Pattern Recognition 43 (2010) 204–221

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Motion trajectory reproduction from generalized signature description

Shandong Wu, Y.F. Li*

Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Kowloon, Hong Kong

ARTICLE INFO

Article history:
Received 22 October 2008
Received in revised form 17 March 2009
Accepted 24 May 2009

Keywords:
Motion trajectory
Signature descriptor
Mutual description
Trajectory reproduction
Computer vision

ABSTRACT

Free form motion trajectories prove to be an informative and compact motion clue in sketching long-term, spatiotemporal motions. Hence, motion trajectories have been used for characterizing human behaviors/activities, robot actions and other objects' movements. However, it is observed that most of the previous studies merely use motion trajectories straightforwardly in the raw data form, which is inflexible as they rely largely on the absolute positions. To solve this problem, we propose to achieve effective motion trajectory descriptions by developing a systematic trajectory description mechanism. To this end, a flexible motion trajectory signature descriptor has been proposed in our previous work, which can offer generalized descriptions to the raw trajectory data thanks to its rich description invariants. Moreover, for an effective descriptor, it is sometimes desired to have mutual description functions, i.e. describing and un-describing capability to support some applications like robot learning. Hence, opposite to describing a motion trajectory using the signature, this paper focuses on the un-describing problem, that is, reproducing a trajectory instance from a given signature description. The moving frame technique is used in formulating the trajectory reproduction method. A nonlinear signature matching-based metric is also developed to measure the quality of the reproductions. Experiments are conducted to verify the effectiveness of the trajectory reproduction. It is shown that the trajectory signature is flexible and easy to implement in both the description and reproduction of trajectory instances.

1. Introduction

1.1. Motion trajectories: definition and role

Dynamic features seem more suitable than static features in characterizing the motions of humans, robots or other moving objects. For example, free form motion trajectories prove to be an effective motion feature for modeling long-term, spatiotemporal motions [1,2]. A motion trajectory consists of a set of position vectors of continuously sampled points for a moving object in a spatiotemporal motion. Hence, motion trajectories are informative, compact, robust and meaningful in outlining the motions. In fact, not limited to the individual trajectory tracked from a single moving part, multiple concurrent motion trajectories can be extracted too for depicting complicated motions. For example, although the full-body motions are usually complicated due to the articulated structure, descriptive motion trajectories are still traceable and useful by identifying the tracked parts in concern, such as the hands, head, feet [3,4]. In this manner, a spatiotemporal motion can be well characterized by visually observing and extracting the consequent motion trajectories of a motion. Note that some short-term motions, such as instant grasps, might not be suitably characterized by motion trajectories. Such a kind of motions will not be covered in this paper.

1.2. Motion trajectories: usage and description

Previous studies show how motion trajectories are used to characterize various kinds of spatiotemporal motions. Motion trajectories have been studied for human behavior analysis, including representing human actions [1], recognizing human gestures [5], modeling human walking [6], and even classifying human facial expressions [7]. Motion trajectories can also be used for learning human motion skills [8] and for detecting abnormal motions [9]. Moreover, multiple motion trajectories extracted from several human body parts were used for gait classification [10] and activity recognition [11]. Meanwhile, motion trajectories were also investigated for robotic applications. Motion paths were usually represented by motion trajectories in robot motion planning [12,13]. Motion trajectories were also used for depicting and analyzing motion-based robot tasks for supporting robot learning [2,14,15]. In addition, the motion trajectories of ordinary moving objects in certain workflows also play an important role for movement recognition and analysis [16].

Although motion trajectory is proved to be a suitable feature for motion characterization in previous works, a motion trajectory

* Corresponding author.
E-mail addresses: sdwu@eecs.ucf.edu (S.D. Wu), meyfli@cityu.edu.hk (Y.F. Li).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.05.019

was usually used directly in its raw data form. The raw data might be sufficient for certain simple work but such a simplistic use of raw data is quite inflexible because it relies heavily on the absolute positions of the motion data. In particular, it should be noted that the raw data also restrain the overall performance of some high-level modeling approaches such as neural networks [6] and hidden Markov models [2,10], when they are directly used based on raw motion data. In addition, it is noticed that motion trajectories were used in some other form of representations in certain cases. Yet, it is only simple and plain descriptors that were used without considering their effectiveness.

The concept of shape descriptor [17,18] motivates the development of motion trajectory descriptors. Some shape descriptors have been built in the past [19]. Simple descriptors are nonimpressive in performance [20,21]. Occlusion is a difficult problem for global descriptors such as geometric moments [22]. Mathematical descriptors such as Bezier curve [23], B-spline [24] and NURBS [25] suffer from the fitting inaccuracy. Some transform functions such as wavelet transform [26], Fourier descriptor (FD) [27], curvature scale space (CSS) [28] and Radon transform [29] can represent shape in a coarse-to-fine manner. Yet, only partially salient features are considered for shape description.

We focus on the research of building systematic and effective motion trajectory-oriented description mechanisms. The previous work includes a flexible 3-D signature descriptor [30,31]. We use 3-D motion trajectories instead of 2-D, as 3-D data can reconstruct and model space motions more accurately than the 2-D data by providing the depth information, which also overcomes the image viewpoint problem that the 2-D data suffer from [32]. More importantly, the signature is a generalized description to the raw trajectory data. The generality lies in the rich descriptive invariants that enable the signature to capture the essential motion. Besides, the signature promises flexible description forms meeting adaptable description requirements [31].

1.3. Mutual description functions: describing and un-describing

Sometimes it is desired for a descriptor to have mutual transformation abilities in describing and un-describing motions. Compared with the forward describing action by data generalization, the un-describing process is about the backward reproduction of the data instances. While both the forward description and backward reproduction are important in fulfilling the missions of a descriptor, it should be noted that the practical description requirements depend on the applications. For example, when recognition is the main goal, the backward un-describing function will not be that necessary, which makes a descriptor efficient in handling complex motions. When both recognition and reproduction are of concern such as in robot task understanding, human–robot interactions and motion reconstruction/synthesis, a descriptor's mutual description abilities will accordingly be desired.

Opposite to describing a motion trajectory, this paper deals mainly with the signature's un-describing function, i.e. reproducing motion trajectories from a given signature. Firstly, we introduce the basic signature definition. Then, we focus on the formulation of the trajectory reproduction algorithm. Invariant trajectory reproduction and reproduction measurement are also elaborated. The most important contribution of this paper is the trajectory reproduction method that completes the signature's mutual description abilities, which can systematically enhance the signature-based motion description mechanism.

The existing descriptors have different performance in their un-describing abilities. Normally efforts need to be paid to find the corresponding un-describing methods for simple descriptors. For the mathematical curves such as B-spline, NURBS, the reproduction is difficult as the descriptions are inaccurate due to the data fitting. For the FD, the inverse Fourier transform is actually the un-describing process. Likewise, the inverse wavelet transform is the un-describing operation of the wavelet transform. The Radon transform also has a corresponding inverse transform. But the reproduction process of the CSS suffers from the curve shrinking problem as it is not easy to precisely reconstruct the original curve length.

1.4. Mutual signature description: applications

The mutual signature descriptions can play a key role in supporting some typical and important applications.

1.4.1. Robot learning

It is desired that a robot can learn essential task knowledge instead of multiple redundant task instances [33,34]. Hence, building generalized and reusable task representations is important, particularly in robot learning by demonstration (LbD) [35,36]. However, the motion trajectories have not been well described to support generalized robot task learning. In addition to using the raw trajectory data [14,37], it is observed that previous trajectory representation schemes include B-spline wavelet [15], NURBS [25] and linear dynamical system [38], which rely inherently on the absolute positions of robot tasks and suffer from the fitting inaccuracy.

It can be found that the generalized signature can serve as a systematic and effective robot task descriptor, which can support robot learning of the essential task knowledge. Meanwhile, the task reproduction, i.e. instantiating a trajectory instance from the signature-based task descriptions, is also a necessary step to complete the procedures of robot learning as it enables robots to re-run the learned tasks. Moreover, efficient robot learning also desires multiple trajectory variants to be reproduced from an individual task signature, which can be easily achieved by making use of the signature's rich invariants. In general, the signature description and trajectory reproduction can be coupled to serve as the core components of a basic motion trajectory-oriented robot learning system.

1.4.2. Motion recognition and reconstruction

Recognizing and reconstructing spatiotemporal motions are important for analyzing behaviors, activities and movements [39,40]. The motion trajectory signature can enrich the description-based schemes for motion analysis [41]. In fact, using the generalized signature and taking advantage of its rich invariants, we can achieve efficient motion recognition overcoming the inflexibility and redundancy of the raw motion data.

Meanwhile, reproducing motions from the compact signature descriptions is also useful for motion trajectory-based behavior reconstruction [42], motion scenes synthesis and editing [43], which is meaningful in supporting virtual (or augmented) reality and in making animation and cartoon plays [44,45]. In practice, for a complicated motion, each of the sub-motions represented by the corresponding motion trajectory can be individually transformed and freely assembled. This has some similarity to human dancing synthesis [46] that uses motion primitives and styles. From the generalized signatures, the motion reconstruction needs trajectories to be reproduced a priori and diverse trajectory variants are expected to be easily produced from one signature. Here, the signature's backward reproduction function and its rich invariants can serve an important role. Hence, the signature can be mutually capable in supporting effective motion analysis, i.e. recognition, reconstruction, synthesis and editing.

This paper is organized as follows. Section 2 briefs the signature description principles. The trajectory reproduction method is presented in Section 3, followed by the analysis of invariant trajectory reproduction in Section 4. A reproduction metric is designed in

Section 5. Experiments proceed in Section 6 and this work is finally concluded in Section 7.

2. Motion trajectory signature description

We assume that 3-D motion trajectories have been acquired from motions. Studying the methods for trajectory tracking is out of the focus of this paper and in fact, there are many methods well developed for trajectory tracking [47].

2.1. Generalized trajectory signature

In our previous work [30], the following signature has been proposed to describe raw motion trajectories.

Definition 1. For a regular 3-D motion trajectory parameterized by $\Gamma(t)=\{X(t),Y(t),Z(t)|t\in[1,N]\}$, where t is the temporal stamp and N is trajectory's length, its signature S is defined in terms of Euclidean differential invariants: curvature (κ), torsion (τ) and their derivatives with respect to the Euclidean arc-length parameter s,

$$S=\{[\kappa(t),\kappa_s(t),\tau(t),\tau_s(t)]|t\in[1,N]\} \quad (1)$$

This signature allows a complete description for all sampled trajectory points. Hence, it does not suffer from the data fitting inaccuracy. For the 3-D motion trajectory Ω shown in Fig. 1, its signature description (the robust approximate signature implementation [30]) is shown in Fig. 2. Here, the trajectory signature defined in Eq. (1) is the basis of our systematic signature mechanism. More flexible signature descriptions were studied in Refs. [31,48]. In this paper, we will mainly use the trajectory signature to formulate the signature's un-describing problem.

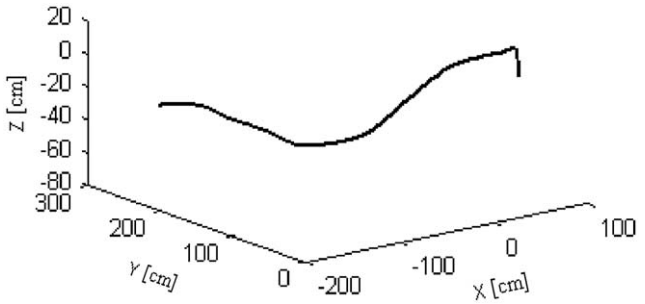

Fig. 1. A piece of 3-D motion trajectory Ω.

2.2. Signature's description completeness

The signature here is defined for regular trajectories. However, we have also provided a solution to enable the signature to describe irregular trajectories [31]. More generally, the signature proves to be able to describe complicated motions. Complicated motions usually give rise to multiple, concurrent, interrelated motion trajectories. While parallel trajectories entail the simultaneous motions of multiple parts, it is the sequential trajectory that captures temporally continuous motions. On the one hand, parallel trajectories can be described one by one and then combined, manually, as a whole. On the other hand, there are two options for describing a temporally continuous trajectory. While a single signature is able to describe an arbitrary length of regular trajectory, this might have difficulty in more detailed descriptions of a semantic rich motion. Hence, a motion trajectory that contains multiple distinctive stages could be segmented into multiple meaningful episodes, from which more detailed motion analysis can be conducted according to the segmented episodes and their relationships. We leave more detailed segmentation study to the future work.

