BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **21**, No 1
Sofia • 2021 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2021-0009

Design of a Multifunctional Operating Station Based on Augmented Reality (MOSAR)

Veronika Ivanova[1]*, Plamen Vasilev*[2]*, Ivilin Stoianov*[3]*, Rumen Andreev*[4]*, Ani Boneva*[4]

[1]*Institute of Robotics, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria*
[2]*Industrial Automation University of Chemical Technologies and Metallurgy, 1756 Sofia, Bulgaria*
[3]*Istituto di Scienze e Tecnologie della Cognizione, Consiglio Nazionale delle Ricerche, Padova, Italy*
[4]*Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria*
E-mails: iwanowa.w@abv.bg plamen.vasilev@uctm.edu ivilinpeev.stoianov@cnr.it rumen@isdip.bas.bg a.boneva@isdip.bas.bg

Abstract**:** *Design principles of a novel Multifunctional Operation Station* (*MOS) using Augmented Reality* (*AR*) *technology* (*MOSAR*) *are proposed in this paper. AR-based design allows more ergonomic remote instrument control in real time in contrast to classical instrument-centered interfaces. Another advantage is its hierarchical software structure including multiple programming interpreters. The MOSAR approach is illustrated with a remote surgical operating station that controls intelligent surgical instruments. The implementation of the Operation Station* (*MOS*) *is based on the multiplatform open-source library Tcl/Tk, and an AR extension has been developed on a Unity platform, using Vuforia SDK.*

Keywords**:** *Multifunctional Operating Station* (*MOS*)*, Virtual Reality* (*VR*)*, Augmented Reality* (*AR*)*, Tcl/Tk, Surgical Robotics.*

1. Introduction

The purpose of this article is to lay down design principles and principle approaches for the implementation of remote-control operating stations Multifunctional Operation Station (MOS), with a specific focus in the surgery domain. Following a description of the state-of-the-art in the domain, we first present design principles for MOS implementation based on the solid Tcl/Tk scripting programming stack. We then discuss new possibilities for MOS provided by Augmented Reality (AR) and Virtual Reality (VR), and develop an extension of the MOS design based on AR/VR (MOSAR) and one demo implementation based on a modern gaming platform, Unity which includes extended AR/VR capabilities and other useful characteristics such as physical simulation of real objects and their interactions [1-5].

Most controlled processes require real-time control. An operating station is a basic tool which serves for the purpose of process monitoring and control. It must allow multitasking, graphical visualization of live audio-video streams, instrument controls, instrumental measurements, and other types of information [1-3, 5]. Data transmission may include different communication protocols designed for IoT communication (e.g., [6]). Multifunctional operator platforms for control and monitoring have a variety of application domains, including interactions with industrial devices [2], intelligent instruments [7] and healthcare, including surgery [3, 4].

In the surgery domain, operator stations are mainly needed to perform remote-controlled surgeries by providing, on one hand visual context and operative information about the state of the patient, organ information, etc, and on the other hand, real-time instrument measurements and delivering control commands. An example in laparoscopic surgery is illustrated in [8]. Other applications of operating stations in surgery include planning surgeries, training, reviewing past surgeries. Indeed, training medical specialists by using simulated surgeries is critical for gaining high-quality surgery skills [9, 10]. Those applications require live video of the surgery, stored videos of past surgeries, or simulated anatomy.

The control software of an operating station should operate under operating systems embedding real-time processing. A number of platforms could be used for developing and implementing the structure and functions of the control software. It could extend open source libraries written in script languages such as Tcl/Tk, Pearl and Python. The libraries could provide resources from the operating system and additional complex functionalities such as image processing, pattern recognition, image co-registration (i.e., spatial localization of multiple types of information), etc.

Notably, operating stations also fall in the domain of application of Virtual-Reality/Augmented-Reality (VR/AR), which is originally developed for the military and gaming industry [11-13**]** but nowadays it is applied also in other domains. For example, enhanced and virtual multimedia environments are ideal tools for the development of systems for improving the knowledge and practical skills of students and practitioners in many medical fields. Such systems could contribute to maintaining and improving the level of knowledge and practical skills of specialists, and introduce them to up-to-date scientific achievements and modern technologies. More general applications of VR/AR include e-Learning [14] and rich human-machine interfaces including real time communication of multimedia content in robotics [15, 16].

A more recent approach for providing even more intuitive and rich graphical user interface is to exploit the extended 3D-graphical modelling and visualization capacities of general-purpose gaming engines such as *Unity* and *Unreal*. Those provide extensive multi-platform libraries written in OOP languages ($C^{\#}$ or C^{++}) and allow high frame-rate and high-definition 3D-visualizations needed for the implementation of VR/AR applications. Indeed, there are already some industrial AR/VR applications based on such platforms [17].

The paper is further organized as follows: Section 2 describes the design principles of a MOS. Then Section 3 describes the Tcl/Tk software stack that could be used for basic MOS implements and Section 4 briefly describes one MOS implementation in the surgery domain. In turn, Section 5 outlines characteristics of Virtual and Augmented Reality and introduces applications of AR in the medical domain, along with relevant characteristics of the Unity gaming engine. Section 6 then describes the design principles of an AR-based MOS, MOSAR and then Section 7 describes a sample Unity-based AR application for the surgery domain. Finally, conclusions are provided in Section 8.

2. Design principles of MOS

Firstly, we describe the functional requirements and structure of a classical MOS for control and monitoring of modular systems, with focus in the medical domain, in particular, surgeriy [1].

A MOS is an intelligent control device aiming to provide multiple functionalities. Those include collecting and integrating readings of various sensors and other types of descriptive information (e.g., derived from video analysis), delivering control signals to manipulators and actuators, an Artificial Intelligence system functioning as an operator adviser, communication with an external authorized remote user via Internet, and so on. The MOS requires a reliable software-and-hardware implementation and capabilities to acquire, process, and store large-volume real-time data as it creates and maintains seamless interfaces with the managed processes. The requirements for the Operator Station, included in the control systems, presuppose reliable software implementation and possibilities for operation of the station with large data sets in real time, as it creates and maintains an uninterrupted interface with the processes being managed. This device, used in the field of Robotics and especially Medical Robotics, must be able to provide the operator with full control over the objects being controlled and opportunities for maximum information about the actions performed and their results. However, in the domain of intelligent medical tools, that is, devices provided with a certain degree of capabilities to take autonomous decisions, a MOS must be able to provide the operator with full control over the management facilities and the ability to maximize his/her awareness of the performed actions and their results.

Following this analysis, we conclude that MOS should support the following basic functions:

- Communication with close and remote devices via RS232, RS485, CAN, GPRS/3G/4G, LAN, WLAN, Bluetooth. Receive data and remotely manually or automatically control the devices via the above communication channels. Ability to remotely update the firmware of devices;
- Support of a database with instantaneous instrument readings and other types of information, e.g., live video;
- Ability to perform manual or automated pre-processing of raw instrument readings received from devices before sending information to storage and post-processing after receiving data from databases;
- Real-time visualization of readings from devices and live video through a GUI or a console;

- Generating various reports from the system archive.

