POLITECNICO DI MILANO

Master's Program in Biomedical Engineering

Department of Electronics, Information and Bioengineering

A Method for Simultaneous Task Recognition and Gesture Classification in Surgical Robotics

NEARlab

Neuroengineering and Medical Robotics Laboratory

Supervisor: Prof. Elena DE MOMI

Assisitant Supervisors: PhD. Nima ENAYATI,
MSc. Aleks ATTANASIO

By:

Francesco GRIGOLI, 864846

Accademic Year 2016-2017

Numerose persone hanno contribuito in termini morali e tecnici al mio percorso di studi culminati con la stesura di questa tesi di laurea magistrale. Inanzitutto vorrei ringraziare la mia famiglia per l'incessante supporto ed affetto che mi hanno saputo dare, in ogni situazione ed in ogni circostanza. Vorrei ringraziare Valentina, che ha sempre appoggiato in maniera critica e propositiva ogni mia scelta facendomi crescere umanamente e professionalmente fornendomi consiglio e conforto. Vorrei ringraziare sentitamente Aleks, senza il cui instancabile contributo tecnico nonché umano questo progetto non sarebbe mai potuto essere portato avanti. Voglio esprimere la mia gratitudine alla Professoressa Elena De Momi e al Dottor Nima Enayati per avermi dato la possibilità di effettuare questo studio a compimento della mia carriera universitaria. Infine vorrei ringraziare i miei amici che ormai da anni mi conoscono e che non hanno mai smesso di sopportarmi ed esortarmi, accetandomi anche nei momenti di difficoltà.

Grazie di cuore a tutti voi, Francesco.

After all this time, to Christine

Sommario

Introduzione

Fin dagli albori della chirurgia mini invasiva (Minimally Invasive Surgery, MIS) furono evidenti sia i suoi numerosi vantaggi rispetto alla chirurgia tradizionale sia i suoi limiti. Infatti, se da un lato la MIS giocava un ruolo chiave nel ridurre i tempi di degenza ospedaliera e le possibili complicazioni legate ad interventi complessi [1], dall'altro, si poteva applicare solo ad un ristretto numero di operazioni chirurgiche a causa della postura chirurgica necessaria e degli strumenti usati [2]. Questi limiti tecnici e tecnologici spinsero il campo della robotica a sviluppare sistemi alternativi che permettessero il pieno sfruttamento delle potenzialità della chirurgia mini invasiva, dando vita alla chirurgia mini invasiva assistita da robot (Robot-Assisted Minimally Invasive Surgery, RAMIS). Al giorno d'oggi sempre più operazioni chirurgiche vengono eseguite attraverso l'uso di protocolli RAMIS, sfruttando l'impiego del popolare robot da Vinci Surgical System (dVSS), prodotto da Intuive Surgical Inc., o di altri sistemi analoghi. Nonostante gli innumerevoli vantaggi portati da questi nuovi protocolli, poichè in RAMIS i robot vengono manipolati in maniera telematica, le operazioni eseguite risultano essere di lunga durata e faticose [3, 4]. Per rimediare a questi svantaggi, i robot hanno iniziato ad essere provvisti di algoritmi di apprendimento automatico grazie ad i quali riescono ad eseguire autonomamente operazioni elementari [5] o a facilitare l'acquisizione delle abilità motorie necessarie ai chirurghi nelle varie operazioni [6]. Infatti, I robot intelligenti possono valutare in modo oggettivo le esercitazioni degli studenti di chirurgia indirizzando e consigliando protocolli di addestramento specifici. Inoltre, possono riconoscere le varie procedure chirurgiche intervenendo in aiuto del chirurgo eseguendo in autonomia operazioni elementari, riducendone la durata dell'esecuzione e la fatica connessa.

Questa tesi vuole presentare un metodo di apprendimento automatico che perme-

tta al robot il riconoscimento simultaneo del tipo di operazione chirurgica eseguita dagli utenti, caratterizzando pienamente ogni movimento effettuato. Per questo scopo sono stati studiati diversi approcci: il primo metodo esplorato cerca di caratterizzare direttamente ed in tempo reale ogni azione eseguita in operazioni chirurgiche senza considerare il particolare tipo di operazione effettuata. Il secondo approccio adottato esegue l'analisi dei gesti chirurgici in tempo reale ma, al contrario, in funzione del tipo specifico di operazione chirurgica effettuata.

Materiali e metodi

I dati chirurgici necessari per la creazione dei metodi sopracitati sono stati ottenuti grazie all'interfaccia API [7] creata per l'acquisizione di dati dal dVSS. Nello specifico questi dati provengono da due manipolatori lato paziente (Patient Side Manipulators, PSMs) composti da braccia seriali a 7 gradi di libertà ciascuna [8]. I dati acquisiti sono relativi all'esecuzione di 3 diverse procedure chirurgiche quali: sutura, passaggio di ago ed esecuzione di un nodo, eseguite per 5 volte da 8 chirurghi differenti. Questi dati cinematici relativi alla posizione dell'estremità dei due PSMs, dalla loro rotazione e velocità lineare insieme con le informazioni angolari riguardanti le pinze, sono raccolti all'interno del set di dati chiamato *JHU-ISI Gesture and Skill Assessment Working Set* (JIGSAWS) [9]. È importante notare che i dati cinematici all'interno del JIGSAWS sono dotati di indicazioni manuali dei gesti che i dati stessi rappresentano. Questi gesti vengono definiti come azioni atomiche singole, indicatrici della particolare procedura chirurgica eseguita [10, 11, 12] e verranno utilizzati per l'addestramento dei modelli usati.

Al fine di addestrare e di valutare i modelli che permettono il riconoscimento della particolare operazione chirurgica e dei gesti contenuti in essa, il set di dati JIGSAWS è stato suddiviso in due diverse configurazioni di cross-validazione. Nella prima, la sessione d'acquisizione i-esima eseguita da ogni chirurgo viene utilizzata per la valutazione degli algoritmi. Nella seconda configurazione di cross-validazione, tutte le sessioni d'acquisizione relative ad un determinato chirurgo, sono a turno escluse

dall'addestramento ed utilizzate per la valutazione degli algoritmi [13]. Usando queste configurazioni di dati differenti, lo scopo della tesi viene raggiunto attraverso lo studio di due diversi approcci. In particolare:

- Il primo metodo proposto va ad individuare e a caratterizzare ogni gesto di un'operazione chirurgica senza considerare di che tipo di operazione si tratti. Al fine di perseguire questo scopo è stato progettato un classificatore di gesti generico (Generic Gesture Classifier GGC), composto da Modelli di Marcov *Hidden* (Hidden Markov Moldes, HMMs) uniti per poter identificare i gesti in ogni possibile operazione chirurgica.

- Il secondo metodo studiato effettua prima di tutto un riconoscimento in tempo reale del tipo di operazione chirurgica eseguita e, successivamente, indirizza la caratterizzazione dei gesti in funzione dell'operazione individuata. Il metodo risulta composto da due algoritmi indipendenti: il primo effettua il riconoscimento dell'operazione (Task-Related Task Recognizer, TRTR) utilizzando tre HMMs a tre stati ciascuno e definiti per il riconoscimento specifico di ogni operazione chirurgica. Il secondo algoritmo, il classificatore di gesti (Task-Specific Gesture Classifier, TSGC) è suddiviso in tre parti ed ognuna è composta da HMMs uniti per poter identificare i gesti di una specifica operazione chirurgica.

Risultati e discussione

I metodi proposti sono in primo luogo ottimizzati per fare in modo che risultino robusti ed affidabili. Successivamente considerando la prestazione temporale degli algoritmi così come la loro accuratezza nella classificazione è stato possibile valutarli e confrontarli con lo stato dell'arte.

Considerando lo scopo per cui l'algoritmo GGC è stato creato, è possibile affermare che esso risulta capace di effettuare la caratterizzazione dei gesti di una generica operazione chirurgica in tempo reale e senza nessuna informazione a priori riguardo al tipo di procedura eseguita. Anche se a differenza dei valori in letteratura questi

risultati sono stati raggiunti in tempo reale, è necessario sottolineare che l'accuratezza finale raggiunta nella classificazione dei gesti risulta essere troppo bassa per l'utilizzo immediato dell'algoritmo in possibili applicazioni.

Il secondo approccio presentato è composto da due algoritmi diversi: il riconoscitore di operazioni chirurgiche TRTR ed il classificatore di gesti TSGC. È possibile affermare, considerando i risultati ottenuti, che il TRTR si è dimostrato più accurato nel riconoscimento delle operazioni chirurgiche degli algoritmi proposti in letteratura. Esso, inoltre si è dimostrato esssere più veloce, infatti è capace di processare in tempo reale i dati acquisiti dal dVSS, ottenendo, già con il 12% dell'operazione completata, il pieno riconoscimento della procedura chirurgica in esecuzione. Riguardo al TSGC, è possibile affermare che l'algoritmo, a differenza di quelli presentati in letteratura, è capace di processare in tempo reale i dati provenienti dal dVSS. Il riconoscimento dei gesti eseguiti avviene con un'accuratezza finale simile, anche se di poco inferire, a quella riportata dallo stato dell'arte.

Summary

Introduction

Since the very beginning of Minimally Invasive Surgery (MIS), its advantages with respect to the traditional open surgery as well as its limitations appeared evident. If, from one side, MIS played a key role in reducing disabilities and hospital stays enhancing life expectancy [1], on the other, the used tools and the particular surgeon required skills limited the positive effects of MIS on a restrict subset of surgical operations [2]. These technological limitations inspired the robotic branch to develop alternative systems exploiting completely MIS, giving birth to the Robotic-Assisted Minimally Invasive Surgery (RAMIS). More and more clinical operations are being performed every day in RAMIS, using the popular robot da Vinci Surgical System (dVSS) from Intuitive Surgical Inc. or analogous setups. However, since RAMIS involved tele-operated robotic assistants, surgical operations results to be tedious and time consuming [3, 4]. To face these drawbacks, surgical robots are starting to implement Machine Learning strategies to actively assist surgeons in improving their expertise level [5] or in automatizing time-demanding elementary operations[6]. Indeed, intelligent robots can objectively evaluate trainee surgeons making more profitable their training curricula, and, at the same time, intelligent robotic surgical assistants can autonomously perform elementary surgical tasks reducing fatigue and execution time.

By implementing a Machine Learning approach, this dissertation wants to present a framework that allows robots to perform simultaneous recognition of different surgical tasks characterizing them by capturing their underlying surgical motion. In particular: a first method tried to characterize every action in surgical trials in real time, without any consideration on the particular kind of operation. A different framework has been subsequently adopted to address real time action recognition as a function of the specific surgical task.

Materials and methods

The surgical data used to setup the aforementioned frameworks are provided by the dVSS and collected thanks to the Human Robot Interface *API* [7] from 2 Patient Side Manipulators (PSMs), robotic serial arms with 7-degree-of-freedom each [8]. Data are acquired while 8 surgeons, with a various expertise level, were repeating 3 very common surgical tasks as suturing, needle passing and knot tying, 5 times each. Kinematic records from the two PSMs were stored in the JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS) [9]. This dataset, not only contains kinematic data such as tool tip positions, rotation matrices, gripper angular velocity, tool tip linear and rotational velocities but also it incorporates manually annotated labels of ground-truth atomic gestures used to train the models. In particular, gestures are defined as meaningful single motion sequences used to deeply characterize each surgical task [10, 11, 12].

The JIGSAWS dataset is divided into two different cross-validation setups used to train and to evaluate the proposed approaches. In the first one the i-th session of each surgeon is used as a test set, while, in the second validation scheme, all sessions from each surgeon in turn are taken as test set [13]. By using these setups, three different Machine Learning models are designed and trained to accomplish the goal of the thesis. In particular:

- A first framework characterizes every gesture in surgical trials without any consideration on the particular kind of operation. This aim is accomplished by a designed Generic Gesture Classifier (GGC) based on Hidden Markov Models (HMMs) [14] linked together, following the gestures in every possible surgical tasks.

- The second studied framework addresses a task-specific gesture classification thanks a first task recognizer. Thus, the method is composed of two disjoint algorithms: a Task-Related Task Recognizer (TRTR) that uses a three state left-to-right HMM to model and classify each surgical task and a Task-Specific Gesture Classifier (TSGC), based on HMMs linked together, following the gestures in one task, able to perform the final gesture classification.

Results and Conclusions

The proposed algorithms are firstly optimized in order to setup models as robust as possible. Subsequently, by considering the time performances and the classification accuracies, algorithms are evaluated and compared with the actual state of art.

Considering the final purpose of the GGC algorithm, this approach results to be able to accomplish real time gesture classification without any prior assumption over the undergoing task. It is necessary to report that even if the GGC time performances are interesting, and not achieved in literature yet, the reached final classification accuracy results too low to allow possible immediate applications.

The second framework previously presented, is composed of two different algorithm: the task recognizer TRTR and the gesture classifier TSGC. It is possible to remark that the TRTR proved to be more accurate than the state of art reaching a real time stable task recognition with less then the 12% of the task accomplished, outperforming the literature. About the TSGC, with respect to the literature, it is able to work in real time reaching good gesture classification accuracies similar, but not as high as the ones reported to be the state of art.

Contents

Chapter 1

INTRODUCTION

As well as antibiotics and sterilization, one of the most important innovation that enhanced the role of medicine in improving general wellness is the introduction of *Minimally Invasive Surgery* (MIS) in 1987. Surgical procedures started to apply new protocols that have been able to reduce the invasiveness of surgery resulting in higher rate of success in operations (see Figure 1.1). Moreover, MIS played a key role: pain and disability were finally reduced, patients expectancy of life after operations increased and social costs related to hospitals were finally controlled [15].
Unlike the classic *open* surgery, MIS is performed by using laparoscopic tools that enter inside patients body through small incisions of about 1 centimetre. Surgeons can control these instruments having a view of the internal operating area thanks to an endoscopic camera.

The aforementioned set up has signed a real revolution in the surgery field allowing, for the first time in history, the entry of precision instruments in operation rooms. Even if this kind of surgery has become a gold standard its technical limitations appeared clear [2]:

- Instead of using hands, as in open surgery, surgeons have to use laparoscopic instruments reducing their degrees of freedom in the patient body from 7 per arm to only 4.

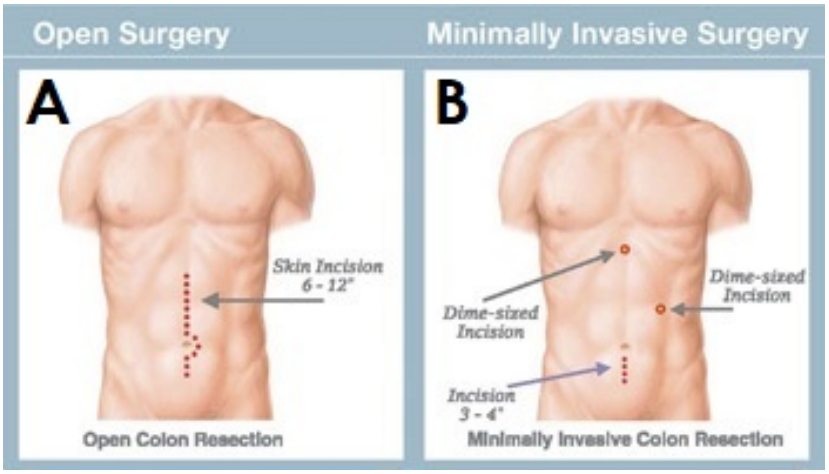

Figure 1.1: *Comparison between incisions necessary to perform colon resection: A) Open surgery. B) MIS. Image form www.houstoncolon.com.*

- Surgeons trembling are rigidly transmitted and amplified through laparoscopic tools up to the patient side.

- Instruments are controlled by surgeons while watching a 2-dimensional screen right in front of them or, most likely on their side. This unnatural operative posture causes fatigue to surgeons making their performances even more challenging. Figure 1.2 represents a typical posture held by surgeons during MIS procedures.

- Finally, the unnatural operative posture leads to work in a counterintuitive way: in order to move tools towards a specific target, surgeons should turn them in the opposite direction. Indeed, this *fulcrum effect* created by the trocar, the containing instrument for surgical tools in incision, in the insertion point, compromises the hand-eye coordination.

These technological limitations inspired research and development in robotics leading to alternative systems able to exploit completely MIS overcoming these important disadvantages.

Figure 1.2: *Maintained posture and used tools during MIS. Image from www.muhc.ca*

1.1 Robot Assisted Minimally Invasive Surgery

Robotics entered in MIS in 1994 with AESOP (voice controlled camera holder). This has been the first robot approved by the United States Food and Drugs Administration (FDA) [16]. Following this first attempt to overpass MIS limitations many other systems have been created, however only in 1997 with the da Vinci Surgical System (dVSS) prototype (Intuitive Surgical Inc.) the technology changes decisively.

By using the concept of telesurgery, first developed for military aims, the dVSS allows surgeons to perform surgical procedures from remote. Surgeons control dexterous manipulators (master), through which the motion is transmitted to a second ones (slave) which perform the surgery in another workspace (for ex. see Figure 1.3).

Thanks to this revolutionary setting the surgeon sits in with an ergonomic position in front of a 3D stereo viewer which shows the scene inside the patient. The aforementioned fulcrum effect is autonomously compensated recovering hand-eye coordination, while, 7 degree of freedom tools enlarge the surgeon's range of motion up to the natural condition enhancing dexterity. In addition to this, appropriate hardware and software filters compensate involuntary movement securing the patient and increasing overall precision.

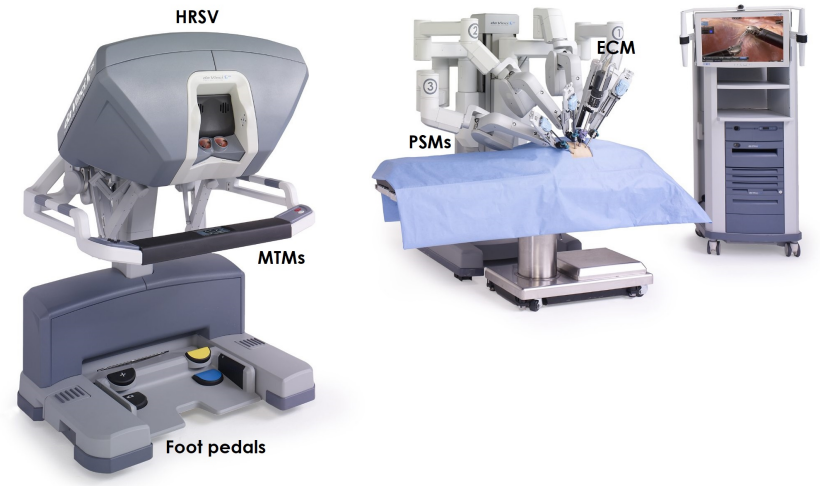

Figure 1.3: *The dVSS robot with all its components: master tool manipulators (MTMs), High-Resolution Stereo Viewer (HRSV), foot pedals, patient-side manipulators (PSMs) and Endoscopic CCD-Camera Manipulator(ECM) [17]*

1.1.1 Open issues

Thanks to its incredible performances more and more clinical operations are being performed every day using dVSS or equivalent systems. Robot-Assisted Minimally Invasive Surgery (RAMIS) has become a standard approach in many surgical fields and the number of robotic systems installed worldwide increases every year (rate of new installations by year represented in Figure 1.4).

Even if RAMIS has already defined a significant step forward in surgery, presenting many advantages with respect to both open surgery and MIS, developments can still be made to make it less demanding than open procedures. Actually, many operations still present significant manipulation complications: for instance, to tie a knot in laparoscopy (robotic or not) can take more than 3 minutes while in open surgery it takes less than 2 seconds [4], almost 200 times less. Even if these problems seem to be irrelevant, they open important issues:

- Since the technological aspect of the machine as well as the tools manipulation is complex, the training of new surgeons is longer, more expensive and problematic to be assessed.

- Surgical operations increase in difficulty, thus surgeons take more time to accom-

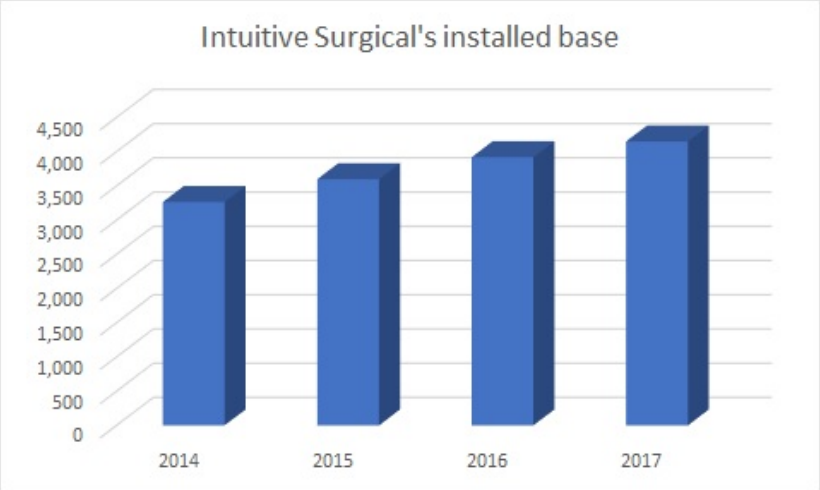

Figure 1.4: *Number of installed robot bases by years, from Intuitive Surgical. Source company reports www.seekingalpha.com*

plish their task forcing more aggressive strategies to anaesthetize patients.

- Costs for RAMIS are incredibility high, between 10,269 CAD$ to 26,608 CAD$ for each prostatectomy performed [18]. Taking into account that a considerable part of these expenses is due to the use of disposable tools, longer and more complex operations lead to higher costs.

- A final drawback of these demanding procedures is the technical workload to which surgeons are exposed that increases with the lengthening of surgical procedures.

In order to face these issues, a new border has been signed: surgical robotics are starting to use Machine Learning to imitate human skills enabling surgeons to overcome all these limits. Using learning strategies, researchers are trying to allow robots to assist surgeons during operations by automatizing some time-demanding, elementary tasks. Moreover, other studies are testing new algorithms that learn how to objectively evaluate trainee surgeons to make more profitable their training curricula.

Intelligent robots can improve patient health by enhancing surgeons performances in terms of both reducing medical errors and improving their expertise level with targeted training programs.

1.2 Aims

By following the future of autonomous robot control, this thesis wants to present a new approach that allows robots to learn how to characterize different surgical procedures. Specifically, the presented algorithms should learn from expert demonstrations of many different surgical operations, capturing their underlying surgical motion. Once the machine learning approach has been defined it can be used to accomplish a real time recognition of the undergoing task and its internal structures.

To face real time task recognition and gesture classification different approaches has been explored. Gesture classification has been studied, in first place from a general point of view, without any considerations on the kind of the undergoing task. Subsequently, a different approach that addresses gesture classification through task recognition has been adopted.

1.2.1 Task recognition

Starting from kinematic data collected from dVSS through a Human Robot Interface (HRI) a first algorithm is defined to identify which task is under performance. The task recognizer looks at the general flow of the kinematic samples identifying online the undergoing surgical task addressing a finer-grained scale characterization of the surgical movements.
The task recognizer is able to understand the surgeon activity with a overall classification accuracy of more than 95%, in about 15 seconds, having from 4% to 15% of the task accomplished.

1.2.2 Gesture segmentation and classification

Since the task recognizer is not able to deeply characterize each surgical procedure and its peculiar and fundamental steps, it is necessary to set up a classifier that works recognizing every movement, or *gesture* (Section 2.2), in winch it is possible to segment the task.

Considering kinematic data collected from dVSS during different surgical procedures a first classifier has been defined to assess real time gesture segmentation and identification without any a priori knowledge about the undergoing task. A second classifier has been subsequently developed to identify activities in surgical procedures considering the previous knowledge about the kind of task under execution. After the initial real time task recognition of the kind of the undergoing surgical procedure, the Task-Specific Gesture Classifier (TSGC) can classify online the peculiar activities accomplished during the operation.

As soon as the task is characterized, the gesture classifier starts to identify gestures in the task: every time frame is classified in less than 0.15 seconds as belonging to one particular gesture. By using this algorithm it is possible to reach an overall accuracy in gesture classification over 3 different surgical task comparable with the actual state of art.

