

CHANGE-POINT DETECTION ON THE LIE GROUP $SE(3)$ FOR SEGMENTING GESTURE-DEFINED SPATIAL RIGID MOTION

Loic Merckel and Toyoaki Nishida

Graduate School of Informatics, Kyoto University, Japan

Keywords: Change-point detection, Lie group, Rigid body motion segmentation, Special Euclidean group, Exponential map.

Abstract: Common CAD interfaces for editing spatial motion of virtual objects, which includes both position and orientation information, are often hampered by complexity and lack of intuitiveness. As the demand for motion data is increasing, e.g., in computer graphics or mixed reality, the development of new interfaces that offer a natural means of specifying arbitrary motion becomes essential. A solution consists in relying on live motion capture systems to record user's gestures through space. In this context, we present a novel method for discovering change-points in a time series of elements in the set of rigid-body motion in space $SE(3)$. The goal is to segment gesture-defined motion with in mind the development of a method for enhancing the user's intent. Although numerous change-points detection techniques are available for dealing with scalar, or vector, time series, the generalization of these techniques to more complex structures may require overcoming difficult challenges. The group $SE(3)$ does not satisfy closure under linear combination. Consequently, most of the statistical properties, such as the mean, cannot be properly estimated in a straightforward manner. We present a method that takes advantage of the Lie group structure of $SE(3)$ to adapt a difference of means method. Especially, we show that the change-points in $SE(3)$ can be discovered in its Lie algebra $se(3)$ that forms a vector space. The performance of our method is evaluated through both synthetic and real data.

1 INTRODUCTION

The growing progress in capturing motions, including both position and orientation data, has motivated some initiatives (e.g., (Merckel and Nishida, 2009)) to develop new interfaces, for generating hand-defined motions, as an alternative to the conventional, and overwhelming, WIMP-based interfaces (van Dam, 1997) of CAD software. An important application is to provide a wide range of users with an effective means of creating animated 3D contents for Mixed Reality (MR) environments. In this vein, (Merckel and Nishida, 2009) present a hand-held MR interface, which consists of a tablet PC equipped with a six-degree-of-freedom (6-DOF) orientation and position sensor, for animating 3D virtual items. To do so, the user naturally describes the 6-DOF trajectory in space by moving the hand-held system as if it was the item.

Such a technique suffer from a two-fold drawback. First, the motion capture process is, to some extent, limited in precision and may contain noise. Second, user's inputs are hampered by what (Sezgin and Davis, 2004) refer to as "*imperfect motor control*",

i.e., the user's movements (gestures) do not strictly reflect what the user intends. Note that some modern motion capture systems has become fairly robust and accurate, and consequently, the former problem might be quasi-negligible compared with the latter one (e.g., some quantitative results are presented by (Fiorentino et al., 2003) concerning a 3D position optical tracker). As a result, when motion is defined via hand-simulating the movements in space, the user may encounters some setbacks for expressing her/his intention.

In a similar spirit, 2D and 3D input devices have been extensively employed for drawing curves (e.g., see (Fiorentino et al., 2003) and references therein). These attempts have to cope with the same issues. A typical scheme, for addressing those flaws, consists in splitting the input curve into primitives, and then, inferring the user's intent in order to enhance each segment ((Qin et al., 2001) and (Fiorentino et al., 2003)). Those curves are a sequence of 2D or 3D points in Euclidean space (\mathbb{R}^2 or \mathbb{R}^3), which is a quite appealing structure for performing a wide variety of data processing algorithms.

The long term goal of our research is to develop an efficient means of amplifying the user's intent during freehand motion definition. Analogously to 3D drawing engines (Fiorentino et al., 2003), the envisaged method consists in discovering some key-points of the motion, then interpolating a smooth trajectory between each consecutive key-points. The goal of the current research is to address the former issue, i.e., to discover key-points in motions. In contrast to planar or spatial curves, a sensor-captured motion results in a discrete time series of displacements, that formally, amounts to a time series of elements in the special Euclidean group of rigid body motion, commonly denoted $SE(3)$. Although an universal definition of key-points is hard to state, a reasonable assumption is to identify key-points with change-points, sometimes referred to as "break-points" (Qin et al., 2001), in the time series. The latter problem, i.e., the interpolation of a smooth motion between key-points, is not discussed in this paper. Nevertheless, it is worth noting that an adaptation of the method introduced by (Hofer and Pottmann, 2004) should fulfill the requirement.

We formulate the problem as a change-points detection problem in time series of elements in $SE(3)$. A major difficulty arises from the particular structure of the group $SE(3)$ that does not satisfy closure under linear combination. Consequently, such a structure sets some serious constraints that prevent numerous of the common time series data processing or mining techniques from being applicable. For example, most of the statistical properties, such as the mean, cannot be properly estimated in a straightforward manner. However, by exploiting the Lie group structure of $SE(3)$, we show how to adapt a *difference of means* method ("which is an adaptation of an image edge detection technique", (Agarwal et al., 2006)). In particular, we show that the change-points on the group $SE(3)$ can be discovered in its associated Lie algebra $se(3)$ that form a vector space. The method discussed by (Agarwal et al., 2006) is suitable only for detecting changes in step functions (i.e., piecewise constant functions). Our adaptation is formulated in a way that does not assume such a simple model, and should perform well with various piecewise-defined functions.

The contribution of the present work lies into two-fold. First, a novel method for detecting change-points in $SE(3)$ is presented and evaluated. Second, an underlying general approach, which can be easily adapted to be applied to various Lie groups is suggested.

The remainder of this paper is organized as follow. In the next section, we discuss about the problematic and related works.

In Section 3, we briefly present the Lie group theory and we describe the structure of the group $SE(3)$. In Section 4 we introduce our method for detecting change-points on $SE(3)$. Then, in Section 5, we propose a set of evaluations using both synthetic and real data. Finally, in Section 6, we summarize our key points.

2 DISCUSSIONS AND RELATED WORKS

The detection of change-points in time series, which consists in partitioning the time series in homogeneous segments (in some sense), is an important issue in several domains ((Basseville and Nikiforov, 1993), (Ide and Inoue, 2005), (Agarwal et al., 2006)). Consequently, numerous attempts at solving this problem exist ((Basseville and Nikiforov, 1993), (Moskvina and Zhigljavsky, 2003), (Ide and Tsuda, 2007), (Gombay, 2008), (Kawahara and Sugiyama, 2009)). However, most of the existing techniques apply only to scalar, or, for certain, vector time series. Furthermore, as pointed out in the introduction section, some of these methods are restricted to be performed only with time series that follow simple models ((Basseville and Nikiforov, 1993), (Agarwal et al., 2006)). Therefore, these methods may not provide us with a suitable solution to deal with time series of elements in more elaborated structures.

