Degeneracies, Dependencies and their Implications in Multi-body and Multi-Sequence Factorizations

Lihi Zelnik-Manor **Michal Irani**
Dept. of Computer Science and Applied Math
The Weizmann Institute of Science
76100 Rehovot, Israel

Abstract

The body of work on multi-body factorization separates between objects whose motions are independent. In this work we show that in many cases objects moving with different *3D motions will be captured as a single object using these approaches. We analyze what causes these degeneracies between objects and suggest an approach for overcoming some of them. We further show that in the case of multiple sequences linear dependencies can supply information for temporal synchronization of sequences and for spatial matching of points across sequences.*

1 Introduction

Dynamic scenes include either a single moving object or multiple moving objects. When a single video camera views a dynamic scene with multiple objects our goal is to separate between objects moving with different motions. We refer to this as the "multi-body factorization problem", which was presented in [3, 4]). When multiple video cameras view the same scene at the same time, in addition to the multi-body factorization problem, we are interested also in matching across cameras. This can be either temporal matching (i.e., temporal synchronization) or spatial matching (i.e., point correspondences). We refer to this as the "multi-sequence factorization problem".

The body of work on multi-body factorization (e.g., [3, 4, 5, 1, 7]), suggest employing multi-frame linear subspace constraints to separate between objects moving with independent motions. We first show that often objects moving with *different* 3D motions will be captured as a single object using previously suggested approaches. We show that this happens even when there is only partial linear dependence between the object motions. In many of these cases, although the motions are partially dependent they are conceptually different and we would like to separate them. Thus, we suggest a preliminary algorithm which can overcome some of these degeneracies and distinguish between such moving objects where other methods fail.

When multiple video sequences of the same dynamic scene are available (which includes either a single object or multiple objects) there can be dependencies between the motions or shapes captured by the sequences. Wolf & Zomet [10] showed that a dependence between the motions can be used for temporal synchronization of sequences. We extend this and show that even partial dependence suffices for synchronization purposes. This can be applied to synchronize sequences with only partial spatial overlap. Additionally, we show that when the sequences view the same set of points we get a dependence which can be used to find the spatial matching of the points across sequences.

2 Dependencies between Image Coordinate Matrices

Let $I_1, \ldots, I_F$ denote a sequence of F frames with N points tracked along the sequence. Let (x_i^f, y_i^f) denote the coordinates of pixel (x_i, y_i) in frame I_f ($i = 1, \ldots, N$, $f = 1, \ldots, F$). Let X and Y denote two $F \times N$ matrices constructed from the image coordinates of all the points across all frames:

$$X = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_N^1 \\ x_1^2 & x_2^2 & \cdots & x_N^2 \\ & & \vdots & \\ x_1^F & x_2^F & \cdots & x_N^F \end{bmatrix} \quad Y = \begin{bmatrix} y_1^1 & y_2^1 & \cdots & y_N^1 \\ y_1^2 & y_2^2 & \cdots & y_N^2 \\ & & \vdots & \\ y_1^F & y_2^F & \cdots & y_N^F \end{bmatrix} \tag{1}$$

Each row in these matrices corresponds to a single frame, and each column corresponds to a single point. Stacking the matrices X and Y of Eq. (1) vertically results in a $2F \times N$ matrix $\left[\frac{X}{Y}\right]$. It has been previously shown that under various camera and scene models [8, 6, 2] the image coordinate matrix of a single object can be factorized into motion and shape matrices: $\left[\frac{X}{Y}\right] = M_{2F\times r} S_{r\times N}$. When the scene contains multiple objects (see [3, 4]) we still get a factorization into motion and shape matrices $\left[\frac{X}{Y}\right] = M_{all} S_{all}$ where M_{all} is the matrix of motions of all objects and S_{all} contains shape information of all objects.

Let W_1 (of size $2F \times N_1$) and W_2 (of size $2F \times N_2$) be two image coordinate matrices ($W_1 = \left[\frac{X_1}{Y_1}\right]$ and $W_2 = \left[\frac{X_2}{Y_2}\right]$). In the case of the multi-body (single sequence) factorization, each of these would correspond to an independently moving object, i.e., $W_1 = W_{obj1}, W_2 = W_{obj2}$. In the case of multi-sequence factorization, each of these matrices corresponds to a single sequence (i.e., the trajectories of all points on all objects in each sequence): $W_1 = W_{seq1}$ and $W_2 = W_{seq2}$. Let r_1 and r_2 be the true (noiseless) ranks of W_1 and W_2, respectively. Then, in both cases these matrices can be each factorized into motion and shape information: $[W_1]_{2F\times N_1} = [M_1]_{2F\times r_1}[S_1]_{r_1\times N_1}$ and $[W_2]_{2F\times N_2} = [M_2]_{2F\times r_2}[S_2]_{r_2\times N_2}$, where M_1, M_2 contain motion information and S_1, S_2 contain shape information. In the multi-sequence factorization case the motion and shape matrices will include information of all the objects in the corresponding scene.

