

Received 21 July 2023, accepted 3 September 2023, date of publication 13 September 2023, date of current version 20 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3314793

## RESEARCH ARTICLE

# Development of a Digital Twin Pipeline for Interactive Scientific Simulation and Mixed Reality Visualization

JUNG-HUN SHIN<sup>1</sup>, (Member, IEEE), SANG-JIN PARK<sup>1</sup>, (Member, IEEE),  
MIN-AH KIM<sup>1</sup>, (Member, IEEE), MYEONG-JIN LEE<sup>2</sup>, (Member, IEEE),  
SEUNG-CHAN LIM<sup>2</sup>, (Member, IEEE), AND KUM-WON CHO<sup>3</sup>, (Member, IEEE)

<sup>1</sup>Korea Institute of Science and Technology Information (KISTI), Daejeon 34141, Republic of Korea

<sup>2</sup>Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34141, Republic of Korea

<sup>3</sup>Kumoh National Institute of Technology (KIT), Gumi-si 39177, Republic of Korea

Corresponding authors: Jung-Hun Shin (shandy77@kisti.re.kr) and Kum-Won Cho (kwcho@kumoh.ac.kr)

This work was supported by the National Research Council of Science and Technology (NST) grant by the Korean Government (MSIT) under Grant CRC21011.

**ABSTRACT** An interactive scientific simulation and visualization pipeline in mixed reality was developed to explore how scientific simulation could proceed in coming workplace. The main technical concern is fast scientific simulator and visualizer in mixed reality with hand gesture. Framework for scientific simulation and visualization was implemented on the server side based on deep learning based fast simulation model and open source visualization toolkit, VTK, respectively. Airfoil aerodynamics was selected as a target scientific simulation task since it is a representative and critical problem in scientific fluid simulation determining the whole aero-product's performance and shows highly nonlinear physics even by small shape deviation and so high computational cost. Therefore, the fast simulator was firstly developed by carefully developing convolutional neural network-based model, using U-net, which learned corresponding high accuracy simulation dataset. All scenes which are supposed to be seen on the mobile device side were authored in a widely used content development tool called Unity3D with a mixed reality toolkit, MRTK for Microsoft HoloLens 2. User interface and user experience were created based on a scenario of a drone rotor which demonstrates running and analyzing the simulation results using designed interaction gestures. Finally, several discussions were derived through the implemented digital twin pipeline. It was concluded, with an adequately implemented mixed reality tools, that scientific researchers or industrial engineers will be helped not only in individual analysis works due to the delicacy of hands and fingers and real time interaction in mixed reality, but also, as a future study, in cooperative analysis works.

**INDEX TERMS** Computational fluid dynamics, computer simulation, convolutional neural network, digital twin, extended reality, mixed reality.

## I. INTRODUCTION

Extended reality (XR) is a comprehensive term including whole spectrum of virtual reality (VR), augmented reality (AR), and their combination called mixed reality (MR). XR technology is widely used in computer graphics related

The associate editor coordinating the review of this manuscript and approving it for publication was Stavros Souravlas.

fields such as game, movie, advertisement, modern art et cetera. In spite of the popularity of media and game in these days, there have also been scientific or engineering studies associated with. Huzaifa et al. [1] developed an XR testbed in the University of Illinois at Champaign which is opened to application fields in the manner of consortium. Among the various application fields, medical and architectural fields are notably active. Peter et al. [2] explained

TABLE 1. Nomenclature.

Symbols or Abbreviations	Meanings
$\rho$	Density
$p$	Pressure
$t, x, y$	Time and spatial axis
$u, v$	Velocity on x, y-axis
2D or 3D	Two-dimensional or three-dimensional
AI	Artificial intelligence
AR	Augmented reality
CFD	Computational fluid dynamics
CNN	Convolution neural network
ELU	Exponential linear unit
I/O	Input and output
LiDAR	Light detection and ranging
MR	Mixed reality
MRO	Maintenance, repair, and operation
MRTK	Mixed reality toolkit
PDE	Partial differential equation

a thorough treatment of XR in medical practice, and Curran et al. [3] explained that XR technique got noticed particularly in the medical training due to its simulation-based learning modality. Alizadehsalehi et al. [4] provided a roadmap for the implementation of XR technologies in the architecture, engineering, and construction industry. In practical aspect, Carasco and Chen [5] discussed improving architectural design comprehension efficiency using MR technology. Besides the active fields, Çöltekin et al. [6] identified and documented research challenges in geo-spatial science if using XR. Doolani et al. [7] discussed application cases and future barriers in the manufacturing training field. Lee and Yoo [8] focused on collaboration function of XR in real world by demonstrating micro contents implemented in web-browser.

