



# On the Use of Virtual Reality for Medical Imaging Visualization

Filipi Pires<sup>1</sup> · Carlos Costa<sup>1</sup> · Paulo Dias<sup>1</sup>

Received: 17 September 2020 / Revised: 3 June 2021 / Accepted: 21 June 2021 / Published online: 29 July 2021  
© Society for Imaging Informatics in Medicine 2021

## Abstract

Advanced visualization of medical imaging has been a motive for research due to its value for disease analysis, surgical planning, and academical training. More recently, attention has been turning toward mixed reality as a means to deliver more interactive and realistic medical experiences. However, there are still many limitations to the use of virtual reality for specific scenarios. Our intent is to study the current usage of this technology and assess the potential of related development tools for clinical contexts. This paper focuses on virtual reality as an alternative to today's majority of slice-based medical analysis workstations, bringing more immersive three-dimensional experiences that could help in cross-slice analysis. We determine the key features a virtual reality software should support and present today's software tools and frameworks for researchers that intend to work on immersive medical imaging visualization. Such solutions are assessed to understand their ability to address existing challenges of the field. It was understood that most development frameworks rely on well-established tool-kits specialized for healthcare and standard data formats such as DICOM. Also, game engines prove to be adequate means of combining software modules for improved results. Virtual reality seems to remain a promising technology for medical analysis but has not yet achieved its true potential. Our results suggest that prerequisites such as real-time performance and minimum latency pose the greatest limitations for clinical adoption and need to be addressed. There is also a need for further research comparing mixed realities and currently used technologies.

**Keywords** Virtual Reality · Medical Imaging Visualization · Volume Rendering · DICOM

## Introduction

The past decade has witnessed the second wave of widespread consumption of virtual reality (VR). Technology breakthroughs and the proliferation of companies developing VR-related products have brought low-persistence lag- and smear-free displays, wireless headsets emerging at a fast pace, and the future seems to be headed toward head-mounted displays (HMDs) that do not require external tracking sensors, allowing completely untethered immersive experiences [1, 2].

According to Perkins Coie's survey report on augmented reality (AR) and VR [3], Healthcare is the sector with the second highest expectation of most investment in VR technology during 2020, right behind Gaming. This supports the idea that there are highly valuable benefits to the use of VR

for medical scenarios. And this is not unfathomable since the use of VR technology in this industry facilitates training and mirroring real-life experiences, particularly interesting for life-critical environments. Experts in the field believe research has only begun to perceive the applicability of virtual environments. However, one of the major setbacks to innovation is the processing, visualization and manipulation of large volumes of clinical data in real time. Volume rendering, a set of techniques used to display two-dimensional (2D) projections of three-dimensional (3D) discretely sampled data sets, is generally the method of choice for data visualization. The data sets are typically groups of 2D slices acquired by Computed Tomography (CT) or Magnetic Resonance Imaging (MRI) scans and are rendered by computing polygon meshes, with the help of algorithms such as the Marching Cubes [4].

Today's volume rendering technologies can facilitate early diagnosis and treatment planning for physicians by providing more realistic representations of patients' physical conditions than traditional 2D images. However, there are several limitations, hence the disproportion between existing

✉ Filipi Pires  
filipesnetopires@ua.pt

<sup>1</sup> Department of Electronics Telecommunications and Informatics, University of Aveiro, Aveiro, Portugal

solutions for education or training and those for surgical planning and other scenarios where requirements for image quality and data control are extremely high. Users are still confined to 2D displays of the 3D volumetric data, reducing their ability to make calculations (such as determining distances between objects) and allowing interaction only through standard input devices (mouse and keyboard).

In this paper, the current state of VR in medical imaging visualizations is explored in search of understanding the availability and quality of current solutions. Similar work has been published for Mixed Reality in general [5]; here, the goal is to present the state-of-the-art of immersive medical software products as well as of development frameworks for VR-based medical imaging visualization. This survey is stated in [State-of-the-Art](#) and [Development Tools and Frameworks](#). Development tools are evaluated to understand their ability to tackle current limitations and potential to improve clinical workflows in a field that still relies heavily on 2D image analysis and that could gain from a shift to 3D virtual immersive analysis. Conducted software experimentations are described in [Development Tools and Frameworks](#). Conclusions and final thoughts on future developments are presented in [Conclusions and Future Work](#).

## State-of-the-Art

Mixed Reality technology is already being used within the medical industry for multiple purposes. In education, it is used for teaching anatomy, as D. Aouam et al. [6] and the creators of the application 4DAnatomy [7] present (one using VR, the other AR) in their interactive systems with the use of pre-modeled human organs; or in low-cost practical exercises with virtual human dummies as suggested by X. Wang et al. [8]. The effectiveness of such tools is thoroughly assessed by M. Cabrera et al. [9] in their study with a focus on VR as well. Immersive solutions are also being used for training medical students, like in PathoGeniusVR [10] where a serious-game approach is adopted, or resorting to surgical simulations with pre-processed models [11] or real body scans [12, 13]. In clinical practice, it is used for personnel training, as the authors of SimCEC [14] explore comparing their solution to similar ones, or as A. Javaux et al. [15] present in their VR trainer with motion and force-sensing capabilities. Surgical planning is one of the application areas with the most significant impact on public health and also the most demanding in terms of image quality, faithfulness to reality, and interaction power. M. Holland et al. [16] adopt non-HMD hardware for their promising cardiovascular blood simulation for hospital decision support; other solutions adopt commercial hardware such as Microsoft's HoloLens [17]; but VR seems to be quite explored as well, in fact, it is the chosen technology when haptic feedback is

a requirement [18, 19]; more recently, VR is starting to be used for telerobotics or robot-assisted surgical systems [20] [21, 22]. There is no doubt that immersive healthcare simulators are much coveted, as the paper on the state of the art of these solutions explores [23]. Other scenarios also resort to VR such as psychological therapy [24–26] or rehabilitation [27–30] where the patient-as-user paradigm is adopted.

Non-educational virtual reality in particular is being increasingly used for two main scenarios: highly realistic simulators and highly interactive medical data visualization stations, both ultimately designed for surgical planning. In order for a system to be adopted in a hospital environment, it needs to support different patient data (and not pre-processed generic 3D human models), and support the data formats used by the industry-standard equipment. A medical image data set consists typically of one or more images representing the projection of an anatomical volume onto an image plane (projection or planar imaging), a series of images representing thin slices through a volume (tomographic or multislice two-dimensional imaging), a set of data from a volume (volume or three-dimensional imaging), or multiple acquisitions of the same tomographic or volume image over time to produce a dynamic series of acquisitions (four-dimensional imaging). Medical image file formats can be divided into two categories: the first is formats intended to standardize the images generated by diagnostic modalities, e.g., Dicom [31]; the second is formats created to strengthen postprocessing analysis, e.g., Analyze [32], Nifti [33], and Minc [34]. We will address the former category, considering the usage scenario to be explored.

Today, the Digital Imaging and Communication in Medicine (DICOM) standard is the backbone of every medical imaging department. The added value of its adoption in terms of access, exchange, and usability of diagnostic medical images is, in general, huge. The fact that one single file contains both the image data and the metadata (patient information and a complete description of the entire procedure used to generate the image) as a header, the support for many compression schemes (JPEG, JPEG-LS, JPEG-2000, MPEG2/MPEG4, RLE and Deflated) and the wide adoption by radiography, CT, MRI and ultrasound systems [35] has led to a great popularization of such format in the medical research community, with over a hundred thousand articles mentioning it worldwide [36]. It has recently seen progress in its encapsulation in STL (for 3D printing) and OBJ and other formats (for Virtual, Augmented and Mixed Realities) with the WG-17 3D extended mandate [37], acknowledging the necessity of a clean dataflow between current industry data collection hardware and innovative data manipulation solutions.

There is already software on the market that takes advantage of the full potential of DICOM. Table 1 presents a list of products that use VR to visualize and interact with medical data, along with the commercial hardware supported.

