ADVANCED ALGORITHMS FOR SURGICAL GESTURE CLASSIFICATION

Giovanni Luca Santosuosso[1], Giovanni Saggio[1], Fabio Sorà[1], Laura Sbernini[2], Nicola Di Lorenzo[2]

[1]Dept. of Electrical Engineering, Univ. of Rome "Tor Vergata",
Via del Politecnico 1, 00133 Rome, Italy

[2]Dept. of Experimental Medicine and Surgery, Univ. of Rome "Tor Vergata",
Via Montpellier 1, 00133 Rome, Italy

ABSTRACT

A novel gesture binary classification procedure is presented to determine surgical ability. To this aim a sensory glove was employed to track surgical hand movements and sensors data were recorded to be processed by a specific algorithm. The classification task was able to discriminate a gesture made by an expert surgeon with respect to a novice one, thanks to a two steps classification strategy. The first one produced a binary tree of parameters associated to a sensor time function; they were elaborated in the second step by a neural network providing a real output whose magnitude was associated to the surgeon ability. Experimental tests correctly classify all operators in a group.

Index Terms **—** Wearable sensors, neural networks, biomedical signal processing, supervised learning, computational intelligent.

1. INTRODUCTION

Manual dexterity and technical skills are surgeon key qualities along with cognitive capabilities, judgment and decision-making. However, surgical training is still based on ''see one, do one, teach one'' scheme in which senior surgeon observes a student and provides verbal feedback [1]. Thus novice's assessment is based on subjective criteria and observation hard to standardize. New approaches for surgical training are desirable to allow trainees to practice in a safe environmental before the operating room and to provide objective feedbacks without direct supervision.

Recent studies have introduced several examples of skill-evaluation tool and suggest that motion analysis has a considerable potentiality as a basis for measuring surgical ability. The major criticism in this field concerns open surgery: this is still the standard procedure and is difficult to measure. In fact the surgeon freely manipulates instrumentation in the tridimensional open space [1]. To this purpose, several examples focus on hand movements tracking: magnetic sensors (e.g. ICSAD uses an electromagnetic tracking system with markers placed on the dorsum of each hand [2]), optical motion tracking systems and videographic recording [3] are used to discriminate between the performance of experts and novices. In these studies several parameters are currently measured (e.g. the number of hand movements, rate of movements, time taken to complete a task, etc.) to assess surgical ability. The aforementioned examples, however, present several limitations as an uninterrupted line of sight between cameras and markers (optical and video tracking), dedicated environments and highly sensitivity to environmental noise (electromagnetic systems).

In this paper we propose the use of a sensory glove to track the surgeon's hands movements during an open surgical simulated task, to solve the limitations of other techniques. Furthermore, a sensory glove records degree of freedom (DOF) data from each hand's joints and is comfortable for the users because it does not affect the natural free-movements of the hands. Potentially, these features make our system more suitable for open surgery.

Finally we develop an algorithm capable to classify the ability to perform a basic suturing task not limited to the use of some parameters but on the basis of the entire gesture. The idea behind our classification algorithm is related to the strategy developed, among others in [4] for the classification of electroencephalogram (EEG) signals, in [5] for power system transients classification, and in [6] for epileptic seizure detection. In the works above two networks were used for classification: a first level network that operates starting from time domain signals yielding a reduced dimension output; the second-level networks were trained using the outputs of the first-level networks as input data; in general they consisted in a standard multilayer feedforward neural network (NN) ([7], [8]). The first level network employed in [4], [5], [6] yielded the discrete wavelet transform coefficients; in this paper we produced a hierarchical binary tree structure, following the idea in [9], [10]. This was the approach in [11], where a modified wavelet transform called the tree-structured wavelet transform or wavelet packets was proposed. In [12], following [13], this algorithm was used for the discrimination of two-dimensional shapes. It also provided a method to identify a subset of the edges that are most salient for discrimination. In [14] this methodology was applied to

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

stream mining applications; specifically for configuring cascaded binary classifier trees. They required the identification of several different attributes in data content and hence relied on a distributed set of cascaded statistical classifiers to filter and process the data dynamically.