In this paper we will examine the meaning of full and partial linear dependencies between W_1 and W_2, and their implications on multi-body (single sequence) and multi-sequence factorizations. We will see that in the multi-body (single sequence) case these dependencies lead to degeneracies (and therefore are not desired), whereas in the multi-sequence factorization these dependencies are *useful* and provide additional information. In particular, there are two possible linear dependencies between W_1 and W_2: (i) Full or partial linear dependence between the *columns* of W_1 and W_2, and (ii) Full or partial linear dependence between the *rows* of W_1 and W_2. We will show that:

1. In the multi-body factorization case:
   - Dependencies between the columns of W_1 and W_2 cause degeneracies and hence misbehavior of multi-body segmentation algorithms. We also propose an alternative approach which can handle these cases.
   - Linear dependencies between the rows have no effect on the multi-body factorization.

2. In the multi-sequence factorization case:
   - Linear dependencies between the columns of W_1 and W_2 provide constraints for temporal correspondence (i.e., temporal synchronization) between sequences.
   - Linear dependencies between the rows of W_1 and W_2 provide constraints for spatial correspondence (i.e., spatial matching of points) across the sequences.

The analysis in the following sections will be based on the following two claims on the full/partial dependencies between the columns and between the rows of W_1 and W_2. We provide here the proofs for the full dependence case. The proofs of these claims for the case of partial dependence are provided in Appendix 5.

Claim 1 *The* columns *of W_1 and W_2 are fully/partially linearly dependent* **iff** *the* columns *of M_1 and M_2 are fully/partially linearly dependent.*

In the case of full linear dependence this reduces to:

$\exists C$ s.t. $W_2 = W_1C$ bf iff $\exists C'$ s.t. $M_2 = M_1C'$ (C is a $N_1 \times N_2$ coefficient matrix, and C' is a matrix of size $r_1 \times r_2$ which linearly depends on C).

Proof:
First Direction: $W_2 = W_1C \Rightarrow M_2S_2 = M_1S_1C \Rightarrow M_2S_2S_2^T = M_1S_1CS_2^T$. $S_2S_2^T$ is invertible (see note below) $\Rightarrow M_2 \equiv M_1C'$ where $C' = S_1CS_2^T(S_2S_2^T)^{-1}$.
Second Direction: $M_2 = M_1C' \Rightarrow M_2S_2 = M_1C'S_2$. $S_1S_1^T$ is invertible (see note below), hence, $M_1C'S_2 = M_1(S_1S_1^T)(S_1S_1^T)^{-1}C'S_2 \equiv M_1S_1C$ where $C = S_1^T(S_1S_1^T)^{-1}C'S_2$. This implies that $W_2 = M_2S_2 = M_1C'S_2 = M_1S_1C = W_1C$ $\diamondsuit$

Claim 2 *The* rows *of W_1 and W_2 are fully/partially linearly dependent* **iff** *the* rows *of S_1 and S_2 are fully/partially linearly dependent.*

In the case of full linear dependence this reduces to:

$\exists C$ s.t. $W_2 = CW_1$ **iff** $\exists C'$ s.t. $S_2 = C'S_1$ (C is a $N_2 \times N_1$ coefficient matrix, and C' is a matrix of size $r_2 \times r_1$ which linearly depends on C).

Proof:
First Direction: $W_2 = CW_1 \Rightarrow M_2S_2 = CM_1S_1 \Rightarrow M_2^TM_2S_2 = M_2^TCM_1S_1$. $M_2^TM_2$ is invertible (see note below), hence, $S_2 \equiv C'S_1$ where $C' = (M_2^TM_2)^{-1}M_2^TCM_1$.
Second Direction: $S_2 = C'S_1 \Rightarrow M_2S_2 = M_2C'S_1$. $M_1^TM_1$ is invertible (see note below), hence, $M_2C'S_1 = M_2C'(M_1^TM_1)^{-1}(M_1^TM_1)S_1 \equiv CM_1S_1$ where $C = M_2C'(M_1^TM_1)^{-1}M_1^T$. This implies that $W_2 = M_2S_2 = M_2C'S_1 = CM_1S_1 = CW_1$ $\diamondsuit$

Note: The assumption that $M_i^TM_i$ and $S_iS_i^T$ ($i = 1, 2$) are invertible is valid also in degenerate cases. M_i is assumed here to be a $2F \times r_i$ matrix and S_i is an $r_i \times N$ matrix, where r_i is the *actual* rank of W_i. In degenerate cases, the rank r_i will be lower than the theoretical upper-bound rank.

In the following sections we examine the meaning and the implications of each of these dependencies, both for the single-sequence (multi-body factorization) case and for the multi-sequence case.