Generation of alarm messages for violations having been occured due to technological processes or communication problems.

In principle, a MOS is organized into the following implementation levels: hardware, low-level control, high-level control, remote control, and applications. Each of these can provide the following development levels: Bases, object-oriented, and element-based. The Bases level includes applications of low-level languages tools and avoids dependencies on external resources. The object-oriented level consists of language tools that implement most of the functionality.

The MOS system could operate on a portable computer equipped with a wireless LAN and gateway capabilities to connect all relevant devices.

The software of the Operating station should support Graphical User Interface (GUI), a global Database, a generator and library of Macro commands, an Image visualizer, an Expert system, wireless LAN and WiFi infrastructures, a TCP client-server connectivity.

Thus, the MOS architecture should include the following functional blocks (Fig. 1).

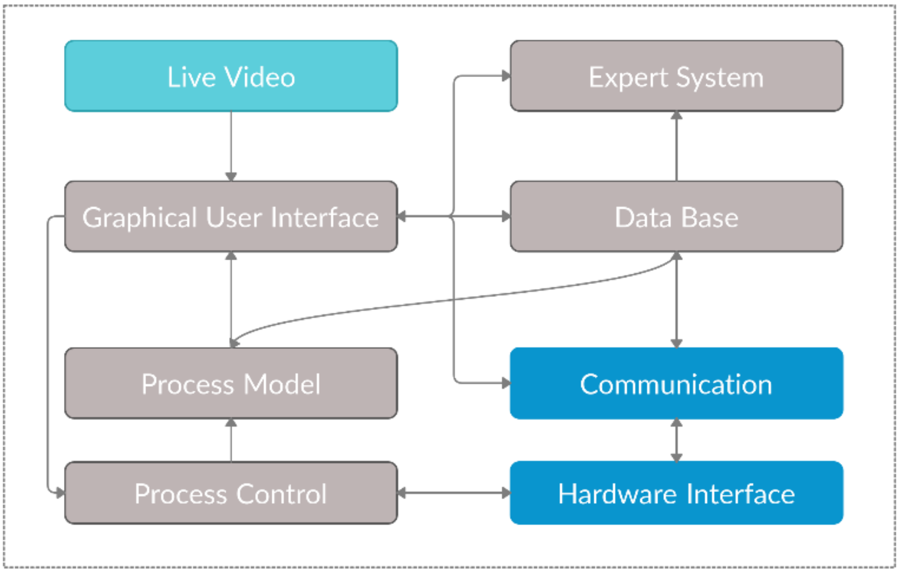


Fig. 1. MOS: basic architecture

(i) **Graphical User Interface (GUI)** providing a system-level interface to control technological processes using programming or graphical tools. The GUI is divided into two parts: first, a *Configurator*, which is a graphical editor for adjusting displays of mnemonic circuits (resolution, access level, etc.), using vector libraries of graphic elements. Graphics must be able to define a general description and behavior depending on the values of certain signals in the configuration table. It is recommended to use IEC/ISO 11581 User system interfaces and symbols for the layout of the graphic elements. Second, the GUI includes an *Operational part,* used to start the configured mnemonic displays and human-MOS interaction, or control. The GUI facilitates the work of the operator in generating command procedures and visualizing the results of their execution.

(ii) **Expert System block** providing domain-specific decision support (here, for example, potential medical diagnoses) based on artificial intelligence techniques. Typically it uses a database of expert knowledge. The necessary components for this block include: (a) Knowledge Database with facts, rules and restrictions; (b) Processing component (Interpreter) – used for signal processing and providing logic functions. The Expert System block interacts with the Graphical User Interface that visualizes possible solutions in a meaningful manner with selection possibility (pop-ups, radio buttons, etc.).

(iii) **Live Video block** includes image processing tools for real-time visualization. The live video unit can be developed as a separate functionality included in the graphical user interface (configurator);

(iv) **Database block** supporting the work with a number of embedded Database packages;

(v) **Process Model block** – it contains a program model of the processes being controlled. The operator could use this model for training before starting to operate on real external devices, controlled through the GUI and the Hardware Interface.

(vi) **Process Control block**– it is used as an internal control mechanism of the program processes and it is based on the use of embedded software state machines (finite automata structures). This block realizes logical operations depending on analog signals level, presence/absence of a discrete signal, reaching counter or timer limit value. For state machine configuration purposes, it is possible to use a built-in module to implement UML (Unified Modeling Language) [18], state diagrams that subsequently generate a pseudocode.

(vii) **Hardware Interface block** provides the connection of the operating station to external devices by means of wired or wireless link. This block enables selection, setting and configuration of interfaces, i.e., Ethernet/IP, CAN, OPC UA, Fieldbus, Profibus DP/PA, Modbus, 3G/4G, etc.

(viii) **Communication block** provides communication with a remote client or server over the Internet. It creates and maintains a bidirectional information channel with external networks, databases and clients. The unit is intended to provide hardware connectivity for I/O devices. Configured physical and logical signals must be entered into a configuration, communication table in the database and be accessible to all other blocks.

All described functional blocks in the form of Implementation Model could be designed using the Tcl/Tk program stack and libraries, also using multiple independent threads. When hierarchical or other types of dependencies between different functions exist, those could be realized using hierarchical structures. Such solutions are useful as they allow one host machine program process to be decomposed to a set of concurrent logical program processes, each of them designed as a Tcl/Tk procedure. Other benefits of this approach are the capabilities for data protection and restricted access to different block resources.

3. Tcl/Tk software stack

The scripting language Tcl/Tk allows developers to access resources of the operating system. It consists of two components: (1) Tcl is a C-like procedure-oriented language, used for standard algorithm programming; (2) Tk - language operators for making requests to the operating system resources and setting the resource parameters. [1, 19, 20]. Tcl/Tk has an embedded mechanism for real-time applications by registering program events and binding Tcl procedures to them [19]. Tcl/Tk also allows linking Tcl/Tk code and libraries with external programs written in C, which allows developers to add novel interfaces.

Typically, a Tcl/Tk application uses the same interpreter for all Tcl and Tk operators. The language model assumes the use of hierarchical structures including more than one interpreter, each of which having their own set of operators and addressing space. At the top of the hierarchy is one fully functional interpreter of Tcl/Tk code – so-called Master interpreter. It can create a number of Slave interpreters and restrict any of their standard commands or create new commands. This process can be continued and the Slaves can create Sub-slaves forming hierarchical structures. [19, 20].

