MAGES 4.0: Accelerating the World's Transition to VR Training and Democratizing the Authoring of the Medical Metaverse

Paul Zikas, Antonis Protopsaltis, Nick Lydatakis, Mike Kentros, Stratos Geronikolakis, Steve Kateros, Manos Kamarianakis, Giannis Evangelou, Achilleas Filippidis, Eleni Grigoriou, Dimitris Angelis, Michail Tamiolakis, Michael Dodis, George Kokiadis, John Petropoulos, Maria Pateraki, and George Papagiannakis, *ORamaVR SA, 1228, Geneva, Switzerland*

In this work, we propose MAGES 4.0, a novel software development kit to accelerate the creation of collaborative medical training applications in virtual/augmented reality (VR/AR). Our solution is essentially a low-code metaverse authoring platform for developers to rapidly prototype high-fidelity and high-complexity medical simulations. MAGES breaks the authoring boundaries across extended reality, since networked participants can also collaborate using different VR/AR as well as mobile and desktop devices, in the same metaverse world. With MAGES we propose an upgrade to the outdated 150-year-old master–apprentice medical training model. Our platform incorporates, in a nutshell, the following novelties: 1) 5G edge-cloud remote rendering and physics dissection layer, 2) realistic real-time simulation of organic tissues as soft-bodies under 10 ms, 3) a highly realistic cutting and tearing algorithm, 4) neural network assessment for user profiling and, 5) a VR recorder to record and replay or debrief the training simulation from any perspective.

The medical metaverse, despite the inflated expectations, is steadily, albeit quietly, being created.[15] Along with it, many technical questions remain, including "who will build the medical metaverse and how?" Building such an ecosystem from few stakeholders would require significant effort involving tasks of tremendous complexity, unless the metaverse authoring process is decentralized and the tools to create it are democratized in the hands of the actual content creators.

The extended pandemic crisis highlighted the need for effective medical training, along with the inadequacy of the 150-year-old surgical training model.[14] Computational medical science aims to accelerate the world's transition to virtual reality (VR) medical training in the metaverse[15] and empower medical professionals to enhance their proficiency and ultimately improve patient outcomes (see S1 in the Sidebar).[12,16,17]

To serve the causes above, we present MAGES 4.0, the medical VR industry's first software development kit (SDK) that allows rapid prototyping of any shared, collaborative networked medical training in VR, in a fraction of time and cost. It draws its robustness from the 20 years of academic research and development, incorporating the latest advancements into a novel medical virtual/augmented reality (VR/AR) SDK.

The computational results achieved with MAGES SDK exceed those typically reached by large teams of domain expert developers of similar engines. Being layered on top of existing game engines, such as Unity3D and Unreal Engine (see Figures 1 (left) and 2), it brings a low-code virtual world authoring platform to developers with even a moderate knowledge of these engines. This allows a small team, of one developer and one designer, as in our latest use case (see

This article has supplementary downloadable material available at https://doi.org/10.1109/MCG.2023.3242686, provided by the authors.
Digital Object Identifier 10.1109/MCG.2023.3242686
Date of publication 6 February 2023; date of current version 23 March 2023.

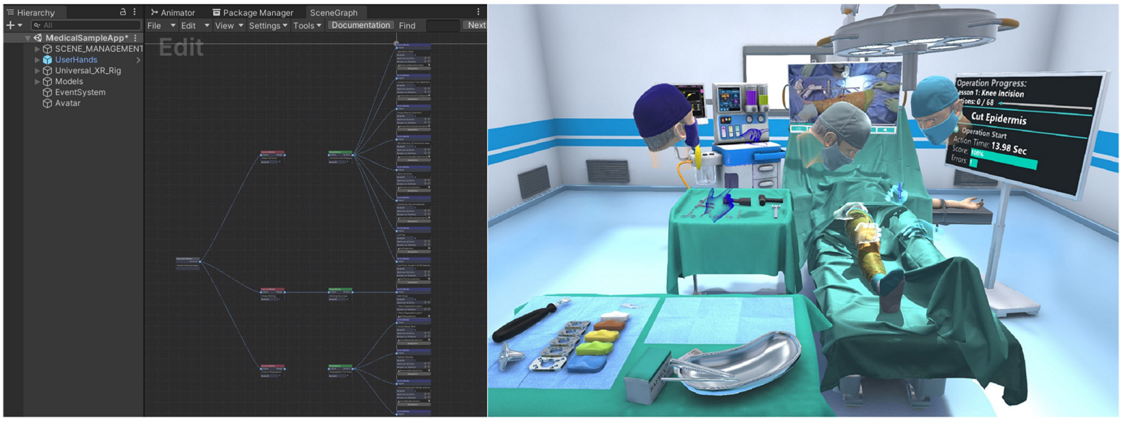

FIGURE 1. (Left) Visual scripting editor. (Right) A surgeon with two students performing a collaborative total knee arthroplasty as generated using MAGES, in Unity3D.

the "Case Studies" section), to create a complete VR medical training simulation even within two weeks.

PREVIOUS WORK

In the past years, a variety of metaverse authoring tools have been introduced,[1] tailored to particular research problems and application verticals, most of them presented as proof of concept. A great number of platforms have been designed for entertainment/multimedia systems,[2] education/vocational training,[4] (see S3 in the Sidebar) cultural heritage, and medical training (see S6 in the Sidebar).[3,12,16,17] Each simulation has various strengths, but because they were often designed with specific use cases in mind, each is also limited in different ways. Most such VR systems provide high-fidelity photorealistic scenes,[2] user immersion and presence that allow human-centric interaction in VR.[5] Existing simulation environments are differentiated as they combine VR training functions in various domains, while others, only very few of them,[3] provide realistic physics simulations on surfaces with rigid body, soft body, cloth, and fluids. Innovative multiuser collaborative virtual environments (VE),[2,3,12,16] provide a great asset in all fields, a fact that was especially highlighted during the COVID-19 pandemic.

Few of the abovementioned VE authoring tools,[3,4] allow reuse/import of content/assets that impacts the tool's usability and effectiveness, as it enables designers to significantly reduce their effort, and generate custom user experience that otherwise would be impossible. In that respect, some novel, innovative, and interoperable authoring platforms allow users with no or limited programming knowledge to design a VE fast and easy,[3] setting early the basic and essential requirements for the building of the metaverse.[1] In that respect, VR software design patterns[6] is a momentum-gaining approach, with already concrete results, that will be utilized in the building process of the next-generation VR training applications for the metaverse.