**Table 1** VR products on the market for medical imaging visualization with support for DICOM

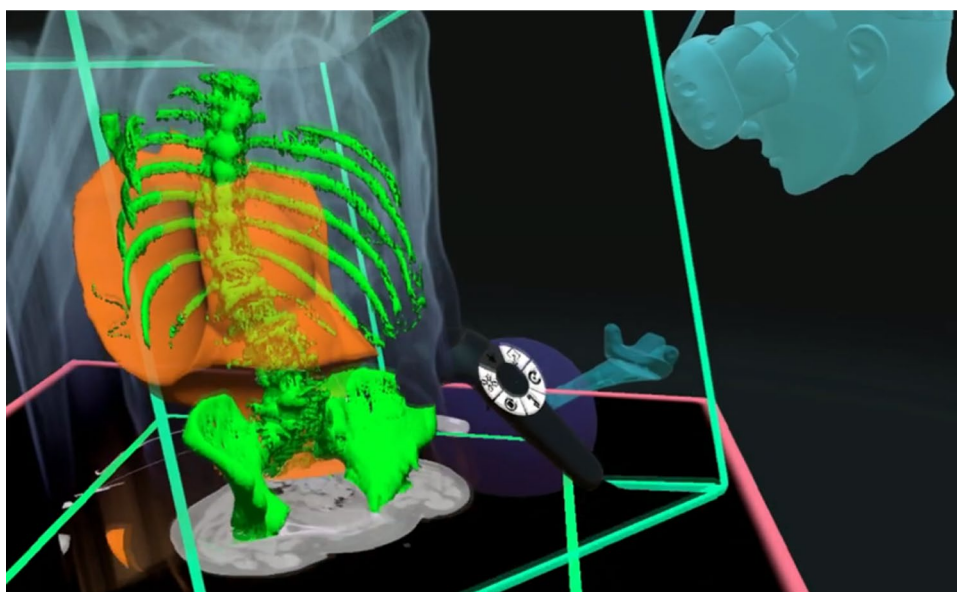
Product	Hardware	Purpose
Elucis	SteamVR HMDs	Data modelling / segmentation
Specto VR	SteamVR HMDs	Realistic data visualization
Immersive View	Oculus Rift	Data visualization
Imaging Reality	SteamVR HMDs	Data visualization
DICOM VR	HTC Vive	Data visualization
Precision VR	SteamVR HMDs	Real-time data visualization
HoloDICOM	HoloLens	Real-time data visualization / segmentation
True 3D	Own Hardware	Data visualization / segmentation

Realize Medical’s Elucis software [38] is a VR-based 3D medical modeling platform for creating patient-specific 3D models. It is used by importing volumes and handling 2D and 3D drawing tools to select and segment individual body parts that can then be exported in OBJ or STL. The algorithms of Elucis allow fast modeling of bones, organs, etc., based on a patient’s volume data and providing features such as mesh gap correction, surface smoothing, and color-mapping tools. By supporting the movement of both the volume and a cutting plane in 3D space, and allowing the usage of multiple planes with different purposes, it is possible for an experienced user to achieve highly precise results. Elucis works with any VR hardware supported by SteamVR. The VR interface offers solid control over the scene, although no means of moving the drawing area was found when exploring the tool, limiting its actual usage. The software’s learning curve should be considered when deciding whether to adopt the software or not, as its complexity is proportional to its power. Figure 1 presents an Elucis user performing a modeling task.

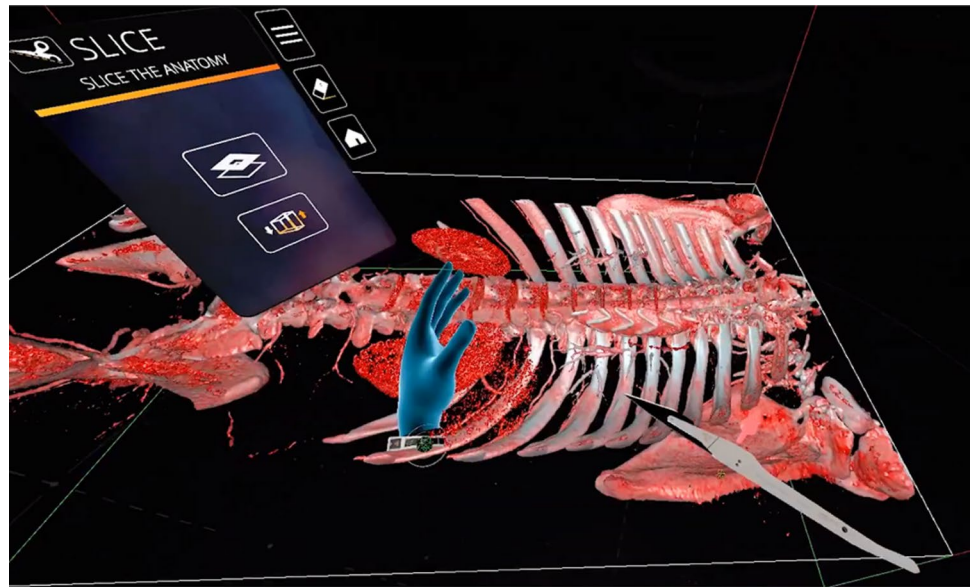
Diffuse’s Specto [39] is another VR product with a publicly available demonstration. SpectoVR, or SpectoVive, is

focused on advanced visualization of medical image datasets with ray tracing techniques for high-quality rendering. Most of the data manipulation is done through their 2D screen interface, while VR is used only for interactive visualization. Like all of the products described in this paper, the software supports DICOM files. Specto provides threshold-based segmentation allowing the user to identify organs and other elements and define different color and opacity values. When immersed, the image quality is highly realistic and the features provided involve basic volume size and position manipulation, lighting changes, and a cutting plane with the option to activate the MRI image placed on the plane. Due to its limited feature list, Specto is most suitable for education and possibly disease diagnosis, requiring powerful hardware for processing. It is currently being used for these purposes and for mummy exhibitions.

ImmersiveTouch is a pioneering company on the medical imaging visualization in immersive environments, with a scientific publication [40] presenting one of the first systems that “integrates a haptic device with a head and hand tracking system”. Today, it provides three very much related products: Immersive View VR (IVVR), Immersive View

**Fig. 1** Screen capture of Elucis demonstration, taken from [38]

**Fig. 2** Screen capture of ImmersiveView VR demonstration, taken from [41] Screen capture of ImmersiveView VR demonstration, taken from [41]



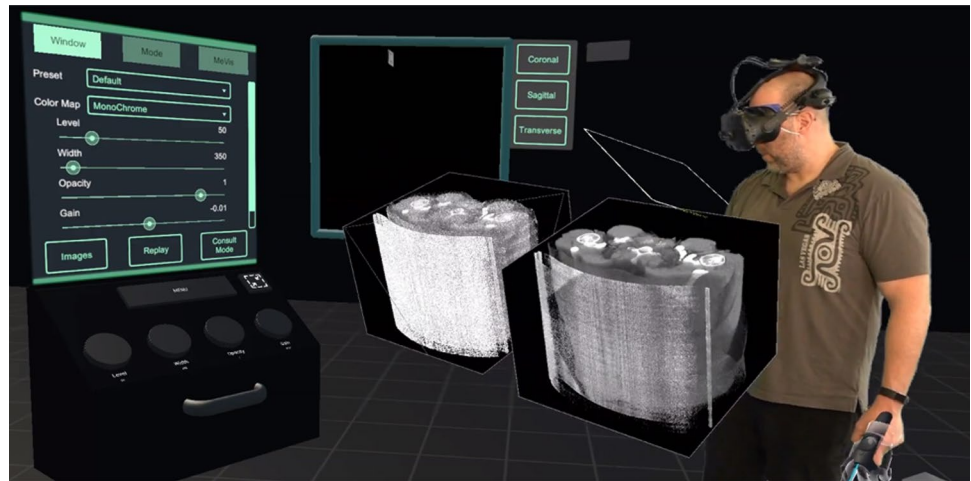
Surgical Planning (IVSP), and Immersive Sim Training (IST). IVVR [41] is the basis of ImmersiveTouch's solutions. Like Specto, it creates explorable 3D VR models from patients' own CT / MRI data, but supporting only Oculus Rift. Their measurement system and 3D drawing tools are something that most systems lack and provide information with millimetric precision valuable when planning for a surgery. The opacity feature is easily controlled and the dynamic cutting tool allows cutting from any angle. The solution also offers a snapshot recording feature. Figure 2 presents a screen capture of IVVR. Unlike most products, data is uploaded to ImmersiveTouch's online platform to be prepared for viewing, making the solution very light. IVSP uses IVVR to generate high-fidelity VR replicas from patient data and allows surgeons to study, assess and plan surgeries and collaborate intra-operatively with their team. IST (and AR Immersive Touch 3) do not use real data and are meant for educational purposes, supporting more HMD hardware and robotic stylus haptic devices.

Imaging Reality [42] creates a virtual workstation with dedicated areas for the medical data volumes and for the controls. It is meant for generic scenarios of medical data visualization and supports most of the commercial HMDs available. The cutting plane lets the user choose whether to see the cut volume with or without the slice overlay and it allows the viewing of the 2D slices in a specific area of the virtual workstation. While going for a fixed virtual menu keeps the virtual space organized, it makes it harder to customize. This solution has an uncommon and optional feature of linking duplicated volumes where each basic manipulation (rotation, scale, etc.) in one volume is applied to the other one as well, but each can have different filtering options (see Fig. 3), allowing a more insightful analysis.