The difference of means method (Agarwal et al., 2006) is relying only on linear operators, and thus, should be easily extended to vector spaces, such as the set of real matrices ($\mathbb{R}^{n \times n}$), in which the notions of mean and distance exist. Still, considering a general metric group structure, the difficulty remains as the closure under linear operators may not hold. However, restricting our attention to metric Lie groups extends the possibilities. One of the great particularity of this class of groups is the local approximation of their structure by the tangent space, which, at the identity, is a Lie algebra forming a vector space. Maps from the group to its algebra, and inversely, exist in a neighborhood of the identity, and are referred to as the logarithmic and the exponential maps (respectively). This particular nature of Lie groups provide a means of extending certain methods relying on linear operations to non-linear groups.

For example, (Lee and Shin, 2002) extend the concept of Linear Time-Invariant (LTI) filters to orientation data (e.g., quaternion group). The same approach is used in the work of (Courty, 2008) to define a bilateral motion filter. (Tuzel et al., 2005) propose an adaptation of the mean shift clustering technique to

Lie groups. This method has been extended by (Subbarao and Meer, 2006) to suit any analytic manifold. (Fletcher et al., 2003) propose a counterpart of the Principal Component Analysis (PCA) method on Lie groups by defining the concept of principal geodesic curves.

In this paper, we attempt to adapt the difference of means method (Agarwal et al., 2006) to suit motion data. Although that the focus is set on the Lie group of spatial rigid motions, our method is quite generic and can be adapted to numerous Lie groups, especially the subgroups of the general linear group $GL_n(\mathbb{R})$.

In the literature, we found two different approaches to extend the concept of mean to a Lie group (see, e.g., (Buss and Fillmore, 2001), (Srivastava and Klassen, 2002), (Govindu, 2004), (Fletcher et al., 2003)). Both of them are based on the observation that the arithmetic mean in Euclidean spaces is the solution to the equation

$$\bar{x} = \arg \min_x \sum_{i=0}^{n-1} \|x - x_i\|^2. \quad (1)$$

Similarly, the mean of a set of points $\{M_i\}$ in a metric Lie group can be formulated as the point \bar{M} that minimizes the sum of squared distances $d(M, M_i)$. Consequently, the concept of mean relies on the choice of the metric. The first approach, denoted the *extrinsic* mean, utilizes the induced metric of an Euclidean space in which the group is embedded (details are given by (Govindu, 2004), (Fletcher et al., 2003), and references therein). The second approach, referred to as the *intrinsic* mean, consists in choosing the Riemannian distance on $SE(3)$ (*intrinsic* distance). The mean is then defined as follow:

$$\bar{M} = \arg \min_{M \in SE(3)} \sum_{i=0}^{n-1} d^2(M - M_i). \quad (2)$$

Employing this definition of the mean, the difference of means method could be adapted. The drawback is that, in practice, the computation of \bar{M} by solving directly the equation (2) is quite complex. Alternatively, an iterative algorithm, based on the work of (Buss and Fillmore, 2001), is proposed by (Fletcher et al., 2003). However, this algorithm is still iterative, and thus may required some significant computation time (the iterative process have to be performed for each point of the time series). Furthermore, such an approach would require a piecewise constant-function as a model for the data (as discussed earlier).

Our approach does not compute the mean, but relies on the operation of a mean filter (LTI filter type). The methodology introduced in this paper presents

some similarities with the work of (Lee and Shin, 2002). They suggest a general scheme for applying linear filters to any orientation representation that form a Lie group structure (such as the quaternion group or the rotation group $SO(3)$). We follow this scheme for applying the mean filter (Note that the adaptation of this scheme to the rigid motion group is straightforward).

In practice, live motion is captured via particular equipments. Many trackers give six or seven components (depending on whether it is based on the Euler angles or on Quaternion for parameterizing the rotation). The mapping from the sensor raw data to $SE(3)$, or other structures trivially homeomorphic to $SE(3)$, such as $\mathbb{R}^3 \times SO(3)$, is usually straightforward (and often employed for storing/recording the motion). One could search for changes, a priori, in the generating process of the series (i.e., during the data acquisition). However, such an approach would be hardware-dependent and restricted to only on-line processing. Consequently, existing data could not be treated. Our approach is independent of the source of the data, and can be performed on-line as well as off-line. It can be remarked that purely optical methods for live motion capture may directly output the time series on $SE(3)$ (Drummond and Cipolla, 2002).

Another conceivable approach consists in parameterizing the group, and search for changes in the parameters space (which can be regarded as bringing the problem back to the approach suggested at the previous paragraph). For example, (Grassia, 1998) gives a comprehensive description of several common parameterizations of the rotation group $SO(3)$, which could be employed to parametrize $SE(3)$. Although it is fairly intuitive that a change in the parameters space would correspond to a change in the series, it is not mathematically obvious (some rational and elements of proof would be required for each parameterization). A particular attention should be given to the parameterization scheme to avoid some anomalies that may occur in the parameters space. For example, it has been proven that “it is topologically impossible to have a global 3-dimensional parameterization without singular points for the rotation group” (Stuelpnagel, 1964) (e.g., the Euler angles suffer from the so called gimbal lock). It can be noted here that the exponential map from the Lie algebra to the Lie group is a sort of parameterization of the group, the parameters space being its Lie algebra (Grassia, 1998). In this regards, our approach could fit into this category.

3 THE LIE GROUP $SE(3)$ AND THE LIE ALGEBRA $se(3)$

3.1 General Overview of Lie Groups

3.1.1 Definitions

A Lie group G is a group that is also a differentiable manifold on which the group operations (i.e., noting the binary operation of G , $G \times G \mapsto G$, $(x, y) \rightarrow x \cdot y$ and $G \mapsto G$, $x \rightarrow x^{-1}$) are differentiable.

The tangent space of G at the identity has a structure of Lie algebra \mathfrak{g} , which is a vector space on which the Lie bracket operator (bilinear, anti-symmetric and satisfying the Jacobi identity) is defined.

The exponential map, denoted \exp , is a map from the algebra \mathfrak{g} to the group G (for a formal definition and proof of existence, see (Huang, 2000)). In general, the exponential map is neither surjective nor injective. Nevertheless, it is a diffeomorphism between a neighborhood of the identity I in G and a neighborhood of the identity 0 in \mathfrak{g} . The inverse of the exponential map $\mathcal{N}_G(G) \mapsto \mathfrak{g}$ is denoted \log (logarithmic map).