On the other hand, even among many aforementioned application fields, XR content has not yet employed engineering simulation workflow. This study is particularly concerned about computational fluid dynamics (CFD) simulation in mixed reality environment. Milgram and Kishino [9] originally created the term ‘mixed reality’ to clarify a spectrum between VR and AR. It has pros of an immersive environment that matches reality, suitable for any simulations. Solmaz et al. [10], Venn et al. [11], and Lin et al. [12] conducted their XR-CFD studies just in visualization aspects, but not in interactive simulation aspect due to its need of separated specialized authoring tools as well as high computational and network costs. Especially in the aspect of computational modeling technique, several loose approximation models proposed by Stam [13], Nguyen [14], Bridson [15] et cetera have been successfully used for interactivity and efficiency. They just employed inviscid and incompressible fluid model. Besides, Data driven simulation techniques have huge potential since accuracy can be increased by learning high accuracy simulation dataset, but there have been still few data driven studies focusing on accuracy. In CFD field, lots

of researcher and engineers have used principal component analysis as model order reduction technique [16]. Recently, Ladický et al. [17], Tompson et al. [18] and Kim et al. [19] recently proposed data driven fast simulation methods based on novel machine learning and deep learning.

The application field in this study, CFD simulation, can be normally seen in aerodynamic of aircraft. The left side of Figure 1 showed that a six-rotor drone and air flows are augmented in an office room shown by an MR device, and the right side demonstrates that an extracted wing section called airfoil which is very critical in the wing performance produces aerodynamic lifting and drag forces. Although airfoil shape design is a classical subject in engineering and manufacturing, it is still being studied by many researchers such as Grey et al. [20] and Wang et al. [21]. Wing design analysis is generally accompanied by CFD simulation around the airfoils belonging to the given wing. Blazek [22] described full numerical implementation of Reynolds averaged Navier Stokes equations for compressible and turbulent flows as well as mesh generation techniques of airfoils. Sa et al. [23] proposed accurate numerical models for simulating transition phenomena which occur in wing surface and showed the accuracy improvement. The simulation dataset used in this study were accumulated and curated for deep learning by using flow simulator developed by Sa et al. [23]. Shin and Sa [24] published a dataset of structural CFD

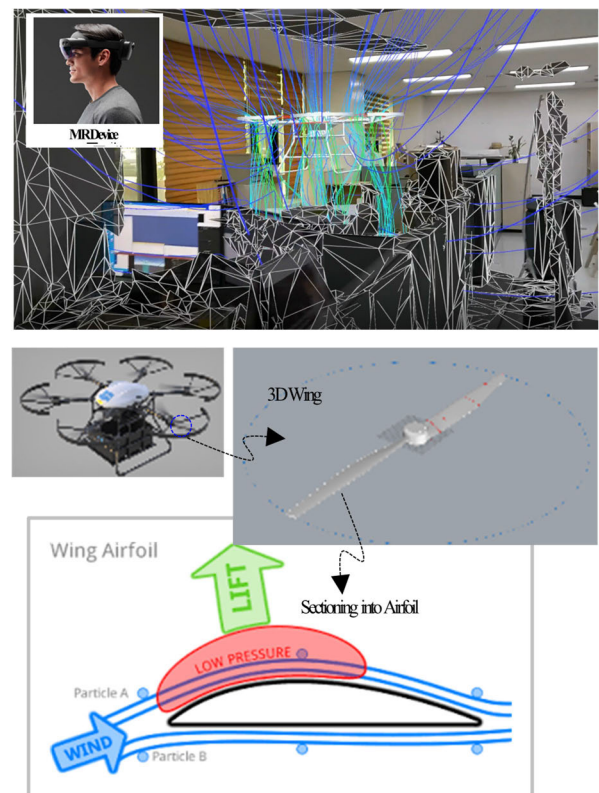


FIGURE 1. Visualization of airflow around a drone (upper) and aero-physics in its wing section (bottom).

mesh of airfoil for machine learning and deep learning algorithms. Shin et al. [25] surveyed whole spectrum of principal component analysis (or proportional orthogonality decomposition), fully-connected neural network, and convolutional neural network (CNN) targeting to predict flows around airfoils. In this study, the authors selected CNN-based U-net algorithm for simulation acceleration. Although there are several researches [26], [27] which used U-net algorithm, the U-net implementation in this study provided an inference model which can predict flow fields in accordance with airfoil shapes as well as near the airfoil boundaries very accurately.

Recently, demands for industrial digital twins have been recently raised. Wright and Davidson [28] discussed and clarified the relationship between digital twin and computational model. Tao et al. [29] searched for the computational tool candidates for digital twin modeling. As an extension of this trend, Therefore, the purpose of this study is to develop CFD simulation-based pipeline which will be tested in the MR environment which consists of an edge server and an MR device. The device is selected to be Microsoft HoloLens 2. Since MR is supposed to combine the best aspects of VR and AR, some interactions between reality and virtuality should be created in MR. Therefore, the developed MR pipeline in this study will consist of meaningful user interactions in a content scenario as well as accelerated CFD model and its visualization in a mobile device. The content scenario which will be explained in the section IV-B is related to the design analysis practice of industrial wing section. The developed MR pipeline is supposed to enable us to run CFD mesher and flow simulator and analyze the result. We chose hand gestures to firstly trigger the mesher and flow simulator in the edge server and query vertex scalar values later. Therefore, in this pipeline, the CFD tools should be installed in an edge server and the results data be rendered in mobile devices. In addition, all vertex query interactions are triggered in the device, but the vertex and its value are respectively searched and extracted in the edge server. This pipeline is expected to facilitate the airfoil shape design practice inside MR environment. The authoring platform which was used in this study is Unity3D [30] with MRTK [31].