Threads are multiple concurrent flows within the same process. All threads within a process share the same memory and other resources. Tcl/Tk supports multi-threading in order to create applications that include a number of tasks processed in parallel [19, 20]. The operating system handles the details of thread creation and coordination. After the application starts, only one thread is executed, referred to as the main thread. Any thread can create another thread; there is no restriction to starting threads only from the main thread. There is a mechanism for message exchange between different threads allowing task synchronizations. Other useful tools for synchronizations between tasks are shared variables and mutexes. Each thread in Tcl/Tk has its own Main interpreter that is not accessed from other threads.

Sockets are network communication channels. They cover the use of the TCP network protocol, although one can find Tcl extensions that create sockets using other protocols. TCP provides a reliable byte stream between two hosts connected to a network. TCP handles all the issues of routing information across the network and it automatically recovers if data is lost or corrupted along the way. TCP is the basis for other protocols like Telnet, FTP and HTTP. A Tcl script can use a network socket just like an open file or pipeline. It uses the socket command to open a socket. Then the commands *gets*, *puts*, and *read* can be used to transfer data. The *close* command closes a network socket.

Network programming distinguishes between clients and servers. A server is a program that runs for long periods of time and controls access to resources. A client typically connects to the server for a limited time in order to gain access to a specific resource. A client opens a socket by specifying the host address and port number for the server of the socket.

Tcl/Tk applications may use global or local program variables and constants, each of them containing string data. The basic language types of data are related to the string interpretations of the strings from the Tcl/Tk operators: text strings, numerical values, objects, lists of objects, indexed arrays of objects, etc. Tcl/Tk operators could be internal (embedded into the language kernel libraries) or external – names of procedures that are defined and located anywhere in the application. Tcl procedures have a number of formal parameters presented as names of string variables and interpreted in the procedure body. Calling procedures is done by name followed by a list of actual parameters. Tk operators are of type "internal" only and they have a fixed number of named parameters with "default" predefined values. As the parameters are named, an "actual" parameter value could be assigned to each of the parameters. If it is not done by the developer, the operator uses its "default" value [20].

4. A MOS for laparoscopic surgery

In this section we illustrate the design and implementation of a TCL/Tk-based MOS for an experimental device with application in laparoscopic surgery. The software runs on Windows and it implements all the functional blocks shown on Fig. 1.

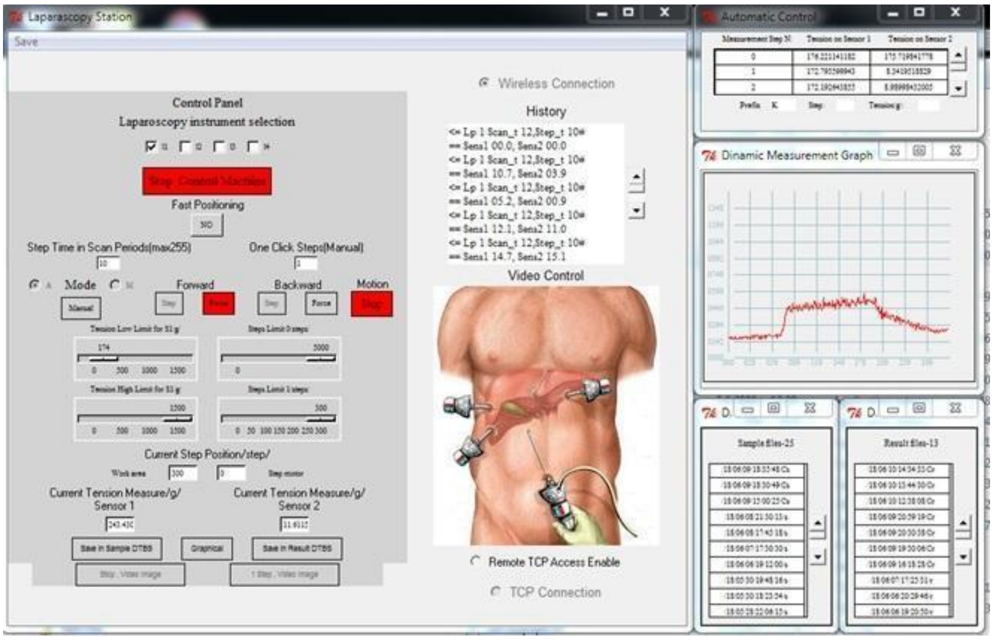
Laparoscopic surgery rapidly evolves as a minimally invasive treatment and has many advantages over conventional surgery: less pain and trauma, shorter recovery time and ultimately better patient outcomes. The interventions are performed using long thin tools that are inserted through a trocar in the human body. The surgeon orients the instruments with the help of laparoscopic images displayed on a monitor. The lack of tactile sense and poor visibility are the most important disadvantages associated with laparoscopy because they limit the surgeon's abilities to examine internal organs of patients. Therefore, visualisation of additional information such as force at the instrument tip is very important for the surgeon to allow him correct manipulations.

For an adequate control of the functionality of each instrument and device, it is important to have an interactive MOS providing a graphical environment for designing, testing, editing and downloading the series of control sequences. The way the managing program is organized influences the movements, the work, the accuracy and the conduction of the experiments, the visual comprehension of the data, and the possibilities for their easy and unambiguous interpretation, comparison and analysis. Therefore, the managing software has to be designed in consideration of the principal requirement to allow appropriate reading and management of the measurements representing the forces exerted by- and on- the laparoscopic instrument.

The implemented operation station offers a graphical user interface providing a variety of graphical tools to allow the operator full control over the instruments being managed. The operator can use program buttons, sliders, data entry, graphical and visualization tools, mode selectors and etc., all embedded into two dialogue windows. There is a possibility for remote transmission of visual and text information by protected e-net channel to remote server. Additionally, an external computer could be connected to the gateway using USB in order to control and monitor the attached instruments event without the MOS. The program can be designed for different kinds of laparoscopic instruments such as a diagnostic tool, therapeutics tool, etc., [21, 22].

Fig. 2 shows the main interface window. On the left side are allocated buttons and sliders for parameter set up while on the right side are visualized measurement results, graphically. The MOS range of commands allows the user to control the device motors, actuators, and sensors for force and position, all of which are connected to a microcontroller. Some of the basic program functions are commands for motion – Start and Stop, command for linear insertion and retraction of the tools. Mode-Automatic and manual, current step positions of the motor, save in samples or save in results, visualization and comparison of the measuring etc. The MOS collects information about different types of measurements at the tip of the tool – surface contact interactions. Previously obtained data from the device is used as a reference to compare with the currently received real-time information. The system graphical visualizes all measurements and quantitatively compares them. This allows the surgeon to give tissue-specific commands on the basis of the measured force interaction between the instrument and the tissue.

In turn, we describe several experiments aiming to demonstrate the operation and the capabilities of the system. Fig. 3 shows the experimental setup, which includes the MOS and a model of a laparoscopic instrument. The experimentation searches for a contact point, detects the presence or absence of contact at the tip of the tool with a surface and measures the interaction force of the tool with the surface [22]. The results are visualized graphically and saved in a database. The measurements could be compared with other data previously received by the MOS.