2.3. Signature description for motion classes

In addition to the trajectory signature for describing a single instance, the description to a motion class/pattern is also important. Fig. 3 shows four different motion classes, each of them is characterized by three similar instances. A common way of description is to manually maintain or combine the trajectory signatures of the multiple instances. However, this kind of description is too redundant.

Model-based motion class description was investigated in our study. A so-called cluster signature is built using Gaussian mixture model (GMM) to serve as an abstract and efficient motion class description [31]. Note that our modeling approach can outperform the traditional work as the GMM is coupled with the generalized trajectory signature instead of raw data. It is also indicated that the signature can be jointly used with more generic model-based methods. This explicitly shows how the introduction of the generalized

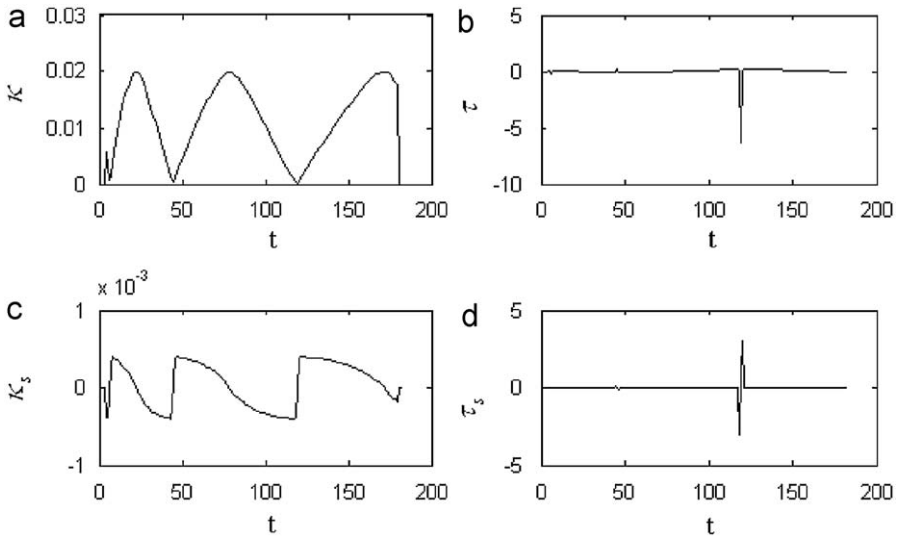

Fig. 2. The approximate signature of trajectory Ω: (a) κ; (b) τ; (c) κ_s and (d) τ_s.

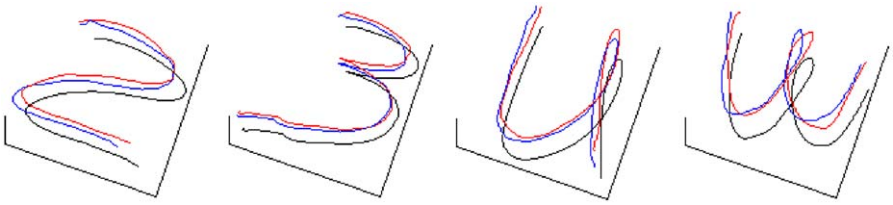

Fig. 3. Four different motion classes.

signature description to high-level modeling methods can overcome the inflexibility caused by direct coupling of raw data.

2.4. Signature's invariants

As the signature is based on local features, it admits computational locality, from which substantial descriptive invariants can be deduced [30]. The signature is invariant with respect to Euclidean transformations and relatively invariant with respect to scaling actions. These two invariants subsequently lead to the signature's insensitivity to spatial 3-D viewpoint changes. Here, a 3-D viewpoint is represented by a viewing angle and viewing distance. In addition, partial invariant can be achieved with respect to speed profile change or occlusion. Note that the abovementioned invariants can be further transmitted to subsequent processing. For instance, the GMM-based motion class description inherits all of these invariants.

The substantial invariants make the signature a generalized description, which can benefit more effective motion recognition since the signature is insensitive to motion positions, sizes, viewpoints, speed profiles and occlusions. Likewise, the reproduction can also enjoy this advantage (see Section 4).

2.5. Signature description ability extension

Although the signature is defined for 3-D data, it can be easily simplified to a 2-D signature for describing 2-D or 1-D data [31]. In particular, 1-D data will be parameterized in a 2-D form by incorporating the time stamp information.

In addition, in this paper, the signature is used to describe position information related motion trajectories. In essence, it is actually able to describe diverse natures of data such as motion orientation or velocity information, provided that the described data are 3-D spatiotemporally continuous.

3. Trajectory reproduction methods

Via the signature description, a motion trajectory is transformed into the signature space. In this section, we will formulate the 'inverse' problem: given a signature description, to instantiate a trajectory instance from the signature profiles. The trajectory reproduction is exactly about the un-describing function of the signature descriptor. As stated in Section 1.4, studying this problem is meaningful as it is not only demanded for robot learning, but also is useful in supporting motion reconstruction, synthesis and editing.

3.1. Reproduction from a trajectory signature

In this subsection, we report a novel method to reproduce a trajectory from a given trajectory signature. Assume a signature $S=\{[\kappa(t),\kappa_S(t),\tau(t),\tau_S(t)]|t\in[1,N]\}$ is given a priori or calculated from a trajectory. Here, the objective is to formulate the algorithm to produce a trajectory instance $\Gamma(t)=\{X(t),Y(t),Z(t)|t\in[1,N]\}$ that has the signature of S, with a predefined initial starting point $\Gamma(1)$ and an initial motion direction $\Theta(1)$. To solve this problem, we use the moving frame technique and the Frenet–Serret formula [49].

Frenet frame is a special moving frame that is associated with a space 3-D curve. For a 3-D curve $\Gamma(s)$ parameterized by the arc-length parameter s, the Frenet frame $F(s)$ is defined by three Frenet vectors T, N and B,

$$F(s)=\{T(s),N(s),B(s)\} \tag{2}$$

where

$$\begin{cases} T(s)=d\Gamma(s)/ds \\ N(s)=\dfrac{dT(s)/ds}{\|dT(s)/ds\|} \\ B(s)=T(s)\times N(s) \end{cases} \tag{3}$$

Furthermore, there is a Frenet–Serret formula that describes the derivatives of the three Frenet vectors,

$$\begin{cases} dT(s)/ds=\kappa(s)\cdot N(s) \\ dN(s)/ds=-\kappa(s)\cdot T(s)+\tau(s)\cdot B(s) \\ dB(s)/ds=-\tau(s)\cdot N(s) \end{cases} \tag{4}$$

From the above equation, we can find that the derivatives of a Frenet frame is expressed in terms of the frame itself and the curvature $\kappa(s)$ and torsion $\tau(s)$. This relation provides a possibility to derive the next Frenet frame from a given Frenet frame.

Note that the Frenet frame $F(s)$ assumes that trajectory $\Gamma(s)$ is parameterized by the arc-length parameter s. This means that the tangent vector of $\Gamma(s)$ is a unit vector, i.e. $\|d\Gamma(s)/ds\|=1$, which implies that the arc-length between two consecutive trajectory points is equal to 1. In other words, the underlying motion speed $v(s)=1$. Thus, Eq. (4) is actually based on a trajectory with a constant speed profile, i.e., for all s, $v(s)=1$. However, the form of trajectory $\Gamma(t)$ is parameterized by the temporal parameter t, and $\Gamma(t)$ could possibly have an arbitrary speed profile. This explains why Eq. (4) cannot be applied directly. In fact, for an arbitrary trajectory defined by $\Gamma(t)$, an equivalent expression of the Frenet–Serret formula picks up an additional factor of motion speed $v(t)$, in the following form:

$$\begin{cases} dT(t)/dt=v(t)\cdot\kappa(t)\cdot N(t) \\ dN(t)/dt=v(t)\cdot(-\kappa(t)\cdot T(t)+\tau(t)\cdot B(t)) \\ dB(t)/dt=v(t)\cdot(-\tau(t)\cdot N(t)) \end{cases} \tag{5}$$

Eq. (5) is more generic than Eq. (4) as it can deal with more flexible trajectory parameterization by $\Gamma(t)$ and an arbitrary speed profile $v(t)$.

The three Frenet vectors are perpendicular to each other, as illustrated in Fig. 4(a), where T is the tangent vector, N is the principal normal vector and B is the binormal vector. Fig. 4(b) illustrates the Frenet frames along a motion trajectory. Note that for clarity, the plotting of the frames is spaced at every 18 points. It is found that the Frenet frame can inherently characterize the shape of a trajectory point.

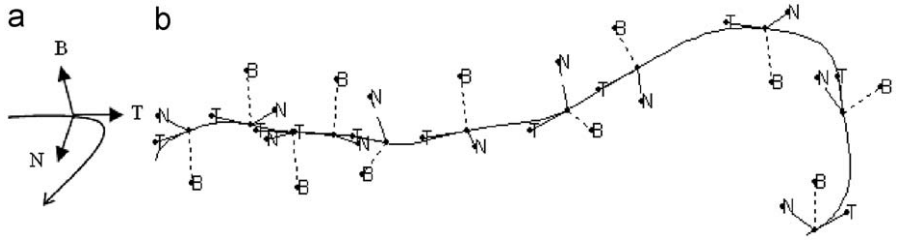

Fig. 4. The illustration of Frenet frame: (a) Frenet frame definition and (b) the Frenet vectors plotted along a motion trajectory.

So far, it can be inferred that if we could determine the Frenet vectors of all the sampled trajectory points and the arc-lengths between two consecutive points, we will be able to reproduce a corresponding motion trajectory accordingly. The key to this problem lies in finding out the varying relation of the Frenet frame between two consecutive points. The Frenet–Serret formula defined in Eq. (5) can exactly account for this. That is to say, given a starting point and an initial motion direction, a motion trajectory can be reproduced by iteratively reproducing all the trajectory points according to Eq. (5) in the temporal sequence.

Now let us focus on Eq. (5). Firstly, the forward finite difference is used to calculate the derivatives on its left side,

$$\begin{cases} dT(t)/dt = T(t) - T(t-1) \\ dN(t)/dt = N(t) - N(t-1) \\ dB(t)/dt = B(t) - B(t-1) \end{cases} \tag{6}$$

Then, denote the Frenet frame at two consecutive trajectory points t_1 and t_2 by $F(t_1) = \{T(t_1), N(t_1), B(t_1)\}$ and $F(t_2) = \{T(t_2), N(t_2), B(t_2)\}$, respectively, we can derive the following equation to calculate $F(t_2)$ from $F(t_1)$:

$$\begin{cases} T(t_2) = T(t_1) + v(t_1)\cdot\kappa(t_1)\cdot N(t_1) \\ N(t_2) = N(t_1) - v(t_1)\cdot\kappa(t_1)\cdot T(t_1) + v(t_1)\cdot\tau(t_1)\cdot B(t_1) \\ B(t_2) = B(t_1) - v(t_1)\cdot\tau(t_1)\cdot N(t_1) \end{cases} \tag{7}$$

The values of curvature $\kappa(t)$ and torsion $\tau(t)$ can be obtained directly from the given signature data. Now we derive the calculation of the motion speed $v(t)$. For a trajectory parameterized by temporal parameter t, its motion speed profile is the first order derivative of the trajectory equation, i.e. $v(t) = \|\dot{\Gamma}(t)\|$. In fact, $v(t)$ can be derived from the signature components $\kappa_s(t)$ and $\tau_s(t)$. Based on the given signature data, we can use the central finite difference to calculate $d\kappa(t)/dt$ and $d\tau(t)/dt$. For example, we can use $\kappa_s(t)$ to derive the following speed formula to be used in Eq. (7):

$$v(t) = \frac{d\kappa(t)/dt}{\kappa_s(t)} = \frac{\kappa(t+1) - \kappa(t-1)}{2\kappa_s(t)} \tag{8}$$

As the Frenet–Serret formula in Eq. (5) supposes that $\{T(t), N(t), B(t)\}$ are unit Frenet vectors, the Frenet frame $\{T(t_2), N(t_2), B(t_2)\}$ obtained by Eq. (7) needs to be further normalized to unit vectors by the following formulae:

$$\begin{cases} \hat{T}(t_2) = T(t_2)/\|T(t_2)\| \\ \hat{N}(t_2) = N(t_2)/\|N(t_2)\| \\ \hat{B}(t_2) = B(t_2)/\|B(t_2)\| \end{cases} \tag{9}$$

Therefore, a successive Frenet frame $F(t_{i+1})$ can be calculated by Eqs. (7)–(9) from a previous Frenet frame $F(t_i)$. In this manner, starting from an initial Frenet frame $\{\hat{T}(1), \hat{N}(1), \hat{B}(1)\}$ with a given signature S, iterating Eqs. (7)–(9), we can obtain an entire sequence of Frenet frame $F_\Gamma(t)$ for a trajectory $\Gamma(t)$,

$$F_\Gamma(t) = \{\{\hat{T}(t), \hat{N}(t), \hat{B}(t)\} | t \in [1, N]\} \tag{10}$$

According to the definition of Frenet frame, $F_\Gamma(t)$ describes the motion direction varying between consecutive trajectory points. Therefore, each Frenet frame in $F_\Gamma(t)$ can actually play the role of controlling the motion direction at each trajectory point in the course of trajectory reproduction. In other words, the overall shape feature of a trajectory is determined by the obtained Frenet frame sequence $F_\Gamma(t)$. In particular, the three vectors of the initial Frenet frame $\{\hat{T}(1), \hat{N}(1), \hat{B}(1)\}$ represent the initial motion direction $\Theta(1)$. For the determination of $\Theta(1)$, we can firstly predefine an arbitrary tangent vector $\hat{T}(1)$, and then $\hat{N}(1)$ and $\hat{B}(1)$ can be calculated according to their definitions.