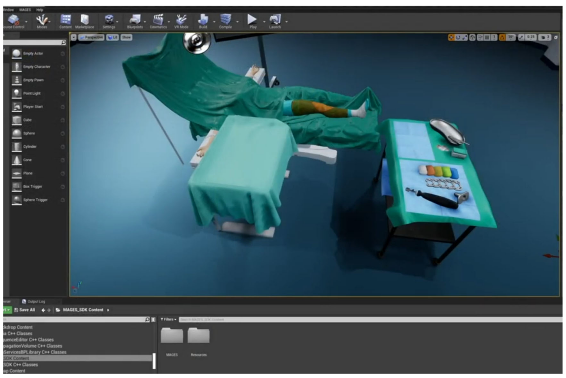

FIGURE 2. Developing a total knee arthroplasty with MAGES in Unreal Engine.

MAGES 4.0 INNOVATIONS

The innovations, presented in this work, extend significantly our previous work,[3] and thus transform our SDK to a powerful low-code VR authoring platform, that allows developers to produce high-fidelity and high-complexity VR simulations fast and effectively, empowering the medical metaverse creation. MAGES SDK is named after its most unique features as follows.

Multiplayer With GA Interpolation (M)

A custom geometric algebra (GA) interpolation engine allows up to 300 simultaneous users in the same session of the VE. To the best of our knowledge, this is the highest number of active concurrent users in the same scene, whereas state-of-the-art applications offer only a few tens of concurrent users with only one active and rest being spectators. This remarkable achievement is accomplished by the transmission over the network layer of GA-based representation forms of the VR scene transformation data, which support higher data compression rates. The use of such alternative forms by MAGES reduces the required data that must be sent over the network by up to 33%, whereas the reduced amount of data sent in these forms yield visually better interpolated results, with the overall quality of experience being sufficient even when the network quality, in terms of bandwidth, seriously deteriorates.[7]

Analytics (A)

A powerful advanced analytic engine system allows tracking and visualization of the student's progress. As this feature enables scoring per user action in a simple manner, it provides an easy way for the tutor or the students to quickly inspect and identify the parts of the training performance. A main insight is that as main recorded data are based on GA, their storage, retrieval, or interpolation is much more efficient than existing methods.[10]

GA Deformable Animation, Cutting, and Tearing (G)

The under-the-hood GA engine used to solve the animation equation is responsible not only for model deformation but for a series of features that exploit it. Specifically, the ability to perform cuts, progressive tears, and drills on skinned soft-body meshes is now feasible in real time, with increased realism suitable even for demanding VR/AR immersive applications (under 10 ms frame rendering time).[13]

Editor With Action Prototypes (E)

An incorporated visual scripting editor along with our custom VR simulation design patterns as basic building blocks, called *actions*, are used to rapidly accelerate content creation for VR/AR simulations. Based on software design patterns,[6] they enable creation of medical training operations, at a fraction of the time and cost, against current practices and standards.

Semantically Annotated Deformable, Soft, and Rigid Bodies (S)

Toward a highly realistic recreation of a virtual surgery operation, we have designed and developed a novel particle system, suitable for real-time elasticity simulations of human tissues and organs in VR/AR.[13]

The abovementioned novel features of MAGES 4.0 were significantly enhanced, compared to our previous work,[3] in order to provide higher fidelity for incisions, increased performance for soft-bodies, extended capabilities for ML analytics (connection with the built in VR recorder), ability to handle more concurrent users via the GA networking layer and support for a larger set of actions for the scripting editor, thus paving the way for MAGES 4.0 to become a complete low-code metaverse authoring tool.

HOW MAGES WORKS IN FIVE STEPS

In this section, we present the five main steps a developer should follow to create a VR training simulation.

1) *Design training storyboard:* Utilizing our visual scripting editor, developers create the steps of each scenario.
2) *Create virtual assets:* Gather all the medical 3-D content (tools, human anatomy, etc.).
3) *Author training actions:* Generate the action scripts using our action prototype design patterns. Developers create programmable actions along with analytics and network behavior using the embedded authoring tools.
4) *Build the medical VR training simulation:* Build and deploy the executable application to a wide range of supported headsets/platforms, operating systems and portable/desktop devices.
5) *License the simulation:* Optionally, connect the training simulation to an analytic server, using MAGES SDK cloud management license.

Code reusability and prototyping are two major principles in software architecture. The structure of software systems and the communication between its modules is described through software design patterns. Software design patterns are reusable solutions for common programming problems that often occur during software development.[8] In Kloiber et al. 's work,[18] the creation of prototyped behaviors in VR with a system to manage and visualize interactive actions was simplified. The system is able to analyze user behaviors in VR to export highly accurate maps of user engagement that can be used later for in-VR action understanding.

ACTION PROTOTYPES

Especially in training simulations, developers need to implement highly interactive behaviors for the students to follow. For this reason, we introduce action prototypes, as novel software design patterns, for low-code behavioral tasks in training scenarios. The novelty in such an approach lies in the fact that these software patterns are tailored specifically for authoring of VR/AR training experiences,[6] e.g., "Insert action," "Question/Answer action," "Tool Action," etc. We classified the majority of the physical tasks into programmable and easily extendable code patterns. We implemented basic behaviors, such as the insertion, removal, usage, or even cognitive behaviors, like questions into separate programmable entities. Developers can inherit those entities and develop their own actions with a few lines of code.

To minimize the need of coding, we developed an authoring tool to automatically generate networked-ready, analytics-ready action scripts. Developers can create such action scripts by selecting the necessary physical objects for each task (e.g., tools, specific parts of the body) and subsequently the system will automatically generate an action script. This script contains the basic-default behavior of the particular object, easily extendable by mounting extension scripts.

TRAINING SCENEGRAPH

A training scenario contains a number of carefully defined steps, in a sequential or multipath manner. MAGES usage depends on an underlying training *scenegraph*, a highly dynamic, acyclic graph representing the training scenarios. Each node defines an *action*, a specific task to be completed by the trainee.

A training scenegraph is not just a static tree, it is a dynamic graph.[6] An educational scenario can lead to multiple paths according to user's actions and decisions. To accommodate this need, the training scenegraph can generate new paths and deviate from the originally intended path while the user explores the training scenario.

To rapidly accelerate the content creation we built a visual scripting authoring tool. This is a low-code system that enables easy authoring of actions while presenting the scenario from a higher level perspective. The visual scripting editor is able to visualize the training scenegraph and, as it consists of interactive nodes, allows the user developer to modify it. This feature is able to generate training simulations in a future proof implementation-agnostic way, via a user friendly GUI [see Figure 1 (left)].

Although VR medical training improves patient outcomes (see S1 in the Sidebar), VR content creation is a lengthy and costly process,[a] as the development of a small simulation requires four man-months for one designer and one developer. The use of the visual scripting editor allows the authoring of a small VR medical simulation by the same personnel in just 14 days (see the "Medical VR Training Examples Built With MAGES" sidebar for such examples), therefore, it is eight times faster and cost efficient than SoA methods.