☑ ☐ ☐ ☐

Step Time in Scan Periods(max255): 10

One Click Steps(Manual): 1

● A Mode ○ M

Tension Low Limit for S1 g: 174 (0 500 1000 1500)

Steps Limit 0 steps: 5000 (0)

Tension High Limit for S1 g: 1200 (0 500 1000 1500)

Steps Limit 1 steps: 300 (0 50 100 150 200 250 300)

Current Step Position/step/

Work area: 300 0 Step motor

Current Tension Measure/g/ Sensor 1: 243.420

Current Tension Measure/g/ Sensor 2: 11.6112

```
<= Lp 1 Scan_t 12,Step_t 10#
== Sens1 00.0, Sens2 00.0
<= Lp 1 Scan_t 12,Step_t 10#
== Sens1 10.7, Sens2 03.9
<= Lp 1 Scan_t 12,Step_t 10#
== Sens1 05.2, Sens2 00.9
<= Lp 1 Scan_t 12,Step_t 10#
== Sens1 12.1, Sens2 11.0
<= Lp 1 Scan_t 12,Step_t 10#
== Sens1 14.7, Sens2 15.1
```

Measurement Step N	Tension on Sensor 1	Tension on Sensor 2
0	176.221141182	175.719841778
1	172.795599943	8.3419518829
2	172.192643835	8.89998432005

Prefix K Step Tension g

Fig. 2. The GUI of a Tcl/Tk operator station for laparoscopic instruments

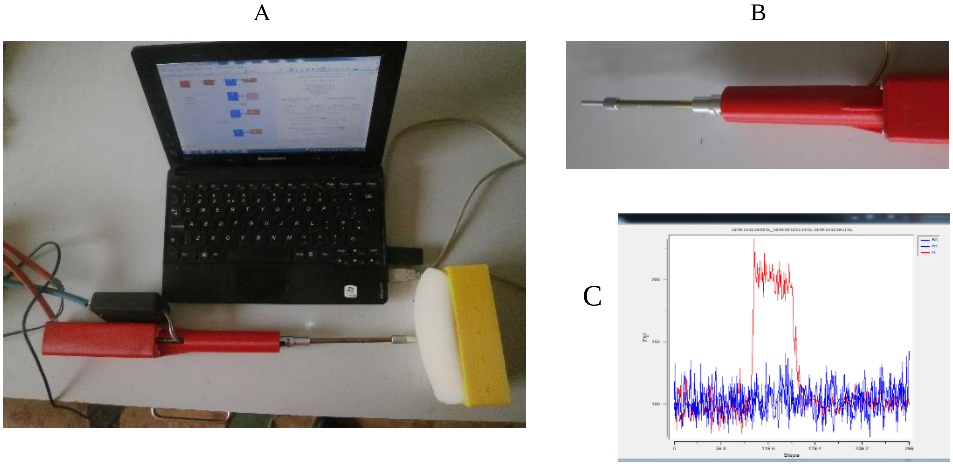


Fig. 3. An experimental implementation model of a laparoscopic instrument and an operator station (A); The contact tip of the laparoscopic instrument (B); Force measured during controlled movement of the instrument in two types of organ tissues (C)

First, Fig. 3C shows the results of an experiment applying the instrument on a muscle tissue (red line) and a control (blue line) [23]. The instrument is initialized as follows: Force set to 0.98 *N*; number of steps: 300; length of micro-step: 24 micrometers. The step of the motor is shown on axis *X*, while the measured force is shown on the axes *Y*. Fig. 3B shows that the minimal observed force is 0.45N, measured from step 0 to step 100 and from step 170 to step 300. The maximal observe force was 1.94N in 100 steps. In this experiment the amplitude was 2.35 N. The average force is 1.21 N. Human tissue data with sample data, which has minimal force of 0.045 N, maximal force of 0.245 N, and an amplitude of 2.36 N have been compared. The results demonstrate the capacity of the device to correctly individuate point of contact with a tissue. Thus, the presented MOS can control and measure force, deformation of tissues, and other tools-objects force interactions.

5. Augmented reality

Augmented Reality (AR) allows to shift the focus during real-time remote control from instruments and gauges to the reality on which the instruments operate. To this aim, live video of anatomy targets is not just displayed on a separate monitor or in a window surrounded by gauges and measurements, but it is the main visual interface on top of which instrumental readings and other types of useful information are displayed. High-definition and high frame-rate AR allows the operators to focus their attention on the task and to precisely remote-control moving instruments instead of frequently to shift attention among videos, pictures, and instruments. Keeping focus on the moving instruments could prevent critical errors during live surgery manipulations. In the AR approach for control stations, the operator does not need to search for relevant information regarding the current manipulation but he/she is provided with context-relevant information right on place. Experimental studies have shown high accuracy in performing robot-assisted surgery due to optimization of real and virtual technology based on augmented reality [24].

Another useful functionality of the AR interface is the possibility to display expected effects of the ongoing manipulation, which could be displayed in terms of instrument trajectory and potential likely outcomes, for example, imminent damage of a sensitive tissue. To be able to do so, the system should be equipped with an image recognition (perceptual) module that should, first, dynamically interpret the input visual stream, and second, predict the consequences of the remote control of a forward model and visually render them [24-27].

AR control interface can mix multiple sources of dynamic information such as live videos from first-perspective and instrument- mounted cameras, echograph, x-ray, MRI, and so on [5, 28]. Altogether, the set of visual stream(s), context-relevant control instruments, results of perception, and the forward predictions constitute the basics of the AR-based control station. A similar approach could be used for a training station, or adding a training mode to the same AR-control station. In the training mode, the sensory input (visual and auditory) could be substituted by a render of a model of the reality, or Virtual Reality (VR), in consideration of the effects of the control actions in the simulated world. This mixture of AR and VR is also known as Mixed-Reality, which operates in modality spanning the AR-AR continuum according to the current requirement

Augmented reality is perfectly suitable for healthcare where it has found multiple applications. Crucial information to a doctor or surgeon could be provided without having direct contact with patients. The applications of Augmented Reality Technologies in healthcare include minimally-invasive surgery, planning, practice, and training of surgical procedures. The first known augmented reality system was created in the U.S. Air Force laboratories in 1992. Its main target was enhancing human performance during surgical procedures. The system included a device that had filmed subcutaneous veins and projected the image of the veins onto the skin of the patient. This system had been used to locate veins until 2005.

AR can provide surgeons with patient monitoring data in the style of a fighter pilot’s heads-up display and allow patient imaging records, including functional videos, to be accessed and visualized in overlay. Examples include a virtual **X-ray** view based on prior **tomography** or on real-time images from **ultrasound** and **confocal microscopy** probes, visualizing the position of a tumour in the video of an endoscope, or radiation exposure risks from X-ray imaging devices.AR can enhance viewing a **fetus** inside a mother’s **womb**. **Laparoscopic** liver surgical systems based on the use of AR technologies to view sub-surface tumours and vessels have been developed by companies such as Siemens and Karl Storz.