Once the motion directions of all the trajectory points are determined, we can subsequently produce the position vectors of these trajectory points. Recall the definition of the tangent vector in Eq. (3), $T(s) = d\Gamma(s)/ds$. By applying the finite difference method, we can derive the following relation:

$$\Gamma(s_2) = \Gamma(s_1) + T(s_2) \tag{11}$$

where s_1 and s_2 are two consecutive points in a trajectory parameterized by the arc-length parameter s. Eq. (11) is for the constant motion speed profile ($v(s) = 1$). Based on this, for a motion trajectory which has an arbitrary speed profile $v(t)$, we further formulate the following equation to reproduce trajectory $\Gamma(t)$ that is parameterized by t,

$$\Gamma(t_2) = \Gamma(t_1) + \hat{T}(t_2)\cdot\frac{v(t_2)}{\|T(t_2)\|} \tag{12}$$

Here, the roles of the three factors in Eq. (12) are specifically explained. As mentioned before, while the unit vector $\hat{T}(t_2)$ controls the motion direction from point t_1 to t_2, it is $v(t_2)$ that controls the actual arc-length traversing between t_1 and t_2. Referring to the signature's relative invariant with respect to scaling actions, the reason for including the factor of $\|T(t_2)\|$ in Eq. (12) lies in the normalization operation in Eq. (9).

Now, given an initial motion point $\Gamma(1) = \{X(1), Y(1), Z(1)\}$, Eq. (12) can be iterated so that a sequence of trajectory points will be reproduced for reproducing trajectory $\Gamma(t)$ that has the signature of S. Here, $\Gamma(1)$ could be an arbitrary 3-D point. In a nutshell, the above trajectory reproduction algorithm is elaborated in terms of two steps, that is, firstly to determine the motion directions of each trajectory point, and then to determine the position vectors of the points.

Here, a length related problem in the trajectory reproduction is particularly pointed out. The initial three and final three points in a trajectory cannot be reproduced since the approximate signature calculation involves multiple neighbor points instead of a single point. In addition, the last fourth point cannot be reproduced, either, due to the finite difference calculation in the reproduction algorithm. Therefore, if it is expected to reproduce a trajectory with the original length, the initial three and last four points should be saved when the signature is calculated.

3.2. Reproduction from a motion class description

The previous subsection deals with the trajectory reproduction from a trajectory signature. It is also necessary to study how to produce trajectories from a given motion class description, i.e. the GMM-based cluster signature. This problem can be tackled by the following two steps.

First, the Gaussian mixture regression (GMR) method is used to instantiate a trajectory signature instance from a GMM-based description [31]. GMR aims at estimating the conditional expectation of the response given a predictor. Here, the predictor is defined as the temporal stamps of a motion and the response will be the profiles of a trajectory signature. Second, after obtaining the GMR-generated trajectory signature, it can then be used to follow the reproduction algorithm in Section 3.1 to produce a trajectory.

3.3. Reproduction of multiple trajectories

The above methods deal directly with the reproduction of an individual trajectory. A complicated motion with multiple trajectories can also be reproduced. As each trajectory is described by a corresponding signature, we can reproduce all of the constitutive trajectories of a motion from the corresponding signatures, respectively. Then the reproduced trajectories can be manually put together to reproduce the entire motion. This means that the current method can also be used for reproducing such kind of complicated motion of multiple trajectories.

4. Invariant trajectory reproduction

Based on the signature's invariants, the reproduction algorithm can be further extended with high adaptability in reproducing trajectory instances. According to Section 1.4, the ability of producing diverse trajectory variants from an individual signature description is specifically advantageous for robot learning and motion synthesis. For the former, a user no longer needs to demonstrate all of the possible instances of the same task in LbD. That is, the trajectory instances of a translated, or rotated, or size-scaled, or length-shortened (occluded), or speed-profile-varied variant can be easily reproduced from the same one signature. This can reduce a lot of laborious work for the task demonstrator. For the latter, various trajectory instances can be easily reproduced to provide elements for flexible motion synthesis. For example, by making use of the signature's Euclidean invariant, a translated or a rotated trajectory instance can be produced for the intended positioning of this instance towards synthesizing a specific motion. Likewise, the different sizes of, length-shortened and speed-profile-changed trajectory variants can also be reproduced according to the corresponding signature invariants. In the following subsections, invariant trajectory reproduction problems are discussed.

4.1. Euclidean invariant reproduction

Consider a subsequent problem to the trajectory reproduction results in Section 3.1. Assume that the signature S is calculated from trajectory $\bar{\Gamma}(t)$, the question is: what is the relation between the instantiated trajectory $\Gamma(t)$ and the original trajectory $\bar{\Gamma}(t)$? This problem involves the signature's Euclidean invariant. In fact, $\Gamma(t)$ and $\bar{\Gamma}(t)$ have the same signature, but they are not always identical in space position. If the initial starting point $\Gamma(1)$ and the initial motion direction $\Theta(1)$ are chosen as those of trajectory $\bar{\Gamma}(t)$, then in principle the produced trajectory $\Gamma(t)$ must be completely identical with $\bar{\Gamma}(t)$.

In practice, we can choose $\Gamma(1)$ and $\Theta(1)$ arbitrarily and trajectory $\Gamma(t)$ will still have the same signature as $\bar{\Gamma}(t)$. While the different choice of $\Gamma(1)$ corresponds to a translation transformation, the different choice of $\Theta(1)$ admits a rotation transformation. That is, $\Gamma(1)$ and $\Theta(1)$ correspond to a Euclidean transformation. Therefore, the signature's Euclidean invariant can ensure that $\Gamma(t)$ has the same signature as $\bar{\Gamma}(t)$. In case $\Gamma(1)$ and $\Theta(1)$ are chosen that are different from those of $\bar{\Gamma}(t)$, the relation between $\Gamma(t)$ and $\bar{\Gamma}(t)$ is called a congruence, which means that $\Gamma(t)$ is a variant of $\bar{\Gamma}(t)$ after a Euclidean transformation.

In general, the formulated trajectory reproduction algorithm in Section 3.1 can produce arbitrary congruent variants of a motion trajectory. It is the signature's Euclidean invariant that enables the trajectory reproduction independent of the absolute positions. Hence, we will be able to choose arbitrary initial parameters for $\Gamma(1)$ and $\Theta(1)$ to produce a trajectory instance which always has the same signature as S.

A specific problem is how to determine the initial parameters of $\Gamma(1)$ and $\Theta(1)$. As $\Gamma(1)$ is a 3-D point, it can be directly determined by a position vector. $\Theta(1)$ defines the initial motion direction in terms of the Frenet vectors $T(1)$, $N(1)$ and $B(1)$. Hence, $\Theta(1)$ can be determined according to the definition of the Frenet frame. For example, we can simply set $T(1)=[1\ 0\ 0]$, $N(1)=[0\ 1\ 0]$ and then $B(1)=[0\ 0\ 1]$. But if we intend to define $\Theta(1)$ exactly as $\bar{\Theta}(1)$ of the original trajectory $\bar{\Gamma}(t)$, we need to calculate $\bar{\Theta}(1)$ a priori from the initial points of trajectory $\bar{\Gamma}(t)$. This can be done using Eq. (3). Because the initial three points are lost in the approximate signature calculation, the actual $\Theta(1)$ corresponds to the motion direction of the fourth point ($[T(4), N(4), B(4)]$) of trajectory $\bar{\Gamma}(t)$. Moreover, since different motion direction indicates a 3-D rotation action, we can apply a rotation action (characterized by a 3-D rotation matrix) to $\bar{\Theta}(1)$ to obtain a parameter for $\Theta(1)$.

According to the above analysis, the Euclidean invariant enables a trajectory instance to be reproduced easily transparent to the absolute positions. In addition, as $\Gamma(1)$ and $\Theta(1)$ can characterize the viewing angle of visual motions, the Euclidean invariant can simultaneously make the motion trajectory reproduction independent of viewing angle changes (assuming a fixed viewing distance).

4.2. Scaling relatively invariant reproduction

According to the signature's scaling relative invariant defined in Ref. [30], if we intend to reproduce a trajectory which is a scaling transformation of the original trajectory, we can just scale the given signature data to be used then in the trajectory reproduction algorithm. Hence, via controlling the scaling factor, various sizes of trajectory instances can be readily instantiated from a given individual signature. These instances may correspond to the trajectory variants with respect to the changes of viewing distances between the motion demonstrator and visual sensors (assuming a fixed viewing angle). Accordingly, the Euclidean invariant plus the scaling relative invariant can nicely account for the reproduction of the motions with different viewpoints (viewing angles plus viewing distances).

4.3. Length adjustable reproduction (occlusion handling)

The length of a trajectory can be represented in terms of points' number or arc-length. We can show that the reproduction length is adjustable for the both representations. This subsection deals with the former and Section 4.4 for the latter. As the trajectory signature is locally calculated, the number of a trajectory's points is controllable in the reproduction algorithm. This means that we can produce a partial trajectory from a complete signature. In particular, when occlusions occur, only a partial trajectory can be obtained. But the signature of the partial trajectory remains unaffected compared with the corresponding part of a complete signature. Hence, the length

adjustable trajectory reproduction can naturally handle the reproduction of occluded trajectories.

Assume that the original trajectory length is t_o, apply a predefined length $t_o^{new} < t_o$ in the reproduction algorithm, then only a trajectory portion will be produced. In addition, as the signature is insensitive to trajectory's direction, we can obtain direction invariant trajectory reproduction. That is, the same motion trajectory can be produced starting from either end of a trajectory signature.

4.4. Speed profile controllable reproduction

Speed is a key motion feature. The signature's speed invariant at a point enables the reproduction algorithm to reproduce an instance with an unchanged speed profile. Moreover, motion speed is actually a flexibly controllable factor in the reproduction process. As the arc-length between two consecutive points essentially indicates the traversing speed, it is feasible to control the speed profile in reproducing a trajectory by adjusting the arc-length reproduction. Here, we discuss this problem from two points.

First, in view of the role the motion speed played in the reproduction algorithm in Eq. (12), we can define a speed control function $\Delta(t)$ with respect to the temporal stamp t, and then apply it to the original speed profile $v(t)$, to achieve the objective of speed controllability in reproducing a trajectory. The new speed profile $v^{new}(t)$ can be calculated by,

$$v^{new}(t) = \Delta(t) \cdot v(t) \tag{13}$$

Then $v^{new}(t)$ will be used in Eq. (12), which changes the trajectory arc-lengths via the speed change. Here, it should be noted that the speed change expressed by Eq. (13) is applied to Eq. (12) instead of Eq. (7). This is because the role of speed in Eq. (7) is for calculating Frenet vectors to determine the motion direction of a reproduction. In essence, it is exactly in Eq. (12) where the speed quantity is controlling the arc-length between two consecutive points. In other words, if the new speed profile in Eq. (13) is applied to Eq. (8), both motion direction and arc-length of the reproduced trajectory will be affected. This is beyond the aim of controlling merely the speed feature.

