathbf{M}_i$ represent a basis for the mode-i subspace.

2. The Kronecker Product and the $\operatorname{vec}()$-operator

The Kronecker product $\otimes$ is a specific case of a tensor product. It is a matrix-valued bilinear product of two matrices and generalizes the bilinear outer product of vectors $\mathbf{ab}^T$ to matrices. Throughout this section, let $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{p \times q}$. Then $\mathbf{A} \otimes \mathbf{B} \in \mathbb{R}^{mp \times nq}$ is a block structured matrix where the $(i,j)^{\text{th}}$ block equals the matrix $\mathbf{B}$ scaled by the $(i,j)^{\text{th}}$ element of $\mathbf{A}$. This implies for example that the first column equals the vectorized outer product $\operatorname{vec}\left(\mathbf{B}_{:,1}\mathbf{A}_{:,1}^T\right)$ of the first two columns of $\mathbf{A}$ and $\mathbf{B}$. The Kronecker product is helpful in rewriting matrix equations of the form $\mathbf{AXB}^T = \mathbf{C}$ which is equivalent to

$$\operatorname{vec}(\mathbf{C}) = \operatorname{vec}\left(\mathbf{AXB}^T\right) = (\mathbf{B} \otimes \mathbf{A}) \operatorname{vec}(\mathbf{X}). \tag{2}$$

Here, the vectorization operator $\operatorname{vec}(\mathbf{A})$ has been used which is usually defined in matrix calculus as the vector which results by stacking all the columns of matrix $\mathbf{A}$ into a column vector. If the number of rows and columns of the matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ are such that $\mathbf{AC}$ and $\mathbf{BD}$ can be formed, then the mixed-product property of the Kronecker product states that

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD}). \tag{3}$$

We define a permutation matrix $\mathbf{T}_{m,n} \in \mathbb{R}^{mn \times mn}$ such that $\text{vec}\left(\mathbf{A}^T\right) = \mathbf{T}_{m,n} \text{vec}\left(\mathbf{A}\right)$. Furthermore, the following identities will be used in Sec. 3 and Sec. 4:

$$\text{vec}\left(\mathbf{A} \otimes \mathbf{B}\right) = \left(\mathbf{I}_n \otimes \mathbf{T}_{q,m} \otimes \mathbf{I}_p\right)\left(\text{vec}\left(\mathbf{A}\right) \otimes \text{vec}\left(\mathbf{B}\right)\right) \tag{4}$$

$$= \left(\mathbf{I}_n \otimes \mathbf{T}_{q,m} \otimes \mathbf{I}_p\right)\left(\mathbf{I}_{mn} \otimes \text{vec}\left(\mathbf{B}\right)\right)\text{vec}\left(\mathbf{A}\right) \tag{5}$$

$$= \left(\mathbf{I}_n \otimes \mathbf{T}_{q,m} \otimes \mathbf{I}_p\right)\left(\text{vec}\left(\mathbf{A}\right) \otimes \mathbf{I}_{pq}\right)\text{vec}\left(\mathbf{B}\right) \tag{6}$$

The Kronecker product provides a link between the Tucker decomposition and ordinary matrix multiplication. Specifically, given a Tucker decomposition $\mathcal{A} = \mathcal{S} \times_1 \mathbf{M}_1 \times_2 \cdots \times_n \mathbf{M}_n$, the flattened tensor $\mathcal{A}_{(i)}$ along mode i is then given by

$$\mathcal{A}_{(i)} = \mathbf{M}_i \mathcal{S}_{(i)} \left(\mathbf{M}_1 \otimes \cdots \otimes \mathbf{M}_{i-1} \otimes \mathbf{M}_{i+1} \otimes \cdots \otimes \mathbf{M}_n\right)^T. \tag{7}$$

3. Linear System for Affine Reconstruction of Camera matrices

The affine camera matrices must fulfill

$$\mathcal{S}_{(f),(10:13,:)}(\Downarrow_k \tilde{\mathbf{S}}_k^T \otimes \tilde{\mathbf{C}}_k)^T = \mathbf{Q}_{kron,(10:13,:)} \mathbf{Q}_{\mathit{aff}}^{-1} \hat{\mathbf{A}} \tag{8}$$

$$= \left[\Rightarrow_k \mathbf{1}_{1 \times N_k} \otimes \tilde{\mathbf{C}}_k^T\right] = \mathbf{Q}_{kron,(10:13,:)} \mathbf{Q}_{\mathit{aff}}^{-1} \left[\Rightarrow_k \hat{\mathbf{A}}_k\right], \tag{9}$$

where we used $\hat{\mathbf{A}}_k$ to denote the submatrix of $\hat{\mathbf{A}}$ due to camera k. Let us first investigate only such a submatrix $\mathcal{S}_{(f),(10:13,:)}(\tilde{\mathbf{S}}_k^T \otimes \tilde{\mathbf{C}}_k)^T$ due to one single camera k. Vectorization of this matrix equation using Eq. (6) and Eq. (3) gives

