Unsupervised Trajectory Segmentation for Surgical Gesture Recognition in Robotic Training

Fabien Despinoy, David Bouget, Germain Forestier, Cédric Penet, Nabil Zemiti, Philippe Poignet, Pierre Jannin

Unsupervised Trajectory Segmentation for Surgical Gesture Recognition in Robotic Training

Fabien Despinoy*, David Bouget, Germain Forestier, Cédric Penet,
Nabil Zemiti, Philippe Poignet, and Pierre Jannin

***Abstract*—Dexterity and procedural knowledge are two critical skills surgeons need to master to perform accurate and safe surgical interventions. However, current training systems do not allow to provide an in-depth analysis of surgical gestures to precisely assess these skills. Our objective is to develop a method for the automatic and quantitative assessment of surgical gestures. To reach this goal, we propose a new unsupervised algorithm that can automatically segment kinematic data from robotic training sessions. Without relying on any prior information or model, this algorithm detects critical points in the kinematic data which define relevant spatio-temporal segments. Based on the association of these segments, we obtain an accurate recognition of the gestures involved in the surgical training task. We then perform an advanced analysis and assess our algorithm using datasets recorded during real expert training sessions. After comparing our approach with the manual annotations of the surgical gestures, we observe 97.4% accuracy for the learning purpose and an average matching score of 81.9% for the fully-automated gesture recognition process. Our results show that trainees workflow can be followed and surgical gestures may be automatically evaluated according to an expert database. This approach tends towards improving training efficiency by minimizing the learning curve.**

***Index Terms*—Unsupervised Trajectory Segmentation, Surgical Gesture Recognition, Machine Learning, Classification, Surgical Skills Training, Robotic Surgery, Teleoperation.**

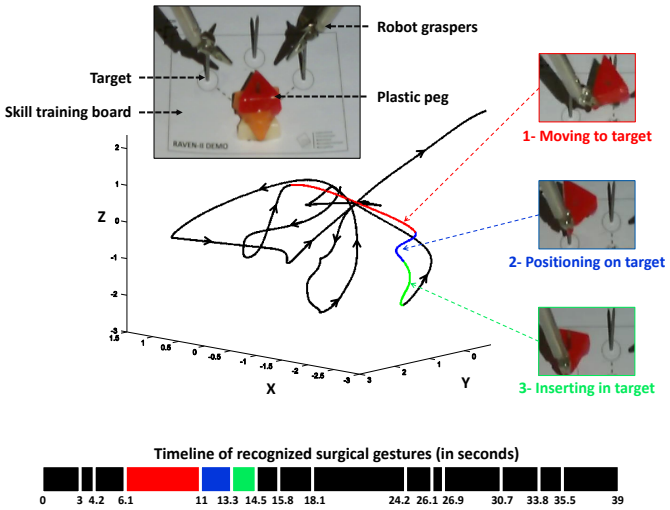


Fig. 1. Example of partial 3D motion segmentation from the left hand, for a pick-and-place task during a robotic surgical training session. Black arrows show the direction of motion. In this figure, three surgical gestures (surgemes) are highlighted: Surgeme 1 (in red) refers to "Moving to target", Surgeme 2 (in blue) is "Positioning on target" and Surgeme 3 (in green) corresponds to "Inserting in target".

I. Introduction

SURGICAL trainees conventionally learn and practice laparoscopic interventions on standard pelvi-trainer systems in order to improve their technical skills. Their performance is evaluated manually, using predefined scoring methods based on rating scales such as OSATS, GOALS or MISTELS [1]. These methods require the participation of an expert who observes and quantifies the trainee's skills. However, recent technological progress has allowed the development and the integration of new surgical devices for training and interventional purposes such as the da Vinci® robot [2, 3]. Thanks to advances in human-machine interface and robot design, these devices have expanded the range of capabilities in terms of comfort and dexterity, thereby improving overall operating conditions for surgeons. Furthermore, the complementary da Vinci® Skills Simulator trainer introduced an advanced evaluation of surgical dexterity and shifted the Halsted's paradigm, "see one, do one, teach one", towards a new educational heuristic: "perfect practice makes perfect" [4]. Indeed, the use of such advanced training system makes the automatic assessment of operator performance possible. At the end of a training session, useful feedback is provided to the operator without any involvement from experts. This feedback is based on the automatic computation of multiple metrics (i.e., completion time, instrument collisions, workspace overlapping, traveled length or economy of motion) which yield performance scores. The evolution of these scores guides and encourages skill honing [5]. However, the mere objective evaluation of technical skills is insufficient. In fact, surgical training mostly relies on the repetition and execution of several different gestures, and qualitative criteria do not provide enough information to replicate them. A reliable solution is

Manuscript received July 1, 2015; revised September 14, 2015; accepted October 13, 2015. Date of current version October 18, 2015. *Asterisk indicates corresponding author.*

This work was supported in part by the French ANR within the Investissements d'Avenir Program (Labex CAMI, ANR-11-LABX0004); by the Equipex ROBOTEX Program (ANR-10-EQPX-44-01); and by the Région Languedoc-Roussillon.

F. Despinoy, N. Zemiti and P. Poignet are with the LIRMM - CNRS, UMR 5506, Université Montpellier, Montpellier, F-34000, France (e-mail: fabien.despinoy@lirmm.f; nabil.zemiti@lirmm.fr; philippe.poignet@lirmm.fr).

D. Bouget, C. Penet and P. Jannin are with the LTSI - INSERM, UMR 1099, Université Rennes 1, Rennes, F-35000, France (e-mail: david.bouget@univ-rennes1.fr; cedric.penet@univ-rennes1.fr; pierre.jannin@univ-rennes1.fr).

G. Forestier is with the MIPS (EA 2332), Université de Haute Alsace, Mulhouse, F-68100, France (e-mail: germain.forestier@uha.fr).

to take the operating gesture workflow into account, in order to provide more intuitive training as well as more accurate gesture and procedural knowledge assessment solutions [6].

In the literature, modeling and evaluating procedural knowledge and surgical activities have been extensively studied [7], and both refer to the notion of Surgical Process Modeling (SPM). The SPM methodology is entirely articulated around the concept of granularity. More precisely, the level of granularity defines the level of abstraction at which the surgical procedure is described. A hierarchical decomposition is employed to structure the different interactions between the surgical team and new technologies (e.g., communication, surgical activities). This formal decomposition was previously introduced for skill assessment purposes using ontological descriptions [8, 9]. However, since surgical gestures represent the lowest granularity level of the SPM decomposition, more precise and relevant information, such as kinematic data from tool motions [10], is necessary to recognize them.

Sugino et al. [11] proposed a method to both identify surgical gestures and assess surgeon expertise. This method relies on velocity and acceleration combined with standard metrics. Similarly, in [12]–[15], the authors used kinematic data during training sessions. Based on multiple transformations from the kinematic observations, such as Descriptive Curve Coding (DCC) or Gaussian models combined with Linear Discriminant Analysis (LDA), they encoded each gesture in a multi-state Hidden Markov Model (HMM). Their dual objective was to segment and recognize surgical gestures based on the temporal model of a surgical task. In a similar vein, more recent works [16]–[18] have proposed combining video and kinematic data, and have proved that mixing information from multiple modalities strongly improves gesture recognition capacities when building a temporal model. Despite such improvements, all these works relied on the assumption that the training sessions already comprised a breakdown of specific, recognizable gestures, which requires significant preprocessing input from experts. To avoid unnecessary assumptions, our work focuses on a non-supervised segmentation technique for gesture recognition in a similar training context.

Many works have proposed unsupervised segmentation techniques as well as trajectory clustering algorithms. However, only a few of them were applied to human gestures, and especially to hand trajectory segmentation purposes. Schulz et al. [19] proposed a method for segmenting joint-angle trajectories based on 3D positions but applied their algorithm to articulated human motions. In a different proposal, Popa et al. [20] addressed the problem of hand gesture recognition based on 2D trajectories. These trajectories were first segmented into strokes. Then, using vector-quantified histograms of motion directions, the authors were able to successfully identify basic gestures, such as drawing a square or a circle on a plane, in real time. In simpler contexts with controlled motions, other authors have also used different representations, such as zero-velocity crossing [21], and velocity and direction [22]. However, these individual representations do not provide enough information. They only capture a specific subspace of the trajectory (e.g., velocity, rotation). Holden et al. [23] proposed a combined method for both segmenting and recognizing surgical gestures during needle insertion interventions using both position and quaternion information. However, even if their proposed algorithm fits real-time requirements thanks to Markov Modeling, the tasks performed only involved simple 3D motions for needle insertion/removal (i.e., up, down, rotate). In fact, laparoscopic surgical gestures do not follow predefined patterns and instead involve complex motions which cannot be easily recognized. Therefore, detecting and recognizing surgeon's gestures requires more acute observations than the ones proposed so far.