In the present paper the two level network is combined with hierarchical classification: the first level network produces a binary tree of parameters representing recursively the 'centers' of the time function edges. The second level network multilayer feedforward NN correctly classifies the gesture as one of an expert or of a novice.

2. MATERIALS AND METHODS

2.1. The sensory glove

Bend and inertial sensors were used to measure motion of fingers joints and wrist. In particular we used fourteen flex sensors (by Flexpoint Sensor Systems, Inc.) and one inertial measurement unit, IMU, with a three axis accelerometer and a three axis gyroscope (Analog Combo Board Razor - 6DOF Ultra-Thin IMU, by SparkFun). Sensors were placed on distal interphalangeal (DIP), proximal interphalangeal (PIP) and metacarpophalangeal (MCP) fingers joints for the bend sensors and on dorsal side of the hand for the inertial units (Figure 1). In detail, a gesture recording was composed of twenty time trajectories, fourteen for the bend sensors and six for the 6DOF of the inertial sensors.

The glove was part of a system of measurement and reproduction in virtual environment for static and dynamic postures of the hand [15]. It was possible, furthermore, to represent recorded movement of the hand in real time via avatar representation.

2.2. Subjects and surgical task

We recruited 18 volunteers, 9 "experts" (i.e. senior surgical residents) and 9 "novices" (i.e. undergraduate medical students with minimal or without surgical experience). Surgical performance was assessed on a basic suturing task performed on a standard foam pad to simulate the skin tissue. Subjects were asked to use a needle driver to make a passage of suture with a needle across a vertical incision pre-prepared on the pad. To standardize the performance a neutral position for the hand (holding the needle driver) and for the pad was defined and drawn on a desk. All the participants were right-handed and have used his/her dominant hand to complete the surgical task.

Each subject had to perform the gesture 10 times with a total of 180 recordings (i.e. 18 volunteers and 10 repeats). Data were separately collected for each repeat and for each sensor. Following raw data were autoscaled and time-normalized to have resampled trajectories of 1000 samples.

2.3. Data processing

FIGURE 1 THE SENSORY GLOVE.

In this section the algorithm developed to data processing is described, following the approach in [9], [10], [12], [13], [14]. The main task of the classification strategy is comparing two different trajectories; in fact a human gesture can be represented as a matrix, each row associated to a sensor time trajectory. In order to validate the affinity degree among trajectories of the same sensor we focus our attention on peculiar trajectories points.

2.3.1. Binary tree generation

We recursively construct a binary tree whose nodes are associated to trajectory portions (see [9], [16]). In particular, we consider trajectory local minima and maxima since they are the most relevant for a correct curve description. Actually to improve methodology clarity we define a sort of "trajectory center" instead of the single local minima and maxima alone. In what follows we determine the function yielding the center of mass coordinates. To be specific, let $Y(h)$, with $1 \leq h \leq N$, be the discrete time function associated to the sensor time output composed of N samples.

First of all we find the couple $[A_x, A_y]$ associated to the average values of dominion and co-dominion of the function $Y(h)$; $A_x = N/2$ while $A_y = \frac{1}{N}\sum_{i=1}^{N} Y(i)$. We assume *n* local minima and/or maxima with coordinates $[E_x(i), E_y(i)]$ for $1 \leq i \leq n$. For each local minimum or maximum we compute the couple $[\bar{x}(i), \bar{y}(i)]$ given by $\bar{x}(i) = E_x(i) - A_x$ and $\bar{y}(i) = E_y(i) - A_y$. Please, notice that $\bar{x}(i)$ is negative if the point is located on the left of A_x and $\bar{y}(i)$ is negative if the point is located below A_y and positive if it is above.

