Recognition of User's Activity for Adaptive Cooperative Assistance in robotic surgery*

Federico Nessi, Elisa Beretta, Giancarlo Ferrigno, Elena De Momi

Abstract— **During hands-on robotic surgery it is advisable to know *how* and *when* to provide the surgeon with different assistance levels with respect to the current performed activity. Gesteme-based on-line classification requires the definition of a complete set of primitives and the observation of large signal percentage. In this work an on-line, gesteme-free activity recognition method is addressed. The algorithm models the guidance forces and the resulting trajectory of the manipulator with 26 low-level components of a Gaussian Mixture Model (GMM). Temporal switching among the components is modeled with a Hidden Markov Model (HMM). Tests are performed in a simplified scenario over a pool of 5 non-surgeon users. Classification accuracy resulted higher than 89% after the observation of a 300 ms-long signal. Future work will address the use of the current detected activity to on-line trigger different strategies to control the manipulator and adapt the level of assistance.**

I. INTRODUCTION

Hands-on robotic surgery, during which the surgeon directly controls the manipulator movement by means of force application, is receiving greater acceptance both in terms of reliability and of assistant acceptability [1], [2]. Using a cooperatively-controlled manipulator, the surgeon is in charge of the procedure's workflow conduction [3] and can thus combine his/her decision making process and experience with advantages provided by the robot (e.g. hand tremor and fatigue reduction, etc.) [4]. Moreover, haptic-based controllers allows force feedback enhancement [5] or active constraints [6] increasing surgeon perception and improving the safety of the contact with soft tissues.

These features can best assist the operator during the execution of specific tasks. However, it is necessary that the system knows *how* and *when* to provide different degrees of assistance in order to get the best performance from the shared human-robot control in the procedure [7]. For example, highly compliant cooperative robots were proved to enhance targeting tasks on soft tissues, but have pointed out criticality in testing configuration due to possible unwanted interaction [8]. Thus, a robot able to adapt its behavior in response to the surgeon detected intention/activity can increase the safety of the cooperation. Furthermore, a manipulator that is able to recognize human's non-verbal cues in order to infer his/her intention can improve the intuitiveness of its usage [9].

*This work was supported by the FP7 ACTIVE project (FP7-ICT-2009-6-270460)

Federico Nessi, Elisa Beretta, Giancarlo Ferrigno and Elena De Momi are with Department of Electronics, Information and Bioengineering (DEIB), Politecnico di Milano, P.zza Leonardo da Vinci 32, 20133, Milan (Italy) {federico.nessi, elisa.beretta, giancarlo.ferrigno, elena.demomi}@polimi.it

In order to perform user's activity recognition, a well-known approach is based on Hidden Markov Models (HMMs) [10]. Successfully applied to speech recognition, HMMs are widely used also in handwriting [11] and gesture recognition [12], as well as motion classification. The basic assumption is that a motion action can be split into a set of primitives, and higher-level activities are a defined temporal series of those primitives.

Regarding hands-on robotic surgery, in [15] a feasibility study was presented to investigate the possibility to discriminate between activities performed using the JHU Steady-Hand robot, e.g. during peg-in-hole insertions, with a HMM-based classifier. Recognition of the primitives sequence was proved to be reliable (accuracy of classification higher than 85%) but runtime classification was not addressed.

HMM-based runtime motion classification algorithms were instead presented in the field of industrial cooperative robotics [14]. The classifier was able to model complex tasks, but proved to be reliable (i.e. more than 80% of correct classifications) only when observing a wide signal percentage (over 60%).

All the presented approaches are strongly affected by the ability to provide a complete set of primitives (or *gestemes*). A different approach was presented in [16] in the field of video surveillance. Using 2-D trajectories of passing-by pedestrians recorded by a camera system, low-level models (i.e. displacements) describing an activity (i.e. trajectory) are fitted with a Gaussian Mixture Model (GMM). This model is then used to cluster incoming data and the output is considered as the emission of a HMM that fully describes each performed activity.