In this paper, we therefore propose a new bottom-up approach to segment and recognize surgical gestures, also called *surgemes* [24]. Surgemes define surgical motion units with explicit semantic sense (i.e., grabbing the needle). Each surgeme is composed of a set of primitives, called *dexemes*, which are numerical representations of sub-gestures necessary to perform a surgeme [25]. Dexemes only involve one hand and are devoid of semantic sense (i.e., go towards, turn left, wait). Fig. 1 shows an example of a surgical training session in which three surgemes are highlighted. Our objective is to identify all surgemes involved in the robotic surgical training task, without relying on any prior information. This bottom-up approach starts from the measurement and computation of multiple kinematic signals, which allow the capture of all the operator's intentions. Next, we automatically detect relevant timestamps in the data that define dexemes. Then, we compute their relevance with respect to spatio-temporal variations and compare three different dissimilarity metrics for this purpose. Further, we apply machine learning techniques to retrieve the entire surgeme workflow by recognizing and associating all the dexemes. Finally, we evaluate the performance of our unsupervised segmentation and recognition approach based on data acquired from laparoscopic surgery training sessions using a robotic platform.

The remaining sections are organized as follows. In Section II, we present the overall method for surgical gesture segmentation and recognition. Section III describes the robotic platform, the acquired datasets and the different validation studies. Section IV presents the qualitative and quantitative assessments of our method. Finally, we discuss our analysis and results in Section V.

II. Methods

The overall pipeline process of the proposed algorithm is presented in Fig. 2. In the first step, unsupervised segmentation of kinematic data is performed through a four-stage process. It provides a relevant selection of dexemes without any prior information. In the second step, another three-stage process is used to learn and recognize the different surgemes involved in the surgical training task.

A. Unsupervised Trajectory Segmentation

The proposed segmentation method involves four distinct stages. The first stage uses the input kinematic data to compute additional 3D-invariant kinematic signals and processes these signals to reduce noise and normalize them. The second stage detects relevant timestamps inside these signals for temporal

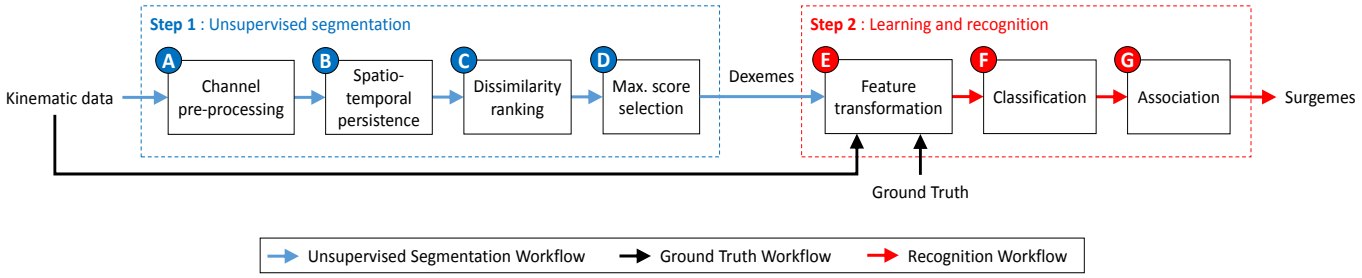


Fig. 2. Overview of the proposed pipeline for segmentation and recognition of surgical gestures. Step 1 performs unsupervised segmentation from kinematic data to define relevant dexemes. Step 2 learns features from these dexemes to recognize them in a future sequence. Surgemes are found by associating dexemes with corresponding labels.

dexeme decomposition. The third stage scores timestamps relevance in terms of spatial and temporal dissimilarity. The last stage selects the best scores to provide the most relevant dexeme segmentation.

1) Channel pre-processing: This part describes computations performed in stage A of Fig. 2. Surgical tool trajectories are represented by a set of rigid transformations, which include the position of each tooltip $\{X_i(t), Y_i(t), Z_i(t) | i \in [1,2]\}$ and the orientation via a rotation matrix, with respect to the robots reference frame. The rotation matrix is converted into quaternion representation $\{Qw_i(t), Qx_i(t), Qy_i(t), Qz_i(t) | i \in [1,2]\}$ in order to avoid singularity issues and to obtain a more compact representation for faster computation time. Additionally, a 3D-invariant signal representation is extracted from the trajectories [26, 27]. This allows us to model motions independently of tooltip's position using the following description. Let $\Gamma(t)$ be a free form motion trajectory with $t \in [1, N]$, where N is the trajectory length. Its 3D Euclidean signature S is defined by four differential invariants: curvature (κ), torsion (τ) and their first order derivatives (κ_s and τ_s) with respect to the Euclidean arc-length parameter, in the following form:

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

where

$$\kappa(t) = \frac{\|\dot{\Gamma}(t) \times \ddot{\Gamma}(t)\|}{\|\dot{\Gamma}(t)\|^3} \tag{2}$$

$$\tau(t) = \frac{(\dot{\Gamma}(t) \times \ddot{\Gamma}(t)) \cdot \dddot{\Gamma}(t)}{\|\dot{\Gamma}(t) \times \ddot{\Gamma}(t))\|^2} \tag{3}$$

$$\kappa_s(t) = \frac{\mathrm{d}\kappa(t)}{\mathrm{d}s} = \frac{\mathrm{d}\kappa(t)}{\mathrm{d}t} \cdot \frac{\mathrm{d}t}{\mathrm{d}s} = \frac{\mathrm{d}\kappa(t)}{\mathrm{d}t} \cdot \frac{1}{\|\dot{\Gamma}(t)\|} \tag{4}$$

$$\tau_s(t) = \frac{\mathrm{d}\tau(t)}{\mathrm{d}s} = \frac{\mathrm{d}\tau(t)}{\mathrm{d}t} \cdot \frac{\mathrm{d}t}{\mathrm{d}s} = \frac{\mathrm{d}\tau(t)}{\mathrm{d}t} \cdot \frac{1}{\|\dot{\Gamma}(t)\|} \tag{5}$$

In our work, we used numerical approximations of each component of S relying on multiple neighbor approximation [28]. This allowed us to reduce computation time for the 3D-invariant descriptors. Furthermore, the grasping angles of the robotic tools are captured in order to obtain a complete description of the left and right surgical tool motions. In the end, a total of 24 variables (i.e., 3 for the position, 4 for the quaternion, 4 for the 3D Euclidean signature and 1 for the grasping angle, for each hand respectively) are acquired and subsequently used as input channels.

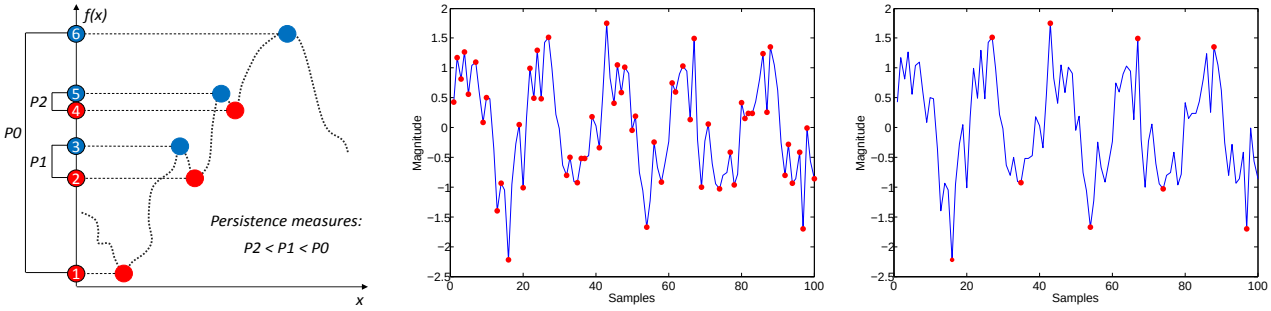
Then, a low-pass filter is used to minimize measurement noise and to capture only voluntary motions. It is set with a corner frequency of 1.5Hz to preserve fundamental hand motion frequencies [29]. It also has unity gain before the corner frequency and a high attenuation beyond 10Hz [30].

Finally, to manipulate and enable a fair comparison between these signals from different acquisitions, a normalization step is performed using mean and variance values. Let R_i be a raw signal of length L, the normalization is specified as:

$$N_i(t) = \frac{1}{\sigma_i^2}(R_i(t) - \mu_i)) \quad t \in [1, L] \tag{6}$$

where μ_i and σ_i^2 are respectively the mean and variance, and N_i is the corresponding normalized signal. At the end of this step, smoothed and normalized data are obtained for the following spatio-temporal analysis.

2) Spatio-temporal persistence: Using previously processed channels as input, this part presents computations performed in stage B of Fig. 2. Our segmentation approach assumes that any dexeme is characterized by a pair of critical points, which are defined by geometrical variations in the input signals (e.g., curves, straight lines). These critical points help to identify the intentions of the operator. For this purpose, we used a new topological simplification technique applied to 2D scalar fields introduced by [31]. The proposed simplification methods aims at successively removing connected critical pairs of points by relying on the notion of persistence [32]. Using bi-Laplacian optimization, the algorithm successively computes the lifetime of connected components. More precisely, it measures the difference in the signal value between specific minima and maxima. This concept is illustrated with Fig. 3(a). Here each maximum is paired with a preceding minimum (note that paired points are not necessarily adjacent to each other in the initial signal). Then persistence of the paired points is computed and ranked. At the end, by choosing a persistence threshold, characteristic points in the signal can be selected. Another example is provided in Fig. 3(b) and Fig. 3(c) where characteristic points in a signal are selected by relying on the persistence threshold only. We applied this selection algorithm to our input channels in order to find extrema of each signal. Persistence thresholds were empirically defined for the experiments.