Let $V_e[i][1] = \bar{x}(i)$ and $V_e[i][2] = \bar{y}(i)$. Now, we can define the center of mass coordinates as $[b_x, b_y]$. In order to define b_x set the *n* coefficients

$$\alpha_i = \frac{|V_e[i][2]|}{\sum_{j=1}^{n}|V_e[j][2]|}, 1 \leq i \leq n. \quad (1)$$

Expression (1) yields

$$b_x = \sum_{i=1}^{n} |V_e[i][1] * \alpha_i| \quad (2)$$

which is the weighted average of $V_e[i][1]$ via the coefficients α_i. The idea behind the definition of the weighting coefficients α_i gives more importance to greater local minima and/or maxima. For the genesis of b_y we have to set the coefficients

$$p_i = \frac{1}{|V_e[i][1]|+\varepsilon} \quad (3)$$

where ε is a suitable design parameter to be chosen sufficiently small. The role of the coefficient defined in (3) is to assign a weight which is greater when the smaller is the distance between a specified local minimum or maximum to the centre of the trajectory. The coefficient in (3) generate normalized weights

$$\overline{\alpha_i} = \frac{p_i}{\sum_{j=1}^{n} p_i}, \quad (4)$$

which finally yields

$$b_y = \sum_{i=1}^{n} |V_e[i][2] * \overline{\alpha_i}|. \quad (5)$$

The couple $[b_x, b_y]$ has to be rescaled to produce values in the range [-1,1] to improve differences detectability. To this purpose let R_x and L_x be the minimum and maximum respectively of the abscissa in $Y(h)$ and R_y and L_y be the minimum and maximum respectively of the trajectory $Y(h)$. Now the normalized value of center of mass

$$\overline{b_x} = \left(\frac{b_x + A_x - R_x}{L_x - R_x}\right), \quad (6)$$

$$\overline{b_y} = \left(\frac{b_y + A_y - R_y}{L_y - R_y}\right). \quad (7)$$

This procedure is also applied on the two sub-trajectories obtained by dividing the original one in two sets given by all the points in $Y(h)$, such that $h \leq \overline{b_x}$ and $h \geq \overline{b_x}$ respectively. In addiction it is recursively applied to subset obtained by further splitting sub-trajectories themselves in new sub-trajectories of smaller dimension. This strategy can be interpreted as a binary tree construction in which the root node contains the couple $[\overline{b_x}, \overline{b_y}]$ of the entire trajectory $Y(h)$, its left child nodes memorizes the centers of mass of the sub-trajectory obtained considering $Y(h)$, with h between 1 and the not normalized center of mass abscissa b_x of the entire trajectory. Its right child nodes collects the centers of mass of the portion of the trajectory $Y(h)$ for h between b_x and L_x. This algorithm is in turn recursively applied on both left and right nodes producing sub-trajectories centers that become smaller the deeper is a node into the tree. The algorithm stops when either the tree reaches the maximum depth n_T assigned by the designer or when there are not local minima and/or maxima inside the portion of the trajectory considered.

2.3.2. Neural training for data classification

The next step in correct gesture classification is the implementation of a feedforward multilayer neural network (see [7], [8], [17],[18]) whose inputs are produced from the binary trees data. To this purpose however, the latter quantities have to be transformed into a suitable vector. To be specific, define the vector of real numbers X, whose dimension is $2 * (2^{n_T} - 1)$. Its role is to record the information contained into the tree nodes, i.e. the couples $\overline{b_x}$ and $\overline{b_y}$, multiplied by a normalization function taking into account the depth inside the binary tree of a node. Recalling that the binary tree overall depth is n_T, set:

$$F(i) = \left[\left(\frac{\delta-1}{n_T-1}\right) * i + \left(\frac{n_T-\delta}{n_T-1}\right)\right] \quad (8)$$

where δ is a design parameter to be chosen less than one, and the integer i is simply the depth of a given node inside the tree.

Notice that the function in (8) yields $F(1) = 1$ for construction i.e. when we consider the root node of the tree. For a node of maximum depth inside the tree we have $F(n_T) = \delta$. Thus the application of the function in (8) for a node of depth i yields a value between 1 and δ. Since δ by construction is less than 1 the function in (8) "weights" the nodes values giving more relevance to the ones closer to the tree's root. The first two entries of the vector X are $X(1) = F(1) * \overline{b_x}$ and $X(2) = F(1) * \overline{b_y}$ where $[\overline{b_x}, \overline{b_y}]$ are the center of mass of the root node. Entries between $X(3)$ and $X(6)$ are the centers associated to the root left and right child nodes multiplied this time by $F(2)$. This strategy is iterated for all the nodes of the tree. We recall that the binary trees main contain ranches that do not reach the maximum depth, this happens for trajectory subparts without an internal local minimum or maximum occurring at depth less than n_T. Thus the entries of the vector X remaining to be determined when the binary transposition is completed are set to zero by default. We choose the maximum depth n_T equal to 5: this is a compromise between accuracy and data compactness; this yield the binary tree to has $2^{n_T} - 1 = 31$ nodes and the vector X to has a length of 62.