3.1.2 Matrix Lie Groups

Matrix Lie groups are subgroups of the general linear group $GL_n(\mathbb{R})$, which is the group of invertible matrices (the group operation being the multiplication). The Lie bracket operator is defined as $[A, B] = AB - BA$ and the exponential map by:

$$\exp(V) = \sum_{k=0}^{\infty} \frac{V^k}{k!}. \quad (3)$$

The inverse, i.e., the logarithmic map, is defined as follow:

$$\log(M) = \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k} (M - I)^k, \quad (4)$$

which is well defined only in a neighborhood of the identity (otherwise, the series may diverge).

Matrix Lie groups are Riemannian manifolds, i.e., they possess a Riemannian metric (derived from a collection of inner products on the tangent spaces at every point in the manifold). Let S be a matrix Lie group. The metric $d: S \times S \mapsto \mathbb{R}^+$ such that

$$d(A, B) = \|\log(A^{-1}B)\|_F, \quad (5)$$

with $\|\cdot\|_F$ the Frobenius norm of matrices, is the length of the shortest curve between A and B (this curve is

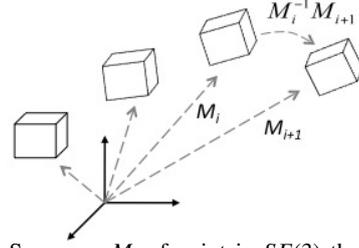


Figure 1: Sequence M_i of point in $SE(3)$ that physically corresponds to a rigid body motion.

referred to as the *geodesics*, whereas its length is the *intrinsic distance*).

3.2 The Special Euclidean Group $SE(3)$

Throughout this paper we consider the special Euclidean group $SE(3)$, which is the matrix Lie group of spatial rigid body motions and is a subgroup of $GL_4(\mathbb{R})$. A general matrix representation has the form

$$SE(3) = \left\{ \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & \mathbf{1} \end{pmatrix} / \mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3 \right\}. \quad (6)$$

The rotation group $SO(3)$ is defined as $\{\mathbf{R} \in \mathbb{R}^{3 \times 3} / \mathbf{R}^T \mathbf{R}^{-1} = I_3, \det(\mathbf{R}) = 1\}$. An element of $SE(3)$ physically represents a displacement, \mathbf{R} corresponds to the orientation, or attitude, of the rigid body while \mathbf{t} encodes the translation (Figure 1).

The Lie algebra $se(3)$ of $SE(3)$ is given by:

$$se(3) = \left\{ \begin{pmatrix} \mathbf{\Omega} & \mathbf{v} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} / \mathbf{\Omega} \in \mathbb{R}^{3 \times 3}, \mathbf{\Omega}^T = -\mathbf{\Omega}, \mathbf{v} \in \mathbb{R}^3 \right\}. \quad (7)$$

The skew-symmetric matrix $\mathbf{\Omega}$ can be uniquely expressed as

$$\mathbf{\Omega} = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}, \quad (8)$$

with $\omega = (\omega_x, \omega_y, \omega_z) \in \mathbb{R}^3$ such that $\forall x \in \mathbb{R}^3, \mathbf{\Omega}x = \omega \times x$. Physically, ω represents the angular velocity of the rigid body, whereas \mathbf{v} corresponds to the linear velocity (Zefran et al., 1998).

(Selig, 2005) presents a closed-form expression of the exponential map (i.e., (3)) and its local inverse (i.e., (4)). The exponential map $se(3) \mapsto SE(3)$ is given by:

$$\exp(V) = I_4 + V + \frac{1 - \cos(\theta)}{\theta^2} V^2 + \frac{\theta - \sin(\theta)}{\theta^3} V^3, \quad (9)$$

where $\theta^2 = \omega_x^2 + \omega_y^2 + \omega_z^2$. Note that it can be regarded as an extension of the well known Rodrigues' formula for rotations (i.e., on the Lie group $SO(3)$).

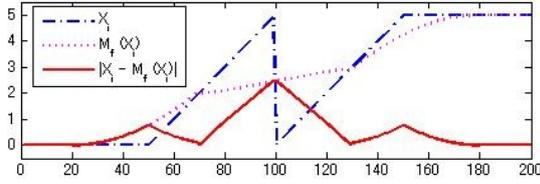


Figure 2: When a change occurs in the time series, $|M_f(X_i) - X_i|$ shows a local maximum.

The logarithmic map $\mathcal{N}_l(SE(3)) \mapsto se(3)$ is yielded by:

$$\log(M) = a(bI_4 - cM + dM^2 - eM^3), \quad (10)$$

with

$$\begin{aligned} a &= (1/8)\csc^3(\theta/2)\sec(\theta/2) \\ b &= \theta\cos(2\theta) - \sin(\theta) \\ c &= \theta\cos(\theta) + 2\theta\cos(2\theta) - \sin(\theta) - \sin(2\theta) \\ d &= 2\theta\cos(\theta) + \theta\cos(2\theta) - \sin(\theta) - \sin(2\theta) \\ e &= \theta\cos(\theta) - \sin(\theta) \end{aligned}$$

and $\text{tr}(M) = 2 + 2\cos(\theta)$. This is valid only for $-\pi < \theta < \pi$.

4 PROPOSED METHOD FOR DETECTING CHANGE-POINT

4.1 Overview of the Method

Let (X_0, \dots, X_{n-1}) be a time series. A simple, but efficient and quite robust technique for discovering the change-points is the *difference of means* method (Agarwal et al., 2006), which is performed only by means of linear operations. The principle is, for each point X_i , to calculate the mean of the N points after X_i (*right mean*), and to calculate the mean of the N points before X_i (*left mean*). The parameter N , the window size, should be carefully selected as mentioned by (Agarwal et al., 2006). Then, the distance d_i between the right and left means of X_i is compared with the other distances yielded by the points in the vicinity of X_i . If d_i is the greatest distance, then X_i is declared as a potential change-point. Some heuristics should be applied to conclude whether or not it is effectively a true positive (see (Agarwal et al., 2006) and below). This technique is hampered by the assumption of a step function as a model for the data (the points where the “steps” are present are then detected).

In order to detect the change-points we adapt the difference of means method. However, we formulate it differently so as to make it suitable for more elaborated models (such as arbitrary piecewise-defined functions). Let M_f be the mean filter and N its mask size. The response for the i^{th} element is given by:

$$M_f(X_i) = \frac{1}{2N+1} \sum_{k=-N}^N X_{i+k}. \quad (11)$$

Our method is based on the observation that if a change occurs at k^* , then $|M_f(X_{k^*}) - X_{k^*}|$ should be a local maximum of the series $(|M_f(X_i) - X_i|)_i$ (Figure 2).