DICOM VR [43] is a more case-specific VR solution for medical data visualization, focused on planning targeted radiation treatment. The 3D representation of the volumes is easy to manipulate using HTC Vive. The cutting tool can be a plane or basic volumes like a square or a sphere, which might be useful in some scenarios; the cut parts present the corresponding slice images only. Threshold presets allow the filtering of data and visualizing only desired parts such as bones, soft tissue or even specific organs like the lungs. The contour feature can be used for segmentation and marking of relevant parts. It can be stored for later editing, and it is color coded with the option of adding labels. However, it is slow and purely manual work, unlike with Elucis where portion delimitations were automatically detected. It has a drilling feature as well, where the user is able to dig wholes in the volume. Moreover, it contains features related to target radiation treatments with LINAC hardware. Nevertheless, the most significant issue might be the dependency of assuming that the DICOM data has the appropriate ranges of values in order for their presets to work. This issue can also be applicable to other products.

PrecisionVR [44] focuses on brain scans and is divided in four products: surgical planner (SRP) responsible for processing scans to create patient-specific VR reconstructions; precision VR viewer allowing visualization for the doctor with the HMD and for the patient in a screen; surgical navigation advanced platform (SNAP) that enhances the surgeon's current operating room workflow by integrating with and enhancing the existing surgical navigation system along with other tools and technologies while using the capabilities of Precision VR; and the VR studio, a collaborative networked, multi-user, multi-disciplinary environment powered by Precision VR. Surgical Theater's software differs from

**Fig. 3** Frame of Imaging Reality's video presentation, taken from [42]



the other products in the sense that it is used in real-time during surgery. The visualized data corresponds to a representation of the patient's status allowing others besides the surgeon to follow the procedure. Every person in the room is able to see the patient's data through the 2D screens. However, not many details regarding the usability of the solution were found.

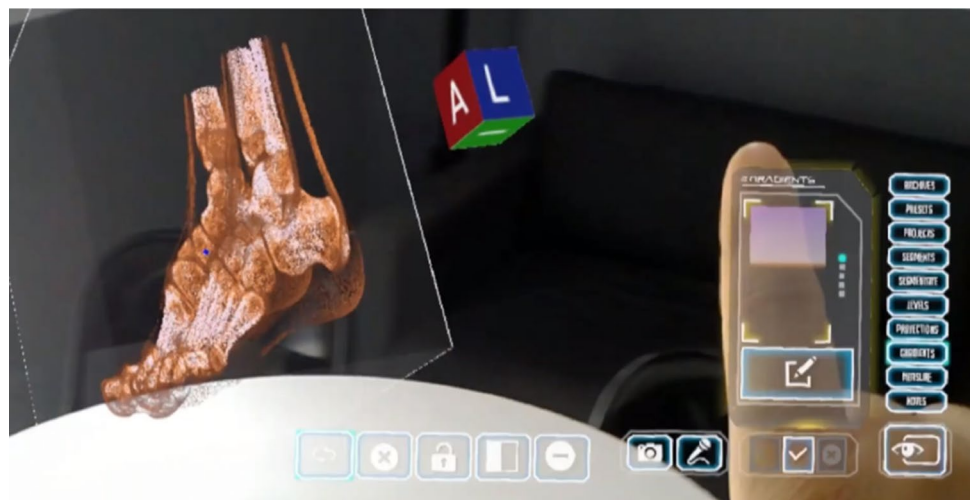
HoloDICOM [45] is another medical imaging visualization software meant to be used in real time. As a standard visualization tool, this product is limited in its root due to the Mixed Reality technology adopted - augmented reality using Microsoft's HoloLens - suffering from the lighting conditions of the real world, visual clutter and other known issues. However, their value lies in allowing a surgeon to actually be able to use the gear during procedures as it remains aware of the real world (as seen in Fig. 4).

The solution is complete in terms of features, doing justice to the term reality augmentation. Users resort to tools for measurements, volume manipulation, screen capture, auxiliary controllable 2D slices and a seemingly intuitive virtual menu. Their segmentation feature supports several

algorithms including the practical management of the threshold and penumbra (thresholding), the different levels of the grayscale ("Window Level"), the Watershed algorithm [46] and the application of Hounsfield units [47].

EchoPixel's True 3D software [48] is one of the most accurately representative products on the market of medical data visualization in the immersive environments of today. Being fully aware of the potential inconvenience of using regular VR gear in medical environments and of the natural reluctance of professionals to adopt such different form of workflow as is the ones proposed by previously mentioned products, the creators of True 3D decided to develop their own dedicated hardware specifically for the medical scenario - their device can be seen in Fig. 5. Special glasses are required to see the 3D volume; multiple users with glasses are supported, while others see the same volume on the 2D screen. True 3D has all the features desirable in a medical visualization tool like cutting planes, segmentation with color codification and labeling, tissue extraction, etc. but offers a 3D pen interaction method with great usability results. The software is currently being used to assist

**Fig. 4** First-person view of HoloDICOM during visualization, taken from [45]



**Fig. 5** Representative image of a user interacting with True3D, taken from [49] Representative image of a user interacting with True3D, taken from [49]



physicians in planning surgical and interventional procedures to treat congenital heart defects [49, 50].

There are alternative products with web-based approaches, such as C. Huang's et al. WebVR-based framework [51] or F. Adochiei's web platform [52], but the great majority are desktop and Steam applications. From the research community it is also possible to find some works, e.g., NextMed [53] (supporting both VR and AR), VRvisu [54] (with the use of smart bracelets), NIVR [55] (focused on neuroimaging visualization), and others.

Table 2 is provided for a global perspective on the main capabilities the products here described provide to their users. It suggests that software meant for medical imaging visualization has some essential features:

- Volume Manipulation - standard control of the patient's volume data, including moving, scaling, and rotating.
- Opacity Adjustment - definition of the opacity value for the whole volume or parts of it (through segmentation); lighting adjustment can also be useful.
- Cutting Plane - freely manipulable volume cutting plane for interior analysis; preferably with options such as allowing the visualization of the correspondent MRI and supporting multiple planes.
- Automatic or Semi-Automatic Segmentation - select and isolate body parts such as bones or organs through parameter tweaking of algorithms such as the window level for images' grayscale, the common threshold and penumbra manipulation or personalized transfer functions.

These features and other such as 3D measurement, drawing, and labeling tools, or more complex and very powerful

segmentation algorithms available provide to physicians a level of control never achieved before. Field-tested products like Elucis, ImmersiveView or True3D, and others not here presented, undoubtedly prove their value for the future of medicine.

## Development Tools and Frameworks

When it comes to implementing software solutions on information visualization, the tools used by developers tend to be generic frameworks capable of providing immense visual methods and interaction commands and of creating user experiences for a wide range of scenarios. However, when it comes to medical image data, the data file formats are so specific and based on standards unique to the industry of healthcare that the manipulation technologies must be built around such constraints. So an individual branch from within the field of information visualization separates itself from the remaining to optimize the processing, manipulation and rendering of medical image data sets.

The Visualization Toolkit (VTK) [56] is one of the most powerful open-source software systems for 3D computer graphics, image processing and visualization with greatest support for medical data formats, namely the DICOM format, offering a wide variety of visualization algorithms including scalar, vector, tensor, texture, and volumetric methods; and advanced modeling techniques such as implicit modeling, polygon reduction, mesh smoothing, cutting, contouring, and Delaunay triangulation. The toolkit, initially presented in a book by W. E. Lorensen et al. [57], has seen improvements by users and developers around the world ever since its creation in 1993, and the same contributors keep applying the system to real-world problems. Kitware Inc.,

**Table 2** VR products on the market for medical imaging visualization with support for DICOM

	Elucis	Specto VR	Immersive View	Imaging Reality
Basic manipulation (scale, rotate, move)	✓	✓	✓	✓
Opacity adjustment	✓	✓	✓	✓
Cutting Plane	✓✓	✓✓	✓✓	✓✓
Threshold	✓	✓	✓	✓
Segmentation				
Additional forms of Segmentation	✓✓	X	X	X
Lighting manipulation	X	✓	X	X
Measurement tool	X	X	✓✓	X
3D drawing and Labelling	X	X	✓	X
Screen capture	X	X	✓	X
Volume duplication	X	X	X	✓
	DICOM VR	Precision VR	Holo DICOM	True 3D
Basic manipulation (scale, rotate, move)	✓	✓	✓	✓
Opacity adjustment	✓	-	✓	✓
Cutting Plane	✓	-	✓	✓
Threshold	✓	-	✓	✓
Segmentation				
Additional forms of Segmentation	✓	-	✓✓	✓✓
Lighting manipulation	X	-	X	X
Measurement tool	X	-	✓	X
3D drawing and Labelling	X	-	X	✓
Screen capture	X	✓	✓	X
Volume duplication	X	-	X	X

Checkmark means the software supports the feature, X it does not and hyphen means that it is unknown. Double checkmarks mean that the product's feature is somehow improved

the owner of VTK, actively contributes to and maintains several leading open-source software packages specialized in computer vision, medical imaging, visualization and 3D data publishing, including VTK, ITK [58], IGSTK [59], 3DSlicer [60], ParaView [61], and more. In fact, VTK is the basis of many visualization applications, including ParaView and 3DSlicer, and other well-known solutions such as InVesalius [62], FreeSurfer [63], OsiriX [64], QGIS [65], etc. while also being responsible for the graphical user interface of the Insight Segmentation and Registration Toolkit (ITK). In 2017, Kitware released a new version of the toolkit with support for VR [66], opening doors to easier development of

specialized immersive medical visualization tools, and soon such innovation quickly spread to their remaining software [67, 68].