Augmented reality and similar computerized utilities have been used to train medical professionals. In healthcare, AR can be used to provide guidance during diagnostic and therapeutic interventions, e.g., during surgery. For example, augmented reality applications for medical training in simulating ultrasound guided needle placement is described in [24]. Other studies like that one described in [29] describe that AR technology both improves university students’ laboratory skills and helps them to build positive attitudes relating to physics laboratory work. Recently, augmented reality has been adopted in **neurosurgery**, which requires heavy amounts of imaging before procedures. Some of the methods are described and may be useful in tracking the movements of the surgical instrument in Augmented Reality [16, 30-33].

Unity is a computationally efficient video gaming engine written on top of C$^{\#}$. It can provide support for a number of critical characteristics for MOS, including live video streaming and 2D/3D visualization, advanced image and video processing, dynamic object modelling and visualization, client-server communication, cloud database. More specifically, the Unity Vuforia SDK is the most widely used platform for AR development, with support for leading phones, tablets, and eyewear. Developers can easily add advanced computer vision functionality to Android, iOS, and UWP apps, to create AR experiences that realistically interact with objects and the environment [34]. Vuforia Engine for Augmented Reality applications embodies sophisticated methods and algorithms for detection and evaluation of points of interest [35], optical flows [36], and fiducial markers [37]. Those, along with other standard Unity platform components such as World Anchor and scripting ability present a core of functionalities for development of comprehensive AR products.

The Vuforia SDK Cloud technology gives the possibility of multiple smart devices to observe the same object from different angles at the same time, which allows numerous participants to be engaged. Vuforia Engine Library contains a rich API documentation section, which includes C$^{\#}$ APIs for Unity, C^{++} APIs for iOS, and UWP and Java for Android.

6. Design principles of MOSAR

Design principles of a MOSAR that enhance previous generation MOS-es with two critical elements are proposed: first, GUI based on VR/AR, which qualitatively enhances the user interface. Second, we substitute the Expert System block with a much more powerful Artificial Intelligence system providing an enhanced knowledge base and human-like visual perception capabilities. Thus, the functional architecture described in Section 2 is enhanced here as follows (also see Fig. 4):

(i) **AR/VR-based GUI block.** The AR-based GUI essentially displays the live stream of the operated environment. Critically, it shows in overlay context-dependent enhancements including instrument readings and controls, organ information extracted from the Artificial Intelligence block, and expert advice. Unity is one of the software platforms that could be used for the implementation of the AR/VR.

(ii) **Artificial Intelligence block.** The AI now replaces the classical Expert System decision support system with much more powerful human-like artificial intelligence. It can provide intelligent advice in support of decisions based on real-time analysis of the video stream and in consideration of the state of the instruments. It includes, for example, databases of expert knowledge, a Knowledge Database with facts, rules and restrictions, a deep neural network video pre-processing and interpretation. The AI block interacts with the GUI, computing and displaying plausible solutions in a meaningful manner, and providing the possibility for subjective decisions.

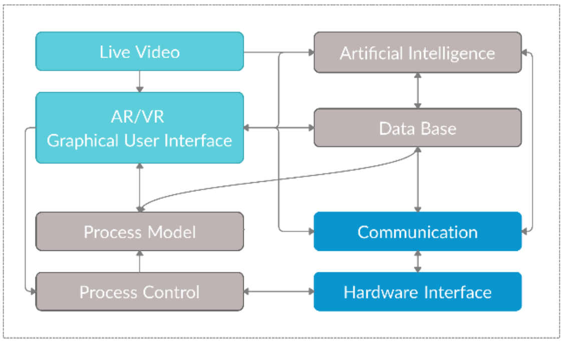


Fig. 4. Multifunctional Operating Station (MOSAR-controller): basic architecture

MOSAR can operate in two regimens, AR and VR. In the Augmented Reality mode, which provides the basic MOSAR functionality, the live video stream is enhanced with instrument readings and tools for control and AI-video analysis. In the Virtual Reality mode, MOSAR is used as a training simulator for surgery, allowing surgeons or practitioners to acquire, develop and maintain the skills required to perform surgical procedures in a range of realistic environments with provided control for safe actions. Moreover, after a live operation or a virtual training session, the surgeons could review their own performance [38-40].

MOSAR can also be developed as a cloud infrastructure application. This requires the use of development tools that allow easy transition from traditional to web-based applications and vice versa, e.g. technologies based on HTML5 or JavaScript such as Cordova or React Native. Cloud-based MOSAR implementations can be installed and launched on different types of mobile and stationary devices from different locations and with different types of network connectivity. Other interesting applications and constructions as well as applications of AR are discussed in [41-43].

7. MOSAR for laparoscopy

To demonstrate the graphical abilities of the Augmented Reality GUI, an application program for MOSAR has been developed ICTPro 1.0 for Android® devices (apk) as an Augmented Reality upgrade of the Multifunctional Operation Station (MOS) for laparoscopic manipulations (described in Section 4). In this preliminary demo version, its principle function is to provide the trainee or the surgeon with a live video stream of the tissue and the laparoscopic instruments during a surgery, enhanced with computer–generated images and accessory information providing information from the laparoscopic instruments, such as tissue elasticity and measurements of the interaction with the body organs, collected along the instrument movement.

The program could be installed on smart devices such as smartphones or smart glasses with OS Android ≥ 7 and exploit their built in MicroElectroMechanical System (MEMS) sensors (accelerometer, gyroscope, camera, solid state compass, GPS, etc.) to evaluate objects and positions situated in the surrounding world. A sample image of the AR interface is shown in Fig. 5. In the demo, the video and the instrumental sets of data have been acquired during a real surgery in a MySQL database.

The MOS maintains prior and acquired information about instruments and other useful information. This information is sent as data to a relational MySQL database that is integrated in MOSAR ICTPro application via a script called LaparoscopicDevice.cs as text or binary data. The data is then augmented to the mixed reality interface as a real-time information about the surgical manipulation process (see Fig. 5). The database includes the following three basic tables (see Fig. 6):

1. *instruments*: containing all characteristics of the medical equipment used;
2. *organs*: contains organ parameters, including probing force, suture force, tissue density, and elasticity;
3. *measurement*: includes organ ID (FK), instrument ID (FK), time stamp, instrument allocation, applied force, movement distance.

Note that this simple database could be further extended with a variety of information, including instrument classification (Equipment Class), Organ Definition, and links towards external (e.g., online) databases.