## II. FRAMEWORK

This chapter describes several essential layers of framework. Figure 2 represents a pipeline for performing simulations in MR environment, visualizing the results, and querying them through gestures. This pipeline consists of mobile client-side and server-side components. Data exchange is carried out through RestAPI, and it can be designed to support remote or multi-user functionality.

Mobile client side provides visualization of simulation results in a mobile environment and the ability to review them through gesture queries. It consists of the three layers: postprocess, rendering, gesture layer. The postprocess layer is a configuration layer for simulation, including object selection, object cross-sectional polygon creation, and simulation parameter setting functions. Cross-sectional polygons

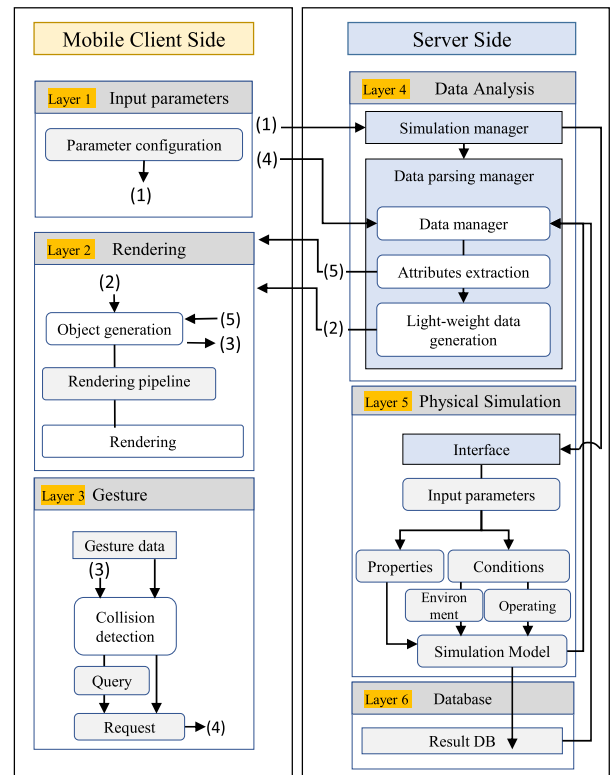


FIGURE 2. Framework of scientific simulation, visualization, and gestural interaction.

and simulation parameters generated in this layer are sent to the server side for simulation requests (1). The rendering layer includes data parsing and visualization object creation functions. It performs the visualization of simulation results (2) and gesture query results (5) through the rendering pipeline. The gesture query layer includes collision detection, simulation request, and vertex exploration query functions. This layer detects collisions between objects (3) and user gestures, and sends the corresponding vertices of the collision points to the Server (4). The mobile client side is implemented based on various authoring SW platform such as Unity3D or Unreal.

The server side consists of the data analysis layer, scientific simulation model layer, and database layer. Within the data analysis layer, the simulation manager serves as an interface for initiating simulations based on mobile client requests or retrieving result data stored in the database layer. The data manager's role is to analyze simulation results into a structure understandable in the proposed processes. Additionally, during the process of attribute extraction, elements (streamline, particles, et cetera) are extracted based on client requests, and the resulting data is reconstructed using REST API request/response methodology through light-weight data generation (see section IV-A). The scientific simulation model can be many schemes such as various fidelities of numerical analysis or a form of reduced order model (ROM). This study focused on CFD simulation which is accelerated using deep learning-based ROM. Since the acceleration is

achieved in order to interact with MR users through several gestures, we would also call the ‘accelerated’ as ‘interactive’. The details of the deep learning model layer are described in the following chapter. Finally, in the database layer, the storage and management of simulation results can be carried out for reuse and sharing. Users can utilize this layer to upload external data for storage or save simulations they have conducted themselves, enabling them to reuse and share the data at any time.

### III. FAST SCIENTIFIC SIMULATOR

Computational fluid dynamics has been widely used for product shape design in the industry as well as for scientific discovery. It normally requires high computing cost and so performed in stationary environments such as cluster and client architecture. Now it is an era of bigdata and artificial intelligence (AI) by development of great strides of infrastructure based on hardware and software. Reduced order model is a simulation acceleration method of recent great interest due to the AI infrastructure to overcome technological limit in scientific fluid simulation. This chapter describes how interactive fluid simulation model was developed.