However, most of the products we have previously discussed go beyond volume processing and provide graphical elements such as virtual and on-screen menus for the user interaction with the system to be more fluid and usable. In fact, part of the core logic of the systems involve aspects outside the traditional scope of toolkits such as VTK. This is where game engines such as Unity [69] or Unreal Engine [70] come in. Although originally meant for game development, these engines are today used for far more purposes than that, due to their convenience and speed of development based on abstractions very appealing to software designers. Such software development environments include rendering and physics engines, animation, memory management, threading, scripting and more. Also what makes them so appealing is the fact that game engines provide platform abstraction, allowing the same application to be run on various platforms with few, if any, changes made to the source code.

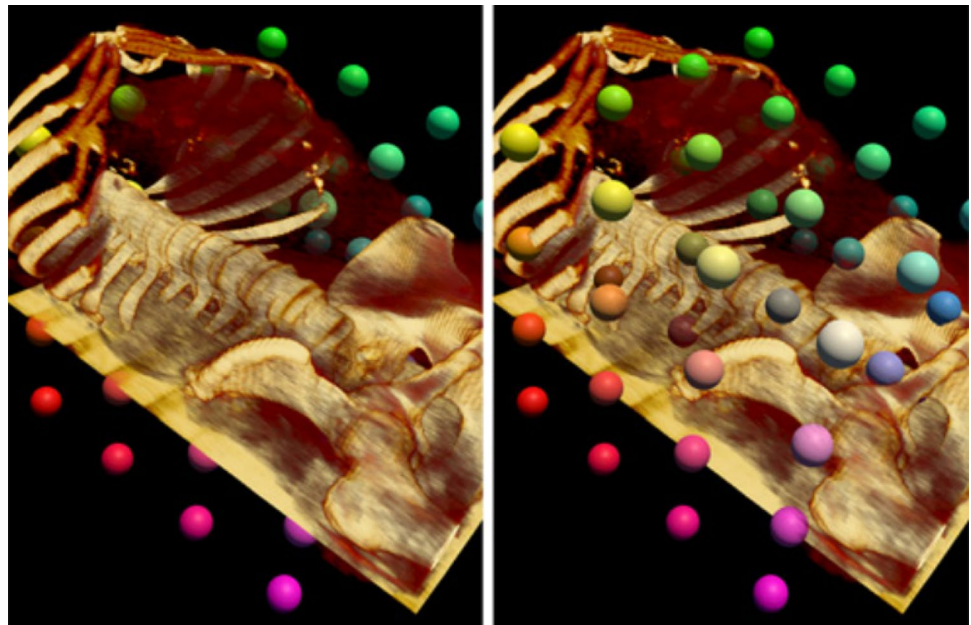
In the past five years, efforts have emerged to combine both of these realities in order to use the advantages of each. One good example of this is a plugin designed for Unity called VtkToUnity [71]. G. Wheeler et al. present a method to interconnect VTK and Unity, enabling them to exploit the visualization capabilities of VTK with Unity's widespread support of immersive hardware for the development of medical image applications for virtual environments. By using the OpenGL renderer and low-level communication interfaces, it was made possible to directly interact with VTK through Unity's C# scripts. The authors also modified some components of VTK so that volumes rendered by the toolkit could be integrated with other virtual elements rendered in Unity without one overlapping the other (see Fig. 6). More implementation details such as camera synchronization, cropping plane colliders, materials and overall evaluated performance are described in their paper.

VtkToUnity is open source<sup>1</sup> and the authors provide an example project for running the Unity project. However, no documentation was found beyond the quickstart instructions provided on the code repository. According to the 3D Heart project, owner of the plugin, the software supports only the Windows 10 Operating System (OS) and should accept any SteamVR compatible VR hardware, although it was only tested with HTC Vive.

MeVisLab [72] is an open-source cross-platform application framework for medical image processing and scientific visualization that has seen progress in the immersive realm.

<sup>1</sup> [https://gitlab.com/3dheart\\_public/vtktounity](https://gitlab.com/3dheart_public/vtktounity)

**Fig. 6** Results of the changes made to VTK by the creators of the Unity plugin, taken from [71]



The solution accepts DICOM and other formats, provides volume rendering capabilities, and their module library includes many VTK/ITK-related modules for advanced development. It is currently being used in a variety of clinical applications, including surgery planning solutions [73], and research projects<sup>2</sup>, with work on creating a plugin for visualization in VR [74]. However, MeVisLab has its own Integrated Development Environment (IDE) for graphical programming to create user interfaces and it cannot yet be integrated with a game engine.

Another interesting example of a software tool for the development of medical imaging visualization solutions is the Immersive Medical Hands-On Operation Teaching and Planning System (IMHOTEP) [75]. The framework, designed for surgical applications, aims to be a development and research platform for VR tools in clinical scenarios. It provides many standard functions that facilitate the use of a VR environment in conjunction with medical data. The core functionalities of the framework are open-source<sup>3</sup>, as well as some basic documentation on how to run the project and contribute as a developer. In the authors' paper published in 2018, Pfeiffer et al. [76] present the evaluation conducted by over seventy clinical personnel participants and their feasibility study where the framework was used during the planning phase of the surgical removal of a large central carcinoma from a patient's liver, both providing very positive results.

IMHOTEP uses Unity as the rendering back-end in order to achieve real-time performance. The adopted modular architecture ensures extendability through application programming interfaces (APIs) and multithreading for information exchange. It is possible to import and parse DICOM data via Kitware's ITK library [58] and render the 3D volume processed by a custom GPU shader program. The platform also supports previously segmented surface meshes (representing structures such as organs, vessels and tumors) generated using external tools. This is possible by importing the mesh into the Blender3D software<sup>4</sup> and then save it in Blender's native *.blend* format, which can be then parsed by the framework and presented in virtual space. Additional data (such as laboratory results and patient history) can be loaded and displayed alongside the image data. Optionally, text can be formatted in HTML for further customization. All features accomplished by integrating the various framework modules.

The Visualization Toolkit, the VTK plugin for Unity and the IMHOTEP framework all have default tools for DICOM volume manipulation and provide functions for the usage and/or implementation of features such as the opacity adjustment, the cutting plane and volume segmentation (even if based only on value thresholds). A superficial analysis suggests that these generic solutions have the potential to be used for the development of immersive medical imaging visualization stations specialized in a given relevant context.

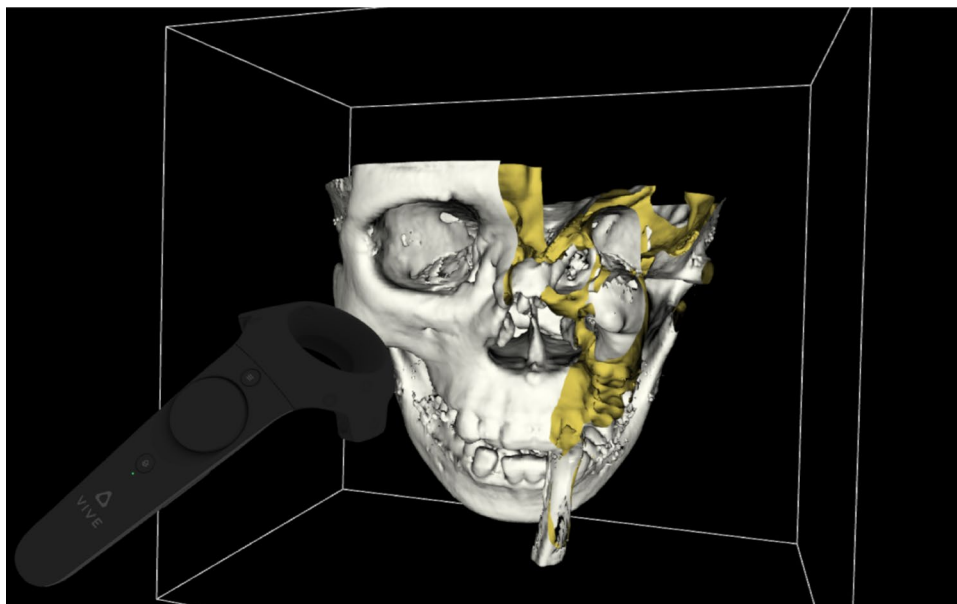
<sup>2</sup> MeVisLab publications: <https://www.mevislab.de/mevislab/publications>

<sup>3</sup> <https://github.com/IMHOTEP-Medical/imhotep>

<sup>4</sup> An open-source 3D creation suite software, supporting modeling, rigging, animation, rendering and more, that can import many common file formats.