(a) Persistence concept in 1D. (b) All extrema: persistence > 0. (c) Extrema with persistence > 1.5.

Fig. 3. Persistence simplification successively achieves the cancellation of critical paired points. The example shows a sinusoidal motion with additive random noise where extrema are ranked using the persistence measure. Final selection is performed using a specific threshold.

3) Dissimilarity ranking: Following this simplification process, we quantified the relevance of the remaining points, considering each one as a possible delimitation solution to define a dexeme. In order to avoid assumptions on the gesture's length (e.g., using temporal windows averaging, which is task- and operator-dependent), we employed a dedicated scoring method to rank delimitation points (see stage C in Fig. 2). For this purpose, we used and compared three different dissimilarity metrics to quantify both shape and time variations of consecutive segments, based on multidimensional time series computation (i.e., each point delimits two segments, where each segment is characterized by multiple input signals and we quantify the dissimilarity between this pair of signals).

The first dissimilarity metric used was the Hausdorff distance [33]. Assuming two sets of points A and B, the Hausdorff distance h is described as:

$$h(A,B) = \max_{a\in A}(\min_{b\in B}\|a-b\|) \tag{7}$$

The symmetry of the metric is restored by using the maximum between $h(A,B)$ and $h(B,A)$.

The second metric studied was the discrete Fréchet distance [34]. The basic Fréchet distance is used for comparing continuous shapes, such as curves and surfaces, and is defined by reparametrizing the shapes. Since it takes the continuity of the shapes into account, it is generally considered a more appropriate distance measure for curves than Hausdorff's. A specific variant of the Fréchet distance is the discrete Fréchet distance, which is naturally used for polygonal curves. Consider two polygonal curves P and Q in $\mathbb{R}^c$ given by their sequences of vertices $\langle p_1,...,p_n\rangle$ and $\langle q_1,...,q_m\rangle$ respectively. A coupling C of the vertices of P and Q is a sequence of pairs of vertices $C = \langle C_1,...,C_k\rangle$ with $C_r = (p_i,q_j)$ for all $r = 1,...,k$ and some $i \in \{1,...,n\}, j \in \{1,...,m\}$ fulfilling $C_1 = (p_1,q_1)$, $C_k = (p_n,q_m)$, $C_r = (p_i,q_j)$ and $C_{r+1} \in \{(p_{i+1},q_j),(p_i,q_{j+1}),(p_{i+1},q_{j+1})\}$ for $r = 1,...,k-1$. Let $|\cdot|$ denote the norm on $\mathbb{R}^c$, then the discrete Fréchet distance is defined as:

$$F(P,Q) = \min_{coupling\ C}\max_{(p_i,q_j)\in C}|p_i - q_j| \tag{8}$$

where C ranges over all coupling of the vertices of P and Q. The main advantage of this metric is that it allows fast computation by only taking into account the distances between vertices.

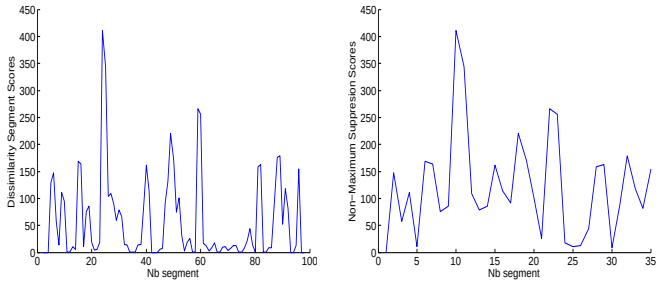
The last metric considered for dissimilarity ranking and the most used in the literature to compare two time series in terms of spatio-temporal variations is Dynamic Time Warping (DTW) with the Euclidean distance [35]. While the Euclidean distance cannot capture flexible similarities, DTW allows one to measure similarities between two sequences which may vary in time or speed. Its main advantage is the computation of a point-to-point association between two temporal sequences, with respect to both time and space variations. Thus, DTW finds the optimal alignment (or coupling) between sequences by aligning similar coordinates of both sequences. The cost of the optimal alignment between sequences $A = \langle a_1,...,a_M\rangle$ and $B = \langle b_1,...,b_N\rangle$ is recursively computed by:

$$D(A_i,B_j) = \delta(a_i,b_j) + min\begin{Bmatrix} D(A_{i-1},B_{j-1}) \\ D(A_i,B_{j-1}) \\ D(A_{i-1},B_j) \end{Bmatrix} \tag{9}$$

where $\delta(a_i,b_j)$ is the norm of the Euclidean distance between a_i and b_j. The overall similarity of the two time series is given by $D(A_{|A|},B_{|B|}) = D(A_M,B_N)$.

By applying these dissimilarity metrics, we were able to compute the relevance of the remaining critical points according to spatio-temporal variations, and rank their scores in order to keep the best ones. However, to avoid a task- or operator-dependent threshold selection, we used a more generic method for maximum score selection.

4) Maximum score selection: Based on the previously computed dissimilarity scores, a pairwise Non-Maximum Suppression (NMS) procedure was employed to avoid biased threshold selection [36] (see stage D in Fig. 2). The NMS performs local maximum search, where a local maximum is greater than all its neighbors (excluding itself). The advantage of this method is that it preserves the topology of the dissimilarity score signal and relies only on the inner score pattern (see Fig. 4). For the maximum selection purpose, we used a 3-Neighborhood parameter. As a result, the best timestamps were selected to define the overall sequence of dexemes. With the proposed selection method, all components (i.e., input channels) were treated equivalently in our segmentation method. There was no specific selection of the nature of the temporal delimitation of the dexemes. Thus, this method does not rely on any prior knowledge regarding the context of execution.



(a) Dissimilarity scores. (b) NMS selection.

Fig. 4. The NMS algorithm computes and selects the most relevant timestamps based on the spatio-temporal dissimilarity scores.

B. Dexeme Learning and Surgeme Recognition

This section focuses on the classification task using a learning paradigm. Dexemes provided from the previous stage of the pipeline are transformed and learned through machine learning algorithms in order to eventually obtain the overall sequence of surgemes performed by the operator.

1) Feature transformation: For this step (stage E of Fig. 2), a descriptive signature is required to represent each dexeme in a specific manner. We based our work on a previous analysis where Chebyshev polynomial and Discrete Fourier series were compared to standard Polynomial approximations [37]. As stated by the authors, the latter decomposition provided sound results for spatio-temporal trajectory classification.

First, a base transformation was applied to dexeme segments, ensuring that all input channels would start from the same origin. Then, a polynomial approximation was applied using the following process. Let $S(x)$ be a data sequence of size $x \in [1, n]$. It can be approximated by a polynomial $y = a_0 + a_1 x + ... + a_m x^m$ of degree $m < n$ using the Least Squares method. Thus, a descriptive signature vector is created for each dexeme by concatenating all polynomial coefficients of the input channels. For our purposes, we tested several degrees of approximation and discussed their impact on the recognition performance in Section IV.

2) Classification algorithms: The objective of this stage is to provide discrete labels from input features (stage F of Fig. 2). We used k-Nearest Neighbors (k-NN) and Support Vector Machines (SVM) [38] to automatically classify dexemes obtained from the unsupervised trajectory segmentation step.

K-NN is a non parametric method for classification and regression. It consists in finding the k closest examples in the training database. Then, for a classification application, a majority voting is performed using the Euclidean distance to assign a class to the current sample. In this work, multiple values of k were considered and discussed according to their recognition performance.

SVM is an optimization algorithm which tries to find, in a binary problem, a hyperplane that maximally separates both classes. For this purpose, it determines a linear function of the form $f(x) = w^T x + b$, where w is the separating hyperplane, x the training samples and b an offset. However, some problems are not linearly separable. In this case, a kernel trick could be used to find a separating hyperplane in a different subspace. Denoting such mapping as $\phi_k(\cdot)$, the kernel SVM classifier is found by solving the optimization problem:

$$\begin{aligned} \underset{w,b,z}{minimise} \quad & \frac{1}{2} w^T w + C \sum_{i=1}^{N} z_i \\ \text{subject to} \quad & y_i(w^T \phi_k(x_i) - b) + z_i \geq 1 \\ & z_i \geq 0, \quad i = 1, ..., N \end{aligned} \tag{10}$$

where z_i is a non-negative slack variable that penalizes the misclassification of sample i, and the parameter C allows the weighting of this penalization. In this work, we used a Radial Basis Function (RBF) kernel. The RBF kernel allows one to classify two samples u and v that are not linearly separable using the following function:

$$K(u, v) = exp\left(\gamma \|u - v\|^2\right) \tag{11}$$

where $\gamma = -\frac{1}{2\sigma^2}$ affects decision boundaries. For both classifiers, the output labels provided determine the assignment of the current dexeme to a distinct surgeme.