#### A. SIMULATION DATASET PRODUCTION

As training dataset, 1,550 geometries were crawled from the UIUC airfoil data repository [32]. These dotted shape data are composed of XY coordinates along the boundary lines. Since how to handle geometry and generate CFD mesh for boundary flow simulation of airfoil were well established in the XFOIL community [33], the authors employed one of them, structure C-type mesh so that the simulation mesh is generated automatically when the airfoil geometries data is given as input. Indeed, this process is not so simple since one must regulate and normalize the dotted geometry data. As shown in Figure 3, firstly, the geometry data is fitted and formulated into cubic spline curve to find its real leading and trailing edges, to control the number and distribution of dots, and to regulate and normalize the data so that the length of  $x$  coordinate becomes one. This curve fitting technique could lead to other advantages to calculate airfoil parameters such as camber and thickness. Next, mesh for CFD simulation need to be generated. In this study, the meshing tool was developed selecting state-of-art partial differential equation-based mesh generation scheme (refer to Appendix A). All meshes had been generated so that the mesh topologies were identical: C-type, structured curved mesh, plot3d format, and size  $689 \times 145$ . The final stage is that a flow simulator for aerodynamic simulation produce flow data of velocity vector, pressure, density, et cetera. These ones are what will be learned to establish our reduced order model of CFD simulation. Using the ROM, the simulation can be done interactively in the MR environment.

Comparing the existing widely-used fluid simulations in the XR area, the fluid simulation of this study is different in that it employed mesh-based accurate physics model since the physics considered viscosity, compressibility and turbulence

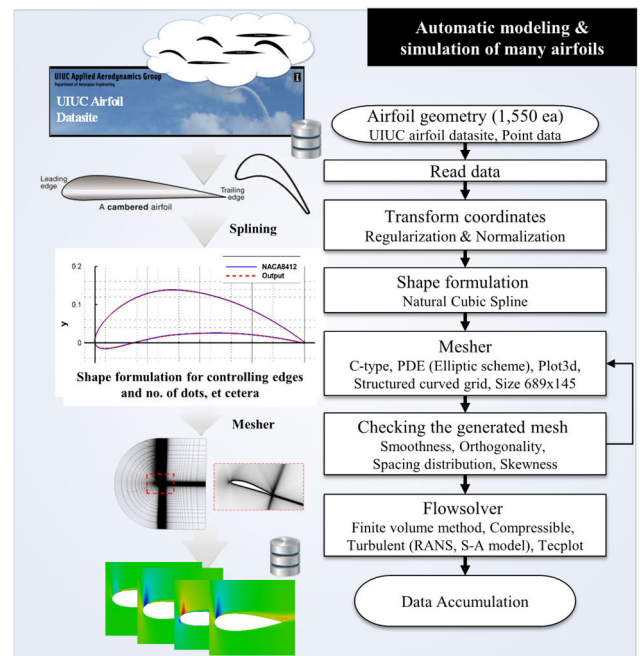


FIGURE 3. Dataset production flow of CFD simulation.

models. Numerically speaking, the physical model employed non-dimensional Reynolds Averaged Navier-Stokes equations (RANS). Now it can be said that RANS is a standard in the industry so that the tool can nearly simulate real fluid physics as a reliable design tool. The fluid flow simulator (normally called flowsolver in CFD domain) that the authors used is KFLOW [23].

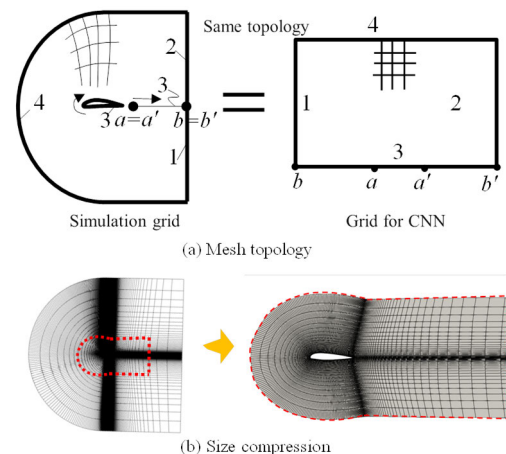


FIGURE 4. Spatial mesh data modeling for CNN-based deep learning.

#### B. DATASET CONSIDERATIONS FOR DEEP LEARNING

Here all mesh data were constrained as well as curated for the deep learning training. The topology was already determined to be identical in all mesh data as rectangular since our deep learning algorithm was already decided to be U-net. The mesh vertices must be able to be transformed into ones of a rectangular mesh as shown in Figure 4(a). By this way, all simulations of airfoils have the same mesh topology which



can be trained in CNN which can learn only structured mesh data such as single photo image. This part is very different and unique comparing with normal CNN work. One will find that the input of the deep learning model is the coordinates of all mesh points and Maintaining simulation mesh guarantees the accuracy in the next section.

