Robust 3D Segmentation of Multiple Moving Objects Under Weak Perspective

Levente Hajder and Dmitry Chetverikov
Computer and Automation Research Institute and Eötvös Loránd University
Budapest, Hungary
hajder@sztaki.hu

Abstract

A scene containing multiple independently moving, possibly occluding, rigid objects is considered under the weak perspective camera model. We obtain a set of feature points tracked across a number of frames and address the problem of 3D motion segmentation of the objects in presence of measurement noise and outliers. We extend the robust structure from motion (SfM) method [5] to 3D motion segmentation and apply it to realistic, contaminated tracking data with occlusion. A number of approaches to 3D motion segmentation have already been proposed [3, 6, 14, 15]. However, most of them were not developed for, and tested on, noisy and outlier-corrupted data that often occurs in practice. Due to the consistent use of robust techniques at all critical steps, our approach can cope with such data, as demonstrated in a number of tests with synthetic and real image sequences.

1. Introduction

The SfM problem has been addressed by the computer vision community since late eighties. The factorisation procedure of Tomasi and Kanade [10] calculates the three-dimensional coordinates of an object from a sequence of feature points tracked across a number of frames. Given the 2D coordinates of the features, the output is the 3D coordinates of the points and the base vectors of the camera planes. The former is usually called the structure data, the latter the motion information.

The Tomasi-Kanade method [10] is applicable to a single (segmented) dynamic rigid object viewed under orthography. More recent studies [2, 13] attempt extending the theory to the nonrigid case. Other studies [7, 9] use the para-perspective or the perspective camera models. A key problem of the factorisation is, however, that of reliable *segmentation*. The procedure [10] is not robust. In particular, it fails if the input 2D data contains points of different moving objects.

A number of methods for 3D motion segmentation of feature points have already been proposed. (In this paper, we only consider motion segmentation methods that explicitely use 3D information.) Most of the approaches work with affine camera model. Costeira and Kanade [3] presented an algorithm based on *rank estimation* of the measurement matrix. The matrix contains the 2D coordinates of points tracked over all frames. A related algorithm was proposed by Gear [4]. Kanatani [6] developed a subspace based method that also needs rank estimation. Machline et al. [17] published a segmentation method applicable to rigid objects and to nonrigid objects that can be represented by linear combinations of rigid objects. The method relies on motion consistency that groups together pixels whose motion follows the same pattern over time. In this work, the measurement matrix is the optic flow field matrix whose columns are clustered based on rank estimation of sub-matrices.

The major drawback of the above algorithms is that they are noise-sensitive because of rank estimation. There is no universal, efficient rank estimation technique applicable to matrices deteriorated by significant noise and outliers. Unfortunately, the 2D coordinates of feature points tracked by standard trackers (e.g., [11]) in real sequences form very noisy measurement matrices; the same applies to measurement matrices based on optic flow.

For noise-free data, or in presence of small noise, the rank estimation based algorithms work reasonably well. They can segment an arbitrary number of independently moving objects. However, the situation changes when real, strongly contaminated tracking data is to be processed. This is why some of the above studies (for example, [3, 6]) use markers or manually selected feature points in their tests.

Torr et al. [12] proposed 3D motion segmentation methods based on the estimation of *fundamental matrices* or trifocal tensors. These algorithms work with real perspective, but compute robust statistics only from two or three images. The segmentation is based on clustering of the tracked feature points according to fundamental matrices or trifocal tensors. A weak point of these methods is handling relatively small objects. When an object is represented by a small portion of all feature points, it is difficult to segment the points of the object by clustering, because too many samples are needed to find the initial cluster. Otherwise, the tests presented in [12] demonstrate that the methods can cope with realistic data such as automatically tracked fea-

ture points.

The epipolar constraint was generalised to the multibody case by Vidal et al. [14] who use the multibody fundamental matrix for SfM in case of multiple moving objects. However, this method needs many image pairs to obtain the multibody fundamental matrix, and it relies on rank estimation to calculate the number of objects. The epipolar constraint was also applied for optical flow based segmentation under weak perspective [15]. The disadvantage of this method is the sensitivity of fundamental matrix computation to noisy co-ordinates of the tracked points.

