Abstract - The paper describes a medical visualization system suitable for both diagnosis and training in medical practice. The work is using open source frameworks and libraries such as WebRTC (Web Real-Time Communication), JSARToolKit (JavaScript Augmented Reality Toolkit), Three.js and NanoDICOM. The system builds a 3D model from the slices of a patient's DICOM imagistic investigation files, superimposes the model on the body area of interest using Augmented Reality and sends the camera images remotely. The main advantage consists in real-time viewing of imagistic patient information by assisting physicians or medical students. Virtually attaching the 3D model to a specific area of the body allows the model to rotate and move in the same time with the patient. The proposed application does not depend on any operating system and runs inside a browser window without any closed source plug-in, allowing it to be accessed on any computer or mobile device that has a suitable browser.

I. INTRODUCTION

The main goal of the project is to enable physicians to share DICOM (Digital Imaging and Communications in Medicine) files retrieved using MRI (Magnetic Resonance Imaging), CT (Computed Tomography) scans or other medical imaging techniques and combine the anatomical data with the image of the real patients.

Doctors will be able to interact remotely with other colleagues or their students in real-time during the patient diagnosis, increasing the accuracy of their results and facilitating the understanding of more complex problems that may arise in the process.

Similar systems are described in references [1], [2] and [3] but none of them are entirely web-based. Our application allows the system to run on any computer eliminating the need to have a specific proprietary operating system pre-installed or any other browser plug-in. This will solve the issues such as dedicated languages, plug-in requirements for interpretation, portability across browsers, devices and operating systems, and advanced rendering support as described in [4].

The open source libraries and frameworks selected as solution allow further development of the system and easy addition of other input and output devices such as optical head-mounted, volumetric or holographic displays for visualization. Adding gesture tracking devices for the easier interaction with the displayed 3D models is easily done only increasing the response time as in [5].

In October 2010, Theo Armour described in his article named “The Brain of Richard App” [6] a solution that uses Three.js in order to display the converted slices of a MRI scan and recombine those in a 3D model viewable inside a web browser. The actual conversion from DICOM to PNG is done in a desktop environment using Irfanview and XnView. The app doesn’t use Augmented Reality to combine the resulting 3D model with the actual image of the patient, neither offers a way to view the data associated with the DICOM files.

NanoDICOM has been used by Thomas Rager in July 2011 to build the WEB-DICOMVIEWER [7], a proof of concept that DICOM files can be parsed with PHP. The application does not allow us to upload files, instead of that it uses its own files which can be open from the user interface.

Many other free desktop applications allow us to view DICOM files. The most successful are 3D Slicer and ITK-SNAP.

Augmented reality is modern technology involving digitally enhanced view of the real world. Using cameras and sensors it adds layers of digital information expressed as video, photo, or sound material on top of objects from the immediate real world.
Augmented Reality applications already made their way in the medical world. One of the most promising applications is written by Mark Shippen and it is called DoctorMole. The app is available as a free for Android and it can detect suspicious moles using the standard ABCDE approach in order to determine risk for malignant lesions as described in [8].

Augmented Reality was also successfully used in “recovery and monitoring of surgical workflow, integrating preoperative and intraoperative anatomic and functional data, improving visual perception in a mixed environment, and developing new user interaction paradigms for taking full advantage of the virtual data, while overlaid onto the real scene” as in [9].

The European Space Agency presented in 2012 a new Augmented Reality unit called CAMDASS (Computer Assisted Medical Diagnosis and Surgery System). CAMDASS is a wearable augmented reality prototype that “can provide just-in-time medical expertise to astronauts. All they need to do is put on a head-mounted display for 3D guidance in diagnosing problems or even performing surgery” as described in [10]. The article also states that the prototype was developed for the Agency by a consortium led by Space Applications Services NV in Belgium with support from the Technical University of Munich and the DKFZ German Cancer Research Centre and they intend to test it first in remote locations such as the Concordia Antarctic base and after that in space.

II. FRAMEWORKS AND LIBRARIES

A. WebRTC (Web Real-Time Communication)

WebRTC is an open source framework that allows Real-Time Communication between the browsers (or peers) without the use of any other plugins. The framework is drafted on the API level by the W3C (World Wide Web Consortium) and on the protocol level by the IETF (Internet Engineering Task Force).