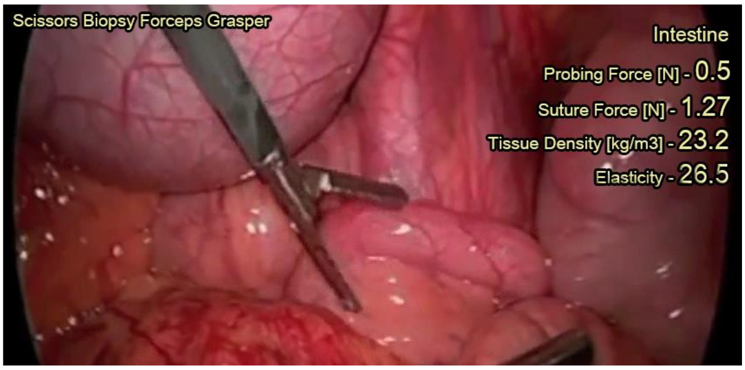
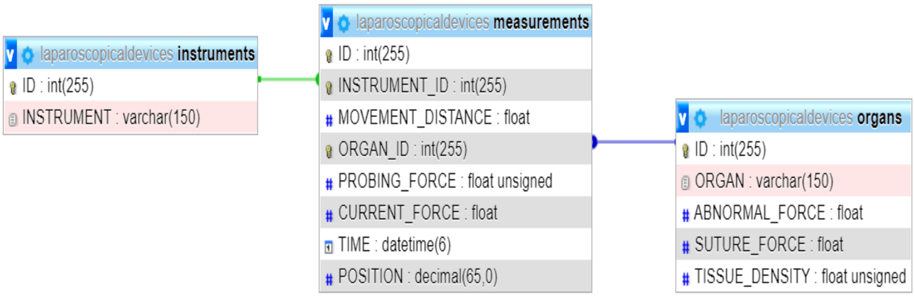


Fig. 5. Frame of AR video stream. The visual interface contains only essential information in order to allow the surgeon to concentrate on its medical task



laparoscopicaldevices **instruments**
ID : int(255)
INSTRUMENT : varchar(150)

laparoscopicaldevices **measurements**
ID : int(255)
INSTRUMENT_ID : int(255)
MOVEMENT_DISTANCE : float
ORGAN_ID : int(255)
PROBING_FORCE : float unsigned
CURRENT_FORCE : float
TIME : datetime(6)
POSITION : decimal(65,0)

laparoscopicaldevices **organs**
ID : int(255)
ORGAN : varchar(150)
ABNORMAL_FORCE : float
SUTURE_FORCE : float
TISSUE_DENSITY : float unsigned

Fig. 6. Laparoscopic devices Database structure

Upon starting, the application initiates a connection to the MySQL database. Then the application starts to acquire live video (e.g., of illustrational image or of live surgery) using the camera of the handheld device, and compares each acquired frame with a key frame from the stored video of the surgery. Falling upon an image with matching points of interest, a video stream is called on top of it and relevant instrument readings and organ characteristics stored in the MySQL database are visualized, matched for timestamp (Fig. 5).

8. Conclusions

This article describes an approach for designing and building a Multi-platform Operator Station based on the high-level script language Tcl/Tk and Augmented Reality. The specific application domain is instrument control in surgery, and more specifically, Laparoscopy. Domain-specific system requirements include the possibility to track tool-tissue force interactions, measuring force interaction and comparing the results with previously obtained information. Intuitive visual interface essentially providing live video of the environment being operated with minimal distraction elements is also crucial, to which requirement we proposed to investigate the more recent Virtual and Augmented Reality technologies.

In general, it is a challenge to design control systems such as MOS. As any other tools used for monitoring and controlling technological processes, MOS is a sophisticated system integrating a variety of subsystems which interact among them and with external information resources. To simplify the developer's work, we propose implementations based on generic and proven technologies such as the open-source Tcl/Tk and its vast set of embedded libraries, and gaming engines providing Virtual Reality capabilities written in versions of the C-language, thus compiled and computationally extremely efficient.

Tcl/Tk allows the possibility of mixing C code with Tcl code, adding and working with precompiled Tcl extension packages, built-in methods for protection of the application data (Safe Tcl), multi-interpreter and multi-threading mechanisms, built-in network connectivity, and multi-platform support. This makes Tcl/Tk a powerful tool suitable for designing complex dynamic objects such as MOS.

There are several promising AR/VR toolboxes that could be integrated with the Unity platform (Version 2017.4.40f1) and which operate on Android OS, among which Vuforia [44] and ARCore [45] are currently most promising [44]. In our preliminary experimentation, Vuforia (Version 7.0.57) provided smoother visual experience and simpler implementation than ARCore (Version 1.20.0), for which we preferred it for our MOSAR demo implementation. However, future versions of the free and fast-developed ARCore might provide sufficient visual quality and developer facilitations as well.

Despite the basic MOS version and its Virtual Reality extension MOSAR have been developed for laparoscopic surgery, they could be adapted to many other application domains where reliability and efficiency are critical factors. For example, control stations of drones, e-Learning, etc., could greatly benefit from control systems based on this technology. More generally, integrating functionalities of an operating control station and augmented reality could become a key new direction for development of handheld applications. Indeed, by using open-source databases (MySQL), libraries (Tcl/Tk), and free (ARCore [45]) or low-cost (Vuforia) VR/AR-engines, software engineers can afford to develop highly attractive cost-effective products.

As a future step we will explore the possibility to use Vuforia SDK to implement a complete MOSAR system including not only a visualization (training/advisory) terminal as the one we presented here but also a fully functional AR Operator Station. This system will be able to provide full onsite or remote control of laparoscopic instruments and other similar medical devices.

Acknowledgments**:** This paper is supported by the National Scientific Program “Information and Communication Technologies for a Single Digital Market in Science, Education and Security (ICTinSES)”, financed by the Ministry of Education and Science and by the European Union H2020-EIC-FETPROACT-2019 Grant 951910 “MAIA” to I.S.