$$\mathbf{G}_k \text{vec}\left(\tilde{\mathbf{C}}_k^T\right) = \left((\mathbf{Q}_{\mathit{aff}}^{-1} \hat{\mathbf{A}}_k)^T \otimes \mathbf{I}_4\right) \text{vec}\left(\mathbf{Q}_{kron,(10:13,:)}\right) \tag{10}$$

$$= \mathbf{H}_k \text{vec}\left(\mathbf{Q}_{kron,(10:13,1:12)}\right) + \mathbf{b}_k \tag{11}$$

where the following matrices were introduced for abbreviation

$$\mathbf{G}_k = (\mathbf{I}_{N_k} \otimes \mathbf{T}_{2,1} \otimes \mathbf{I}_4)(\mathbf{1}_{N_k \times 1} \otimes \mathbf{I}_{2 \cdot 8}) \tag{12}$$

$$\mathbf{H}_k = \left(\left((\mathbf{Q}_{\mathit{aff}}^{-1})_{1:12,:} \hat{\mathbf{A}}_k\right)^T \otimes \mathbf{I}_4\right) \tag{13}$$

$$\mathbf{b}_k = \left(\left((\mathbf{Q}_{\mathit{aff}}^{-1})_{(13,:)} \hat{\mathbf{A}}_k\right)^T \otimes \mathbf{I}_4\right) \text{vec}\left(\mathbf{Q}_{kron,(10:13,13)}\right). \tag{14}$$

Combining each of the linear systems due to a camera in one single linear system leads to

$$\left[\searrow_k \mathbf{G}_k \quad -\Downarrow_k \mathbf{H}_k\right] \begin{pmatrix} \Downarrow_k \text{vec}\left(\tilde{\mathbf{C}}_k^T\right) \\ \text{vec}\left(\mathbf{Q}_{kron,(10:13,1:12)}\right) \end{pmatrix} = \left(\Downarrow_k \mathbf{b}_k\right). \tag{15}$$

The operator $\searrow_k$ stacks multiple matrices into a block-diagonal matrix. However, closer inspection of Eq. (9) reveals that successive rows result in the very same linear constraints. To avoid an unnecessary increase in unknowns, we therefore only consider one row for setting up the linear system which results in slightly changed matrices

$$\mathbf{G}_k = (\mathbf{I}_{N_k} \otimes \mathbf{T}_{2,1} \otimes \mathbf{I}_1)(\mathbf{I}_{N_k} \otimes \mathbf{I}_2) \tag{16}$$

$$\mathbf{H}_k = \left(\left((\mathbf{Q}_{\mathit{aff}}^{-1})_{1:12,:} \hat{\mathbf{A}}_k\right)^T \otimes \mathbf{I}_1\right) \tag{17}$$

$$\mathbf{b}_k = \left(\left((\mathbf{Q}_{\mathit{aff}}^{-1})_{(13,:)} \hat{\mathbf{A}}_k\right)^T \otimes \mathbf{I}_1\right) \text{vec}\left(\mathbf{Q}_{kron,(i,13)}\right). \tag{18}$$

The resulting over-constrained linear system reads like

$$\left[\searrow_k \mathbf{G}_k \quad -\Downarrow_k \mathbf{H}_k\right] \begin{pmatrix} \Downarrow_k \text{vec}\left(\tilde{\mathbf{C}}_{k,(i,:)}^T\right) \\ \text{vec}\left(\mathbf{Q}_{kron,(i,1:12)}\right) \end{pmatrix} = \left(\Downarrow_k \mathbf{b}_k\right), \tag{19}$$

which consists of only $2K + 1 \cdot 12$ unknowns instead of $4 \cdot 2K + 4 \cdot 12$ unknowns. Note that $\mathbf{Q}_{kron,(10:13,13)} = (0, 0, 0, 1)^T$ and therefore $\mathbf{b}_k$ is only non-zero for the last row which is associated with the camera translation. The system matrix in Eq. (19) however has a three-dimensional nullspace, and therefore provides four linear independent solutions for the four rows.

4. Linear System for Affine Reconstruction of Points

This derivation closely follows the one from Sec. 3. Let $\mathbf{X}_k$ denote the non-homogeneous part of the points $\mathbf{S}_k^T = \left[\mathbf{X}_k^T \quad \mathbf{1}_{N_k \times 1}\right]$ and $\mathbf{P}_k$ stands for the non-translational columns of the camera matrix $\mathbf{C}_k = \left[\mathbf{P}_k \quad \mathbf{t}_k\right]$. Using this notation, a valid affine reconstruction must fulfill

$$\mathcal{S}_{(f),(1:9,:)}(\Downarrow_k \tilde{\mathbf{S}}_k^T \otimes \tilde{\mathbf{C}}_k)^T = \mathbf{Q}_{kron,(1:9,:)} \mathbf{Q}_{\mathit{aff}}^{-1} \hat{\mathbf{A}} \tag{20}$$

$$= \left[\Rightarrow_k \tilde{\mathbf{X}}_k \otimes \tilde{\mathbf{P}}_k^T\right] = \mathbf{Q}_{kron,(1:9,:)} \mathbf{Q}_{\mathit{aff}}^{-1} \left[\Rightarrow_k \hat{\mathbf{A}}_k\right] \tag{21}$$