**Fig. 7** Screen capture of the VTK prototype, displaying a patient's skull with a cutting plane



## Experimentation

In all of the development tools and frameworks explored, publicly available datasets were used, either provided with the tools themselves or from online medical databases with anonymized patient samples<sup>5, 6, 7, 8</sup>.

As most solutions rely on the Visualization Toolkit and it now supports the usage of VR hardware, this was the first tool to be used. Its installation, however, was not trivial. The toolkit needs to be compiled on the machine where it will run and several dependencies must be installed in order to achieve this. Additionally, the OpenVR module (used for interaction with the VR hardware) must be compiled simultaneously and cannot be added later. This module caused several errors during compilation and installation, proving to be unstable on the latest releases of VTK.

Once successfully installed, using VTK v7.1.0 and OpenVR v1.0.3, all development was purely through programming. Sequences of DICOM images can be imported and rendered as 3D volumes, with basic manipulation out-of-the-box with HTC Vive's controllers. By selecting specific contour value ranges, segmentation of bones and soft tissue was possible, although creating many artifacts and varying much in quality among different data sets. The selection of each predefined range was made through the use of UI buttons attached to the user's screen. All UI elements were based on already existing VTK widgets, but

their integration with VR always required some tweaking for proper usage. Opacity adjustment, on the other hand, quickly showed positive results. For the cutting plane, attempts to adapt some implementations applied to 3D meshes were unsuccessful or poor in usability; also, the standard means of rendering forced the software to re-processing the cut volume each frame, resulting in a slow frame rate and a need for optimization. Fig. 7 presents a screen capture of the work developed using VTK.

The overall perspective on the usage of VTK is that the toolkit is, although very powerful in terms of image processing, inadequate for the development of standalone applications. The limited control over interaction methods and the primitive tools for the development of user interfaces made working with VTK slow and inefficient.

Attempts on resorting to MeVisLab were conducted to understand if it could be a reliable tool to interact with medical data in a virtual environment. However, the steep learning curve associated made development difficult and slow. The authors developed a GUI definition language (MDL) that allows the integration of Python scripts, but there is a demand for developers to have previous knowledge about MeVisLab before creating VR interfaces. Moreover, their network-based pipelines provide means to build macro modules out of existing ones that, in conjunction with scripts and GUIs, make up new applications.

It was clear that currently the support for VR is limited to visualization, meaning that it is still not possible to easily interact with 3D data in virtual space. It is worth mentioning that, although very promising in terms of control over medical data, MeVisLab is mostly meant for image processing and not actual interaction. For example, the software has dedicated features for Machine Learning (ML) only. The authors of the VR plugin proved the feasibility of integration

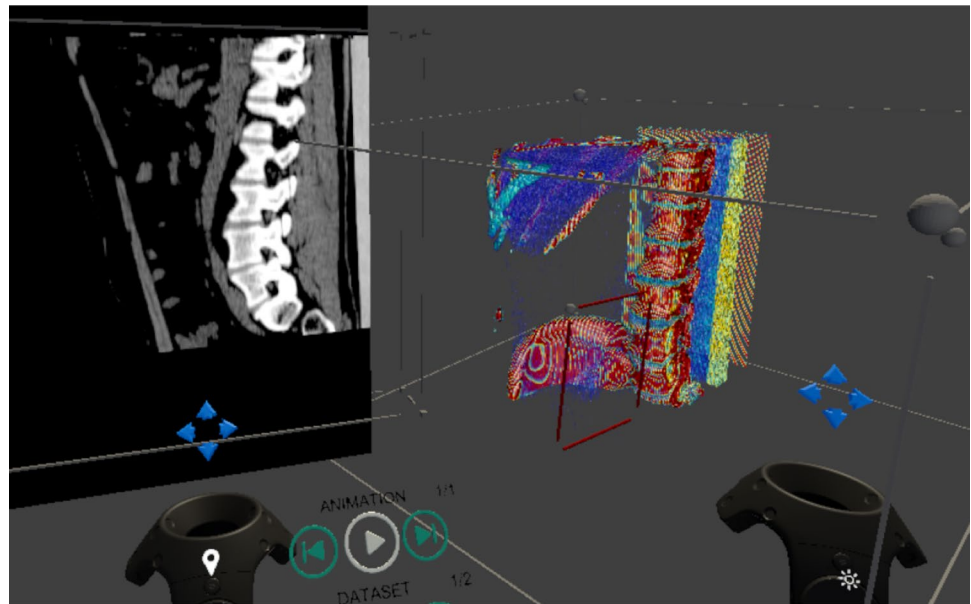
<sup>5</sup> <https://www.dicomlibrary.com/>

<sup>6</sup> <https://www.dclunie.com>

<sup>7</sup> <https://oasis-brains.org/>

<sup>8</sup> <https://www.sci.utah.edu/cibc-software/cibc-datasets.html>

**Fig. 8** Screen capture of the VtkToUnity prototype, displaying a torax with the implemented cutting plane and a transfer function applied to the volume



of VR into MeVisLab and they have even proposed the integration with game engines<sup>9</sup>, but this reality is still taking the first steps. For this reason, MeVisLab was found limited for the development of immersive imaging visualization solutions.

It was of our knowledge that the usage of game engines would provide an abstraction layer with many benefits. This would not only reduce the amount of code needed but would also make the interaction with the hardware more transparent to the developer. Also, development tools based on such engines would most likely deal with optimization processes for rendering, allowing full focus on the actual development of immersive medical applications.

This was the case with the VtkToUnity plugin. G. Wheeler et al. first presented the plugin in 2018. Its development has significantly evolved since then, resulting in major changes during our experimentations. The initial installation of the plugin was a complex process involving the compilation of VTK (with all difficulties mentioned above), with some modifications and the application of a patch created by the authors, but later on they were able to publish a version containing all dependencies and direct integration with Unity.

Although no documentation is provided beyond an installation tutorial, the authors provide Unity scenes containing usage examples. The Unity Steam VR plugin significantly increases flexibility and simplifies operations with HTC Vive's controllers, so the authors provide in those scenes prefabs (reusable assets with some level of complexity) with interaction scripts to, for example, pick up and place objects

using the trigger, or control the transfer function windowing using the touchpad.

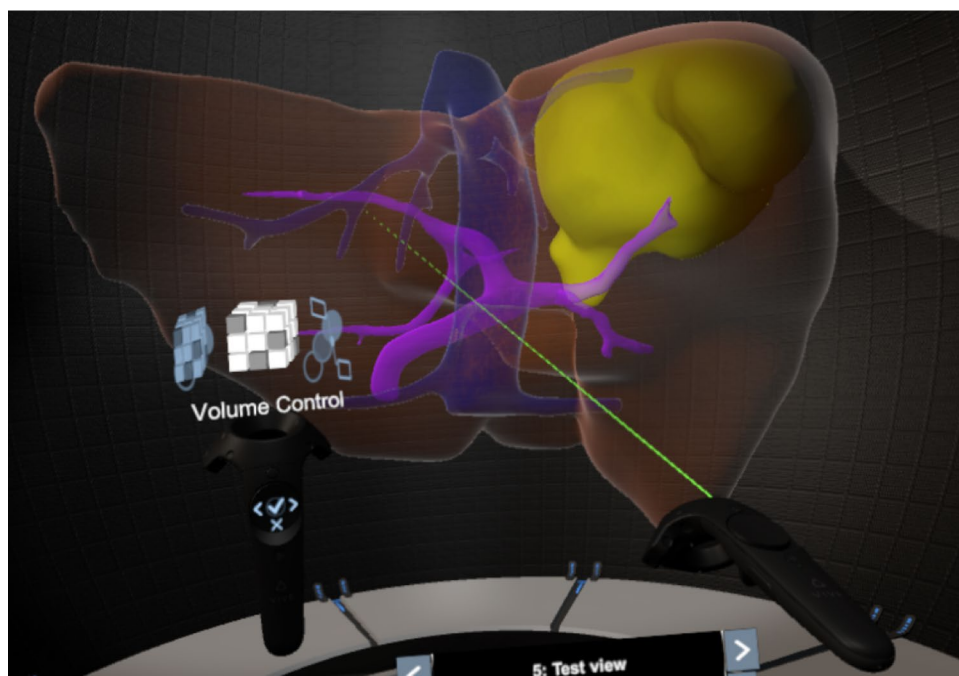
Inside a Unity project with the plugin, several features are provided: importing DICOM volumes and manipulating scale, rotation and position; adjusting volume opacity, brightness and transfer function (grayscale or with a color gradient); filtering data according to the window level and window width (thresholding); isolating 2D slices; capturing screenshots and more. During experimentation, a personalized volume cutting plane was achieved, where the user can choose to see the interior of the volume or the corresponding 2D slice in black and white—this can be seen in Fig. 8.

