# 3D motion consistency analysis for segmentation in 2D video projection

Wei Zhao, Nico Roos and Ralf Peeters

Department of Data Science and Knowledge Engineering,
Maastricht University,
6200 MD Maastricht

**Abstract.** Motion segmentation for 2D videos is usually based on tracked 2D point motions, obtained for a sequence of frames. However, the 3D real world motion consistency is easily lost in the process, due to projection from 3D space to the 2D image plane. Several approaches have been proposed in the literature to recover 3D motion consistency from 2D point motions. To further improve on this, we here propose a new criterion and associated technique, which can be used to determine whether a group of points show 2D motions consistent with joint 3D motion. It is also applicable for estimating the 3D motion information content. We demonstrate that the proposed criterion can be applied to improve segmentation results in two ways: finding the misclassified points in a group, and assigning unclassified points to the correct group. Experiments with synthetic data and different noise levels, and with real data taken from a benchmark, give insight in the performance of the algorithm under various conditions.

## 1 Introduction

Motion provides an important clue for the analysis of video sequences. It can be used for either detecting and segmenting the moving objects present in the scene, or recovering the 3D structure of a scene [18, 5, 2, 4].

When an image is taken by a camera, it maps the 3D world onto a 2D image plane by a projective transformation. The motions of the scene objects as well as the camera, jointly cause the changes of corresponding pixels in the image. We can detect these 2D motions by estimating the displacements of pixels between frames, or by tracking salient feature points from video sequences [11, 4, 20, 13]. Motion segmentation aims at grouping together the points (or pixels) that have the same motion in the video sequence. The key issue of motion segmentation is the definition of "same motion", which can be a 3D motion in the three-dimensional world, or simply a 2D motion of image pixels [24]. Motion segmentation is difficult because the detected motions of points (or pixels) are combined with displacements caused by camera motion and parallax caused by 3D structures [23].

Many motion segmentation approaches group together pixels undergoing the same 2D motion between successive images in the sequence [2, 21, 1, 15, 19, 26].

However, 3D geometric properties are typically affected by the transformation, such as shapes, angles and distances [9]. 3D motion consistency of points in the world coordinate frame is therefore not assured to be preserved in the 2D image coordinate frame. It implies that different parts of one and the same object, e.g. the three visible sides of a cube, can show different 2D motion patterns in the image plane. As a result, 2D motion based methods will fail to properly segment out the object. Another class of motion segmentation methods, tries to capture the 3D motion consistency with the help of constraints derived from geometric or physical models, such as rigidity of an object, 2D homography, an epipolar constraint, or a trilinear constraint [5, 10, 7, 24]. These constraints, with some success, allow to group points (or pixels) moving with the same 3D motion, based on their 2D motion information at the projected image.

In this paper, we propose a new criterion for measuring the 3D rigid motion consistency of a group of points, based on their 2D motions. This criterion can be used with singular value decomposition (SVD), to measure the 'quality' of segmented groups of points in a way that will be made more precise later. It is also applicable for recovering the parameters of 3D rigid motion giving the 2D motion information of a collection of points from the same object. This is used to detect misclassified points and to assign unclassified points to the correct group.

## 2 Related work

A key problem of motion segmentation is to determine whether a set of points all have the same motion. Normally the motion of points (or pixels) is estimated by detecting their 2D positions at each frame from a video sequence. The 2D motion of each point (pixel) in image plane is a projection of a 3D motion in the scene. The "same motion" can be defined based on either their 2D motion consistency, or on their 3D motion consistency.

For segmentation of 2D motions in the image space, one straightforward way is to define the "same motion" with a parametric motion model. Parametric approaches use a 2D affine transformation to describe joint 2D motion [21, 2, 27]. The affine motion model neglects perspectivity effects, and is largely limited to approximate the rigid motion of planar surfaces far away from the camera. Non-parametric models, such as Gaussian processes, are more flexible and suitable for curved surfaces [22]. However, these methods usually segment 3D objects into multiple parts, because of discontinuities in projected 2D motions, caused by perspective effects, depth discontinuities, occlusions, etc. [24]

3D motion segmentation searches for multiple-view geometric constraints to measure the 3D motion consistency. The two-view-based approaches model the motion by a fundamental matrix, based on the epipolar geometry [19, 12]. Other approaches are based on the three-view geometry, and encapsulate the trilinear relations of corresponding points in three images [16, 25]. These methods handle 3D motion consistency by preserving the 3D relations of points. Motion segmentation based on such geometric constraints tends to suffer from a "chicken and egg" problem, as it requires prior knowledge of the number of objects [5].