1.3 Overall organization

In order to present all the proposed algorithms the discussion will follow this flow:

- Firstly, the state of the art on recognition and gesture classification is exposed with particular attention to what has been achieved and what is the current knowledge on this particular field.
- Subsequently the necessary algorithms to accomplish task and gesture classification are explained considering and motivating all their parts.
- Chapters *Results* and *Discussion* will show the results achieved by using the

presented methods for recognition and classification. A further possible interpretation is then provided.

- Final conclusions about the work and possible future developments are then taken into consideration in the last chapter of this work.

Chapter 2

LITERATURE REVIEW

2.1 Automatic task recognition

Autonomous task recognition, thanks to the increasing amount of data collected by sensory systems, opens a wide new world of applications.

Inspired by the success in speech recognition of the Hidden Markov Models (HMMs), Sánchez et al. [19] applies this algorithm to detect hospital activities in order to improve the provided information services. The aim of this project was to develop a smart information system able to adapt the users interface according to their working profile (nurses, physicians etc.) and to their most probable queries to the hospital information services. Using contextual data regarding people involved in hospital procedures and also the tools they commonly use to accomplish generic tasks, three HMMs have been trained, one for each working profile. Each one of these models was composed by two HMMs running in parallel: one was necessary to model the used tools, the other to characterize all possible interactions between people. The final model was able to classify undergoing activities with an accuracy around 92%, enough to improve the performance of the hospital information system. Following the idea of improving everyday life of residents, Singla et al. [20] designed a smart home environment relying on artificial intelligence, able to adapt itself to inhabitants routine. In order to attack the problem of activity recognition in a real environment, they built a Markov Model

(MM) to probabilistically determine all the most likely phases of a more complex activity under observation. Thanks to this recognition they were able to track and recognize the correct actions with an overall accuracy of 88.6%.

Even in Surgical environments, the automatic recognition of surgeons activity has been proposed initiating new many possible application to be introduced.
A robust task classifier to model a laparoscopic cholecystectomy has been proposed by Han et al. [21]. Using 17 kinematic signals collected from 12 surgeons performing the aforementioned operation, authors built an initial HMM model in which every action from every training example is represented by one state leading to overfitting. By iteratively merging two states, a more compact model is generated, able to better generalize over new data and eventually characterize every action in the task, with an offline classification accuracy of 99%. A Convolutional Neural Network (CNN), called EndoNet, is designed to carry out activity recognition to characterize laparoscopic cholecystectomy [22]. Based on videos, EndoNet addresses task recognition, not only considering the steps in which each task can be divided but also the involved tools. After 5 convolutional layers and 2 fully-connected layers (Figure 2.1) the tool is detected; then this information is concatenated with an additional fully-connected layer which is used as an input for one-vs-all multi-class Support Vector Machine (SVM) classifier which gives the observations for a two-levels HMM. Finally, this latter performs the task classification considering both inter-phase and intra-phase dependencies, achieving the 92% of accuracy in offline phase recognition.
Considering stages in actual trauma resuscitations [23] accomplishes a multi-class classification. Without any pre processing they use Radio Frequency Identification signals (RFID) to train a network composed of three convolutional layers followed by 3 fully (dense) connected layers. The last fully connected layer provides the classification of one out of five possible stages in which trauma resuscitation can be decomposed with an overall classification accuracy of about 72%.

Other approaches see the task recognition, achieved in many different ways, as a starting key-point for further investigations and applications.

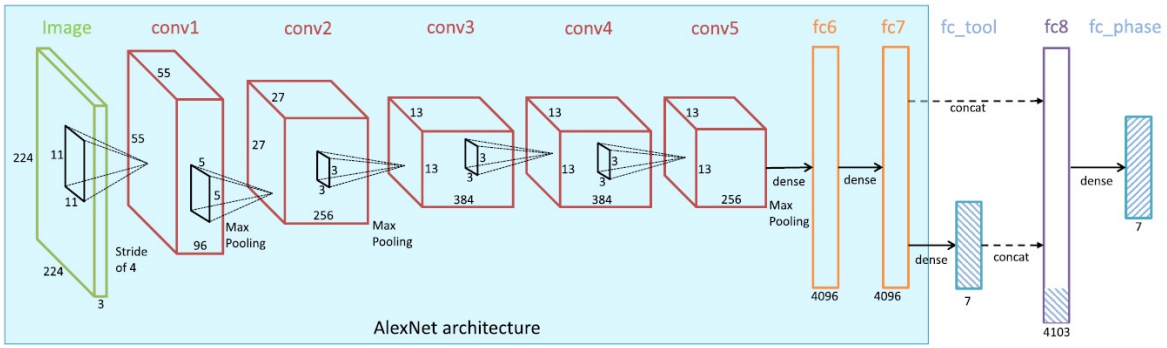

Figure 2.1: *EndoNet [22] architecture for phase recognition; this algorithm is based on the AlexNet network for tool detection.*

A first applicative scenario aims to reduce the workload of surgeons by automating parts of the task. Firstly, robotic assistants have to consider the particular situation to interact with surgeons and eventually, autonomously take over. In [24], experts motion structure is gathered using a Gaussian Mixture Model (GMM) over kinematic data aligned trough Dynamic time Warping (DTW); subsequently the task is actuated trough smoothed trajectory generated in open loop by a Gaussian Mixture Regression (GMR). Murali et al. [6] tries to reduce the surgeon's fatigue by autonomously repeating task as debridement, cutting and suturing. Considering video recordings from two fixed stereo cameras they define a Finite State Machine (FSM) able to learn structures behind activities and to repeat sequences of movements. Even if they demonstrate the effectiveness of their method for automating surgical tasks, they were able to perform operations only at half of speed of expert humans. Another work, [3], uses an *apprenticeship learning* approach to extract reference trajectories from human demonstrations. Using a Kalman smoother over a Gaussian distribution estimated from the human demonstrations, authors fit parameters of the robot dynamic model. By using this fitted dynamic model with a robot controller, performances are enhanced along the task execution. Speeding up performances they succeed in repeating the reference trajectory up to 10 times the human velocity.

Another popular scenario uses the automatic recognition of performed surgical tasks to go towards user's skill assessment. Building trainees performance models and statistically comparing them with others it is possible to understand the particular expertise level contained in the trials. Considering the analogy between spoken languages and

surgical performances, in [25], Markov Models are used to characterize surgical performances of 30 surgeons with a different expertise level. Kinematic data coming from a system called "BLUE DRAGON" [26] are merged with tools information and then grouped into 15 states used to describe tasks. The characterization of states by unique sets of forces, torques and velocities, make the model able to recognize them even if surgical performances were executed in a slightly different way, just like language models recognize words pronounced with different accents. A reference learning curve was finally defined: measuring the quantitative statistical distance between expert and trainee MM models, an objective measure of their expertise level was found. Improving the model of novices and experts, both Megali et al. [27] and Reiley et al. [10] proved the reliability of comparisons between models to obtain an objective measure about users' skills. In particular: Megali et al. [27] builds HMMs out of performed trajectories considering the evolution of frequency content in kinematic time series exploiting Short Time Fourier Transform (STFT). Even if with a small dataset, authors characterize each hidden state of the model clustering the frequency features and, by using a specific metrics, provide a quantitative evaluation of the surgical performance. In parallel Reiley et al [10], using again motion data processed trough STFT, compares two methods implementing HMMs to analyse skill assessment: one modelling the whole task and the other considering a finer-grained scale: the gesture level [11] (see Section 2.2). They finally indicate that HMMs are a useful method to classify skill of unknown trials. Moreover they suggest that using HMMs built at the gesture level, it is possible to improve the accuracy in skill evaluation.

The great majority of these presented works propose HMMs as the best classification algorithm to face the task recognition. However, the recent explosion of interest in small devices (as wearable ones), typically limited in computational power, have created a growing need for more efficient algorithm in terms of time and computational cost. The basic k nearest neighbour algorithm (k-NN) with Dynamic Time Warping (DTW) as embedded measurement system, in all its forms, has become popular[28, 29] especially for comparing and classifying time series.
Starting from the kinematic data of three surgical tasks, acquired from the dVSS and

collected into the JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS) [9], Fard [5] accomplishes to classify surgical tasks by using a kNN-DTW algorithm. She brings very promising results, especially considering real time performances: after 6 to 20% of samples in a demonstration, the model was able to understand which of the three possible tasks was that sequence representing.

Task recognition is a challenging problem, the same action can be performed in various ways in a non-predictable environment requiring different amount of time. In addition to this, surgical procedures are usually complex and composed of many different meaningful steps. This common structure makes the surgical task recognition more confusing and less effective. Moreover, usually task classification is accomplished in a top hierarchy level, too high with respect to the one in which it is necessary to investigate to characterize a so critical and crucial procedure.

2.2 Gesture segmentation and classification

To characterize more precisely an elaborate task, many works address the analysis on a finer-grained scale, allowing the recognition to be more robust since each part of the whole surgical operation is defined and classified.
The finer-grained scale investigation aims to distinguish meaningful single motion sequences called *gestures or surgemes*[11, 10, 12]. These gestures are considered as modular building blocks of every task and just reassembling them it's possible to create new operations, like recombining syllables makes possible to form new words.

In order to better understand the different approaches used to classify surgical gestures, it is possible to divide them into two groups taking advantage of a peculiar aspect of these algorithms: the *learning* process. This stage is usually performed through which is broadly defined as Learning from demonstration (LfD) [30, 31]: starting from human-executed demonstrations, it is possible to sharpen predictive parameters associated to a particular task useful to classify the unknown sequences.

The LfD learning process uses data from human-executed demonstrations to learn about the classification. Several works integrate autonomous systems to fit the predictive parameters directly from the training data without considering any other external information; these algorithms are called *unsupervised* since no information is given on the structure of the dataset. Instead, other researches provide with algorithms a manual segmentation and annotation of tasks adding information regarding the classification of values. These algorithms perform the training phase in a *supervised* way by adapting their predictive parameters to fit the provided division into gestures of the dataset.

2.2.1 Unsupervised gesture classification

As anticipated before, the *unsupervised* learning process is performed autonomously by algorithms which iteratively adapts their parameters to understand the real division of training datasets in order to assess gesture classification. The lack of human intervention leads to many advantages:

- Users cognitive workload is remarkably reduced since their are not request to manually segment and label the whole dataset
- Labels do not contain errors due to human distraction
- Parameters are adapted considering the evolution of the dataset in a fine-grained scale maybe too fine to be detected by humans.

Researchers have proposed a variety of possible solutions to accomplish *unsupervised* gesture classification, some of these use adapted versions of already existing algorithm, others approach classification from a new perspective.

Using a limited dataset composed of human actions and facial expressions, actions are found without assuming any knowledge on either their number or their interpretation: these latter are used to represent the whole performance [32]. A meaningful

action primitive is defined as "*[...] one which can be illustrated visually and described textually (e.g., left arm moving upward, right leg moving outward)[...]*". Main regions of interest in images are highlighted with bounding boxes constructed by considering the optical flow difference in consecutive frames. Starting from these regions, defined in D disjoint videos, a K-means algorithm is used to cluster each box into K possible clusters which will become the Gaussian components of a Gaussian Mixture Model. A certain action primitive, as it is defined, should be common and repeated in subsequent analysed videos. Considering this concept, clusters related in time are grouped into Gaussian mixture distributions. Finally complete actions are recognized with a competitive accuracy, thanks to the relationships between primitives inferred through an HMM.

Also Wu et al. [33] focuses on modelling human activities comprising multiple actions with an unsupervised setting. By using videos representing human daily activities, researchers were able to model long-term movements considering temporal relations and pairwise co-occurrences. In particular they define action-words as a sequence of short-term actions in the video and activities as about a set of action-topics indicating which actions are present into recordings. Following the proposed workflow (Figure 2.2), video are sequenced into overlapping temporal snippets, subsequently, visual features concerning human skeletons are extracted from clips and clustered through a k-means algorithm to form an action-dictionary in which each action-word is represented by each k-th centre. Videos can be seen as sequences of action-words taken from the dictionary and grouped in action-topics considering movements composed of more than a single word. An unsupervised learning model is set up based on both the correlations between topics and time distributions of occurring action-words. The model can assign action-words to topics allowing a proper segmentation and recognition by merging continuous clips with the same assigned topic and considering the meaning of the considered topic.

A different unsupervised approach is used in [34], it is called Transition State Clustering (TSC). In a robot-assisted minimally invasive surgery (RAMIS) context, this work tries to segment demonstrated trajectories to facilitate robot learning. The TSC algorithm fits local linear dynamic models to the demonstrations, after, to improve robustness,

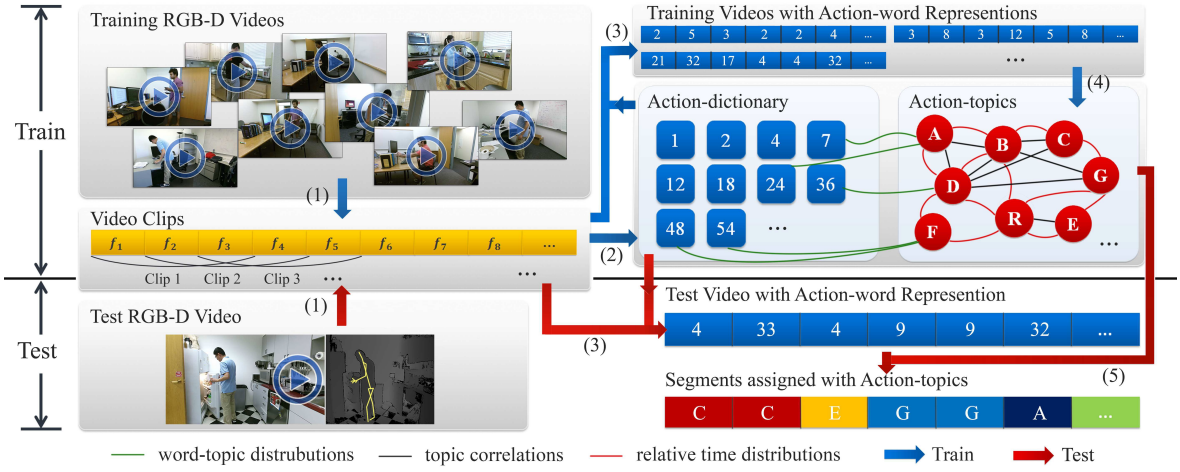

Figure 2.2: *Pipeline used in [33] to: (1) first decompose the video in temporal clips. (2,3) Clustering these clips and map them into action-words in an action-dictionary. (4) Learn the model from the action-words representations. (5) Assign action-words in the video with action-topics.*

multiple clustering and pruning steps are done to remove sparse groups. More in detail, the model is composed of four different levels:

- The first one a GMM is fitted over each demonstrated trajectory and identifies consecutive times with other most-likely mixture components; mixture components which are sets of candidate transitions.
- The consecutive levels cluster these latter over demonstrations: the second level applies GMMs with a fixed number of clusters previously defined by the Dirichlet process (DP) over kinematic data.
- The third one applies a DP-GMM over sensory features.
- The last level of DP-GMM fits over the time axis to incorporate also time related information

The final result contains sub-clusters indexed both in the state space and in time. A final pruning process is necessary to increase the algorithm robustness in identifying those clusters that correspond to common state and time transition conditions.
By using both video and kinematic data provided by the aforementioned JIGSAWS dataset [9] an evolution of the precedent TSC method has been presented: the Transition State Clustering with a Deep Learning (TSC-DL) algorithm [35]. Promising

results of the TSC-DL approach are reached integrating visual features with traditional kinematic data in order to give more information to the internal TSC process that can accomplish segmentation in a more reliable way. Visual features are derived frame-by-frame from video recordings through pre-trained Convolutional Neural Networks, as the AlexNet [36] in Figure 2.1. After a dimensionality reduction, visual features are paired with kinematic variables and fed into a TSC algorithm which performs hierarchical clustering and pruning, firstly in the visual space and then in the kinematic one, as aforementioned. The final accuracy in classifying gestures achieved by the TSC-DL algorithm is around the 73% on suturing tasks and around 55% on needle passing operations.

2.2.2 Supervised classification

In some cases datasets are provided with additional information about data they contain. Using this knowledge it is possible to set up learning processes able to iteratively adjust their parameters to minimize loss functions. This allow decoded classes to become closer and closer to the ones contained in the provided manual annotations of the dataset. To manually encode all the labels and annotations necessary to train a robust classifier can be tedious and stressing. However, sometimes, having the support of an expert can mange ambiguous data and save time that otherwise would be put in efforts to set up autonomous/unsupervised algorithms. In Robotic Assisted Minimally Invasive Surgery, supervised classifiers have spread considerably in the last decades showing different solutions to face all the critical points in a so demanding field.

Lin at al. [11] in order to analyse the skill level of trainee surgeons proposed a less computationally expensive algorithm able to distinguish gestures online. This method reduces 72 kinematic variable retrieved by the Da Vinci system to a more compact space, improving the overall efficiency. After a first feature normalization, Lynear Discriminant Analysis (LDA)[37] is applied to project high-dimensional feature vectors into a lower-dimensional space while optimizing the loss of class discriminatory information. Using a Bayes classifier, they proved the reliability of the LDA reduction

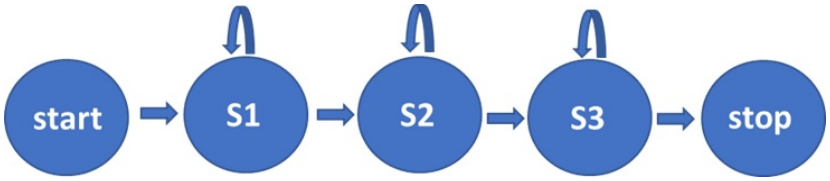

Figure 2.3: *Three states left-to-right HMM model, as proposed in [11], to model each gesture in surgical tasks. Every state S of the HMM represents a sub-phase of a particular gesture deeply characterizing every internal movement.*

approach to simplify the gesture recognition problem, reducing the complexity of the system.

Removing the assumption of working in an online framework a Markov/semi-Markov Conditional Random field (MsM-CRF) model is proposed for a more complete description of the surgical scenario [38]. Both kinematic data and video recordings form the JIGSAWS set are used in this synergistic model to capture local and global cues thanks to its Markov and semi-Markov components. In particular: the unary CRF component with its integrated SVM is able to classify gestures out of single frames. Since gestures are composed of many frames, a first control on results is done considering the temporal coherence of these low level labels. The unary semi-CRF component assigns gesture labels to a group of frames, thereby it gathers global features related to overall gestures. Moreover, considering how it has been defined, two consecutive segments should not have the same label increasing robustness of the whole process.

In accordance with [39] it is necessary to improve gesture classification because: "*[...] as the pool of subjects increases, variation in surgical techniques and unanticipated motion increases [...]*". Following this idea, the authors in [39] first demonstrated that the model proposed in [11] is too simplistic and not able to generalize over more complex dataset. To continue, they introduced in the projected LDA space the use of a Hidden Markov Model (Figure 2.3) to characterize each gesture in suturing tasks performed by different surgeons. This new model, with a higher level of complexity (three state left-to right HMM), was able to better understand all the possible variations related to different users as it could deeply characterize gestures considering also intra-gestures changes.

Taking into account the knowledge about sub-gestures called *dexemes*, Varadarajan et

al. [12] tries to highlight their meaning by applying twice the LDA. In addition, considering encouraging results in the sub-gesture research, he removes the assumptions that in order to optimally characterize gestures models must be composed of three-state left-to-right HMMs. Varadarajan understood that even if the first aim of LDA is to reduce the dimensionality of features without losing information, with the new approach that exploits dexemes, it was necessary to investigate whether it was better to perform LDA to discriminate between sub-gestures rather than entire gestures. To apply LDA at this finer level, a manual segmentation of the dataset into dexemes is required but not provided. In order to overcome this issue a three-state left-to-right HMMs is trained, afterwards feeding this model into the Viterbi algorithm [40] a dexeme segmentation of each gesture is estimated. The resulting dexeme labels are finally input to a second LDA reduction granting a proper training of more performing HMMs based on dexemes. The second effort in [12] is done to deal with the fact that each gesture is not only time dependent but also context related, meaning that a temporal structure left-to-right of the HMMs is not enough to capture all the variance in gestures. For this purpose Data-Derived HMMs (DD-HMM) are built to collect context-dependent variations of gestures using greedy algorithms: starting with a single-state HMM for each gesture, its parameters jointly with the number of states are estimated via Successive State Splitting (SSS) [41]. The last model presented has a peculiar number of states for each HMM modelling a gesture, considering also different transitions between them (Figure 2.4). Despite of promising concepts behind the DD-HMMs construction, accuracy in gesture recognition is basically not improved and maintained similar to state of art results. However, this new algorithm proved its ability in automatically discovering and modelling gestures, supporting dexemes analysis even if the labelling of gestures is very coarse grained or absent.

2.3 Datasets

Recent developments in robotic surgical techniques have provided a significant source of data acquired during surgical operations. These data defined a new output from op-

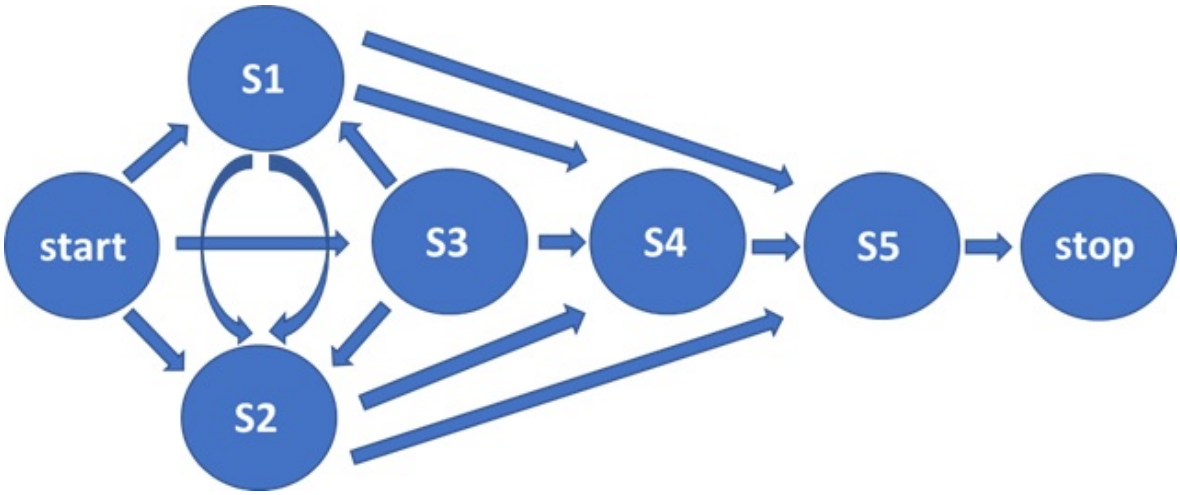

Figure 2.4: *DD-HMM proposed in [12] used to characterize a single gesture by using 5 internal states and their possibles transitions.*

eration room, making the use of robotics fundamental to reach a new point of view and achieve a new comprehension of what are the main clues during surgical procedures. Robotic surgical systems have paved the path for future approaches providing us internal kinematic data and stereoscopic images able to better characterize the environment of study.

The *Da Vinci Surgical System* (dVSS) [8], from Intuitive Surgical, Mountain View, CA, thanks to its research interface (da Vinci API) [7] allows to retrieve information on kinematics variables from both Patient Side Manipulators (PSMs) and Master Tool Manipulator (MTMs) joining them with images obtained by cameras.
In order to access these data for further analysis and studies, many data set considering different tasks have been collected. In particular, this work is based on data provided by the JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS) [9]. This dataset contains data taken from 8 surgeons with a different expertise level, performing three basilar and common surgical tasks: Suturing, Needle Passing and Knot-Tying. For more details about the JIGSAWS working set see Chapter 3.

2.4 Overall approach

In order to accomplish simultaneous task and gesture classification, two different approaches are studied: the first approach directly performs real time gesture classification without any information on the specific kind of task under analysis. The second framework, instead, performs gesture classification after a previous task recognition accomplished by a another algorithm.