ANALYTICS EDITOR

User assessment is crucial in medical simulations as students can identify their skills, while teachers can note any difficulties or points of improvement. For this reason, we integrated into MAGES an analytics system based on Zia et al. 's work[10] to assess and track the progress of each individual. In our training scenarios, the corresponding medical subject-matter-experts are the ones that direct which specific performance parameters to track throughout the medical operation. Those can vary from "wrong angle during incisions" or "tool placements," to "time constrained" decisions or even "contamination" issues.

MAGES features a novel analytic system to configure and present the assessment data. We built an easy to use interface on top of our visual scripting engine that enables analytic functionality to our actions, via minor modifications. For this reason, we introduced the scoring factor, a component that tracks data from objects or from the student's actions. Some of the scoring factors we implemented are velocity scoring factors (to track if a user is moving a fragile object too fast), error collider scoring factors (to check for possible contamination), as well as angle scoring factors (to check the proper placement of implants). Custom scoring factors are also supported allowing developers to extend the existing classes and create their own factors. Developers can assign multiple scoring factors in each action. The final score for each action is calculated as a weighted average of all assigned scoring factors.

After each simulation, users can visualize their analytics from within the VR/AR environment. Analytic reports are also uploaded automatically to our online portal allowing supervisors and teachers to view their students' progress. The implementation of the portal upload feature is as simple as a single API call in the MAGES configuration file.

AUTOMATIC HAND POSTURES

Interactive virtual characters are nowadays common place in VR/AR applications. Designing a virtual human-like hand, which is able to touch and grasp objects with realistic hand and finger adjustments, often trying to imitate the human brain way of reasoning, is a matter of

[a][Online]. Available: https://roundtablelearning.com/cost-of-virtual-reality-training-full-vr-2020

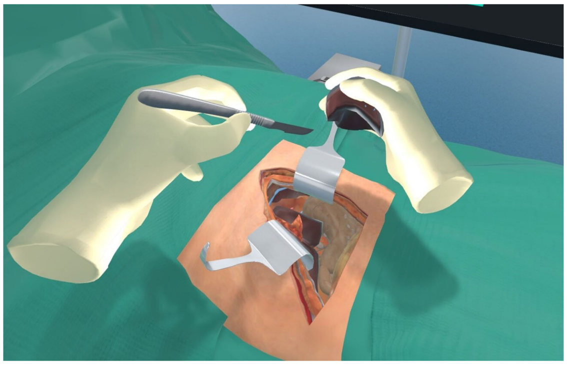

FIGURE 3. Student has just dissected the left part of the liver, during a hemihepatectomy operation. MAGES provides a realistic simulation of hand postures when grasping various objects.

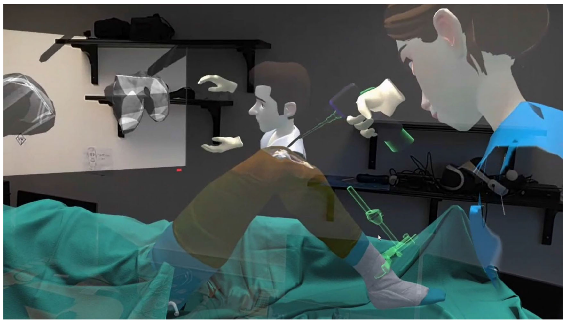

FIGURE 4. Collaborative total knee arthroplasty simulation between a Magic Leap (point of view) and an oculus quest 2 headset (the medical assistant).

necessity. Grasping hand animation itself includes several problematic aspects that make finding a satisfying solution extremely hard. Especially concerning human-like hands, the main challenge is trying to imitate the human-brain way of thinking and replicate the unconscious movements a person makes to grab an object. Although it is true that a 3-D physics engine can quite easily manage collisions and anatomical constraints, the real challenge is to design and implement a controller that supervises all the motors of the hand.[19] Predefined animation cannot be used in general on a physical body; therefore, even producing a simple movement requires particular care.

Reducing the number of possible grasp movements is fundamental. The applied methodology should also respect the physiology of the human hand. This applies to anatomical joint limits, angle limits due to tendon links between fingers (dynamic constraints) as well as angle limits that force a natural posture.

In MAGES, we designed an algorithm for intuitive object grasping with easy configuration and high accuracy, mimicking the human way of reasoning. This algorithm is flexible enough to support hand structures having an arbitrary number of fingers and an arbitrary number of phalanges for each finger. Given the hand bones structure, the position of the character and an object, the algorithm finds the most suitable grasping position for the character's hand on that object. Grasp poses are generated by a combination of a generic grasp movement and a target object that the fingers collide with. Movements are stored using only an initial pose and a final pose, so that the algorithm can generate an arbitrary number of interpolations between them. The algorithm runs in real time without imposing a performance overhead to the main program, even with complex objects (see Figure 3).

To simulate grasp poses, we implemented a procedure that, based on a starting hand position and a grasp movement, generates the grasping pose, by considering the object and computing the final positions and rotations for all bones. When the user grabs an object, starting from the root joint of the hand, we interpolate the joints to reach the final pose. If a bone gets in contact with the object, it is excluded from the following calculations. This algorithm runs recursively for each joint, resulting in a firm grip.

EXTENDED REALITY, MULTIMODAL DEVICE SUPPORT

Access to virtual worlds should be permitted in a seamless way, regardless the nature of the XR device used. For this reason, MAGES is designed to allow virtual sessions for both VR and AR headsets, mobile and desktop devices, without requiring complex configurations and additional work from the developer, allowing such users to collaborate in the same session (see Figure 4). In this regard, if a developer needs to build an application for an AR headset (e.g., HoloLens, Magic Leap, etc.), the only additional work required is to import the respective external SDK. Our universal XR camera interoperates with any HMD regardless of the targeted reality, by automatically culling unnecessary VE objects (e.g., operating room walls) when rendering for an AR device, breaking the boundaries between different XR technologies, that up until recently could not be combined.

EDGE-CLOUD REMOTE VISUAL RENDERING

Although the VR hardware landscape evolves rapidly, with SoC solutions that aim to reduce the performance gap to high-end desktop GPUs, still standalone/untethered VR solutions are less capable in supporting high-quality strong interactive VR services, due to their reduced GPU capabilities and battery life. This favors the exploitation of software solutions that offload computationally intensive tasks from end devices to cloud/edge resources to support processing and storage. Recently, the NVIDIA's CloudXR solution has emerged for remote rendering and streaming from OpenVR applications on a remote server cloud or edge, attained via a client application dependent on the HMD and the OS installed on the client's local machine. In our approach, we overcome the dependencies that regard the device's proprietary API's, thus supporting cross-platform access to VR devices. This is accomplished by handling rendering and streaming inside the offloaded application based on Unity Engine.

A signaling server is further integrated to complete the handshake and the interactive connectivity establishment[b] candidate exchange between the HMD and the offloaded application. The signaling server establishes a communication channel by a TURN server hosted inside the offloaded application, and decides which encoding format is the most appropriate to use for the video stream.

