2014 5th IEEE RAS & EMBS International Conference on
Biomedical Robotics and Biomechatronics (BioRob)
August 12-15, 2014. São Paulo, Brazil

Towards a Cognitive Camera Robotic Assistant

I. Rivas-Blanco, B. Estebanez, M. Cuevas-Rodriguez, E. Bauzano, and V.F. Muñoz, *Member*, *IEEE*

Abstract— **This paper presents a cognitive architecture for a camera robotic assistant aimed at providing the proper camera view of the operating area in an autonomous way. The robotic system is composed of a miniature camera robot and an external robotic arm. The camera robot is introduced into the abdominal cavity and handled by the external robot through magnetic interaction. The cognitive architecture is provided with a long-term memory, which stores surgical knowledge, behaviors of the camera and learning mechanisms, and a short-term memory that recognizes the actual state of the task and triggers the corresponding camera behavior. To provide the proper camera view, each state of the task is characterized by a Focus of Attention (FOA), defined by an object, a position of the object in the image, and a zoom factor. The architecture also includes a learning mechanism to take into account particular preferences of surgeons concerning the viewpoint of the scene. The architecture proposed is validated through a set of in-vitro experiments.**

I. Introduction

In the last decades, Minimally Invasive Surgery (MIS) has become a widely accepted technique as an alternative to traditional open surgery procedures. In this field, robotic assistants have found a wide range of applications, from surgeon extenders to auxiliary surgical supports [1]. Surgeon extenders, which main reference worldwide is the da Vinci Surgical System [2], are operated directly by the surgeon. Although these devices enhance surgeons' abilities in terms of accuracy, accessibility and dexterity, they require long training periods and challenge surgical tasks remain tedious and time consuming. On the other hand, auxiliary surgical supports devices work side-by-side with the surgeon and perform functions such as holding the endoscope or retraction. Traditionally, these devices use direct control interfaces, such as head-trackers [3], eye-trackers [4]-[5], or voice-activated control [6]. However, although these methods have succeeded in substituting medical stuff, they introduce extraneous devices that distract the surgeon from the important surgical tasks. Therefore, it is in this field where there is still much to be done to provide robotic assistants with capabilities to interact with surgeons in a similar way as a human assistant would do.

An ideal robotic assistant should combine human and robot capabilities under the co-worker concept [7], i.e., the robot should collaborate with surgeons in a natural and autonomous way, thus requiring less of the surgeons' attention. A robotic co-worker needs human interaction, perceptual and actuation systems to interact with the environment, and learning mechanisms. Combining these characteristics into a cognitive architecture would provide a robotic assistant with the capability of collaborating autonomously with surgeons during a particular task, either performing preprogrammed basic actions or controlling the endoscope. A natural human-robot communication is based on emulating how humans communicate among each other, basically, voice commands or by gestures. Jacob et al. [8] have combined these two interfaces in a robotic scrub nurse that assists the surgeon by passing surgical instruments. The intelligence of the robot is based on a finite-state machine that evolves depending on the input received from the surgeon. However, collaboration during surgical procedures is a more complex scenario and requires providing the robot with surgical knowledge to be able to identify the actual state of the task to act on this recognition. In this sense, Padoy et al. [9] have proposed a collaborative system for suturing based on recognizing the actions performed by the surgeon through Hidden Markov Models, and triggering previously learned motions. Bauzano et al. [10] propose a similar approach but for a Hand Assisted Laparocopic Surgery scenario.

Similar approaches based on tasks workflow analysis have been addressed for automating the motion of the endoscope. Ko et al. [11] propose an intelligent interaction architecture that suggests the proper camera view depending on the current surgical state. Surgical knowledge is built assuming a state-transition diagram, where transition conditions depend only on the surgical tool in use. Weede et al. [12] enhance the decision capability of the robot by long-term prediction of the surgical instruments motion. This method is based on building a knowledge base of the position of the instruments for a particular procedure from recordings of former interventions. Using Markov chains, the system predicts the area where the surgical tools are going to move. The ideal field of view should include all predicted points and both tools' tips. Although both previous works represent a cognitive solution for the positioning of the endoscope, none of them include online learning algorithms to improve the behavior of the robotic assistant.