To derive an analogue filter M_f^G (G referring to the group) of M_f to be applied to time series in $SE(3)$, we follow the construction protocol introduced by (Lee and Shin, 2002). The key idea is to interpret each displacement $\log(M_i^{-1}M_{i+1})$ between two consecutive points M_i and M_{i+1} of a series in $SE(3)$ as a linear displacement $V_{i+1} - V_i$ in the algebra $se(3)$. The obtained filter M_f^G remains a “LTI type” filter in terms of properties (the proof given by (Lee and Shin, 2002) is employing a closed-form of the exponential map valid for quaternions, however a proof using (9) is very similar). A point X_k is declared to be a potential change-point if the distance $|M_f(X_k) - X_k|$ is the largest one in a neighborhood of X_k . Analogously, since there is a one-to-one correspondence between a displacement in $se(3)$ and a displacement in $SE(3)$ (Lee and Shin, 2002), M_k is declared to be a potential change-point if $d(M_k, M_f^G(M_k))$ is a local maximum of the series $(d(M_i, M_f^G(M_i)))_i$.

To summarize, the pipeline of the approach consists in transforming the sequence in $SE(3)$ to the vector space $se(3)$ via logarithmic mapping, performing the required linear operations (i.e., applying the mean filter), and finally, interpreting the results back to $SE(3)$ (via exponentiation) for discovering the change-points (Figure 3). In the remainder of this section, we detail each step.

4.2 Transformation between $SE(3)$ and $se(3)$

Let (M_0, \dots, M_{n-1}) be a time series in $SE(3)$. One can remark that

$$\forall i \in \llbracket 1, n-1 \rrbracket, M_i = M_0 \prod_{j=0}^{i-1} M_j^{-1} M_{j+1}. \quad (12)$$

The equality (12) shows that any element of the time series can be regarded as a cumulation of

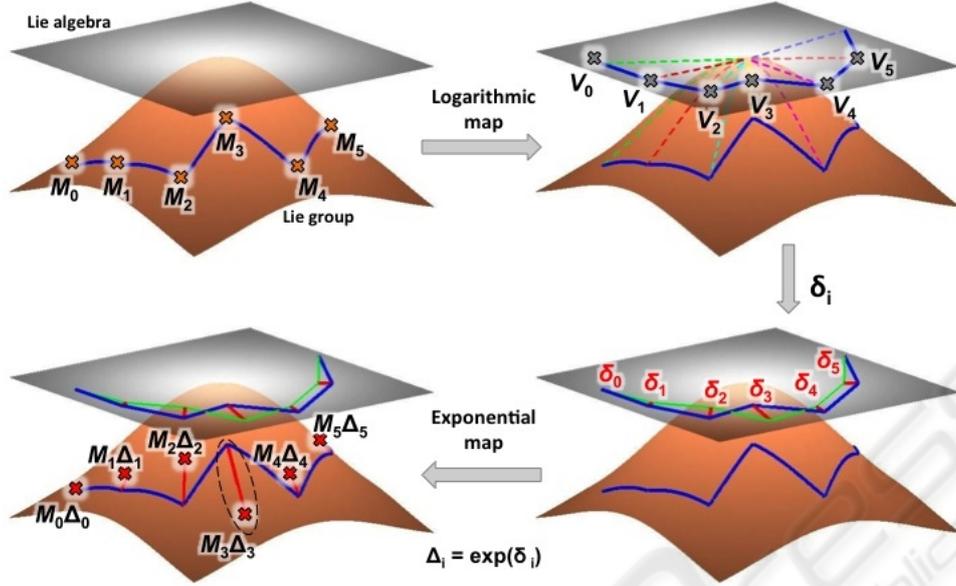


Figure 3: Conceptual view of the change-point detection on $SE(3)$. As discussed in the text, in practice, if the Riemannian metric is used, the set of $\Delta_i = \exp(\delta_i)$ does not need to be calculated.

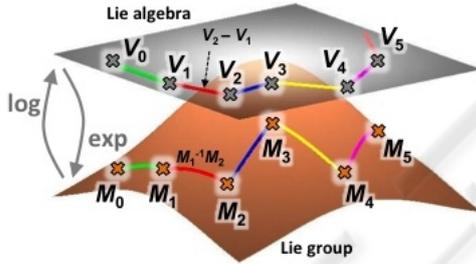


Figure 4: The points $M_i^{-1} M_{i+1} \in SE(3)$ (assumed to be in a neighborhood of I_4) are mapped onto $V_{i+1} - V_i \in se(3)$ by the logarithmic map. The inverse mapping can be achieved by the exponential map.

small displacements $M_j^{-1} M_{j+1}$ from the initial element M_0 . Note that we assume these displacements *small* enough so that

$$\varphi_j = \log(M_j^{-1} M_{j+1}) \quad (13)$$

exists. Equation (12) can then be written

$$\forall i \in \llbracket 1, n-1 \rrbracket, M_i = M_0 \prod_{j=0}^{i-1} \exp(\varphi_j). \quad (14)$$

Similarly to the approach presented by (Lee and Shin, 2002), we construct the following sequence in $se(3)$: given an initial condition V_0 , $\forall i \in \llbracket 0, n-2 \rrbracket$, $V_{i+1} = \varphi_i + V_i$ (Figure 4). We get the two following relations for $i \in \llbracket 1, n-1 \rrbracket$:

$$M_i = M_0 \prod_{j=0}^{i-1} \exp(V_{j+1} - V_j), \quad (15)$$

and

$$V_i = V_0 + \sum_{j=0}^{i-1} \log(M_j^{-1} M_{j+1}). \quad (16)$$

4.3 Application of the Mean Filter

The filter \mathcal{M}_f^G is defined so that $\log(M_i^{-1} \mathcal{M}_f^G(M_i)) = \mathcal{M}_f(V_i) - V_i$, which can be written

$$\mathcal{M}_f^G(M_i) = M_i \exp\left(\frac{1}{2N+1} \sum_{k=-N}^N V_{i+k} - V_i\right). \quad (17)$$

By following a similar development as in the work of (Lee and Shin, 2002), we obtain:

$$\mathcal{M}_f^G(M_i) = M_i \exp(\zeta^R(\varphi_i) - \zeta^L(\varphi_i)), \quad (18)$$

with

$$\zeta^R(\varphi_i) = \sum_{k=0}^{N-1} \frac{N-k}{2N+1} \varphi_{i+k}, \quad (19)$$

and

$$\zeta^L(\varphi_i) = \sum_{k=0}^{N-1} \frac{k+1}{2N+1} \varphi_{i-N+k}. \quad (20)$$