3 Single Sequence Multi-Body Factorization

In this section we examine the implications of linear dependencies between the columns of W_1 and W_2 (of $obj1$ and $obj2$) on multi-body factorization. We show (Section 3.1) that full or partial dependence between the columns of W_1 and W_2 result in grouping together such objects in previous multi-body factorization approaches. In Section 3.2 we suggest an approach which can overcome some of these degeneracies and can separate between partially dependent objects. Section 3.3 shows that a dependence between the rows of W_1 and W_2 has no effect on the segmentation.

3.1 Dependencies between Object Motions

Let W_1, W_2 be the image coordinates sub-matrices corresponding to two objects ($W_1 = W_{obj1}$ and $W_2 = W_{obj2}$). We wish to classify the columns of the combined matrix

$[W_1|W_2]$ according to objects. Let π_1, π_2 be the linear subspaces spanned by the columns of W_1 and W_2, respectively, and r_1 and r_2 be the ranks of W_1 and W_2, respectively (i.e., $r_1 = rank(W_1)$ and $r_2 = rank(W_2)$). The two subspaces π_1 and π_2 can lie in three different configurations[1]:

I. Linear Independence: When π_1 and π_2 are two disjoint linear subspaces $\pi_1 \cap \pi_2 = \{0\}$, then $rank([W_1|W_2]) = r_1 + r_2$. According to Claim 1 this occurs when the motions M_1 and M_2 of the two objects are linearly independent. Algorithms for separating independent linear subspaces can separate the columns of W_1 and W_2.

II. Full Linear Dependence: When one subspace is a subset of (or equal to) the other (e.g., $\pi_2 \subseteq \pi_1$), then $W_2 = W_1 C$ and $rank([W_1|W_2]) = max(r_1, r_2)$. According to Claim 1 this occurs when the motions M_1 and M_2 of the two objects are fully linearly dependent, i.e., $M_2 = M_1 C'$. In this case all subspace based algorithms should group together the columns of W_1 and W_2.

III. Partial Linear Dependence: When π_1 and π_2 intersect partially ($\{0\} \subset \pi_1 \cap \pi_2 \subset \pi_1 \cup \pi_2$), then $max(r_1, r_2) < rank([W_1|W_2]) < r_1 + r_2$. According to Claim 1 this occurs when the motions M_1 and M_2 of the two objects are partially linearly dependent. In this case subspace based approaches can in general separate between the objects, however, most previously suggested algorithms will group them into a single object.

Costeira and Kanade [3] have estimated the SVD of $[W_1|W_2]$, i.e., $[W_1|W_2] = U\Sigma V^T$ (where U and V are unitary matrices) and showed that the "shape interaction matrix" $Q = VV^T = \begin{bmatrix} S_1^T \Lambda_1^{-1} S_1 & 0 \\ 0 & S_2^T \Lambda_2^{-1} S_2 \end{bmatrix}$ has a block diagonal structure. The algorithm they suggested (as well as those suggested in [3, 5, 1, 7]) relied on the block diagonal structure of Q which occurs iff V is block diagonal. However, the columns of V are the eigenvectors of $[W_1|W_2]^T[W_1|W_2] = \begin{bmatrix} S_1^T M_1^T M_1 S_1 & S_1^T M_1^T M_2 S_2 \\ S_2^T M_2^T M_1 S_1 & S_2^T M_2^T M_2 S_2 \end{bmatrix}$. Hence, V and therefore Q will have a block diagonal structure only if the motion matrices M_1 and M_2 are linearly independent. When M_1 and M_2 are partially dependent the off-diagonal blocks $S_1^T M_1^T M_2 S_2$ and $S_2^T M_2^T M_1 S_1$ are non-zero. Hence, algorithms like [3, 5, 1, 7], which rely on the block diagonal structure of Q will fail to separate between the objects. Note, that partial dependence occurs even if only a single column of the motion matrix of one object is linearly dependent on the columns of the motion matrix of the other object. This can occur quite often in real sequences.

An example where this occurs is illustrated in Fig. 1. The synthetic sequence displays a planar scene with two objects (one marked in red and the other marked in cyan) moving with the same rotations but with independent translations. Putting together all the image coordinates of all points corresponding to the red object yields a matrix W_{red} where $rank(W_{red}) = 3$. Similarly, the matrix of image coordinates of the cyan object has $rank(W_{cyan}) = 3$. Combining the image coordinates of all points on both objects into a single matrix gives $W = [W_{red}|W_{cyan}]$ with $rank(W) = 4$. This implies that the subspaces corresponding to the red and to the cyan objects intersect. Fig. 1.e shows that the matrix Q has no block-diagonal structure. Therefore, most previous subspace based segmentation algorithms will group the red object and the cyan object as one although their motions are obviously different.

3.2 Handling Partially Dependent Objects

As was shown by the example in Fig. 1 in many cases we would like to separate between objects with partial dependence. In this section we suggest an approach for doing so. When most of the trajectory vectors lie outside the intersection area and are relatively far from it (i.e., are unique to one object or the other) one would like for these to be interpreted as two separate objects.