The second consideration is that the training and inferring performance will be better when the data size becomes small. This indicate that compressed data lead to small number of deep learning model parameters. Fortunately, CFD for airfoils is normally interested in the boundary areas around airfoils and flow wake area behind the tail of airfoil. Therefore, as shown in Figure 4(b), it can be concluded that the mesh should be cut so that it contains the interesting areas.

**C. FAST SIMULATION MODEL**

Autoencoder, a representative method applying CNN to image reproduction, is an AI model possible of multiple application by giving asymmetric variation to the network in addition to having simple functions such as reproducing and restoring images. U-Net is a kind of Autoencoder network which equalizes the shapes of two corresponding layers which lies symmetrically between convolution and deconvolution processes (refer to Appendix B). It is widely used due to its high accuracy. Figure 5 shows such U-net deep learning process for CFD data. Correlation between input and output variables in CFD simulation can be learned by appropriately putting variables of interest in input and output which have the same dimension. As mentioned earlier in the previous section, the methodology is different from typical CNN implementation in that our U-net model uses original mesh as it was by putting the coordinates as input. This network model can produce output by perturbing the encoded latent vector lying in the middle of the network. In other words, one can develop such a conditional or parametric prediction model by adding concerned variable like scalar values of time or flow conditions to the latent vector. One can equalize the vector size by putting fully-connected layers before or after the latent vector. There are several frameworks which ones can implement various CNN networks readily such as Tensorflow-Keras [34] and Pytorch [35]. This study

**TABLE 2. Specifications of implemented U-net model.**

item	Value	Note
Input layer size	328 x 40 x 2	( $x, y$ )
Opout layer size	328 x 40 x 4	( $\rho, u, v, p$ )
No. of convolutional layers	3	= No. deconvolutional layers
Kernel size	3 x 3	
Stride	1	
Pooling	Maximum	
Padding	No	
Batch normalization	Yes	
Loss function	MSE	
Learning rate	10-3	
Batch size	32	
Activation function	ELU	
No. of total training	10,000	
Latent vector size	64	

used Tensorflow-Keras framework. The specifications of the U-net implementation are written in Table 2.

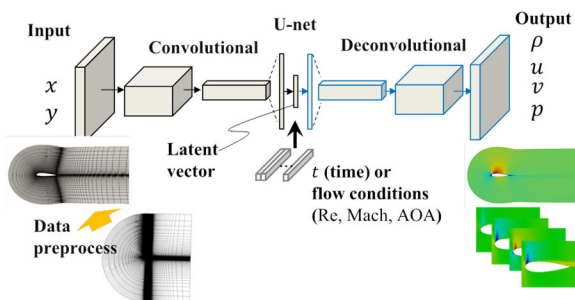
As shown in Figure 6, the simulation result is accurate with high-accuracy RANS simulation due to its appropriate implementation with surface-fitted mesh strategy explained in the section III-B as well as high performance U-net algorithm. The loss function was put to be the summation of mean square errors in whole computational area and along the airfoil boundary. It should be noted that single image training, not using physics-informed way, was enough to simulate CFD in such normal flow conditions without any stall and shock. It took about 30 hours for the training performed in NVIDIA Quadro RTX 6000. The inference time is less than 0.2 seconds comparing that full RANS simulation time is about 5 minutes per one flow condition.

**IV. IMPLEMENTATIONS**

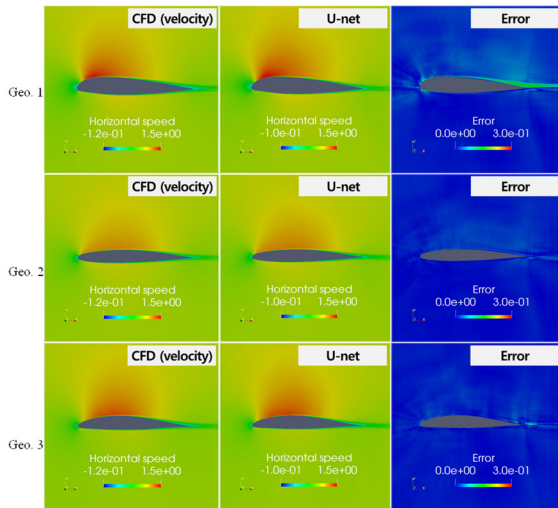
In this chapter, the authors describe a gesture-based query approach for effectively reviewing simulation results in an MR environment. To achieve this, the scientific visualization framework was developed as roughly described in Chapter 2 that integrates the reduced order model based on deep learning technique (called U-net) as described in Chapter 3. developed.