The term $\zeta^R(\varphi_i) - \zeta^L(\varphi_i)$ can thus be interpreted as a difference of weighted means. Let this difference of weighted means be denoted by

$$\delta_i = \zeta^R(\varphi_i) - \zeta^L(\varphi_i). \quad (21)$$

4.4 Change-Point Detection

As previously discussed, the change-points should correspond to local maximums of the series $(d(M_i, \mathcal{M}_f^G(M_i)))_i$. Considering the Riemannian distance, (18) leads to

$$\begin{aligned} d(M_i, \mathcal{M}_f^G(M_i)) &= \left\| \log \left(M_i^{-1} \mathcal{M}_f^G(M_i) \right) \right\|_F \\ &= \left\| \zeta^R(\varphi_i) - \zeta^L(\varphi_i) \right\|_F \\ &= \|\delta_i\|_F. \end{aligned} \quad (22)$$

Equality (22) shows that the value of $\mathcal{M}_f^G(M_i)$ does not need to be explicitly computed. If $\|\delta_i\|_F$ is greater than at any other points in the vicinity of M_i , then M_i is declared as potentially a change-point. Formally, this can be expressed as: For a selected $n \in \mathbb{N}^*$, e.g., $n = N$, if

$$\|\delta_i\|_F > \max_{j \neq i, j \in \llbracket i-n, i+n \rrbracket} \left\{ \|\delta_j\|_F \right\}, \quad (23)$$

then M_i is a candidate for a change-point. Such an approach yields a candidate every $2n$ points, therefore, each candidate must be examined for avoiding false positive. We can adapt the test suggested by (Agarwal et al., 2006). For example, given a candidate M_i , if $\|\delta_i\|_F > x \|\zeta^L(\varphi_i)\|_F$, $x \in]0, 1]$, then M_i is declared to be a valid change-point (the value x have to be empirically selected).

5 RESULTS

In this section, we attempt to evaluate our method. First, we assess the method based on simulation study. Then, we conduct a set of experiments using real data acquired via a motion capture device.

It follows from the previous section that one of the important step of the detection process consists in finding the local maxima of the function $\eta : M_i \in SE(3) \mapsto \|\delta_i\|_F \in \mathbb{R}^+$ (or equivalently, $i \mapsto \|\delta_i\|_F$). In practice, a realistic scenario is that the motion is disturbed by noise (e.g., due to motion capture device imperfection), which consequently, affects the function η . In order to limit the potential detection errors caused by this noise, we perform two different smoothing filters. Those filters are intended to reduce

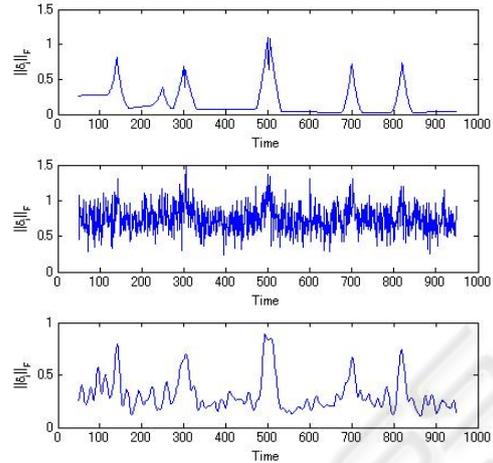


Figure 5: Data smoothing for enhancing the results (these data are obtained from the original signal depicted in Figure 6). Plot of the function $t \mapsto \|\delta_i\|_F \in \mathbb{R}^+$ without noise (top), with strong Gaussian noise added to the original signal (middle), after smoothing both the motion and the function η itself (bottom).

Table 1: Definition of the noise level (NL). Both the position signal and the orientation signal (Figure 6) are disturbed with a Gaussian noise of mean zero and standard deviation σ_T and σ_R , respectively.

	0	1	2	3	4	5	6
σ_T (cm)	0	0.5	1	1.5	2	2.5	3
σ_R ($^\circ$)	0	2	2.5	5	10	20	40

the high frequency components of the data. Therefore, the change-points should not be affected (assuming a Gaussian noise of mean zero, the response of the mean filter \mathcal{M}_f should remain unchanged after attenuating the high frequency components due to noise). Figure 5 illustrates the benefits of this two-steps smoothing.

First, the motion data is smoothed using an adaptation of the orientation filter suggested by (Lee and Shin, 2002). Although we have observed that this filter greatly enhances the motion by significantly reducing additive noise, it is not removed.

Second, to avoid the detection of spurious maxima, we smooth the function η to attenuate the high frequency components. This smoothing operation is performed via the Savitzky-Golay filter. The choice of this filter is motivated by its great property of preserving important features of the signal such as the extrema and the width of peaks.

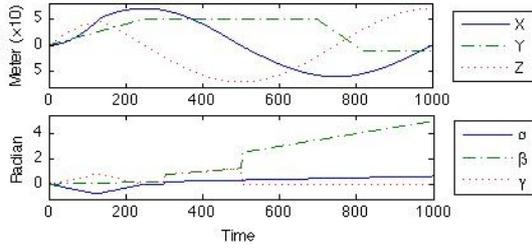


Figure 6: Original signal for generating the simulated motion. the top graph shows the positions, whereas the bottom graph gives the orientation (by means of Euler angles).

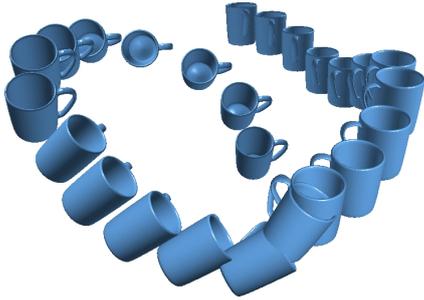


Figure 7: A teacup displayed at each 40 pose of the motion generated by the signal shown in Figure 6.

5.1 Synthetic Data

We generate a synthetic motion (M_1, \dots, M_{1000}) . The original signal is shown in Figure 6, and Figure 7 depicts a visual representation by means of a teacup. By construction, there are a total of six change-points located at index 140, 250, 300, 500, 700 and 820.

5.1.1 Effects of Noise Level and Windows Size

We study the effects of both, (i) the noise level, and (ii) the choice of the window size N , on the change-point detection results in terms of false positives and false negatives. Table 1 gives our definition of seven noise levels to which we refer to throughout the simulations (NL 0, ..., NL 6). A change-point M_n is considered as discovered when the algorithm finds a

Table 2: Results of the change-points detection, depending on the noise level and the window size, in terms of false positive (FP) and false negative (FN). Each cell gives the couple (FP, FN).