Although the Unity XR SDK is exploited for rendering, several custom modifications had to be applied. One major road blocker for VR offloading using Unity3D Render Streaming is that Unity3D does not support stereo render-to-texture. To cope with this limitation, we re-engineered the render streaming by using two complete Unity3D cameras in the scene. Each camera included customized settings for the left and right eye in order to render separate textures for each eye, which are then combined. Furthermore, a custom data stream is used to send transformation and controller data from the HMD to the offloaded application. With the simulation of two Unity3D cameras, the frame rate between the offloaded component and the HMD reaches 60 fps on average. The offloaded application runs inside of a Windows virtual machine (VM) on a linux host. By using KVM with GPU-passthrough, performance can come close to bare-metal speeds. In addition, a custom TURN server is hosted inside the VM, which allows WebRTC to consistently establish communication with the client, even if they happen to be behind a NAT.

DISSECTED EDGE-CLOUD REMOTE PHYSICS ENGINE

A typical monolithic game engine pipeline involves the execution of physics-related calculations, performed on CPU, alongside scene rendering calculations, performed on CPU/GPU. Both these heavy computations are either performed on high-end untethered HMDs, or on VR-ready PCs with a tethered HMD.

[b]For more information, also refer to the RFC 8445 standard.

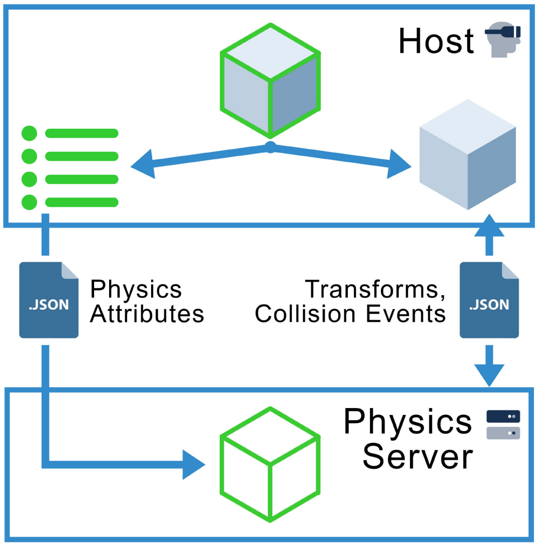

FIGURE 5. High-level architecture diagram of the physics dissection. The physics are calculated only on the physics server while the render service receives the rendered image.

The designed methods and techniques for the dissection of the Unity3D game engine pipeline creates two autonomous, deployed separately, bidirectionally communicating components: the *host* and the *physics server*. The host is responsible of maintaining the game logic and of processing the graphics rendering, while the physics server is responsible for performing physics computations (see Figure 5). The main goal of the dissected Unity3D pipeline is to allow any game object on the host's scene to be fully simulated by the separated physics server (see Figure 6).

The dissection approach is based on splitting the host's game objects into graphics objects, residing in the host, and physics objects, residing in the physics server. This is accomplished by eliminating the physics

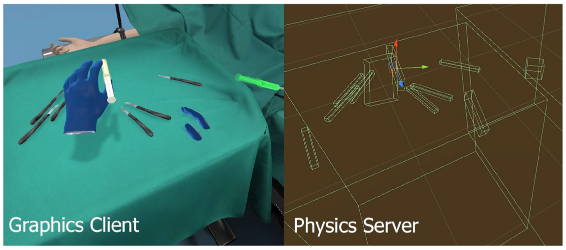

FIGURE 6. Physics are calculated only to the server while the client receives the rendered image.

attributes (e.g., mass, collider dimensions, etc.) from all host's game objects, which are subsequently created in the physics server as physics objects with the same parameters, in an easily transmittable and compressible format. The physics server acts on a completely passive nature since it retains no knowledge regarding the host's scene, the game loop, or the behaviors. The imposed intercalls are streams of transformations between the two services.

The dissected physics server runs inside a Windows VM on a linux host. Further containerization of this service is under consideration. The results of the dissected pipeline show a minor uptake of 0.03 ms on the latency from the new physics service, also producing a frame rate of 60 fps. The produced results have confirmed our plans of hosting extremely high-intensity physics computations in a separate edge service, and allow the physics server to serve multiple rendering services, collaborating in the same multiuser session.

VR SOFT BODY, SKINNED SIMULATION

In the real world, there exist certain deformable objects (e.g., soft or hard tissues) which are "naturally" deformed when external forces are applied on them. MAGES incorporates a framework that aims to achieve this, via the so-called *soft-body mesh deformation*. The idea behind this methodology is to create a layer of control points, called *particles*, on top of the mesh model that, when translated, would affect the vertices of a model in a weighted manner. Ultimately, the visual effect we aim for is similar to what would happen in reality if we pinched and pulled the material at that point, toward the displacement direction of the particle.

The particles are spawned and bound via springs to some initial positions on the mesh's surface, uniformly distributed via a Poisson sampling mechanism. Particles lying sufficiently close are "connected," forming the desired layer of control points. Furthermore, we assign vertices to particles that have a distance from them below a specific model-dependent threshold.

After this initial setup, every time a particle is displaced away from its initial position it will trigger several events. First, the particle's displacement will amount to a weighted displacement of all particles it is assigned to, with each vertex's displacement being inversely proportional to its distance from the particle. Furthermore, the particle's movement will affect connected particles by a fraction also inversely proportional to their in-between distance. As neighboring particles are displaced, they will, in turn, affect vertices in a diminishing extent. Finally, particles that are moved away from their initial position tend to return with a velocity that is proportional to the displacement, similar to a spring. With these fundamental rules applied to the model, we are able to simulate elasticity on the model's surface.

TABLE 1. Running times required to tear and cut various soft-body models: a liver and a heart. Results taken from Kamarianakis et al. 's work.[13]

Characteristics	Liver	Heart
Number of vertices	515	2527
Number of triangles	768	4968
Number of particles	191	179
Tear operation	Performance per tear segment	
Perform tear	0.4 ms	3 ms
Update particles	2.3 ms	7.3 ms
Total time	3.25 ms	11.19 ms
Output	140 fps	90 fps
Cut operation	Performance	
Intersection points	128	356
Total time	10.9 ms	17.2 ms
Output	90 fps	55 fps

HIGHLY REALISTIC PROGRESSIVE TEARING AND CUTTING

By exploiting the GA-based interpolation engine, we are able to simulate realistic unconstrained consecutive tears or cuts, on a soft-body model, similar to the ones performed in real life by a surgeon in the operating room (see Figure 7). Based on pure geometric operations on the surface mesh, we are able to perform

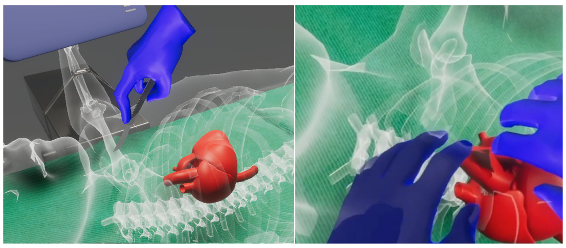

FIGURE 7. Interactive heart model featuring our deformable and tearing algorithms. The student performed an incision to observe the internal structures.

such actions and obtain real-time results in XR, even in low-spec devices, such as mobile VR HMDs.