References

1. Ivanova, V., A. Boneva, Y. Doshev, S. Ivanov, P. Vasilev. Multifunctional Operating Station Based on Tcl/Tk and its Applications. – In: Proc. of 6th IEEE International Conference “Big Data, Knowledge and Control Systems Engineering” (BdKCSE’19), 27 February 2020, pp. 1-7, E ISBN: 978-1-7281-6481-6, Print on Demand (PoD) ISBN: 978-1-7281-6482-3, DOI: 10.1109/BdKCSE48644.2019.9010662.
2. Vasilev, P., E. Janev, G. Elenkov. Communication Interface Module for WEB – Based Control – In: Proc. of VIII International Congress “Machines, Technologies, Materials”, Year XIX, Vol. **3**, 2011, pp. 79-82. Print ISSN: 1310-3946 (in Bulgarian).
3. Mountney, P., J. Fallert, S. Nicolau, L. Soler, W. P. Mewes. An Augmented Reality Framework for Soft Tissue Surgery – Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014. – Lecture Notes in Computer Science, 2014, pp. 423-431. ISBN: 978-3-319-10403-4, 8673, DOI:10.1007/978-3-319-10404-1_53.
4. Ha, H.-G., J. Hong. Augmented Reality in Medicine – J. Hanyang Med Rev, Vol. **36**, 2016, No 4, pp. 242-247. pISSN 1738-429X, eISSN 2234-4446. **https://doi.org/10.7599/hmr.2016.36.4.242**
5. Sheng, B., S. Masood, Y. Jung. Chapter Twenty, Virtual and Augmented Reality in Medicine – Biomedical Information Technology, Elsevier Inc, 2020, pp. 673-686. ISBN 978-0-12-816034-3. **https://doi.org/10.1016/B978-0-12-816034-3.00020-1**
6. Ilchev, S., Z. Ilcheva. Internet-of-Things Communication Protocol for Low-Cost Devices in Heterogeneous Wireless Networks. – In: Proc. of 18th International Conference on Computer Systems and Technologies (CompSysTech’7), Ruse, Bulgaria, 23-24 June, 2017, ACM ICPS Vol. **1369**, pp. 272-279. ISBN: 978-1-4503-5234-5, DOI: 10.1145/3134302.3134329.
7. Ilchev, S., Z. Ilcheva. Thermoelectric Cooling Driver for Laser Projection Systems – In: Proc. of Big Data, Knowledge and Control Systems Engineering Conference (BdKCSE’19), 21-22 November 2019, Sofia, Bulgaria, pp. 1-9, IEEE. Electronic ISBN: 978-1-7281-6481-6, Print on Demand(PoD) ISBN: 978-1-7281-6482-3. DOI: 10.1109/BdKCSE48644.2019.9010606.
8. Borissova, D., I. Mustakerov. Methodology for Design of Web-Based Laparoscopy e-Training System. – European Journal of Open, Distance and E-Learning, ERIC Number: EJ954909, 2011, No 2, pp. 1-9. ISSN-1027-5207. **https://eric.ed.gov/?id=EJ954909**

9. Rokhsaritalemi, S., A. Sadeghi-Niaraki, S.-M. Choi. A Review on Mixed Reality: Current Trends, Challenges and Prospects – J. MDPI, Appl. Sci, Vol. **10**, 2020, No 636, pp. 1-26.
https://doi.org/10.3390/app10020636

10. Jagadeesan, J., B. Xavier, F. King, A. Hosny, D. Black, S. Pieper, A. Tavakkoli. A Novel Mixed Reality Navigation System for Laparoscopy Surgery – Med Image Comput Comput Assist Interv, Vol. **11073**, September 2018, pp. 72-80. DOI: 10.1007/978-3-030-00937-3_9.

11. Ajey, L. Virtual Reality and Its Military Utility. – Journal of Ambient Intelligence and Humanized Computing, Vol. **4**, 2013, No 1, Springer, pp. 17-26. DOI: 10.1007/s12652-011-0052-4.

12. Liu, X., J. Zhang, G. Hou, Z. Wang. Virtual Reality and Its Application in Military. – In: Proc. of 2nd International Symposium on Resource Exploration and Environmental Science, IOP Conf. Series: Earth and Environmental Science,Vol. **170**, 032155, IOP Publishing, 2018, pp. 1-8. DOI: 10.1088/1755-1315/170/3/032155.

13. Adrian, I., D. Trandabăţ. Enhancing the Attractiveness of Learning through Augmented Reality. – In: Proc. of International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2018, 3-5 September 2018, Belgrade, Serbia, Procedia Computer Science, Vol. **126**, Elsevier, 2018, pp. 166-175.
https://www.sciencedirect.com/science/article/pii/S1877050918311943

14. Popchev, I. P., D. A. Orozova. Towards Big Data Analytics in the e-Learning Space. – Cybernetics and Information Technologies, Vol. **19**, 2019, No 3, pp. 16-24. DOI: 10.2478/cait-2019-0023.
http://www.cit.iit.bas.bg/CIT_2019/v-19-3/10341-Vol-19_Issue_3-02_paper.pdf.

15. Ma, Z., W. Liu. Data Receiving Method Based on Multimedia Timing in RealTime System – Cybernetics and Information Technologies, Vol. **17**, 2017, No 1, pp. 126-134. DOI: 10.1515/cait-2017-0010.
http://www.cit.iit.bas.bg/CIT_2017/v-17-1/10_paper.pdf

16. Yi, Y., X. Hu. Robot Simultaneous Localization and Mapping Based on Self-Detected Waypoint – Cybernetics and Information Technologies, Vol. **16**, 2016, No 2, pp. 212-221. DOI: 10.1515/cait-2016-0031.
http://www.cit.iit.bas.bg/CIT_2016/v-16-2/10341-Vol-16_Issue_2-16_paper.pdf

17. Mojtaba, N., A. Heydarian, V. Balali, K. Han. Trend Analysis on Adoption of Virtual and Augmented Reality in the Architecture, Engineering, and Construction Industry. – MDPI, Vol. **5**, 2020, Issue 1, 26, pp. 1-18.
https://doi.org/10.3390/data5010026

18. **https://www.uml.org/what-is-uml.htm**

19. Welch, B. B., K. Jones, J. Hobbs. Practical Programming in Tcl and Tk, Prentice Hall Professional. – Computers, 2003, pp. 1-882.

20. **http://www.tcl.tk/**

21. Ivanova, V., D. Bachvarov, A. Boneva, R. Andreev, N. Dobrinkova. System for Analysis and Control of Mechanical Properties of Biological Tissues, Utility Model. Registration No 3323/31.10.2019. p 6293 (in Bulgarian).
www.bpo.bg/images/stories/buletini/binder-2019-11-15.pdf

22. Ivanova, V., D. Bachvarov, A. Boneva. An Advanced Robot System for Diagnostic and Therapeutics Tasks with Application in Laparoscopic Surgery. – Journal of Computer Engineering and Information Technology, Vol. **7**, 2018, Issue 1, pp. 1-9. SciTechnol, ISSN: 2412-8856 (Online), DOI 10.4172 /2324-9307.1000202.