	$N = 10$	$N = 20$	$N = 30$	$N = 40$	$N = 50$
NL 0	(0,0)	(0,0)	(0,0)	(1,0)	(1,1)
NL 1	(0,1)	(0,0)	(0,0)	(1,0)	(1,1)
NL 2	(2,1)	(0,0)	(0,0)	(1,1)	(1,1)
NL 3	(19,0)	(1,0)	(0,0)	(0,0)	(0,1)
NL 4	(27,0)	(10,0)	(0,0)	(0,0)	(0,1)
NL 5	(28,1)	(13,0)	(4,0)	(1,0)	(0,1)
NL 6	(28,1)	(10,1)	(4,0)	(3,0)	(4,2)

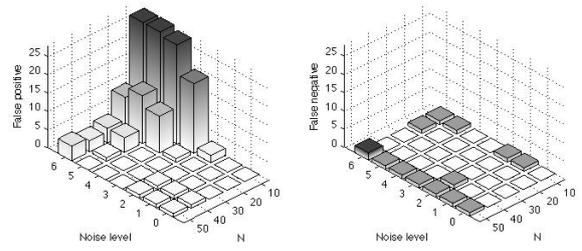


Figure 8: Visual representation of the results summarized in Table 2.

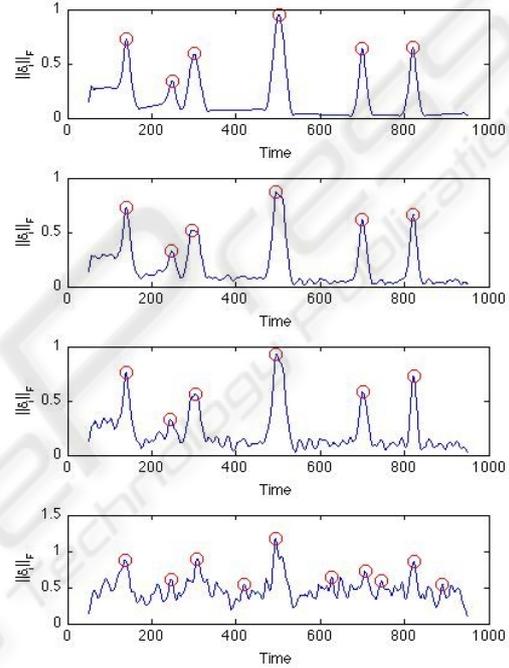


Figure 9: Plot of the function $t \mapsto \|\delta_t\|_F$ for $N = 30$ and (from top to bottom) noise level 0, 1, 3, 6. The circles correspond to the detected change-points.

point in its vicinity, i.e., $M_{n \pm k}$, with $k < 10$. If several points are in the vicinity, then only the nearest is counted as valid (i.e., the other points are considered as false positives).

Table 2 summarizes the results for 35 simulations during which the window size N varies from 10 to 50 by step of 10, whereas the noise level increases from level 0 to level 6. Figure 8 gives a visual representation of the same results.

We can observe that when the window size N is reaching 50, a change-point is systematically missed, independently of the noise level. This result is actually expected for the window size becomes larger than the distance between two subsequent change-points. Thus, only one will be discovered. At the other extremity, when N is small, the method is unstable, and appears to be very sensitive to noise. Especially, a

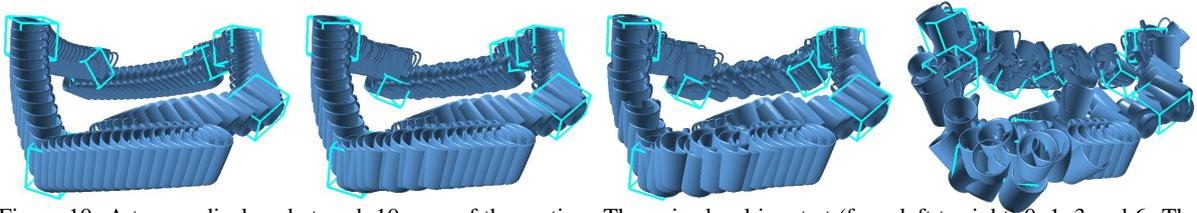


Figure 10: A teacup displayed at each 10 pose of the motion. The noise level is set at (from left to right) 0, 1, 3 and 6. The cups encapsulated in a frame correspond to the discovered change-points.

large number of false positives are yielded as soon as the noise level reaches 3. Alternatively, a moderate window size (e.g., between 30 and 40) provides good results. When the noise level is high, we can observe only few false positives.

Figure 9 shows the plots of the function $t \mapsto \|\delta_t\|_F$ for the window size set at 30 and the noise level set at 0, 1, 3 and 6. Figure 10 gives the corresponding representation of the motion (via a teacup) in which the change-point are emphasized.

Globally, the simulation suggests that, assuming an adequate selection of the window size, one can expect a low number of false positives and a very low, if any, number of false negatives.

5.1.2 Comparison Against SST

Singular Spectrum Transform (SST) is a robust change-point detection method based on the PCA ((Moskvina and Zhigljavsky, 2003), (Ide and Tsuda, 2007)). One of its great advantage compared with various previous attempts is its capability to be applied to analyze “complex” data series (in terms of “shape”) without restrictive assumptions about the data model. For example, it can deal with data series for which the distribution depends on the time, such as arbitrary piecewise-defined functions (e.g., connected affine segments).

Since SST is designed for manipulating only scalar time series, to compare the results obtained by the both SST and our proposed method, first, we apply the SST method to each of the six components of the signal that served for generating our synthetic motion (Figure 6), and second, we consolidate the results to determine the change-points. We use the software written by the authors of the algorithm discussed by (Moskvina and Zhigljavsky, 2003) (which is freely distributed¹). During the experiment we use the setting suggested by the software.

Figure 11 presents the results. We can observe that the performance, in terms of false positive and false negative, of both our method and SST are com-

¹<http://www.cardiff.ac.uk/math/subsites/stats/change-point>

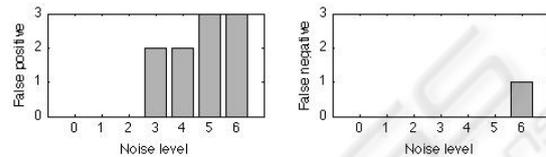


Figure 11: False positive and false negative yielded by the SST method applied to the signal depicted in Figure 6.