Although Wu et al.[9] described diverse ways on how to cut a 3-D model, most of these methods are not suitable for VR,[20] since the necessary computations must be performed in real time, within a few ms to preserve user immersion. The latest developments[13] allow for complex operations, such as cutting or tearing on a rigged mesh model, to be run in real time, and are incorporated in MAGES. The significance of this framework lies in the fact that it overcomes current state-of-the-art limitations, where similar tears on a rigged 3-D model in VR are predefined via linear-blend skinning animations, in order to allow them to play back in real time.

Paired with the soft-bodies framework, our tearing and cutting algorithms allow the simulation of realistic, surgical-grade continuous tears, especially valuable in the context of medical VR training. Furthermore, our algorithms are based on simple geometric predicates on the rigged mesh, and therefore do not require specific model preprocessing. Ultimately, our framework allows the user to freely cut or tear in a consecutive way any 3D mesh model, under collaborative networked VE, reaching up to 140 fps, depending on scene complexity (see Table 1). Tears can now be simulated to "open," replicating the tissue behavior in real-life incisions, providing immersive visual results for soft-body materials, and of course, rigged models can be further reanimated and torn or cut again, enabling a number of complex surgical operations to be implemented in XR.

CONVOLUTIONAL NEURAL NETWORK AUTOMATIC ASSESSMENT

Despite the effort, only few systems involve procedures for assessing user progress inside the immersive environment, that either evaluate only trivial tasks or require a huge amount of time by manually being reviewed or assessed. On the other hand, the need for real-time automated evaluation of user's actions is constantly increasing. The state-of-the-art methods for similar tasks either require the development of complicated task-specific computer vision algorithms or support very simple tasks.[10]

MAGES proposes a deep-learning-based system that is able to assess, in real time, user actions within a VR training scenario. The method enables the rapid development of trained assessment functions, since it utilizes data augmentation to minimize the amount of labeled data (e.g., poor/mediocre/excellent performance) that need to be collected. Using this system, we are able to assess actions (e.g., tears and cuts) performed in VR medical operations by the trainees.

The developed scoring system evaluates user's actions, based on the trajectories of the used virtual tool. We utilize a supervised learning process that feeds score-labeled trajectories, randomly sampled, to a lightweight 15-layer CNN model, able to provide high-accuracy results (see Figure 8). The CNN consists of four sets of two convolutional, two ReLU, and a batch normalization layer, along with global average pooling, flattening and densing layers, which form the classifier. The model outputs the probabilities $p(i)$ of a trajectory belonging to each of the used classes. In Table 2, a comparison of the accuracy of different models is presented for various numbers of classes. In the case of 2 or 3 classes, all models perform correctly; however, when 6 classes are used, only our proposed CNN method achieves

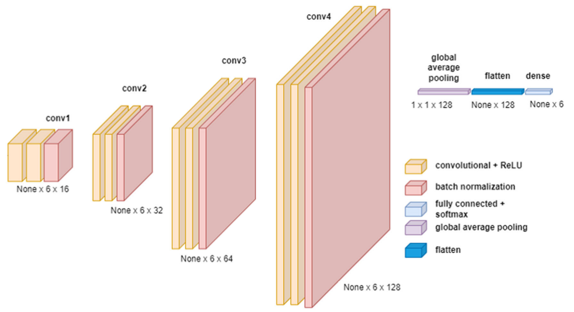

FIGURE 8. Layers of the proposed CNN model.

TABLE 2. Table comparing the accuracy of traditional ML techniques and the proposed CNN.

No. of classes	Logistic regression	KNN	SVM	CNN (ours)
2	100%	100%	100%	100%
3	100%	100%	100%	100%
6	39%	66%	83%	**95%**

TABLE 3. Measuring the FPS burden of a VR application due to the recorder feature. Results taken from Kamarianakis et al. 's work.[11]

Performance	Session without VR recording	Session with VR recording
Average FPS	89.56	85.13
Minimum FPS	76.56	68.78
Maximum FPS	93.29	92.57

SIDEBAR: MEDICAL VR TRAINING EXAMPLES BUILT WITH MAGES

In this section, we present honorable medical simulations built with MAGES SDK.

Total knee and hip arthroplasty: In collaboration with the University of Southern California and New York University, we created both orthopedic operations as a part of their curricula (see Figures 9, 10, and 11). We also conducted a clinical trial that proves skill transfer from virtual to the real world[S1] (3 months, three full-time developers, one full-time designer).

FIGURE 9. (Top) MAGES 4.0 collaborative total knee arthroplasty simulation with 25 trainees in the same VR operating room. (Bottom) The MAGES 4.0 swab insertion process from the COVID-19 nasopharyngeal swab test and personal protective equipment simulation.

FIGURE 10. MAGES 4.0 total knee arthroplasty simulation.

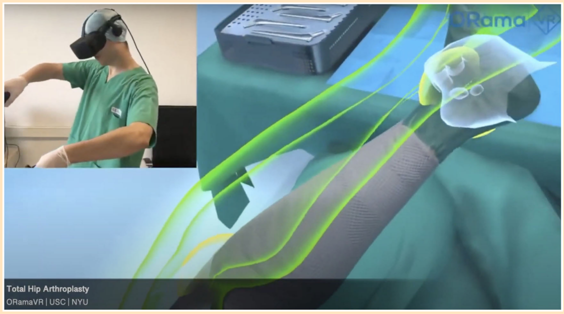

FIGURE 11. MAGES 4.0 total hip arthroplasty simulation.

Emergency trauma scenarios: This is a collection of emergency on-the-field simulations (extrication from car, first aid) and an intubation process (see Figures 12 and 16) as part of an online course for the University of Athens (1 month, two full-time developers, one full-time designer).

FIGURE 12. MAGES 4.0 emergency trauma simulation.

FIGURE 13. MAGES 4.0 social worker simulation.

FIGURE 14. MAGES 4.0 behavioral health simulation.

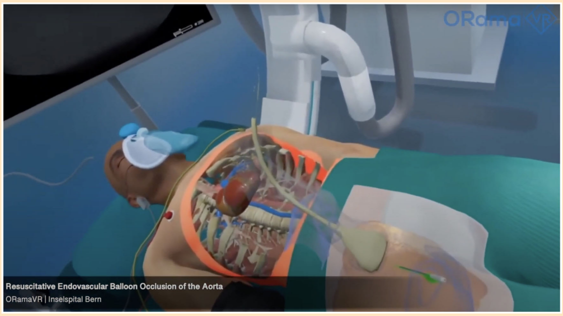

FIGURE 15. MAGES 4.0 REBOA simulation.