The approach we suggest is similar to that suggested by Kanatani [7] which groups points into clusters in an agglomerative way. In Kanatani's approach the cost of grouping two clusters was composed of two factors. The first factor is based on the increase in the rank when grouping the two clusters. Although intended for independent objects, this applies also to the case of partially dependent objects. This is because adding to a group of points of one object more points from the same object will not change the rank (i.e., when the initial group of points spans the linear subspace corresponding to that object), whereas adding even a single point from another object will increase the rank, even if the objects are only partially dependent. When the clusters are large enough the agglomerative process is likely to continue correctly, however, for the initial stages to be correct we need additional information. For this Kanatani [7] used a second factor which is based on the values within the shape interaction matrix Q (high values indicate low-cost whereas low values indicate a high-cost). However, as was shown in Section 3.1, when the objects are partially dependent, the Q matrix looses its block diagonal structure. Hence, relying on this factor can lead to erroneous segmentation. Instead, we use the matrix $\hat{Q}$ defined next.

The values in $W^T W = [W_1|W_2]^T[W_1|W_2]$ are dot products between trajectory vectors (i.e., columns in W_i). These values decrease with the increase of the angle between the trajectory vectors. The angles between trajectory vectors of points on the same object are expected to be *on the average* significantly smaller than the angles between trajectory vectors of points on different objects (i.e., when the trajectory vectors lie relatively far from the intersection area). Therefore, although the off-diagonal blocks

[1] These claims build upon the observations previously made by Boult & Brown [1] and by Gear [4].

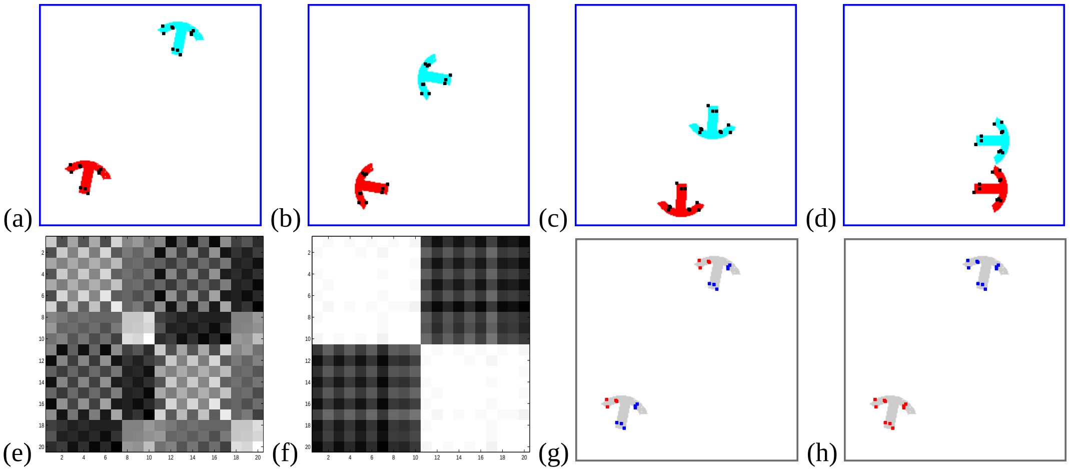


Figure 1. Degeneracies in multi-body (single sequence) factorization: *(a)-(d) Four sample frames (at times t=1,17,34,50) from a synthetic sequence showing partially dependent objects (tracked points marked in black). Both objects rotate counter-clockwise, but the red object also translates horizontally whereas the cyan object translates vertically. Although the motions of the red object and the cyan object are different, they are partially dependent. Therefore, their image coordinate matrices are partially dependent: $rank(W_{red}) = 3$, $rank(W_{cyan}) = 3$ and $rank([W_{red}|W_{cyan}]) = 4)$. (e) The distance matrix used by previous algorithms for segmentation (the "shape interaction matrix" Q) after sorting the points according to objects. There is no block diagonal structure hence the points will not be separated correctly. (f) The matrix $\hat{Q}$ we suggest to use, after sorting the points according to objects. The evident block-diagonal structure implies a good prior for segmentation. (g) Segmentation into two objects when using the classical shape interaction matrix Q. Points in differently detected clusters are marked by different colors (red and blue). The segmentation is obviously erroneous and mixes between the objects. (h) The corresponding segmentation result when using our proposed matrix $\hat{Q}$. The points on the red and the cyan objects are separated correctly.*