In this work, we propose a cognitive architecture for a camera robotic assistant aimed at providing the proper camera view of the scene for each state of the surgical task. Unlike previous work, this architecture includes a learning mechanism to improve the behavior of the robot. Long-term memory is divided into procedural, semantic and episodic memory, where different classes of knowledge are stored, while short-term memory represents the inference engine of the system. Besides instrument tracking, camera view is improved by using the Focus of Attention (FOA) of the current state of the task. We validate this approach on a suturing task. This paper is organized as follows. The camera robotic system is described in section II. Section III illustrates the cognitive architecture, including a detailed description of all its modules. Experimental results are reported in section IV, and conclusions are discussed in section V.

II. Camera Robotic System

The camera robotic system used in this work is composed of a miniature camera robot handled by an external robotic arm. The camera robot, provided with a set of magnets, is introduced into the abdominal cavity through an entry port (by which the surgical tools are inserted), and attached at the abdominal wall through magnetic interaction. To this aim, a magnetic holder is attached at the end effector of the external robot. Thus, motion of the camera robot is controlled by translating the external robot along the abdominal wall. Advantages of this system versus a conventional laparoscope are that motion of the camera is not restricted by the entry port, so more camera views of the abdominal cavity can be obtained, and the incision for the laparoscope is avoided. Moreover, automatic guidance of the camera can be performed by automating the motion of the external robot. Tracking of the surgical tools is performed by an optical 3D tracker by means of reflective markers attached at the tools. An overview of the complete system is shown in Fig. 1, along with the references frames of the different components of the system. Additionally, a detail of the camera robot prototype is depicted. Camera robot size is 30 x 22 x 90 mm, and it is composed of a High Definition camera (Logitech HD Webcam C310), fourteen white LEDs to illuminate the operating area, and two permanent magnets at the bottom.

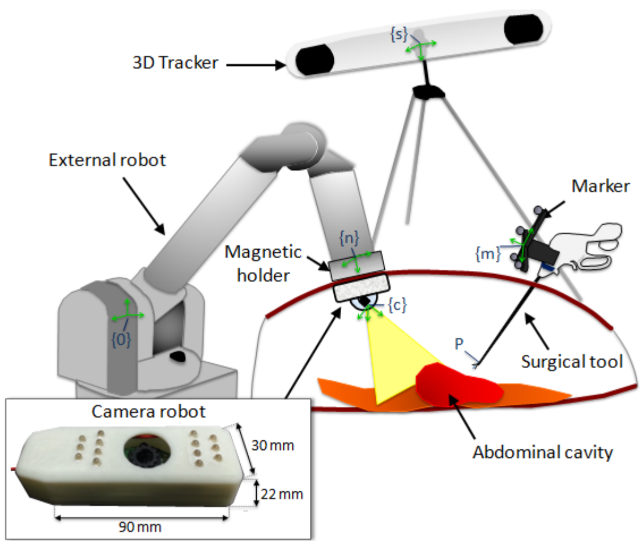

Figure 1. Robotic system overview

III. The Cognitive Architecture

As mentioned in section I, an ideal robotic assistant should combine human and robot capabilities to provide surgeons the proper camera view of the operating area without disrupting their concentration on the surgical procedure. Human assistants capabilities are mainly based on their surgical background and their learning abilities. Surgical background provides knowledge of the surgical workflow to be able to identify the actual state of the task and to know where the FOA is, while learning abilities allow them to learn particular preferences of surgeons in order to improve their behavior. Therefore, an intelligent robotic assistant should include these human capabilities. Cognitive architectures specify the underlying infrastructure for an intelligent system. As stated by Langley [13], a cognitive architecture should include: short-term and long-term memories to store information about the system beliefs, goals and knowledge, representations of elements contained in the memory organized into mental structures, and functional processes that operate on these structures and learning mechanisms.