In this work, we assess a strategy to detect surgeon's activities during cooperation with a surgical robotic assistant, exploiting a GMM-HMM based algorithm without the need to define gestemes. Because of the nature of the hands on cooperation, we model both 3-D driving forces and 3-D resulting trajectories of the manipulator. Furthermore, our algorithm is optimized for runtime recognition. The objective is to provide the intraoperative detection of the current activity, that can be exploited to automatically adapt the manipulator's dynamic behavior during human collaboration, improving safety and guidance feeling.

II. MATERIALS AND METHODS

A. Activity Model

During hands-on robotics, user's guidance can be described by the human driving forces ($\mathbf{f}$) and the resulting trajectory of the end-effector ($\mathbf{x}$). In particular, driv-

ing forces and end-effector trajectory can be expressed as vectors of n-samples over time, i.e. $\mathbf{f} = (\mathbf{f}^1, \ldots, \mathbf{f}^n)$ and $\mathbf{x} = (\mathbf{x}^1, \ldots, \mathbf{x}^n)$. Defining the user's current activity as a variable $a \in \{1, \ldots, A\}$, both $\mathbf{f}$ and $\mathbf{x}$ can be assumed to depend on the current activity. Thus, vectors $\mathbf{x}$ and $\mathbf{f}$ can be considered as a unique 6-dimensional vector $\mathbf{d}$ describing the user's action on the manipulator, i.e.

$$\mathbf{d} = (\mathbf{d}^1, \ldots, \mathbf{d}^n) = \begin{pmatrix} \mathbf{x} \\ \mathbf{f} \end{pmatrix}^T. \tag{1}$$

The use of both forces and end-effector trajectory is motivated by the fact that the combination of the two should be able to model user's activity that do not produce end-effector movement, e.g. exploitation of redundancy on a specific manipulator.

Following [16], the vector $\mathbf{d}$ is produced by a sequence of increments, with respect to the current action, i.e.

$$\mathbf{d}^t = \mathbf{d}^{t-1} + \Delta\mathbf{d}^t \tag{2}$$

and increments $\Delta\mathbf{d}^t$ at time t can be modeled as

$$\Delta\mathbf{d}^t = \mathbf{T}_{z_t} + \mathbf{C}_{z_t}^{1/2} \cdot \mathbf{w}_t \tag{3}$$

thus being the emission of a probability distribution (*low-level* model) labeled $z_t \in \{1, \ldots, M\}$ and characterized by the $\mathbf{T}_{z_t}$ mean and $\mathbf{C}_{z_t}$ covariance matrices. Vector $\mathbf{w}_t$ is the sample of a zero-mean and identity covariance Gaussian random vector, i.e. $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{I})$. Thus, model (2) is fully characterized by M low-level models describing the increments, defined by the $\mathbf{T} = (\mathbf{T}_1, \ldots, \mathbf{T}_M)$ and $\mathbf{C} = (\mathbf{C}_1, \ldots, \mathbf{C}_M)$ matrices and one at the time responsible of the emission of the current displacement $\Delta\mathbf{d}^t$.

Under this assumption, the vector of increments over time $\Delta\mathbf{d} = (\Delta\mathbf{d}^1, \ldots, \Delta\mathbf{d}^t)$ characteristic of each activity a is the sequence of emissions of low-level models $\mathbf{z} = (z_1, \ldots, z_t)$. This is equal to consider the $\Delta\mathbf{d}$ vector as a sample of a M-state continuous HMM in which the z-th state emission probability density is characterized by one of the $\mathbf{T}$ and $\mathbf{C}$ matrices. The transition matrix $\mathbf{B}_a$ of the HMM thus represents the model of the high-level user's activity a, i.e. $p(\mathbf{z}|a) = p(\mathbf{z}|\mathbf{B}_a)$.

B. Training

Data used in the training process is composed by a set of trajectories and guidance forces for all the activities that need to be described. Each couple of signals is labeled, thus it is know *a priori* the activity that have generated each data.