A variety of solutions are proposed to avoid estimating the motion model explicitly, thus solving this "chicken and egg" problem. The factorization method introduced by Tomasi and Kanade [14], factorizes the trajectory matrix of points tracked in a video sequence into a motion matrix and a shape matrix. The rigidity of objects ensures the uniqueness of the shape matrix, and the feature trajectories belonging to an object are linearly dependent. Then the "same motion" is defined as belonging to a low-dimensional subspace, and trajectories lying in the same subspace are regarded as belonging to the same object. Many developments of the factorization methods are made by the following researchers [14, 3, 24, 5, 7]. The factorization methods can group together points moving with a consistent "behavior" over a long period of time, because they use the full temporal trajectory of every tracked point [24]. However, current factorization methods often fail to segment motions between only two frames [8].

In this paper, we investigate an efficient way of measuring 3D motion consistency using the 2D image motion between just two frames. The proposed criterion can be used for measuring the quality of segmented groups. Moreover, it is able to estimate the 3D structure in an efficient way.

## 3 Background

### 3.1 3D rigid body motion

When a rigid object is moving in 3D space, all the points on the object will follow the same motion model. It can be modeled as a combination of 3D rotation and translation. Suppose a point moves from position $\boldsymbol{p} = [X, Y, Z]^\mathsf{T}$ to $\boldsymbol{p'} = [X', Y', Z']^\mathsf{T}$, then

$$\boldsymbol{p'} = R\boldsymbol{p} + \boldsymbol{t} \tag{1}$$

where $R$ is a 3D rotation matrix and $\boldsymbol{t} = [t_1, t_2, t_3]^\mathsf{T}$ is a translation vector carrying the displacements in the directions of the three axes. The matrix $R$ can be parameterized as: $R = R_z(\varphi_z)R_y(\varphi_y)R_x(\varphi_x)$. where $\varphi_z \in (-\pi, \pi]$, $\varphi_y \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, and $\varphi_x \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ are yaw, pitch, and roll angles respectively. For convenience and conciseness we shall write $s_x = \sin\varphi_x, s_y = \sin\varphi_y, s_z = \sin\varphi_z$ and $c_x = \cos\varphi_x, c_y = \cos\varphi_y, c_z = \cos\varphi_z$. So:

$$R = \begin{bmatrix} c_z & -s_z & 0 \\ s_z & c_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_y & 0 & s_y \\ 0 & 1 & 0 \\ -s_y & 0 & c_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_x & -s_x \\ 0 & s_x & c_x \end{bmatrix} \tag{2}$$

We shall write $r_{ij}$ to denote the entry in row $i$ and column $j$ of $R$.

### 3.2 Projections

The physical camera projects the 3D world onto a 2D image plane through some projection mechanism. Accurate knowledge of this projection mechanism may in principle be used to provide 3D information for understanding the images. However, in practice the lens system in a real camera is too complex to perform 3D reconstruction for. Instead, approximate camera models are developed for different applications, starting from corresponding simplifying assumptions.

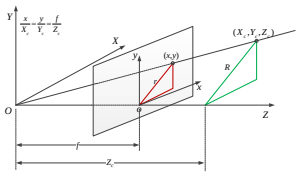


Fig. 1: Perspective camera model

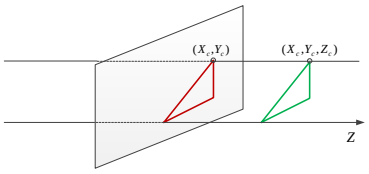


Fig. 2: Orthographic camera model

**General perspective projection** General perspective projection is an idealized mathematical model for a real camera, which is widely used in computer vision applications. It assumes that the camera is sufficiently small compared to the viewed scenes and objects.

Fig. 1 shows the simplest central-projection camera: the pinhole camera model. The $XYZ$ coordinate frame is centered at the camera, with the $Z$-axis being the principal axis of the camera. The projected image plane coincides with the focus plane, and employs the $xy$ coordinate frame. The origin of this image frame, $o$, is the projection of the camera center $O$ on the image plane; their distance is indicated by $f$. A point $\boldsymbol{X_c} = [X_c, Y_c, Z_c]^\mathsf{T}$ in the camera frame, is mapped to the point $\boldsymbol{x} = [x, y]^\mathsf{T}$ in the image frame by

$$\boldsymbol{x} = \frac{f}{Z_c} P \boldsymbol{X_c} \qquad \text{where } P = \left[\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{smallmatrix}\right] \tag{3}$$

**Orthographic projection model** If the camera is sufficiently far away from the viewed scene, one may assume an infinite focal length. Then the points in the camera frame are mapped to the image frame by parallel projection, as illustrated in Figure 2. For this orthographic projection model, the coordinate mapping takes the form:

$$\boldsymbol{x} = P \boldsymbol{X_c} \tag{4}$$

## 4 3D motion consistency

To analyze 3D motion consistency, we address the situation in which we have a given set of *matched pairs of feature points* from two image frames. We aim to find a 3D rigid body motion consistent with all those matched pairs. Combining such a 3D motion with the camera projection mechanism, we can set up an equation relating the coordinates for each matched pair. With sufficiently many points from the same object, an overdetermined system of equations will be obtained. Due to the rigid body motion assumption, this system will have certain structural properties. By using matrix factorization techniques we then can analyze how to recover a 3D rigid body motion in the best possible way.