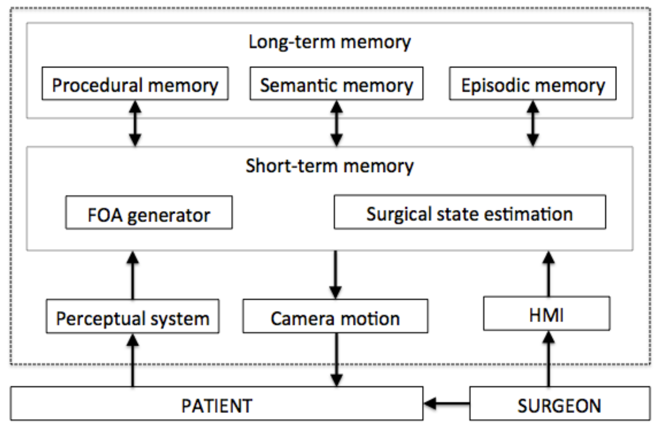

Figure 2. Cognitive Architecture

Fig. 2 shows the cognitive architecture proposed in this work, which is aimed at controlling the camera robot in an intelligent way to provide the surgeon the proper camera view during a particular procedure. Long-term memory stores scenario and surgical knowledge (*semantic memory*), learned behaviors of the camera (*procedural memory*) and experiences of users over time (*episodic memory*). *Surgical state estimation* is performed in the short-term memory, using the actual state of the environment provided by the *perceptual system* and the surgical knowledge. Depending on the actual state of the task, the *FOA generator* triggers the corresponding camera behavior, which changes the camera view of the patient's abdominal cavity through the *camera motion* module. Finally, the surgeon interacts with the system through an appropriate Human-Machine Interface (*HMI*), which lets the user to adjust the camera view according to his/her preferences. Next, each module of the cognitive architecture depicted in Fig. 2 is described in detail.

A. Long-term memory

Long-term memory is broken down into procedural memory, semantic memory and episodic memory [14]. Firstly, *semantic memory* contains declarative facts that the system "knows" and it is essential for reasoning and deciding autonomously. It contains the surgical knowledge the system needs to identify the actual state of the task, as well as surgical scenario knowledge. Thus, semantic memory is divided into two classes of knowledge: *scenario* and *surgical knowledge*. This knowledge division allows the system to organize the memory into different mental structures. The *scenario knowledge* contains static information of the different objects of the surgical scenario. This information is required to classify the objects and to establish the relations between them. Therefore, each object of the scenario knowledge is defined as a data structure, which first field is the object's class. In this work we have considered four object's classes: camera (O_1), sensor (O_2), surgical tool (O_3), and image (O_4). Objects definition is as follows:

- O_1 = {*camera*, ${}^nT_c$}
- O_2 = {*sensor*, ${}^0T_s$}
- O_3 = {*surgical tool, marker*, ${}^mP$}
- O_4 = {*image*, ${}^IR_c$, *α*, *aspect ratio*, *H*}

Transformation matrixes between the external robot end effector and the camera (${}^nT_c$), and between the external robot base and the 3D tracker (${}^0T_s$) are static and known prior to the surgical procedure. This information will be used to compute where the position tracker is with respect to the camera. The 3D tracker acquires the position and orientation of a reflective marker attached at the surgical tools. In a common surgical procedure, several surgical tools are required, so for each of them the system needs to know which marker has been attached to it and the tip position with respect to the marker reference frame (${}^mP$). In this way, the system will be able to compute the tool's position using the marker's information from the 3D tracker. As shown in Fig. 3, the image provided by the camera robot has its own reference frame *{I}*, with the singularity that images are in 2D. So rotation matrix between *{I}* and *{c}* is defined as the following 2x2 matrix:

$$\begin{pmatrix} x_I \\ y_I \end{pmatrix} = {}^IR_c \begin{pmatrix} x_c \\ y_c \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \end{pmatrix} \quad (1)$$

The field of view of the camera can be defined by two properties: the angle of view (α) and the *aspect ratio*. The angle of view is the angular extent of a given scene, while the aspect ratio describes the proportional relationship between the image width (w) and height (l). The last field of the image object, H, is the distance from the camera lens to the scene, i.e., the abdominal cavity depth, and it is used to compute the image width and height:

$$w = 2 \cdot H \cdot tan(\alpha/2) \quad (2)$$

$$l = w/aspect\ ratio \quad (3)$$

Using these data, position of *{c}* frame with respect to the image frame (${}^IO_c$) is:

$${}^IO_c = (w/2 \quad l/2) \quad (4)$$

On the other hand, the *surgical knowledge* represents the basis for the surgical state estimation algorithm. It contains the surgical task model and a surgeons' gestures library. Most surgical tasks, as suture, can be modeled as a state-transition diagram, where the overall task is divided into a sequence of basic actions called gestures. In this work, we have considered a suturing task which can be divided into six states: *start, stitching, pulling out, knot tying, thread cutting*, and *end*. *Start* represents the initial state of the task; *stitching* involves inserting the needle in the tissue with the right grasper while pressing on the tissue with the left grasper; after stitching, the *pulling out* state involves extracting the needle with the left grasper and pulling out to pass the thread through the tissue, while pressing on with the right tool; afterwards, in the *knot tying* state a loop is created with the right grasper, and the remaining thread is passed through the loop; finally, the right grasper is substitute by a scissors tool for *thread cutting*, which leads to the *end* state. Fig. 4 shows the state-transition diagram of the suture task described above, while transitions between states are