2.4.1 Generic gesture classification

The first part of this work aims to develop an efficient real time gesture classifier that does not consider any information about the kind of the particular undergoing surgical task, to accomplish gesture classification. This Generic Gesture Classifier (GGC) is composed by a set of three states left-to-right HMMs. Each of these models attempts to characterize an underlying gesture of the task by fitting a GMM to it. Since a surgical procedure is a flow of continuous gestures, to allow a continuous classification, these HMMs are linked together considering every possible transition among gestures.

2.4.2 Task recognition to address gesture classification

The second used approach intents to identify the undergoing surgical performance to address an accurate and specific gesture classification. The first algorithm of this jointed framework is an efficient real time task recognizer able to perform task recognition, addressing a further task-specific gesture classification accomplished by a second model.

Using data provided by the JIGSAWS dataset [9], the real time task recognizer is based on HMMs. Groups of 3 Gaussians each are used to fit and characterize any of the three surgical task in the working set. These Gaussians are linked together to compose a final 9 states left-to-right HMM useful to recognize any of the proposed tasks. The task recognition is accomplished thanks to the Viterbi algorithm [40]. It is important to report that since the classification is done in real time for every single sample, if a classification error has occurred, the algorithm is able to recover by changing online task estimation (see Chapter 3 for further information).

After the identification of the surgical task, the task recognizer addresses the gesture classification by activating a task-specific gesture classifier. It si possible to define the task-specific gesture classifier as a composite HMM [13]. This model is able to identify gestures in surgical tasks linking them together and describing their relationships. Each state of this composite algorithm will be a particular pre-trained HMM describing a single gesture. Using the underlying structure of the task to link gestures together it is possible to increase the final classification accuracy. In fact, these structures, as grammatical constrains, narrows the research field of classification algorithm that can ignore impossible transitions. It is important to notice that, since the composite HMM contains inter-gesture links, peculiarity of each task, the model is naturally task-dependent.

Chapter 3

MATERIALS AND METHODS

The aim of this project is to implement an efficient algorithm able to accomplish *simultaneous task recognition and gesture classification*, considering different surgical operations as Suturing, Needle-Passing and Knot-Tying tasks.
Two different framework are studied for this purpose. In the first one a Generic Gesture Classifier (GGC) is developed to accomplish real time gesture classification over all the surgical procedures. In the second one a task recognizer autonomously addresses real time the gesture identification to a task-specific gesture classifier.

The discussion about the used algorithms is organized as follows:

- To start, the first section analyses the dataset used for this work: the JIGSWAS working set.
- The second part describes the core of the algorithms implemented in this dissertation with their sub-parts: Markov Chains and Hidden Markov Models. This section is focused on defining the mathematical aspect of models as well as their key processes.
- The third and the fourth section show all derivations of composite HMMs made to accomplish: a gesture classification without considering any particular task, surgical procedure recognition and a subsequent task-specific gesture classification.

- Finally, the last paragraph indicates how the different approaches are evaluated and compared with other works in literature.

3.1 Dataset

In this dissertation all the studies are accomplished by using the data provided by the JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS) [9] acquired from the da Vinci Surgical System (dVSS) through a specific Human Robot interface called *API* (see Section 7.1 for further information).

3.1.1 Surgical tasks and data description

Data in the JIGSAWS are collected from 8 surgeons, with a different expertise level, while they are performing three elementary surgical tasks on bench-top models.

As anticipated before, the included tasks in the JIGSAWS are: Suturing (SU), Needle-Passing (NP) and Knot-Tying (KT) which are very popular and part of surgical training courses. In a more descriptive way we can summarize them as it follows:

- Suturing (SU): surgeons pick up the needle, approach a vertical line in the phantom (to simulate incisions) and pass the needle though the tissue in correspondence of paired dots. After the first needle pass, surgeons extract the needle from the tissue, change their hand, and go further to the other marked points (Figure 3.1 A).

- Needle-Passing (NP): surgeons pick up the needle and pass it through 4 small rings from left to right (Figure 3.1 B).

- Knot-Tying (KT): surgeons tie a single loop knot by taking one end of a suture tied to a flexible phantom (Figure 3.1 C).

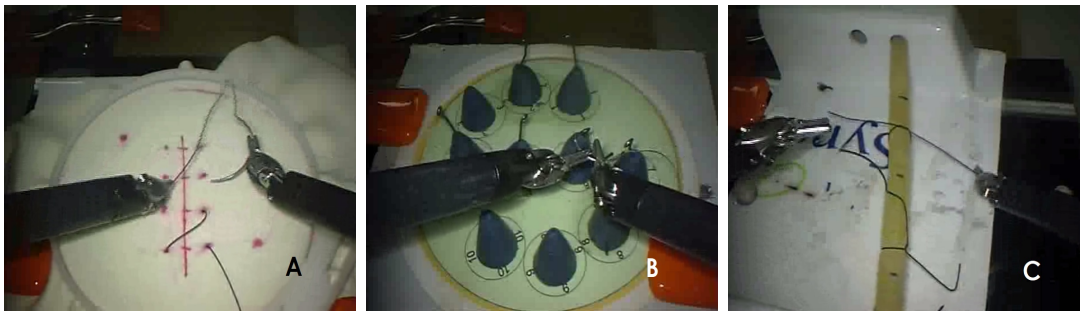

Figure 3.1: *Snapshots of surgical tasks, respectively taken from: A)Suturing, B) Needle Passing and C)Knot-Tying videos contained in [9]*

Surgeons who are performing the tasks in the JIGSAWS have a different expertise level and surgical experience. In particular,

- Two surgeons are considered experts since they reported to have more than 100 hours of robotic surgical practice.
- Two surgeons can be considered intermediate, having between 10 to 100 hours of experience.
- Finally, 4 surgeons are considered novices as they reported to have less than 10 hours of practice.

Doctors repeat each task 5 times and each repetition is considered a *trail*. Authors reported some problems during the acquisition of certain trials which lead to their elimination from the resulting final dataset composed of 39 trials of SU, 36 trials of KT, and 28 trials of NP.

Column indices	Number of variables	Description of variables
1-3	3	Left MTM tool tip position (xyz)
4-12	9	Left MTM tool tip rotation matrix (R)
13-15	3	Left MTM tool tip linear velocity ($x'y'z'$)
16-18	3	Left MTM tool tip rotational velocity ($\alpha'\beta'\gamma'$)
19	1	Left MTM gripper angle velocity (θ)
20-38	19	Right MTM kinematics
39-41	3	PSM1 tool tip position (xyz)
42-50	9	PSM1 tool tip rotation matrix (R)
51-53	3	PSM1 tool tip linear velocity ($x'y'z'$)
54-56	3	PSM1 tool tip rotational velocity ($\alpha'\beta'\gamma'$)
57	1	PSM1 gripper angle velocity (θ)
58-76	19	PSM2 kinematics

Table 3.1: *Kinematic variables included in [9], table taken from the same paper.*

As described in Table 3.1, 19 motion variables from each MTM and PSM of the dVSS are collected through the API interface with a sampling frequency of 30Hz. Considering together data from left and right PSMs as well as the left and right MTMs, 76 kinematic variables are finally used in the JIGSAWS to describe movements in each frame. The 76 motion variables in Table 3.1, collected with respect to a common reference system, include: Cartesian positions, rotation matrix, linear velocities, angular velocities (described in terms of Euler angles) and a gripper angle.
Video data are synchronized with the same sampling rate of the kinematic variables, such that each video frame corresponds to a kinematic data sample captured.

In order to set up the algorithms proposed in this dissertation, only the kinematic variables from the two PSMs will be taken into consideration.

Gesture index	Gesture description
G1	Reaching for needle with right hand
G2	Positioning needle
G3	Pushing needle through tissue
G4	Transferring needle from left to right
G5	Moving to center with needle in grip
G6	Pulling suture with left hand
G7	Pulling suture with right hand
G8	Orienting needle
G9	Using right hand to help tighten suture
G10	Loosening more suture
G11	Dropping suture at end and moving to end points
G12	Reaching for needle with left hand
G13	Making C loop around right hand
G14	Reaching for suture with right hand
G15	Pulling suture with both hands.

Table 3.2: *Gesture vocabulary [13], taken from the same paper.*

3.1.2 Manual Annotations

The JIGSAWS working set provides the manually labelled ground-truth set of gestures in each trial. In particular the dataset divides each repetition of every task in a set of gestures taken from a specified common vocabulary of 15 atomic gestures summarized in Table 3.2. Annotations about gestures are taken by watching videos with experienced gynaecological surgeons: each annotation includes the name of the gesture and its start and end in terms of video and kinematics frames.

Gestures used to describe a surgical task are considered to be taken from a common vocabulary, however not all of them are used in each task. Every procedure is

characterized by a specific subset of gestures: it may contain gestures observed in more than one task as specific gestures of that particular task. For instance, considering Figure 3.2 that shows all gestures in each task, it is possible to observe that while G1 is included in every operation, G12 is performed only in KT. Specifically each subset of gestures used to characterize the performances is formed as follow:

- The SU subset contains the gestures from G1 to G6, and from G8 to G11.
- The NP task contains the gestures from G1 to G6, and G11.
- The KT group encloses the gesture G1 and from G11 to G15.

Figure 3.2 not only represents the characteristic gestures in each task, but also it shows the constraining links among them. These links define a *grammatical* structure which is also specific for a certain surgical operation. This concept should be more clear considering, for example, G1: this gesture is common to all the tasks, however only the KT task contains the transition between G1 and G12 which becomes characteristic for this task. As described in Section 3.5, the property of having specific links within gestures will be exploited in setting up a *Task-specific gesture classifiers*.

3.2 Markov Chains and Hidden Markov Models

All presented models to segment and classify surgical procedures are obtained considering variations of the standard Gaussian Hidden Markov Model (HMM). Considering the basic HMM, the main characteristic they share, that differentiates them from standard HMMs, is the usage of a Markov Chain (MC) to link together HMMs each one representing a gesture within the surgical task. By using a so structured model it is possible to describe completely a surgical task characterizing, at the same time, two different levels: the internal or *elementary architecture* given by HMMs that describe gestures and the overall flow of different gestures called *composite architecture*, provided by MCs.

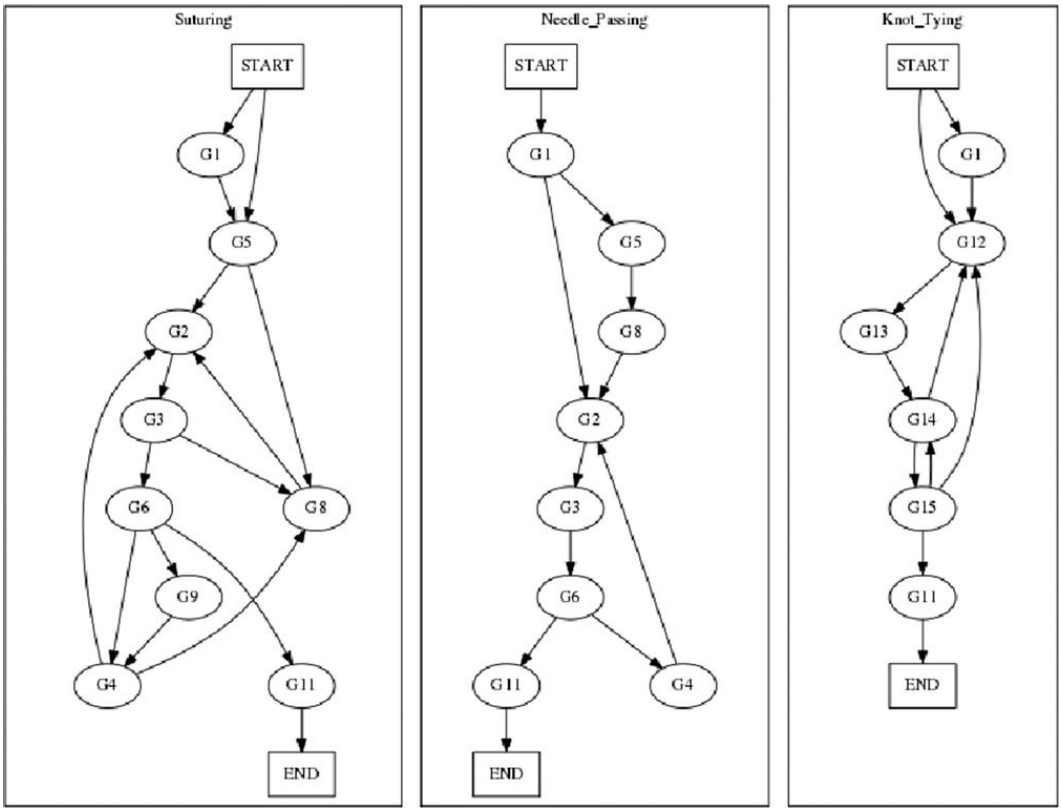

Figure 3.2: *Gesures and their "grammatical" links divided by task [13].*

Models that are going to be described include and share many algorithm to accomplish segmentation and classification problem. In particular, they all implement MCs and HMMs: these two algorithms define the core of all presented models and so they are foremost described to have a general overview of the main processes. Other algorithms allow the tuning of models and grants an efficient classification; these processes will be subsequently discussed in further sections.

3.2.1 Markov Chain

It is possible to consider a MC as the simplest Markov Model [42]: an autonomous stochastic system is called Markov Chain process or Markov Chain if it goes through fully observable states. MC processes can be mathematically described by considering the transition probabilities between their internal states. Figure 3.3 represents a MC that can be explained through mathematical equations: a set of N states is defined by the process $S = s_A, s_B ... s_N$. The process starts in one of these states, defined by an

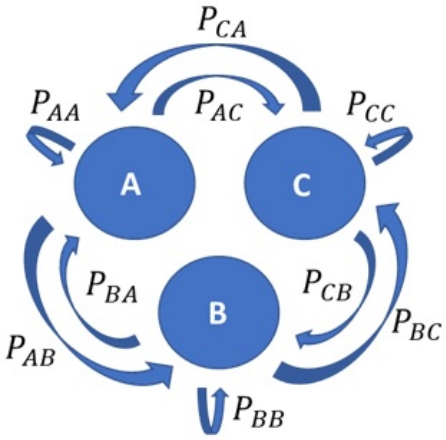

Figure 3.3: *Markov Chain: 3 fully observable states are linked together by defined probabilities*

initial probability distribution, and moves forward from one state to the other by single steps. Each *transition probability* $p_{i,j}$ denotes the likelihood through which the model can pass from state s_i to state s_j. It is important to mention that these probabilities respect the Markov assumption and do not depend on previous state of the chain, they are fixed and characteristic of processes. Transition probabilities useful to describe the process can be represented by a transition matrix (see Table 3.3) in which every row defines a different state from which it is possible to start, and each column a possible state to which is possible to go.

Considering that all values in a row are probabilities that define the ability to go from s_i to one of the other possible states $s_{A,B,..N}$, their sum should be equal to one: Equation (3.1).

$$P_i = p_{iA} + p_{iB} + ... + p_{iN} = \sum_{j=1}^{N} p_{ij} = 1 \tag{3.1}$$

$s_i \setminus s_j$	S_A	S_B	S_C
S_A	p_{AA}	p_{AB}	p_{AC}
S_B	p_{BA}	p_{BB}	p_{BC}
S_C	p_{CA}	p_{CB}	p_{CC}

Table 3.3: *Transition matrix defined for the process in Figure 3.3*

Once that the MC is defined it is possible to predict, given the state s_i in a particular moment, what will be the probability of translating to state s_j, m steps forward in the future. This probability is computed by considering independent transitions over each previous state. Considering a N states MC the $p_{ij}^{(m)}$ probability to go from s_i to s_j in m steps can be computed as follow:

$$p_{ij}^{(m)} = \sum_{k=1}^{N} p_{ik} p_{kj} \tag{3.2}$$

In general, with Equation (3.3), it is possible to calculate the distribution of a given process one steps forward in the future $t+1$ by using the starting probability P_1 and the transition matrix T in Table 3.3.

$$P_{t+1} = P_t * T = (P_{t-1} * T) * T = P_1 * T^t \tag{3.3}$$

This work uses MCs to characterize transitions from a gestures to another. For this purpose the transition matrix in one task is computed by considering the frequency rate of transitions between gestures in training sets. This computation is possible thanks to the JIGSAWS dataset which provides labels of gestures in each task.

3.2.2 Hidden Markov Model

As specified before, the first algorithm used in our composite approach allows to catch the probability to move between subsequent gestures in a specific surgical task. Even if this is an important feature of the whole segmentation and classification process, it is necessary to go deeper to define the main aspects of gestures in order to recognize them if presented to the algorithm.
Following previous works as [13, 12], each gestures will be characterized by considering its internal organization captured through a specific HMM.

A Hidden Markov Model is defined as: "*[...] a doubly stochastic process with an underlying stochastic process that is not observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observed*

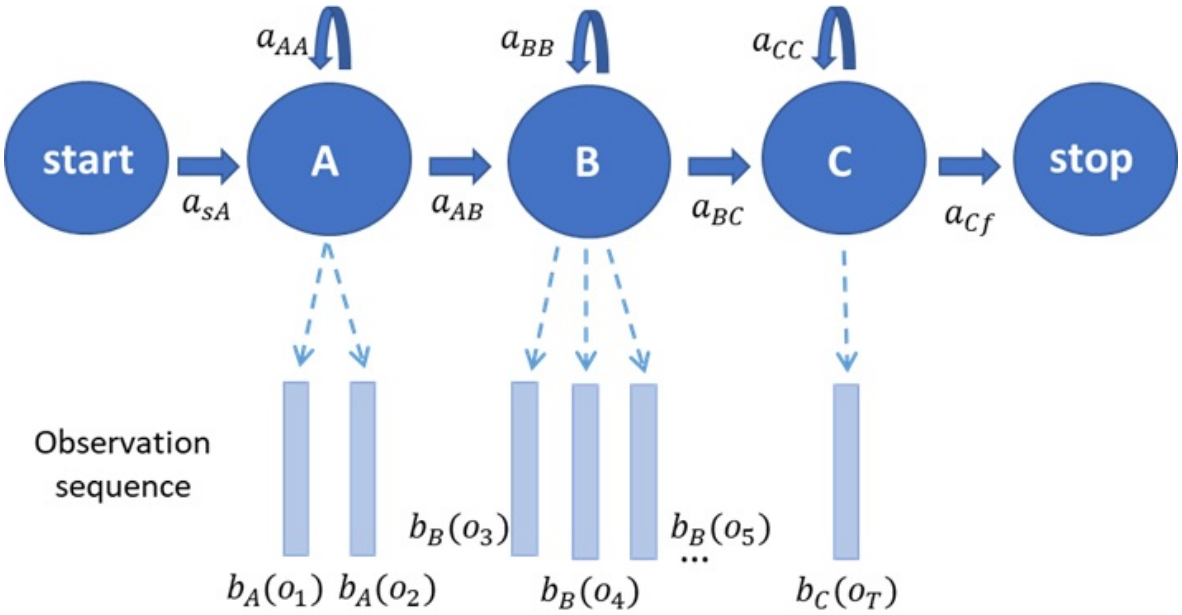

Figure 3.4: *Left-to-right Hidden Markov model with 3 states, it is defined as λ. This particular structure will be used to characterize each gesture in the JIGSAWS surgical tasks*

symbols [...]" [14].

In HMM based problems, a set kinematic vectors or *observations* $O = \{o_1, o_2..o_T\}$ in which o_t is the vector at time t, is assumed to be generated by a HMM λ (Figure 3.4) [43]. These observation vectors are related to gestures appearing in surgical tasks and modelled by the hidden states of the HMM. Every time step in the model corresponds to a kinematic vector generated from the probability density function $b_j(o_t)$ of an unknown hidden state. As described before for MC, transitions from state i to state j are probabilistic and also in HMM they are represented by a transition matrix A and its elements a_{ij} as set up in Table 3.3. Considering the example in Figure 3.4, a stochastic process moves within a three states model following the state sequence $S = \{start, A, ..., A, B, ..., B, C, ..., C, stop\}$, in order to generate the sequence $O = \{o_1, o_2, ..., o_T\}$ (the start and stop states, are considered non-emitting, so no observations are generated from them). The probability of the observations O is finally generated from the model by moving through the states $S = \{start, A, ..., A, B, ..., B, C, ..., C, stop\}$. In particular it can be computed as the product between the transition a_{ij} and the output probabilities. Considering Figure 3.4 that refers to the model λ, the probability that O is generated from the model is:

$$P(O, S|\lambda) = a_{sA} b_A(o_1) a_{AA} b_A(o_2) a_{AB} b_B(o_3) ... a_{BC} b_C(o_T) a_{Cf} \tag{3.4}$$

Up to now the output probabilities $b_j(o_t)$ also called emitted probabilities, are still not determined. These particular values define the possibility for a certain observation (o_t) to belong to a certain state j: if the observation (o_t) is very similar to the ones considered by the state j, $b_j(o_t)$ will have high value, meaning that the state recognizes or can *emit* that kinematic vector, otherwise $b_j(o_t)$ will be really low. Considering that kinematic data are continuous parameters, $b_j(o_t)$ will take the form of a continuous multivariate Gaussian density function represented by Equation (3.5), with mean μ, covariance Σ and n as dimensionality of O.

$$\mathcal{N}(O; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2}(O-\mu)' \Sigma^{-1} (O-\mu)} \tag{3.5}$$

Due to this hypothesis, the emission probability $b_j(o_t)$ is finally defined as

$$b_j(o_t) = \prod_{s=1}^{S} \left[\sum_{m=1}^{M_s} c_{jsm} \mathcal{N}(o_{st}; \mu_{jsm}, \Sigma_{jsm}) \right]^{\gamma_s} \tag{3.6}$$

where M_s is the number of mixture components in the s-th Gaussian mixture model, also called stream, c_{jsm} is the weight of the m-th component and the exponent γ_s is a stream weight used to highlight a particular stream. In the algorithms presented here, it is going to be considered just one mixture component, and just one stream of Gaussians, thus $M_s = s = 1$. This assumption is proved to be effective [13, 12] in decreasing the complexity of the final model and reducing chances for possible overfitting. Due to this simplification, all the parameters that depend only on m or s are always set equal to 1 (ex. c_{jsm}, γ_s).

Even if every member of Equation (3.6) is explained, in a real case that faces unknown data streams, only observation sequences O are sampled and no further information is provided about the emitting state sequence S. Without knowing S, it is impossible to associate the correct emission probability b_j to each sample o_t, making the Equation (3.4) unusable to find $P(O, S|\lambda)$.

Although the formulation of the problem is not directly tractable, by using iterative techniques it is possible to learn the model parameters starting form expert demonstrations, with a normal LfD approach (Section 2.2), and then, it is possible to solve the problem $P(O, S|\lambda)$ estimating the state sequence S of unseen surgical trials. Rather than on the final likelihood $P(O, S|\lambda)$, this dissertation is more focused on optimizing the accuracy over the state sequence S estimation: this sequence represent, considering the JIGSAWS data, the time flow of surgical gestures of the operations performed by 8 surgeons. Behind the gesture estimation all the HMM potential is shown: indeed, they are not only able to classify underlying gestures among unknown trials, but also they are able to segment these performances recognizing when each gesture starts and finishes.

In order to be more clear, the usage of an Hidden Markov Model will be divided into its two peculiar parts: the learning and the inferring, the former is about the way the algorithm allows learning from expert demonstration, the latter will describe in detail how to classify gestures and tasks.

3.2.2.1 Learning: the Baum-Welch algorithm

To better understand, mathematical expressions used to characterize HMMs will always use the following notation:

N	Number of states
T	Number of observations
O	A sequence of observations
o_t	The observation at time $t, 1 \leq t \leq T$
a_{ij}	The probability of a transition from state i to j
μ_j	Vector of means for the state j
Σ_j	Covariance matrix for the state j
λ	The set of all parameters that define a HMM

Table 3.4: *Mathematical notations proposed in [43].*

It is possible to define more functional definitions of Hidden Markov model: in particular the model λ in Figure 3.4 can be seen as a set of parameters $\lambda = (\Pi, A, B)$ that must be tuned by LfD in order to make the whole model usable. In this compact notation Π represents the possibility to start in every state of the model, A contains the transitions probabilities as anticipated before and B defines the emission likelihoods of the states.