### 4.1 Theorems

Consider a point on an object undergoing a rigid body motion. Suppose it moves, in the camera coordinate system, from some position $\boldsymbol{p}$ at time $t$ to another

position $\boldsymbol{p'}$ at time $t'$. Then according to Eqn. (1) we have that $\boldsymbol{p'} = R\boldsymbol{p} + \boldsymbol{t}$, where $R$ is a rotation matrix and $\boldsymbol{t}$ a translation vector. The translation vector can be eliminated by working relative to a selected point for which the movement is known (e.g., a center of mass or any other point on the object): if it also holds that $\boldsymbol{p'_0} = R\boldsymbol{p_0} + \boldsymbol{t}$, then

$$\boldsymbol{p'} - \boldsymbol{p'_0} = R(\boldsymbol{p} - \boldsymbol{p_0}). \tag{5}$$

If the scene is far away from the camera, and focal length is small compared to the distance of the object to the camera, then for every two points $\boldsymbol{p}$ and $\boldsymbol{p'}$ at distances $Z_c$ and $Z'_c$, we can assume $\frac{f}{Z_c} \approx \frac{f}{Z'_c}$. Hence, we can ignore the effect of the scaling factor $\frac{f}{Z_c}$ and the camera projection can be approximated by an orthographic projection. Thus the point $\boldsymbol{p}$ at position $[x, y, z]^\mathsf{T}$ in the camera frame is mapped (up to a fixed factor) to position $[x, y]^\mathsf{T}$ in the image frame. The following theorems apply, subject to this orthographic projection assumption. The general situation is discussed in Section 4.3.

**Theorem 1.** *A set of $m + 1$ matched pairs of 2D points $(x_i, y_i)^\mathsf{T}$ and $(x'_i, y'_i)^\mathsf{T}$ (with $i = 0, \ldots, m$) can consistently be interpreted as the 2D coordinates of orthographic projections onto the image plane of $m + 1$ pairs of 3D points in camera space which are related by a single 3D rigid body motion, if and only if the $m \times 4$ data matrix*

$$M = \begin{bmatrix} \tilde{x}_1 & \tilde{y}_1 & \tilde{x}'_1 & \tilde{y}'_1 \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{x}_m & \tilde{y}_m & \tilde{x}'_m & \tilde{y}'_m \end{bmatrix} \tag{6}$$

*where $\tilde{x}_i = x_i - x_0$, $\tilde{y}_i = y_i - y_0$, $\tilde{x}'_i = x'_i - x'_0$ and $\tilde{y}'_i = y_i - y_0$, has a nontrivial null space containing a vector $\boldsymbol{v}$ of which the four entries satisfy*

$$v_1^2 + v_2^2 = v_3^2 + v_4^2. \tag{7}$$

Typically, we will be interested in situations with sufficiently many matched pairs of data points (i.e., $m \geq 4$), for which non-rigid motions would otherwise produce the trivial null space $\{0\}$. For rigid body motion, the non-trivial null space of $M$ will normally be of dimension 1, unless the rigid body motion is of a special type. The following theorem addresses the nature of the family of rigid body motions consistent with such data.

**Theorem 2.** *If the condition under Theorem 1 is satisfied, then there exists a family of rigid body motions, consistent with the data, having at least one real degree of freedom for the translation (corresponding to an arbitrary translation in the z-direction) and at least one real degree of freedom for the 3D rotation.*

***Proof*** From 2 and 5, we have

$$\begin{bmatrix} \tilde{x}'_1 \ldots \tilde{x}'_m \\ \tilde{y}'_1 \ldots \tilde{y}'_m \end{bmatrix} = \begin{bmatrix} c_z & -s_z \\ s_z & c_z \end{bmatrix} \begin{bmatrix} c_y & s_y s_x & s_y c_x \\ 0 & c_x & -s_x \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \ldots \tilde{x}_m \\ \tilde{y}_1 \ldots \tilde{y}_m \\ \tilde{z}_1 \ldots \tilde{z}_m \end{bmatrix} \tag{8}$$