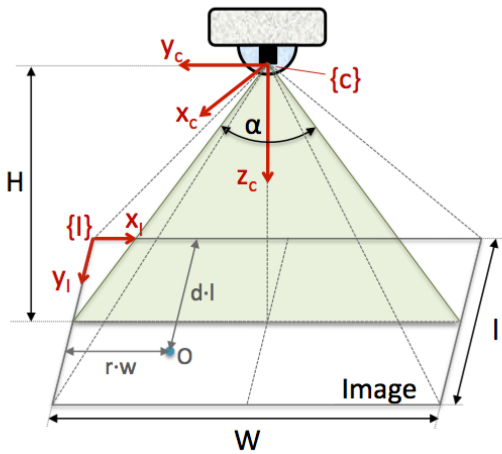

Figure 3. Image parameters

described in Table I. Each state of the diagram is characterized by a surgeon's gesture, so transitions to the next state are triggered when a gesture has been completed, except the first transition, which is triggered by a voice command when the surgeon is ready to start the procedure. Surgeon's gestures are modeled using Hidden Markov Models (HMMs) because of its high flexibility for representing the surgeon's behavior. Gestures patterns are obtained in a training phase, where a gesture's library is built. This library is used to match the data acquired by the 3D tracker concerning the motion of the instruments handled by the surgeon with the data trained offline.

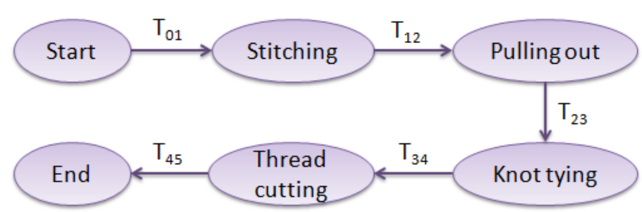

Figure 4. Suture Task Model

TABLE I. TRANSITIONS DESCRIPTION

Transition	Description
T_{01}	Voice command '*start*'
T_{12}	Recognition of *stitching* gesture
T_{23}	Recognition of *pulling out* gesture
T_{34}	Recognition of *knot tying* gesture
T_{45}	Recognition of *thread cutting* gesture

Secondly, *procedural memory* contains knowledge of how to perform particular behaviors of the camera robot. Behaviors are functions containing camera robot primitives that change the viewpoint of the image. When positioning the endoscope in a surgical procedure, the main factor to take into account is the position of the instruments. A solution traditionally assumed is to center the surgical tools in the image. Weedy et al. [12] enhanced camera positioning by predicting the movements of the tools, but different surgeons may handle the instruments in different ways, so this solution could lead to positioning errors when it is used by a surgeon who hasn't trained the system. In this work, we have defined a FOA for each task state. A FOA is described by an object O, a position of O in the image, and a zoom factor (*zoom*). This way, we can not only track a surgical tool, but also to determine the area of interest of the task. For example, during stitching, the task takes place at the right side of the tool that is pressing on the tissue, so in this case the FOA would be to position the tool in the left of the image. As depicted in Fig. 3, to position an object O in a desired location of the image, we use two parameters, r and d. These values, both between 0 and 1, are used to define how to the right and how down, respectively, O is in the image. Therefore, only one function is required in the procedural memory, with different inputs depending on the actual state of the task. Steps of the FOA function, which inputs are *{O, r, d, zoom}* are:

1) To acquire tool's marker transformation matrix (${}^{s}T_{m}$) from the 3D tracker.
2) To compute marker orientation matrix (${}^{c}R_{m}$) and origin (${}^{c}O_{m}$) with respect to the camera frame {c}:

$$ {}^{c}T_{m} = {}^{n}T_{c}^{-1}\ {}^{0}T_{n}^{-1}\ {}^{0}T_{s}\ {}^{s}T_{m} = \begin{pmatrix} & {}^{c}R_{m} & & {}^{c}O_{m} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5) $$

   where ${}^{0}T_{n}$ is the direct kinematics of the external robot, and ${}^{n}T_{c}$, and ${}^{0}T_{s}$ are defined in the semantic memory.
3) To calculate tool position with respect to the camera frame (${}^{c}P$):

$$ {}^{c}P = {}^{c}O_{m} + {}^{c}R_{m}\,{}^{m}P \quad (6) $$

4) To correct image width and height with the actual camera zoom (*actual_zoom)*:

$$ w' = w/actual\_zoom \quad ; \quad l' = l/actual\_zoom \quad (7) $$

5) To compute tool position with respect to the image frame (${}^{I}P$), in 2D:

$$ {}^{I}P = {}^{I}O_{c} + {}^{I}R_{c}\,{}^{c}P \quad (8) $$

6) Desired tool position (${}^{I}P_{d}$) according to parameters r and d:

$$ {}^{I}P_{d} = \begin{pmatrix} r & 0 \\ 0 & d \end{pmatrix} \begin{pmatrix} w' \\ l' \end{pmatrix} = \begin{pmatrix} r \cdot w' \\ d \cdot l' \end{pmatrix} \quad (9) $$

7) Incremental motion of the robot in the x_0-y_0 plane to place the tool in the desired location in the image:

$$ {}^{o}\Delta P = {}^{0}R_{I}(\,{}^{I}P - {}^{I}P_{d}) \quad (10) $$

   where ${}^{0}R_{I}$ is the rotation matrix between planes x_0-y_0 and x_I-y_I.
8) Finally, digital zoom is applied to provide a camera zoom according to parameter *zoom*.

Finally, *episodic memory* represents the experience of users over time. It stores a users' profile library containing the users' preferences as regards to the camera viewpoint for each task state. When the system is started, it asks for a user name. If the user name does not exist in the profiles library, a new profile is created with the default FOA parameters (O, r, d, and *zoom*) for each state. During the operation, the user interacts with the system through voice commands. The HMI is provided with six voice commands: *left*, *right*, *up* and *down*, to change the camera position, and *zoom in* and *zoom out* to change the zoom factor of the image. This way, the user can communicate the system his/her preferences for each state. When a motion voice command is activated, the external robot moves the camera a fix quantity (*inc_motion*) in the corresponding direction. This predefined quantity is corrected with the actual camera zoom in order to provide a coherent motion in the image:

$$ inc\_motion' = inc\_motion/zoom \quad (11) $$

Then, the system acquires the corresponding tool's marker position and computes the position of the tool's tip using equations (5)-(8). Afterwards, parameters r and d of the actual task state are updated as follows:

$$ \begin{pmatrix} r \\ d \end{pmatrix} = \begin{pmatrix} 1/w' & 0 \\ 0 & 1/l' \end{pmatrix} {}^{I}P \quad (12) $$

On the other hand, when a zoom voice command is activated (*inc_zoom*), the zoom factor of the actual state is updated as:

$$ zoom = zoom \cdot inc\_zoom \quad (13) $$

Therefore, after the task is performed the user's profile has been update with his/her particular preferences. When the user performs the task again, the system loads his/her profile, instead of the default FOA parameters, thus decreasing the number of voice commands the user has to communicate to the system.

B. Short-term memory

Short-term memory represents the inference engine of the system, where reasoning and deciding procedures take place. This memory is divided into two blocks: *surgical state estimation* and *FOA generator*, which triggers the corresponding robot behavior from the perceptual memory according to the actual state of the task.

The *surgical state estimation* module requires a model of the task being performed (from the *surgical knowledge*) and a recognition system to evaluate the transitions among the different states. As described in the previous subsection, the task model is a state-transition diagram, which transitions are triggered when a particular gesture is recognized. The *surgical state estimation* module acquires data of the surgical tools from the perceptual system (3D tracker), and outputs the actual state of the task. Surgeons' gestures are modeled using Hidden Markov Models (HMMs), a stochastic technique commonly used to evaluate surgeon's skills [15] and to predict surgical states [16]-[17]. As described in previous works [18]-[19], a surgical gesture is associated with a pattern λ_k described by the following parameters: $\lambda_k = (S, E, A, B, \pi)$, where S is the set of basic actions that characterized a particular gesture, E is the set of observable characteristics that describe each basic action, A is the probability distribution matrix relating the basic actions, B is the probability distribution matrix establishing the most probable observable characteristic in each state, and π is the initial states distribution. In this work, the observable characteristics are defined by the interaction between the surgical tools, namely, the tip's distance, the angle between the tools and their velocities. Gestures patterns λ_k are trained off-line to build the gestures library. During the operation, the recognition system identifies a certain gesture by matching the observable characteristics sequence (E) acquired form the tracker with the gestures library. When a gesture is recognized, a transition state of the task model is triggered.