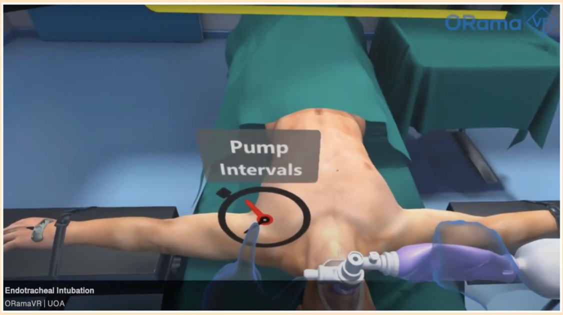

FIGURE 16. MAGES 4.0 intubation simulation.

STARS: Patient education and empowerment through knowledge. We developed this informative and stress relief application for patients in collaboration with ICS-FORTH and the Human Computer Interaction Lab[S2] (1.5 months, two full-time developers, one full-time designer).

VRADA: In collaboration with the Aristotle University of Thessaloniki, the University of Thessaly, and Biomechanic solutions, we created a VR bike simulation with cognitive questions that allows older people with mild cognitive impairment symptoms to simultaneously practice physical and cognitive skills on a dual task. We published a clinical trial with our results[S3] (2 months, one full-time developer, one full-time designer).

Social worker: A cognitive training simulation for soft skills in collaboration (see Figure 13) with the Fayetteville State University (14 days, one full-time developer, one part-time designer).

Behavioral health, chronic care assessment: Two cognitive training simulations for soft skills in collaboration (see Figure 14) with the Western Governors University (3 months, two full-time developers, one part-time designer).

COVID-19 PPE & swab testing: During the COVID-19 pandemic, we developed this application to educate medical personnel on the proper use of personal protective equipment and how to take COVID samples for testing (see Figure 9). It was performed in collaboration with Inselspital University Hospital of Bern and New York University. We conducted two clinical trials,[S4,S5] to explore the effectiveness of VR simulations versus traditional learning methods (3 months, three full-time developers, two full-time designers).

Anatomical viewer: Non Nocere developed an anatomical viewer for laminectomy and discectomy using MAGES SDK (see Figure 18).

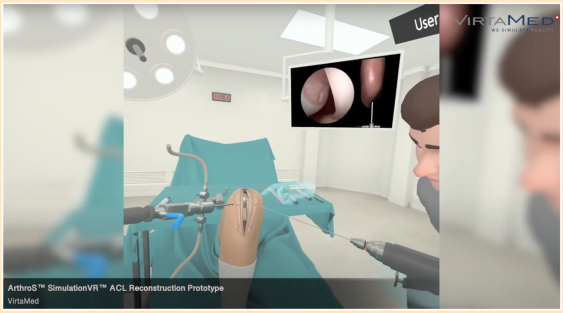

FIGURE 17. MAGES 4.0 arthroscopy simulation.

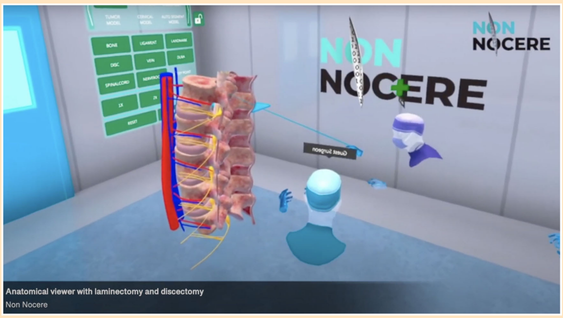

FIGURE 18. MAGES 4.0 anatomical viewer simulation.

REBOA: In collaboration with the Inselspital University Hospital of Bern, we developed a endovascular balloon occlusion of the Aorta simulation (see Figure 15)

(3 months, three full-time developers, one full-time designer).

Collaboration with VirtaMed AG.: VirtaMed AG is currently developing a Knee Arthroscopy simulation (see Figure 17) using MAGES SDK.[c]

IDS: In collaboration with the University of Geneva, University Hospital of Geneva, and MiraLAB Sarl, in the Swiss funded project Intelligent Digital Surgeon,[d] we are currently advancing our ML assessment feature. The main objective of the project is to identify and analyze the immersed trainee's behavioral model and provide personalized real-time feedback, assessment, and recommendations like a real surgical instructor. A deep learning model will be derived to identify the trainee's behavioral model, by recognizing and analyzing the trainee's hand/arm gestures, and to assist the feedback decision engine, by providing personalized assessment, real-time feedback, instructions, and recommendations.

PROFICIENCY: We also participate in the multipartner (the lead of the three clinical partners Kantonsspital St. Gallen, Centre Hospitalier Universitaire Vaudois, and Balgrist University Hospital, and VirtaMed AG, Microsoft Mixed Reality, AI Zurich Lab, and Atracsys LLC) Swiss funded PROFICIENCY[e] project that aims to define a fully novel, standardized, and proficiency-based surgical training curricula installed and demonstrated on two example surgical modalities, laparoscopy, and arthroscopy, while fully generalizable to other interventions.

[c]https://business.vive.com/uk/stories/solving-medical-training-challenges-present-and-future/
[d]https://innosuisse.miralab.com/
[e]https://surgicalproficiency.ch/

REFERENCES

S1. J. Hooper et al., "Virtual reality simulation facilitates resident training in total hip arthroplasty: A randomized controlled trial," *J. Arthroplasty*, vol. 34, no. 10, pp. 2278–2283, 2019, doi: 10.1016/j.arth.2019.04.002.

S2. H. Kondylakis et al., "A digital health intervention for stress and anxiety relief in perioperative care: Protocol for a feasibility randomized controlled trial," *JMIR Res. Protocols*, vol. 11, no. 11, 2022, Art. no. p.e38536.

S3. M. Hassandra et al., "A virtual reality app for physical and cognitive training of older people with mild cognitive impairment: Mixed methods feasibility study," *JMIR Serious Games*, vol. 9, no. 1, 2021, Art. no. e24170, doi: 10.2196/24170.

S4. T. Birrenbach et al., "Effectiveness and utility of virtual reality simulation as an educational tool for safe performance of COVID-19 diagnostics: Prospective, randomized pilot trial," *JMIR Serious Games*, vol. 9, no. 4, 2021, Art. no. e29586, doi: 10.2196/29586.

S5. P. Zikas et al., "Virtual reality medical training for COVID-19 swab testing and proper handling of personal protective equipment: Development and usability," *Frontiers Virtual Reality*, vol. 2, 2022, doi: 10.3389/frvir.2021.740197.