In this paper we propose a new robust method for 3D motion segmentation. The proposed method is an extension of the weak-perspective SfM algorithm of Hajder et al. [5] which is applicable to strongly contaminated tracking data, when the inlier ration is below 50%. Due to the use of robust techniques, our 3D motion segmentation approach can also handle such data. The advantage of our method over the subspace based methods [3, 6] is in the use of 3D motion coherence. Motion based segmentation under weak perspective is a special type of subspace clustering. The subspace methods do not consider constraints on 3D motion of objects.

The structure of this paper is as follows. In section 2 we introduce basic notions and present formulas related to structure from motion under orthography and weak perspective. We will need these equations later on in section 3, where the proposed method is described. Experimental results are given in section 4, conclusions and outlook in section 5.

2 SfM under weak perspective

Given P feature points of a rigid object tracked across F frames, $x_{fp} = (u_{fp}, v_{fp})^T$, $f = 1, \ldots, F$, $p = 1, \ldots, P$, the goal of SfM is to recover the structure of the object. For orthogonal projection, the 2D coordinates are calculated as

$$x_{fp} = R_f s_p + t_f, \tag{1}$$

where $R_f = [r_{f1}, r_{f2}]^T$ is the orthogonal rotation matrix, s_p the 3D coordinates of the point and t_f the offset. Under the weak perspective model, the equation is

$$x_{fp} = q_f R_f s_p + t_f, \tag{2}$$

where q_f is the nonzero scale factor of weak perspective. The offset vector is eliminated by placing the origin of 2D coordinate system at the centroid of the feature points.

For all points in the f-th image, the above equations can be rewritten as

$$W_f = (x_{f1} \ldots x_{fP}) = M_f \cdot S \tag{3}$$

where M_f is called the motion matrix, $S = (s_1, \ldots, s_P)$ the structure matrix. Under orthography $M_f = R_f$, under weak respective $M_f = q_f R_f$.

For all f, equations (3) form $W = M \cdot S$, where $W^T = [W_1^T, W_2^T, \ldots, W_F^T]$ and $M^T = [M_1^T, M_2^T, \ldots, M_F^T]$. The task is to factorise the measurement matrix W and obtain the structural information S. This can be done in two steps. In the first step the rank of W is reduced to three by the singular value decomposition (SVD), since the rank of W is at maximum three: $W^{2F\times P} = \hat{M}^{2F\times 3} \cdot \hat{S}^{3\times P}$. This factorisation is determined only up to an affine transformation because an arbitrary 3×3 non-singular matrix Q can be inserted so that $W = \hat{M}QQ^{-1}\hat{S}$. Therefore $\hat{M}$ contains the base vectors of the frames deformed by an affine transformation. The matrix Q can be determined optimally by least squares optimisation both for orthogonal [10] and weak-perspective [16] case imposing the constraint on the frame base vectors. The estimated motion vectors can be written as $R = \hat{M}Q$, where $R = [r_{11}, r_{12}, \ldots, r_{F1}, r_{F2}]^T$.

3. Proposed algorithm

The motion segmentation algorithm uses two basic assumption: Each object is rigid, connected and does not contain narrow parts (compactness).

The main idea of the algorithm is as follows. Select and track as many feature points as possible. Divide the first frame of the sequence into regions, for example, discs or squares. The regions may overlap. (In our implementation, we use non-overlapping squares.) A feature is identified by its region in the first frame. Then apply the robust factorisation [5] to the tracked 2D features of each region separately. Check if there is a correct dominant 3D motion in a region. Select the correct region having the least motion error. Use this region as the seed and grow it by aggregating in the neighbouring regions those points that have similar 3D motion. Stop at motion borders, remove the aggregated features, then iterate the procedure until no more correct region is available.

The factorisation method of Hajder et al. [5] is a robust procedure based on the Least Trimmed Squares [8]. It can find dominant 3D motion in presence of noise and a large amount of outliers, by detecting and discarding the outliers. For the details, the reader is referred to the paper [5]. Details of other parts of the proposed segmentation approach are given below. In the end of this section, we summarise the algorithm.

3.1. Selecting region with least motion error

The tracked points of a region are processed by the robust SfM algorithm [5]; the outliers are detected and removed from the measurement matrix. The remaining data can potentially represent a correct 3D motion, but there is no guarantee for that. When the algorithm [5] has been applied to every region of the first frame, we need a measure of motion error to be able to compare the regions and select the most

promising one. The motion error for a region is obtained as follows:

1. Select randomly four points from the set of the region's features.
2. Calculate motion and structure by factorisation.
3. Normalise all base vectors. Replace the third element of each base vector by its absolute value because of reflection. Rotate the base vectors: let the base vectors of the first frame be parallel to $[1,0,0]^T$ and $[0,1,0]^T$ vectors.
4. Create a concatenated vector by concatenating the base vectors of the camera planes.
5. Repeat steps 1–4.
   Steps 1–5 yield two concatenated vectors.
6. Calculate the norm of the difference between the two concatenated vectors. Divide the difference by the number of the base vectors.
7. Repeat 20 times steps 1–6.
8. Calculate the error as the average of the 20 norms obtained.