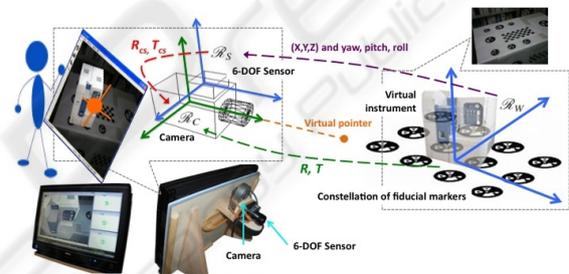


Figure 12: 3D items animation engine.

parable (assuming a proper selection of the window size N , see Table 2). Both methods yield a limited number of false positives when the noise level reach a certain level, with possibly a very few false negatives. This results should be interpreted loosely, for the SST algorithm requires an adequate setting of five parameters. A better tuning of those parameters might lead to improving the performances. However, we observed that the setting suggested by the software usually provide good results.

5.2 Real Data

We have integrated our change-point detection method into a 3D items animation engine, such as the one described by (Merckel and Nishida, 2009), which is an hand-held MR system (Figure 12). It consists of a tablet PC equipped with a video camera and the IS-1200 VisTarcker² (Foxlin and Naimark, 2003), which is a 6-DOF (position and orientation) vision-inertial tracker.

This engine is intended for providing experts of

²Manufactured by InterSense, Inc., <http://www.intersense.com>

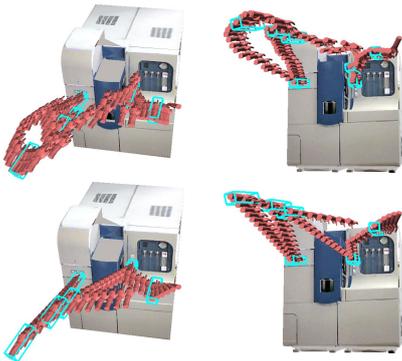


Figure 13: Motion of the pointing-hand in the context of a virtual instrument. The objective is to push sequentially two different buttons located on the surface of the instrument. The scene is shown from two different view-points. Original user’s inputs (top). Interpolated motion between change-points (bottom). The hands encapsulated in a frame correspond to the discovered change-points.

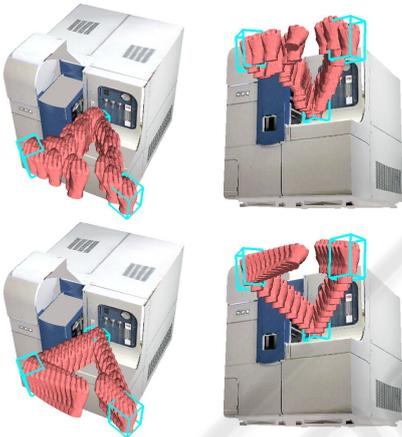


Figure 14: Motion of the picking-hand in the context of a virtual instrument. The objective is to express that the handle of the front cover must be grasped and pulled. The scene is shown from two different view-points. Original user’s inputs (top). Interpolated motion between change-points (bottom). The hands encapsulated in a frame correspond to the discovered change-points.

complex instruments with an efficient means of communicating knowledge, to end-users, about 3D tasks that must be performed for properly operating the instruments. Especially, it allows to animate existing 3D items (CAD models) in the context of a subject instrument (that can be a physically concrete instrument, or alternatively, a virtual representation). For animating an items, the user move the tablet PC in the real-world as if it was the item. The motion is then captured via the VisTracker. In other words, the engine allows to acquire freehand motion.

Note that the motion is not directly given by the sensor. As depicted in Figure 12, the motion is de-

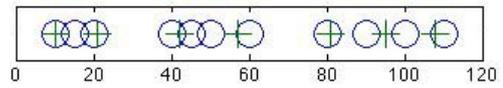


Figure 15: Detected change-points in the pointing-hand motion. The 7 crosses are the ones found using our method, while the 11 circles are the ones found using the SST method.

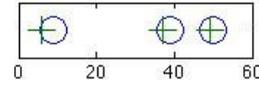


Figure 16: Detected change-points in the pointing-hand motion. The 3 crosses are the ones found using our method, while the 3 circles are the ones found using the SST method.

finied by the time sequence of poses R and T (i.e., the camera pose). These poses are also required for registering the virtual objects (the item being animating and the virtual instrument) in the current scene. The VisTracker measures the coordinates $(X, Y, Z)^T$ and the orientation (yaw, pitch and roll) of the sensor reference frame \mathcal{R}_S in the world reference frame \mathcal{R}_W . The relative position T_{c_s} and orientation R_{c_s} of the VisTracker (\mathcal{R}_S) and the video camera (\mathcal{R}_C) need to be known to compute the camera pose. To determine this transformation, we perform an initial calibration process that consists in computing n poses (R_i, T_i) from different viewpoints using a purely optical method (chessboard pattern recognition), simultaneously, recording the sensor data $(R_{s,i}, T_{s,i})$, and finally, finding the R_{c_s} and T_{c_s} that minimized the cost function $\sum_{i=1}^n (\|R_i - R_{c_s}R_{s,i}\| + \|T_i - (R_{c_s}T_{s,i} + T_{c_s})\|)$.

Figure 13 and 14 show the captured motion for two different scenarios. In the first one (Figure 13), the user is moving a pointing-hand model to sequentially push two different buttons. In the second one (Figure 14), the user is moving a picking-hand model for expressing a situation in which the handle of a cover has to be grasp and pulled. In both figures, the top row depicts the original user’s input, while the bottom row represents an enhanced version. We can observed that the discovered change-points are pertinent in the sense that, in both cases, the motion segmentation correspond to the user’s intent (the buttons are pushed, and the handle is grasped and pulled), and consequently, the motion is successfully and greatly enhanced (e.g., the unintended “jerky” movements are removed). The performed interpolation here consists of a naive “screw-motion” joining the change-points and ignoring the all other points input by the user. To better represent the user’s intention, a more elaborated method, such as the one discussed by (Hofer and Pottmann, 2004), should be considered.