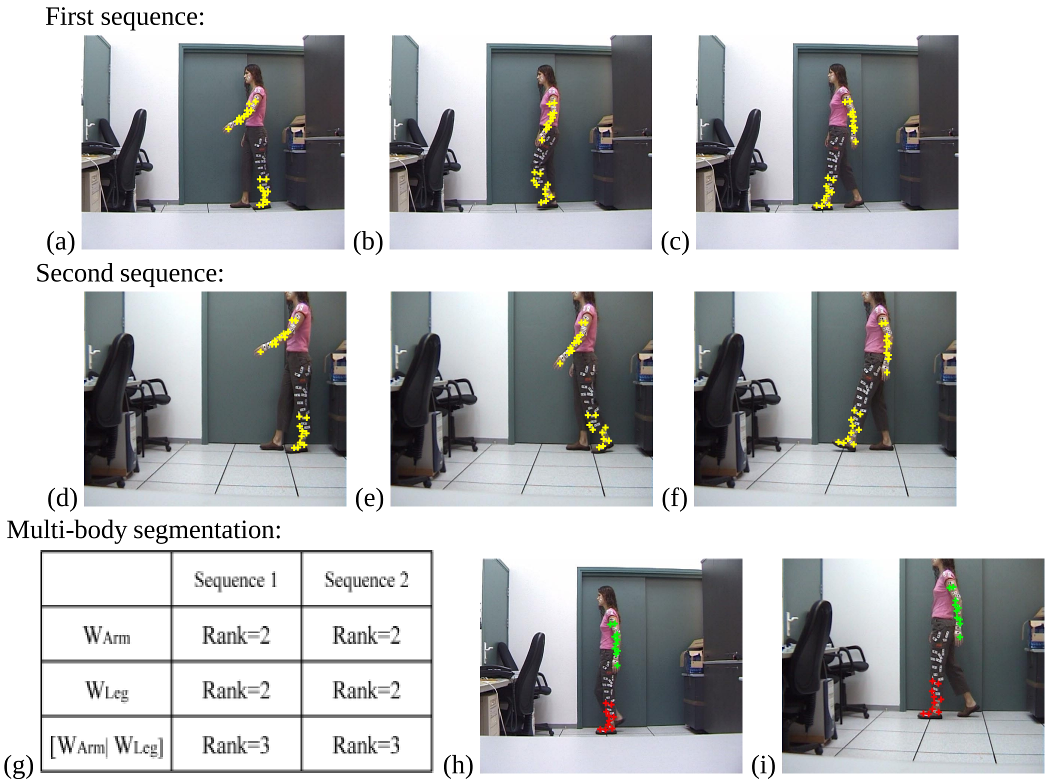
of W^TW are non-zero, they will usually have significantly smaller values than the main-diagonal blocks. We can thus write $W^TW = \begin{bmatrix} A & C \\ C^T & B \end{bmatrix}$, where on the average the values in A, B are larger than the values in C. According to Weiss [9], when A, B and C are constant the eigenvectors (denoted v) of the normalized version of W^TW (see [9]) will have $v(i) = v(j)$ for points i and j on the same object. In our case A, B and C are not constant, however, the values in A, B are larger than the values in C. We can thus expect the more dominant eigenvectors to account for the constant part which separates between the blocks and the less dominant eigenvectors to account for the within object variations. The shape interaction matrix Q is constructed using *all* eigenvectors hence has no block diagonal structure. Instead, we suggest to construct an affinity matrix $\hat{Q}$ where $\hat{Q}_{ij} = \sum_k e^{(v_k(i)-v_k(j))^2}$ where v_k are the most dominant eigenvectors of W^TW. Fig. 1.f shows the matrix $\hat{Q}$ for the synthetic sequence of Figs. 1.a-d (using a single, most dominant, eigenvector). It can be seen that $\hat{Q}$ has an obvious block diagonal structure, whereas Q does not.

Figs. 1.g,h show a comparison of the agglomerative clustering algorithm described above, once using the matrix Q, and once using the matrix $\hat{Q}$. Using $\hat{Q}$ gave correct segmentation results, whereas using Q mixed the points of the two objects.

Fig. 2 shows an example of partial dependence in a real sequence. In this example, two cameras viewed a person stepping forward. Figs. 2.a-c show example frames from the first camera sequence, and Figs. 2.d-f show example frames from the second camera sequence. The non-rigid motion performed by the person can be viewed as a group of rigid sub-parts each moving with a different motion. In both sequences we tracked points on two sub-parts: the arm and the shin (lower leg). The tracked points are marked in yellow in Figs. 2.a-c,d-f. In both sequences the rank of the image-coordinate matrix $W = [W_1|W_2]$ for all the points on both parts is higher than the rank of the image-coordinate matrix $W_i (i = 1, 2)$ for each of the individual parts but is lower than the sum of them, i.e., $rank(W_i) < rank([W_1|W_2]) < rank(W_1) + rank(W_2)$ (see Fig. 2.g). Fig. 2.h and Fig. 2.i show the result of applying this clustering scheme to the points tracked in the sequences of Figs. 2.a-c and Figs. 2.d-f, respectively. The result of the segmentation (forcing it into two objects) shows that the points on the arm and the shin were all classified correctly to the two different body parts.