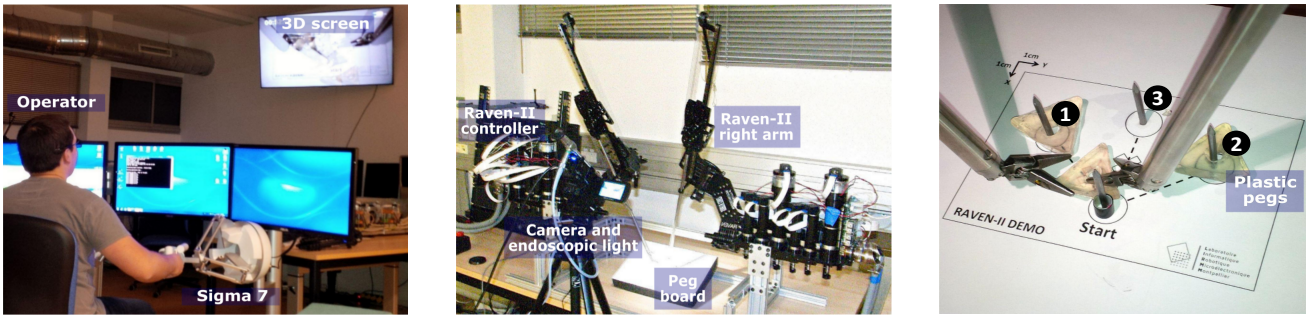
3) Dexeme association: The last stage executes a temporal association of consecutive dexeme labels in order to produce a surgeme sequence as the output of the overall pipeline (stage G of Fig. 2). However, since a surgeme is composed of several dexemes, smoothing is applied to the dexeme sequence provided to prevent outliers (i.e., modifying dexemes which were classified differently than their neighbors). We used a 3-Neighborhood smoothing for robust estimation of the surgeme sequence.

III. Setup and Validation Studies

To evaluate the proposed approach, several operators were asked to execute predefined surgical training tasks on the telesurgical robotic platform presented in Section III-A. By recording the surgical tool motions involved in these training tasks, we created two distinct datasets presented in Section III-B. A verification study for the segmentation method and two validation studies for the assessment of both segmentation and classification steps are introduced in detail in Sections III-C and III-D.

A. Robotic Setup

For surgical training purposes, we used an advanced robotic teleoperation platform (see Fig. 5). The Raven-II robot was chosen because it closely mimics the da Vinci® systems motions [39]. Composed of two serial arms, each with 7 Degrees of Freedom (DoF), the Raven II allows the operator to move surgical needle graspers via a cable-driven mechanical architecture. Moreover, two Sigma 7 master interfaces were employed to teleoperate the Raven-II. Also offering 7 DoFs, these interfaces provide enough dexterity to precisely handle the distant robot with a "Position-Position" control loop running at 1kHz. This setup made surgical training on phantoms possible and allowed the acquisition and computation of the 24 kinematic variables required for our gesture recognition process



(a) The master console with two Sigma 7 and a 3D screen for visual feedback.

(b) The Raven-II robot and a 3D camera enable teleoperated control of laparoscopic tools.

(c) The two surgical graspers and the training board for the pick-and-place task.

Fig. 5. The telesurgical robotic platform is composed of two main sites: master and slave sides. The master interfaces allow the operator to fully control the Raven-II surgical tools in order to complete the pick-and-place training task.

B. Trajectory Datasets

Two different datasets were acquired to validate the proposed approach. Each dataset consisted of multiple trajectories involving the same training task, executed by three experts. These experts were two urologists who regularly perform surgery with the da Vinci® robot and a teleoperation system engineer. Participants were briefed about the setup and the specificities of the tasks to ensure that they would perform them appropriately and consistently (i.e., in the same manner and without doing mistakes).

The first dataset consisted of three trajectories (one from each different operator). The aim of this task was to draw the letter "R" with one hand (left or right, depending on the operator's hand dexterity). We used this dataset to qualitatively assess the segmentation method developed in this work.

The second dataset consisted of nine trajectories and videos, three per operator. The aim was to execute a surgical training task directly inspired by SAGES and FLS guidelines [40]. This training task involved peg transfers to several target locations following the workflow described hereafter:

1) Pick the first peg with the left tool and insert it into target 1 (leftmost pin of Fig. 5(c)),
2) Pick the second peg with the right tool and insert it into target 2 (rightmost pin),
3) Pick the last peg with the left or right tool and progress towards the center of the peg board. Grab it with the other available tool in order to insert it into target 3 (uppermost pin).

Next, in order to define the ground truth annotation, the acquisition modalities were manually synchronized and the manual segmentation of the kinematic data was achieved using video clipping. Note that, even if the ground truth was built by an expert, it could suffer from uncertainties and subjectivity. Here, we supposed that the expert have the required knowledge to correctly (i.e., optimally) perform this task, without any bias. From the annotations, twelve surgemes were thusly identified in this pick-and-place task (see Table I). They presented with the possibility that they would appear more than once during a session. We used this dataset to quantitatively assess the segmentation and the classification steps involved in our gesture recognition pipeline.

TABLE I
SURGEME'S VOCABULARY FOR THE PICK-AND-PLACE TRAINING TASK

N°	Definition	N°	Definition
1	Wait	7	Positioning on target
2	Reaching peg	8	Inserting in target
3	Precise positioning	9	Releasing peg
4	Grabbing peg	10	Moving to wait
5	Extracting peg	11	Moving back to center
6	Moving to target	12	Moving to end position

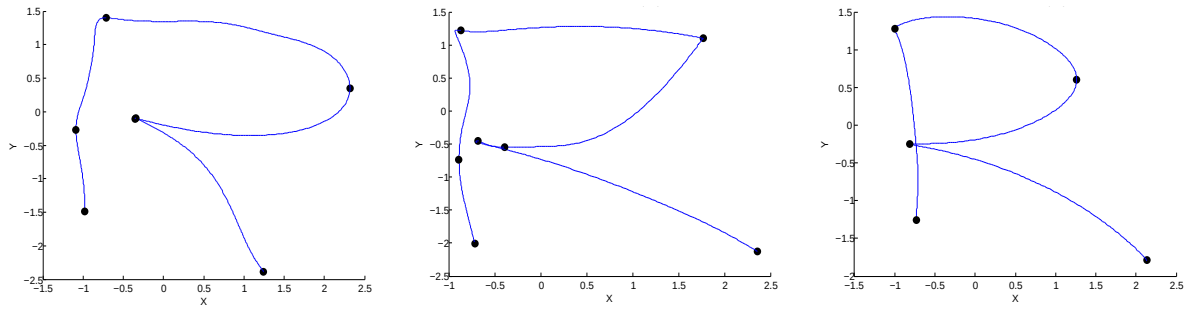
C. Segmentation Verification Study

This qualitative study aimed to visually analyze the output of the proposed segmentation algorithm. Due to the subjective nature of segmentation, it is difficult to assert its soundness without any specific medical application in mind. However, the robustness of any segmentation with respect to shape variability is one of the most important aspects of this type of assessment. Our operators executed trajectories with one hand to draw the letter "R". The only inputs given to the segmentation algorithm were the position $\{X(t), Y(t)\}$ and the computed short invariant signature $S = \{\kappa(t), \kappa_s(t)\}$. The total invariant signature was not computed because only planar shapes were considered, leading to null torsion. The results based on the three dissimilarity metrics are detailed in Section IV-A.

D. Validation Studies for Surgeme Recognition

In order to assess the performance of the classification process, we carried out two studies. The first study consisted in using ground truth annotations as input to assess the classification performance (Ground Truth and Recognition Workflows in Fig. 2). The second study involved the combination of the proposed segmentation step with the classification step to assess the overall pipeline for surgical gesture recognition (Unsupervised Segmentation and Recognition Workflows in Fig. 2).

1) Ground truth and classification: The first validation was achieved using manual annotations of the surgemes in order to provide "gold-standard" results for the classification step. We



(a) Operator N°1. (b) Operator N°2. (c) Operator N°3.

Fig. 6. Verification of the segmentation algorithm based on the "R" letter trajectories using DTW as the dissimilarity metric.

used a cross-validation process on the overall database with leave-one-out sessions, and averaged scores to present means and standard deviations as results. For both classification algorithms, recognition performance was assessed according to the following parameters:

- Degree of the feature transformation where $m \in \{1, 3, 5, 7, 9\}$,
- Number of neighbors for the $k-$NN classifier where $k \in \{1, 3, 5, 7\}$,
- Tradeoff penalty value for the SVM classifier where C is evenly spaced in log-space from 10^{-5} to 10^{10}.

We employed four metrics to quantitatively assess recognition performance using the ground truth segmentation as input. Those metrics were accuracy, precision, recall, and F-score with identical weighting (also known as F_1) metrics:

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \tag{12}$$

The results of this study are presented in Section IV-B.

2) Unsupervised segmentation and classification: This second validation study was completed using our entire pipeline, which combines the output of the unsupervised segmentation with the classification step. As in the previous study, we used a cross-validation process on the overall database with leave-one-out sessions, and averaged scores to present means and standard deviations as results. For both classification algorithms, recognition performance was assessed with respect to the feature transformation order and the inner parameter of the classifiers (i.e., number of neighbors for the $k-$NN and tradeoff penalty for the SVM). Moreover, the assessment of recognition performance was completed with the three dissimilarity metrics used for the segmentation process (Hausdorff, Fréchet and DTW).