Every row $\tilde{x}'_i$ is a linear combination of $\tilde{x}_i$, implying that $M$ has a rank of at most 3. We can rewrite the equation in terms of $M$:

$$M \begin{bmatrix} c_y & 0 \\ s_y s_x & c_x \\ -c_z & s_z \\ -s_z & -c_z \end{bmatrix} = \begin{bmatrix} \tilde{z}_1 \\ \vdots \\ \tilde{z}_m \end{bmatrix} \begin{bmatrix} -s_y c_x & s_x \end{bmatrix} \tag{9}$$

Multiplying the result by $\left[\begin{smallmatrix} c_x & s_x \\ -s_y s_x & s_y c_x \end{smallmatrix}\right]$:

$$M \begin{bmatrix} c_y c_x & c_y s_x \\ 0 & s_y \\ -c_z c_x - s_z s_y s_x & -c_z s_x + s_z s_y c_x \\ -s_z c_x + c_z s_y s_x & -s_z s_x - c_z s_y c_x \end{bmatrix} = \begin{bmatrix} \tilde{z}_1 \\ \vdots \\ \tilde{z}_m \end{bmatrix} \begin{bmatrix} -s_y & 0 \end{bmatrix} \tag{10}$$

If $s_y = 0$, we have a nontrivial $\boldsymbol{v} = [1, 0, -c_z, -s_z]^\mathsf{T}$ such that $M\boldsymbol{v} = 0$; and if $s_y \neq 0$, we have a nontrivial vector $\boldsymbol{v} = [c_y s_x, s_y, -c_z s_x + s_z s_y c_x, -s_z s_x - c_z s_y c_x]^\mathsf{T}$ such that $M\boldsymbol{v} = 0$. In both cases $\boldsymbol{v}$ satisfies $v_1^2 + v_2^2 = v_3^2 + v_4^2$. This proves one implication of Theorem 1.

Conversely, if a non-zero vector $\boldsymbol{v} = [v_1, v_2, v_3, v_4]^\mathsf{T}$ is given in the kernel of $M$ which happens to satisfy $v_1^2 + v_2^2 = v_3^2 + v_4^2$, we can proceed by the following two cases with respect to the value of $v_2$:

*Case 1.* Assume that $v_2 \neq 0$. Then let $\varphi_y$ have an arbitrary nonzero value in the interval $\left(-\arctan\left|\frac{v_2}{v_1}\right|, \arctan\left|\frac{v_2}{v_1}\right|\right)$. Next, compute $\varphi_x = \arcsin\left(\frac{v_1}{v_2}\tan(\varphi_y)\right)$. Let $\lambda = \frac{v_2}{\sin\varphi_y}$ be a scaling factor, which is nonzero. Then $v_2 = \lambda s_y$ and $v_1 = \lambda c_y s_x$. Consequently $\lambda$ can be computed from $v_1^2 + v_2^2 = \lambda^2(1 - c_y^2 c_x^2)$.

Note that $\left[\begin{smallmatrix} v_3 \\ v_4 \end{smallmatrix}\right] = \lambda \begin{bmatrix} -c_z & s_z \\ -s_z & -c_z \end{bmatrix} \left[\begin{smallmatrix} s_x \\ s_y c_x \end{smallmatrix}\right]$ should hold, which can be rewritten in terms of $c_z$ and $s_z$: $\begin{bmatrix} -v_3 & -v_4 \\ -v_4 & v_3 \end{bmatrix} \left[\begin{smallmatrix} c_z \\ s_z \end{smallmatrix}\right] = \lambda \left[\begin{smallmatrix} s_x \\ s_y c_x \end{smallmatrix}\right]$. The values of $s_z$ and $c_z$ are obtained, which uniquely specify $\varphi_z \in (-\pi, \pi]$.

*Case 2.* Assumes that $v_2 = 0$. Then let $\varphi_y = 0$ and note that $v_1 \neq 0$. Now choose $\varphi_x$ to have an arbitrary nonzero value in the interval $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$. Then set $\lambda = v_1$, $\varphi_z$ is determined through $\left[\begin{smallmatrix} v_3 \\ v_4 \end{smallmatrix}\right] = \lambda \begin{bmatrix} -c_z \\ -s_z \end{bmatrix}$. It follows that $v_1^2 + v_2^2 = v_3^2 + v_4^2 = \lambda^2$.

In either of the two cases 1 and 2, a nonzero scaling factor $\lambda$ and suitable values for $\varphi_z$, $\varphi_y$ and $\varphi_x$ are obtained which make that the vector $\boldsymbol{v}$ is of the form $\boldsymbol{v} = \lambda \begin{bmatrix} c_y s_x \\ s_y \\ -c_z s_x + s_z s_y s_x \\ -s_z s_x - c_z s_y c_x \end{bmatrix}$ or simplified form $\boldsymbol{v} = \lambda \begin{bmatrix} 1 \\ 0 \\ -c_z \\ -s_z \end{bmatrix}$ when $s_y = 0$.