3.3 Dependencies between Object Shapes

In Section 3.1 it was shown that a dependence between the *columns* of W_1 and W_2 causes degeneracies which af-



	Sequence 1	Sequence 2
WArm	Rank=2	Rank=2
WLeg	Rank=2	Rank=2
[WArm| WLeg]	Rank=3	Rank=3

Figure 2. Multi-body segmentation. *(a)-(c) Frames number* $1, 30, 60$*, from a sequence showing a walking person. Tracked points are marked in yellow. (d)-(f) Frames* $1, 30$ *and* 60 *from a second sequence showing the same scene, taken by a different camera (the two cameras where activated at different times). (g) The rank of the arm and leg together is lower than the sum of the individual ranks of each, since their motions are partially dependent. (h) Segmentation result of the first sequence, using our approach: different colors mark the different detected objects (red = leg, green = arm). (i) Segmentation result of the second sequence using our algorithm.*

fect the quality of multi-body segmentation. In this section we show that a dependence between the *rows* of W_1 and W_2 has no effect on the multi-body segmentation.

As was shown in Section 2, a dependence between the rows of W_1 and W_2 implies that $S_2 = C'S_1$. This implies that both objects share the same shape up to a selection of the object coordinate frame. When only the shapes are the dependent and the motions are still independent, the two objects will still be separated in all multi-body segmentation algorithms, since:

$$[W_1|W_2] = [M_1|M_2]\begin{bmatrix} S_1 & 0 \\ 0 & CS_1 \end{bmatrix}$$

and $rank([W_1|W_2]) = r_1 + r_2$.

4 Multi-Sequence Factorization

In this section we examine the case of multiple sequences, i.e., $W_1 = W_{seq1} = M_{seq1}S_{seq1}$ and $W_2 = W_{seq2} = M_{seq2}S_{seq2}$. Note that the two sequences can each contain multiple objects moving with different motions.The motion and shape matrices will accordingly include information of all the objects in the corresponding scene. As opposed to the multi-body factorization case, we show that in the multi-sequence case dependencies between the image coordinate matrices W_1 and W_2 of the two sequences do not cause degeneracies. On the contrary, they produce useful information! In particular, dependencies between the columns of W_1 and W_2 supply information for temporal synchronization of the sequences (Section 4.1) and dependencies between the rows of W_1 and W_2 supply information for spatial matching of points across sequences (Section 4.2).

4.1 Temporal Synchronization

Wolf & Zomet [10] showed how subspace constraints can be used for temporal synchronization of sequences when two cameras see the same moving objects. We reformulate this problem in terms of dependencies between W_1 and W_2, and analyze when this situation occurs. We further extend this to temporal synchronization in the case of *partial* dependence between W_1 and W_2 (i.e., when the fields-of-view of the two video cameras are only partially overlapping).

As was shown in Claim 1, the columns of W_1 and W_2 are linearly dependent when $M_2 = M_1C'$. Stacking W_1

and W_2 horizontally gives $[W_1|W_2] = M_1[S_1|C'S_2]$ and therefore $rank\left([W_1|W_2]\right) \leq r_1$. Note, however, that we get this low rank *only* when the rows of W_1 correspond to frames taken at the same time instance as the frames of the corresponding rows of W_2. This can be used to find the temporal synchronization between the two sequences, i.e., the alignment of rows with the minimal rank gives the temporal synchronization [10]. Furthermore, even if the motions that the two sequences capture are only partially linearly dependent, we will still get the lowest rank when the rows of the matrices W_1 and W_2 of the two sequences are correctly aligned temporally. Partial dependence between the motions captured by different sequences occurs when the fields of view of the corresponding cameras have only partial spatial overlap. Note that the cameras are assumed to be fixed with respect to each other (they can move jointly, however), but the points on the objects need *not* be the same points in both sequences.

Figs. 3.a-d show temporal synchronization results on the sequences of Fig. 2. The two stationary cameras viewed the same scene but from different view points and were *not* activated at the same time. We then used an approach similar to that suggested by Wolf & Zomet [10]. We tested the rank of the combined matrix $[W_1|W_2]$ for all possible temporal shifts. The rank was approximated by looking at the rate of decay of the singular values. Let $\sigma_1 \geq \sigma_2 \geq \sigma_3, \ldots$ be the singular values of $[W_1|W_2]$. We set $rank([W_1|W_2]) = i-1$ where i is the index of the largest singular value for which $\sigma_i/\sigma_1 < 0.01$. Since the data is noisy we might get this rank for more than one shift. Hence, we additionally estimated the residual error as $Error = \sum_{i=rank+1}^{N} \sigma_i$. The graph in Fig. 3.a shows that the minimum error was achieved at a temporal shift of 14 frames between the sequences. Figs. 3.b-d verify the correctness of this result.