The *FOA generator* inputs the actual state of the task and outputs the FOA parameters of the actual state to the *procedural memory*, which triggers the corresponding FOA function to provide the optimal viewpoint for the actual state. Therefore, the *FOA generator* consists of a table relating each task state with its corresponding FOA parameters. Default values for our particular work are described in Table II. For the stitching state, the FOA is on the right side of the left grasper, as it is the area where the needle will be inserted, with a zoom factor of 1.25; for pulling out, zoom is turned off and the right grasper, which is pressing on the tissue, is placed at the right side of the image, as pulling out of the needle is performed with the left tool; for knot tying, the left grasper, which holds the needle, is placed at the centered of the image in the horizontal side and at the bottom in the vertical one, as the knot will be performed in the upper side; finally, for thread cutting, the right grasper is substitute by a scissors, and the left grasper, which still holding the needle, is centered in the image. For the two last states, a zoom factor of 1.5 is applied, as these are the most challenging states.

TABLE II. FOA GENERATOR (DEFAULT PARAMETERS)

Task state	**FOA parameters**			
	O	***r***	***d***	***zoom***
Stitching	Left grasper	0.4	0.6	1.25
Pulling out	Right grasper	0.75	0.6	1
Knot tying	Left grasper	0.5	0.7	1.5
Thread cutting	Left grasper	0.5	0.5	1.5

IV. EXPERIMENTS

A. Implementation

The experimental set-up is shown in Fig. 5, along with a camera snapshot in a certain instant. The external robot is a 7 DOFs Barrett WAM (Barrett Technology, Inc.), a cable-driven robot which exhibits zero backlash and has low friction and low inertia. The 3D Tracker is a Polaris Spectra (NDI), a real-time 3D optical tracker used in a variety of surgical applications. The camera properties are: angle of view of 50º, aspect ratio of 16:9, and resolution of 1280 x 720 pixels. The height of the abdomen simulator is 220 mm. The cognitive architecture has been implemented in MATLAB 2013. Video display is integrated in the cognitive architecture program, using a MATLAB Timer Object with a period of 0.04 seconds, so camera image is displayed at 25 fps. The *surgical state estimation* algorithm requires a high computational time, as it has to acquire data from the 3D tracker in real-time. In order to not compromise the video display quality, this module has been performed in an external computer. Communication between the cognitive architecture and the computer running the *surgical state estimation* is performed by *User Data Protocol (UDP)* protocol. On the other hand, the Barrett WAM is provided with a built-in real-time control library (*libbarrett-1.2.1*) written in C++. Communication between the cognitive architecture and the WAM is performed by *TCP/IP* protocol. The architecture sends the incremental motion of the robot output by the FOA function in the WAM reference frame. The open-source framework ROS [20] is used to receive this data and send it to the WAM node.