We also quantified the recognition performance with the four previous assessment metrics (i.e., accuracy, precision, recall, and F_1 score). Additionally, we used another metric referred to *matching*. The latter provides a temporal matching ratio of surgemes (in percentage) between the ground truth annotations and output of our pipeline. The matching score is defined as:

$$Match(t_i, g_i) = \frac{|\cap(t_i, g_i)|}{length(g_i)} \tag{13}$$

where $\cap$ denotes the overlapping between the unsupervised segmented sequence t_i and its corresponding ground truth sequence g_i of surgeme labels, normalized by the length of the current sequence. The results of this study are presented in Section IV-C.

IV. Results

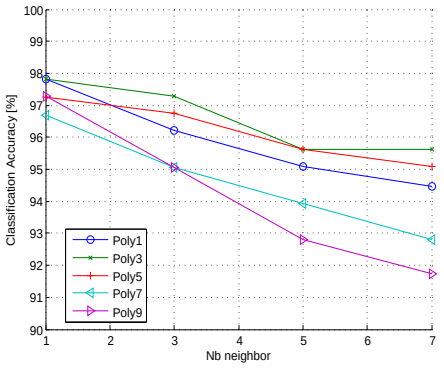
In this section, we report the results obtained from the studies presented in Section III. Experiments were run on an Intel Core i7-3770 @ 3.40GHz. The proposed pipeline performs dexeme segmentation and surgeme recognition from a trajectory in less than 5 seconds.

A. Segmentation Study Results

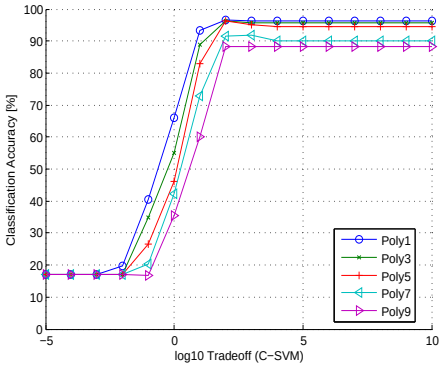
The qualitative assessment of the unsupervised segmentation method was applied to the first trajectory dataset. In the results, no difference in temporal segmentation (lower than 1.2% of the trajectory length) is noted between the three dissimilarity metrics. Fig. 6 shows results for the segmentation with the DTW dissimilarity metric only. Each point on the graphs is referenced as a delimiter between two segments. As shown, even if the letters do not have the same shape, the segmentation algorithm performed similarly on these three cases. The different parts of the letter are well identified. Only the vertical straight line is well distinguished with the operator N°3 (Fig. 6(a)), where in the two other cases, this vertical line is broke down in two parts because of the deformations. Otherwise, the semicircle is well split in two parts and the oblique line appeared as a single segment in all cases. Upon completion, the multiple-segment composition allows the reconstruction of the main parts of the shape so the operator's intentions can be recognized.

B. Validation Study Results: Ground Truth and Classification

The results reported herein were obtained with the first study described in Section III-D. We used the second trajectory dataset with the ground truth annotation combined with the classification step. In the $k-$NN case, Fig. 7(a) shows high recognition accuracies, especially for a 1-NN classification. An average recognition rate of 97.4% is obtained independently of the polynomial approximation order. However, 1-NN voting is sensitive to noise or misclassification, hence, the preference for the 3- or 5-NN classification methods is a better compromise with respect to our database length. Concerning

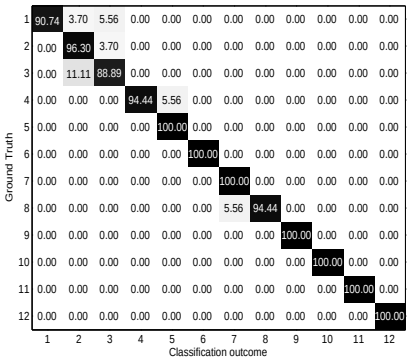


(a) $K-$NN classification accuracy depending on the number of neighbors.

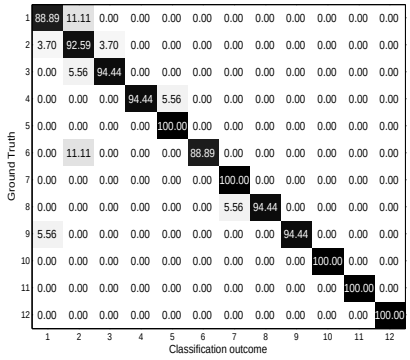


(b) SVM classification accuracy depending on the tradeoff penalty.

Fig. 7. Assessment of the classification step using $k-$NN and SVM classifiers applied to the ground truth consistency. Results are given with respect to both inner parameters and polynomial fitting order.



(a) 5-NN with *Poly5* approximation.



(b) SVM with *Poly5* approximation.

Fig. 8. Confusion matrices comparing the classification outcome with the ground truth annotation for each surgeme. Values indicate the percentage of accuracy at which the actual surgeme was recognized as belonging to each predicted surgeme.

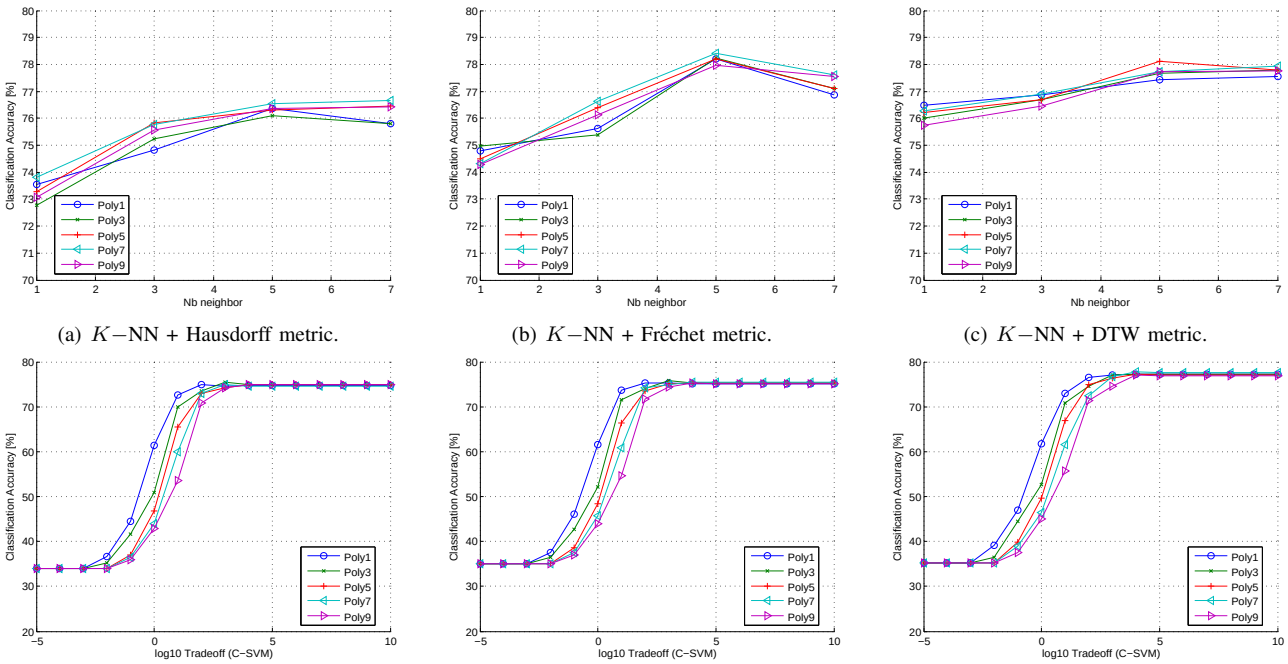
TABLE II
PRECISION, RECALL AND F_1 SCORES FOR EACH SURGEME WITH THE GROUND TRUTH ANNOTATION AND THE CLASSIFICATION PROCESS.