In *Case 1* (where $s_y \neq 0$), this allows one to construct a corresponding vector $(\tilde{z}_1, \ldots, \tilde{z}_m)^\mathsf{T}$ to satisfy the required identity. Because $\left(\begin{smallmatrix} c_x & s_x \\ -s_y s_x & s_y c_x \end{smallmatrix}\right)$ is invertible, $(\tilde{z}'_1, \ldots, \tilde{z}'_m)^\mathsf{T}$ can be obtained by reconsidering the omitted third row of $R$. Clearly, translations in the $z$-direction cannot be observed at all, while coordinate values in all directions can only be obtained relative to an arbitrarily chosen origin. For $z_0$ and $z'_0$ one can introduce arbitrary values, which shows that the entry $t_3$ of translation vector $\boldsymbol{t}$ is completely free.

In *Case 2* (where $\varphi_y = 0$), both columns in the matrix following $M$ in Equation 10 are collinear; both columns are of the form $k(1, 0, -c_z, -s_z)^\mathsf{T}$ (because $c_y = 1$, and $k$ can be either $c_x$ or $s_x$).

With $\varphi_x$ from the indicated range (which ensures that $c_x$ and $s_x$ are both nonzero), we have that both columns are collinear, and that the relationship in Equation 10 is properly satisfied. However the matrix $\left(\begin{smallmatrix} c_x & s_x \\ -s_y s_x & s_y c_x \end{smallmatrix}\right)$ is no longer invertible, so to rewind our steps, we should reconsider Equation 9. With $s_x \neq 0$ it follows that:

$$\begin{bmatrix} \tilde{z}_1 \\ \vdots \\ \tilde{z}_m \end{bmatrix} = \frac{1}{s_x} M \begin{bmatrix} 0 \\ c_x \\ s_z \\ -c_z \end{bmatrix} \tag{11}$$

Then we can proceed as in Case 1 to construct a rotation and translation which is consistent with the given observed data. This proves the converse implication of Theorem 1.

It also proves Theorem 2 upon noting that in both *Cases 1* and *2* a real degree of freedom for $R$ (for the angles $\varphi_y$ and $\varphi_x$, respectively) and for the coordinate $t_3$ was encountered. In special cases, i.e., when the rank of $M$ is less than 3, more degrees of freedom may occur.

### 4.2 Applicability of theoretical results

**Consistency with 3D rigid body motion** In computer vision applications, motion based image segmentation is an important and fundamental topic. The aim is to partition visual elements (pixels or feature points) into groups, based on their motion features. Segmentation algorithms are used in tasks like object detection and tracking, where objects are represented by groups of points (or pixels). For videos from a monocular camera, the key challenge of motion segmentation is to segment the points w.r.t. their 3D motions, while only 2D projection-coordinates of points are available.

Theorem 1 can be used to determine whether the movements of a group of 2D points (matched point from consecutive images) are consistent with a 3D rigid body motion. Giving $m + 1$ pairs of points, we can decompose the $m \times 4$ data matrix $M$ using the SVD:

$$M = UDV^\mathsf{T} \tag{12}$$

in which $U$ is an $m \times m$ orthogonal matrix, $V$ is a $4 \times 4$ orthogonal matrix, and $D = \operatorname{diag}\{d_1, d_2, d_3, d_4\}$ is an $m \times 4$ diagonal matrix with entries $d_1 \geq d_2 \geq d_3 \geq d_4 \geq 0$ on its main diagonal. Theorem 1 establishes that at least $d_4 = 0$ should hold if the movement of 2D points is consistent with a 3D rigid body motion. However, when working with real data, deviations may occur for various reasons, such as inaccuracies in feature extraction and motion detection. Moreover, the orthographic projection hypothesis - which disregards the perspective - is an approximation.

The value of $d_4$ can be taken as a measure for the (lack of) quality of 3D rigid body motion consistency, for the group of points being analyzed. According to Theorem 1, in case of a rigid 3D body motion, every vector $\boldsymbol{v}$ in the kernel satisfies $v_1^2 + v_2^2 = v_3^2 + v_4^2$ if $d_3 > 0$. This property can be also used as a quality measure for the rigid body motion consistency. Note that a vector $\boldsymbol{v}$ is obtained as the last column of matrix $V$ if $d_4 = 0$ and $d_3 > 0$.