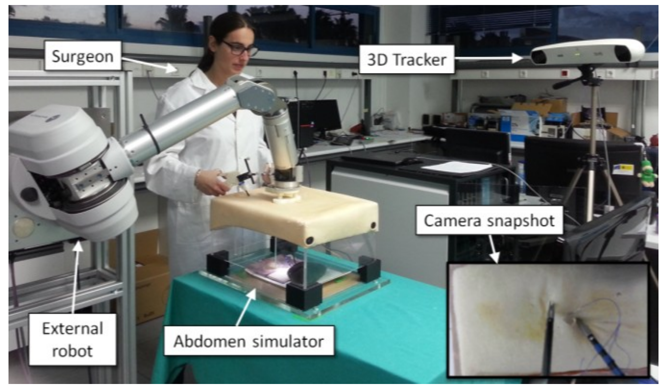

Figure 5. Experimental set-up

B. Experimental results

To test the cognitive architecture described in section III, a set of in-vitro experiments have been performed. Five non-expert users have been asked to perform a suture task according to the task model described in Fig. 2. First, users have performed three trials of the task using just voice commands to get the optimal view of the operating area in each state. Number of voice commands and overall time have been recorded. Results are reported in Table III, where T_1, T_2 and T_3 represent the users' trials, and *mean* is the mean parameter value of the three trials. Secondly, users have been asked to perform the same task but using the cognitive architecture proposed in this work. Results are shown in Table IV. As regards to the number of voice commands, comparing mean values of Table III with results of the first trial of Table IV, it can be seen that performance of the task with the cognitive architecture requires less voice commands, as the system provides a predefined camera view for each state in an autonomous way. After the first trial, the system learns the preferences of users through the *episodic memory*. Consequently, the number of voice commands in the third trial is reduced in a 100%, 67%, 60%, 85% and 67% for user 1, 2, 3, 4 and 5, respectively.

With respect to time, results do not show a direct relation between the use of the cognitive architecture and the overall performance time. While some users reduce considerably the overall time (as *user 1*), others increase it (as *user 2*). This is mainly due to the inherent complexity of the task, which makes that time widely varies even in different trials of the same user, and also due to that users have to adjust the camera view in the first trial when using the cognitive architecture. However, results do show that with the cognitive architecture, overall time is reduced in a 47%, 8%, 21%, 9% and 15% in the third trial with respect to the first one, as the system has already learned the users' preferences.

TABLE III. EXPERIMENTAL RESULTS USING JUST VOICE COMMANDS

User	# Voice commands				Time (s)			
	T_1	*T_2*	*T_3*	*Mean*	*T_1*	*T_2*	*T_3*	*Mean*
1	16	12	11	13	132	87	87	102
2	12	10	11	11	123	107	85	105
3	7	9	8	8	94	90	80	88
4	14	10	8	11	104	119	103	109
5	11	9	8	9	127	95	95	106

TABLE IV. EXPERIMENTAL RESULTS USING THE COGNITIVE ARCHITECTURE

User	# Voice commands			Time (s)		
	T_1	*T_2*	*T_3*	*T_1*	*T_2*	*T_3*
1	4	1	0	88	87	47
2	6	4	2	120	114	111
3	5	4	2	93	87	80
4	7	4	1	100	96	94
5	6	2	2	92	91	83

V. CONCLUSIONS

This paper has presented a cognitive architecture for a camera robotic assistant. The goal of the cognitive architecture is to provide the user the optimal camera view of the operating area for each state of the task in an autonomous way. To this aim, each task state has been characterized by a FOA, which is defined by an object, a position of the object in the image, and an image zoom factor, which delimits the area of interest of a particular state. When a task state is recognized, the system triggers the corresponding camera behavior to change the viewpoint of the scene. The architecture has been tested through a set of in-vitro experiments, aimed at comparing the number of voice commands and the overall performance time when using just voice commands versus using the cognitive architecture. Results show that the number of voice commands is significantly reduced in the second case, thus reducing the user workload, who can focus in the surgical task instead of interacting with the system to change the camera view. Moreover, a learning mechanism has been implemented in order to take into account particular preferences of users. Results show that both the number of voice commands and the overall performance time are reduced for the five users after three trials. The experiments show promising results of the cognitive architecture, which could be used for different tasks models by adapting the FOA parameters and augmenting the system knowledge. The learning mechanism can also be improved by adapting the FOA default parameters for new users taking advantage of experiences of other users.

REFERENCES

[1] R. Taylor, “A perspective on medical robotics,” *Proceedings of the IEEE*, vol. 94, no. 9, pp. 1652–1664, September 2006.

[2] Intuitive Surgical, Inc. (2007). Company profile of Intuitive Surgical Inc. Sunnyvale, CA, USA, Website, URL http://www.intuitivesurgical.com/

[3] D. Sharma, C. Brown, C. Kouriefs, H. Sood, P. Grange, and H. Patel, “Initial experience with the freehand robotic camera holder in laparoscopic urology,” Journal of Endourology, vol. 23, no. 1, p. A249, 2009.

[4] D. Noonan, G. Mylonas, J. Shang, C. Payne, A. Darzi, and Y. G., “Gaze contingent control for an articulated mechatronic laparoscope,” in *Proceedings of the IEEE/RAS International Conference on Biomedical Robotics and Biomechatronics*, pp. 759–764, 2010.