Surgeme	Method	Classifier	Precision [Mean ± SD %]	Recall [Mean ± SD %]	F1 [%]
Wait	Ground truth +	5-NN	100,00 ± 0,00	90,74 ± 14,10	95,15
	Classification	SVM	95,00 ± 10,00	88,89 ± 23,57	91,84
Reaching peg	Ground truth +	5-NN	88,89 ± 16,67	96,30 ± 11,11	92,44
	Classification	SVM	85,93 ± 22,47	92,59 ± 14,70	89,13
Precise positioning	Ground truth +	5-NN	89,81 ± 15,47	88,89 ± 22,05	89,35
	Classification	SVM	96,30 ± 11,11	94,44 ± 16,67	95,36
Grabbing peg	Ground truth +	5-NN	100,00 ± 0,00	94,44 ± 16,67	97,14
	Classification	SVM	100,00 ± 0,00	94,44 ± 16,67	97,14
Extracting peg	Ground truth +	5-NN	96,30 ± 11,11	100,00 ± 0,00	98,11
	Classification	SVM	96,30 ± 11,11	100,00 ± 0,00	98,11
Moving to target	Ground truth +	5-NN	100,00 ± 0,00	100,00 ± 0,00	100,00
	Classification	SVM	**88,89 ± 33,33**	**88,89 ± 33,33**	**88,89**
Positioning on target	Ground truth +	5-NN	96,30 ± 11,11	100,00 ± 0,00	98,11
	Classification	SVM	96,30 ± 11,11	100,00 ± 0,00	98,11
Inserting in target	Ground truth +	5-NN	100,00 ± 0,00	94,44 ± 16,67	97,14
	Classification	SVM	100,00 ± 0,00	94,44 ± 16,67	97,14
Releasing peg	Ground truth +	5-NN	100,00 ± 0,00	100,00 ± 0,00	100,00
	Classification	SVM	100,00 ± 0,00	94,44 ± 16,67	97,14
Moving to wait	Ground truth +	5-NN	100,00 ± 0,00	100,00 ± 0,00	100,00
	Classification	SVM	100,00 ± 0,00	100,00 ± 0,00	100,00
Moving back to center	Ground truth +	5-NN	100,00 ± 0,00	100,00 ± 0,00	100,00
	Classification	SVM	100,00 ± 0,00	100,00 ± 0,00	100,00
Moving to end position	Ground truth +	5-NN	100,00 ± 0,00	100,00 ± 0,00	100,00
	Classification	SVM	100,00 ± 0,00	100,00 ± 0,00	100,00
Average performance	Ground truth +	5-NN	97,61 ± 4,53	97,07 ± 6,72	97,29
	Classification	SVM	96,56 ± 8,26	95,68 ± 11,52	96,07

the SVM classification, the RBF kernel with a default value $\gamma = \frac{1}{Nb\ class}$ (the best compromise between performance and boundary complexity) provides 96.2% accuracy with the best parameter combination (feature approximation order + SVM tradeoff) as presented in Fig. 7(b). C-SVM with a tradeoff parameter effectively improves classification performance by deforming decision boundaries to accommodate a larger penalty for error/margin tradeoff. The performances are close to the $k-$NN results when $C \in [10^3, 10^{10}]$, which indicates strong misclassification penalization.

For both classifiers, the polynomial approximation order shows a noticeable impact on classifier performance. In the case of *Poly1*, *Poly3* and *Poly5*, high recognition accuracy is obtained. Conversely, when considering *Poly7* and *Poly9*, accuracy decreases as the polynomial order increases. Finally, *Poly3* and *Poly5* produce the best recognition scores for surgical gesture classification. In the following studies, we only focused on the *Poly5* approximation, because it allows high recognition results and accurate data generalization.

A detailed analysis of the recognition performance for each surgeme is presented in Table II. Results were averaged from all sessions, and means and standard deviations are presented. Moreover, the averaged confusion matrices with *Poly5* and both classifiers are presented in Fig. 8. They show the distribution of the classification outcome with respect to the manual annotation. In these results, we notice both high precision and recall scores for each surgeme with low standard deviations, except for the recognition of the surgeme *Moving to target* with the SVM classifier. Indeed, this surgeme was often confused with the *Reaching peg* surgeme, mainly because both



(a) $K-$NN + Hausdorff metric. (b) $K-$NN + Fréchet metric. (c) $K-$NN + DTW metric.

(d) SVM + Hausdorff metric. (e) SVM + Fréchet metric. (f) SVM + DTW metric.

Fig. 9. Assessment of the unsupervised segmentation and classification steps using $k-$NN and SVM classifiers with respect to their inner parameters. Effects of the polynomial fitting order and of the dissimilarity metric used for the segmentation are considered.

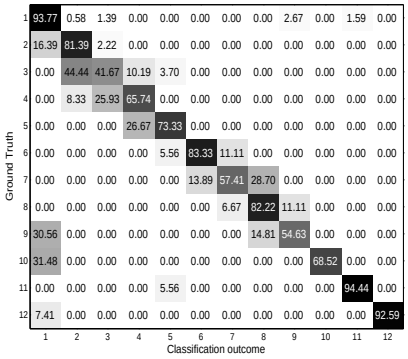
gestures involved the same straight line dexemes. These results highlight that the second part of our pipeline (i.e., feature transformation and classification process) is well-adapted to the surgeme recognition problematic in a training context.

C. Validation Study Results: Unsupervised Segmentation and Classification

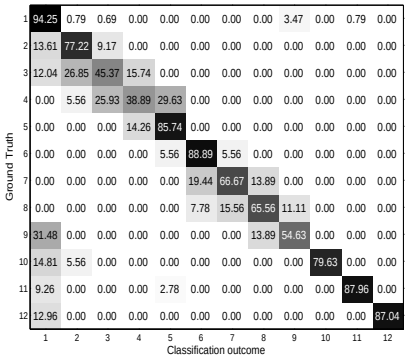
This section presents results obtained with the second study detailed in Section III-D. The second trajectory dataset was used with our overall recognition pipeline (i.e., both unsupervised segmentation and classification steps). First, the assessment of the various dissimilarity metrics used for the unsupervised segmentation is addressed. As shown in Fig. 9, classification accuracies are not highly impacted by dissimilarity metrics, independently of the considered classifier.

For the $k-$NN classifier, the best performances are obtained with a 5-neighbor voting. As in the previous section, the polynomial order does not significantly impact recognition accuracy. But as noted above, *Poly5* and *Poly7* provide the best results. All told, the best segmentation and recognition configurations are given by the DTW + *Poly5* and Fréchet + *Poly7* combinations, with 78.2% and 78.4% accuracy respectively. Compared to $k-$NN, SVM with the RBF kernel gives similar results when the C parameter strongly penalizes misclassification. Here, DTW exceeds the other metrics with a 77.5% recognition score and more stable results when modifying classifier parameters. Moreover, lower polynomial order approximations are preferred because they indicate that a good feature generalization is performed.

After comparing results for the ground truth and the proposed segmentation method using a *Poly5* approximation order, we conclude that the unsupervised method (with both

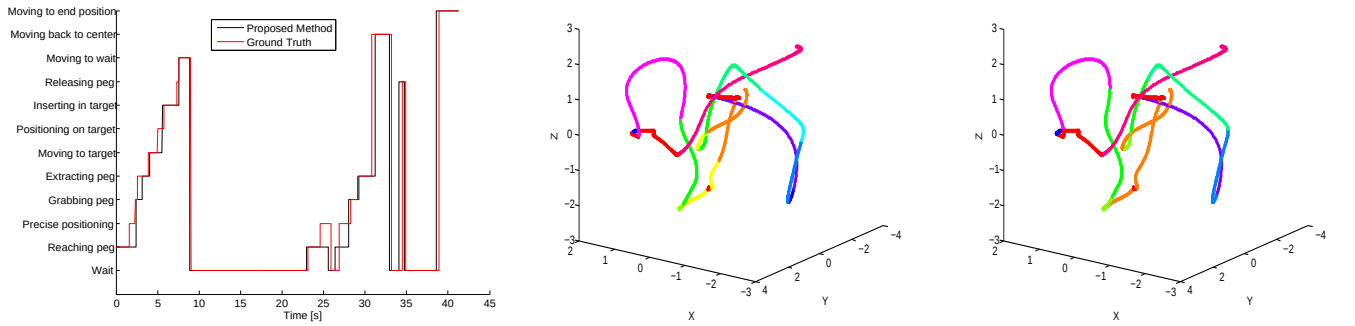


(a) 5-NN with *Poly5* and DTW metric.



(b) SVM with *Poly5* and DTW metric.

Fig. 10. Confusion matrices comparing automatic segmentation and classification outcome with ground truth annotation for each surgeme. Values indicate the percentage of accuracy at which the actual surgeme was recognized as belonging to each predicted surgeme.



(a) Surgeme timelines of the ground truth annotations and the recognized surgemes.

(b) Ground truth trajectory annotated.

(c) Trajectory with the unsupervised surgeme recognition.

Fig. 11. Example for one trajectory: surgeme timelines comparison of the ground truth annotations and the automatic segmentation and classification workflow. The 3D trajectories from the left hand are colored in order to distinguish each specific surgeme. The matching score for this example is 81.1%.

TABLE III
PRECISION, RECALL AND F_1 SCORES FOR EACH SURGEME WITH OUR UNSUPERVISED SEGMENTATION AND THE CLASSIFICATION PROCESS.