**Reconstruction of 3D rigid body motion** Theorems 1 and 2 also enable us to estimate the parameters of a 3D rigid body motion for a given set of matched pairs. Starting from data matrix $M$ (Eqn.6) with a 1-dimensional null space, using Equation 7, there will be one real degree of freedom when computing the 3D rotations $\varphi_z$, $\varphi_y$, $\varphi_x$. There is also one degree of freedom (the translation in the $z$ direction) in determining $\boldsymbol{t}$. However, the values $\tilde{z}_1, \ldots, \tilde{z}_m$ completely depend on the degree of freedom for the 3D rotation.

We may determine the value of $\varphi_y$ or $\varphi_x$ by minimizing a criterion function, such as the sum of squares of values $\tilde{z}_1, \ldots, \tilde{z}_m$. The idea is that the norm of the vector of changes in the (unobserved) $z$-direction, consistent with the computed rotation and translation, is minimized so that no unnecessarily large deviations are included in the rigid body motion.

### 4.3 Error analysis

The proposed theorems are based on the orthographic projection, which is an approximation of the perspective projection. In this subsection, we analyse the errors of orthographic projection w.r.t. to the perspective projection.

Suppose a 3D point is moving from $(X_c, Y_c, Z_c)^\mathsf{T}$ at time $t$ to $(X'_c, Y'_c, Z'_c)^\mathsf{T}$ at time $t'$, and two images are captured at the two time points. The coordinates of a projected point at time $t$ and $t'$ under the perspective projections are $(x_p, y_p)^\mathsf{T}$ and $(x'_p, y'_p)^\mathsf{T}$ respectively. The coordinates of the same projected point under orthographic projection are $(x, y)^\mathsf{T}$ and $(x', y')^\mathsf{T}$. According to the Equations 4 and 3: $[x, y]^\mathsf{T} = [X_c, Y_c]^\mathsf{T}$, $[x_p, y_p]^\mathsf{T} = \frac{f}{Z_c}[X_c, Y_c]^\mathsf{T}$, $[x', y']^\mathsf{T} = [X'_c, Y'_c]^\mathsf{T}$ and $[x'_p, y'_p]^\mathsf{T} = \frac{f}{Z'_c}[X'_c, Y'_c]^\mathsf{T}$. The perspective projection scales the $[X_c, Y_c]^\mathsf{T}$ with a factor $\frac{f}{Z_c}$. We can compensate for the scaling of $[x_p, y_p]^\mathsf{T}$ by multiplying $[x_p, y_p]^\mathsf{T}$ with $\mu = \frac{Z_c}{f}$. So, $[x, y]^\mathsf{T} = \mu[x_p, y_p]^\mathsf{T}$. By applying the same scaling to $[x'_p, y'_p]^\mathsf{T}$, we can compute the error caused by orthographic projection:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} - \mu \begin{bmatrix} x'_p \\ y'_p \end{bmatrix} = (1 - \frac{Z_c}{Z'_c}) \begin{bmatrix} X'_c \\ Y'_c \end{bmatrix} \tag{13}$$

If the changes in $z$ direction caused by translation and rotation are small, then $\frac{Z_c}{Z'_c} \approx 1$, and the error is approximately 0.

## 5 Experiments

In this section, we evaluate the applicability of the theorems on synthetic data in subsection 5.1, and subsequently on real video data in subsection 5.2. The

theorems can also be used to estimate the 3D rigid motion of an object with one degree of freedom. We evaluate this aspect in subsection 5.3.

### 5.1 Improving motion segmentation using synthetic data

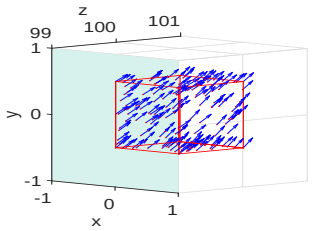


Fig. 3: 3D motion flows

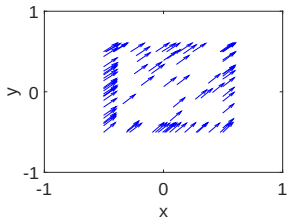


Fig. 4: 2D motion fields

There are two possible ways to apply our results in motion segmentation,

1. Giving the result of a segmentation, an object is represented as a group of points. Usually there are miss-classified points in each group, which make the result in an low precision. We can use the criterion in this paper to find out the miss-classified point in a class of points.
2. Given a group of points that are belonging to an object, and a set of new points without assignments, we can identify whether the new points belong to the object. Failing to identify these points result is a low recall.

We generated a 3D synthetic scene containing a cube, which follows a 3D transformation. Randomly chosen points on the surface of the cube are tracked. We also randomly generate some noise points that have arbitrary 3D motions. The motion of each point is represented by its initial position and the new position after transformation. Figure 3 illustrates the 3D motion flows of the points on the cube surface in camera space, while Figure 4 shows the orthographic projection of these motion vectors on the image that is parallel to the xy plane.