The main an well known approach to accomplish the tuning of Π, A, B parameters is the Baum-Welch estimation algorithm [40]. The following mathematical explanation of the Baum-Welch process will be made in a simplistic manner. In particular: no mixture components as well as stream of Gaussian distribution Equation (3.5) are taken under consideration, since they are not considered in any model presented by this dissertation. Moreover, the given formulae refers to the case in which only one set of data observations is used to train each HMM. Considering that the JIGSAWS dataset contains 5 trials for each user for every task, this assumption is clearly not valid.
To have a more detailed mathematical proof of what it is implemented in the proposed algorithms see Appendix 7.1.2 where it is possible to find all the equations necessary to accomplish the learning of HMMs by using multiple observations.

To provide a good estimation of the model parameters Π, A, B, the Baum-Welch algorithm first requires rough initialization of them. Afterwards, the process iterates means and variances in the model considering every state j characterized by an output Gaussian distribution Equation (3.7).

$$b_j(o_t) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_j|}} e^{-\frac{1}{2}(o_t - \mu_j)' \Sigma_j^{-1} (o_t - \mu_j)} \tag{3.7}$$

Since the underlying state sequence is unknown, it is not possible to assign directly observations to individual states, therefore it is not possible to just estimate means and variances from the data by using Equations (3.8) and (3.9).

$$\widehat{\mu}_j = \frac{1}{T}\sum_{t=1}^{T} o_t \tag{3.8}$$

$$\widehat{\Sigma}_j = \frac{1}{T}\sum_{t=1}^{T}(o_t - \mu_j)(o_t - \mu_j)' \tag{3.9}$$

Even if Equations (3.8) and (3.9) cannot be used by the iterative algorithm, they provide an efficient way to initialize the mean μ_j and the variance Σ_j of each state in the so called *flat start* or *flat initialization*. By using this initialization technique, μ_j and Σ_j for each state are set equal to the mean and the covariance of the whole dataset. The iterative Baum-Welch algorithm will change them in next steps.

Once the algorithm is initialized, every observation o_t is assigned to every state in accordance to the probability of having o_t emitted by the states of the model when the kinematic vector is sampled. By iteratively applying Equations (3.10) and (3.11), means and covariances of HMMs are estimated.

$$\widehat{\mu}_j = \frac{\sum_{t=1}^{T} L_j(t) o_t}{\sum_{t=1}^{T} L_j(t)} \tag{3.10}$$

$$\widehat{\Sigma}_j = \frac{\sum_{t=1}^{T} L_j(t)(o_t - \mu_j)(o_t - \mu_j)'}{\sum_{t=1}^{T} L_j(t)} \tag{3.11}$$

In Equations (3.10) and (3.11) the continuous re-estimation of means and covariances is possible thanks to the $L_j(t)$ term which describes the probability to be in state j at time t.

By using the same re-estimation concept applied to tune μ and Σ, it is possible to fit also the transition values contained in the A matrix. The formulae for this complex process, directly adapted to the case of having multiple trials of observations, are provided in Appendix 7.2, (see Equations (7.1),(7.3) and (7.4)).

The last fundamental part of the Baum-Welch algorithm is focused on computing the likelihood of being in state j at time t $L_j(t)$ by using a *Forward-Backward* algorithm.

The first variable considered in the Forward-Backward step is the *forward term* $\alpha_j(t)$, defined as follow:

$$\alpha_j(t) = P(o_1, o_2, ...o_t, x(t) = j|\lambda) \tag{3.12}$$

$\alpha_j(t)$ represent the joint probability of observing the t kinematic vectors and being in state j at time t. It and can be computed in a recursive way by using the Equation (3.13). It is important to note that, even if it is considered as such, $\alpha_j(t)$ is not exactly a probability since the output distributions are densities and not likelihood.

$$\alpha_j(t) = \left[\sum_{i=2}^{N-1} \alpha_i(t-1)a_{ij}\right] b_j(o_t) \tag{3.13}$$

The recursion that leads to $\alpha_j(t)$ can be explained by considering that the probability of being in state j at time t, seeing the observation o_t, is computed by summing the α probabilities for all previous states i mediated by their transition terms a_{ij}. It is necessary to consider that, since the first state (*start* in Figure 3.4) and the last one (*stop* in Figure 3.4) are non-emitting state, they act as the two boundaries of the Forward-Backward process.

The initial conditions for the $\alpha_j(t)$ computation are

$$\alpha_1(1) = 1 \tag{3.14}$$

$$\alpha_j(1) = a_{1j}b_j(o_1),\ 1 < j < N \tag{3.15}$$

while considering the final non emitting state, the final condition is given by Equation (3.16) that, considering the definition of $\alpha_j(t)$, leads to Equation (3.17).

$$\alpha_N(T) = \sum_{i=2}^{N-1} \alpha_i(T)a_{iN} \tag{3.16}$$

$$P(O|\lambda) = \alpha_N(T) \tag{3.17}$$

The second and last term considered in the computation of $L_j(t)$, the probability to be in state j at time t, is the so called *backward probability*, represented by Equation

(3.18). Again, this term is not properly a probability.

$$\beta_j(t) = P(o_{t+1}, ..., o_T | x(t) = j, \lambda) \tag{3.18}$$

As in the computation of $\alpha_j(t)$, even in calculating $\beta_i(t)$ a recursive method is applied:

$$\beta_i(t) = \sum_{j=2}^{N-1} a_{ij} b_j(o_{t+1}) \beta(t+1) \tag{3.19}$$

with the initial and final conditions given by Equation (3.20) and (3.21):

$$\beta_i(T) = a_{i,N},\ 1 < i < N \tag{3.20}$$

$$\beta_1(1) = \sum_{j=2}^{N-1} a_{1j} b_j(o_1) \beta_j(1),\ 1 < i < N \tag{3.21}$$

$\alpha_j(t)$, as it is defined, can be interpreted as a joint probability, while $\beta_i(t)$, thanks to its definition, is more like a conditional probability. This difference in meaning allows the probability of state occupation to be determined by taking the product of $\alpha_j(t)$ and $\beta_i(t)$:

$$\alpha_j(t)\beta_i(t) = P(O, x(t) = j|\lambda) \tag{3.22}$$

Finally, $L_j(t)$ can be written as

$$L_j(t) = P(x(t) = j|\lambda) = \frac{P(O, x(t) = j|\lambda)}{P(O|\lambda)} = \frac{1}{P}\alpha_j(t)\beta(t) \tag{3.23}$$

A final consideration on the Baum-Welch algorithm can be done considering that computations of the forward and backward probabilities involve the products of a large amount of probabilities leading to tiny values. Hence, to avoid numerical problems, the same formulae can be transposed on a logarithmic scale amplifying the range of possible results from the interval $[0, 1]$ to $]-\infty, 0]$.

3.2.2.2 Inferring: the Viterbi Algorithm

The Viterbi algorithm [40] allows to identify the best maximum likelihood state sequence and, afterwards, to perform the classification.
To understand how it works it is possible to consider that the Baum-Welch algorithm, previously described, uses the recursive formulae of forward probabilities to find the total likelihood or log-likelihood $P(O|\lambda)$. By properly adapting this approach, it possible to succeed in identifying the best maximum likelihood state sequence.

Given λ in state j at time t, the maximum likelihood $\phi_j(t)$ of kinematic observations, from o_1 to o_t, can be iteratively computed by using the Equation (3.24), or in terms of log-likelihood to avoid underflow risks using Equation (3.25)

$$\phi_j(t) = \max_i \left[\phi_i(t-1)a_{ij}\right] b_j(o_t) \tag{3.24}$$

$$\psi_j(t) = \max_i [\psi_i(t-1) + \log(a_{ij})] + \log(b_j(o_t)) \tag{3.25}$$

The initial conditions used to compute $\phi_j(t)$ at any time are

$$\phi_j(1) = 1 \tag{3.26}$$

$$\phi_j(1) = a_{1j} b_j(o_1),\ 1 < j < N \tag{3.27}$$

The process stops at the very end of the sequence O (3.28), in other words once reached the o_T sample.

$$\phi_N(T) = \max_i [\phi_i(T) a_{iN}] \tag{3.28}$$

To have a clearer idea about how the Viterbi algorithm works, it is possible to refer to Figure 3.5: the algorithm tries to find the best path through a matrix where the vertical axis represents every states of the HMM while the horizontal defines the kinematic frames (time). The probability of any path is computed simply by multiplying transition probabilities with output probabilities (respectively lines and dots in Figure 3.5) along that path. At time t, each partial path $\phi_i(t-1)$ is known for all states i

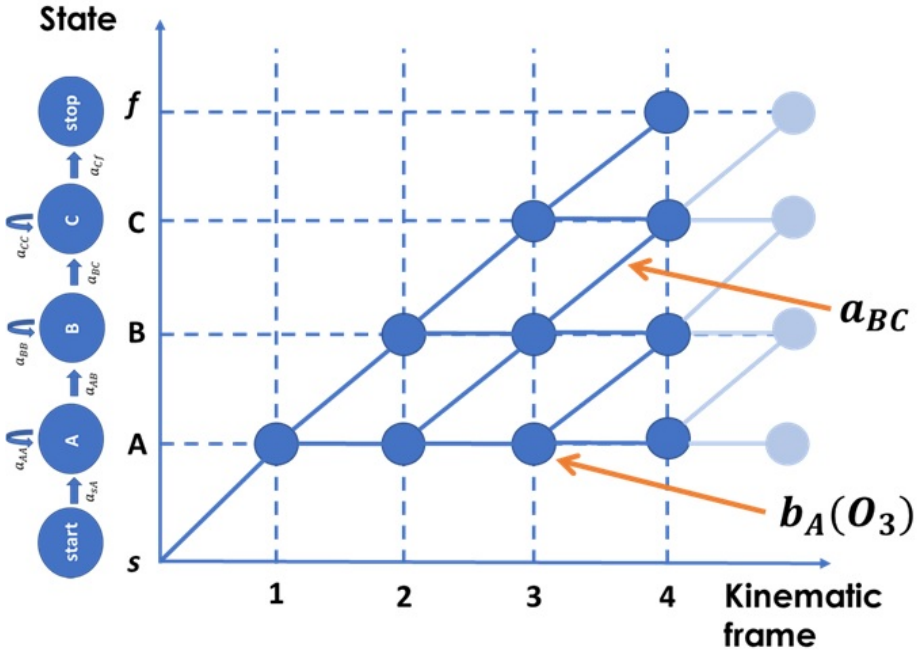

Figure 3.5: *Viterbi algorithm based on model in Figure 3.4, for the recognition of a single gesture*

and by using Equation (3.24) or Equation (3.25) $\phi_i(t)$ is computed and partial paths are extended by one time frame.

3.2.3 Final model

The two models described in the previous section, are joint together to set up what is called *composite HMM*. This final model, at the same time, uses the transitions probabilities defined through a Markov Chain to identify the possibility to pass from one gesture to another while it deeply characterizes each gesture by considering its relative HMM. In Figure 3.6B it is possible to visualize the general aspect of the final model: each transition between gestures (orange lines) links two subsequent gestures deeply characterized by their complete 3 states left-to-right HMMs.

Taking into account the general model and its parts highlighted in Figure 3.6, it is possible to modify it in order to differentiate its final usage as a function of the scope of each composite HMM. To define a method for simultaneous task recognition and gesture classification in surgical robotics, 3 different models have been derived from

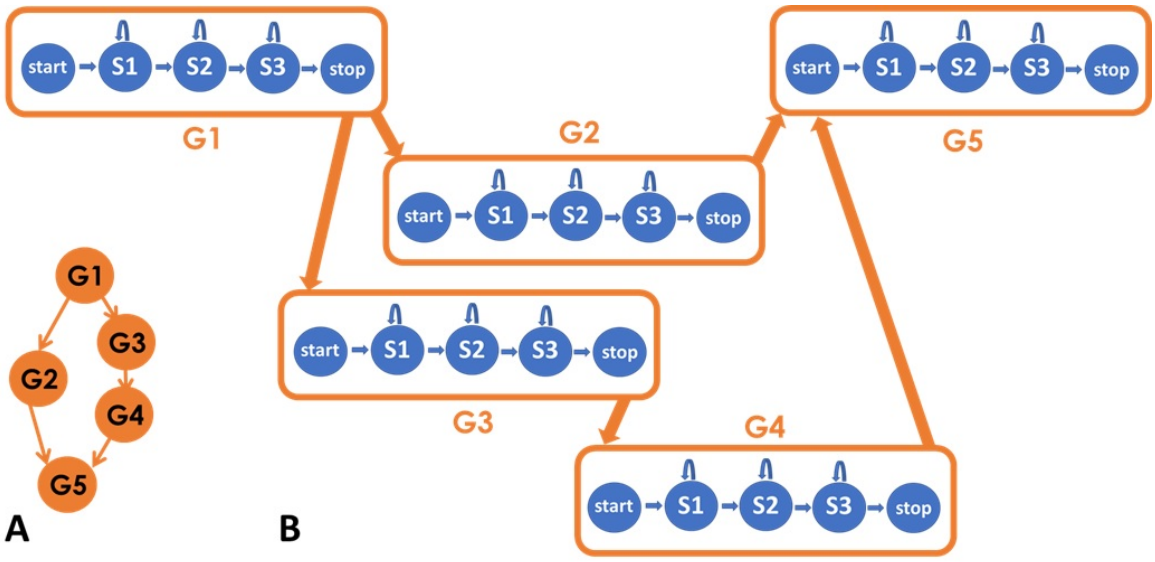

Figure 3.6: *A. Overall structure of a composite model, 5 gestures linked together by a Markov Chain. B. extended structure of a composite HMM: 5 gestures, each one defined by a 3 states left-to-right HMM, are linked together by a Markov Chain.*

the general composite HMM.

- Generic gesture classifier: A three states left-to-right HMM is used to model each gesture of the same kind in the JIGSAWS. Linked together to form the composite HMM they must be able to perform real time classification over gestures without having information regarding the undergoing task.

- Task-related task classifier: for each task in the JIGSAWS, a three states left-to-right HMM is modelled. The final purpose of these models is to catch the general flow of each surgical procedure to recognize in real time the overall meaning of it classifying the task. Every three state left-to-right HMM has 3 Gaussians as emitting states and their interpretations have no connections to gestures.

- Task-Specific gesture classifier: based on [13], this model uses the a priori knowledge of which task is in progress to classify in real time gestures, by using a different composite HMM for each task. A three states left-to-right HMM is used to model each gesture of a specific task. Finally they are subsequently linked together by using the known structure of that particular task Figure 3.2.

The structure of each implemented composite HMM is carefully described in the following sections.

3.3 Generic gesture classifier

With the Generic Gesture Classifier (GGC) gestures are classified in real time without knowing which surgical procedure is undergoing. This generic approach is essential for real case scenario, in which the characterization of tasks in a fine-grained scale is needed but no information is previously given about the operation.

3.3.1 Model description

All trials in the JIGSAWS dataset are merged together, without considering the task they are part of. To incorporate the temporal context into features the concatenation of $2P+1$ consecutive kinematic samples into fifty percent overlapping frames is performed [12, 44]. Subsequently two possible approaches are followed: in the first, kinematic features are standardized in each frame using Equation (3.30), in the other they are not. Features are reduced through LDA and finally subdivided basing on their provided labels into gestures forming a dataset to train the model. To characterize each gesture in the dataset a three states left-to-right HMM is set up considering each state described by a Gaussian. Every possible inter-gesture transition of every task (Figure 3.2) is captured by a Markov Chain, and merged with the gestures HMMs in a unified system (as the one in Figure 3.6) that considers the relation between states and gestures.

3.3.2 Model application

The Viterbi algorithm (Described in Section 3.2.2.2) processes unseen trials to have a final gesture classification. It is important to report that all the possible sequences of gestures G are computed in real time for every new sample at time t. By taking the sequence of gestures g, with $g \in G$ which maximises the likelihood up to time t (Equation (3.29)), it is possible to have a real time classification af all the gestures en-

countered. Moreover, if in the next sample at time $t+1$ the sequence of gestures g that maximizes Equation (3.29) changes, the algorithm will recover its inference providing a new classification.

$$L_{inferred\ sequence}(1,t) = \max_{g} P(O_{1,t}|\lambda_g) \tag{3.29}$$

3.4 Task-related task recognizer

The purpose of the Task-Related Task Recognizer (TRTR) is to understand online which task is in progress by looking to general features of the trial capturing the main structure of the process.

3.4.1 Model description

Using the JIGSAWS dataset, three different 3 states left-to-right HMMs are trained each one over trials that belong to a different task. These HMMs are merged together in one system which unifies the starting and the transition probabilities of all independent models. The starting probability Π of the merged model λ is equal in each one of these 3 HMMs, thus Π becomes $\Pi = [0.33, 0.33, 0.33]$ where each value define the probability of starting in one different HMM. The transition probabilities are also merged in one unified matrix A defining the probabilities to change state inside the same HMM since no transitions are allowed between different models. It is important to report that the correspondence between states in the transition matrix A and the Gaussians which describe emission probabilities related to each state b is maintained.

3.4.2 Model application

Before the training as well as before the decoding data should be pre-processed. In particular,

1. The whole dataset is unified in just one set of samples that contains every trial from every task. Every sample in this set is labelled with a sign correspondent to the task to which it belongs and concatenated with others $2P + 1$ into fifty percent overlapping frames, to help incorporating the temporal variance into features [12, 44].

2. To make the kinematic variables in the training set comparable, a standardization (see Equation (3.30)) is provided in order to have zero mean and unit variance variables.

$$\widehat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j} \, , 1 < j < length(variables) \, , 1 < i < length(set) \tag{3.30}$$

   where μ_j is the mean of the j kinematic variable and σ_j its variance.

3. The total number of kinematic variables is reduced thanks to LDA [37] considering the task-label of each sample. This process improves the overall efficiency of the algorithm maintaining unchanged its discriminant power.

Once that each task-related HMM is trained through the Baum-Welch algorithm over all the samples of a specific task, the final system is used for real time task recognition.

Every sample is sent to the Viterbi algorithm which considers the HMM of each task, keeping trace of the best path within HMMs. At any time t the path with the highest probability is classified as the best path and the related task is recognized. The following expression allows the maximization of the path as a function of surgical tasks: $\max_k P(O_{1,t}|\lambda_k)$, where k represents the surgical task (i.e. SU, NP, KT).

3.5 Task-specific gesture classifier

A Task-Specific Gesture Classifier (TSGC) is built to accomplish real time gesture classification by using the a priori knowledge of the relationships between gestures (see Figure 3.2). TSGC is not generic and, in order to perform profitably gesture classification it strongly depends on the TRTR (described in Section 3.4) which addresses the

classification in a generic scenario with multiple possible surgical tasks.

The particular relationships between gestures that characterize each task (represented in Figure 3.2) are used as additional information to narrow the final classification. As anticipated before, this approach can provide a robust classification if paired with a TRTR classifier that works without any assumption on the surgical task performed, addressing the gesture classification.

3.5.1 Model description

The trials in the JIGSAWS dataset are grouped according to their task, subsequently these task-related groups are used to train three different classifiers, one for each surgical procedure. In this way, all samples of the same task are processed together: they are concatenated first into groups of $2P+1$ subsequent fifty percent overlapping frames, then the kinematic variables could be standardized with Equation (3.30) or not, forming 2 different training sets. Finally, original kinematic features, concatenated and, in case, standardized are reduced through LDA and subdivided basing on their original gestures.

It is necessary to highlight that since the LDA process reduces features in the task-related groups of trials considering their subdivision into gestures, and since the gesture content in each one of these task-related groups is different, the final reduction of features, for each task, will be different. In other words, by applying the LDA over task-dependent datasets, the reduction will be different, leading to a differentiation of gestures as a function of the task.

Different sets of three states left-to-right HMMs are trained over data of each distinct gesture of each task. HMMs representing gestures of the same surgical procedure are merged together in one unique system by using a Markov Chain that defines task-related constraints as described in Figure 3.2.

3.5.2 Model application

Through the Viterbi algorithm (Section 3.2.2.2) the gesture classification is achieved. As in the other algorithms presented before, for every new sample at time t, it is possible to infer the underlying gesture sequence real time, by finding the sequence g that maximizes probability $\max_g P(O_{1,t}|\lambda_g)$.

3.6 Evaluation protocol

In order to compare the aforementioned techniques with the ones from literature, it is necessary to use common cross-validation settings and evaluation metrics.

3.6.1 Cross-validation settings

As proposed in [9] two different cross-validation settings are taken into account:

- The first setting is called Leave-One-User-Out (LOUO). Here, all trials performed by a single surgeon are left out as a test set while the remaining ones are used as a training set. Considering that the JIGSAWS dataset includes data from 8 different users, by taking out all the sessions of a surgeon in turn, it is possible to define 8 paired homogeneous groups of training-test datasets. They can give important information about the way the algorithm can capture variations within surgeons since, for every group, the test user is unseen and its style unknown.

- The second cross-validation framework, instead, is called Leave-One-Super-Trial-Out (LOSO). In this case, the i-th trial of each subject is left out as the test set. Due to the fact that each surgeon repeats the same task 5 times, it is possible to set up 5 different training-test groups. Since only one instance is taken out for the test set, the training sets of these groups contain trials of every user, making them more reliable and less sensitive to changes in performances executed with different abilities and styles.

3.6.2 Measurement metrics

The definition of the cross-validation settings makes the computations of following metrics meaningful insuring their comparability thanks to the homogeneous substrate over which they are calculated.

Performances of implemented algorithms are measured in terms of micro average accuracy, macro average accuracy and precision, as defined in [7]. In particular, for each training-test set group g, a confusion matrix C_g is computed considering classes classified during the gesture recognition as: $C_g[i, j]$ equal to $class_i$ predicted as $class_j$. The complete confusion matrix, considering all cross-validation groups g-th, is computed by summing-up all confusion matrices C_g:

$$C = C_1 + C_2 + ... + C_G = \sum_{g=1}^{G} C_g \tag{3.31}$$

By using the complete confusion matrix (Equation (3.31)), it is possible to define the first measurement metrics: the micro average accuracy, computed as the average of the total number of correct predictions over the total number of predictions. Equation (3.32) provides the formulae for micro average accuracies computation, here n is the total number of possible classes.

$$micro = \frac{\sum_{i=1}^{n} C[i, i]}{\sum_{i,j=1}^{n} C[i, j]} \tag{3.32}$$

It is possible to extend the micro average accuracy metrics considering the classification performances of the algorithm over every kind of surgical operation under analysis. In particular, averaging the summation of the *micro* average accuracy computed considering the three surgical tasks, it is possible to find the *consolidated accuracy* performance of the model over all the operations (Equation (3.33)).

$$consolidated\ accuracy = \frac{\sum_{i=1}^{tot_{tasks}} micro(i)}{tot_{tasks}} \tag{3.33}$$

Here, tot_{tasks} is the total number of analysed tasks. Considering the data used in this dissertation $tot_{tasks} = 3$, and the analysed surgical tasks are: suturing, needle passing and knot tying.

Another meaningful metrics is the macro average accuracy computed considering the *positive rates* for each class by the Equation (3.34). Its standard deviation is defined through Equation (3.35).

$$macro = \frac{1}{n}\sum_{i=1}^{n}\frac{C[i,i]}{\sum_{j=1}^{n}C[i,j]} \tag{3.34}$$

$$macro\ std = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\frac{C[i,i]}{\sum_{j=1}^{n}C[i,j]} - macro\ average)^2} \tag{3.35}$$

Finally it is possible to consider the precision (Equation (3.36)) and its standard deviation (Equation (3.37)) of the classification algorithm:

$$precision = \frac{1}{n}\sum_{i=1}^{n}\frac{C[i,i]}{\sum_{k=1}^{n}C[k,i]} \tag{3.36}$$

Precision is defined as the sensitivity for confusion matrices: $\frac{True\ positives}{Predicted\ Positives}$.

$$precision\ std = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\frac{C[i,i]}{\sum_{k=1}^{n}C[k,i]} - precision)^2} \tag{3.37}$$

In order to be sure that all results are directly comparable and not corrupted, they are all obtained by running a script in MATLAB 2017b on the same machine. The computer has the 64 bits version of Windows 10 Enterprise as operating system and the machine hardware is composed of a processor Intel Xeon CPU E5-2667 v4 (32CPUs) with a clock frequency of 3.2GHz.