In [5], the above motion error is analysed both theoretically and experimentally. An expression is derived for the mean of the squared difference between two random vectors on a semi-sphere. In this model, for infinitely large noise the expected value of the error is 1.5. Tests on simulated data with different levels of noise confirmed that the error tends to 1.5 as noise grows. Figure 1 shows the plot of the average square error versus the noise level. (The horizontal axis is $100r/R$, where R is the size of the synthesised moving object, and r is the variance of the Gaussian noise.) As noise grows, the error increases, then levels off at a value close to 1.5. The motion data becomes random; the base vectors of the frames spread randomly over a semi-sphere of unit radius.

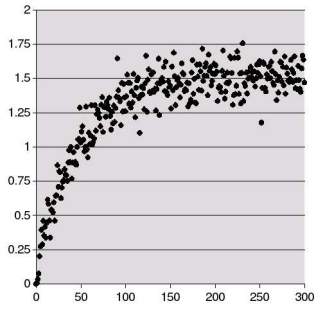

Figure 1: Errors of motion estimation versus 2D noise level.

In the proposed segmentation algorithm, we use the above analysis to decide if the motion of a region is *correct*: If the motion error is below a threshold T_{err}, the feature points belong to the same moving object. In the tests below, we set $T_{err} = 0.5$. A motion error value is obtained for each region, and the region with the smallest error is selected as the seed.

3.2. Finding points with known motion

After a correct seed motion has been selected, we try to extend it to the points of the neighbouring regions. In this section we show how to determine if a feature point is moving according to a known 3D motion. Due to the ambiguity of factorisation, this problem is not trivial.

Given a measurement matrix W, factorisation yields a 3D motion matrix and a 3D structure matrix:

$$W = (\hat{M}Q)(Q^{-1}\hat{S}), \tag{4}$$

where $M = (\hat{M}Q)$ represents the 3D motion and $S = (Q^{-1}\hat{S})$ represents the 3D structure. The factorisation is ambiguous: the formula described in section 2 provides a correct result, but this result is not unique. If the coordinate system of the structure is rotated by a matrix A, the motion vectors by A^T, where A is an Euclidean transformation matrix ($AA^T = I$), then $W = (MA^T)(AS)$ is also a correct factorisation. It can be proved that if $rank(S) = 3$, then all possible factorisations can be written in the form of $W = (MA)(A^TS)$. (See appendix A for a proof.)

We have a correct 3D motion matrix M and we would like to separate the feature points with this motion from other feature points. The segmentation process is based on the error value of a feature. This error, ϵ_p, is different from the motion error discussed in section 3.1. For better clarity, we will call ϵ_p the *incoherence value*. It is defined as follows.

Let w_p be the p^{th} column of the measurement matrix. w_p contains the tracked 2D coordinates of a feature point over all frames. According to motion M, the 3D coordinates of the p^{th} point can be estimated by the least square method: $\hat{s}_p = M^\dagger w_p$. The 2D coordinates are given by $\hat{w}_p = MM^\dagger w_p$. The error value ϵ_p is determined as

$$\epsilon_p = \|w_p - \hat{w}_p\| = \|(E - MM^\dagger)w_p\| \tag{5}$$

The incoherence ϵ_p is essentially the reprojection error of point p for motion M. It has the beneficial property of being invariant to Euclidean transformations of the motion matrix. In appendix A, we prove that Tomasi-Kanade factorisation is ambiguous up to an Euclidean transformation. Therefore, if M is a correct motion matrix and A is orthogonal, then $\tilde{M} = MA$ is also a correct motion matrix. The error value $\tilde{\epsilon}_p$ according to the transformed motion matrix $\tilde{M}$ is equal to ϵ_p:

$$\tilde{\epsilon}_p = \|(E - MA(MA)^\dagger)w_p\| = \epsilon_p, \tag{6}$$

because $(MA)^\dagger = A^TM^\dagger$, as shown in appendix B.