In particular, in case the speed control function $\Delta(t)$ is a constant, e.g. $\Delta(t) = \delta$ (δ is a real number), the new speed profile will be scaled as $v^{new}(t) = \delta \cdot v(t)$. Now to review the effect on the trajectory reproduction caused by $v^{new}(t)$, comparison is made with a scaling action with the same parameter δ. According to the scaling relative invariant [30] and Eq. (8), here, we can derive the speed profile $\hat{v}^{new}(t)$ as follows for the trajectory to which the scaling action δ is applied,

$$\hat{v}^{new}(t) = \frac{\hat{\kappa}(t+1) - \hat{\kappa}(t-1)}{2\hat{\kappa}_{\hat{s}}(t)} = \frac{\frac{1}{\delta}\kappa(t+1) - \frac{1}{\delta}\kappa(t-1)}{2\frac{1}{\delta^2}\kappa_s(t)} = \delta \cdot v(t) \tag{14}$$

We find that $v^{new}(t) = \hat{v}^{new}(t)$. This implies that a constant quantity scaled new speed profile actually has the same effect on the trajectory reproduction as a corresponding scaling action (both with the same parameter δ). Note that the derivation of Eq. (14) is targeting Eq. (8) and the used signature data is actually scaled according to the scaling relative invariant. While the curvature κ and torsion τ are scaled by $1/\delta$, the speed profile is simultaneously scaled by δ. Hence, these two scaling effects can be eliminated in Eq. (7). This ensures that only trajectory arc-length is scaled while trajectory shape (motion direction) is preserved.

By Eq. (13), although the speed profile can be controlled, the whole trajectory arc-length is also affected, simultaneously. This can be easily understood from the perspective of scaling transformation as it has the same effect as a constant function $\Delta(t)$. Referring to Section 4.3, this is actually the other kind of reproduction with controllable trajectory length. Here, the length is about trajectory arc-length.

In addition, we also present a second kind of speed control approach that keeps the whole trajectory arc-length unchanged. The principle of this approach is to apply a follow-up linear interpolation to a normally reproduced trajectory. Via customizing the interpolation step, the speed profile can be controlled while the whole arc-length can also remain unchanged.

5. Metric for measuring reproductions

Following the trajectory instantiation results, we need a metric to measure the quality of the reproductions. Errors are unavoidable in the reproduction algorithm as the discrete derivative is approximated by finite difference. We plan to generate the new signature of a reproduction and then compare it with the reference signature to measure the degree of reproduction error.

In case a reproduced trajectory has the same length as the original trajectory (ignoring the lost points at two ends), two signatures can be linearly matched in a pointwise manner. Yet, as analyzed in Section 4, a reproduction may have an arbitrary length (in terms of points' numbers or arc-length) and distribution of trajectory points. In particular, when the reproduced trajectories are input to robots to replay, noise and motion inconsistency normally exist. This means that the new signature of a re-tracked trajectory might differ from the reference signature in quantity, length or points' distribution. In such a case, direct linear signature comparison is no longer applicable. The reproduction metric should be able to account for nonlinear matching of two arbitrary signatures.

We use and extend a nonlinear inter-signature matching algorithm to serve as the reproduction metric, in which the dynamic time warping (DTW) [31] method is employed that is able to find a reasonable matching between two arbitrary trajectory signatures by nonlinear path warping.

For two signatures $S^{*1} = \{[\kappa^{*1}, \kappa_s^{*1}, \tau^{*1}, \tau_s^{*1}]_p | p \in [1, P]\}$ and $S^{*2} = \{[\kappa^{*2}, \kappa_s^{*2}, \tau^{*2}, \tau_s^{*2}]_q | q \in [1, Q]\}$ with respective length P and Q, applying the DTW algorithm in Ref. [31], the best alignment of the two signatures can be found giving rise to the DTW distance at $u(P, Q)$, which can serve as a measurement of the reproduction error. But note that the DTW distance relies on the signature length P and Q. Hence, here we further formulate an average DTW distance $\bar{u}$ for the use of measuring the average reproduction error,

$$\bar{u} = 2u(P, Q)/(P + Q) \tag{15}$$

The distance ($u(P, Q)$ and/or $\bar{u}$) can provide quantitative measurement of the quality of the reproductions. Furthermore, according to the DTW-based inter-signature matching result, we can obtain the corresponding inter-trajectory alignment referring to the temporal stamps of the signature data. The alignment is depicted by the warping paths, which can be visualized to give an intuitive perception to the difference and consistency of two motions. This means that the DTW method can also offer a qualitative measurement of the reproductions.

Normally, the following two steps can be carried out to evaluate a trajectory reproduction:

- *Step one*. Examine the (average) DTW distance to see how close a reproduction is to the reference motion trajectory. The reproduction would be acceptable if this distance falls below a predefined error threshold.
- *Step two*. A human (or a robot) can observe and perceive certain motion features of a reproduction (or the re-tracked counterpart of a reproduction) according to the inter-trajectory warping paths. For example, in robot task re-running, the speed profile can be

checked via the DTW visualization to see if it is consistent with that of the reference. In the event that a certain trajectory portion is not replayed as expected, the visualization can supervise the robot to pay more attention when repeating the trajectory. In this way, a robot can self-improve the performance of task reproductions in multiple trials.

6. Experiments

The objective of the experiments is to illustrate the effects of the signature and the reproduction algorithm. Experiments are conducted for a simple LbD where the robot's task is to learn some signed language. The sign language is a special interaction modality between humans. The purpose of the LbD learning is to let a robot learn the sign language so that the robot can interact with humans (or other robots) using the signs. The sign language is performed by continuous movements of human hands. A binocular vision system is used to observe human demonstrations and the underlying 3-D sign motion trajectories are extracted to characterize the demonstrations. Based on this, the signature description and the reproduction algorithm are tested. It should be noted that at this stage, the sign reproductions have not yet been input to a real robot to follow. However, based on these results, it should be easy to do that in the future, according to the configurations of a specific robot system.

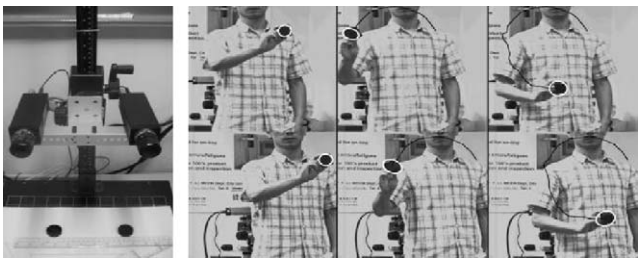

Fig. 5. The vision system setup and snaps of the stereo trajectory tracking.

6.1. Sign trajectory observation and acquisition

In this experiment, we only focus on the regular sign trajectories resulting from the movements of a single hand. The binocular vision system has two TM-765 cameras and the Continuously Adaptive Mean Shift (CAMShift) algorithm is employed to track spatiotemporally discrete 3-D trajectories [31]. Fig. 5 shows the vision system setup and several stereo trajectory tracking snaps of a sign word being demonstrated.

To reduce the noise associated with the extracted trajectories, trajectory smoothing was conducted and proved to be an effective way to enhance the signature's computational reliability. In addition, if a trajectory contains some sharp turns that might cause computational instability, the smoothing can also appropriately smooth these turns to reduce the unexpected influence. In our study two smoothers [30] are designed, one is an adaptive moving average filter and the other is a wavelet smoother. Normally the former is more useful than the latter in dealing with irregular and uneven noise. But, the latter is easier to operate with fewer parameters. In practice, they can be used individually or jointly. As sometimes it is difficult or impossible to estimate the noise strength or types a priori, users can try the two filters respectively, or combined them, to compare and then choose to achieve a relatively satisfactory smoothing effect. Hence, these two smoothers can provide options to account for various situations of noise. Note that although the trajectory shape may also be affected by the smoothing, both the two smoothers are able to balance the noise reduction and shape preservation by interactively setting tunable smoothing parameters. Fig. 6 illustrates some smoothed noisy trajectories using the two filters. It is observed that the smoothing effect is acceptable, with only minor shape deviation.

6.2. Trajectory reproduction examples

In the reproduction experiments, we demonstrate both basic trajectory reproduction and the invariant reproduction. Fig. 7 shows a 3-D trajectory Π that characterizes a kind of human signed word. Its signature can serve as a general description to this word. Fig. 8 illustrates the approximate signature of trajectory Π. In the following

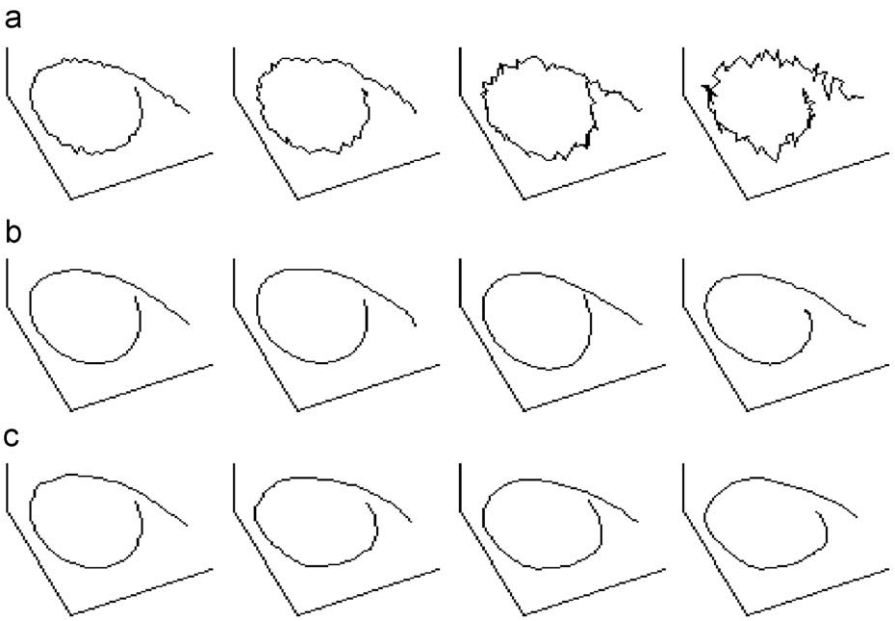

Fig. 6. Noise reduction to smooth (a) noisy trajectories using; (b) the moving average filter (span = 3, 7, 11 and 15) and (c) the wavelet smoother (wavelet DB4 and the coefficients at level 2–5).

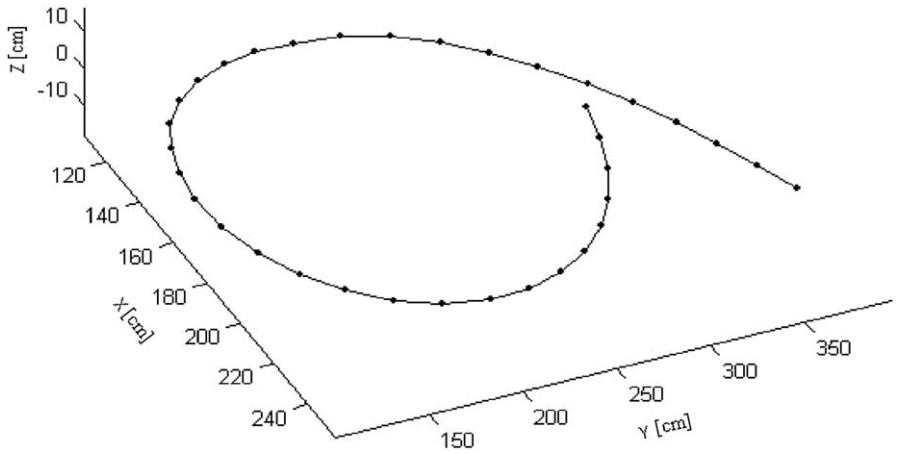

Fig. 7. Motion trajectory Π that characterizes a kind of human signed word.