Chapter 4

RESULTS

In order to have the best possible time performances as well as classification accuracies, every algorithm presented in this dissertation is optimized considering its internal parameters. Its performances are presented and compared with the state of the art by analysing metrics presented in Section 3.6.

Model parameters have a direct role in defining performances of classification algorithms. Thus, the study of every possible combination of the number of concatenated samples (P) with the number of features (dim) used to characterize each frame is essential for a subsequent task recognition or gesture classification. The tuning of these parameters affects different aspects of the model, in particular it is possible to optimize performances of models by considering how different setups change the overall accuracy and the ability of the model in classifying sequences.

- Overall consolidated accuracy: It is possible to infer a first set of parameters by analysing how the *consolidated accuracy* in recognition, computed through Equation (3.33), changes varying P and dim. Every combination of P and dim is studied considering the LOSO and LOUO validation scheme.

- Unclassified sequences: By increasing the complexity of the model, the Viterbi algorithm can fail in processing surgical trials. This phenomenon can be measured as the percentage of unclassified sequences over the total number of trials. This

number has to be maintained as lower as possible to have a good reliability of the model.

Once the optimized model is defined, its best performances are presented in terms of micro average accuracy, macro average accuracy and in terms of precision as defined in Section 3.6. It is important to report that, since these algorithms should work online, also time performances will be taken in consideration. Performances will consider:

- Time: to accomplish real time classification it is necessary to consider that a new sample is provided with a frequency of 30 Hz. By concatenating P samples with an overlapping value of 50%, the real frequency to which the algorithm should work is $30Hz/P$. This puts limits in the allowed computational time and it becomes a key feature.

- Accuracy and relevance: The micro average accuracy is evaluated as presented in Section 3.6. The robustness of the optimized algorithm is measured, as previously mentioned, as the percentage of unclassified sequences over the total number of trials.

4.1 Generic gesture classifier

Following the framework proposed in Figure 4.1, GGC allows to accomplish a real time gesture classification without considering which particular surgical task is performed. Every performance is sampled at 30 Hz and every sample, of 38 kinematic variables, is processed to be concatenated, standardized and reduced in dimensions.

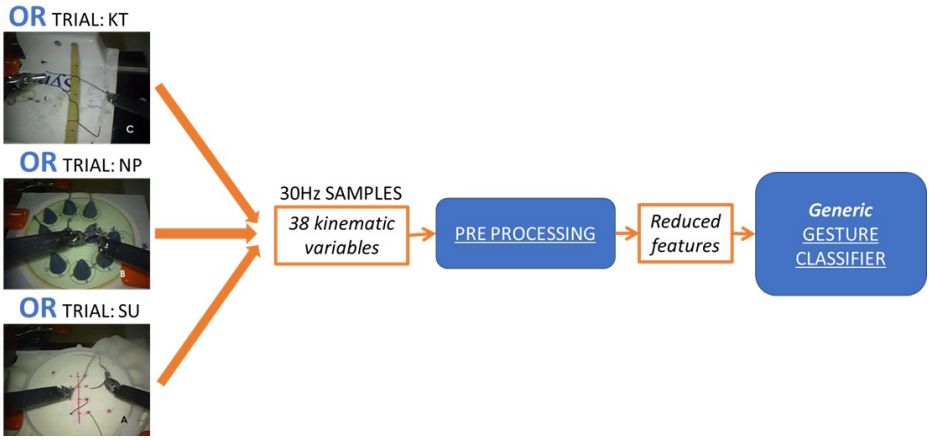

Figure 4.1: *Framework allowing gesture classification through generic gesture classifier. Surgical trials are collected and pre-processed. Reduced vectors are then classified into 1 out of 15 possible gestures, according to Table 3.2.*

Subsequently, the prepared data are input to the general model which performs real time gesture classification assigning labels according to Table 3.2.

4.1.1 Parameters

To study the performances of the Generic Gesture Classifier (GGC) it is possible to define two different approach: in the first one, after a concatenation of P samples, features are standardized following Equation (3.30). On the other hand, in the second approach, features are not standardized. It is possible to find a more detailed explanation about a possible standardization in Section 7.3.1, to be more synthetic, here, only the standardized case will be presented since it proved to be more considerable.

- Overall consolidated accuracy: By optimizing the variation of the *consolidated accuracy* over the JIGSAWS tasks as a function of P and *dim* parameters, it is possible to define a model.

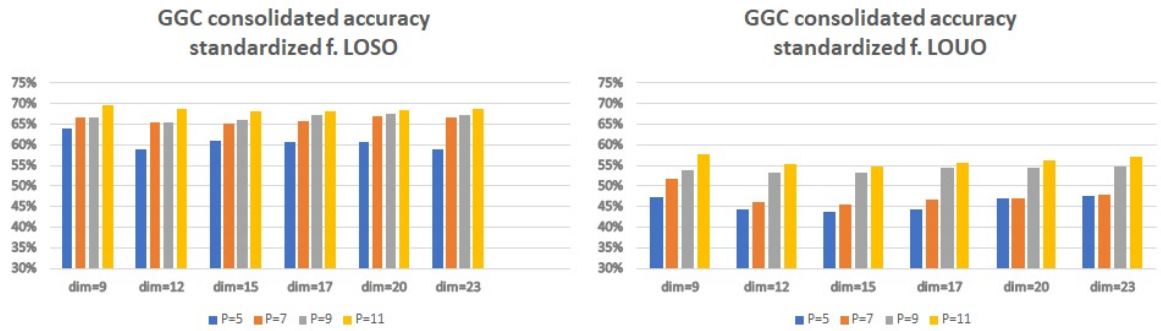

Figure 4.2: *Consolidated accuracy over all the three tasks, studied for every combination of concatenation length (P) and number of features (dim) on the LOSO and LOUO validation schema.*

By considering Figure 4.2, it is possible to see that the set of parameters which optimize the consolidated accuracy for both LOSO and LOUO cross-validation schema is $P = 11$ and $dim = 9$. It is important to report that the maximum value of consolidated accuracy is higher for the LOSO setup than for the LOUO.

- Unclassified sequences: In order to have a robust algorithm it is necessary to study the variation of the percentage of unclassified task as function of the P and *dim* parameters. This index gives an important clue to the final choice of the model parameters.

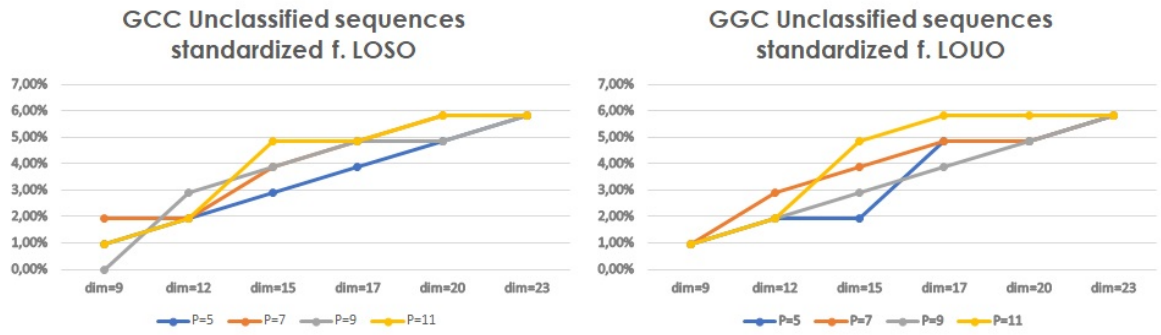

Figure 4.3: *Percentage of unclassified sequences, obtained by GGC considering different P and dim parameters.*

In particular, referring to Figure 4.3, it appears that for the LOSO cross validation schema a choice of parameters that maintains as lower as possible the percentage of unclassified sequences can be $P = 9$ and $dim = 9$, while for LOUO

scheme a good choice is once again $P = 11$ and $dim = 9$.

Considering both the two presented measures, the best possible set up is:

- LOSO: $P = 11$ and $dim = 9$, reporting that this choice penalizes a bit the percentage of unclassified sequences.
- LOUO: $P = 11$ and $dim = 9$

4.1.2 Performances

- Time: By considering the two set of parameters for LOSO and LOUO cross-validation schema, in Table 4.1 is presented the total computational time that takes to process a single frame. Since the algorithm should work in real time applications the required computational time cannot overpass the threshold defined considering the overlapped concatenation. This particular time threshold can be computed as the inverse of the sampling frequency divided by the concatenation value P, in particular the available time is given by:

$$available\ time = \frac{P}{30Hz} \tag{4.1}$$

	LOSO (P=11, dim=9)	LOUO (P=11, dim=9)
To classify 1 sample	0.092s	0.086 s
available time	0.367s	0.367s

Table 4.1: *Time spent by the GGC algorithm to classify one sample, considering the LOSO and LOUO validation scheme and the respective threshold time for real time applications.*

At it is confirmed in Table 4.1, the GGC algorithm respects the time constraints in both the two cross-validation approaches.

- Accuracy and relevance: Using the metrics defined in Section 3.6 the GGC algorithm can be evaluated in terms of micro/macro average accuracy, precision

and their respective standard deviations. In Table 4.2 the GGC algorithm is presented considering its performances over both the LOSO and LOUO validation scheme.

	LOSO (P=11, dim=9)			LOUO (P=11, dim=9)		
Evaluation	Suturing	Needle passing	Knot tying	Suturing	Needle passing	Knot tying
Micro (%)	79.28	55.94	73.52	63.45	43.29	66.08
Macro $\pm$ *std* (%)	76.95 ± 10.36	46.48 ± 26.83	64.12 ± 32.29	51.26 ± 16.63	36.48 ± 24.35	57.92 ± 19.82
Precision $\pm$ *std* (%)	54.33 ± 30.94	42.77 ± 29.72	41.96 ± 42.64	43.10 ± 33.24	33.47 ± 30.60	36.85 ± 40.21

Table 4.2: *Performances of the GGC, considering the evaluation metrics defined before (Section 3.6) computed over the LOSO and LOUO validation schema. The reported values are expressed as percentages.*

It is necessary to highlights that the overall performances of the GGC algorithm increases, if the LOSO cross-validation set up is considered.

Always considering LOSO and LOUO cross-validation schema, the percentage of unclassified sequences for the specific set of parameters previously defined is presented in Table 4.3. Considering both the two validation sets, less than 1% of the total number of trials is lost.

	LOSO (P=11, dim=9)	LOUO (P=11, dim=9)
Unclassified seq.	0.97%	0.97%

Table 4.3: *Percentage of unclassified trials, considering the LOSO and LOUO validation schema, for the particular set of parameters previously defined.*

4.1.3 Comparisons

Considering the micro average accuracy, performances of GCC algorithm can be compared with other approaches presented in literature (Table 4.4). In [38] a Markov and semi Markov Conditional Random Field (MsM-CRF) is proposed to classify gestures

in surgical tasks. The same purpose is accomplished in [13] thanks to the use of three different composite HMMs, one for each task in the JIGSAWS.

	LOSO			LOUO		
	MsM-CRF*	GMM-HMM**	GGC	MsM-CRF*	GMM-HMM**	GGC
Suturing	82.1%	82.2%	79.3%	72.6%	74.0%	63.5%
Needle passing	76.8%	70.6%	56.0%	57.1%	64.1%	43.3%
Knot tying	81.1%	80.1%	73.5%	68.8%	72.5%	66.1%

Table 4.4: *Comparison between state of art gesture classification micro accuracies, expressed as percentages. Results from: MsM-CRF* [38], GMM-HMM** [13], and the proposed GGC approach.*

As it clearly appears from Table 4.4 the gesture classification results achieved by the GGC algorithm are inferior with respect to the state of art. However, it is necessary to report that with respect to literature the GGC performances are completely achieved in real time.

- Statistical analysis: It is possible to compare more in detail the gesture classification results achieved with the proposed GGC approach with the one presented in [13] and referred to the composite GMM-HMM model.

	LOSO			LOUO		
	Suturing	Needle passing	Knot tying	Suturing	Needle passing	Knot tying
Macro p-value	5.89×10^{-14}	1.47×10^{-33}	1.08×10^{-32}	8.55×10^{-19}	3.01×10^{-34}	9.14×10^{-23}
Precision p-value	2.50×10^{-33}	2.54×10^{-34}	2.54×10^{-34}	6.14×10^{-33}	2.57×10^{-34}	2.54×10^{-34}

Table 4.5: *H_0 test results between the population achieved by using a GMM-HMM as proposed in [13] and the one obtained with proposed GGC approach.*

The hypothesis H_0: *the population coming from GMM-HMM algorithm belongs to the same distribution of the population achieved by using the GGC algorithm* is tested over every surgical procedure by using both macro average and precision metrics. In particular, in Table 4.5 are shown results of the test in terms of p-values. Considering the macro average and the precision over every kind of surgical task, it is possible to see that the proposed GGC algorithm works in a

different way with respect to the state of art. In Section 7.4.1 more details are furnished about the tested populations.

4.2 Task recognition and task-specific gesture classification

Figure 4.4 summarises how to perform a real time surgical gesture classification addressed by the task recognition. Kinematic data form the two PSMs of the dVSS are acquired at 30 Hz during generic surgical performances. These data are concatenated, possibly standardized and reduced to have samples composed of a limited number of features. Once data are pre-processed, the task recognition is performed online and the trial is addressed to the proper task-specific gesture classifier for the final segmentation and classification into gestures.

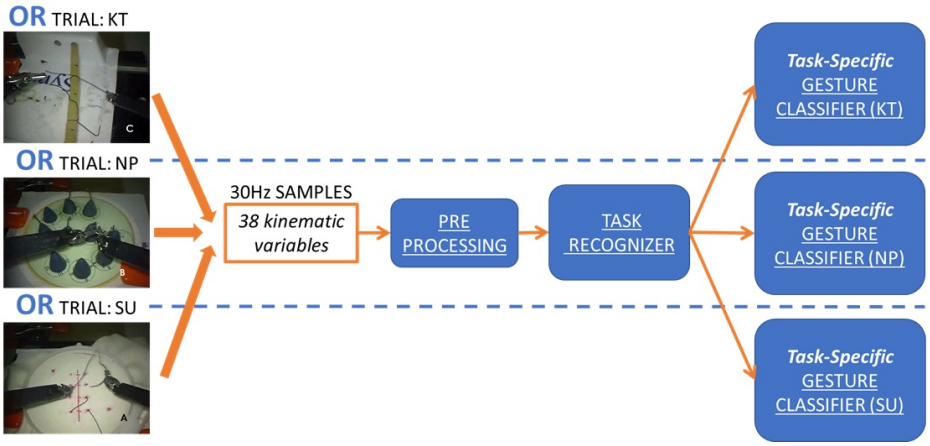

Figure 4.4: *Framework used to perform gesture classification after task recognition. Any of the surgical trials performed is collected and pre-processed, allowing task recognition over reduced data. The task recognizer feeds the correct task-specific gesture classifier which performs classification considering only possible gestures in that specific task Figure as described in Figure 3.2.*

In order to achieve the proposed framework, two disjointed systems have been created: one performing task recognition, the other classifying trials into gestures. The results coming from these two algorithms will be presented considering the choice of their constitutive parameters and performances of different used models also compared with the state of art.

4.2.1 Task-related task recognition

In order to recognize the undergoing surgical task performed by a surgeon, a Task-Related Task Recognizer (TRTR) is set up. In particular this algorithm performs a first data pre-processing and a following task classification through Viterbi process. Since it has to accomplish real time task recognition time performances as well as the final classification accuracies are key features to evaluate the overall approach.

4.2.1.1 Parameters

- Overall consolidated accuracy: The first element taken into consideration to tune model parameters is the consolidated accuracy. It changes by varying P and dim, and every possible combination of them is studied by considering the LOSO and LOUO validation schema.

Figure 4.5: *Performances of the TRTR are studied in terms of overall consolidated accuracy. This metrics computed through Equation* (3.33) *is studied for every combination of concatenation length (P) and number of features (dim) on the LOSO and LOUO validation schema.*

By considering the consolidated accuracy in Figure 4.5, it is possible to define two distinct sets of model parameters, one for LOSO and one for LOUO cross-validation scheme. For the LOSO scheme it seems that a good choice for the number of concatenated samples (P) and the number of considered features (dim) can be $P = 5, dim = 12$ *or* $dim = 15$ while for LOUO could be $P = 7, dim = 9$. From Figure 4.5 it is clear that the maximum peak in consolidate accuracy for LOSO and for LOUO is different: LOUO set up reaches a lower maximum level of consolidate accuracy meaning that its performances will be worse.

- Time performances: The task recognition should be completed online. Moreover the algorithm should classify the correct task as soon as possible to start the final gesture classification. This ability is measured in terms of percentage of the trial that is necessary to have to manage task recognition. Varying P and dim the real time performances of the TRTR changes, as it is possible to see in the consolidated graph, for every different cross-validation set up, in Figure 4.6.

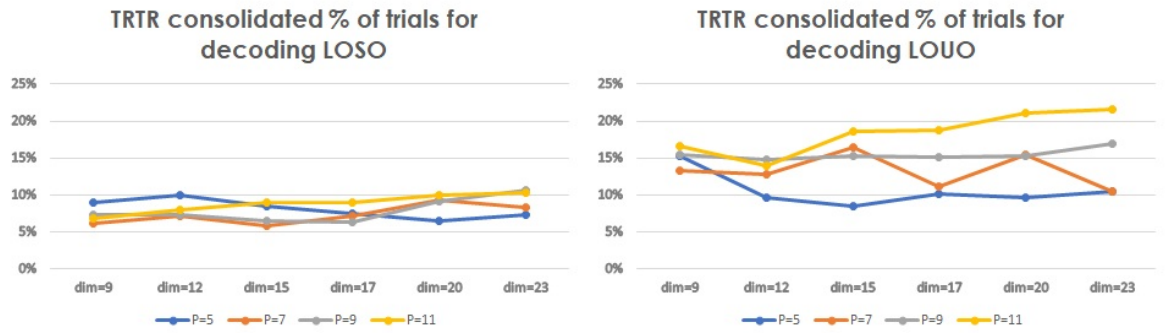

Figure 4.6: *Percentage of the samples of one trial that is necessary to have to accomplish task recognition, consolidated over the three surgical tasks and function of the P and dim parameters.*

From Figure 4.6 it appears that for LOSO schema, the choice of parameters does not affect the real time performances that much and the consolidated percentage of samples that it is necessary to have to infer the task is almost constant. On the other hand, for LOUO setup, the number of concatenated samples should be as lower as possible. For LOSO schema, the best choice would be to set $P = 7, dim = 15$, but taking into account the relative advantages of this choice and the loss in consolidated accuracy represented in Figure 4.5, P and dim will be maintained set as $P = 5, dim = 12$ *or* $dim = 15$. The best set of parameters for LOUO could be $P = 5, dim = 15$, however, considering the choices taken over the consolidated accuracy, it is better to maintain $P = 7, dim = 9$: the real time performances will decrease a bit, but the accuracy is maintained as high as possible.

- Unclassified sequences: As aforementioned, it is possible to consider robustness as a key factor to define the TRTR parameters.

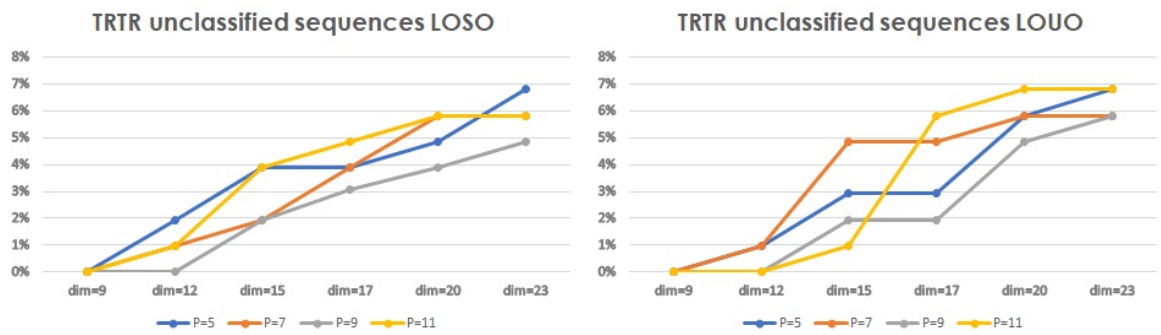

Figure 4.7: *Performances of the TRTR are studied in terms of percentage, of unclassified sequences.*

Considering Figure 4.7, it is clear that a low level of model complexity it is necessary for a good reliability. Thus the best choice for the LOSO setup, also caring for the previous consideration can be $P = 5, dim = 12$, while, for the LOUO cross-validation set, $P = 7, dim = 9$ can be the best choice. Thus the final choice of parameters is:

- LOSO: $P = 5$ and $dim = 12$
- LOUO: $P = 7$ and $dim = 9$

Reporting that these choices penalize the percentage of the trial that it is necessary to recognize the task.

4.2.1.2 Performances

- Time: In Table 4.6 is presented the computational time required to classify one sample and the available computational time allowed by the choice of the LOSO and LOUO parameters. Due to the overlapped concatenation, the latter is computed as described in Equation (4.1).

	LOSO (P=5, dim=12)	LOUO (P=7, dim=9)
To classify 1 sample	0.011s	0.009s
Available time	0.167s	0.233s

Table 4.6: *Time to classify a sample with TRTR algorithm taking in consideration LOSO and LOUO schemes and respective times for real time application.*

Since the task recognition is the first step of the framework in Figure 4.4, it has to classify as fast as possible the task in order to address gesture classification. In Table 4.7 is presented the percentage of trials that it is necessary to have for every task to accomplish task recognition using TRTR.

	LOSO (P=5, dim=12)	LOUO (P=7, dim=9)
Suturing	10.80%	13.04%
Needle Passing	15.81%	16.30%
Knot Tying	3.45%	10.58%

Table 4.7: *Percentage of samples of the trial necessary to recognize the surgical procedure using TRTR.*

In order to recognize the task from a trial belonging to one specific task, the TRTR needs to process form the 4% up to 16.3% of the undergoing task.

- Accuracy and relevance: In Table 4.8 the micro average accuracy of TRTR is shown. In particular, for every kind of surgical procedure the micro average accuracy is presented as a function of the LOSO and LOUO cross-validation setup.

	LOSO (P=5, dim=12)	LOUO (P=7, dim=9)
Suturing	100%	94.87%
Needle Passing	96.43%	85.71%
Knot Tying	97.22%	97.22%

Table 4.8: *TRTR micro accuracy, expressed as percentages, over the JIGSAWS surgical tasks.*

The percentage of unclassified trials over the LOSO and LOUO test set is presented in Table 4.9. As anticipated before this measure is essential to define the robustness of the TRTR it is fundamental to maintain this measure as low as possible.

	LOSO (P=5, dim=12)	LOUO (P=7, dim=9)
Unclassified seq.	1.94%	0%

Table 4.9: *Percentage of unclassified trials, considering the LOSO and LOUO validation schema, for the particular set of parameters previously defined (Section 4.2.1.1).*

4.2.1.3 Comparisons

In Table 4.10 the TRTR algorithm is compared with the state of the art. Considering the micro average accuracy, it is possible to compare the TRTR with two different task recognizers presented in [5] which are a standard HMM and a DTW-kNN.

	LOSO			LOUO		
	HMM*	DTW-kNN*	TRTR	HMM*	DTW-kNN*	TRTR
Suturing	96.4%	100%	**100%**	80.7%	84.6%	**94.9%**
Needle Passing	83.5%	89.3%	**96.4%**	80.8%	85.7%	**85.7%**
Knot tying	**97.3%**	97.2%	97.2%	90.9%	95.8%	**97.2%**

Table 4.10: *Comparison between the state of art task recognition micro average accuracies due to HMM* and DTW-kNN*, both defined in [5], and the proposed TRTR. The micro average accuracy is expressed as a percentage*

As it is possible to see from Table 4.10, the TRTR approach outperform the state of task in recognizing almost all the proposed surgical tasks.