3.3. Summary of proposed algorithm

The main steps of the proposed 3D motion segmentation algorithm are as follows:

1. **Tracking**. Compute a dense feature point set and track the features over the sequence. Divide the first frame into regions. Identify each feature by its region in the first frame.
2. **Computing motion errors**. For each region,
   (a) detect outliers by the algorithm [5] and discard them from the measurement matrix;
   (b) calculate motion error according to section 3.1.
3. **Selecting correct seed region**. Select the region with the minimal motion error. If this minimal error exceeds a pre-defined limit ($T_{err} = 0.5$), stop the algorithm and indicate that there is no more correct 3D motion in the sequence. Otherwise, calculate the motion matrix M for the points of the selected region.
4. **Calculating incoherence values**. For each region,
   (a) detect outliers and discard them;
   (b) for each point, calculate by (5) its incoherence value with respect to the motion matrix M;
   (c) calculate the average incoherence for the region;
   (d) create an *incoherence map* whose pixels represent the normalised incoherence values of the regions.
5. **Growing the seed region**. Grow the seed in the incoherence map by aggregating the connected pixels with similar incoherence values.
6. **Iterating the procedure**. Remove the feature points of the segmented area from the initial dataset, then go to step 2.

4. Experimental results

The proposed 3D motion segmentation algorithm was tested both on synthetic and real video sequences. In all cases, feature points were detected in the first frame by the well-known KLT feature (corner) detector [11]. Then a simple template matching method (shift-corrected SSD) was used to track the points. When setting the parameters of the algorithms, we tried to obtain as many tracks as possible, at the expense of higher possibility of incorrect or lost tracks. This was done for two reasons: (1) One needs dense features for a good segmentation; (2) We wanted to test the robustness of the method against a large number of outliers.

4.1. Test on synthetic sequence

The first test sequence consists of a cube and a sphere moving (shifting and rotating) separately against a textured background, all viewed with a moving camera. That is, the background is dynamic. The first and the last frames of the sequence are shown in figure 2. The animation was generated by the PovRay ray-tracer software with a resolution of 1000×800 pixels. The sequence consists of 10 frames. The sphere occludes the cube in all frames of the sequence.

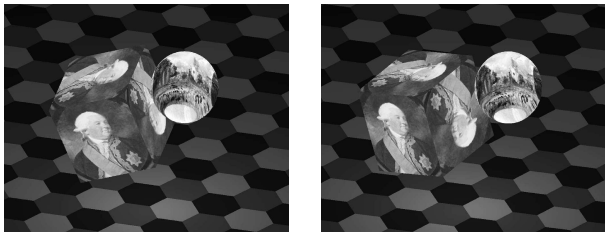

Figure 2: First and last frames of synthetic sequence.

Figure 3 shows the 3D motion errors of the regions, computed in step 2 of the algorithm: the brighter the pixel, the larger the error. If a pixel is white, motion error cannot be computed because of the lack of features in the region. The locations of the cube and the sphere in the first frame are visible. The error is high at the occlusion border because the motion data of the objects are mixed in the measurement matrix of the factorisation. The motion error is larger at the background than at the objects because the camera motion is smaller than the motion of the cube and the sphere.

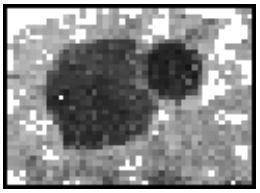

Figure 3: Motion errors for synthetic sequence.

Note that in this case the error map itself could be used to segment the objects. However, the result improves after using the incoherence map. Figure 4 displays the incoherence maps w.r.t. the motion matrices of the cube and the sphere, respectively. The segmented regions are shown in figures 5 and 6.

4.2. Test on real sequences

The segmentation method was also tested on two real image sequences. The 'Bear' sequence (figure 7) was acquired by a 2Mpixel digital camera. The sequence has 15 frames. Both the camera and the object are moving. The resolution is relatively high: 800×600 pixels. The segmented region of the Bear is shown in figure 8.

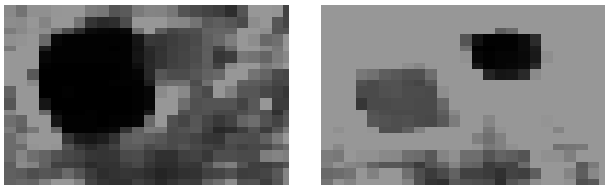

Figure 4: Incoherence maps for two detected motions. Left: w.r.t. cube motion. Right: w.r.t. sphere motion.