Surgeme	Method	Classifier	Precision [Mean ± SD %]	Recall [Mean ± SD %]	F1 [%]
Wait	Auto Seg. + Classification	5-NN	85,01 ± 6,25	93,77 ± 4,23	89,17
		SVM	83,72 ± 5,47	94,25 ± 6,64	88,67
Reaching peg	Auto Seg. + Classification	5-NN	**72,75 ± 18,73**	**81,39 ± 21,03**	**76,83**
		SVM	**75,93 ± 14,10**	**77,22 ± 24,89**	**76,57**
Precise positioning	Auto Seg. + Classification	5-NN	**31,48 ± 24,22**	**41,67 ± 39,53**	**35,86**
		SVM	**32,96 ± 19,82**	**45,37 ± 36,59**	**38,18**
Grabbing peg	Auto Seg. + Classification	5-NN	64,81 ± 18,99	65,74 ± 17,40	65,27
		SVM	56,30 ± 31,77	38,89 ± 19,98	46,00
Extracting peg	Auto Seg. + Classification	5-NN	**89,07 ± 16,81**	**73,33 ± 11,37**	**80,44**
		SVM	**72,04 ± 13,28**	**85,74 ± 13,95**	**78,29**
Moving to target	Auto Seg. + Classification	5-NN	**85,19 ± 22,74**	**83,33 ± 25,00**	**84,25**
		SVM	**82,59 ± 26,97**	**88,89 ± 22,05**	**85,63**
Positioning on target	Auto Seg. + Classification	5-NN	62,96 ± 41,48	57,41 ± 37,14	60,06
		SVM	61,30 ± 39,30	66,67 ± 41,46	63,87
Inserting in target	Auto Seg. + Classification	5-NN	**68,52 ± 24,22**	**82,22 ± 26,82**	**74,75**
		SVM	**62,96 ± 33,10**	**65,56 ± 36,09**	**64,23**
Releasing peg	Auto Seg. + Classification	5-NN	77,04 ± 28,06	54,63 ± 25,04	63,93
		SVM	74,07 ± 26,50	54,63 ± 13,89	62,88
Moving to wait	Auto Seg. + Classification	5-NN	**100,00 ± 0,00**	**68,52 ± 24,22**	**81,32**
		SVM	**100,00 ± 0,00**	**79,63 ± 24,69**	**88,66**
Moving back to center	Auto Seg. + Classification	5-NN	**92,59 ± 14,70**	**94,44 ± 16,67**	**93,51**
		SVM	**96,30 ± 11,11**	**87,96 ± 19,14**	**91,94**
Moving to end position	Auto Seg. + Classification	5-NN	**100,00 ± 0,00**	**92,59 ± 14,70**	**96,15**
		SVM	**100,00 ± 0,00**	**87,04 ± 20,03**	**93,07**
Average performance	Auto Seg. + Classification	5-NN	**77,45 ± 18,02**	**74,09 ± 21,93**	**75,13**
		SVM	**74,85 ± 18,45**	**72,65 ± 23,28**	**73,17**

classifiers) does not outperform the recognition capacity with the ground truth consistency. However, it provides relatively accurate results with respect to the dissimilarity metric used for the segmentation process. For automatic segmentation purposes, DTW provides better segmentation output than the other metrics, especially when it is combined with the *Poly5* parameter. They offer the best compromise between feature generalization and recognition accuracy. This combination allowed us to achieve 78.2% recognition for the 5-NN classifier and 77.1% for the SVM.

We also carried out a performance analysis of each surgeme. Only the results using the best combinations are presented here. Both 5-NN and SVM with a tradeoff penalty $C = 10^3$ were tested with the *Poly5* approximation order and the DTW dissimilarity metric for segmentation. Confusion matrices (see Fig. 10), as well as precision, recall and F_1 have been computed and averaged (see Table III).

From this assessment, we note that the 5-NN classifier offers the best compromise for the current application, and especially for the most important surgemes, such as *Extracting peg*, *Moving to target* and *Inserting in target*. The confusion matrices presented in Fig. 10 also show high recall percentages for most surgemes. While average performances in Table III are around 20% lower than the performances based on ground truth consistency, average recall reaches about 74% and average precision goes up to 77%.

The last assessment focuses on the timeline comparison between manual segmentation and the proposed processing outputs (see Fig. 11). The 5-NN classifier reaches an average matching score of 81.9% (± 2.4%) with the *Poly5* and DTW dissimilarity metric. An instance of the results is presented in Fig. 11(a), and the corresponding 3D trajectories with the ground truth annotation (Fig. 11(b)) could be compared to the processed one with our overall pipeline (Fig. 11(c)). As illustrated, trajectories are similarly colored and the transition delay between surgemes is visible in 3D. These figures mainly illustrate that most errors are due to misidentification of transition times between surgemes, rather than incorrect task classification. They also indicate that the proposed workflow is capable of recognizing the intentions of a new operator within a very small detection delay.

V. DISCUSSION

A. Kinematic Channels Impact

Kinematic data offers relevant information for low-level recognition of human gestures. However, recognition performance is largely affected by the kinematic data that are used as pipeline input. In a prior step, we performed the same study relying only on 3D positions. However, solely leveraging 3D positions as input channels led to poor segmentation and classification performance. In the same way, Cifuentes et al. [41] used only quaternion information to retrieve operator gestures (i.e., intention could be correlated to orientation as well). Based on this analysis, we concluded that the operators intentions could be determined using rigid transformation data. Gao et al. [42] advocated this approach and presented surgical gesture recognition works which dealt with 78 kinematic variables. However, these studies relied mainly on dimensionality reduction to characterize surgical gestures, and failed

to provide information about the relevance of the channels considered for processing. In our work, we used 24 kinematic channels, including 3D invariant variables, to represent tool motions. With this shorter signature length, we were able to capture and recognize surgical gestures, and also improve the processing steps by reducing the computation time of the algorithm. A short-term perspective of this approach is to compare the different Euclidean descriptors present in the literature, including the ones in the PCL library, in order to optimize the segmentation process as well as the recognition performance.

B. Persistent-Extrema Selection Impact

Relying on the topological simplification of signals, and ranking attributes by their persistence measure, the persistent-extrema selection is a crucial element of the proposed pipeline. An empirical tuning of the persistence threshold was performed so that the same parameters would be used in all experiments. However, data-driven machine learning techniques could help to estimate the best threshold for a dataset, by optimizing this value with respect to the classification performance. Based on our observations, this threshold value affects the minimal size of the dexemes in a non-linear way and generates high variations on the recognition performance.

C. Dissimilarity Metric Impact

In this paper, Hausdorff and Fréchet metrics were finally excluded because of their lower performance, especially for classification purposes. DTW enables the best association (segmentation + classification), and had already been identified as a superior metric [43, 44]. However, the DTW algorithm has two main drawbacks. The first one is its long computation time that could, however, be reduced with predefined constraints. The second shortcoming is that the DTW requires that particular attention be paid to data dimensionality. Ten Holt et al. [45] addressed the problem of gesture recognition by using a multi-dimensional DTW (MD-DTW). Their results confirmed that MD-DTW outperformed 1D-DTW in case of normalized input signals, motivating our pre-processing step for normalization. Moreover, issues related to noisy measures do not have to be considered in our work, thanks to the low-pass filtering step for fundamental motion retrieving. Nonetheless, future work could focus on the comparison of 1D-DTW and MD-DTW, by quantifying the classification performance variations.

D. Feature Transformation and Approximation Order Impacts

In this work, we focused on computing a relevant descriptive signature with potential fast backward computing for further robotic assistance. Polynomial approximation provides a discriminant signature and yielded excellent results for spatio-temporal trajectory classification [37]. Motivating our choice, we discussed the approximation order for feature transformation. As suggested in the results, this approximation order impacts the performance of the proposed pipeline. However, one can argue that a high approximation order allows for well-fitted data and reduces information loss. Nevertheless, the general idea is to understand human gestures from a training database. Since there are multiple ways to perform a single action, learned motions must be generalized as much as possible to accurately capture and understand human gestures from an unknown training dataset. In our case, a 5-degree polynomial approximation provided optimal results in order to avoid overfitting side-effects. The resulting signatures, which comprised 72 variables (6 coefficients and 12 signals for one surgical tool), were sufficient for offline generalization but also not too large for further online processing.

E. Surgeme Recognition Assessment

In the last validation study, we found that the $5-$NN classifier offered the best compromise for the present application. With this classifier, the most important surgemes such as *Extracting peg*, *Moving to target* and *Inserting in target* were appropriately recognized. The confusion matrices presented in Fig. 10 showed high recall percentages for these surgemes. Conversely, the *Precise positioning*, *Grabbing peg*, *Positioning on target* and *Releasing peg* surgemes generated poorer results and high standard deviations, partly due to their short durations and their specific features, which are not easily reproduced (e.g., operators did not use the same positioning process for each session). We also noticed that the *Wait* surgeme caused unwanted gestures. It was often confused with other surgemes because, even during the waiting phase, small motions were produced as a result of hand mimicking. By using such bottom-up approach, we addressed the lowest decomposition level of gesture where different surgemes reveal similar dexemes that are hard to distinguish from each other, resulting in misclassification. Nevertheless, the final matching assessment between ground truth annotations and the proposed method shows that an accurate recognition of the most important surgemes involved in the pick-and-place task is possible. Moreover, an extension of this unsupervised segmentation method to other training tasks could be directly performed since the proposed algorithm does not rely on any prior information about the surgical task.

VI. Conclusion

Surgical gesture recognition is a key component for the next generation of surgical training systems, including context-aware computer-assisted systems. It could offer an advanced quantitative evaluation of surgical gestures for more appropriate operator feedback (e.g., haptic or visual feedback, path-following capability). In this paper, we proposed a new approach for automatically segmenting and recognizing surgical gestures during robotic training sessions. While the first step performs trajectory decomposition into dexemes, the second step recognizes these dexemes to assign them to surgical gestures. We believe that accurately detecting, and then understanding, the surgical gestures of a new trainee without any human intervention in the training process are realistically attainable goals with this approach. We assessed our algorithm on a real training dataset from surgical experts. An accuracy of 97.4% was achieved in the learning task and an average matching score of 81.9% was obtained for a fully automated

gesture recognition process. To conclude, this work proposes a new approach for automated surgical gesture recognition. It directly encourages the educational heuristic "perfect practice makes perfect", by providing an implement to improve training efficiency with potential surgical skill assessment and gesture-specific feedback.