**A. PIPELINE**

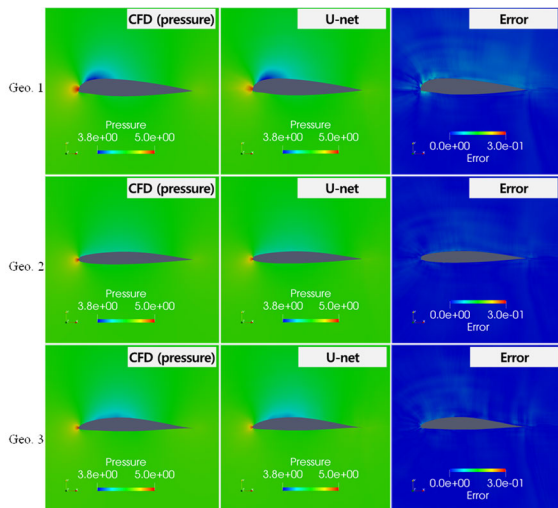
Figure 7 provides a detailed diagram of the pipeline for the gesture-based query approach in the Mobile Client Side of Figure 2 and a detailed interface with the Server Side. The Local Server which is the same as the Server Side of Figure 2 provides simulation analysis results based on client requests. This pipeline enables visualization of simulation results in the MR environment and allows for their query using hand gestures. The content development framework for mixed reality was selected as Unity3D, so the simulation and scientific visualization server was set to communicate with compiled mobile application using C# script. Both ROM simulator and visualization toolkit (VTK) framework were deployed in the local server. The MR content scenario proceeds with two types of hand gestures: grab and touch, which



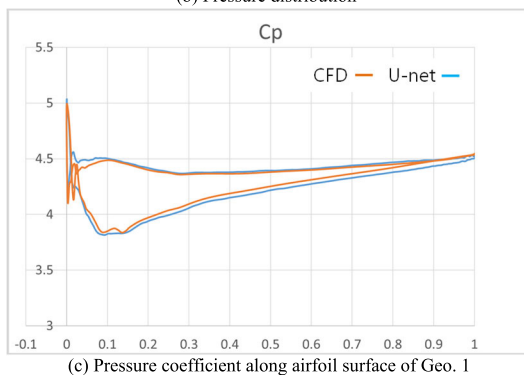
**FIGURE 5. Deep learning architecture of U-net with latent variables.**



(a) Horizontal velocity distribution



(b) Pressure distribution



(c) Pressure coefficient along airfoil surface of Geo. 1

FIGURE 6. Comparison of CFD model versus reduced order model.

are implemented using MRTK framework. MRTK is provided as Unity3D add-on package.

In this study, the authors have considered the characteristics of data exchange forms and have utilized the REST API request/response methodology to facilitate efficient responses to requests. To achieve this, the MR Environment incorporates the WebRequest function provided by Unity within the request syntax, and the local server is equipped

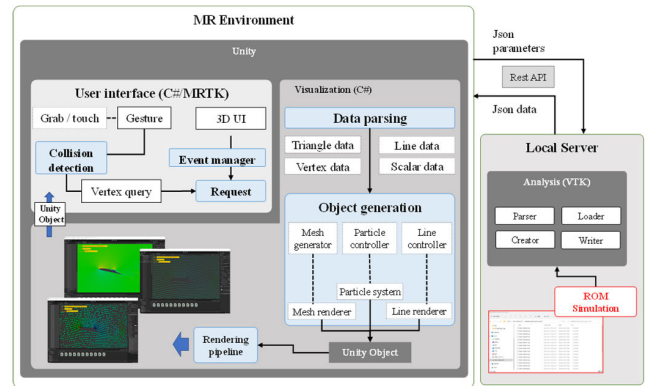


FIGURE 7. Implemented MR pipeline.

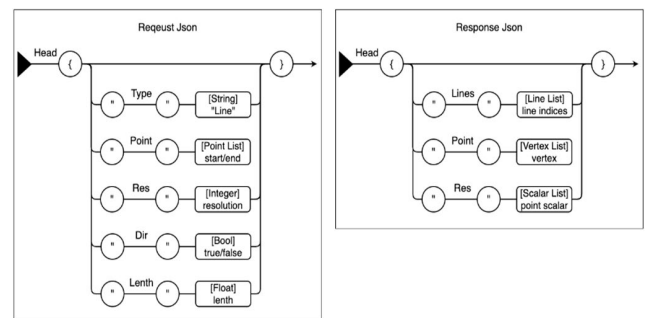


FIGURE 8. Structure of request and response in the form of REST API.

with Flask framework [36], enabling it to swiftly respond to WebRequests. This setup allows the Local Server to promptly check and respond to requests from the MR Environment during runtime. The data structure utilized for these requests and responses adopts the JSON format. Figure 8 depicts the structure of requests and responses on the REST API for the analysis of simulation results. Additionally, to support interactive review of simulation results in the MR environment, user hand gesture interaction methods were derived based on Hololens 2, which is widely used in the industry, and utilized MRTK for effective hand gesture implementation. MRTK is a tool that allows for the implementation of various interactions in the MR environment, providing several hand gesture methods such as pointer and touch and grab pointer (named point, touch or grab). In this study, query events of simulation results were implemented using the ray, touch, and grab pointer gesture methods. Figure 9 explains the process for handling touch and grab pointer gestures and querying results for vertices of simulation results.