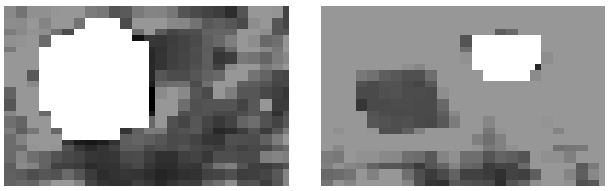

Figure 5: Segmentations of incoherence maps.

The 'Car' video (figure 9) shows a car taking a bend. The quality of the sequence is poor, the resolution is only 320×200 pixels, and the images are noisy. Despite the low quality of the video, the segmentation algorithm can separate the feature points of the car from the points of the moving background, as demonstrated in figure 10.

5. Summary and conclusions

We have presented a novel method for 3D motion segmentation of a sequence showing multiple moving objects. Compared to the previous methods using rank estimation, our method has the advantage of being robust and applicable to real tracking data in presence of significant noise and a large number of outliers. Compared to the methods by Torr et al. [12], our method has the advantage of being capable to handle relatively small objects as well. Another positive feature is that the algorithm has a small number of parameters that are easy to interpret and set.

In particular, we have developed principled methods for estimating and thresholding the motion error of a region and for determining, in an invariant way, the feature points whose motion is consistent with a given motion matrix.

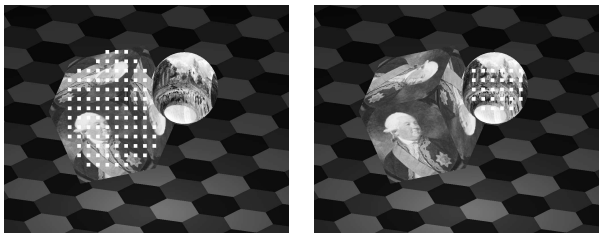

Figure 6: Segmented regions of cube (left) and sphere (right).

Figure 7: First and last frames of 'Bear' sequence.

Figure 8: Segmented region of Bear.

The robustness of the proposed method is due to: (1) robust seed selection (searching regions containing correct 3D motion); (2) robust coherence measure that provides a map which is segmented by region growing.

The property of robustness does not come at no cost. Since the robust techniques used at all critical steps of our approach require multiple testing of the data, the method needs a significant computational effort; however, this effort is prohibitive neither for testing nor for application.

We are currently working on quantitative, comparative performance evaluation of the proposed method. At the same time, we would like to extend the method to articulated, non-rigid objects.

Acknowledgment. This work was supported by the EU Network of Excellence MUSCLE (FP6-507752).

References

[1] Å. Björck. *Numerical Methods for Least Squares Problems*. Siam, 1996.

[2] M. Brand and R. Bhotika. Flexible Flow for 3D Nonrigid Tracking and Shape Recovery. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 312–322, December 2001.

[3] J. Costeira and T. Kanade. A Multibody Factorization Method for Independently Moving Objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.

[4] C. Gear. Mutibody Grouping from Motion Images. *International Journal of Computer Vision*, 29:133–150, 1998.

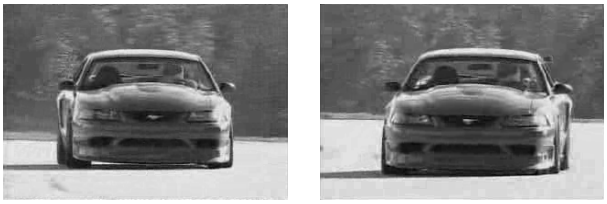

Figure 9: First and last frames of 'Car' sequence.

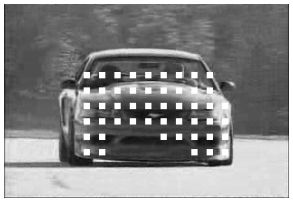

Figure 10: Segmented region of Car.

[5] L. Hajder, D. Chetverikov, and I. Vajk. Robust Structure from Motion under Weak Perspective. In *2nd Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, Sept 2004.

[6] K. Kanatani. Motion Segmentation by Subspace Separation and Model Selection. In *ICCV*, pages 586–591, 2001.

[7] C. J. Poelman and T. Kanade. A Paraperspective Factorization Method for Shape and Motion Recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):312–322, March 1997.

[8] P. Rousseeuw and A. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, NY, 1987.