23. Ivanova, V.-A.-G. Laparoscopic Executive Tools for Robots. PhD Thesis, Scientific Field: Mechanical Engineering, Professional Area: Scientific Specialty Robots and Manipulators, Institute of Robotics, Bulgarian Academy of Sciences, Scientific Advisor: Professor Dr. Veselin Pavlov, Asoc. Professor Dr. Ivan Chavdarov, Sofia 2020, pp. 1-136 (in Bulgarian).
http://ir.bas.bg/competitions/atanasova/avto_ata.pdf

24. Itamiya, T., T. Iwai, T. Kaneko. The Holographic Human for Surgical Navigation Using Microsoft HoloLens. – J. EPiC Series in Engineering, Vol. **1**, 2018, pp. 26-30.
https://doi.org/10.29007/wjjx

25. Kim, H. S., N. Hong, M. Kim, S. G. Yoon, H. W. Yu, H.-J., Kong, S.-J. Kim, Y. J. Chai, H. J. Choi, J. Y. Choi, K. E. Lee, S. Kim, H. C. Kim. Application of a Perception Neuron® System in Simulation-Based Surgical Training. – Journal of Clinical Medicine, Vol. **8(1)**, 2019, No 124, pp. 1-11. ISSN: 2077-0383. **https://doi.org/10.3390/jcm8010124**

26. Lee, D., J. W. Yi, J. Hong, Y. J. Chai, H. C. Kim, H.-J. Kong. Augmented Reality to Localize Individual Organ in Surgical Procedure. – Health Information Research, Vol. **24**, October 2018, No 4, pp. 394-401. DOI: 10.4258/hir.2018.24.4.394. **https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6230535/pdf/hir-24-394.pdf**

27. Faure, F., C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, S. Coti. SOFA: A Multi-Model Framework for Interactive Physical Simulation. – Part of the Studies in Mechanobiology, Tissue Engineering and Biomaterials Book Series, Vol. **11**, Springer, Berlin, Heidelberg, 2012, pp 283-321. Online ISBN: 978-3-642-29014-5, DOI: https://doi.org/10.1007/8415_2012_125.

28. **https://developers.google.com/ar/discover**

29. Akçayır, M., G. Akçayır. Advantages and Challenges Associated with Augmented Reality for Education: A Systematic Review of the Literature. – J. Educational Research Review, Vol. **20**, February 2017, pp. 1-11. DOI:10.1016/j.edurev.2016.11.002.

30. Belhaj, S. K., E. Ghribi, Y. Labyed, M. N. Saidi, A. Tamtaoui, N. Kaabouch. Mixed-Reality Aided System for Glioblastoma Resection Surgery Using Microsoft HoloLens. – IEEE Xplore, IEEE, 2019, pp. 079-084. e-ISSN: 2154-0373, e-ISBN: 978-1-7281-0927-5, DOI: 10.1109/EIT.2019.8833738.

31. Pfeiffer, M., C. Riediger, S. Leger, J.-P. K¨uhn, D. Seppelt, R.-T. Hoffmann, J. Weitz, S. Speidel. Non-Rigid Volume to Surface Registration Using a Data-Driven Biomechanical Model. – Computer Science, Engineering ArXiv, 2020, arXiv:2005.14695v1. **https://arxiv.org/pdf/2005.14695.pdf**

32. Mendizabal, A., E. Tagliabue, T. Hoellinger, J.-N. Brunet, S. Nikolaev, S. Cotin. Data-Driven Simulation for Augmented Surgery. Bilen Emek Abali, Ivan Giorgio. – Developments and Novel Approaches in Biomechanics and Metamaterials, Vol. **132**, 2020, pp. 71-96. 978-3-030-50464-9. ff10.1007/978-3-030-50464-9ff. **https://hal.inria.fr/hal-02538101/document**

33. Pellicer-Valero, O. J., M. J. Rupérez, S. Martínez-Sanchis, J. D. Martín-Guerrero. Real-Time Biomechanical Modeling of the Liver Using Machine Learning Models Trained on Finite Element Method Simulations – J. Expert Systems with Applications, Vol. **143**, 2019, No 1, Elsevier, Ltd., pp. 1-12. ISSN 0957-4174. **https://doi.org/10.1016/j.eswa.2019.113083**

34. **https://library.vuforia.com/getting-started/overview.html**

35. Schmid, C., R. Mohr, C. Bauckhage. Evaluation of Interest Point Detectors. International. – Journal of Computer Vision, Springer Verlag, Vol. **37**, 2000, No 2, pp. 151-172. ff10.1023/A:1008199403446ff. ffinria-00548302.

36. Fleet, D. J., Y. Weiss. Optical Flow Estimation. – Handbook of Mathematical Models in Computer Vision, Springer, 2006, pp. 237-257. ISBN: 978-0-387-26371-7.

37. Jurado, S., Garrido, R. Muñoz-Salinas F. J., Madrid-Cuevas M. J., Marín-Jiménez. Automatic Generation and Detection of Highly Reliable Fiducial Markers under Occlusion. – J. Pattern Recognition, Vol. **47**, June 2014, Issue 6, pp. 2280-2292.

38. Talbot, H., N. Haouchine, I. Peterlik, J. Dequidt, C. Duriez, H. Delingette, S. Cotin. Surgery Training, Planning and Guidance Using the SOFA Framework. – Eurographics, HAL Id: hal-01160297, Zurich, Switzerland, May 2015, pp. 1-5. **https://hal.inria.fr/hal-01160297/document**

39. Boubaker, M., M. Haboussi, J.-F. Ganghoffer, P. Aletti. Finite Element Simulation of Interactions between Pelvic Organs: Predictive Model of the Prostate Motion in the Context of Radiotherapy. – Journal of Biomechanics, Vol. **42**, Elsevier Ltd, 2009, pp. 1862-1868. ISSN: 0021-9290, DOI: 10.1016/j.jbiomech.2009.05.022.

40. Fouard, C., A. Deram, Y. Keraval, E. Promayon. CamiTK: A Modular Framework Integrating Visualization – Image Processing and Biomechanical Modeling, Studies in Mechanobiology. – Tissue Engineering and Biomaterials, Vol. **11**, Springer, Berlin, Heidelberg, 2012, pp. 283-321. Online ISBN: 978-3-642-29014-5, DOI: 10.1007/8415_2012_118.
https://link.springer.com/chapter/10.1007/8415_2012_125
41. Khor, W. S., B. Baker, K. Amin, A. Chan, K. Patel, J. Wong. Augmented and Virtual Reality in Surgery – the Digital Surgical Environment: Applications, Limitations and Legal Pitfalls. – J. Annals of Translational Medicine, Vol. **4**, December 2016, No 23, pp. 1-10. DOI: 10.21037/atm.2016.12.23.
https://pubmed.ncbi.nlm.nih.gov/28090510/
42. Makhataeva, Z., H. A. Varol. Augmented Reality for Robotics: A Review. – J. Robotics, Vol. **9**, 2020, Issue 2, pp. 1-28.
https://doi.org/10.3390/robotics9020021
43. Bauer, M., B. Bruegge, G. Klinker, A. MacWilliams, T. Reicher, S. Riß, C. Sandor, M. Wagner. Design of a Component-Based Augmented Reality Framework. IEEE Xplore, IEEE, INSPEC Accession Number: 7121584, 2001, pp. 45-54. Print ISBN: 0-7695-1375-1, DOI: 10.1109/ISAR.2001.970514.
44. **https://nxrev.com/2020/05/vuforia-vs-arcore/**
45. ARCore. (Retreived on 02.11.2020).
https://developers.google.com/ar

Received: 03.11.2020; Accepted: 27.01.2021