4.2 Spatial Matching

When the video cameras are *not* fixed with respect to each other but they view the same set of 3D points in the dynamic scene, the spatial correspondence between these points across sequences can be recovered. In this case the subspaces spanned by the *rows* of W_1 and W_2 are equal, i.e., $W_2 = CW_1$. As was shown in Section 2, this occurs if and only if $S_2 = CS_1$. Note, that in this case there is no need for the cameras to be fixed with respect to each other (they can move independently), as only dependence between shapes (S_1 and S_2) is assumed. Stacking W_1 and W_2 vertically gives: $\left[\frac{W_1}{W_2}\right] = \left[\frac{M_1}{M_2C'}\right] S_1$ and $rank\left(\left[\frac{W_1}{W_2}\right]\right) \leq max(r_1, r_2)$. Note, that we get this low rank *only* when the columns of W_1 and W_2 correspond to the same points and are ordered in the same way. This can be used to find the spatial correspondence between the points in the two cameras, i.e., the permutation of columns in W_2 which leads to a minimal rank of $\left[\frac{W_1}{W_2}\right]$ gives the correct spatial matching.

Fig. 3.e examines this low rank constraint on the two sequences of Fig. 2. The graph shows the residual error for 1000 permutations, 999 of which were chosen randomly, and only one was set to be the correct permutation (the residual error here is again $Error = \sum_{i=rank+1}^{N} \sigma_i$, where σ_i are the singular values of $\left[\frac{W_1}{W_2}\right]$). The graph shows that the correct permutation yields a significantly lower error.

We next suggest a possible algorithm for obtaining such a spatial matching of points across sequences. However, the cross-sequence matching constraint presented above is not limited to this particular algorithm. One could, for example, start by selecting r point matches ($r = max(r_1, r_2)$) either manually or using image-to-image feature correspondence algorithms (taking only the r most prominent matches). The rest of the points could be matched automatically by employing only temporal information: given the matrix $\left[\frac{W_1^{match}}{W_2^{match}}\right]$ of the already matched points we add a new point (a new column) by choosing one point from the first sequence and testing the residual error when matching it against all the remaining points from the second sequence. The match which gives the minimal residual error is taken as the correct one. Fig. 3.f illustrates that the correct spatial matching was found.

5 Conclusions

In this paper we presented an analysis of linear dependencies and their implications on multi-body (single sequence) and multi-sequence factorizations. Our contributions are:
(i) a single unified framework for analyzing multi-body and multi-sequence factorization methods,
(ii) an analysis of degeneracies in multi-body factorization,
(iii) an approach to separating objects in such degenerate cases,
(iv) an extension of the temporal synchronization of [10] to the case of partially overlapping fields of view, and,
(v) an approach to spatial matching of points across sequences.

References

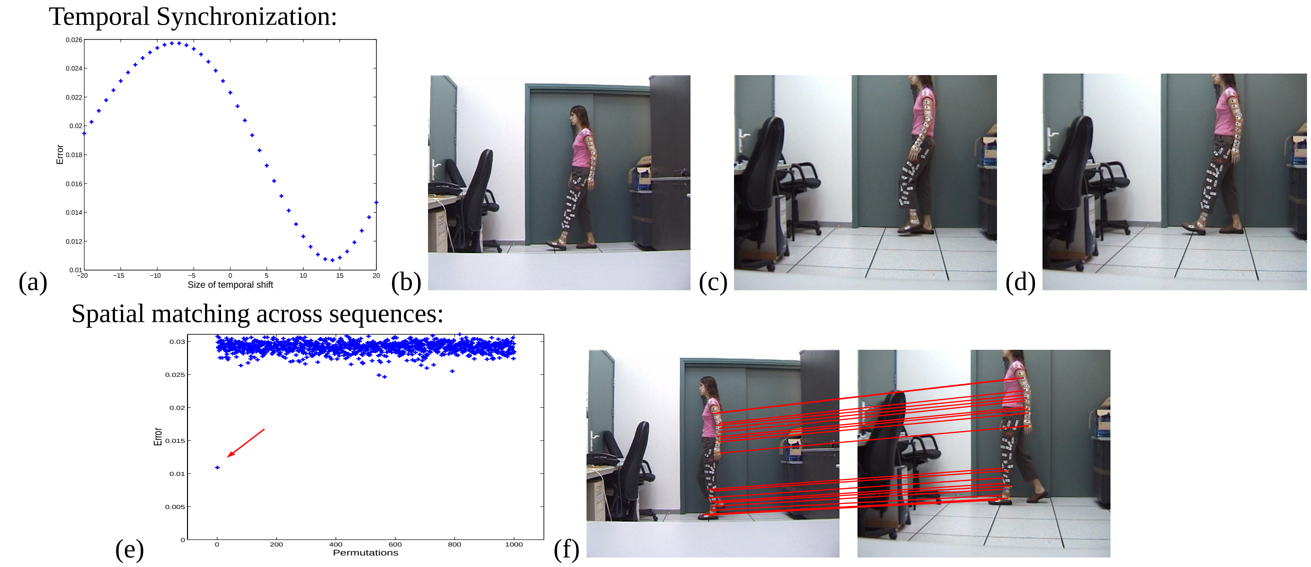
[1] T.E. Boult and L.G. Brown. Factorization-based segmentation of motions. In *in Proc. of the IEEE Workshop on Motion Understanding*, pages 179–186, 1991.