The data provided by the binary tree are eventually used to train the neural network based classifier whose task is to distinguish between a gesture of an expert and a gesture of a novice. To this purpose we assume that an output of the network close to 1 represent a gesture of an expert and an output close to -1, on the contrary, represent a gesture of a novice.

As mentioned above data set is made up of 180 gesture recordings, 90 for the experts group and 90 for the novices group, and each gesture is composed of 20 time trajectories. Data processing is performed separately for each sensor, thus we have 20 vector X for each gesture recording. A separated neural network is implement and trained for each sensor. Resulting 20 data sets, each one is associated to a sensor and composed by 180 vector X, are divided one by one into three sub-sets: training set (70%), validation set (20%) and test set (10%). Notice that data from each subject are distributed into the three sub-sets with the aim of generalize the input of the network and improve subjects variance detectability.

The first stage of the classifier is a feedforward backpropagation neural network of three layers. Let p and t be respectively the dominion and the co-dominion of the training set for each sensor. p includes vectors X and each element of t is defined opportunely: 1 correspond to an expert and a -1 correspond to a novice. Notice that t is the scalar vector that represent the desired network outputs. The network architecture is show in the partial Matlab script below (see [17], [18] for the functions employed):

```
net{1,x}=newff(p,t,[100,50,50],{'tansig',...
'tansig','tansig','tansig'},'traingdx');
net{1,x}.trainParam.show=25;
net{1,x}.trainParam.goal=1e-1;
net{1,x}.trainParam.epochs=1000;
net{1,x}.trainParam.lr=0.0001;
net{1,x}.trainParam.lr_inc=1.02;
net{1,x}.trainParam.mc=0.7;
net{1,x}=init(net{1,x});
net{1,x}=train(net{1,x},p,t,[],[],val,test);
```

where `x` range between 1 and 20 (number of sensors) and `val` and `test` are respectively the validation and test sub-sets.

The second and last stage of the classifier is another neural network whose input are the outputs of the first stage. This simple network is implemented to take the mean value of its input. To this purpose synaptic weights are $1/n$ where n is the number of sensors. After that all values are summed. This strategy is adopted to obtain a unique value for each participant from the 20 sensors and to generalize the results to the whole gesture.

3. RESULTS

By applying the algorithm and the neural network described in section 2 we classified participants in term of his/her surgical ability. Recalling that we chose the output of the neural network based classifier range from -1 and 1. Performance was automatic assigned to the experts group when result was positive and to the novices group if result was negative. Table 1 summarizes the results. In particular it is important to remark that the test set elements are precisely 18 gestures each one associated to a different individual, thus 9 experts and 9 novices. They are not included in the neural training algorithm, so that since data associated to test set are correctly classified by the NN then we can state that our strategy has learned correctly the distinction between a gesture associated to a novice and an expert one. It can be worth mentioned that we have 0% error of misclassification because each participant was correctly classified.

TABLE 1 RESULTS FOR EXPERTS AND NOVICES GROUPS.

Experts	Novices
0.219	-0.531
0.329	-0.197
0.288	-0.346
0.337	-0.282
0.281	-0.011
0.208	-0.120
0.333	-0.195
0.336	-0.203
0.285	-0.129

4. CONCLUSIONS

A novel classification algorithm has been tested on data collected from a sensory glove to determine surgical skill. In this framework a two steps procedure was implemented. First local maxima and/or minima associated to a sensor time function were elaborated to produce a binary tree retaining peculiar features of gestures. These data were further processed by a three layers feedfoward neural network. The overall output was a number in the range [-1,1], by associating numerical outputs whose magnitude is proportional to the surgeon skill. We trained the algorithm with a supervised learning strategy with a training set associated to gestures of eighteen subjects: nine experts and nine novices. The trained network has been tested to classify another set of gestures from the two groups of surgeons. All the gestures made by individuals in the experts group were associated to positive numbers while the ones by non experts produce negative numbers. In our view, this can be considered as an important experimental validation of a novel classification algorithm conceived to handle classification problems whose dominion is a high dimension space.