Following again the approach presented in [16], we estimate each low-level model z directly from the $\Delta\mathbf{d}$ computed over the complete set of training data, i.e. from training trajectories and forces of all the considered activities, as

$$\Delta\widehat{\mathbf{d}}^t = \mathbf{d}^t - \mathbf{d}^{t-1} \tag{4}$$

Each element in the $\Delta\widehat{\mathbf{d}}$ is considered as a sample of a M-mixtures GMM $\boldsymbol{\theta}$ in which the emission density parameters $\widehat{\mathbf{T}}_z$ and $\widehat{\mathbf{C}}_z$ of each component z estimates one of the $\mathbf{T}$ and $\mathbf{C}$ matrices. The optimal M number of components for the mixture model (i.e. the number of low-level models) is computed in an unsupervised way using the algorithm presented in [17]. The method is able to suppress redundant components starting from an initialization with an over-estimated number of gaussians.

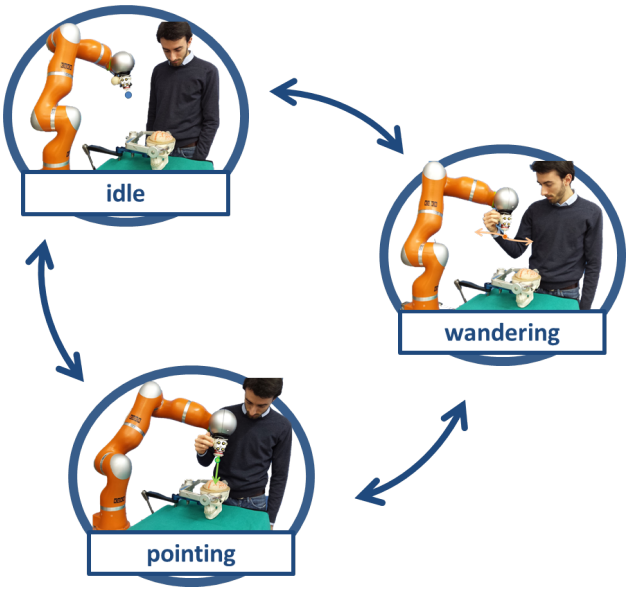

Fig. 1. Example of a user performing a mapping procedure over a brain phantom with the help of a hands on controlled manipulator. The three identified activities are underlined as possible steps of the procedure.

A M-states continous HMM is then trained for each activity a. We estimate the $\mathbf{B}_a$ transition matrix using a modified version of the Baum-Welch algorithm in which the emission probability density parameters of each state z were imposed to $\widehat{\mathbf{T}}_z$ and $\widehat{\mathbf{C}}_z$ [14], [16].

The training algorithm is implemented in MATLAB environment (R2014b, MathWorks).

C. Classification of current activity

In this work we are interested in runtime recognition of performed activity, thus we need to deal with classification of incomplete set of data. For each new chunk of data $\mathbf{d}^*$ the classifier computes the class-conditional likelihood of $\mathbf{d}^*$, that is $p(\mathbf{d}^*|\boldsymbol{\theta}, \mathbf{B}_a)$, for each trained HMM, i.e. for each modeled activity a using the forward-backward algorithm. At each time the activity that maximizes the class-conditional likelihood on the available chunk of data is classified as the current activity.

The runtime classification algorithm is implemented in C/C++ language inside the Robotic Operating System[1] (ROS). Every iteration, the algorithm computes $p(\mathbf{d}^*|\boldsymbol{\theta}, \mathbf{B}_a)$ for each activity model over data buffered in an array with First In First Out (FIFO) logic that is refreshed with new data incoming from the robot controller, deploying a full-parallel implementation based on multi-core hardwares.

[1] `www.ros.org`

D. Experimental scenario and validation

The proposed activity recognition algorithm was experimentally tested on a LWR4+ (Kuka, Augsburg, Germany), a 7 degrees-of-freedom redundant manipulator with flexible joints, inside a scenario that addresses hands on robotic assistance for open-skull neurosurgical procedures [18], [8].

Since the procedure encompasses simple targeting gestures, three exemplary activities were identified (as reported in Figure 1), i.e.