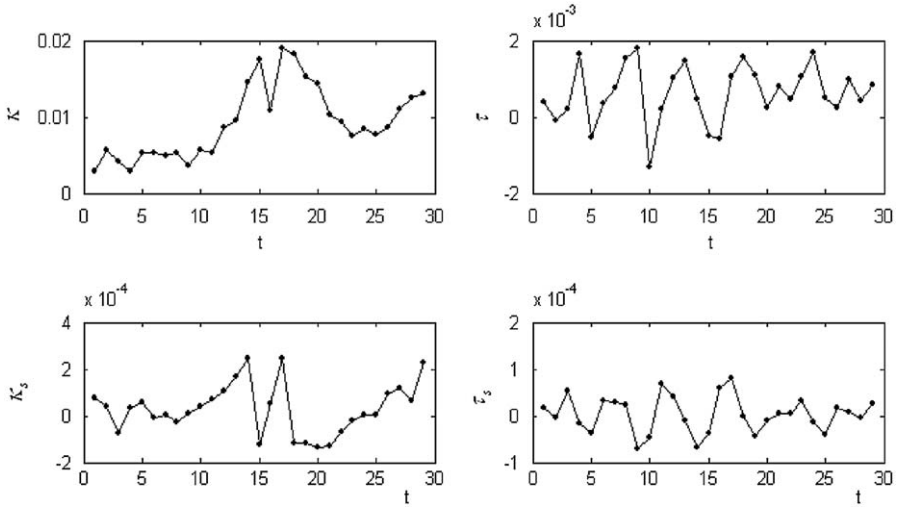

Fig. 8. The signature profiles (κ, τ, κ_s and τ_s) of trajectory Π.

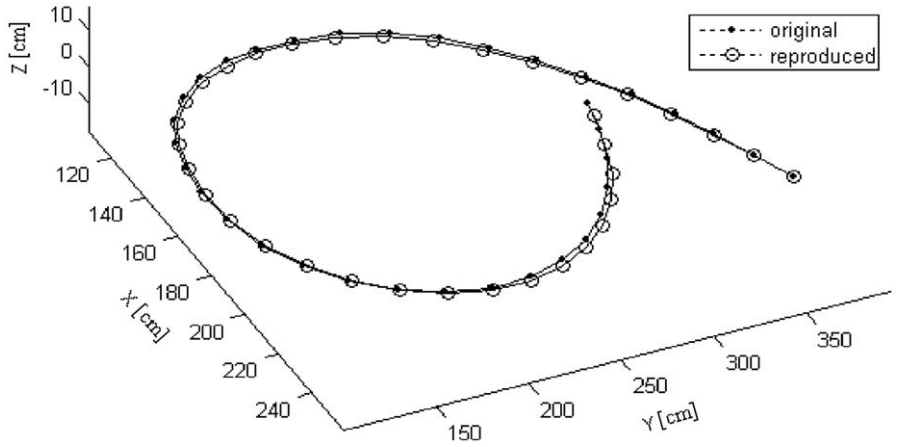

Fig. 9. A reproduction example of trajectory Π.

we will mainly use this example to test the trajectory reproduction algorithm.

First, we predefine the initial motion direction and starting point as those of trajectory Π, then a trajectory is instantiated as shown in Fig. 9 where it is compared with the original trajectory. It is observed that the reproduction is quite close to the original. Calculating the signature of the reproduced trajectory and comparing it with the original signature, we can find that they are also close, as illustrated in Fig. 10. Both Figs. 9 and 10 can verify that the developed trajectory reproduction algorithm works well.

In fact, we tested basic trajectory reproduction using more than 50 different motions. More examples of the basic trajectory reproduction are given in Figs. 11–13, where three reproductions of length of 27, 55 and 99 trajectory points are shown. The initial motion direction and starting point of each of the three reproductions are chosen as those of the original trajectory. It is observed that the reproduction effects are good too.

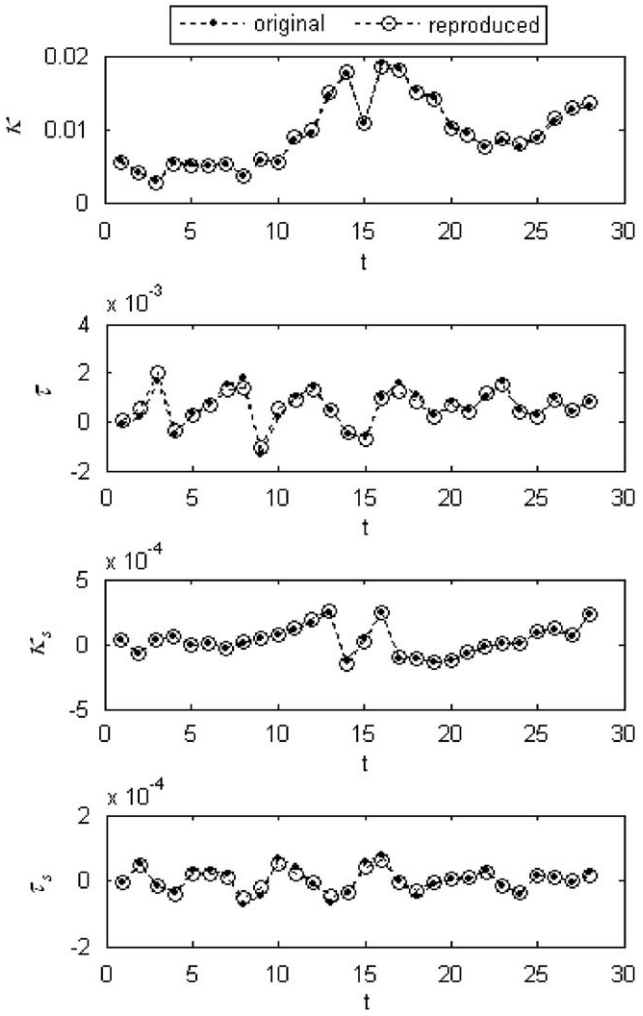

Fig. 10. Signature comparison between the reproduced trajectory and the original trajectory.

6.3. Invariant trajectory reproduction examples

In addition to basic trajectory reproduction, we also test the reproduction algorithm by more reproduction examples to verify the invariant reproduction. Various parameters are adopted for the reproduction referring to the analysis in Section 4.

With the initial motion point at [10 20 30] and the same initial motion direction as Π, Fig. 14 shows a reproduced trajectory example that illustrates the Euclidean (translation) invariant reproduction. Likewise, Fig. 15 also demonstrates the Euclidean invariant reproduction with respect to a rotation transformation, in which the reproduced trajectory has the same initial motion point as Π, and its initial motion direction is defined by a Frenet frame of $T(1)=[1\,0\,0]$, $N(1)=[0\,1\,0]$ and $B(1)=[0\,0\,1]$. Fig. 16 is one more rotation invariant reproduction example with the initial motion direction at $T(1)=[-0.119\ \ 0.588\ \ 0.8]$, $N(1)=[-0.98\ \ -0.199\ \ 0]$ and $B(1)=[0.159\ \ -0.784\ \ 0.6]$. In Fig. 17, a reproduction example illustrates both the translation and rotation invariant reproduction. Its initial motion point is at [10 20 30] and the initial motion direction is at $T(1)=[1\,0\,0]$, $N(1)=[0\,1\,0]$ and $B(1)=[0\,0\,1]$. Here, note that the observations of the figures are affected by the display viewpoints.

The following two examples (Figs. 18 and 19) show the scaling relatively invariant trajectory reproduction. The initial motion point of both the reproduced trajectories is at $[-50\ 30\ -160]$, and the initial motion direction is the same as Π. But the scaling factor is 0.58 (Fig. 18) and 1.7 (Fig. 19), respectively, in reproducing the two trajectories.

In Fig. 20, a speed controllable reproduction example is illustrated. While the parameters of the initial motion direction and starting point are the same as the example in Fig. 16, the speed control function here is defined by a constant function $\Delta(t)=3.2$. We can find that the reproduced trajectory has the same effect as the scaling transformation. This verifies the discussion in Section 4.4.

As observed, the trajectory arc-length of the example in Fig. 20 is stretched simultaneously along with controlling the speed. In the following two figures Figs. 21 and 22, we illustrate the arc-length preserved speed control in two reproduction instances. Applying the linear interpolation with a respective step of 0.5 (Fig. 21) and 2 (Fig. 22), the speed profiles of the reproduced trajectories,

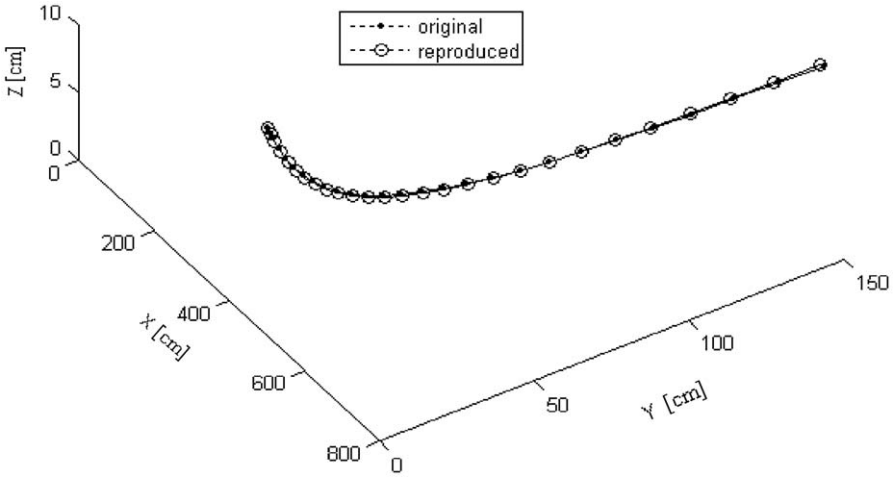

Fig. 11. Trajectory reproduction example 1 (length: 27 points).

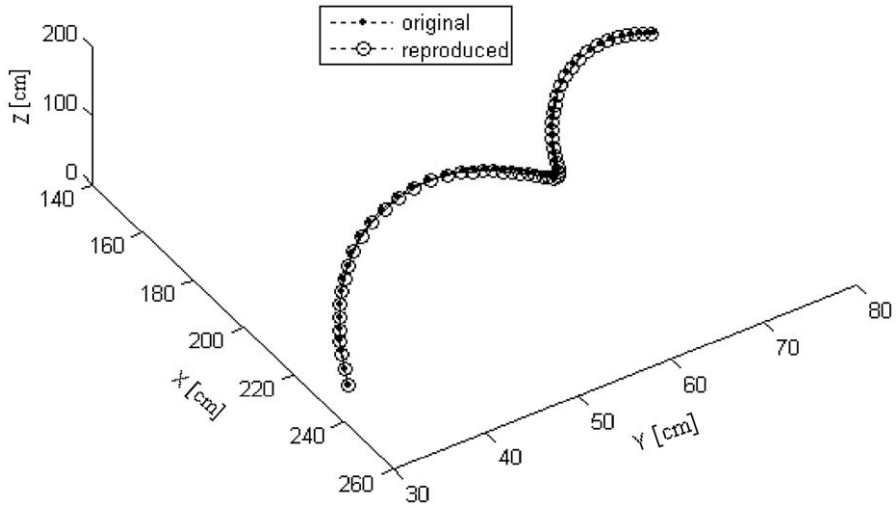

Fig. 12. Trajectory reproduction example 2 (length: 55 points).

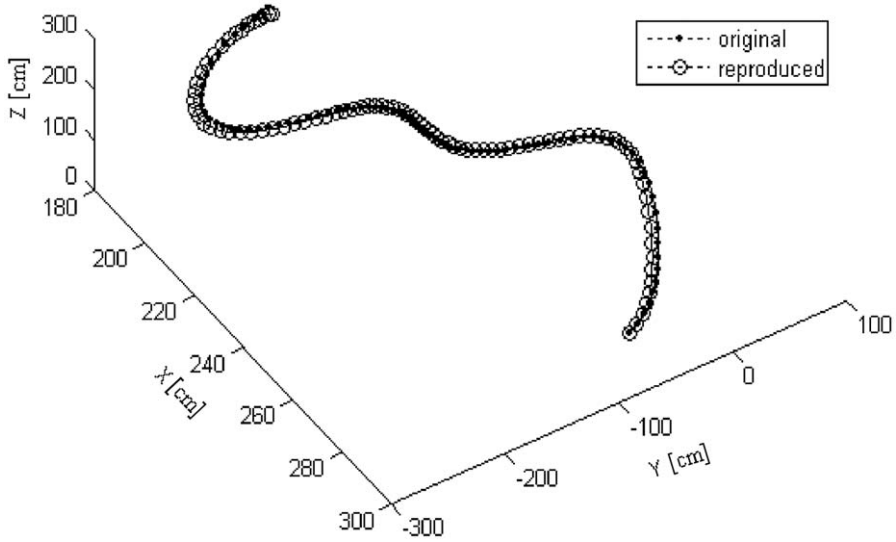

Fig. 13. Trajectory reproduction example 3 (length: 99 points).