S6. P. Zikas et al., "COVID-19 VR strikes back: Innovative medical VR training," in *ACM SIGGRAPH 2021 Immersive Pavilion (SIGGRAPH '21)*. New York, NY, USA: ACM, 2021.

an acceptable performance. The model was trained with the VR recorder's exported transformations/trajectories data, that were obtained after recording and labeling multiple sessions.

The usefulness of this feature is especially highlighted in cases where we have a predefined operation, or exemplary task from a teacher, and students must be trained to replicate the teacher's movements precisely. Furthermore, by using transfer learning, these assessment functions can be reconfigured to support similar tasks, thus reducing even more the amount of training data.

Various underlying techniques were used to make this achievable. Data collection involved capturing transformation data (translation and rotation) of the active virtual tool (e.g., a scalpel) on distinct frames. The dataset was made uniform to amend for the fact that the execution of an action is user dependent. Data augmentation techniques were applied to increase the training data toward enhancing the training process of the 15-layer CNN. Since low training and inference times are preferred, the lightweight model used is able to provide high-accuracy scoring results. Incorporating the trained CNN in the MAGES allows assessing the user's actions with minimal performance overhead.

VR RECORDER FOR AUTOMATIC ASSESSMENT AND DEBRIEF

Recording and replaying a training session in VR can serve as an additional and powerful educational tool. The bibliography shows great interest on this matter with the majority of projects focusing on video recording and motion capturing.[10]

MAGES incorporates a different, highly accurate and lightweight solution to record and replay any collaborative training VR/AR session.[11] The underlying algorithm allows for the first time to record the session by capturing the transformation of objects (position, rotation) and user-driven events (interaction, decisions). This results in a highly accurate recording of the scenario that requires minimal storage space (approximately 1 MB per minute per user) and minimal performance overhead (see Table 3). In addition, we efficiently handle and record the user audio and synchronize it using a timestamp algorithm.

After recording the session, users can select the recordings and replay them. They can join in the same recorded session, not only as viewers of a 3-D video but with the ability to navigate within the scene and relive the session from any perspective in space, at any given time. In this way, they may pay attention to details that they missed when they initially "played" the scenario.

MEDICAL METAVERSE CASE STUDIES

We can create a complete simulation at a fraction of the time and cost with respect to existing solutions. To be more precise, we were able to build a training simulation for soft skills in collaboration with the Fayetteville State University within 14 days, employing one full-time developer and one part-time designer. Using other frameworks, the respective work would require 120 days with the same human resources, to be completed. This corresponds to an eight times faster and eight times more cost-efficient authoring process, based on current freelance rates for one 3-D designer and one 3-D developer in the USA today. For more simulation applications built with MAGES, along with the required amount of resources, the reader may refer to the "Medical VR Training Examples Built With MAGES" sidebar.

CONCLUSIONS AND FUTURE WORK

In this work, we presented MAGES 4.0, a novel development kit that allows rapid creation of any collaborative medical VR/AR simulation. The abovementioned novelties rapidly accelerate content creation while maintaining a stable, multiplaftorm authoring tool for the upcoming medical metaverse.

In the near future, we aim to improve the developer and user experience in MAGES. Our goal is to reach a point where a novice developer can rapidly uptake and create with all existing actions, cloud, and analytics in a single day. In addition, experienced developers should also be able to extend the codebase (actions, prototypes, mechanics) with ease at a fraction of time with minimal effort. Furthermore, we consider to support the universal scene description, an open metaverse file format that empowers the collaboration in 3-D virtual worlds. Our hope is that this work provides a solid contribution on how to accelerate the world's transition to medical VR training and the proliferation of the medical metaverse.

ACKNOWLEDGMENTS

The project was partially funded by the European Union's Horizon 2020 research and innovation programme under grant agreements 871793 (ACCORDION), 101016509 (CHARITY), and 101016521 (5G-EPICENTRE) and from the Innosuisse—Swiss Innovation Agency No. 100.133 IP-ICT (Intelligent Digital Surgeon).

REFERENCES

1. H. Coelho, P. Monteiro, G. Gonçalves, M. Melo, and M. Bessa, "Authoring tools for virtual reality experiences: A systematic review," *Multimedia Tools Appl.*, vol. 81, no. 19, pp. 28037–28060, 2022.
2. M. Nebeling et al., "XRDirector: A role-based collaborative immersive authoring system," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2020, pp. 1–12.
3. G. Papagiannakis et al., "MAGES 3.0: Tying the knot of medical VR," in *Proc. ACM SIGGRAPH Immersive Pavilion*, 2020, pp. 1–2.
4. F. Cassola, M. Pinto, D. Mendes, L. Morgado, A. Coelho, and H. Paredes, "A novel tool for immersive authoring of experiential learning in virtual reality," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces Abstr. Workshops*, 2021, pp. 44–49.
5. L. Zhang and S. Oney, "FlowMatic: An immersive authoring tool for creating interactive scenes in virtual reality," in *Proc. 33rd Annu. ACM Symp. User Interface Softw. Technol.*, 2020, pp. 342–353.
6. P. Zikas et al., "Immersive visual scripting based on VR software design patterns for experiential training," *Vis. Comput.*, vol. 36, pp. 1965–1977, 2020, doi: 10.1007/s00371-020-01919-0.
7. M. Kamarianakis, I. Chrysovergis, N. Lydatakis, M. Kentros, and G. Papagiannakis, "Less is more: Efficient networked VR transformation handling using geometric algebra," *Adv. Appl. Clifford Algebras*, vol. 33, no. 1, pp. 1–34, 2023.