- *pointing*, when approaching the patient's brain during targeting task;
- *wandering*, when moving the manipulator in space without a specific target;
- *idle*, when not moving the manipulator.

each one possibly requiring activation/deactivation of different control modalities.

1) Model training: A dataset (i.e. both trajectory $\mathbf{x}$ and forces $\mathbf{f}$) of 30 trials performed by one expert user for each activity was recorded (200 Hz) and processed off-line to model $\boldsymbol{\theta}$ and train each HMM.

2) Runtime validation: During runtime validation, 5 non-surgeon users were asked to perform targeting toward 5 arbitrary points over a brain phantom with the help of the manipulator. Each task started with the robot in a fixed position and users were instructed to follow a specific sequence of actions (i.e. *idle*, *wandering*, *pointing*, *idle*, *pointing*, *wandering*, *idle*) and to bring the robot back to the starting position at the end. To provide ground-truth regarding current performed action, user were provided with a button to press when switching from one activity to another. The classifier was run at 20 Hz over 50 samples long buffer (i.e. 250 ms of signal) on a dual-core Intel Xeon@2.66 GHz processor. The sample accuracy $\mathbf{e}_{sample}$, i.e. the percentage of samples in the classification that share the same label with the ground-truth [15], were computed over the complete set of trials for each user.

III. RESULTS AND DISCUSSION

A. Model training

The results of the different activity models training is shown in Figure 2. The unsupervised algorithm used fitted the data with a mixture of 26 Gaussian distributions ($M = 26$). A projection over the xy plane (with respect to the base of the manipulator) of both fitted data and GMM components is shown in Figure 2a. Estimated transition matrices $\mathbf{B}_a$ are shown for each modeled activity in Figure 2b. The *idle* matrix shows how one state acts as an attraction well, representing the emission of the zero-displacement Gaussian distribution (i.e. the stopped robot). On the other hand, the *pointing* matrix is similar to a diagonal matrix, showing this activity can be assumed as generated by a single low-level model (i.e. represents a linear trajectory). Conversely. the *wandering* activity shows a more sparse matrix, underlying the random nature of the movement.

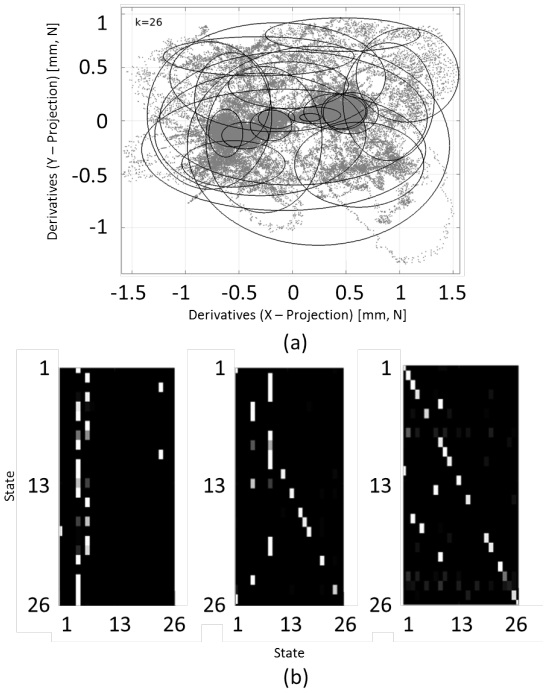

Fig. 2. Results of the model fitting for the three activities. The upper picture represents a 2-D projection of the displacements and of the 26-components GMM that is used to fit the data. The lower picture represents the transition matrix $\mathbf{B}_a$ for the three trained HMM, i.e. from left to right, *idle*, *wandering*, *pointing*.