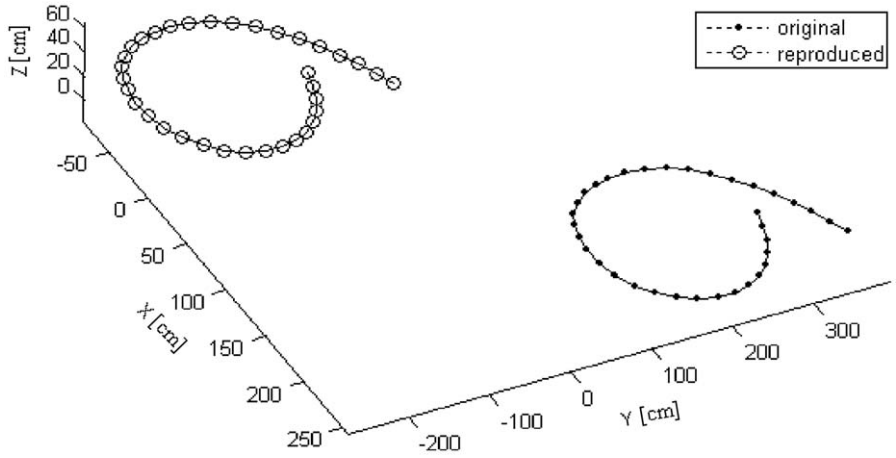

Fig. 14. Translation invariant trajectory reproduction.

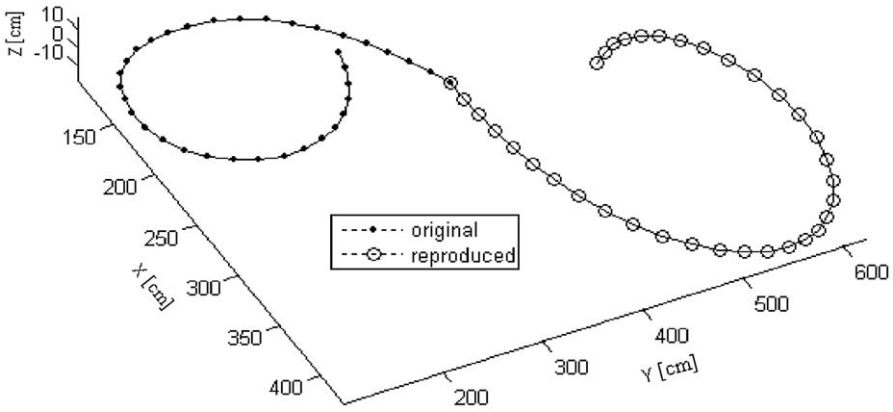

Fig. 15. Rotation invariant trajectory reproduction.

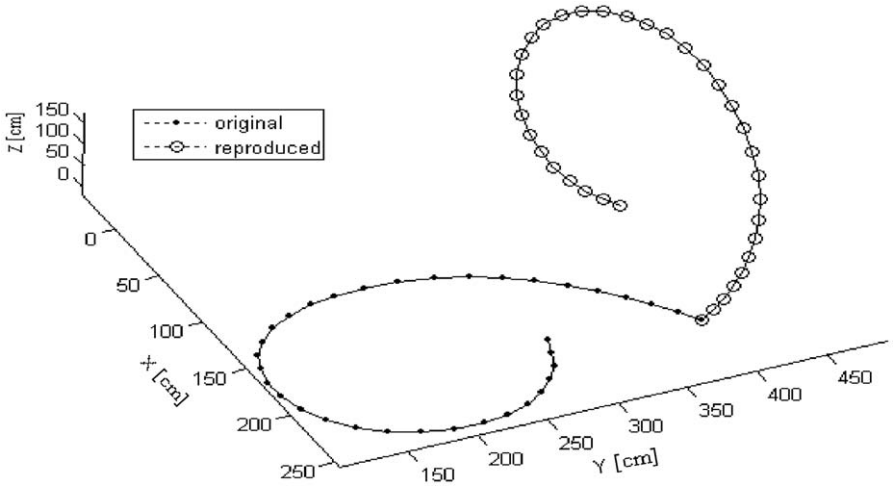

Fig. 16. Another example of rotation invariant trajectory reproduction.

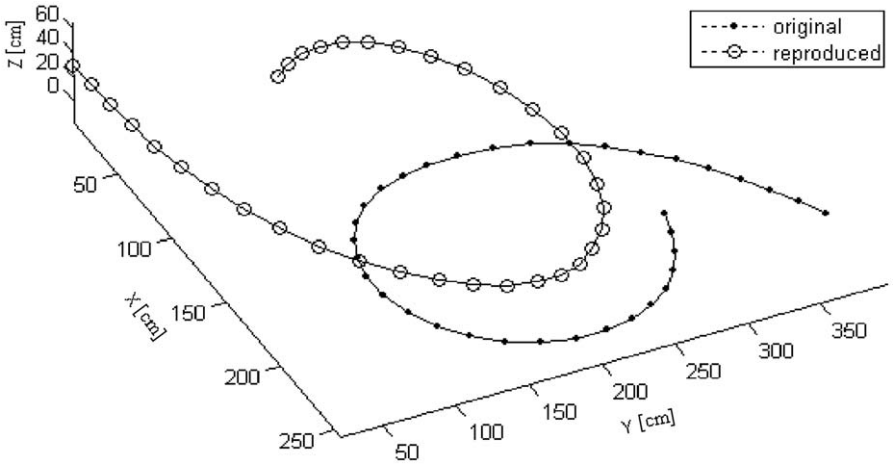

Fig. 17. Euclidean (both translation and rotation) invariant trajectory reproduction.

as observed, are halved and magnified accordingly. We can find that the arc-lengths of the two reproduced trajectories remain unchanged. As the interpolation step can be set arbitrarily, this kind of speed control shows the effectiveness of the trajectory reproduction in making a robot re-run a task with a flexibly configured speed profile.

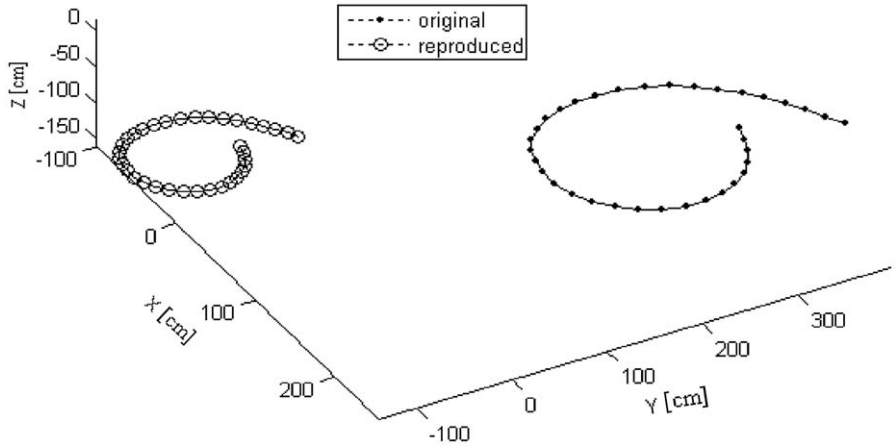

Fig. 18. Scaling relatively invariant trajectory reproduction with scaling factor 0.58.

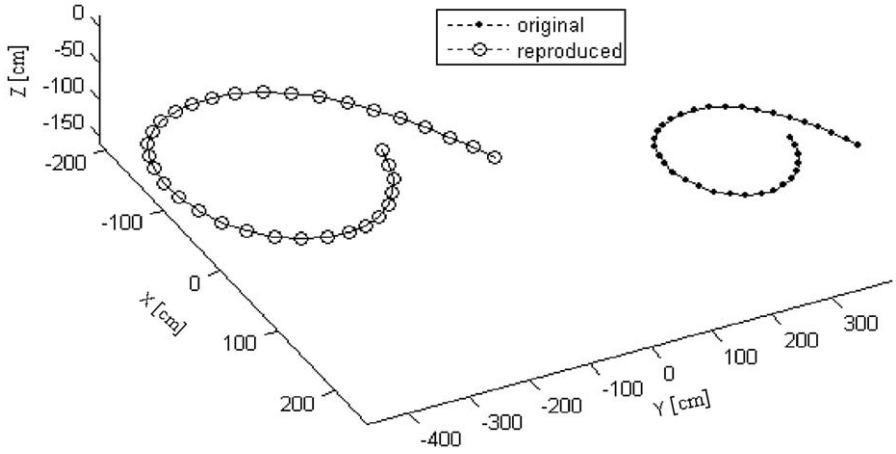

Fig. 19. Scaling relatively invariant trajectory reproduction with scaling factor 1.7.

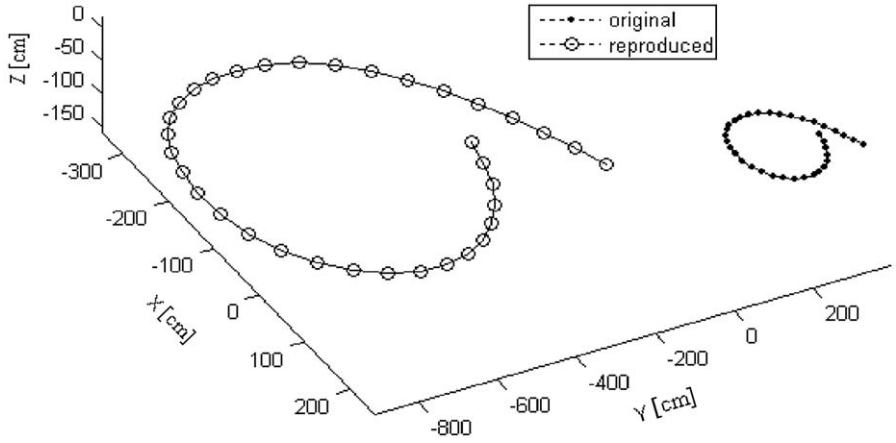

Fig. 20. Speed controllable trajectory reproduction with speed control function $\Delta(t) = 3.2$.

Figs. 23 and 24 demonstrate the length adjustable trajectory reproduction (or the reproduction under occlusion). While the last twelve trajectory points are not reproduced in the example in Fig. 23, the first nine points are skipped in reproducing the example given in Fig. 24. Thanks to the computational locality, the reproduction is independent of the skipped points. This also verifies that the signature can well account for occlusion in trajectory description and reproduction.

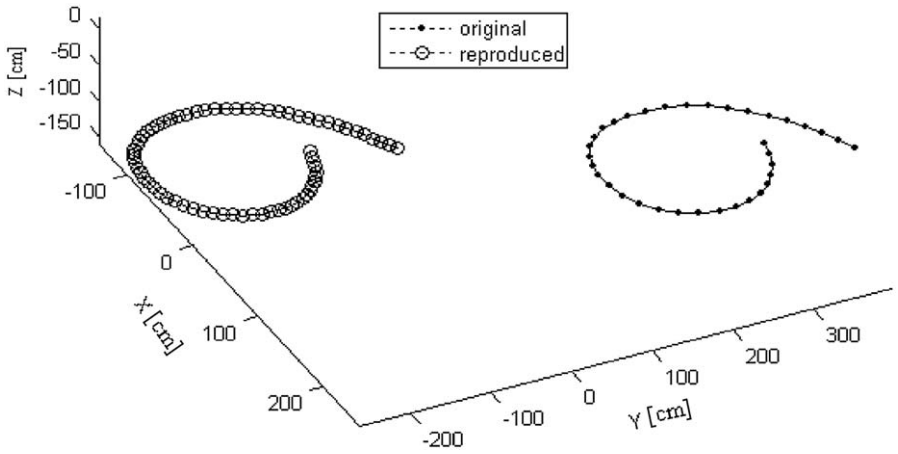

Fig. 21. Speed controllable trajectory reproduction (arc-length preserved) with interpolation step 0.5.