For the first experiment, we choose 100 points from the cube object and $n$ noise points. We aim at allocating these points into two subgroups: the "objects" and "noises" using the criterion in this paper. The accuracy is defined as the percentage of points that are successfully classified, which is illustrated in Fig. 5 w.r.t. different noise ratios (i.e. the percentage of noise points in the mixture set). We compare our method with a state-of-art method—sparse subspace clustering (SSC) [6], whose result is illustrate as the blue line in Fig. 5. The result shows that our method is more stable than SSC.

For the second experiment, $m$ ($m \in [4, 100]$) points on the cube are chosen to represent the object, which is used to determine the classification of 200 other points (half from the object and half from the noises) based on the criterion in our paper. We computed the error using $error = \left\| d_4 + \sqrt{v_1^2 + v_2^2 - v_3^2 - v_4^2} \right\|$. We assigned a point to the group if its error is lower than a threshold value (which is 0.005 in this experiment). We compared it with another error estimation method, which is computed by the error to the affine motion model that is estimated using the

$m$ points represent the object [26]. The performance is evaluated by the accuracy of correctly classifying the undetermined points, as shown in Fig. 6.

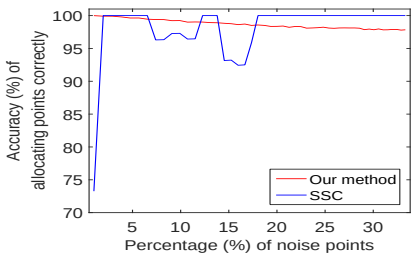


Fig. 5: Identifying the mis-classified points in a group.

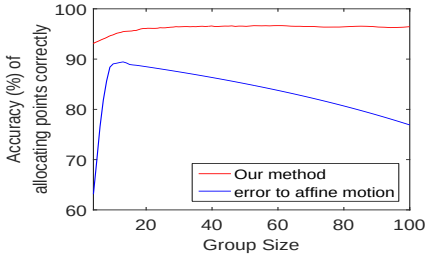


Fig. 6: Determining whether a new point belonging to an object.

### 5.2 Improving motion segmentation using video data

In this experiment, we used the real video sequences from the Hopkins155 benchmark data set [17], for which the feature points on the objects' surfaces and their motions are provided. We chose 25 video sequences from the category named "checkerboard". Each video contains 29 frames, which records a scene with 3 objects following distinct 3D motions (rotation and translation). There are 75 objects in total. In each experiment we chose one object and used the motion vectors between the frame pair $\{f_1, f_i\}$ $(i \in [2, 29])$. We computed the average accuracy of finding the misclassified points from a group over all objects w.r.t. different frame pairs and noise ratio, which is illustrated in Fig. 7. For the second experiment, we computed the average accuracy of allocating a point to a group of points (which represent an object), w.r.t. the group size (i.e. the number of points in the group) and the distance between frames, as shown in Fig. 8.

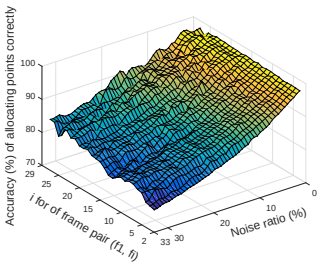


Fig. 7: The accuracy of finding mis-classified points as function of the noise ratio and the distance between two frames.

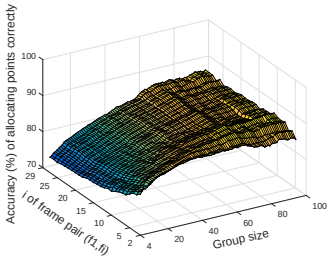


Fig. 8: The accuracy of allocating a point correctly to an object w.r.t. the group size of the object and the distance between two frames.

### 5.3 Recovering the 3D rigid body motion

*Exp.* 5 addressed the reconstruction of a 3D rigid body motion. We investigated whether it is possible to handle the one degree of freedom for the 3D rotation by

minimizing the sum of square of $\tilde{z}_1, \ldots, \tilde{z}_m$. Our initial experiments with points on the surfaces of a cube in the synthetic scene showed that for randomly chosen rotations smaller than $\pi/4$ rad, we can recover the rotation angles $\varphi_x$, $\varphi_y$ and $\varphi_z$ with average accuracies of 74.3%, 74.3%, 94.6% respectively.

## 6 Conclusion

This paper presented two theorems specifying properties of a 2D projection of a 3D rigid body movement. The theorems state that the data matrix of 2D projection of points on a 3D rigid body making a 3D movement, has a non-trivial kernel with a specific structure. The theorems also show that we can reconstruct the original 3D body movement with one degree of freedom for the translation in $z$-direction and one degree of freedom for the 3D rotation.