TABLE I
SAMPLE ACCURACY DURING VALIDATION

	User 1	User 2	User 3	User 4	User 5
Accuracy	0.9126	0.8928	0.8921	0.8971	0.8990

B. Runtime validation

A typical segmentation of user's activity during each trial is represented in Figure 3. The overall sample accuracy $\mathbf{e}_{sample}$ for each user is reported in Table I. The percentage of correctly classified samples is over 89% for every user. In fact, as shown in Figure 3a most of the misclassifications occurred during the activity changes due to the algorithm's reaction time. These results are comparable with off-line gesteme-based classifiers, i.e. classification over a complete set of data for each activity [15]. Gesteme-based online classifiers showed an accuracy higher than 80% only when observing more than the 60% of the activity [14].

Because of the small part of the signal observed (i.e. 50 samples), the algorithm shows fast response ($\sim$300 ms for each user, Figure 3b) to sudden changes but some classification errors occurred even far from the activity transitions.

In Table II the confusion matrix computed among all the trials from users is shown. The algorithm can better identify when the robot is stopped (i.e. *idle*, $\sim$93% accuracy) than during actual manual guidance (i.e. *pointing*, $\sim$88% of accuracy and *wandering*, $\sim$86% of accuracy). In particular,

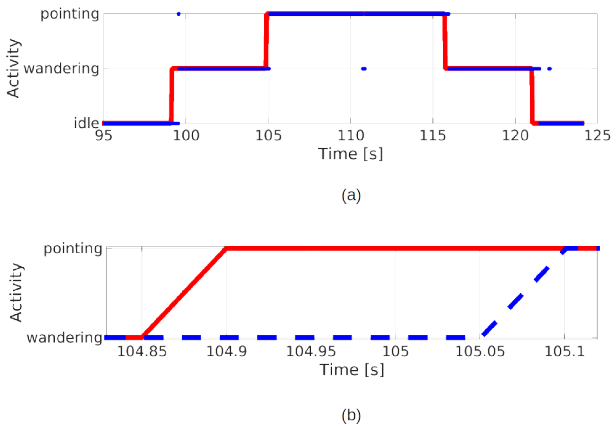

Fig. 3. Typical segmentation of a user's targeting task is reported in (a). Algorithm reaction to a transition is reported in (b). Red line represents provided ground-truth regarding current activity, while blue spots represent algorithm classification.

TABLE II

CONFUSION MATRIX DURING RUNTIME VALIDATION

	Classification		
True Activity	pointing	wandering	idle
pointing	0.8824	0.0819	0.0357
wandering	0.0668	0.8660	0.0672
idle	0.0145	0.0478	0.9377

misclassification between *wandering* and *pointing* resulted to be higher than 8%.

IV. CONCLUSIONS

In this paper we presented the first results in user's activity recognition during hands-on robotic surgery obtained with an algorithm that does not rely on primitives identification to perform classification. The obtained accuracy in classification exceeds 89% with a reaction time approximately of $\sim$300 ms.

It has to be stated that those results are obtained from tests performed only on 5 non-surgeon subjects. A more in depth study will need to evaluate the algorithm's performances over a pool of novice and expert surgeons.

Critical aspects not addressed by this paper were reaction time to activities transition and size of the data buffer used for classification. In particular, wide buffers should guarantee a more robust classification during one single activity, with slower response to a sudden transitions.

To overcome this problem, an Adaptive Windowing algorithm will be implemented in future work, to classify over short buffer when the distance between the *winner* model likelihood and the others overcomes a threshold, increasing the buffer size elsewhere. Moreover, runtime classification could be exploited to trigger different control strategies on the manipulator based on the current detected activity. For example, a high-accuracy variable damping control applied during *pointing* in contrast to a more compliant control during *wandering* could enhance the cooperation feeling.

REFERENCES

[1] B. Davies, S.J. Harris, F. Rodriguez y Baena, P. Gomes, and Jakopec M., "Hands-on robotic surgery: is this the future?," in: Medical Imaging and Augmented Reality, Springer Berlin Heidelberg, pp. 27-37, 2004.

[2] B. Davies, M. Jakopec, S.J. Harris, F. Rodriguez y Baena, A. Barrett, A. Evangelidis, P. Gomes, J. Henckel, and J. Cobb, "Active-constraint robotics for surgery," in: Proceeding of the IEEE. 94(9), pp. 1696-1704, 2006.