For comparison, we have performed the SST

method to the 6 components of the motion signal output by the VisTracker. In the first scenario, we have discovered a total of 7 change-points using our method, whereas the SST method has yielded 11 change-points. Figure 15 shows the relative distributions of those two sets of points. One can observe that the two results are, to a fairly large extent, well-correlated. Although the SST method has discovered 4 points more than our method, the relative distributions (Figure 15) suggests that we could “cluster” together the 11 points in a way that correspond to the 7 points discovered by our approach. Especially, regarding that the 11 points are an interpolation through consolidation of the SST results independently obtained from the 6 signals received from the VisTracker. A change in one of the signal at time t , and a change in another signal at time $t + \epsilon$ (with ϵ small), might have the same cause. Even though, two change-points may be discovered. Our method searches for changes in a particular mixture of the 6 signals (that leads to a series in $SE(3)$), which may yield a single change at, e.g., $t + \epsilon/2$. This phenomenon is unlikely to occur in the synthetic data, for the cause leading to a change is, by construction, well synchronized between the signals. Moreover, the artificial noise follows a neat Gaussian distribution (in practice, the stochastic imperfection of the real data due to various causes is unlikely to follow a perfect Gaussian law).

Considering the second scenario, Figure 16 shows that the both methods give comparable results. Only a slight shift in the change-point positions can be observed between the two approaches.

6 CONCLUSIONS

We have proposed a method for detecting change-points in rigid-body motion time series. This method can be regarded as an adaptation of the difference of means method to time series in $SE(3)$. It is based on the key observation that the absolute gain of the mean filter yields a local maximum when a change occurs. By exploiting this result and the particular Lie group structure of $SE(3)$, we have shown that the change-points in $SE(3)$ can be discovered in its Lie algebra $se(3)$ through the following process: The initial time series in $SE(3)$ is transformed in a corresponding time series in $se(3)$ (via logarithmic mapping). Then for each point in the vector space $se(3)$, we calculate the norm of the difference between a weighted mean of the point to the left and a weighted mean of the point to the right. Finally, the potential change-points correspond to the maximum values.

A set of evaluations has been conducted showing that, assuming an adequate parameter setting (mainly the window size of the mean filter), the method should yield a low number of false positives and a very low, if any, number of false negatives.

REFERENCES

- Agarwal, M., Gupta, M., Mann, V., Sachindran, N., Anerousis, N., and Mummert, L. (2006). Problem determination in enterprise middleware systems using change point correlation of time series data. In *2006 IEEE/IFIP Network Operations and Management Symposium NOMS 2006*, pages 471–482. IEEE.
- Basseville, M. and Nikiforov, I. V. (1993). *Detection of Abrupt Changes - Theory and Application*. Prentice-Hall, Inc, Englewood Cliffs, N.J.
- Buss, S. R. and Fillmore, J. P. (2001). Spherical averages and applications to spherical splines and interpolation. *ACM Trans. Graph.*, 20(2):95–126.
- Courty, N. (2008). Bilateral human motion filtering. In *the 16th European Signal Processing Conference*, Lausanne, Switzerland.
- Drummond, T. and Cipolla, R. (2002). Real-time visual tracking of complex structures. *IEEE Transactions on PAMI*, 24:932–946.
- Fiorentino, M., Monno, G., Renzulli, P. A., and Uva, A. E. (2003). 3D sketch stroke segmentation and fitting in virtual reality. In *International conference on the Computer Graphics and Vision*, pages 188–191, Moscow, Russia.
- Fletcher, P. T., Lu, C., and Joshi, S. (2003). Statistics of shape via principal geodesic analysis on lie groups. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 95–101. IEEE Comput. Soc.
- Foxlin, E. and Naimark, L. (2003). Vis-tracker: A wearable vision-inertial self-tracker. *Virtual Reality Conference, IEEE*, 0:199.
- Gombay, E. (2008). Change detection in autoregressive time series. *Journal of Multivariate Analysis*, 99:451–464.
- Govindu, V. M. (2004). Lie-algebraic averaging for globally consistent motion estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 684–691. IEEE Comput. Soc.
- Grassia, F. S. (1998). Practical parameterization of rotations using the exponential map. *journal of graphics, gpu, and game tools*, 3(3):29–48.
- Hofer, M. and Pottmann, H. (2004). Energy-minimizing splines in manifolds. *ACM Transactions on Graphics*, 23(3):284–293.
- Huang, J.-S. (2000). *Introduction to Lie Groups*, chapter 7, pages 71–89. World Scientific Publishing Company.
- Ide, T. and Inoue, K. (2005). Knowledge discovery from heterogeneous dynamic systems using change-point correlations. In *SIAM International Conference on Data Mining*, pages 571–576.

- Ide, T. and Tsuda, K. (2007). Change-point detection using krylov subspace learning. In *SIAM International Conference on Data Mining*, pages 515–520.
- Kawahara, Y. and Sugiyama, M. (2009). Change-point detection in time-series data by direct density-ratio estimation. In *SIAM International Conference on Data Mining*.
- Lee, J. and Shin, Y. S. (2002). General construction of time-domain filters for orientation data. *IEEE Transactions on Visualization and Computer Graphics*, 8:119–128.
- Merckel, L. and Nishida, T. (2009). Towards expressing situated knowledge for complex instruments by 3D items creation and animation. In *The 8th International Workshop on Social Intelligence Design*, pages 301–315, Kyoto, Japan.
- Moskvina, V. and Zhigljavsky, A. (2003). An algorithm based on singular spectrum analysis for change-point detection. *Communications in Statistics: Simulation and Computation*, 32:319–352.
- Qin, S.-F., Wright, D. K., and Jordanov, I. N. (2001). On-line segmentation of freehand sketches by knowledge-based nonlinear thresholding operations. *Pattern Recognition*, 34:1885–1893.
- Selig, J. M. (2005). *Lie Algebra*, chapter 4, pages 51–83. Monographs in Computer Science. Springer New York, New York.
- Sezgin, T. M. and Davis, R. (2004). Scale-space based feature point detection for digital ink. In *Making pen-based interaction intelligent and natural, AAAI fall symposium*.
- Srivastava, A. and Klassen, E. (2002). Monte Carlo extrinsic estimators of manifold-valued parameters. *IEEE Transactions on Signal Processing*, 50(2):299–308.
- Stuelpnagel, J. (1964). On the parametrization of the three-dimensional rotation group. *SIAM Review*, 6:422–430.
- Subbarao, R. and Meer, P. (2006). Nonlinear mean shift for clustering over analytic manifolds. In *Computer Vision and Pattern Recognition*, pages 1168–1175. IEEE.
- Tuzel, O., Subbarao, R., and Meer, P. (2005). Simultaneous multiple 3d motion estimation via mode finding on lie groups. In *Tenth IEEE International Conference on Computer Vision*, pages 18–25. IEEE.
- van Dam, A. (1997). Post-wimp user interfaces. *Communications of the ACM*, 40(2):63–67.
- Zefran, M., Kumar, V., and Croke, C. (1998). On the generation of smooth three-dimensional rigid body motions. *IEEE Transactions on Robotics and Automation*, 14:576–589.