8. R. Nystrom, *Game Programming Patterns*. 3rd ed., Rossendale, U.K.: Genever Benning, 2014.
9. J. Wu, C. Dick, and R. Westermann, "Efficient collision detection for composite finite element simulation of cuts in deformable bodies," *Vis. Comput.*, vol. 29, no. 6, pp. 739–749, 2013.
10. A. Zia et al., "Automated video-based assessment of surgical skills for training and evaluation in medical schools," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 11, no. 9, pp. 1623–1636, 2016.
11. M. Kamarianakis, I. Chrysovergis, M. Kentros, and G. Papagiannakis, "Recording and replaying psychomotor user actions in VR," in *Proc. ACM SIGGRAPH Posters*, 2022, pp. 1–2, doi: 10.1145/3532719.3543253.
12. D. Abeysinghe, C. Chua, and W. Huang, "Using XR to support collaborative learning in health," in *Proc. 23rd Int. Conf. Inf. Vis.–Part II*, 2019, pp. 92–95.
13. M. Kamarianakis, A. Protopsaltis, D. Angelis, M. Tamiolakis, and G. Papagiannakis, "Progressive tearing and cutting of soft-bodies in high-performance virtual reality," in *Proc. ICAT-EGVE Int. Conf. Artif. Real. Telexistence Eurographics Symp. Virtual Environ.*, 2022, pp. 45–53, doi: 10.2312/egve.20221275.
14. R. K. Reznick and H. MacRae, "Teaching surgical skills–changes in the wind," *New England J. Med.*, vol. 355, no. 25, pp. 2664–2669, 2006.
15. G. Bansal, K. Rajgopal, V. Chamola, Z. Xiong, and D. Niyato, "Healthcare in metaverse: A survey on current metaverse applications in healthcare," *IEEE Access*, vol. 10, pp. 119914–119946, 2022.
16. J. Schild, D. Lerner, S. Misztal, and T. Luiz, "EPICSAVE–Enhancing vocational training for paramedics with multi-user virtual reality," in *Proc. IEEE 6th Int. Conf. Serious Games Appl. Health*, 2018, pp. 1–8.
17. J. Schild, C. Elsenbast, and G. Carbonell, "ViTAWiN-developing multiprofessional medical emergency training with mixed reality," in *Proc. IEEE 9th Int. Conf. Serious Games Appl. Health*, 2021, pp. 1–9.
18. S. Kloiber et al., "Immersive analysis of user motion in VR applications," *Vis. Comput.*, vol. 36, no. 10, pp. 1937–1949,2020.
19. S. Oprea, P. Martinez-Gonzalez, A. Garcia-Garcia, J.A. Castro-Vargas, S. Orts-Escolano, and J. Garcia-Rodriguez, "A visually realistic grasping system for object manipulation and interaction in virtual reality environments," *Comput. Graph.*, vol. 83, pp. 77–86, 2019.
20. I. Berndt, R. Torchelsen, and A. Maciel, "Efficient surgical cutting with position-based dynamics," *IEEE Comput. Graph. Appl.*, vol. 37, no. 3, pp. 24–31, May/Jun. 2017.

PAUL ZIKAS is vice president of engineering at ORamaVR SA, 1228, Geneva, Switzerland. Currently he is developing his Ph.D. research topics at the University of Geneva, 1205, Geneva, Switzerland. Zikas received his M.Sc. degree in computer science from the University of Crete, Iraklio, Greece. He is the corresponding author of this article. Contact him at paul.zikas@oramavr.com.

ANTONIS PROTOPSALTIS is a lead research scientist at ORamaVR SA, 1228, Geneva, Switzerland. He is also a special teaching fellow with the University of Western Macedonia, 50100, Kozani, Greece. Protopsaltis received his Ph.D. degree in computer science from the University of Ioannina, Ioannina, Greece. Contact him at antonis.protopsaltis@oramavr.com.

NICK LYDATAKIS is head of platform at ORamaVR SA, 1228, Geneva, Switzerland. Lydatakis received his M.Sc. degree in computer graphics from the Computer Science Department, University of Crete. Contact him at nick.lydatakis@oramavr.com.

MIKE KENTROS is lead SDK developer and ML at ORamaVR SA, 1228, Geneva, Switzerland. Currently, he is a Ph.D. candidate with the University of Crete, 70013, Iraklio, Greece. Kentros received his M.Sc. degree in computer science from the University of Crete. Contact him at mike.kentros@oramavr.com.

STRATOS GERONIKOLAKIS is lead SDK developer, responsible for multiplatform support and products distribution, with ORamaVR SA, 1228, Geneva, Switzerland. Geronikolakis received his M.Sc. degree in computer science from the University of Crete, specializing in computer graphics. Contact him at stratos.geronikolakis@oramavr.com.

STEVE KATEROS is vice president of product management at ORamaVR SA, 1228, Geneva, Switzerland. He was a researcher in ICS-FORTH. Kateros received his M.Sc. degree in computer science from the University of Crete, specializing in computer graphics and virtual reality. Contact him at steve.kateros@oramavr.com.

MANOS KAMARIANAKIS is an SDK R&D developer and ML at ORamaVR SA, 1228, Geneva, Switzerland. Kamarianakis received his Ph.D. degree in applied mathematics from the University of Crete. Contact him at manos.kamarianakis@oramavr.com.

GIANNIS EVANGELOU is a lead VR developer and level designer at ORamaVR SA, 1228, Geneva, Switzerland. Currently, he is working toward his M.Sc. degree in information systems and human–computer interaction. Evangelou received his B.Sc. degree in computer science from the University of Crete, specializing in game design and computer graphics. Contact him at giannis.evangelou@oramavr.com.

ACHILLEAS FILIPPIDIS is a content and SDK developer with ORamaVR SA, 1228, Geneva, Switzerland. Filippidis received his B.Sc. degree in computer science from the University of Crete, specialized in computer graphics, and continues his academic path with an M.Sc. Contact him at achilles.filippidis@oramavr.com.

ELENI GRIGORIOU is a product manager and UI design at ORamaVR SA, 1228, Geneva, Switzerland. She is working toward her M.Sc. degree in computer graphics. Grigoriou received her B.Sc. degree in computer science from the University of Crete, specialized in big-data interaction in VR. Contact her at eleni.grigoriou@oramavr.com.

DIMITRIS ANGELIS is a machine learning developer at ORamaVR SA, 1228, Geneva, Switzerland. Angelis received his B.Sc. degree in computer science from the University of Crete, specialized in classification algorithms. Contact him at dimitris.angelis@oramavr.com.

MICHAIL TAMIOLAKIS is a content and soft-body developer at ORamaVR SA, 1228, Geneva, Switzerland. Tamiolakis received his B.Sc. degree in computer science from the University of Crete, specialized in soft-body simulations. Contact him at michalis.tamiolakis@oramavr.com.

MICHAEL DODIS is a networking and Unreal Engine developer at ORamaVR SA, 1228, Geneva, Switzerland. Contact him at michael.dodis@oramavr.com.

GEORGE KOKIADIS is an MR developer at ORamaVR SA, 1228, Geneva, Switzerland. Kokiadis received his M.Sc. degree in multimedia technology from the University of Crete. Contact him at george.kokiadis@oramavr.com.

JOHN PETROPOULOS is a VR content developer with ORamaVR SA, 1228, Geneva, Switzerland. He is working toward his M.Sc. degree in computer graphics. Petropoulos received his B.Sc. degree in computer science. Contact him at john.petropoulos@oramavr.com.

MARIA PATERAKI is an R&D executive at ORamaVR SA, 1228, Geneva, Switzerland. She is also an assistant professor in photogrammetry at the National Technical University of Athens, and an affiliated researcher of the CVRL at ICS-FORTH. Contact her at maria.pateraki@oramavr.com.

GEORGE PAPAGIANNAKIS is co-founder and CEO/CTO of ORamaVR. His academic credentials include serving as professor of computer graphics at the Computer Science Department, University of Crete, 70013, Iraklio, Greece, and as affiliated research fellow at the HCI lab at ICS-FORTH, where he leads the CG Group and as visiting professor of CS at the University of Geneva, Geneva, Switzerland. Contact him at george.papagiannakis@oramavr.com.