## B. CONTENT SCENARIO

A process of the MR simulation scenario was developed as shown in Figure 10(a). 3D drone model was created for overall object handling simulation, and wings are especially designed by the procedures in OpenVSP which is a powerful aircraft design tool [37]. The tool enables ones to extract point data of airfoils in any sectional positions in a wing. Therefore, in this content scenario, it was assumed that ones

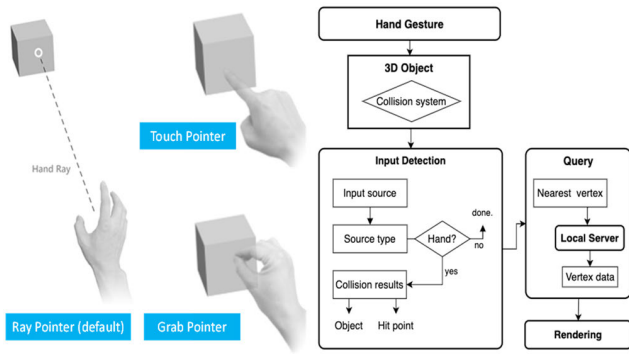
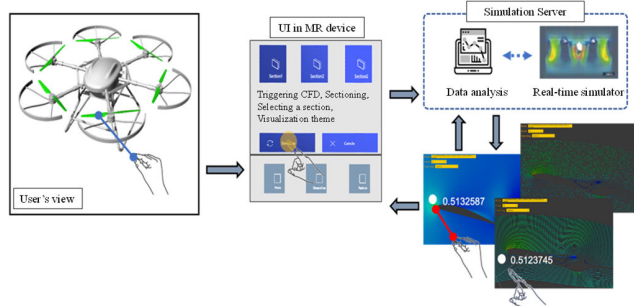
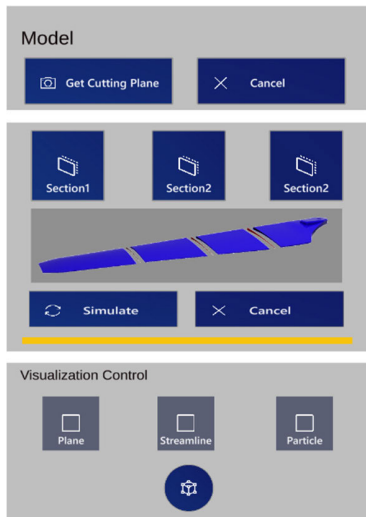


FIGURE 9. A query process using pointer gestures for review of simulation results.



(a) Scenario flow



(b) Main control panel UI

FIGURE 10. Demo scenario - interactive air flow analysis around airfoils of three sections in a selected wing.

could readily know the shapes of airfoils in any sectional positions as the form of point distribution data which can be inputs to the mesher described in the section III-A. Wing designers and engineers always need to check its performance metrics associated with not only how strongly the fluid flow is lifting and dragging the wing but also flow velocities and pressures in particular locations. Users can start the scenario using the grab gesture. When the target wing is selected, they select a sectional position by pushing buttons. Next, they can set visualization theme using touch gestures again. Now the server conducts the simulation based on the request when

triggering a button and so analyzes the results as querying point values by gesture. The analyzed results are visualized in the client’s MR environment. Subsequently, the users can also change the visualization themes (v-flied, particle, streamline) in the runtime or check the result values at the required locations through gestures. Figure 10(b) shows the main control panel UI which contains buttons for running simulation application, deriving airfoils from wing, and selecting visualization theme. Checking simulation test result can be performed differently depending on the distance between the user and the visualized simulation result. For distant targets, they can move their fingertips’ rays to the desired location on the visualized results and select vertex positions using grab gestures. For closer targets, users can select vertex positions using touch gestures. The detailed test results can be shown in the section V-A.

V. RESULTS AND DISCUSSIONS

A. TEST RESULTS

In order to check if 3D drone model and simulation result object were well loaded into real space, preliminary augmentations were performed as shown in Figure 11. Actually, 3D model did not need to be augmented if surface detection resolution of the MR device, Hololens 2 having a sence such as LiDAR, is denser so that real drone and virtual drone model are fully digital-twined, but the authors manually augmented the 3D drone model considering current situation.