This solution fulfills most of the features listed in Table 2, allowing even the implementation of the duplicate volume controls, all based on the standard development process of Unity. Measurement tools, 3D drawings and labeling can also be added as pure Unity features, although they require the implementation of a calibration mechanism to inform Unity about the volume's real-life scale (achievable through DICOM's metadata, for example). Additionally, the authors' performance tests show that their solution achieves high values of frames per second (fps), reaching 90 fps with objects at two meters of distance and 60 fps at half a meter. Image quality is adjustable as well, meaning that applications built with the plugin can run on machines with different processing capacities, with less powerful machines trading quality for performance.

G. Wheeler et al. are now working on providing the plugin as a Unity asset, increasing once more its integration with the engine, but perhaps VtkToUnity's greatest challenge is the implementation of alternative segmentation algorithms. Threshold-based algorithms are light and usable in real-time, but are limited in accuracy and precision; when importing

<sup>9</sup> <http://www.janegger.de/Opportunities/MeVisLab-Unity%203D.pdf>

**Fig. 9** Screen capture of the IMHOTEP prototype, displaying a liver with a tumor, each mesh with different opacity levels



3D scans of patients' thorax, it was possible to filter bone structures but tweaking the window values did not allow to fully separate heart from lungs and other tissues. More sophisticated solutions involving user interaction such as active shape models [77] or those implemented in products such as Elucis, HoloDICOM or True 3D, although requiring time and skill, are able to deliver more precise results, but face integration issues with this plugin as they involve specific functions VTK might provide but the authors' connector lack.

Pfeiffer et al. present the IMHOTEP framework as another means to bring the data processing capabilities of Kitware's software to the development environment of the Unity game engine. As described in [Development Tools and Frameworks](#), IMHOTEP is designed specifically for the purposes we are exploring and has a modular architecture that connects software for 3D modeling, data visualization, image processing and application development, making it a powerful development framework.

Their default Unity project already contains a complex set up of a workstation that places the user on a control area with three main views: the lateral walls containing patients' information on the left and a 2D view of the data (a slice) on the right; and the front view presenting a large virtual spherical room. Once the data is uploaded, the room generates either a 3D volume or a pre-treated 3D surface mesh of the patient's scan. The authors adopted a manipulation approach different from the traditional direct control of objects in 3D space; in IMHOTEP, the objects are distant from the user and amplified, the user can still rotate and scale the data but not move it like an object in

space, and he/she can also save default views - this is yet to be formally evaluated whether this is more intuitive and/or more appropriate for the target audience or not.

If previously segmented surface meshes exist, these can be displayed in the virtual scene and individual colors can be assigned to the various meshes before they are opened and displayed by the framework. The meshes (representing structures such as organs, vessels and tumors) are generated using external tools and can only be imported if they are previously loaded into the Blender3D software and saved in the native .blend file format. Although the usage advantage of these meshes is not questionable, they fall out of the scope of our study, as they involve additional external tools for processing the patient's data and segmenting it into different 3D objects.

If 3D volumes (such as the slices resulting from a CT or MRI scan in DICOM format) are available, they can be displayed via volumetric rendering. This is realized by a custom GPU shader program, and transfer functions are once again used as a threshold-based segmentation tool. Experimentations showed poor performance results when importing the same test volumes inside IMHOTEP, suggesting the need for future rendering optimization.

Like the VtkToUnity plugin, IMHOTEP contains prefabs for the operations linked to the VR controllers, but in the framework's case, these are very easily incremented since the UI elements attached to the controllers only appear when the user wishes to and each feature can be selected through a virtual circular carousel, as seen in Fig. 9. Moreover, the authors have instructions in their documentation on how to create new controller features.

The 2D slice presented on the right wall corresponds to a plane intersecting the object in front of the user. This image can also have its brightness adjusted, and the plane can be placed in any angle and position of the patient's data - once again, the interaction method is different than what would be expected and less intuitive. As this was already implemented, creating a cutting plane derived from this intersection plane required less effort. It was possible to tweak the mesh shaders in order to achieve a cutting feature, but it was not found particularly interesting for segmented surface meshes and due to performance issues we were not able to map it to the 3D volume view.

Developing a medical imaging visualization application based on IMHOTEP does not necessarily require the use of their predefined tools. For example, A. Lowele et al. [78] created a VR system called FaMaS-VR that allows to interactively simulate atrial excitation propagation and place ablation lesions. Their system is based on the IMHOTEP framework with a fast marching algorithm for the simulations implemented by the authors. Although requiring high capacity software to run, FaMaS-VR showed very positive quality results. The authors also understood that the current state of their system is most suited "for training and educational purposes" but propose means to properly pave the way for clinical application.

## Conclusions and Future Work

This paper presents an analysis of the current state-of-the-art of medical imaging visualization applications in virtual reality. Based on this, we derived the main features a system should have to be useful under the context of clinical practice, namely the free control over patients' data, opacity adjustment and interactive tools such as volume cutting planes and automatic and semi-automatic segmentation algorithms. Our research also showed that most of the existing solutions do not have sufficient performance to be adopted in the surgical environment and serve primarily as tools for education and training.

Based on our findings, we conducted an exploratory study on existing development tools and frameworks that could provide similar visualization applications or even expand the existing knowledge on the field. It was clear that most technologies rely on well-established toolkits specialized for healthcare and standard data formats such as DICOM. In terms of development environments, game engines are undoubtedly the elected choice, as they offer immense support and facilitate the combination of software modules for more valuable results. Technologies such as the VtkToUnity plugin and the IMHOTEP

framework are already powerful options to study and compare immersive medical imaging visualization applications against traditional 2D visualization stations. They are built entirely for scenarios of image analysis and diagnosis, but show great potential for future growth to other scenarios.

Also, our research and experimentation results suggest that there are still many limitations that must be overcome for the adoption of VR solutions in medicine. Real-time performance, high precision and minimum latency are demanding prerequisites that present the first challenge for such systems. Although new generation hardware devices continue to address these technological challenges, real-time high-fidelity image rendering has still much to be improved. Automatic setup and calibration also become critical when millimeter accuracy is required. Human-Computer Interaction (HCI) is another important aspect to take into consideration. For image analysis and disease diagnosis, it is crucial that the practitioner can quickly perform tasks and go through several patients at an equivalent pace when compared to the traditional 2D-based analysis. For real-time in-surgery solutions, AR tends to be the elected technology since VR prevents doctors from being aware of both the patient and the virtual information, which is naturally unfeasible. For personnel training, the use of haptic devices is highly advisable and should be explored in greater depth, as it can also push the research on remote procedures - where VR becomes a potentially beneficial in-surgery option.

Even though the past five years have presented an exponential growth of immersive solutions, particularly in medicine, the technology has not yet achieved its true potential and lacks studies comparing Mixed Reality and currently used technologies and determining the feasibility of these technologies complementing each other. Future work should focus on supporting the research community with tools that ease the use of medical data formats and allow developers to focus on faster algorithms, more intuitive system interactions and specialized user experiences.

**Funding Information** This work was partially funded by FCT - Foundation for Science and Technology, in the context of the projects EHDEN-H2020/806968 and UIDB/00127/2020.

## Declarations

**Conflict of Interest** The authors declare that they have no conflict of interest.

**Ethical Approval** This article does not contain any studies with human participants or animals performed by any of the authors.