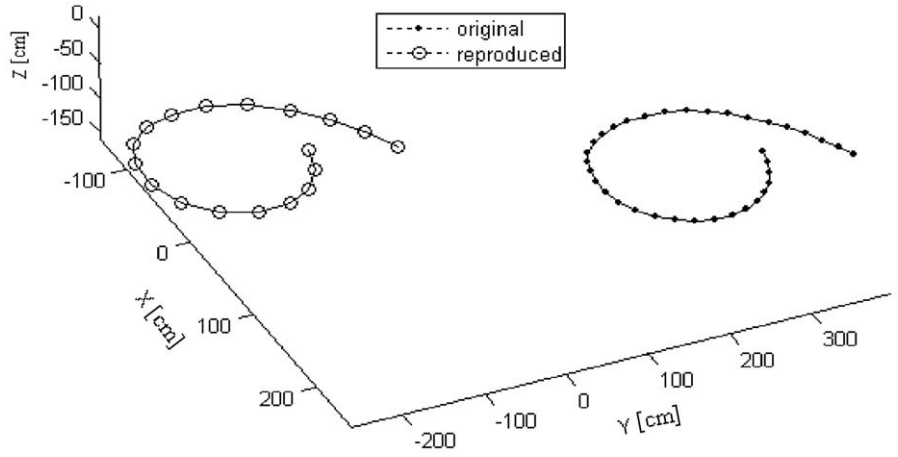

Fig. 22. Speed controllable trajectory reproduction (arc-length preserved) with interpolation step 2.

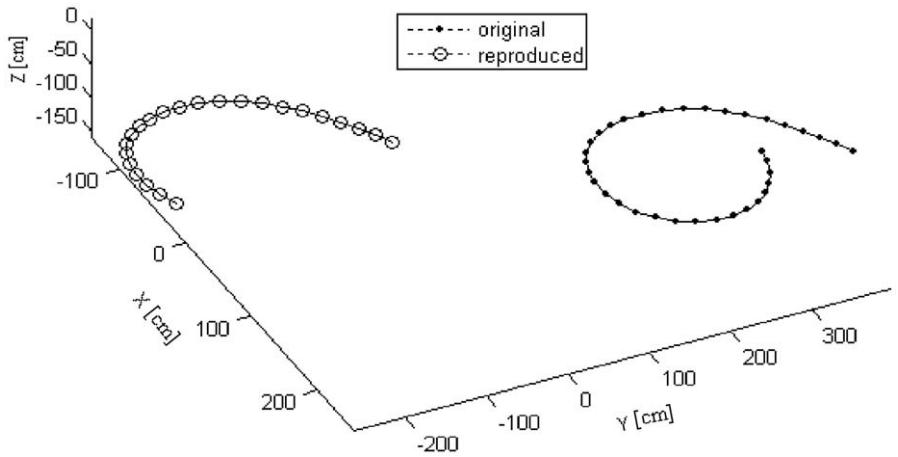

Fig. 23. Length controllable trajectory reproduction (the last twelve points are skipped).

6.4. DTW-based reproduction measurement

In this subsection, we demonstrate the quality measurement of the reproductions using the DTW-based nonlinear signature matching. Here, we select some representatives from the above examples to show and observe their DTW matching results in terms of both the qualitative and quantitative measurements.

First, the inter-trajectory warping paths are given to illustrate the qualitative perception and measurement. In the following four figures, Fig. 25 shows the DTW matching of the example in Fig. 17.

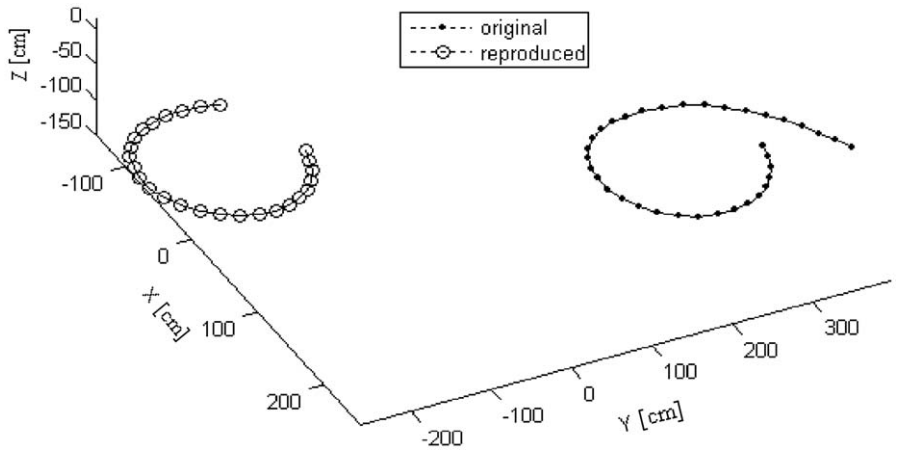

Fig. 24. Length controllable trajectory reproduction (the first nine points are skipped).

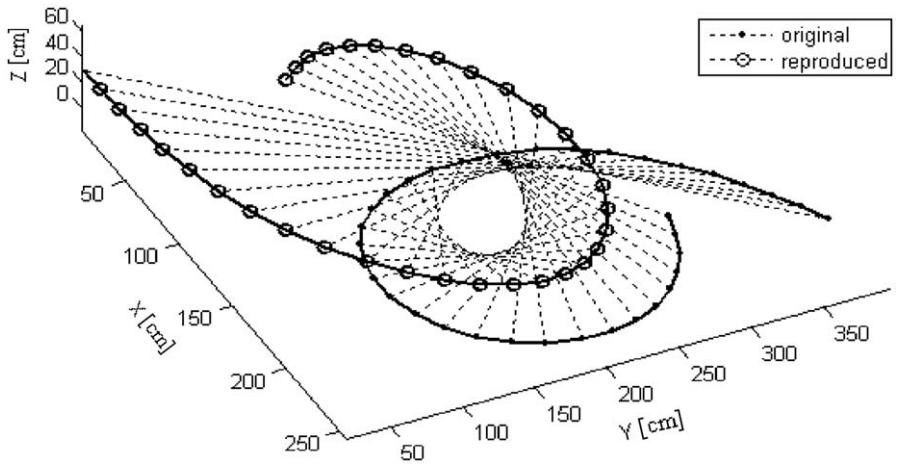

Fig. 25. Inter-trajectory warping paths based on the DTW-based signature matching (for the example in Fig. 17).

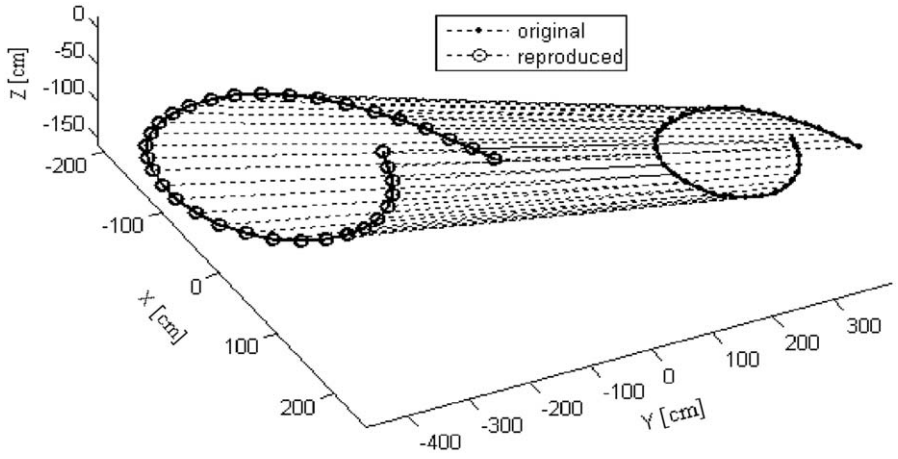

Fig. 26. Inter-trajectory warping paths based on the DTW-based signature matching (for the example in Fig. 19).

Similarly, Figs. 26, 27 and 28 correspond to the examples in Figs. 19, 22 and 24, respectively. We observe that the matching results are quite reasonable. In particular, note that the matching in Figs. 27 and 28 actually admits the one-to-many nonlinear alignment. Perceiving these matching examples can intuitively validate the developed trajectory reproduction algorithm.

In addition, the signature's DTW distances are also recorded as the quantitative measurement of the reproductions. The average DTW

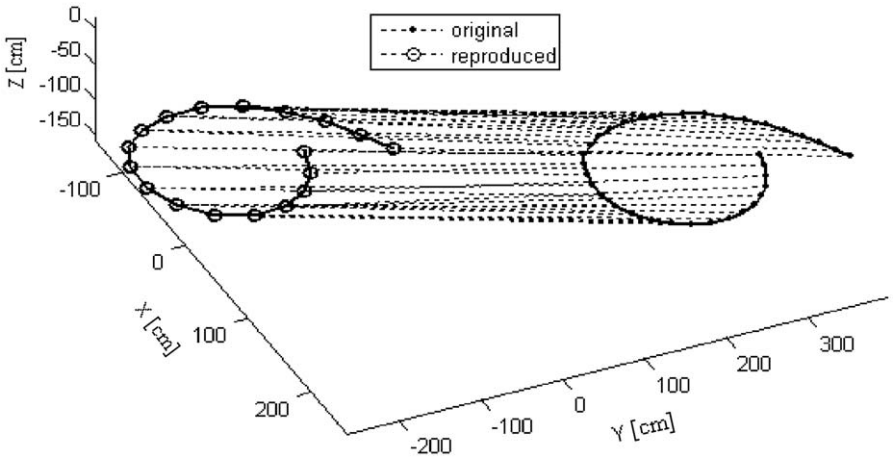

Fig. 27. Inter-trajectory warping paths based on the DTW-based signature matching (for the example in Fig. 22).

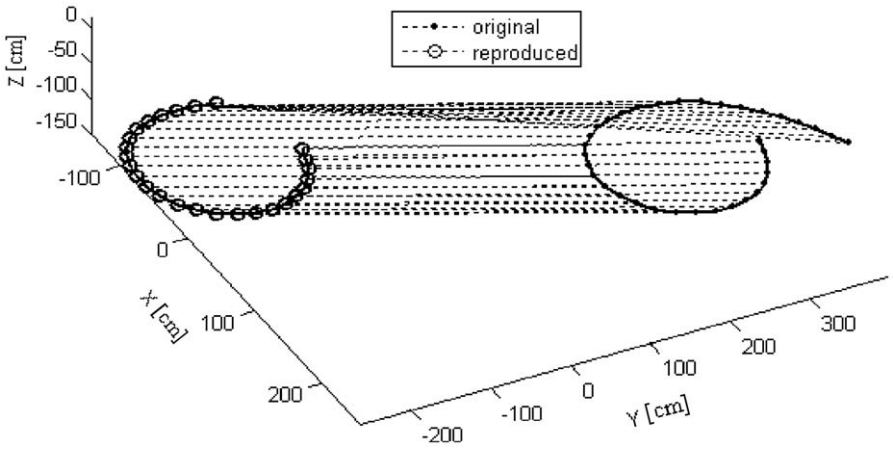

Fig. 28. Inter-trajectory warping paths based on the DTW-based signature matching (for the example in Fig. 24).

distances of Figs. 25–28 are 0.01062, 0.01637, 0.031049 and 0.136, respectively. This means that the similarity between the original and reproduced trajectories is high. Furthermore, it will be meaningful to give a possible statistical measurement of the matching error between references and reproductions. As some matching cases depend on the degree of speed control (Fig. 27) or occlusion (Fig. 28), here, we do not give them a generic measurement of the average DTW distance. But for the cases of the matching examples in Figs. 25 and 26, a series of different classes of motion trajectory instances are prepared to repeat the DTW matching and then to calculate the average DTW distances (Eq. (15)). As a result, an average distance data 0.0149 is obtained for 50 reproduction examples, whose lengths range from 23-point to 187-point. This explains why the reproduction algorithm is reliable with minor and acceptable errors.

The above examples and data show that both the two DTW-based measurements are helpful for analyzing the quality of the trajectory reproductions. In general, the reproduction algorithm is an important basis for replaying a robot task in the sign language-based LbD, and the DTW-based matching can offer a useful reproduction/learning metric.

6.5. Comments

In the above experiments, the generalized signature description is tested and different reproduction examples are presented and compared. The reproduction algorithm can be actually run online and it is shown that real time reproduction is achieved for the experimental trajectory examples. Yet, it should be noted that the actual reproduction time of a trajectory depends inherently on the trajectory's length.

This paper focuses on the backward reproduction ability of the signature description. Meanwhile, note that the signature's forward description-based sign language recognition has been explored in our previous study [31], where flexible signature recognition solutions are designed and better recognition performance is achieved compared with other descriptors.

Referring to the review and discussion in Section 1 and in Ref. [31], the signature's description mechanism and mutual description functions show more merits than the existing work. Also, based on the above experiment results plus some previous experiments data [30,31], the signature's overall advantages over other descriptors are revealed more clearly.