REFERENCES

[1] K. Ahmed *et al.*, "Observational tools for assessment of procedural skills: A systematic review," *The American Journal of Surgery*, vol. 202, no. 4, pp. 469–480, 2011.

[2] G. S. Guthart and K. Salisbury, "The Intuitive Telesurgery System: Overview and Application," *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 618–621, 2000.

[3] C. Freschi *et al.*, "Technical review of the da Vinci surgical telemanipulator," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 9, no. 4, pp. 396–406, 2013.

[4] S. Tsuda *et al.*, "Surgical Skills Training and Simulation," *Current problems in surgery*, vol. 46, no. 4, pp. 271–370, 2009.

[5] W. M. Brinkman *et al.*, "Da Vinci Skills Simulator for Assessing Learning Curve and Criterion-based Training of Robotic Basic Skills," *Urology*, vol. 81, no. 3, pp. 562–566, 2013.

[6] R. McCormick, "Conceptual and Procedural Knowledge," *International Journal of Technology and Design Education*, vol. 7, no. 1, pp. 141–159, 1997.

[7] F. Lalys and P. Jannin, "Surgical process modelling: A review," *International Journal of Computer Assisted Radiology and Surgery*, vol. 9, pp. 495–511, 2014.

[8] L. Riffaud *et al.*, "Recording of Surgical Processes: A Study Comparing Senior and Junior Neurosurgeons During Lumbar Disc Herniation Surgery," *Neurosurgery*, vol. 67, no. 2, pp. 325–332, 2010.

[9] G. Forestier *et al.*, "Classification of surgical processes using dynamic time warping," *Journal of Biomedical Informatics*, vol. 45, no. 2, pp. 255–264, 2012.

[10] J. K. Aggarwal and L. Xia, "Human activity recognition from 3D data: A review," *Pattern Recognition Letters*, vol. 48, pp. 70–80, 2014.

[11] T. Sugino, H. Kawahira, and R. Nakamura, "Surgical task analysis of simulated laparoscopic cholecystectomy with a navigation system," *International Journal of Computer Assisted Radiology and Surgery*, vol. 9, pp. 825–836, 2014.

[12] C. E. Reiley *et al.*, "Automatic Recognition of Surgical Motions Using Statistical Modeling for Capturing Variability," *Studies in health technology and informatics*, vol. 132, no. 1, pp. 396–401, 2008.

[13] B. Varadarajan *et al.*, "Data-Derived Models for Segmentation with Application to Surgical Assessment and Training," *Medical Image Computing and Computer-Assisted Intervention*, vol. 5761, no. 1, pp. 426–434, 2009.

[14] N. Padoy and G. D. Hager, "Human-Machine Collaborative Surgery Using Learned Models," *IEEE International Conference on Robotics and Automation*, pp. 5285–5292, 2011.

[15] N. Ahmidi *et al.*, "String Motif-Based Description of Tool Motion for Detecting Skill and Gestures in Robotic Surgery," *Medical Image Computing and Computer-Assisted Intervention*, vol. 8149, pp. 26–33, 2013.

[16] L. Tao *et al.*, "Surgical Gesture Segmentation and Recognition," *Medical Image Computing and Computer-Assisted Intervention*, vol. 8151, pp. 339–346, 2013.

[17] L. Zappella *et al.*, "Surgical gesture classification from video and kinematic data," *Medical Image Analysis*, vol. 17, no. 7, pp. 732–745, 2013.

[18] C. Lea, G. D. Hager, and R. Vidal, "An Improved Model for Segmentation and Recognition of Fine-grained Activities with Application to Surgical Training Tasks," *IEEE Winter Conference on Applications of Computer Vision*, pp. 1–7, 2015.

[19] S. Schulz and A. Woerner, "Automatic Motion Segmentation for Human Motion Synthesis," *International Conference on Articulated Motion and Deformable Objects*, vol. 6169, pp. 182–191, 2010.

[20] D. Popa *et al.*, "Trajectory Based Hand Gesture Recognition," *International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics*, pp. 115–120, 2007.

[21] J. F.-S. Lin and D. Kulić, "Online Segmentation of Human Motion for Automated Rehabilitation Exercise Analysis," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 1, pp. 2881–2884, 2014.

[22] W. W. Kong and S. Ranganath, "Automatic Hand Trajectory Segmentation and Phoneme Transcription for Sign Language," *IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 1–6, 2008.

[23] M. S. Holden *et al.*, "Feasibility of Real-Time Workflow Segmentation for Tracked Needle Interventions," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 6, pp. 1720–1728, 2014.

[24] H. C. Lin *et al.*, "Towards automatic skill evaluation: Detection and segmentation of robot-assisted surgical motions," *Computer Aided Surgery*, vol. 11, no. 5, pp. 220–230, 2006.

[25] C. E. Reiley and G. D. Hager, "Task versus Subtask Surgical Skill Evaluation of Robotic Minimally Invasive Surgery," *Medical Image Computing and Computer-Assisted Intervention*, vol. 5761, pp. 435–442, 2009.

[26] E. Calabi *et al.*, "Differential and Numerically Invariant Signature Curves Applied to Object Recognition," *International Journal of Computer Vision*, vol. 26, no. 2, pp. 107–135, 1998.

[27] M. Boutin, "Numerically Invariant Signature Curves," *International Journal of Computer Vision*, vol. 40, no. 3, pp. 235–248, 2000.

[28] S. D. Wu and Y. F. Li, "Flexible signature descriptions for adaptive motion trajectory representation, perception and recognition," *Pattern Recognition*, vol. 42, pp. 194–214, 2009.

[29] S. Yang *et al.*, "Performance of a 6-Degree-of-Freedom Active Microsurgical Manipulator in Handheld Tasks," *IEEE International Conference on Engineering in Medicine and Biology Society*, pp. 5670–5673, 2013.

[30] I. D. Loram, P. J. Gawthrop, and M. Lakie, "The frequency of human, manual adjustments in balancing an inverted pendulum is constrained by intrinsic physiological factors." *The Journal of physiology*, vol. 577, pp. 417–432, 2006.

[31] T. Weinkauf, Y. Gingold, and O. Sorkine, "Topology-based Smoothing of 2D Scalar Fields with C1-Continuity," *Eurographics Conference on Visualization*, vol. 29, no. 3, pp. 1221–1230, 2010.

[32] H. Edelsbrunner, D. Letscher, and A. Zomorodian, "Topological Persistence and Simplification," *Discrete & Computational Geometry*, vol. 28, no. 4, pp. 511–533, 2002.

[33] J. Chen *et al.*, "Clustering of Trajectories Based on Hausdorff Distance," *International Conference on Electronics, Communications and Control*, pp. 1940–1944, 2011.

[34] K. Buchin *et al.*, "Detecting Commuting Patterns by Clustering Subtrajectories," *Algorithms and Computation*, vol. 5369, no. 642, pp. 644–655, 2008.

[35] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[36] A. Neubeck and L. V. Gool, "Efficient Non-Maximum Suppression," *International Conference on Pattern Recognition*, vol. 3, pp. 850–855, 2006.

[37] A. Naftel and S. Khalid, "Classification and Prediction of Motion Trajectories using Spatiotemporal Approximations," *Annual German Conference on Advanced in Artificial Intelligence*, 2009.

[38] K. P. Bennett and C. Campbell, "Support Vector Machines: Hype or Hallelujah?" pp. 1–13, 2000.

[39] B. Hannaford *et al.*, "Raven-II: An Open Platform for Surgical Robotics Research," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 4, pp. 954–959, 2013.

[40] A. M. Derossis *et al.*, "Development of a Model for Training and Evaluation of Laparoscopic Skills," *The American Journal of Surgery*, vol. 175, no. 6, pp. 482–487, 1998.

[41] J. Cifuentes *et al.*, "An Arc-length Warping Algorithm for Gesture Recognition Using Quaternion Representation," *IEEE International Conference on Engineering in Medicine and Biology Society*, pp. 6248–6251, 2013.

[42] Y. Gao *et al.*, "JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS): A Surgical Activity Dataset for Human Motion Modeling," *Modeling and Monitoring of Computer Assisted Interventions*, pp. 1–10, 2014.

[43] B. Morris and M. Trivedi, "Learning Trajectory Patterns by Clustering: Experimental Studies and Comparative Evaluation," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 312–319, 2009.

[44] X. Wang *et al.*, "Experimental comparison of representation methods and distance measures for time series data," *Data Mining and Knowledge Discovery*, vol. 26, no. 2, pp. 275–309, 2013.

[45] G. A. Ten Holt, M. J. T. Reinders, and E. A. Hendriks, "Multi-Dimensional Dynamic Time Warping for Gesture Recognition," *Annual Conference of the Advanced School for Computing and Imaging*, 2007.