We used the theorems to measure the 3D rigid motion consistency of a group of 2D projection points. It can achieve above 95% accuracy in identifying misclassified points when the rate of the misclassified points is below 10%, and remains around 90% when the noise rate increases to 20%. We also used the theorems to determining whether new points belong to a known object. If we known more than 50 points belonging to the object, new points can be classified with an accuracy around 90%. These results suggest that the theorems can be used to improve the segmentation accuracy of existing motion segmentation algorithms.

Recovering the 3D rotation angle of a moving object has also been evaluated. The initial results are promising but further research is required.

## References


1. Altunbasak, Y., Eren, P.E., Tekalp, A.M.: Region-based parametric motion segmentation using color information. Graphical models and image processing 60(1), 13–23 (1998)
2. Borshukov, G.D., Bozdagi, G., Altunbasak, Y., Tekalp, A.M.: Motion segmentation by multistage affine classification. IEEE Transactions on Image Processing 6(11), 1591–1594 (1997)
3. Boult, T.E., Brown, L.G.: Factorization-based segmentation of motions. In: Visual Motion, 1991., Proceedings of the IEEE Workshop on. pp. 179–186. IEEE (1991)
4. Bovik, A.C.: Handbook of image and video processing. Academic press (2010)
5. Costeira, J., Kanade, T.: A multi-body factorization method for motion analysis. In: Computer Vision, 1995. Proceedings., Fifth International Conference on. pp. 1071–1076. IEEE (1995)
6. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 2790–2797. IEEE (2009)
7. Gruber, A., Weiss, Y.: Multibody factorization with uncertainty and missing data using the em algorithm. In: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. vol. 1, pp. I–I. IEEE (2004)
8. Gruber, A., Weiss, Y.: Incorporating non-motion cues into 3d motion segmentation. In: European Conference on Computer Vision. pp. 84–97. Springer (2006)

9. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521540518, second edn. (2004)
10. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
11. Horn, B.K., Schunck, B.G.: Determining optical flow. In: 1981 Technical Symposium East. pp. 319–331. International Society for Optics and Photonics (1981)
12. Jian, Y.D., Chen, C.S.: Two-view motion segmentation with model selection and outlier removal by ransac-enhanced dirichlet process mixture models. International Journal of Computer Vision 88(3), 489–501 (2010)
13. Jodoin, P.M., Pierard, S., Wang, Y., Van Droogenbroeck, M.: Overview and benchmarking of motion detection methods (2014)
14. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. International Journal of Computer Vision 9(2), 137–154 (1992)
15. Torr, P.H., Szeliski, R., Anandan, P.: An integrated bayesian approach to layer extraction from image sequences. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(3), 297–303 (2001)
16. Torr, P.H., Zisserman, A.: Concerning bayesian motion segmentation, model averaging, matching and the trifocal tensor. In: European Conference on Computer Vision. pp. 511–527. Springer (1998)
17. Tron, R., Vidal, R.: A benchmark for the comparison of 3-d motion segmentation algorithms. In: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. pp. 1–8. IEEE (2007)
18. Ullman, S.: The interpretation of structure from motion. Proceedings of the Royal Society of London B: Biological Sciences 203(1153), 405–426 (1979)
19. Vidal, R., Soatto, S., Ma, Y., Sastry, S.: Segmentation of dynamic scenes from the multibody fundamental matrix. Urbana 51(61801), 1–2
20. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. International journal of computer vision 103(1), 60–79 (2013)
21. Wang, J.Y., Adelson, E.H.: Layered representation for motion analysis. In: Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93., 1993 IEEE Computer Society Conference on. pp. 361–366. IEEE (1993)
22. Weiss, Y.: Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In: Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on. pp. 520–526. IEEE (1997)
23. Yuan, C.: Motion segmentation and dense reconstruction of scenes containing moving objects observed by a moving camera. ProQuest (2007)
24. Zelnik-Manor, L., Machline, M., Irani, M.: Multi-body factorization with uncertainty: Revisiting motion consistency. International Journal of Computer Vision 68(1), 27–41 (2006)
25. Zhang, J., Shi, F., Liu, Y.: Motion segmentation by multibody trifocal tensor using line correspondence. In: Pattern Recognition, 2006. ICPR 2006. 18th International Conference on. vol. 1, pp. 599–602. IEEE
26. Zhao, W., Roos, N.: Motion based segmentation for robot vision using adapted em algorithm. In: Proceedings of the 11th International Conference on Computer Vision Theory and Applications (VISIGRAPP 2016). pp. 649–656 (2016)
27. Zhao, W., Roos, N.: An em based approach for motion segmentation of video sequence. In: 24th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG 2016. pp. 61–69 (2016)