[2] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume II, pages 690–696, 2000.

[3] J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. In *International Conference on Computer Vision*, pages 1071–1076, Cambridge, MA, June 1995.

[4] C.W. Gear. Multibody grouping from motion images. *International Journal of Computer Vision*, 2(29):133–150, 1998.

[5] N. Ichimura. A robust and efficient motion segmentation based on orthogonal projection matrix of shape space. In *IEEE Conference on*


Temporal Synchronization:

Spatial matching across sequences:

Figure 3. Multiple Cameras. *Temporal synchronization and spatial matching applied to the sequences of Fig. 2. (a) The minimal rank residual error (see Section 4.1) is obtained at the correct temporal shift, which was 14 frames. (b) The 46'th frame of the first sequence. (c) The 46'th frame of the second sequence. As the second camera was turned on approximately half a second before the first one, this does* not *correspond to the 46'th frame of the first sequence. (d) The 60'th frame from the second sequence, which was recovered as the correct temporally corresponding frame (i.e., a shift of 14 frames). (e) The residual error for 1000 different pairings of the features across the two sequences (see Section 4.2). All wrong pairings give high residual error (i.e., high rank), whereas the correct pairing (marked by a red arrow) gives a small residual error (i.e., low rank). The correctly recovered pairing of feature points across the two sequences is shown in (f).*

Computer Vision and Pattern Recognition, pages 446–452, South-Carolina, June 2000.

[6] M. Irani. Multi-frame correspondence estimation using subspace constraints. *International Journal of Computer Vision*, 48(3):173–194, July 2002. A shorter version appeared in ICCV'99.

[7] K. Kanatani. Motion segmentation by subspace separation and model selection. In *ICCV*, volume 1, pages 301–306, Vancouver, Canada, 2001.

[8] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9:137–154, November 1992.

[9] Y. Weiss. Segmentation using eigenvectors: A unifying view. In *International Conference on Computer Vision*, pages 975–982, Corfu, Greece, September 1999.

[10] L. Wolf and A. Zomet. Correspondence-free synchronization and reconstruction in a non-rigid scene. In *Workshop*, Copenhagen, June 2002.

Appendix

Claim: The *columns* of W_1 and W_2 are partially linearly dependent **iff** the *columns* of M_1 and M_2 are partially linearly dependent.

proof:

The columns of W_1 and W_2 are partially linearly dependent **iff** we can find a basis B which spans the columns of the combined matrix $[W_1|W_2]$ such that $B = [B_1|B_{12}|B_2]$ and $W_1 = [B_1|B_{12}]C_1$; $W_2 = [B_{12}|B_2]C_2$, where C_1 and C_2 are coefficient matrices. This occurs **iff** $M_1 = [B_1|B_{12}]C_1S_1^T(S_1S_1^T)^{-1}$ and $M_2 = [B_{12}|B_2]C_2S_2^T(S_2S_2^T)^{-1} \Leftrightarrow M_1 = [B_1|B_{12}]C'_1$ and $M_2 = [B_{12}|B_2]C'_2$ where $C'_1 = C_1S_1^T(S_1S_1^T)^{-1}$ and $C'_2 = C_2S_2^T(S_2S_2^T)^{-1} \Leftrightarrow$ the columns of M_1 and M_2 are partially linearly dependent. $\diamondsuit$.

Claim: The *rows* of W_1 and W_2 are partially linearly dependent **iff** the *rows* of S_1 and S_2 are partially linearly dependent.

Proof:

The rows of W_1 and W_2 are partially linearly dependent **iff** we can find a basis B which spans the rows of the combined matrix $\left[\frac{W_1}{W_2}\right]$ such that $B = \begin{bmatrix} B_1 \\ B_{12} \\ B_2 \end{bmatrix}$ and $W_1 = C_1 \begin{bmatrix} B_1 \\ B_{12} \end{bmatrix}$; $W_2 = C_2 \begin{bmatrix} B_{12} \\ B_2 \end{bmatrix}$ where C_1 and C_2 are coefficient matrices. This occurs **iff** $S_1 = (M_1^TM_1)^{-1}M_1^TC_1 \begin{bmatrix} B_1 \\ B_{12} \end{bmatrix}$ and $S_2 = (M_2^TM_2)^{-1}M_2^TC_2 \begin{bmatrix} B_{12} \\ B_2 \end{bmatrix} \Leftrightarrow M_1 = C'_1 \begin{bmatrix} B_1 \\ B_{12} \end{bmatrix}$ and $M_2 = C'_2 \begin{bmatrix} B_{12} \\ B_2 \end{bmatrix}$ where $C'_1 = (M_1^TM_1)^{-1}M_1^TC_1$ and $C'_2 = (M_2^TM_2)^{-1}M_2^TC_2 \Leftrightarrow$ the rows of S_1 and S_2 are partially linearly dependent. $\diamondsuit$