Vectorization of the submatrix equation due to camera k using Eq. (5) and Eq. (3) leads to

$$\mathbf{G}_k \text{vec}\left(\tilde{\mathbf{X}}_k\right) = \left((\mathbf{Q}_{\mathit{aff}}^{-1} \hat{\mathbf{A}}_k)^T \otimes \mathbf{I}_9\right) \text{vec}\left(\mathbf{Q}_{kron,(1:9,:)}\right) \tag{22}$$

$$= \mathbf{H}_k \text{vec}\left(\mathbf{Q}_{kron,(1:9,1:12)}\right) + \mathbf{b}_k \tag{23}$$

where the following matrices were introduced for abbreviation

$$\mathbf{G}_k = (\mathbf{I}_{N_k} \otimes \mathbf{T}_{2,3} \otimes \mathbf{I}_3)(\mathbf{I}_{3N_k} \otimes \operatorname{vec}(\tilde{\mathbf{P}}_k^T)) \tag{24}$$

$$\mathbf{H}_k = \left(\left((\mathbf{Q}_{\mathit{aff}}^{-1})_{1:12,:}\hat{\mathbf{A}}_k\right)^T \otimes \mathbf{I}_9\right) \tag{25}$$

$$\mathbf{b}_k = \left(\left((\mathbf{Q}_{\mathit{aff}}^{-1})_{(13,:)}\hat{\mathbf{A}}_k\right)^T \otimes \mathbf{I}_9\right) \operatorname{vec}\left(\mathbf{Q}_{kron,(1:9,13)}\right). \tag{26}$$

Combining again each of the linear systems due to a camera in one single linear system leads to

$$\left[\Searrow_k \mathbf{G}_k \quad -\Downarrow_k \mathbf{H}_k\right] \begin{pmatrix} \Downarrow_k \operatorname{vec}\left(\tilde{\mathbf{X}}_k\right) \\ \operatorname{vec}\left(\mathbf{Q}_{kron,(1:9,1:12)}\right) \end{pmatrix} = \left(\Downarrow_k \mathbf{b}_k\right). \tag{27}$$

A similar observation as in Sec. 3 holds true for Eq. (27). More specifically, successive row-triples in Eq. (21) result in the very same linear constraints. To avoid an unnecessary increase in unknowns, we therefore only consider one row triple for setting up the linear system which results again in slightly changed matrices

$$\mathbf{G}_k = (\mathbf{I}_{N_k} \otimes \mathbf{T}_{2,1} \otimes \mathbf{I}_3)(\mathbf{I}_{N_k} \otimes \operatorname{vec}(\tilde{\mathbf{P}}_k^T)) \tag{28}$$

$$\mathbf{H}_k = \left(\left((\mathbf{Q}_{\mathit{aff}}^{-1})_{1:12,:}\hat{\mathbf{A}}_k\right)^T \otimes \mathbf{I}_3\right) \tag{29}$$

$$\mathbf{b}_k = \left(\left((\mathbf{Q}_{\mathit{aff}}^{-1})_{(13,:)}\hat{\mathbf{A}}_k\right)^T \otimes \mathbf{I}_3\right) \operatorname{vec}\left(\mathbf{Q}_{kron,(3i-2:3i,13)}\right). \tag{30}$$

The resulting over-constrained linear system reads like

$$\left[\Searrow_k \mathbf{G}_k \quad -\Downarrow_k \mathbf{H}_k\right] \begin{pmatrix} \Downarrow_k \operatorname{vec}\left(\tilde{\mathbf{X}}_{k,(i,:)}\right) \\ \operatorname{vec}\left(\mathbf{Q}_{kron,(3i-2:3i,1:12)}\right) \end{pmatrix} = \left(\Downarrow_k \mathbf{b}_k\right), \tag{31}$$

which consists of only $\sum_k N_k + 3 \cdot 12$ unknowns instead of $3\sum_k N_k + 9 \cdot 12$ unknowns. Note that $\mathbf{Q}_{kron,(1:9,13)} = \mathbf{0}_{9\times 1}$ and therefore $\mathbf{b}_k$ is zero for every row-triple. However, the system matrix has a four dimensional nullspace, which should not come as a surprise since each basis vector of this nullspace provides a solution to a different row triple (one solution corresponds to the homogeneous coordinate of the points, which we do not need to solve for).

References

[1] R. Angst and M. Pollefeys. Static multi-camera factorization using rigid motion. In *Proc. ICCV*, 2009. 1

[2] J. Carroll and J.-J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition. *Psychometrika*, 35(3):283–319, September 1970. 1

[3] R. Harshman. Foundations of the parafac procedure: Models and conditions for an explanatory multi-modal factor analysis. *Working Papers in Phonetics*, 16, 1970. 1

[4] L. D. Lathauwer, B. D. Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21(4):1253–1278, 2000. 1

[5] L. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, September 1966. 1