It is possible to compare the time performances of the TRTR with the ones presented in [5] and refereed to the aforementioned DTW-kNN algorithm. In particular, in Table

4.11 is represented the percentage of the trial that is necessary to have to accomplish task recognition considering the two approaches.

	LOSO		LOUO	
	DTW-kNN*	TRTR	DTW-kNN*	TRTR
Suturing	12.0%	**10.8%**	20.0%	**13.0%**
Needle Passing	**15.0%**	15.8%	22.0%	**16.3%**
Knot tying	5%	**3.45%**	**8%**	10.6%

Table 4.11: *Comparison between the state of art percentages of the trial that is necessary to have to accomplish task recognition and ones needed form the proposed TRTR. DTW-kNN* proposed in [5].*

Table 4.11 summarizes that, in order to achieve the values of micro average accuracy presented in Table 4.10, the proposed TRTR approach is faster. The percentage of trial that it is necessary to have in order to permanently classify tasks is inferior.

4.2.2 Task-specific gesture classification

Thanks to the task recognizer the final gesture classification is addressed to a proper classifier which accomplishes segmentation into gestures and a subsequent gesture identification through the Viterbi algorithm. The Task-Specific Gesture Classifier (TSGC) described in Section 3.5, has to be able to perform real time gesture classification. Even in this case the algorithm should work online, thus, it has to compute every identification in time span defined by the sampling frequency of 30 Hz.

It is important to report that the TSGC algorithm is still considered a disjointed system from the TRTR. Because of that, the TSGC algorithm does not consider the time previously used by the TRTR to perform task recognition. In a ideal synergistic system in which the TRTR works together with TSGC it is required also to consider the impact of the task recognizer since its computational time affects the whole framework

presented in Figure 4.4.

4.2.2.1 Parameters

With the prior assumption that the two systems composing the framework presented in Figure 4.4 are disjointed, every model performs its own data pre-processing optimizing independently its P and *dim* parameters.
As for the GGC algorithm, even in the TSGC is possible to optimize the accuracy over gesture classifications as well as time performances by considering a possible standardization of kinematic variables. A more complete explanation about the possible advantages in performing features standardization over LOSO and LOUO validation schema is provided in Section 7.3.2, here, to be more synthetic, only the optimal case of having standardized variables is presented.

- Overall consolidated accuracy: In Figure 4.8 it is possible to see the consolidated gesture classification accuracy over three surgical tasks (SU,NP,KT) as a function of the models parameters and validation scheme.

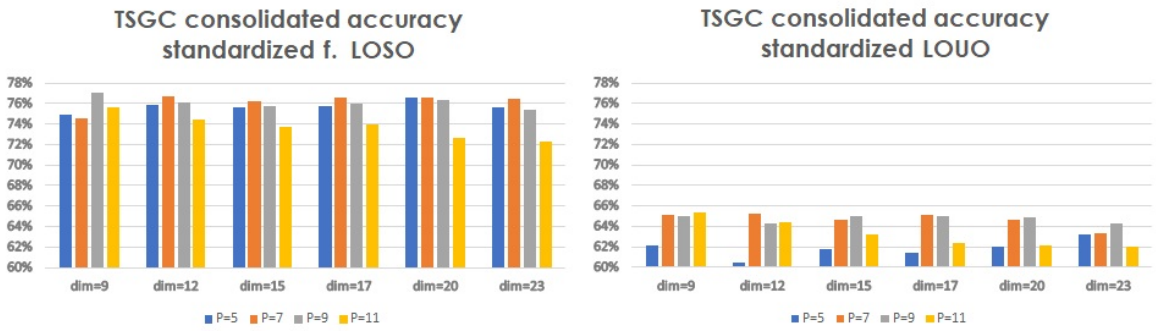

Figure 4.8: *Consolidated accuracy results, over SU,NP and KT surgical tasks, obtained by TSGC considering different P and dim parameters as well as the LOSO and LOUO validation schema.*

In order to maintain gesture classifier micro average accuracy high in surgical tasks, the best set parameters identified for the LOSO set up is $P = 9$ and $dim = 9$, while, for the LOUO schema an optimal choice is to set $P = 11$ and $dim = 9$. As for the GGC algorithm, even using TSGC, it has to been reported that the maximum level of consolidated accuracy is reach for the LOSO setup.

- Unclassified sequences: In Figure 4.9 appears that in order to have consolidated percentages of unclassified sequences over all the analysed surgical tasks as low as possible, the system complexity should be small. In particular, for the LOSO cross-validation schema $P = 9$ and $dim = 9$ appear to be the best choices, while for the LOUO setup the combination is $P = 5$ *or* $P = 7$ and $dim = 9$.

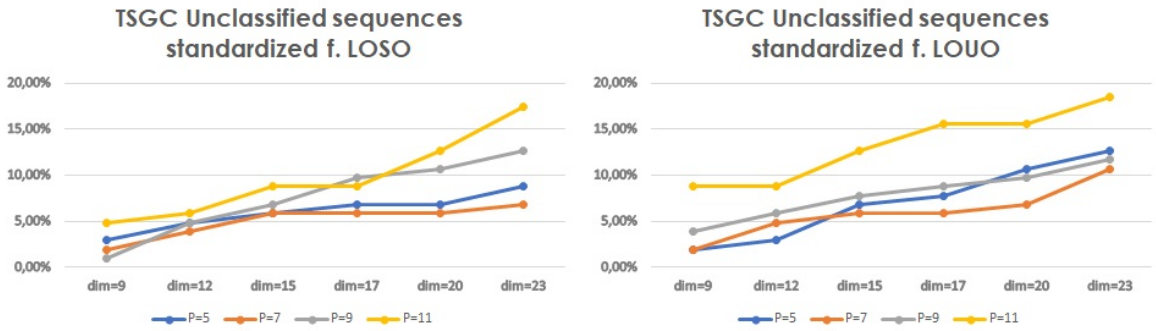

Figure 4.9: *Percentage of unclassified sequences, obtained by TSGC considering different P and dim parameters.*

Considering both the overall accuracy and the percentage of unclassified sequences, the best configuration is:

- LOSO: $P = 9$ and $dim = 9$
- LOUO: $P = 7$ and $dim = 9$

4.2.2.2 Performances

- Time: In Table 4.12 it is shown the total computational time required to process and classify a single frame. Considering possible real time application for the TSGC algorithm, the computational time required to process each sample should remain below the available time span, computed through Equation (4.1).

	LOSO (P=9, dim=9)	LOUO (P=7, dim=9)
To classify 1 sample	0.061s	0.063s
Available time	0.300s	0.233s

Table 4.12: *Time to classify a sample with the TSGC, and the respective available time span for real time applications.*

At it is confirmed in Table 4.1, the TSGC algorithm respects the time constraints in both the two cross-validation approaches.

- Accuracy and relevance: In Table 4.13 performances of the TSGC algorithm are presented following the metrics explained in Section 3.6 over the LOSO and LOUO validation schema.

	LOSO (P=9, dim=9)			LOUO (P=7, dim=9)		
Evaluation	Suturing	Needle passing	Knot tying	Suturing	Needle passing	Knot tying
Micro (%)	82.05	68.40	80.86	64.66	52.12	80.89
Macro $\pm$ *std* (%)	80.75 $\pm$ 8.71	50.74 $\pm$ 23.13	76.46 $\pm$ 15.09	52.57 $\pm$ 17.42	39.24 $\pm$ 23.12	79.76 $\pm$ 12.61
Precision $\pm$ *std* (%)	79.01 $\pm$ 17.89	66.35 $\pm$ 22.62	83.46 $\pm$ 11.71	65.60 $\pm$ 27.73	54.39 $\pm$ 25.55	81.53 $\pm$ 10.18

Table 4.13: *Performances of the TSGC, considering the evaluation metrics defined before (Section 3.6).The reported values are expressed as percentages.*

Considering the performances presented in Table 4.13 it is clear how the performances of the TSGC algorithm decrease in classifying gestures belonging to suturing and to needle passing trials, for the LOUO cross-validation schema.

The TSGC robustness in decoding sequences is also described as a percentage unclassified sequences over the total number of trials in Table 4.14.

	LOSO (P=9, dim=9)	LOUO (P=7, dim=9)
Unclassified seq.	0.97%	1.94%

Table 4.14: *Percentage of unclassified trials, for the particular set of parameters previously defined.*

4.2.2.3 Comparisons

In Table 4.15, considering the micro average accuracy, performances of the TSGC algorithm are compared with the ones obtained through a composite GMM-HMM gesture classifier presented in [13] and the ones achieved by a MsM-CRF in [38].

	LOSO			LOUO		
	MsM-CRF*	GMM-HMM**	TSGC	MsM-CRF*	GMM-HMM**	TSGC
Suturing	82.1%	82.2%	82.0%	72.6%	74.0%	64.7%
Needle passing	76.8%	70.6%	68.4%	57.1%	64.1%	52.1%
Knot tying	81.1%	80.1%	80.9%	68.8%	72.5%	80.9%

Table 4.15: *Comparison between the state of art gesture classification micro average accuracies. Results from: MsM-CRF* [38], GMM-HMM** [13], and the proposed TSGC approach. The micro average accuracies are expressed as percentages.*

Considering micro accuracy results in Table 4.15 it is possible to see that, even if the proposed TSGC algorithm reaches the state of art results over LOSO validation schema, its performances considering the LOUO set up are inferior. However, it is necessary to report that with respect to literature the TSGC performances are completely measured in real time.

- Statistical analysis: It is possible to compare more in detail results achieved with the proposed TSGC with the ones presented in [13] and referred to a composite GMM-HMM model.

	LOSO			LOUO		
	Suturing	Needle passing	Knot tying	Suturing	Needle passing	Knot tying
Macro p-value	6.80×10^{-30}	4.40×10^{-30}	6.10×10^{-3}	3.71×10^{-9}	2.70×10^{-33}	2.17×10^{-32}
Precision p-value	6.49×10^{-15}	2.00×10^{-2}	2.19×10^{-5}	1.30×10^{-4}	8.14×10^{-12}	3.66×10^{-32}

Table 4.16: *H_0 test results between the population achieved by using a GMM-HMM in [13] and the one obtained with proposed TSGC approach.*

The hypothesis H_0: *the population coming from GMM-HMM algorithm belongs to the same distribution of the population achieved by using the TSGC algorithm* is tested over every kind of surgical procedure by using the macro average and precision metrics. In particular, in Table 4.16 results of the p-value test are shown. Considering the macro average measure as well as the precision over every kind of task, for both LOSO and LOUO cross-validation settings, H_0 is

rejected.

In Appendix 7.4.2 the box plots representing the tested populations are shown.

Chapter 5

DISCUSSION

The final aim of this dissertation is to characterize in real time different surgical procedures in order to making them understandable for robots. To attack this problem two different frameworks are presented: the first one, represented in Figure 4.1, tries to directly accomplish real time gesture classification without any information about which particular task is performed. The second one, represented in Figure 4.4, uses the knowledge on the undergoing task, autonomously acquired, to address a real time task-specific gesture classification.

In order to better describe and discuss the performances of the presented frameworks, every algorithm composing them will be considered separately. Moreover, each algorithm will be reviewed divided into sections, as previously done for Chapter 4.

- Parameters, every algorithm is optimized to work considering two different cross-validation setups called LOSO and LOUO (see Section 3.6). These two setups are optimized independently with respect to the variable P, which refers to the number of samples concatenated together in every overlapped frame, and *dim* that indicates the number of kinematic features considered. In particular, the first variable P gives a measure about the temporal context incorporated into features [12, 44], while *dim* defines the model complexity.

- Performances, are measured considering the computational time required by the

algorithm to process one sample, this is a fundamental property of every algorithm since they have to be applied in real time applications. Afterwards, by using the metrics defined in Section 3.6 the algorithm is evaluated. It is important to report that also the ability to recognize trials is considered and measured as the percentage of unclassified sequences over the total number of trials.

- Comparisons, every algorithm will be finally compared with the state of art.

5.1 Generic gesture classifier

The final aim of the GGC is to accomplish a real time gesture classification over the different surgical tasks presented in the JIGSAWS dataset. The algorithm works without considering any information about the task that is under performance, thus, it can be easily applied to real scenario in which surgeons change the surgical operation continuously.

It is possible to discuss the results obtained with the GGC approach showed in Section 4.1 considering the aforementioned division into sections.

- Parameters: it is possible to see from Figure 4.2 that the gesture classification micro accuracy, consolidated considering the three surgical tasks does not change that much as a function of P and *dim*. However, it seems that the slight change in consolidated micro accuracy highlights the tendency of models in favouring simple configurations (low number of features *dim*) with a high temporal content (high P).
  The necessity of having simple configurations is remarked also in Figure 4.3, where it is clear that to maintain an high robustness of the model the *dim* parameter must be maintained as low as possible.

- Performances: as mentioned before, considering that the final classification is computed over 50% overlapped frames constituted by $2P+1$ samples, the choice of P defines the available time span for real time gesture classifications through Equation (4.1). It is clear from Table 4.1 that the GGC algorithm respects time

constraints spending, for every real time classification, less than a quarter of the total available time.

Less than 1% of the trials results unclassifiable, reporting a good reliability of the algorithm (see Table 4.3).

Analysing Table 4.2 it is possible to see that the choice of the cross-validation setup affects the gesture classification micro accuracy over the three surgical tasks considered. In particular, the LOSO setup results in higher accuracies, explicable considering that it is less sensitive to different surgical execution styles. This depends on the fact that in every group of training just one session for every surgeon is hidden from the algorithm, allowing it to know all the different surgical styles. Considering macro average accuracy and precision, it is possible to state that all these values are low and sparse underlining an high variance in detecting positive rates for every class as well as in sensitivity.

- Comparisons: comparing the achieved gesture classification micro accuracies of the GGC algorithm with the ones presented in [13] and in [38](see Table 4.4), it is clear that the proposed algorithm is less effective: its accuracy is, in average, 10% lower than the state of art.

  Considering the statistical analysis computed over the hypothesis H_0: *the population coming from GMM-HMM [13] algorithm belongs to the same distribution of the population achieved by using the GGC algorithm*, for every task and both the macro average and precision metrics, is rejected (p-values of the test reported in Figure 4.5). Moreover, referring to box plots in Section 7.4.1 representing these populations, it is clear that the performances are not only different but also lower.

After all these considerations it is possible to say that the algorithm is able to accomplish online gesture classification over different surgical procedures with different levels of micro average accuracy. Even if the real time performances are interesting and not achieved in literature, the micro accuracy in classifying gestures reached by the algorithm is far for the state of art. A possible explanation can be that without any a priori assumption about the undergoing task, the classification into gestures can be ambiguous.

5.2 Task recognition and gesture classification

The framework presented in 4.4 aims to address a real time gesture classification thanks to a simultaneous task recognition. This performance identification must be able to address the undergoing surgical trial to a specific gesture classifier trained for that particular task. Thus, to accomplish real time gesture classification two disjoint models are designed: the first one to perform real time task recognition and the second one for the final task-specific gesture classification.

5.2.1 Task recognition

The goal of the first algorithm in the framework showed in Figure 4.4 is to accomplish real time task recognition with a high accuracy. Since the TRTR algorithm has to address a further gesture classification, it has to recognize permanently the undergoing task as soon as possible.

It is possible to discuss the results obtained with the TRTR algorithm showed in Section 4.2.1 considering the aforementioned division into sections.

- Parameters: in Figure 4.5,4.6 and 4.7 appears that the TRTR algorithm works better if the complexity of model is maintained low, with a low level of temporal context incorporated in each time frame. More in detail, it is clear that even if the percentage of samples that it is necessary to have to infer the task basically does not change by varying P and dim (see Figure 4.6), the consolidated micro accuracy as well as the percentage of unclassified sequences improve lowering the two parameters.
  This determines a first distinction from the aforementioned GGC algorithm, which needs to incorporate lot of temporal context in order to discriminate between gestures.

- Performances: as it is shown in Table 4.6 the time constraints defined by the choice of P are respected: the TRTR satisfies the real time threshold. The algorithm is able to process one sample in less than 0.02 seconds and with the

12% of the samples, in average, the whole trial is permanently recognised and addressed for further gesture classifications.

Once again it is necessary to highlight the difference in performances between the LOSO and the LOUO cross-validation settings over the two most challenging surgical task: suturing and needle passing. Indeed, as it appears from Table 4.7, classification micro accuracies decrease of about 10%. This can be explained by considering that suturing as well as needle passing involve complex actions which may vary a lot by using different surgical styles diversely evaluated by the two validation setups.

Less than 2% of the trials results unclassifiable, reporting a good reliability of the algorithm (see Table 4.9).

- Comparisons: comparing the task recognition micro accuracies achieved by the TRTR and the ones presented in [5] computed through a traditional HMM and a DTW-kNN algorithm, it is possible to define that the proposed TRTR approach outperforms the state of art. Due to a lack of comparable data in literature, it is not possible to accomplish a more detailed statistical analysis, however, the tendency in classification accuracies seems to confirm this idea. The state of art seems to be outperformed also considering the percentage of trial that it is necessary to have to accomplish a permanent classification of surgical tasks. The proposed TRTR approach requires less samples to understand the undergoing task, turning out to be faster.

Considering the discussed results, it is possible to confirm that the presented TRTR algorithm is able to perform real time task recognition with final classification micro average accuracy that overpass the actual state of art. Moreover, it is necessary to remark the quick response in permanently recognizing the undergoing task that will define the delay with which the subsequent gesture classifier will start. Indeed with about the 12% of samples of the undergoing trial, the user performance can be accurately addressed to the correct TSGC for the final gesture classification.

5.2.2 Task-specific gesture classification

The second system proposed in framework of Figure 4.4 aims to accomplish real time gesture classification considering also task-related information. Three different classifiers are trained accordingly to the task analysed. During a surgical performance anyone of them may activate according to the task identified accomplishing a real time task-specific gesture classification.

It is possible to discuss the results obtained by TSGC algorithm showed in Section 4.2.1 considering the aforementioned division into sections.

- Parameters: by considering Figure 4.8 it clear that the choice of P and *dim* parameters does not strongly affect the consolidated classification micro accuracy. However, as it is shown in Figure 4.9, the choice of model parameters is fundamental to optimize performances, making the model reliable. The model seems to be optimized when it encapsulates a medium level of temporal context with a low degree of complexity.

- Performances: considering the time performances of the TSGC algorithm presented in Table 4.12, it is possible to see that the choice of P does not compromise the real time execution of the model, indeed, the time spent to classify each sample respects widely the time constraints.
  Once again, the reliability of the TSGC algorithm is proved in Table 4.14 in which it is possible to see that less than the 2% of the total number of trials is lost during gesture classification.
  The evaluation metrics proposed in Section 3.6 are refereed to the TSGC algorithm in Table 4.13. It is necessary to report a drop in the classification micro accuracy results for suturing and needle passing procedures in the LOUO cross-validation set. As previously done it is possible to explain this decrease in performances due to the lack of different styles in the LOUO setup. Considering macro average and precision in classification it is possible to say that this values are increased with respect to the ones presented in Table 4.2 for the GGC algorithm. These values are not only increased, their variance is remarkably reduced

defining a more robust algorithm.

- Comparisons: referring to the Table 4.15, it is possible to say that the TSGC algorithm, especially for the LOSO setup, achieves the state of art micro average accuracy performances in gesture classification. The inferior results achieved by TSGC in suturing and needle passing for the LOUO setting, can be explained, by the different content in LOUO setup of various styles that seem to be fundamental to allow TSGC in detecting gestures in these complex procedures. The statistical analysis accomplished to test H_0: *the population coming from GMM-HMM [13] algorithm belongs to the same distribution of the population achieved by using the TSGC algorithm* with respect to macro average and precision metrics is presented in Table 4.16. Here it is possible to notice that the p-values reject the hypothesis with a considerable confidence. The performances of the TSGC, in terms of macro average and precision, are definitely different from the state of art as confirmed by the p-values, however, by considering also the box plots in Figure 7.13, 7.14,7.15 and 7.16 representing the tested populations, it is possible to say that these performances are not necessarily worse.

Regarding the discussed results achieved by the TSGC algorithm, it si possible to say that the aim of the model has been reached. The gesture classifier can work in real time maintaining constant the initial delay required for task recognition. This interesting time performances are still not achieved in literature, over the same dataset. The TSGC reaches good classification micro accuracies, similar but not higher than the state of art, denoting the need of future developments in this direction.

5.3 Final considerations

It is possible to compare the two frameworks proposed in this dissertation and represented in Figure 4.1 and 4.4.

	LOSO		LOUO	
	GGC	TRTR+TSGC	GGC	TRTR+TSGC
Suturing	79.3%	**82.0%**	**63.5%**	61.4%
Needle passing	56.0%	**66.0%**	43.3%	**44.7%**
Knot tying	73.5%	**78.7%**	66.1%	**78.7%**

Table 5.1: *Classification micro accuracies achieved by the framework composed by the GGC an the one composed by TRTR+TSGC.*

It is possible to compare this two approaches analysing their final classification micro accuracies as summarized in Table 5.1. The final micro average accuracy of the first framework is expressed as the final performance reached by the GGC algorithm, while the classification micro accuracy reached by the second approach is computed as the multiplication between the final micro accuracy reached by TRTR and the one achieved by TSGC. As it is possible to see, the TRTR-TSGC framework (represented in Figure 4.4) outperforms the GGC approach which tries to classify gestures without any information about the undergoing task. Moreover, the TRTR-TSGC framework outperforms the GGC one also in terms of macro average and precision, as it is possible to see from the box plots in Section 7.5 representing the generated populations.

In the end we can consider the best presented framework as the one composed by the TRTR and TSGC algorithms. This approach simultaneously provides the recognition of any task performed by users and a precise characterization of its internal gestures.

Chapter 6

CONCLUSIONS AND FUTURE WORK

In this dissertation two possible frameworks are presented to implement *simultaneous task recognizer and gesture classifier* based on composite Hidden Markov Models. In particular, it has been showed that in order to characterize each movement of a surgical task with a significant accuracy, it is necessary to model gestures and the relationship lying within them. As discussed before, a possible way to increase gesture classification performances can be addressing the classification using information related to a specific undergoing task.

This thesis proposed a framework that, in two steps, allows a real time task recognition which addresses an accurate segmentation and classification of gestures in performed surgical operations. Both the steps are executed online, while the surgeon is performing the operation. However, the two algorithms are still disjointed and they need a manual switching to complete gesture classification (See Section 6.1).
The whole system achieves similar accuracies in gesture classification to the state of art, but, in addition, it is able to generalize its classification to different tasks without knowing a priori what the user is going to perform. Moreover, the entire process works totally online allowing the implementation of further applications.

6.1 Future Developments

Even if the main goal of the dissertation has been reached, many future developments can be done to improve performances of the overall framework. In particular it is possible to identify some limits limits that can be overpassed:

- The final aim of having simultaneous task recognition and gesture classification has been treated with success by distinguishing two sub-problems: the online task recognition and the subsequent online gesture classification. This distinction leads to an important simplification of the problem and even if the final goal has been achieved thanks to it, this assumption has led to have two disjointed systems that need to communicate to have a final gesture classification. A further improvement can be done integrating and connecting these two main algorithms, allowing an automatic update of the final gesture classification in case of task recognition errors or in case of changes in task (i.e. the surgeon goes form suturing to knot tying in a continuous way).

- Future studies can done to improve the final classification accuracy of the TSGC algorithm, to finally outperform the state of art.

- As previously explained, the inference time performances of these algorithms are fundamental features to be considered in order to perform task recognition and gesture classification in real time. This puts some limits on the systems used to acquire data. A future work can be addressed to optimize the data pre-processing phase as well as the subsequent Viterbi classification stages. In other words, developments can be addressed to optimize and to improve the time performances of the whole framework in order to be less restrictive in the acquiring system choice.