Currently WebRTC is implemented by Google, Mozilla and Opera in their browsers and it consists of three APIs accessible using JavaScript:
- **MediaStream** (also known as **getUserMedia**), which allows the browser to get access to the camera and microphone in order to capture media streams.
- **RTCPeerConnection**, allowing two different users to communicate streaming data (audio or video) directly from peer to peer.
- **RTCDataChannel**, facilitating the exchange of any kind of data (not only audio or video) between the connected peers.

WebRTC uses a mechanism to send control messages and to coordinate the communication between the peers. This process known as signaling (is not a part of **RTCPeerConnection API**) and it is used to exchange the following information between peers:
- **Session control messages** in order to start or close communication and to report eventual errors.
- **Network configuration** to communicate the computers IP addresses and ports used for communication.

- **Media capabilities** with the purpose to establish what codecs and resolutions can be handled by the caller’s browser and the browser it wants to communicate with, as explained by Sam Dutton from Google in his article [11].

B. JSARToolKit (JavaScript Augmented Reality Toolkit)

JSARToolKit is a JavaScript library written for Augmented Reality applications. It was indirectly derived from C ARToolKit, a video tracking library written in C, calculating the distance from the camera to a physical marker in real time, allowing us to draw a 3D model on top of it.

JSARToolKit uses the HTML5 `<canvas>` element for drawing the detected markers. In order to have images on this canvas we need to draw each frame of the video stream on it with the use of `drawImage()` function and loop trough the frames of the video with `requestAnimationFrame()`.

C. Three.js

Three.js is a JavaScript API that allows the rendering and the animation of 3D models inside a browser window. It allows developers to easily build a scene with different camera types, lights, textures, materials and shaders, rendered with various renderers such as Canvas, SVG or WebGL (accessing the functionalities of the user’s graphic card).

D. NanoDICOM

NanoDICOM is an open source server-side DICOM parser written in PHP that allows the reading of attached data and visualization of the images inside a web browser. It has been successfully used to build online DICOM viewers and its role in the current project is to retrieve the raw data from the images and based on the application we contribute on will transform those images into PNG format, needed for Three.js to reconstruct the 3D model based on the uploaded slices. It will also provide the base for an integrated online DICOM viewer used by physicians to see details in different slices or to colorize different areas of the slices before the 3D reconstruction in order to emphasize areas of interest as in [12].

III. THE SYSTEM UNDER DEVELOPMENT

The proposed system will use the patients DICOM images retrieved following an actual CT or MRI scan obtained as slices.

The images will be uploaded to the server where the slices will be transformed into PNG images in order to make them editable by the online image editor. This conversion is necessary because, for now, Three.js can handle only PNG, JPG and GIF from the known raster image formats.

Raster graphics are the most common type of image files. The files are formed by a set of pixels where each pixel represents an individual color within the image. Web graphics and digital photos are stored as raster graphics, either compressed or not (bmp, png, jpg, tif, gif).
The JSARToolKit library needs raster objects in order to detect and identify the individual markers that appear inside the images retrieved from the web camera.

The marker type that JSARToolKit uses is also called ID marker because each marker has a specific number (ID) associated with it. This association allows us to have multiple markers tracked in the same time in the real-time video feed and gives us the possibility to attach different 3D models to each one of them.

The markers are very similar to QR codes. The major difference is that we don’t need to track QR codes across multiple frames. We interpret their pattern once in order to obtain usually a string of text.

The markers are also used as a point of reference in order to establish the 3D model’s position and orientation in the visualized scene. This allows us to rotate or scale the 3D model associated with a specific marker attached to a real object when it changes its position.

The ID markers usually have a black, square contour that helps us in identifying the marker inside the tracked scene. The marker we use is presented in Fig. 2 and a specific pattern inside this square that may vary.

After editing the images with a built-in online JavaScript and PHP image editor, the set will be sent back to the application that runs in the browser and reconstructed in a 3D model using Three.js.

In order to accomplish this task we chose a solution that uses the THREE.PlaneGeometry() function. This allows us to apply each PNG image as a 2D texture to a PlaneGeometry object and reconstruct a 3D model of the patient’s organ from those planes as we describe synthetically in Fig.3, with an example we currently work on.