**Informed Consent** This article does not contain patient data.

## References

1. HTC Corporation, HTC VIVE Cosmos: own your VR world, in <https://vive.com>. Retrieved April 2020.
2. Oculus Go, Standalone VR Headset: Our all-in-one headset made for entertainment., in *oculus.com*. Retrieved April 2020.
3. Perkins Coie & XR Association. (2019). Augmented and Virtual Reality Survey Report. Industry Insights into the Future of Immersive Technology, vol. 3.
4. Lorensen, William & Cline, Harvey. (1987). Marching Cubes: A High Resolution 3D Surface Construction Algorithm. ACM SIGGRAPH Computer Graphics. 21. 163–169. <https://doi.org/10.1145/37401.37422>.
5. Chen, Long & Day, Thomas & Tang, Wen & John, Nigel. (2017). Recent Developments and Future Challenges in Medical Mixed Reality. 10.1109/ISMAR.2017.29.
6. Aouam, Djamel & Zenati-Henda, Nadia & Benbelkacem, Samir & Hamitouche, Chafiaa. (2020). An Interactive VR System for Anatomy Training. IntechOpen, <https://doi.org/10.5772/intechopen.91358>.
7. 4D Anatomy: AR Mobile Application, by *4D Interactive Anatomy*. Retrieved April 2020.
8. Xin Wang and Xiuyue Wang. 2018. Virtual Reality Training System for Surgical Anatomy. In Proceedings of the 2018 International Conference on Artificial Intelligence and Virtual Reality (AIVR 2018). Association for Computing Machinery, New York, NY, USA, 30–34. <https://doi.org/10.1145/3293663.3293670>.
9. Cabrera, Mildred & Carrillo, Jos & Nigenda, Juan & Gonzalez, Ricardo & Valdez, Jorge & ChavarRa, Belinda. (2019). Assessing the Effectiveness of Teaching Anatomy with Virtual Reality. 43–46. <https://doi.org/10.1145/3369255.3369260>.
10. Makled, Elhassan & Yassien, Amal & Elagroudy, Passant & Magdy, Mohamed & Abdennadher, Slim & Hamdi, Nabila. (2019). PathoGenius VR: VR medical training. 1–2. <https://doi.org/10.1145/3321335.3329694>.
11. Papagiannakis, George & Lydatakis, Nick & Kateros, Steve & Georgiou, Stelios & Zikas, Paul. (2018). Transforming medical education and training with VR using M.A.G.E.S. 1–2. <https://doi.org/10.1145/3283289.3283291>.
12. Huber, Tobias & Wunderling, Tom & Paschold, Markus & Lang, Hauke & Kneist, Werner & Hansen, Christian. (2017). Highly Immersive Virtual Reality Laparoscopy Simulation: Development and Future Aspects. International Journal of Computer Assisted Radiology and Surgery. 13. <https://doi.org/10.1007/s11548-017-1686-2>.
13. Gonzalez IZard, Santiago & Juanes, Juan. (2016). Virtual reality medical training system. Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM 16). 479–485. <https://doi.org/10.1145/3012430.3012560>.
14. Paiva, Paulo & Machado, Liliane & Valena, Ana & Batista, Thiago & Moraes, Ronei. (2018). SimCEC: A Collaborative VR-Based Simulator for Surgical Teamwork Education. Computers in Entertainment. 16. 1–26. <https://doi.org/10.1145/3177747>.
15. Javaux, Allan & Bouget, David & Gruijthuijsen, Caspar & Stoyanov, Danail & Vercauteren, Tom & Ourselin, Sebastien & Deprest, Jan & Denis, Kathleen & Vander Poorten, Emmanuel. (2018). A mixed-reality surgical trainer with comprehensive sensing for fetal laser minimally invasive surgery. International Journal of Computer Assisted Radiology and Surgery. 13. <https://doi.org/10.1007/s11548-018-1822-7>.
16. Holland, Mark & Pop, Serban & John, Nigel. (2017). VR Cardiovascular Blood Simulation as Decision Support for the Future Cyber Hospital. 233–236. <https://doi.org/10.1109/CW.2017.49>
17. Tan, Wenjun & Ge, Wen & Hang, Yucheng & Wu, Simeng & Liu, Sixing & Liu Ming (2018). Computer assisted system for precise lung surgery based on medical image computing and mixed reality. Health Inf Sci Syst 6, 10. <https://doi.org/10.1007/s13755-018-0053-1>.
18. Ayoub, Ashraf & Pulijala, Yeshwanth. (2019). The application of virtual reality and augmented reality in Oral & Maxillofacial Surgery. BMC Oral Health. 19. <https://doi.org/10.1186/s12903-019-0937-8>.
19. Li, Feiyan & Yonghang, Tai & Li, Qiong & Peng, Jun & Xiaoqiao, Huang & Chen, Zaiqing & Shi, Junsheng. (2019). Real-Time Needle Force Modeling for VR-Based Renal Biopsy Training with Respiratory Motion Using Direct Clinical Data. Applied Bionics and Biomechanics. 2019. 1–14. <https://doi.org/10.1155/2019/9756842>.
20. Malpani, Anand & Vedula, Sunil & Lin, Henry & Hager, Gregory & Taylor, Russell. (2020). Effect of real-time virtual reality-based teaching cues on learning needle passing for robot-assisted minimally invasive surgery: a randomized controlled trial. International Journal of Computer Assisted Radiology and Surgery. 15. <https://doi.org/10.1007/s11548-020-02156-5>.
21. Mehralivand, Sherif & Kolagunda, Abhishek & Hammerich, Kai & Sabarwal, Vikram & Harmon, Stephanie & Sanford, Thomas & Gold, Samuel & Hale, Graham & Romero, Vladimir & Blom, Jonathan & Merino, Maria & Wood, Bradford & Kambhamettu, Chandra & Choyke, Peter & Pinto, Peter & Turkbey, Baris. (2019). A multiparametric magnetic resonance imaging-based virtual reality surgical navigation tool for robotic-assisted radical prostatectomy. Trk roloji Dergisi/Turkish Journal of Urology. 45. 357–365. <https://doi.org/10.5152/tud.2019.19133>.
22. Pan, Zhaoxi & Tian, Song & Guo, Mengzhao & Zhang, Jianxun & Yu, Ningbo & Xin, Yunwei. (2017). Comparison of medical image 3D reconstruction rendering methods for robot-assisted surgery. 94–99. <https://doi.org/10.1109/ICARM.2017.8273141>.
23. Ruthenbeck, Greg & Reynolds, Karen. (2014). Virtual Reality for Medical Training: The State of the Art. Journal of Simulation. 9. <https://doi.org/10.1057/jos.2014.14>.
24. Sharkey, P.M. & Merrick, J.. (2014). Virtual reality: Rehabilitation in motor, cognitive and sensorial disorders. <http://site.ebrary.com/id/10929399>.
25. Mitrousia, V. & Giotakos, O. (2016). Virtual reality therapy in anxiety disorders. Psychiatriki. 27. 276–286. <https://doi.org/10.22365/jpsych.2016.274.276>.
26. Wiederhold, B.. (2018). Are We Ready for Online Virtual Reality Therapy?. Cyberpsychology, Behavior, and Social Networking, 21: 341–342. <https://doi.org/10.1089/cyber.2018.29114.bkw>.
27. Dias, Paulo & Silva, Ricardo & Amorim, Paula & Lains, Jorge & Roque, Eulalia & Pereira, Ines & Pereira, Fatima & Santos, Beatriz & Potel, Mike. (2019). Using Virtual Reality to Increase Motivation in Poststroke Rehabilitation. IEEE Computer Graphics and Applications. 39. 64–70. <https://doi.org/10.1109/MCG.2018.2875630>.
28. Moreira, Gabrielly & Machado, Ingrid & Lima, Elis & Loureiro, Ana Paula & Manfra, Elisangela. (2020). The Use of Virtual Reality Rehabilitation for Individuals Post Stroke. 1. 21–27.
29. August, K. & Sellathurai, M. & Bleichenbacher, D. & Adamovich, S., VESLI Virtual Reality Rehabilitation for the Hand, 2020.
30. Weiss, P.L.T. & Weintraub, N. & Laufer, Y., Virtual reality therapy in paediatric rehabilitation, 2016.
31. W. Dean Bidgood, Jr & Steven C. Horii & Fred W. Prior & Donald E. Van Syckle. (1997). Understanding and Using DICOM, the Data Interchange Standard for Biomedical Imaging, Journal of the American Medical Informatics Association, Volume 4, Issue 3, Pages 199–212. <https://doi.org/10.1136/jamia.1997.0040199>.
32. R.A. Robb & D.P. Hanson & R.A. Karwoski & A.G. Larson & E.L. Workman & M.C. Stacy. (1989). Analyze: A Comprehensive, operator-interactive software package for multidimensional medical image display and analysis. Computerized Medical Imaging and Graphics, 13 (6): 433–454. [https://doi.org/10.1016/0895-6111\(89\)90285-1](https://doi.org/10.1016/0895-6111(89)90285-1).