- The presented model has been developed by considering the data provided by the JIGSAWS. Even if this dataset furnishes motion data from 3 different surgical tasks accomplished by 8 surgeons, it is too limited and fragmented to support more robust analysis. This limitation in the dataset does not limit the presented

framework, however a larger dataset could improve the accuracy allowing more general conclusions.

- Due to limitations imposed by the dataset it was not possible testing any machine learning approach implementing Neural Networks (NN). NN algorithms have proved to be very effective in gesture segmentation and classification: in particular it would be interesting to implement Long-Shot Term Memory NN (LSTM-NN) to compare final performances.

6.2 Possible applications

The proposed approach can be considered as a part of the so called *machine learning of human skills field*. This field have provided a new approach in the application of Robot-Assisted Minimally Invasive Surgery. The future possibility to have intelligent robots able to learn and to understand surgeons in operation rooms defines new scenarios enhancing surgeons activity performances with a great advantage for patients health.

It is possible to think different applications for such a new technology. In particular, a learning algorithm that allows robot to accomplish a real time recognition of surgical tasks, with different level of granularity, can be fundamental in automating surgical procedures and tools or, again, in evaluating surgeons to improve their training.

6.2.1 Automating surgical procedures and controls

Automating robotic surgical assistants is one of the main research fields in RAMIS. A key-point that these studies share is the learning algorithm: in order to interact with a surgeon, a robot should accurately understand what are the surgeons intentions. Indeed, only in this way robots can properly predict the surgeon's next movement helping him in real time along surgical procedures. The proposed approach enables the real time segmentation and the characterization of every single movement accomplished by

surgeons on a different scale, providing the starting point for a proper interaction.

Many works address robots ability in understanding surgeons movements to automating elementary time-consuming surgical tasks or parts of them. In this way the surgeons workload as well as surgical costs are notably reduced: all tedious and time consuming basic operations can be autonomously executed by robots improving the surgical outcome [6, 24].

Also the surgical environment can improve thanks to the use of intelligent technologies. Supporting tools as endoscopic cameras, monitors as well as surgical bed or surgical instruments are increasing everyday their importance in hospital scenarios. However, only by automatizing them using learning algorithms with different level of granularity it is possible to define a synergistic environment.

6.2.2 Skill assessment

Nowadays, despite all the advantages in technology, the surgical training is still based on subjective evaluation carried out by expert who observe trainees during surgical performances and subsequently provide their judgement. Because of its subjective nature this evaluation is limited in consistency and reliability. In addition, these procedures are costly and not completely targeted to what trainees eventually would really need [5].

By using machine learning techniques applied to surgical robots, it is possible to exploit the collected data to perform a more robust and accurate analysis of the surgical trials executed by trainee surgeons. Considering that, machine learning algorithms, as the ones proposed here, are able to model surgical trials performed by specific users. Once the trainee model is defined, it can be compared with one built on an expert to have a valid and objective skill estimation [25].
Always comparing models, applications can be focused on detecting and identifying movement not perfectly accomplished. Trainees can understand in real time where they failed and, thanks to this identification a targeted training curricula can be ad-

dressed in order to fill specific skill gaps.

Bibliography

[1] V. R. Fuchs, "New priorities for future biomedical innovations," *New England Journal of Medicine*, vol. 363, no. 8, pp. 704–706, 2010.

[2] A. R. Lanfranco, A. E. Castellanos, J. P. Desai, and W. C. Meyers, "Robotic surgery: a current perspective," *Annals of surgery*, vol. 239, no. 1, p. 14, 2004.

[3] J. Van Den Berg, S. Miller, D. Duckworth, H. Hu, A. Wan, X.-Y. Fu, K. Goldberg, and P. Abbeel, "Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 2074–2081, IEEE, 2010.

[4] H. Mayer, F. Gomez, D. Wierstra, I. Nagy, A. Knoll, and J. Schmidhuber, "A system for robotic heart surgery that learns to tie knots using recurrent neural networks," *Advanced Robotics*, vol. 22, no. 13-14, pp. 1521–1537, 2008.

[5] M. Jahanbani Fard, "Computational modeling approaches for task analysis in robotic-assisted surgery," 2016.

[6] A. Murali, S. Sen, B. Kehoe, A. Garg, S. McFarland, S. Patil, W. D. Boyd, S. Lim, P. Abbeel, and K. Goldberg, "Learning by observation for surgical subtasks: Multilateral cutting of 3d viscoelastic and 2d orthotropic tissue phantoms," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 1202–1209, IEEE, 2015.

[7] S. Di Maio and C. Hasser, "The da vinci research interface," *Miccai workshop on systems and architectures for computer assisted interventions*, 2008.

[8] G. S. Guthart and J. K. Salisbury, "The intuitive telesurgery system: overview and application," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 1, pp. 618–621, IEEE, 2000.

[9] Y. Gao, S. S. Vedula, C. E. Reiley, N. Ahmidi, B. Varadarajan, H. C. Lin, L. Tao, L. Zappella, B. Béjar, D. D. Yuh, *et al.*, "Jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling," in *MICCAI Workshop: M2CAI*, vol. 3, 2014.

[10] C. E. Reiley and G. D. Hager, "Task versus subtask surgical skill evaluation of robotic minimally invasive surgery," in *International conference on medical image computing and computer-assisted intervention*, pp. 435–442, Springer, 2009.

[11] H. C. Lin, I. Shafran, D. Yuh, and G. D. Hager, "Towards automatic skill evaluation: Detection and segmentation of robot-assisted surgical motions," *Computer Aided Surgery*, vol. 11, no. 5, pp. 220–230, 2006.

[12] B. Varadarajan, C. Reiley, H. Lin, S. Khudanpur, and G. Hager, "Data-derived models for segmentation with application to surgical assessment and training," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 426–434, Springer, 2009.

[13] N. Ahmidi, L. Tao, S. Sefati, Y. Gao, C. Lea, B. B. Haro, L. Zappella, S. Khudanpur, R. Vidal, and G. D. Hager, "A dataset and benchmarks for segmentation and recognition of gestures in robotic surgery," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 9, pp. 2025–2041, 2017.

[14] L. Rabiner and B. Juang, "An introduction to midden markov models," *ieee assp magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[15] M. J. Mack, "Minimally invasive and robotic surgery," *Jama*, vol. 285, no. 5, pp. 568–572, 2001.

[16] J. H. Palep, "Robotic assisted minimally invasive surgery," *Journal of Minimal Access Surgery*, vol. 5, no. 1, p. 1, 2009.

[17] I. Surgical, "davinci surgery." `http://http://www.davincisurgery.com/da-vinci-surgery/da-vinci-surgical-system/`, 2018. [Online; accessed 3-March-2018].

[18] S. I. Decisions, "Cadth technology report," 2011.

[19] D. Sánchez, M. Tentori, and J. Favela, "Activity recognition for the smart hospital," *IEEE intelligent systems*, vol. 23, no. 2, 2008.

[20] G. Singla, D. J. Cook, and M. Schmitter-Edgecombe, "Incorporating temporal reasoning into activity recognition for smart home residents," in *Proceedings of the AAAI workshop on spatial and temporal reasoning*, pp. 53–61, 2008.

[21] T. Blum, N. Padoy, H. Feußner, and N. Navab, "Modeling and online recognition of surgical phases using hidden markov models," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 627–635, Springer, 2008.

[22] A. P. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. de Mathelin, and N. Padoy, "Endonet: A deep architecture for recognition tasks on laparoscopic videos," *IEEE transactions on medical imaging*, vol. 36, no. 1, pp. 86–97, 2017.

[23] X. Li, Y. Zhang, I. Marsic, A. Sarcevic, and R. S. Burd, "Deep learning for rfid-based activity recognition," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, pp. 164–175, ACM, 2016.

[24] C. E. Reiley, E. Plaku, and G. D. Hager, "Motion generation of robotic surgical tasks: Learning from expert demonstrations," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pp. 967–970, IEEE, 2010.

[25] J. Rosen, J. D. Brown, L. Chang, M. N. Sinanan, and B. Hannaford, "Generalized approach for modeling minimally invasive surgery as a stochastic process using a discrete markov model," *IEEE Transactions on Biomedical engineering*, vol. 53, no. 3, pp. 399–413, 2006.

[26] J. Rosen, J. D. Brown, M. Barreca, L. Chang, B. Hannaford, and M. Sinanan, "The blue dragon-a system for monitoring the kinematics and the dynamics of endoscopic tools in minimally invasive surgery for objective laparoscopic skill assessment," *Studies in health technology and informatics*, pp. 412–418, 2002.

[27] G. Megali, S. Sinigaglia, O. Tonet, and P. Dario, "Modelling and evaluation of surgical performance using hidden markov models," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 10, pp. 1911–1919, 2006.

[28] F. Petitjean, G. Forestier, G. I. Webb, A. E. Nicholson, Y. Chen, and E. Keogh, "Dynamic time warping averaging of time series allows faster and more accurate classification," in *Data Mining (ICDM), 2014 IEEE International Conference on*, pp. 470–479, IEEE, 2014.

[29] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "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.

[30] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[31] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Springer handbook of robotics*, pp. 1371–1394, Springer, 2008.

[32] Y. Yang, I. Saleemi, and M. Shah, "Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 7, pp. 1635–1648, 2013.

[33] C. Wu, J. Zhang, S. Savarese, and A. Saxena, "Watch-n-patch: Unsupervised understanding of actions and relations," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pp. 4362–4370, IEEE, 2015.

[34] S. Krishnan, A. Garg, S. Patil, C. Lea, G. Hager, P. Abbeel, and K. Goldberg, "Transition state clustering: Unsupervised surgical trajectory segmentation for robot learning," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1595–1618, 2017.

[35] A. Murali, A. Garg, S. Krishnan, F. T. Pokorny, P. Abbeel, T. Darrell, and K. Goldberg, "Tsc-dl: Unsupervised trajectory segmentation of multi-modal surgical demonstrations with deep learning," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 4150–4157, IEEE, 2016.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[37] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of human genetics*, vol. 7, no. 2, pp. 179–188, 1936.

[38] L. Tao, L. Zappella, G. D. Hager, and R. Vidal, "Surgical gesture segmentation and recognition," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 339–346, Springer, 2013.

[39] C. E. Reiley, H. C. Lin, B. Varadarajan, B. Vagvolgyi, S. Khudanpur, D. Yuh, and G. Hager, "Automatic recognition of surgical motions using statistical modeling for capturing variability," *Studies in health technology and informatics*, vol. 132, p. 396, 2008.

[40] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[41] B. Varadarajan, S. Khudanpur, and E. Dupoux, "Unsupervised learning of acoustic sub-word units," in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pp. 165–168, Association for Computational Linguistics, 2008.

[42] P. A. Gagniuc, *Markov Chains: From Theory to Implementation and Experimentation*. John Wiley & Sons, 2017.

[43] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, *et al.*, "The htk book," *Cambridge university engineering department*, vol. 3, p. 175, 2002.

[44] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

Chapter 7

APPENDIX

7.1 Acquisition system

The da Vinci Surgical System (dVSS) is a tele-robotic surgical platform released by Intuitive Surgical to enable surgeons to perform Robot Assisted Minimally Invasive Surgery (RAMIS) operations in different fields: from thoracic surgery to the urologic one. The dVSS has become a gold standard in RAMIS, allowing a precise control during surgical operations thanks to its high quality stereo viewer and its human robot interface (HRI) [17].

7.1.1 The robot hardware

The dVSS is a teleguided master-slave robot equipped with 3 robotic serial arms with 7-degree-of-freedom each, provided by Intuitive Surgical. It is made up of two sub-parts: the surgeon's console and the patient side cart.

The master-side console includes: two Master Tool Manipulators (MTMs), used by surgeons during performances, a High-Resolution Stereo Viewer (HRSV) displaying images from the endoscopic camera allowing a 3D vision of the operation and foot pedals to switch modality of work.

The patient-side robot is composed of three patient-side manipulators (PSMs) with 7 degree of freedom each, considering also the interchangeable tools surgeons place inside the patient's body. An Endoscopic CCD-Camera Manipulator(ECM) records and sends images to the stereo viewer (Figure 1.3) [8].

7.1.2 da Vinci API

The research interface (da Vinci API) [7] makes the robot internal data exchange accessible for further analysis, allowing real-time data sampling through Ethernet cable. The API interface of the dVSS is used to retrieve the camera pose, video data, kinematic data from both MTMs and PSMs and other noticeable system events. [9]. All these records are converted in a readable format and stored into the JIGSAWS working set.

7.2 HMM: Baum-Welch algoritm for set o training observations

If a set of training observations $O^r, 1 \leq r \leq R$ is used to tune the HMM, the iterative re-estimation of transition probabilities in A is accomplished through

$$\widehat{a}_{ij} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r-1} \alpha_i^r(t) a_{ij} b_j(o_{t+1}^r) \beta_j^r(t+1)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r} \alpha_i^r(t) \beta_i^r(t)} \tag{7.1}$$

in which $1 < i < N$ and $1 < j < N$. P_r is defined as the total probability of the r-th observation:

$$P_r = P(O^r|\lambda) \tag{7.2}$$

The transitions from the non-emitting start state are re-estimated by

$$\widehat{a}_{1j} = \frac{1}{R} \sum_{r=1}^{R} \frac{1}{P_r} \alpha_j^r(1) \beta_j^r(1), \ 1 < j < N \tag{7.3}$$

while transitions of the emitting states up to the stop non-emitting exit state are computed by

$$\widehat{a}_{iN} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \alpha_i^r(T) \beta_i^r(T)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{Tr} \alpha_i^r(t) \beta_i^r(t)}, \ 1 < i < N \tag{7.4}$$

$L_j(t)$, the likelihood of being in state j at time t is modified accordingly by considering d the particular r-th observation. Thus, it become Equation (7.5) that can be efficiently used to compute Equation (3.10) and Equation (3.11) in the overall tuning of the HMM parameters.

$$L_j^r(t) = \frac{1}{P_r} \alpha_j(t) \beta_j(t) \tag{7.5}$$

7.3 Standardization

In order to understand if the standardization of the kinematic variables is profitable to improve performances of the presented algorithms, it is possible to study how it impacts on the consolidated accuracy and on the unclassified sequences as function of P and *dim* parameters.

7.3.1 Generic gesture classifier

The standardization role is studied over LOSO and LOUO cross-validation sets.
In particular:

- LOSO

  1) Overall consolidated accuracy

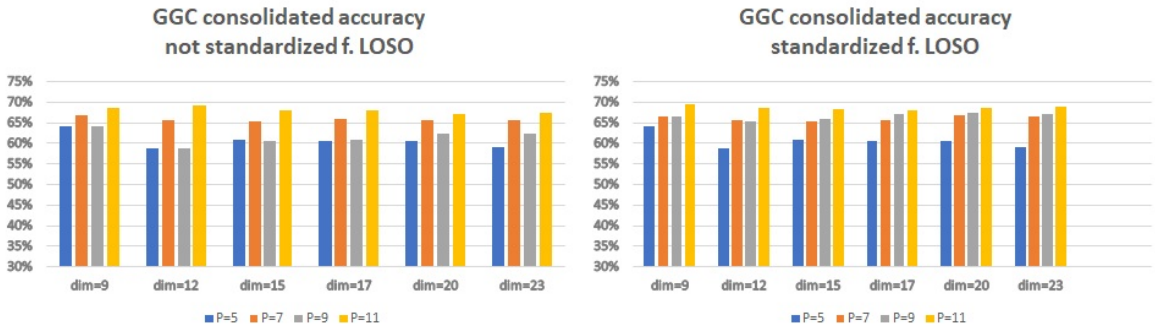

Figure 7.1: *GGC consolidated accuracy performances over LOSO test set, as function of standardization or not of the kinematic variables and P and dim parameters.*

  2)Unclassified sequences

Figure 7.2: *Percentage of unclassified sequences by GGC over LOSO test set, as function of standardization or not of the kinematic variables and P and dim parameters.*

As it appears from Figure 7.1 and Figure 7.2, the approach with standardized features improve the performances of the GGC algorithm both in terms of consolidated accuracy and unclassified sequences.

- LOUO 1) Overall consolidated accuracy

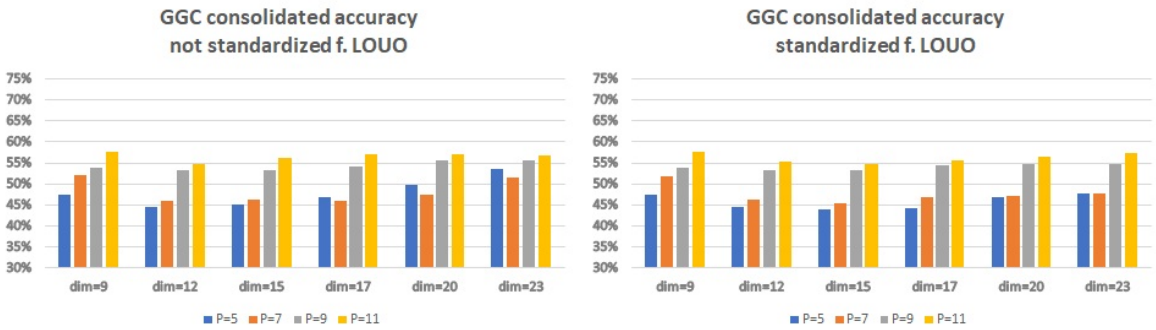

Figure 7.3: *GGC consolidated accuracy performances over LOUO test set, as function of standardization or not of the kinematic variables and P and dim parameters.*

2)Unclassified sequences

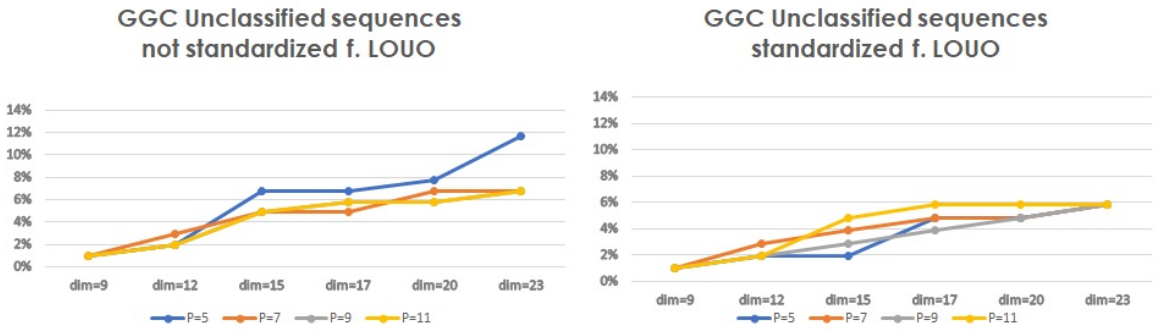

Figure 7.4: *Percentage of unclassified sequences by GGC over LOUO test set, as function of standardization or not of the kinematic variables and P and dim parameters.*

Also from the test represented in Figure 7.3 and 7.4 it appears clear that the use of standardized features increases performances.

7.3.2 Task-specific gesture classifier

The Standardization role is studied over LOSO and LOUO cross-validation sets.
In particular:

- LOSO

  1) Overall consolidated accuracy

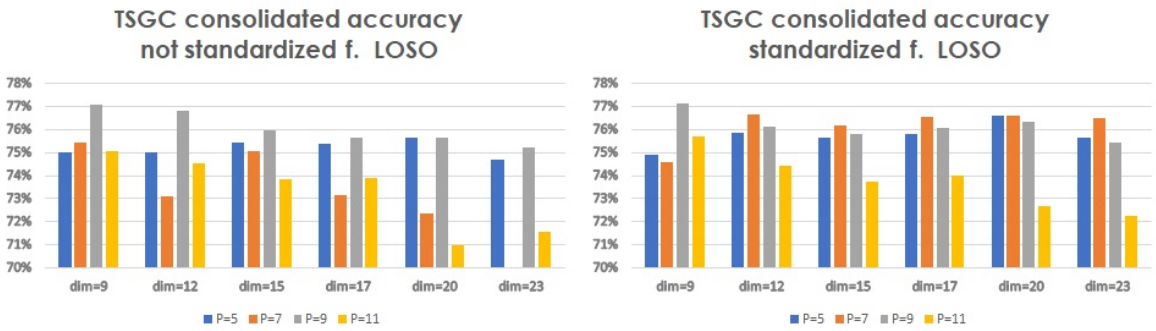

Figure 7.5: *TSGC consolidated accuracy performances over LOSO test set, as function of standardization or not of the kinematic variables and P and dim parameters.*

2)Unclassified sequences

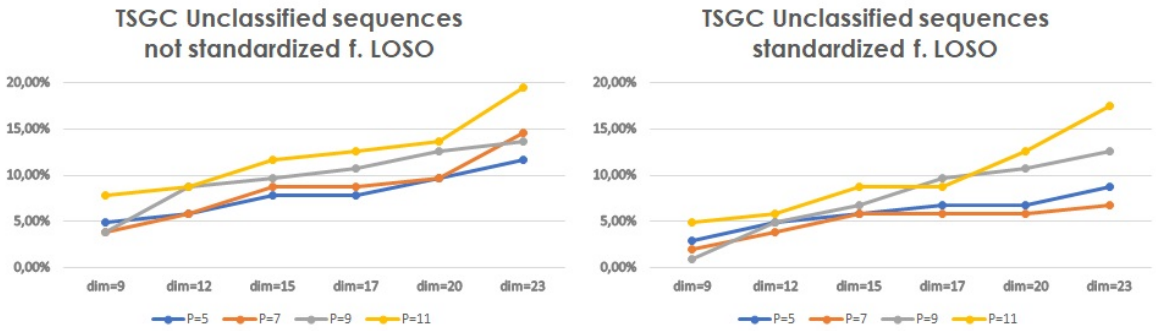

Figure 7.6: *Percentage of unclassified sequences by TSGC over LOSO test set, as function of standardization or not of the kinematic variables and P and dim parameters.*

As it is possible to see form Figure 7.5 and Figure 7.6, by adopting the standardization of the kinematic variables in the pre-processing stage both consolidated accuracy performances and percentages of unclassified sequences improve.

- LOUO

  1) Overall consolidated accuracy

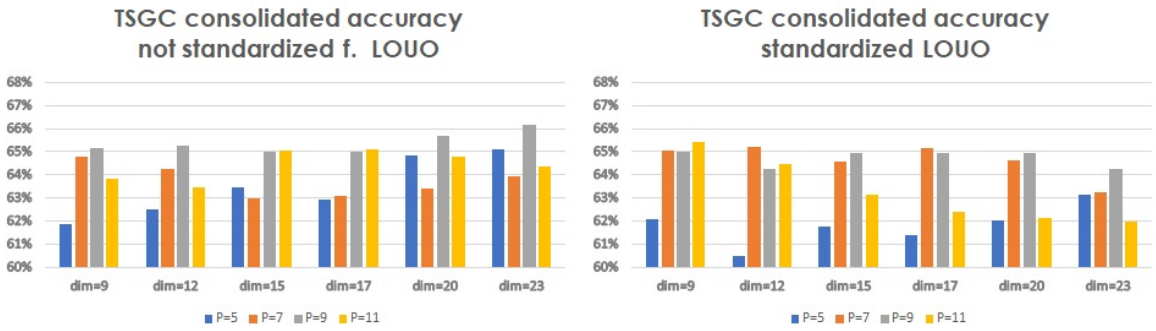

Figure 7.7: *TSGC consolidated accuracy performances over LOUO test set, as function of standardization or not of the kinematic variables and P and dim parameters.*

2)Unclassified sequences

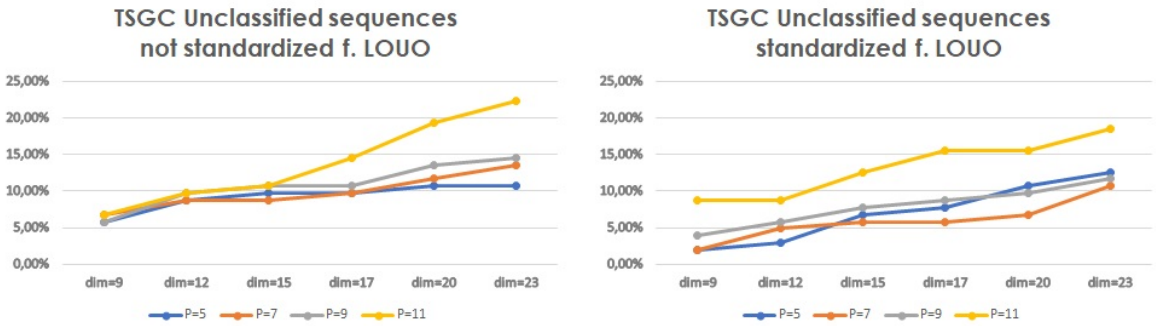

Figure 7.8: *Percentage of unclassified sequences by TSGC over LOUO test set, as function of standardization or not of the kinematic variables and P and dim parameters.*

As it is possible to see form Figure 7.7 and Figure 7.8, by adopting the standardization of the kinematic variables in the pre-processing stage both consolidated accuracy performances and percentages of unclassified sequences improve.