[5] C. Staub, S. Can, B. Jensen, A. Knoll, and S. Kohlbecher, “Human-Computer interfaces for interaction with surgical tools in robotic surgery”, *IEEE/EMBS Int. Conf. on Biomedical Robotics and Biomechatronics,* Rome, Italy, pp. 81-86, 2012.

[6] V.F. Muñoz, I. García-Morales, C. Perez del Pulgar, J.M. Gomez-DeGabriel, J. Fernandez-Lozano, A. García-Cerezo, C. Vara-Thorbeck, and R. Toscano, “Control movement scheme based on manipulability concept for surgical robotic assistant”, *IEEE Int. Conf. on Robotics and Automation,* pp. 245-250, Orlando, Florida, 2006.

[7] S. Haddadin, M. Suppa, S. Fuchs, T. Bodenmüller, A. Albu-Schäffer, and G. Hirzinger, “Towards the robotic co-worker”, *Springer Tracts in Advanced Robotics,* Vol. 70, pp.261-282, 2011.

[8] M.G. Jacob, Y-T. Li, J.P. Wachs, “Gestonurse: a multimodal robotic scrub nurse”, *ACM/IEEE Int. Conf. on Human-Robot Interaction*, Boston, MA, pp. 153-154, 2012.

[9] N. Padoy, and G.D. Hager, “Human-machine collaborative surgery using learned models”, *IEEE Int. Conf. on Robotics and Automation*, pp. 5285-5292, Shangai, China, 2011.

[10] E. Bauzano, B. Estebanez, I. Garcia-Morales, and V.F. Muñoz-Martinez, “Collaborative Human-Robot System for HALS suture procedures”, *IEEE Systems Journal,* Vol. PP (99), pp. 1-10, 2014.

[11] S.Y. Ko, J. Kim, D.S. Kwon, and W.J. Lee, “Intelligent interaction between surgeon and laparoscopic assistant robot system”, *IEEE Int. Workshop on Robots and Human Interactive Comunication,* pp. 60-65, 2005.

[12] O. Weede, H. Monnich, B. Muller, and H. Worn, “An intelligent and autonomous endoscopic guidance system for minimally invasive surgery,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 5762 –5768, 2011.

[13] P. Langley, J. E. Laird and S. Rogers. Cognitive Architectures: Research Issues and Challenges. Cognitive Systems Research, Vol. 10 (2), pp.141- 160, 2009.

[14] R. Salgado, F. Bellas, P. Caamano, B. Santos-Diez, and R.J. Duro, “A procedural Long Term Memory for cognitive robotics”, *2012 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS),* Madrid, pp. 57-62, 2012.

[15] 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 Trans. Biomedical Engineering, vol. 53, no. 3, pp. 399-413, 2006.

[16] O. Weede, D. Stein, B. Müller, and H. Wörn, “A cognitive path-guidance-system for minimally invasive surgery”, *IEEE Int. Symposium on Intelligent Systems and Informatics,* pp. 139-144, Subotica, Serbia, 2010.

[17] C.E. Reiley, E. Plaku, and G.D. Hager, “Motion generation of robotic surgical tasks: learning from expert demonstrations”, 32nd *Annual Int. Conf. of the IEEE EMBS*, pp. 967-970, Buenos Aires, Argentina, 2010.

[18] B. Estebanez, P. del Saz-Orozco, I. Rivas-Blanco, E. Bauzano, V. Muñoz, and I. Garcia-Morales, “Manuevers recognition in laparoscopic surgery: artificial neural network and Hidden Markov model approaches”, *IEEE RAS/EMBS Int. Conf. on Biomedical Robotics and Biomechatronics,* pp. 1164-1169, Rome, Italy, 2012.

[19] B. Estebanez, P. del Saz-Orozco, V. Muñoz, and I. García-Morales, “Maneuvers recognition system for laparoscopic surgery”, *IEEE 19th Int. Workshop on Robotics in Alpe-Adria-Danube Region (RAAD),* pp. 87-92, 2010.

[20] M. Quigley, K. Conley, B. Gerkey, J. Faust, T.B. Foote, J. Leibs, R. Wheeler, and A.Y. Ng, "ROS: an open-source Robot Operating System", International Conference on Robotics and Automation, ser. Open-Source Software workshop, 2009.