33. NIfTI Documentation, in <https://nifti.nimh.nih.gov>. Retrieved April 2020.
34. The McConnell Brain Imaging Centre, MINC software library and tools, in <https://bic.mni.mcgill.ca>. Retrieved April 2020.
35. Kahn, Jr, Charles & Carrino, John & Flynn, Michael & Peck, Donald & Horii, Steven. (2007). DICOM and radiology: Past, present, and future. *Journal of the American College of Radiology : JACR*. 4. 652-7. <https://doi.org/10.1016/j.jacr.2007.06.004>.
36. National Electrical Manufacturers Association, Digital Imaging and Communications in Medicine, in <https://dicomstandard.org>. Retrieved April 2020.
37. National Electrical Manufacturers Association, WG-17 3D, in <https://dicomstandard.org>. Retrieved April 2020.
38. Realize Medical, Elucis: The Future of Medical Modeling, in <https://realizemed.com>. Retrieved April 2020.
39. Diffuse, Spectro, in <https://diffuse.ch>. Retrieved April 2020.
40. C. Luciano & P. Banerjee & L. Florea & G. Dawe, Design of the ImmersiveTouch: a high-performance haptic augmented virtual reality system, 2014.
41. ImmersiveTouch, ImmersiveView VR, in <https://immersivetouch.com>. Retrieved April 2020.
42. Imaging Reality, About Imaging Reality, in <https://imagingreality.com>. Retrieved April 2020.
43. DICOM VR, DICOM VR: a new way of viewing volumetric medical imaging and planning targeted radiation treatment in virtual reality, in <https://dicomvr.com>. Retrieved April 2020.
44. Surgical Theater, Precision VR, in <https://surgicaltheater.net>. Retrieved April 2020.
45. HoloDICOM, HoloDICOM: diagnostic medical image in augmented reality, in <https://holodicom.com>. Retrieved April 2020.
46. Seal, Arindrajit & Das, Arunava & Sen, Prasad. (2015). Watershed: An Image Segmentation Approach. *International Journal of Computer Science and Information Technologies*, 6(3):2295-2297. 0975-0946.
47. Kemmling, Andre & Minnerup, Heike & Berger, Karsten & Knecht, Stefan & Groden, Christoph & Nlte, Ingo. (2012). Decomposing the Hounsfield Unit Probabilistic Segmentation of Brain Tissue in Computed Tomography. *Clinical neuroradiology*. 22. 79-91. <https://doi.org/10.1007/s00062-011-0123-0>.
48. EchoPixel, True3D, in <https://echopixeltech.com>. Retrieved April 2020.
49. Mohammed, Mohammed & Khalaf, Mosbah & Kesselman, Andrew & Wang, David & Kothary, Nishita. (2018). A Role for Virtual Reality in Planning Endovascular Procedures. *Journal of vascular and interventional radiology : JVIR*. 29. 971-974. 10.1016/j.jvir.2018.02.018.
50. Kaley, Vishal & Aregullin, Enrique & Samuel, Bennett & Vettukattil, Joseph. (2018). Transcatheter Intervention for Paravalvular Leak in Mitroflow Bioprosthetic Pulmonary Valve. <https://doi.org/10.12945/j.jshd.2018.006.18>.
51. Huang, Chenxi & Zhou, Wen & Lan, Yisha & Chen, Fei & Hao, Yongtao & Cheng, Yongqiang & Peng, Yonghong. (2018). A Novel WebVR-based Lightweight Framework for Virtual Visualization of Blood Vasculum. *IEEE Access*. PP. 1-1. 10.1109/ACCESS.2018.2840494.
52. Adochiei, Felix & Ciucu, Radu & Adochiei, Ioana & Grigorescu, Dan & Seritan, George & Miron, Casian. (2019). A WEB Platform for Rending and Viewing MRI Volumes using Real-Time Raytracing Principles. 1-4. <https://doi.org/10.1109/ATEE.2019.8724963>.
53. Gonzalez Izard, Santiago & Plaza, Oscar & Torres, Ramiro & Juanes, Juan & Garcia-Pealvo, Francisco. (2019). NextMed, Augmented and Virtual Reality platform for 3D medical imaging visualization: Explanation of the software platform developed for 3D models visualization related with medical images using Augmented and Virtual Reality technology. 459-467. <https://doi.org/10.1145/3362789.3362936>.
54. Reddivari, Sandeep & Smith, Jason & Pabalate, Jonathan. (2017). VRvisu: A Tool for Virtual Reality Based Visualization of Medical Data. 280-281. <https://doi.org/10.1109/CHASE.2017.102>.
55. Ard, Tyler & Krum, David & Phan, Thai & Duncan, Dominique & Essex, Ryan & Bolas, Mark & Toga, Arthur. (2017). NIVR: Neuro imaging in virtual reality. 465-466. <https://doi.org/10.1109/VR.2017.7892381>. Sandeep Reddivari and Jason Smith and Jonathan Pabalate. (2017). V Rvisu: a tool for virtual reality based visualization of medical data. *Proceedings of the Second IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE 17)*. IEEE Press, 280-281. <https://doi.org/10.1109/CHASE.2017.102>
56. The Visualization Toolkit, in <https://vtk.org>. Retrieved April 2020.
57. Schroeder, William & Martin, K. & Lorensen, William. (2006). *The Visualization Toolkit, An Object-Oriented Approach To 3D Graphics*. ISBN 978-1-930934-19-1.
58. Ibanez, L. & Schroeder, W. & Ng, L. & Cates, J., *The ITK Software Guide: The Insight Segmentation and Registration Toolkit*, 2005. Kitware Inc.
59. Enquobahrie, Andinet & Cheng, Patrick & Gary, Kevin & Ibanez, Luis & Gobbi, David & Lindseth, Frank & Yaniv, Ziv & Aylward, Stephen & Jomier, Julien & Cleary, Kevin. (2007). The Image-Guided Surgery Toolkit IGSTK: An open source C++ software toolkit. *Journal of digital imaging : the official journal of the Society for Computer Applications in Radiology*. 20 Suppl 1. 21-33. <https://doi.org/10.1007/s10278-007-9054-3>.
60. 3DSlicer: a multi-platform, free and open source software package for visualization and medical image computing, in <https://slicer.org>, by Kitware, Inc. Retrieved April 2020.
61. ParaView, in <https://paraview.org>, by Kitware, Inc. Retrieved April 2020.
62. Amorim, Paulo & Franco de Moraes, Thiago & Pedrini, Helio & Silva, Jorge. (2015). InVesalius: An Interactive Rendering Framework for Health Care Support. [https://doi.org/10.1007/978-3-319-27857-5\\_5](https://doi.org/10.1007/978-3-319-27857-5_5).
63. B. Fischl. (2012). FreeSurfer. *NeuroImage*. 62. 774-81. <https://doi.org/10.1016/j.neuroimage.2012.01.021>.
64. OsiriX DICOM Viewer, in <https://osirix-viewer.com>, by Pixmeo. Retrieved April 2020.
65. Nicolas Baghdadi & Clment Mallet & Mehrez Zribi. (2018). QGIS and Generic Tools. <https://doi.org/10.1002/9781119457091>.
66. O'Leary, Patrick & Jhaveri, Sankhesh & Chaudhary, Aashish & Sherman, William & Martin, Ken & Lonie, David & Whiting, Eric & Money, James & McKenzie, Sandy. (2017). Enhancements to VTK enabling scientific visualization in immersive environments. 186-194. <https://doi.org/10.1109/VR.2017.7892246>.
67. S. Arikatla & J. Fillion-Robin & B. Paniagua & M. Holden & Z. Keri & M. Jolley & A. Lasso & A. Enquobahrie, Bringing Virtual Reality to 3D Slicer, in <https://blog.kitware.com>, 2018. Retrieved April 2020.
68. K. Martin & D. DeMarle & S. Jhaveri & U. Ayachit, Taking ParaView into Virtual Reality, in <https://blog.kitware.com>, 2018. Retrieved April 2020.
69. Unity Technologies, Unity for All, in <https://unity.com>. Retrieved April 2020.
70. Epic Games, Unreal Engine: Make something Unreal, in <https://unrealengine.com>. Retrieved April 2020.
71. Wheeler, Gavin & Deng, Shujie & Toussaint, Nicolas & Pushparajah, Kuberan & Schnabel, Julia & Simpson, John & Gomez, Alberto. (2018). Virtual Interaction and Visualisation of 3D Medical Imaging Data with VTK and Unity. *Healthcare Technology Letters*. 5. <https://doi.org/10.1049/htl.2018.5064>.
72. MeVis Medical Solutions AG, MeVisLab: A powerful modular framework for image processing research and development, in <https://mevislab.de>. Retrieved April 2020.
73. Muhler, Konrad & Preim, Bernhard. (2010). Reusable Visualizations and Animations for Surgery Planning. *Computer Graphics*

- Forum. 29. 1103-1112. <https://doi.org/10.1111/j.1467-8659.2009.01669.x>.
74. Gunacker, Simon & Gall, Markus & Schmalstieg, Dieter & Egger, Jan. (2018). Multi-Threaded Integration of HTC-Vive and MeVis-Lab. <https://doi.org/10.13140/RG.2.2.18864.05121>.
75. National Center for Tumor Disease, Dresden, University Hospital, Immersive Medical Hands-On Operation Teaching and Planning System, in <https://imhotep-medical.org>. Retrieved April 2020.
76. Pfeiffer, Micha & Kenngott, Hannes & Preukschas, Anas & Huber, Matthias & Bettscheider, Lisa & Muller, Beat & Speidel, Stefanie. (2018). IMHOTEP - Virtual Reality Framework for Surgical Applications. International Journal of Computer Assisted Radiology and Surgery. 13. <https://doi.org/10.1007/s11548-018-1730-x>.
77. Olveres, Jimena & Nava, Rodrigo & Escalante-Ramirez, Boris & Cristobal, Gabriel & Mara, Carla. (2013). Midbrain volume segmentation using Active Shape Models and LBPs. Proceedings of SPIE - The International Society for Optical Engineering. 8856. <https://doi.org/10.1117/12.2024396>.
78. Loewe, Axel & Poremba, Emanuel & Oesterlein, Tobias & Pilia, Nicolas & Pfeiffer, Micha & Doessel, Olaf & Speidel, Stefanie. (2017). An Interactive Virtual Reality Environment for Analysis of Clinical Atrial Arrhythmias and Ablation Planning. <https://doi.org/10.22489/CinC.2017.125-118>.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.