Since we can manipulate the opacity of the slices in Three.js the result will be a transparent 3D model, superimposed on the real image of the patient’s organ in the camera feed. We could also highlight each slice of the reconstructed model by altering its opacity or displaying that slice separately in another part of the user interface.

The reconstructed 3D model will be virtually attached to a detected marker with the aid of JSARToolKit that would use the feed of a web camera provided by the MediaStream API of WebRTC.

A real marker will be attached to the patient’s skin with the aid of a double-sided tape. This will allow the model to move in the same time with the image of the patient or to rotate when the subject changes his position as we describe in Fig. 4.

Since the size of the displayed image depends on the size of the viewed marker the application will provide a control mechanism to adjust the dimensions of the augmented 3D model or to bring closer the virtual camera used in Three.js in order to see the details of the mapped slices.

The mixed image can be afterwards transmitted to a remote browser with WebRTC where another physician could give a second opinion regarding the subject’s condition, making difficult cases accessible to remote specialists and increase the success rate of the future interventions where applicable. The easy manner in which a professional may access the application recommends it for medical staff.

It will be possible also to write annotations related to each slice for future reference and display them next to the resulting video image.
The system will be also suitable to teach students remotely by showing them real patients conditions and problems. This would provide a valuable experience and a highly effective medical training mechanism. The steps to be followed in the development of the system are presented in Fig. 5.

![Diagram](image)

**Figure 5. The steps to obtain and stream the combined images**

### IV. MEDICAL BENEFITS

The system we work on will be used as clinical supporting tool in rehabilitation medicine. The proposed system is intended firstly for medical students to train and have a better relation between reality and symptoms. The image obtained from slices will be modeled 3D and superimposed on the real knee as soon as possible after the imagistic process. The solution is easy to use and economical from the costs point of view. Now-a-days students are keen in using modern visual tools so making available a user friendly instrument to train is coming to cover a real need.

After future developments the system will be ready to be a useful tool for a second opinion in diagnosis and further work may drive to a tool for better recovery including feedback from patients. The system we work on will be used as clinical supporting tool in rehabilitation medicine.

In nowadays medicine rehabilitation engineering and assistive technology depend on understanding the confluence of two factors describing an individual’s functionality. This confluence is of medical restoration and patient’s participation. Whether we are referring to stroke, spinal cord injuries, upper and lower extremities soft-tissue injuries, trauma disorders, osteoarthritis, inflammatory arthritis, or cognitive impairment, rehabilitation needs to address the International Classification of Functioning, Disability and Health [13].

MRI, CT, and ultrasound imaging have an expanding role in the diagnosis and follow-up of pathology of the musculoskeletal, neuromuscular, and central nervous systems. The rehabilitation specialist should keep abreast of the current and future applications of these imaging modalities to ensure that the patient involved in the recovery process will maximally benefit from the variety of new imaging technologies that are becoming available [14].

The proposed system aims to identify the patient’s affected limb or joint disorders based on imaging scanning. The addressed disorders include bones, muscular structures of upper and lower limb, and intra-articular lesions of knee, ankle, elbow or wrist joints (injuries of the ligaments or tendons, meniscal tears).

The imaging scanning is correlated to real-time patient’s functioning of the affected structure. This is important as the rehabilitation specialist can detect the sites of lesions and can make an accurate assessment of patient’s capacity to perform certain tasks related to affected structures.

The system is non-invasive and easily manageable by the doctor. It offers a great amount of information about a patient’s medical status and about neuromotor capacities of the involved limb, and thus of the overall functionality.

### V. CONCLUSIONS AND FUTURE DEVELOPMENT

The system brings into medical practice a novel approach since the application will be able to run inside a browser window, without the need to install any plug-in or to have limitations imposed by the operating system. These aspects are very important if we take into account that the targeted user groups (doctors, medical students, medical staff) do not need to have prior training in computer science or engineering.

Although the system will be capable to run on classic computer architecture, it will be easy to include other input and output devices such as optical head-mounted displays, motion tracking devices or holographic displays with the purpose of better image visualization and manipulation.

The system is under development in cooperation with the medical staff from Rehabilitation Department at “Victor Babes” University of Medicine and Pharmacy Timisoara, Romania.

### REFERENCES