7. Conclusions

Free form 3-D motion trajectory is an appropriate motion feature for characterizing long-term, spatiotemporal motions. In this paper we introduce a trajectory signature and then formulate the trajectory reproduction problem to complete the signature's mutual description functions. On the one hand, the signature is advantageous

by serving as a systematic and generalized method for motion description; on the other hand, the trajectory reproduction is the basis to support motion reproduction, re-running, reconstruction and synthesis. The rich invariants make the signature flexible in both the mutual descriptions. The DTW-based signature matching proves to be an intuitive metric for measuring the reproductions. The experiments verify the validity and effectiveness of the reproduction algorithm.

This research reveals the feasibility and extensibility to potentially support wider applications. Robot learning will be a representative application of the mutual signature. The motion trajectories resulting from LbD demonstrations can be sequentially extracted, described, learned and then reproduced. While a single trajectory of the end-effector can be described by a single trajectory signature, the whole body locomotion can be described by the multiple signatures of the trajectories tracked from the multi-dimensional motions. The GMM-based class description can provide essential knowledge extracted from multiple different demonstrations of the same task class. In addition, this study can also benefit motion trajectory-based analysis of human behaviors and object motions, such as scene analysis in surveillance [50], behavior-based human communications like the interactions among the dummies and the visual interactions in virtual reality, visual teaching and evaluation of specific human behaviors like sportsmen's performance and dancing movements, motion-based content retrieval from image/video databases [51].

The future work will include two directions. One is the description and reproduction of semantic rich motion trajectories. This relates to the discussion in Section 2.2 about the detailed analysis and segmentation of a meaningful motion. The other will target the complicated motions that yield multiple interrelated trajectories. Basically this paper aims at a motion that is characterized by a single trajectory, or a motion class represented by multiple similar single trajectory-based instances. According to Sections 2.2 and 3.3, the current methods in describing and reproducing multiple interrelated trajectories still need user's manual interference, which is not that effective. In the future, we plan to improve this by investigating more efficient description and reproduction strategies, in which the spatiotemporal relations among the trajectories might need to be properly and naturally accommodated in certain way.

Acknowledgments

This work was fully supported by a grant from the Research Grants Council of Hong Kong [Project no. CityU117507]. The authors would like to thank Prof. Jianwei Zhang of University of Hamburg, Germany for the valuable discussions on this research topic. The first author is grateful to Xiaohua He for her support to this study.

References

[1] C. Rao, A. Yilmaz, M. Shah, View-invariant representation and recognition of actions, International Journal of Computer Vision 50 (2) (2002) 203–226.
[2] M. Bennewitz, W. Burgard, G. Cielniak, S. Thrun, Learning motion patterns of people for compliant robot motion, The International Journal of Robotics Research 24 (1) (2005) 31–48.
[3] B. Heisele, U. Kressel, W. Ritter, Tracking non-rigid, moving objects based on color cluster flow, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, June 1997, pp. 257–260.
[4] J. Min, R. Kasturi, Extraction and temporal segmentation of multiple motion trajectories in human motion, in: Proceedings of the IEEE Computer Vision and Pattern Recognition Workshop on Detection and Recognition of Events in Video, Washington, DC, USA, June 2004, pp. 118–122.
[5] A. Psarrou, S. Gong, M. Walter, Recognition of human gestures and behaviour based on motion trajectories, Image and Vision Computing 20 (5–6) (2002) 349–358.
[6] M.J. Black, A.D. Jepson, A probabilistic framework for matching temporal trajectories: condensation-based recognition of gestures and expressions, in: Proceedings of the European Conference on Computer Vision, Freiburg, Germany, 1998, vol. 1, pp. 909–924.
[7] K.K. Lee, M. Yu, Y. Xu, Modeling of human walking trajectories for surveillance, in: Proceedings of the IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 2003, pp. 1554–1559.
[8] D. Suc, I. Bratko, Skill modeling through symbolic reconstruction of operator's trajectories, IEEE Transactions on Systems, Man and Cybernetics, Part A 30 (6) (2000) 617–624.
[9] N. Suzuki, K. Hirasawa, K. Tanaka, Y. Kobayashi, Y. Sato, Y. Fujino, Learning motion patterns and anomaly detection by human trajectory analysis, in: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, October 2007, Montreal, Canada, pp. 498–503.
[10] D. Meyer, J. Psl, H. Niemann, Gait classification with HMMs for trajectories of body parts extracted by mixture densities, in: Proceedings of the British Machine Vision Conference, Southampton, England, 1998, pp. 459–468.
[11] J. Min, R. Kasturi, Activity recognition based on multiple motion trajectories, in: Proceedings of the International Conference on Pattern Recognition, Cambridge, UK, August 2004, vol. 4, pp. 199–202.
[12] T. Kollar, N. Roy, Trajectory optimization using reinforcement learning for map exploration, The International Journal of Robotics Research 27 (2) (2008) 175–196.
[13] E.A. Merchan-Cruz, A.S. Morris, Fuzzy-GA-based trajectory planner for robot manipulators sharing a common workspace, IEEE Transactions on Robotics 22 (4) (2006) 613–624.
[14] Y. Sato, K. Bernardin, H. Kimura, K. Ikeuchi, Task analysis based on observing hands and objects by vision, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 2002, pp. 1208–1213.
[15] A. Ude, C.G. Atkeson, M. Riley, Programming full-body movements for humanoid robots by observation, Robotics and Autonomous Systems 47 (2–3) (2004) 93–108.
[16] W. Chen, S.F. Chang, Motion trajectory matching of video objects, in: Proceedings of the SPIE/IS&T Storage and Retrieval for Media Databases, San Jose, CA, January 2000, pp. 544–553.
[17] V.V. Kindratenko, On using functions to describe the shape, Journal of Mathematical Imaging and Vision 18 (33) (2003) 225–245.
[18] J. Xie, P.-A. Heng, M. Shah, Shape matching and modeling using skeletal context, Pattern Recognition 41 (5) (2008) 1756–1767.
[19] D. Zhang, G. Lu, Review of shape representation and description techniques, Pattern Recognition 37 (1) (2004) 1–19.
[20] E. Bribiesca, A chain code for representing 3D curves, Pattern Recognition 33 (5) (2000) 755–765.
[21] A. Masood, Optimized polygonal approximation by dominant point deletion, Pattern Recognition 41 (1) (2008) 227–239.
[22] T. Suk, J. Flusser, Projective moment invariants, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (10) (2004) 1364–1367.
[23] M.C. Shin, L.V. Tsap, D.B. Goldgof, Gesture recognition using Bezier curves for visualization navigation from registered 3-D data, Pattern Recognition 37 (5) (2004) 1011–1024.
[24] F.S. Cohen, Z. Huang, Z. Yang, Invariant matching and identification of curves using B-splines curve representation, IEEE Transactions on Image Processing 4 (1) (1995) 1–10.
[25] J. Aleotti, S. Caselli, Robust trajectory learning and approximation for robot programming by demonstration, Robotics and Autonomous Systems 54 (5) (2006) 409–413.
[26] G. Chuang, C.C. Kuo, Wavelet descriptor of planar curves: theory and applications, IEEE Transactions on Image Processing 5 (1) (1996) 56–70.
[27] P.R.G. Harding, T.J. Ellis, Recognizing hand gesture using Fourier descriptors, in: Proceedings of the International Conference on Pattern Recognition, vol. 3, Cambridge, UK, 2004, pp. 286–289.
[28] F. Mokhtarian, S. Abbasi, J. Kittler, Robust and efficient shape indexing through curvature scale space, in: Proceedings of the British Machine Vision Conference, Edinburgh, UK, 1996, pp. 53–62.
[29] S. Tabbone, L. Wendling, J.P. Salmon, A new shape descriptor defined on the Radon transform, Computer Vision and Image Understanding 102 (1) (2006) 42–51.
[30] S.D. Wu, Y.F. Li, On signature invariants for effective motion trajectory recognition, The International Journal of Robotics Research 27 (8) (2008) 895–917.
[31] S.D. Wu, Y.F. Li, Flexible signature descriptions for adaptive motion trajectory representation, perception and recognition, Pattern Recognition 42 (1) (2009) 194–214.
[32] V. Parameswaran, R. Chellappa, View invariance for human action recognition, International Journal of Computer Vision 66 (1) (2006) 83–101.
[33] Y. Kuniyoshi, M. Inaba, H. Inoue, Learning by watching: extracting reusable task knowledge from visual observation of human performance, IEEE Transactions on Robotics and Automation 10 (6) (1994) 799–822.
[34] S. Calinon, F. Guenter, A. Billard, On learning, representing and generalizing a task in a humanoid robot, IEEE Transactions on Systems, Man and Cybernetics, Part B 36 (5) (2006) 286–298.
[35] J. Takamatsu, K. Ogawara, H. Kimura, K. Ikeuchi, Recognizing assembly tasks through human demonstration, The International Journal of Robotics Research 26 (7) (2007) 641–659.
[36] H. Friedrich, S. Munch, R. Dillmann, S. Bocionek, M. Sassin, Robot programming by demonstration (RPD): supporting the induction by human interaction, Machine Learning 23 (2–3) (1996) 163–189.
[37] C.A. Acosta-Calderon, H. Hu, Imitation towards service robotics, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, 2004, pp. 3726–3731.

[38] K.R. Dixon, P.K. Khosla, Learning by observation with mobile robots: a computational approach, in: Proceedings of the IEEE International Conference on Robotics and Automation, New Orleans, LA, USA, April 2004, vol. 1, pp. 102–107.
[39] R. Poppe, Vision-based human motion analysis: an overview, Computer Vision and Image Understanding 108 (1–2) (2007) 4–18.
[40] J.K. Aggarwal, S. Park, Human motion: modeling and recognition of actions and interactions, in: Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission, Thessaloniki, Greece, pp. 640–647.
[41] J. Feng, Combining minutiae descriptors for fingerprint matching, Pattern Recognition 41 (1) (2008) 342–352.
[42] B. Li, Q. Meng, H. Holstein, Articulated motion reconstruction from feature points, Pattern Recognition 41 (1) (2008) 418–431.
[43] O. Arikan, D.A. Forsyth, J. O'Brien, Motion synthesis from annotations, ACM Transactions on Graphics 33 (3) (2003) 402–408.
[44] Z. Popovic, Editing dynamic properties of captured human motion, in: Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, April 2000, vol. 1, pp. 670–675.
[45] T. Kwon, Y.S. Cho, S.I. Park, S.Y. Shin, Two-character motion analysis and synthesis, IEEE Transactions on Visualization and Computer Graphics 14 (3) (2008) 707–720.
[46] A. Nakazawa, S. Nakaoka, T. Shiratori, K. Ikeuchi, Analysis and synthesis of human dance motions, in: Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Tokyo, Japan, July–August 2003, pp. 83–88.
[47] T.B. Moeslund, E. Granum, A survey of computer vision-based human motion capture, Computer Vision and Image Understanding 81 (3) (2001) 231–268.
[48] S.D. Wu, Y.F. Li, J.W. Zhang, A hierarchical motion trajectory signature descriptor, in: Proceedings of the IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, May 2008, pp. 3070–3075.
[49] J.J. Stoker, Differential Geometry, Wiley-Interscience, New York, 1969.
[50] T. Xiang, S. Gong, Activity based surveillance video content modeling, Pattern Recognition 41 (7) (2008) 2309–2326.
[51] M.R. Daliri, V. Torre, Robust symbolic representation for shape recognition and retrieval, Pattern Recognition 41 (5) (2008) 1782–1798.

About the Author—Shandong Wu received the B.Sc. and M.Sc. degrees both in Computer Science from School of Computer Engineering and Science at Shanghai University, Shanghai, China, in 1998 and 2003, respectively. He was a Ph.D. student at Department of Manufacturing Engineering and Engineering Management at City University of Hong Kong, Hong Kong and received the Ph.D. degree in 2008. Dr. Wu is currently a Postdoctoral Associate of Computer Vision Lab in the School of Electrical Engineering and Computer Science at the University of Central Florida, USA. His main research area consists of computer vision and robotics particularly human behavior description and recognition and robot learning by demonstration.

About the Author—Y.F. LI received the Ph.D. degree in Robotics from Oxford University, Oxford, UK, in 1993. From 1993 to 1995, he worked as a Postdoctoral Researcher in Department of Computer Science at University of Wales, Aberystwyth, UK. He joined City University of Hong Kong in 1995. He has published over 100 papers in refereed international journals and conferences. His research interests include robot vision, visual tracking, robot sensing and sensor-based control, mechatronics, and automation. He is a Senior Member of the IEEE. Dr. Li is an Associate Editor of IEEE Transactions on Automation Science and Engineering.