[9] P. Sturm and B. Triggs. A Factorization Based Algorithm for Multi-Image Projective Structure and Motion. In *ECCV*, volume 2, pages 709–720, April 1996.

[10] C. Tomasi and T. Kanade. Shape and Motion from Image Streams under orthography: A factorization approach. *Intl. Journal Computer Vision*, 9:137–154, November 1992.

[11] C. Tomasi and J. Shi. Good Features to Track. In *IEEE Conferences on Computer Vision and Pattern Recognition*, pages 593–600, June 1994.

[12] P. H. S. Torr, A. Zisserman, and D. W. Murray. Motion clustering using the trilinear constraint over three views. In *Europe-China Workshop on Geometrical Modelling and Invariants for Computer Vision*, pages 118–125, 1995.

[13] L. Torresani, D. Yang, E. Alexander, and C. Bregler. Tracking and Modelling Nonrigid Objects with Rank Constraints. In *IEEE Computer Society Conference on Computer Vision and Patter Recognition*, 2001.

[14] R. Vidal. Segmentation of Dynamic Scenes from the Multibody Fundamental Matrix. In *ECCV Workshop on Vision and Modeling of Dynamic Scenes*, June 2002.

[15] J. Weber and J. Malik. Rigid Body Segmentation and Shape Description from Dense Optical Flow Under Weak Perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):139–143, 1997.

[16] D. Weinshall and C. Tomasi. Linear and Incremental Acquisition of Invariant Shape Models From Image Sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):512–517, 1995.

[17] L. Zelner-Manor, M. Machline, and M. Irani. Multi-body Segmentation: Revisiting Motion Consistency. In *ECCV Workshop on Vision and Modeling of Dynamic Scenes*, June 2002.

A. Ambiguity of factorisation

The Tomasi-Kanade factorisation method factorises the measurement matrix W into a motion matrix M and a structure matrix S: $W = MS$, where $M = [M_1^T M_2^T ... M_F^T]^T$ represent the motion data and S contains the 3D coordinates of the object. M_l is the motion information of the l^{th} frame: $M_l^T = [\mathbf{i}_l^T, \mathbf{j}_l^T]$. $\mathbf{i}_l$ and $\mathbf{j}_l$ are 3D base vectors of the k^{th} image plane. Motion submatrices can be completed with the third base vector perpendicular to the fist two base vectors $\mathbf{i}_l$ and $\mathbf{j}_l$: $\tilde{M}_l = [\mathbf{i}_l^T, \mathbf{j}_l^T, \mathbf{k}_l^T]$, where $\mathbf{k}_l = \mathbf{i}_l \times \mathbf{j}_l$ and $\tilde{M}$ is an orthogonal matrix.

The factorisation of the measurement matrix W is ambiguous. Let us assume that we have a valid factorisation $W = MS$. All valid factorisation of W can be written in the form of $W = (MA)(A^{-1}S)$, if $rank(S) = 3$. Since MA is a motion matrix, it must the fulfil the motion constraints. Let MA be denoted by $N = MA = [N_1^T N_2^T ... N_F^T]^T$. $\tilde{N}_l$ denotes the completed new motion matrix of the l^{th} image of the sequence. It is known that $N_l = M_l A$ and $\tilde{N}_l = \tilde{M}_l A$. $\tilde{N}_l$ is orthogonal, so we have $\tilde{N}_l^T \tilde{N}_l = A^T \tilde{M}^T \tilde{M} A = I$. This is true if and only if $A^T A = I$, because the original completed motion matrix is orthogonal.

The following conclusion is drawn: The Tomasi-Kanade factorisation is ambiguous up to an arbitrary orthonormal transformation.

B. Pseudoinverse of matrix product

Given a matrix M, its Moore-Penrose pseudoinverse $M^\dagger$ and an orthogonal matrix A, the task is to determine the pseudoinverse of MA. It is known [1] that the pseudoinverse of M can be written as

$$M^\dagger = V(V^T V)^{-1}(U^T U)^{-1} V^T, \tag{7}$$

where $M = UV^T$ is a minimal dyadic decomposition matrix M. The dyadic decomposition of MA is

$$MA = U(V^T A) \tag{8}$$

The Moore-Penrose pseudoinverse based on dyades can be written as follows:

$$(MA)^\dagger = A^T V(V^T A A^T V)^{-1}(U^T U)^{-1} V^T = A^T M^\dagger, \tag{9}$$

because $A^T A = I$.