[3] M. Jakopec, S.J. Harris, F. Rodrigurez y Baena, P. Gomes, J. Cobb, and B. Davies, "Preliminary results of an early clinical experience with the acrobot system for total knee replacement surgery," in: Medical Image Computing and Computer-Assisted Intervention - MICCAI 2002, Springer Berlin Heidelberg, pp. 256-263, 2002.

[4] MacLachlan R.A., Becker B.C., Tabares J.C., Podnar G.W., Lobes L.A., and Riviere C.N. "Micron: An actively stabilized handheld tool for microsurgery," in: IEEE Transactions on Robotics, vol. 28(1), pp. 195-212, 2012.

[5] A. Uneri, M.A. Balicki, J. Handa, P. Gehlbach, R.H. Taylor, and I. Iordachita, "New steady-hand eye robot with micro-force sensing for vitreoretinal surgery," in: Proc of IEEE Int Conf on Biomedical Robotics and Biomechatronics, pp. 814-819, 2010.

[6] J.G. Petersen, and F. Rodriguez Baena, "A dynamic active constraints approach for hands-on robotic surgery," in: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, pp. 1966,1971, 2013.

[7] M. Li, and A.M. Okamura, "Recognition of operator motions for real-time assistance using virtual fixtures," in: Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS 2003), Proceedings of the 11th Symposium on, pp. 125-131, 2003.

[8] E. Beretta, F. Nessi, G. Ferrigno, F. Di Meco, A. Perin, L. Bello, F. Cardinale, and E. De Momi, "Enhanced torque-based impedance control to assist brain targeting during open-skull neurosurgery: a feasibility study," in: Journal of Medical Robotics and Computer Assisted Surgery (JMRCAS), 2015, under revision.

[9] O.C. Schrempf, U.D. Hanebeck, A.J. Schmid, and H. Worn, "A novel approach to proactive human-robot cooperation," in: Robot and Human Interactive Communication, 2005 (ROMAN 2005), IEEE International Workshop on, pp. 555-560, 2005.

[10] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in: Proceedings of the IEEE, vol. 77(2), pp. 257-286, 1989.

[11] A. Brakensiek, A. Kosmala, D. Willett, W. Wang, and G. Rigoll, "Performance evaluation of a new hybrid modeling technique for handwriting recognition using identical on-line and off-line data," in: Document Analysis and Recognition (ICDAR'99), Proceedings of the Fifth International Conference on, pp. 446-449, 1999.

[12] R.H. Liang, and M. Ouhyoung, "A real-time continuous gesture recognition system for sign language," in: Automatic Face and Gesture Recognition, Proceedings of the Third IEEE International Conference on, pp. 558-567, 1998.

[13] A. Castellani, D. Botturi, M. Bicego, and P. Fiorini, "Hybrid HMM/SVM model for the analysis and segmentation of teleoperation tasks," in: Robotics and Automation, 2004 (ICRA'04), Proceedings of the IEEE International Conference on, vol. 3, pp. 2918-2923, 2004.

[14] D. Aarno, and D. Kragic, "Motion intention recognition in robot assisted applications" in: Robotics and Autonomous Systems, vol. 56(8), pp. 692-705, 2008.

[15] C.S. Hundtofte, G.D. Hager, and A.M. Okamura, "Building a task language for segmentation and recognition of user input to cooperative manipulation systems," in: Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS 2002), Proceedings of the 10th Symposium on, pp. 225-230, 2002.

[16] J.C. Nascimento, M.A. Figueiredo, and J.S. Marques, "Trajectory classification using switched dynamical hidden Markov models," in: Image Processing, IEEE Transactions on, vol. 19(5), pp. 1338-1348, 2010.

[17] M.A. Figueiredo, and A.K. Jain, "Unsupervised learning of finite mixture models," in: Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 24(3), pp. 381-396, 2002.

[18] E. Beretta, E. De Momi, F. Rodriguez y Baena, and G. Ferrigno, "Adaptive hands-on control for reaching and targeting tasks in surgery," in: International Journal of Advanced Robotic System, 2015, accepted.