7.4 Literature comparisons

7.4.1 Generic gesture classifier

Considering both the cross-validation schema, performances of the GMM-HMM algorithm proposed in [13] and the ones of the GGC model are compared in terms of macro average accuracy and precision. The distances between the generated populations, represented in the following figures, are studied and compared considering the p-value over the Hypothesis, H_0:

the population coming from GMM-HMM algorithm belongs to the same distribution of the population achieved by using the GGC algorithm.

- LOSO set up

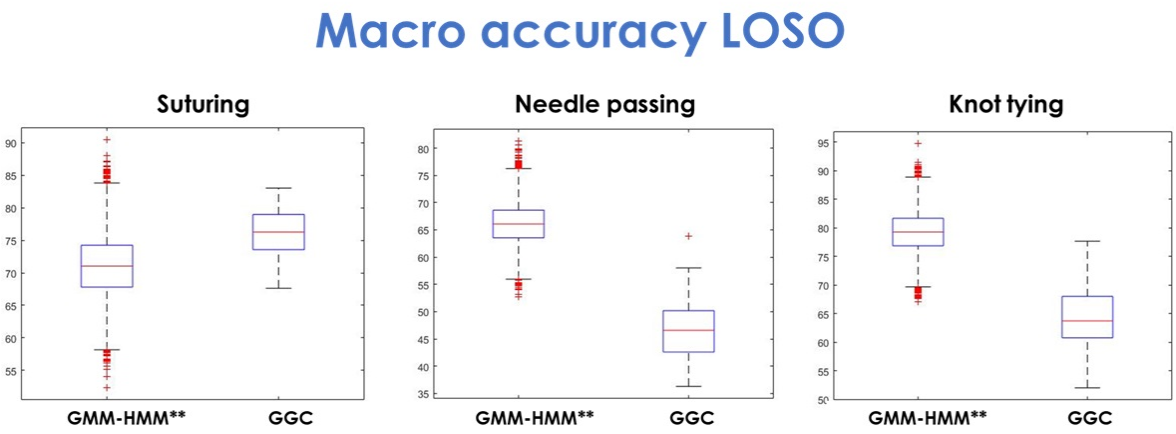

Figure 7.9: *Population resulting from the use of GMM-HMM** algorithm [13] and from the proposed GGC considering the macro average metrics.*

In Figure 7.9 populations are generated from the two algorithm considering macro average.

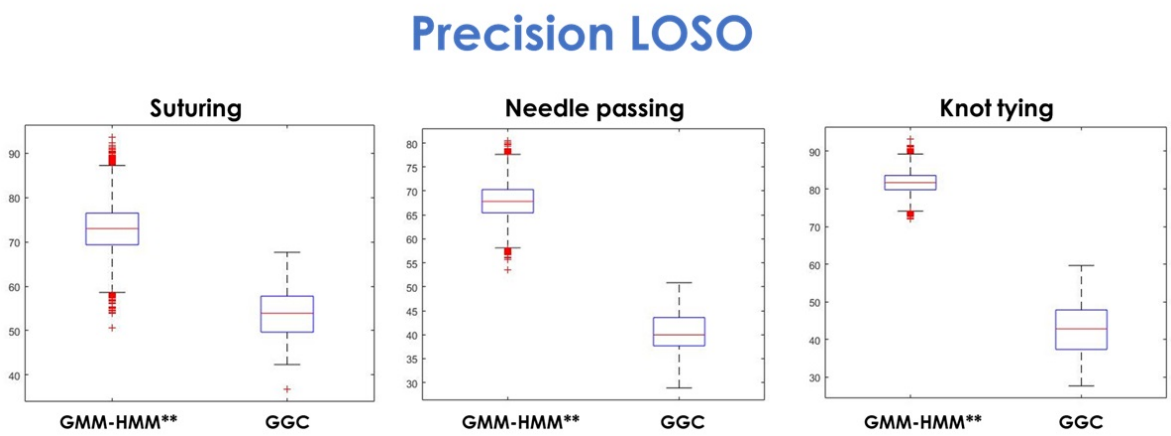

Figure 7.10: *Population resulting from the use of GMM-HMM** algorithm [13] and from the proposed GGC considering the precision metrics.*

In Figure 7.10 populations are dependent on the precision metrics.

- LOUO set up

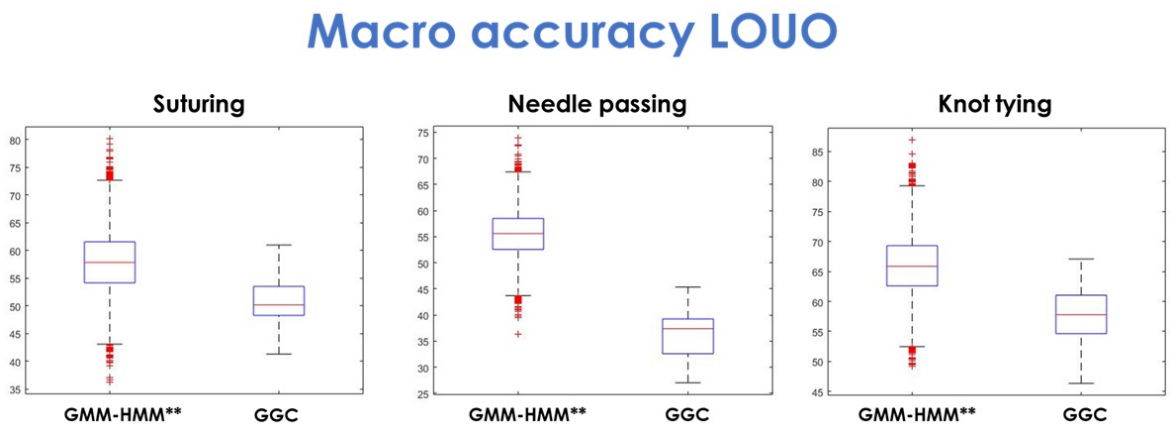

Figure 7.11: *Population resulting from the use of GMM-HMM** algorithm [13] and from the proposed GGC considering the macro average metrics. Both these algorithms consider a LOUO validation schema*

In Figure 7.11 populations are generated from the two algorithms considering macro average.

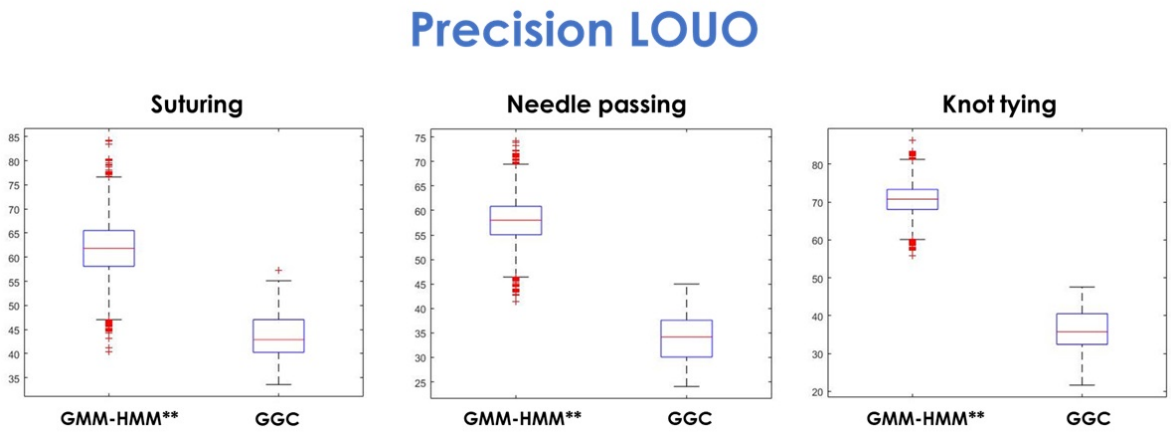

Figure 7.12: *Population resulting from the use of GMM-HMM** algorithm [13] and from the proposed GGC considering the precision metrics. Both these algorithms consider a LOUO validation schema*

In Figure 7.12 populations are dependent on the precision metrics.

7.4.2 Task-specific gesture classifier

By using the LOSO and the LOUO validation schema performances of the GMM-HMM algorithm proposed in [13] and the ones of the TSGC model are compared in terms of macro

average accuracy and precision. For every surgical task in the JIGSAWS dataset, each model generate populations according to the particular metrics analysed.

- LOSO set up

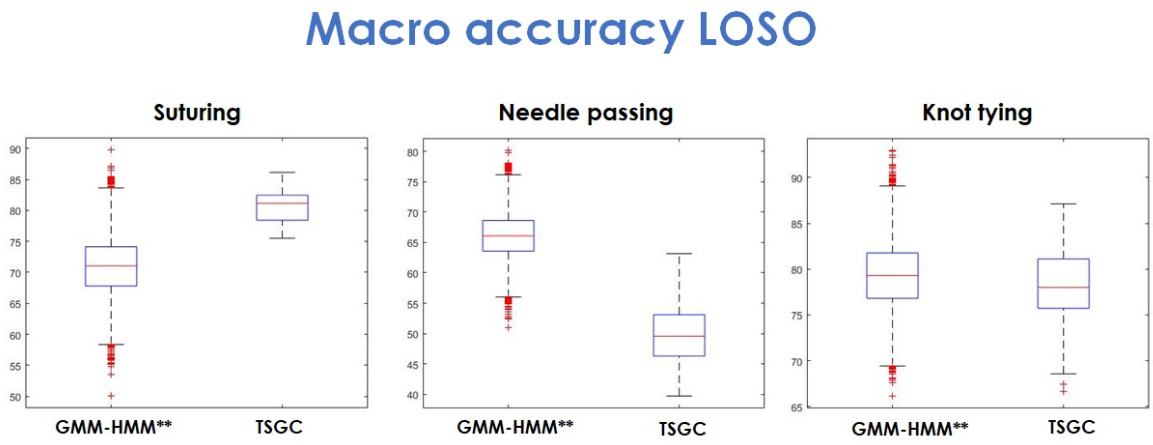

Figure 7.13: *Population resulting from the use of GMM-HMM** algorithm [13] and from the proposed TSGC considering the macro average metrics. Both these algorithms consider a LOSO validation schema*

In Figure 7.13 populations are generated from the two algorithm considering macro average.

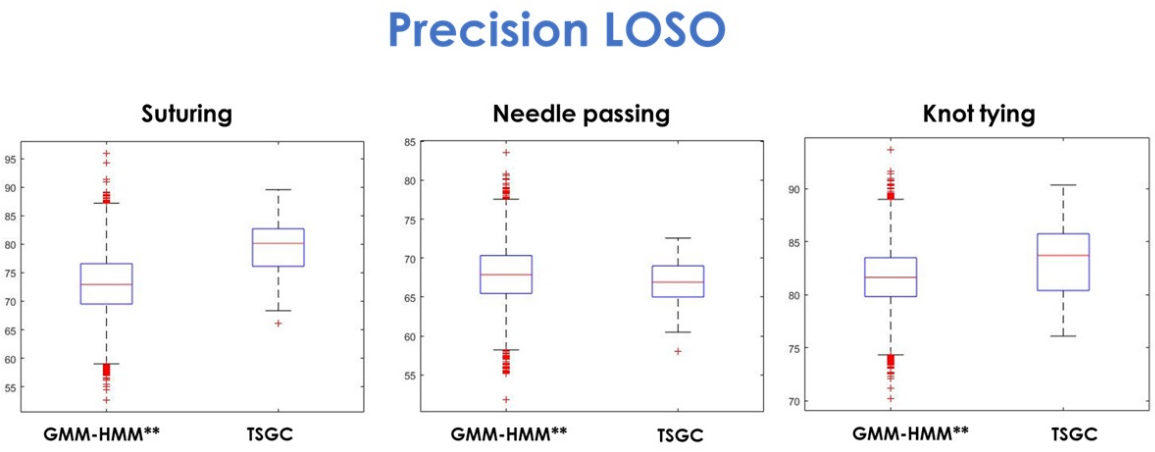

Figure 7.14: *Population resulting from the use of GMM-HMM** algorithm [13] and from the proposed TSGC considering the precision metrics. Both these algorithms consider a LOSO validation schema*

In Figure 7.14 populations are dependent on the precision metrics.

- LOUO set up

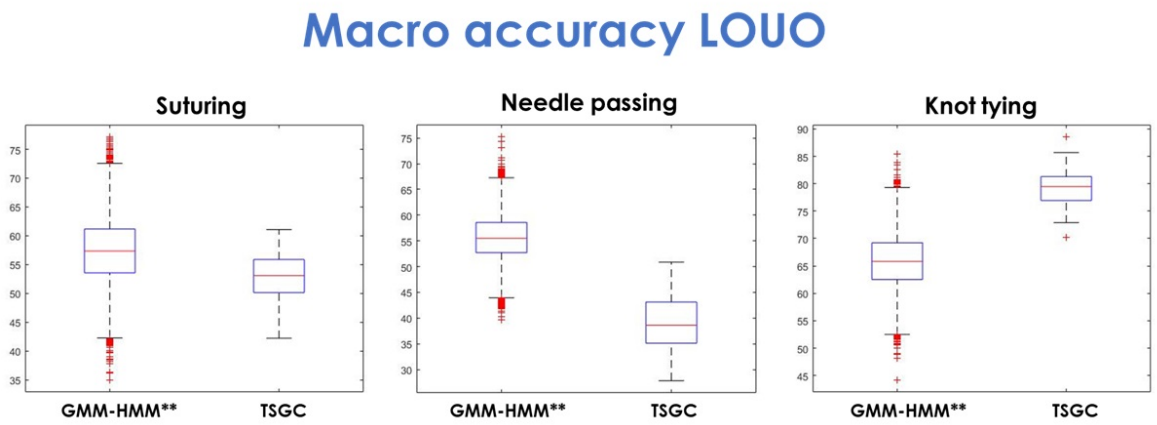

Figure 7.15: *Population resulting from the use of GMM-HMM** algorithm [13] and from the proposed TSGC considering the macro average metrics. Both these algorithms consider a LOUO validation schema*

In Figure 7.15 populations are generated from the two algorithm considering macro average.

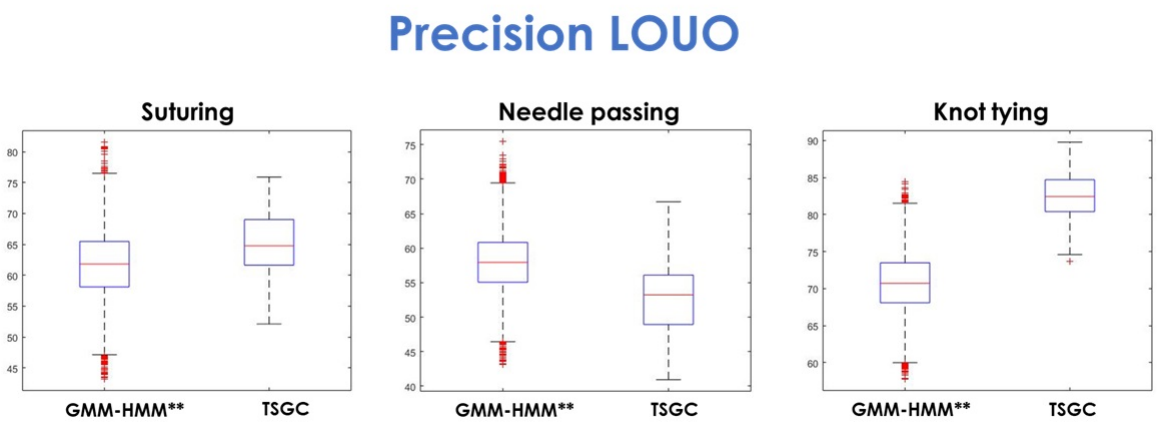

Figure 7.16: *Population resulting from the use of GMM-HMM** algorithm [13] and from the proposed TSGC considering the precision metrics. Both these algorithms consider a LOUO validation schema*

In Figure 7.16 populations are dependent on the precision metrics.

7.5 Final considerations

It is possible to compare the two proposed approaches, the one composed by the GGC algorithm and the one composed by the couple TRTR+TSGC in terms of generated populations with respect to micro average accuracy and precision.

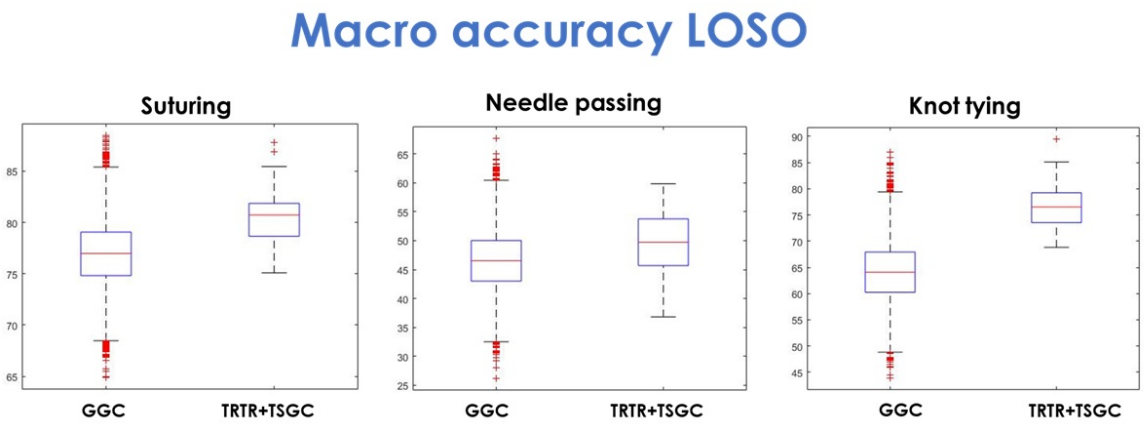

Figure 7.17: *comparison between the population generated by GGC and by TRTR+TSGC, considering micro average*

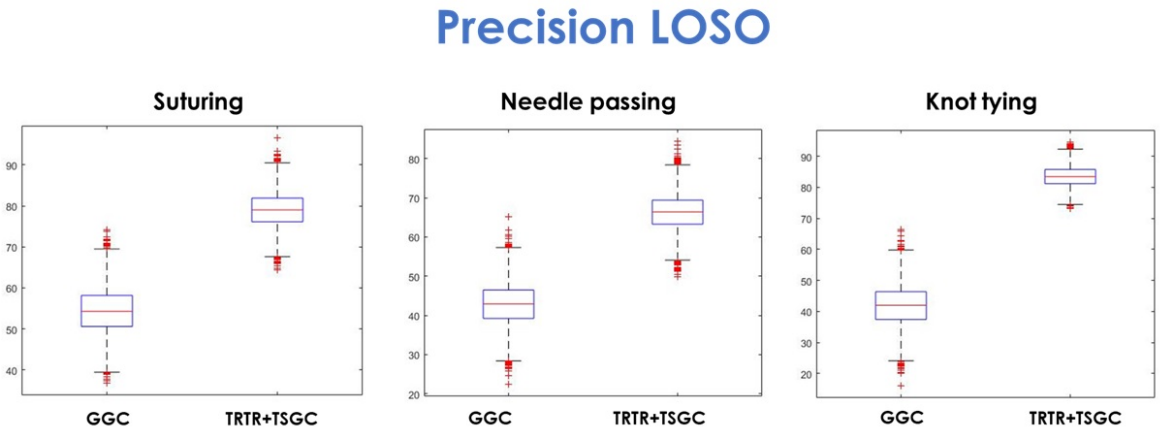

Figure 7.18: *comparison between the population generated by GGC and by TRTR+TSGC, considering precision*

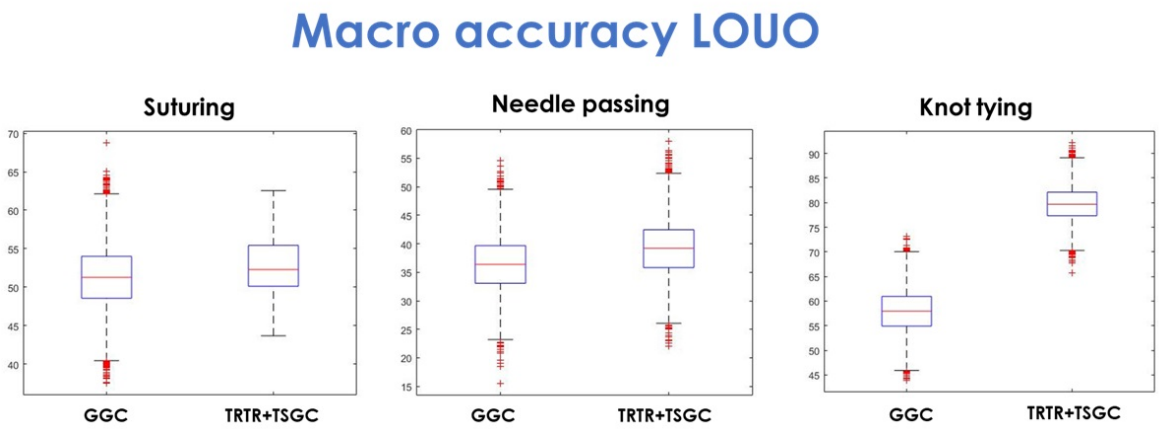

Figure 7.19: *comparison between the population generated by GGC and by TRTR+TSGC, considering micro average*

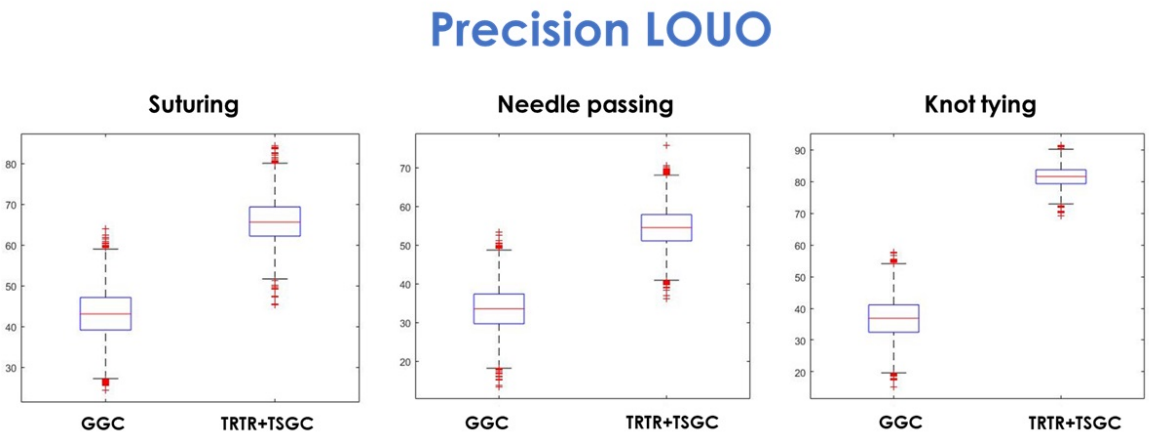

Figure 7.20: *comparison between the population generated by GGC and by TRTR+TSGC, considering precision*

As it appears in the figures above, the populations generated by the TRTR+TSGC framework clearly overpass the ones due to GGC, denoting a clear superiority in terms of macro average and precision.