5. REFERENCES

[1] C.E. Reiley, H.C. Lin, D.D. Yuh and G.D. Hager, "Review of methods for objective surgical skill evaluation", *Surgical Endoscopy*, 25(2):356-66, February 2011.

[2] V. Datta, S. Mackay, M. Mandalia and A. Darzi, "The use of electromagnetic motion tracking analysis to objectively measure open surgical skill in laboratory-based model", *Journal of the American College of Surgeons,* 193(5):479–485, 2001.

[3] R. Sonnadara, N. Rittenhouse, A. Khan, A. Mihailidis, G. Drozdzal, O. Safir and S.O. Leung, “A novel multimodal platform for assessing surgical technical skills”, *American Journal of Surgery*, 203(1), 32–36, 2012.

[4] E.D. Übeyli, “Combined neural network model employing wavelet coefficients for EEG signals classification”, *Digital Signal Processing*, Volume 19, Issue 2, Pages 297-308, March 2009.

[5] H. Zhengyou, G. Shibin, C. Xiaoqin, Z. Jun, B. Zhiqian and Q. Qingquan, “Study of a new method for power system transients classification based on wavelet entropy and neural network”, *International Journal of Electrical Power & Energy Systems*, Volume 33, Issue 3, Pages 402-410, March 2011.

[6] L. Guo, D. Rivero, J. Dorado, J.R. Rabuñal and A. Pazos, “Automatic epileptic seizure detection in EEGs based on line length feature and artificial neural networks”, *Journal of Neuroscience Methods*, Volume 191, Issue 1, Pages 101-109, 15 August 2010.

[7] G.B Huang, Y.Q. Chen and H.A. Babri, “Classification ability of single hidden layer feedforward neural networks”, *Neural Networks, IEEE Transactions on*, vol.11, no.3, pp.799,801, May 2000.

[8] R.P. Lippmann, “Pattern classification using neural networks”, *Communications Magazine, IEEE*, vol.27, no.11, pp.47,50, Nov. 1989.

[9] P. Yin, A. Criminisi, J. Winn and I. Essa, “Bilayer Segmentation of Webcam Videos Using Tree-Based Classifiers”, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol.33, no.1, pp.30,42, Jan. 2011.

[10] B.C. Lovell and A.P. Bradley, “The multiscale classifier”, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol.18, no.2, pp.124,137, Feb 1996.

[11] T. Chang and C.-C.J. Kuo, “Texture analysis and classification with tree-structured wavelet transform”, *Image Processing, IEEE Transactions on*, vol.2, no.4, pp.429,441, Oct 1993.

[12] V.Y.F. Tan, S. Sanghavi, J.W. Fisher and A.S. Willsky, “Learning Graphical Models for Hypothesis Testing and Classification”, *Signal Processing, IEEE Transactions on*, vol.58, no.11, pp.5481,5495, Nov. 2010.

[13] Y. Amit, D. Geman and K. Wilder, “Joint induction of shape features and tree classifiers”, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol.19, no.11, pp.1300,1305, Nov 1997.

[14] B. Foo, D.S. Turaga, O. Verscheure, M. van der Schaar and L. Amini, “Resource Constrained Stream Mining With Classifier Tree Topologies”, *Signal Processing Letters, IEEE*, vol.15, pp.761,764, 2008.

[15] G. Saggio, S. Bocchetti, C.A. Pinto and G. Orengo, “Wireless Data Glove System developed for HMI”, *ISABEL2010 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies,* Rome, Italy, 7-10 November 2010.

[16] M. Athimethphat and B. Lerteerawong, “Binary classification tree for multiclass classification with observation-based clustering”, *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 9th International Conference on*, 2012.

[17] S.N. Sivanandam and S. Sumathi, *Introduction to Neural Networks Using MATLAB 6.0*, Tata McGraw-Hill, 2005.

[18] T.J. Anastasio, *Tutorial on Neural Systems Modeling*, Sinauer Associates, Inc, 2009.