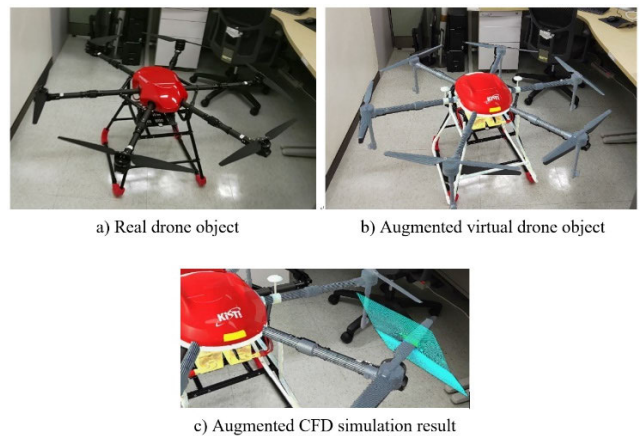
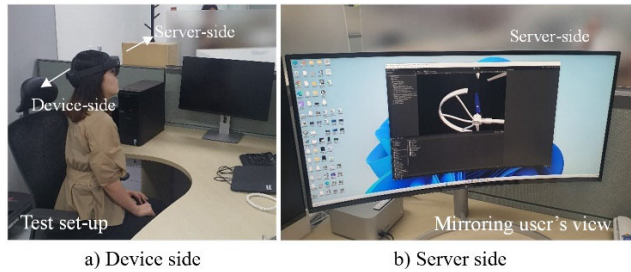


FIGURE 11. Simulation result augmentation.

Figure 12 shows test set-up which consists of device side and server side in an office room. A user wearing Hololens 2 sits down on a chair and control augmented objects by hand gesture in the device side. There is an edge server in the opposite server-side which computations of simulation, visualization, analysis, and monitoring with mirroring the user’s view are performed in the server. Consequently, Figure 12(b) shows the mirrored view results in an MR environment.

Consequently Figure 13 shows seamless sequences scene by scene, which means that all steps of ROM simulation, visualization, and analysis work immediately and



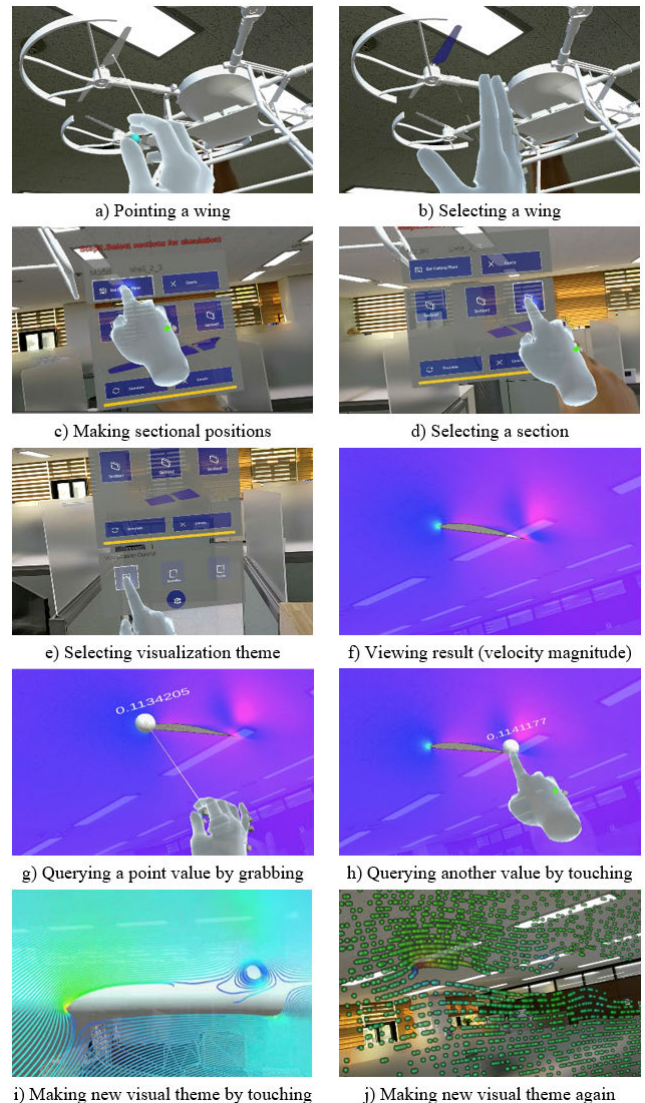


**FIGURE 12.** Test set-up with mirroring user's view in the server side.

interactively: a) pointing a wing, b) grabbing the wing, c) extracting three airfoils from the wing, d) selecting an airfoil, e) selecting visualization theme such as velocity and color map, f) visualizing a simulation result, g) querying a single node, h) querying another node, i) changing visualization theme (streamlines), and j) changing visualization theme (particle tracer). Each event for objects incorporating Unity's collision system is triggered through hand gestures. All objects representing simulation results incorporate a collision system. Initially, when a user makes a hand gesture on an object of interest in the simulation results, the collision system determines whether a collision has occurred. If a collision is detected, the system verifies the type of gesture source, stop event processing if the source is not a hand. If the gesture source type is a hand, the system conveys the collided object and collision point to the query component. This component then identifies the vertex closest to the collision point within the mesh information of the collided object and forwards it to the server. The server subsequently responds with the simulation result corresponding to that vertex. These processed simulation results are visualized within the MR environment via the rendering pipeline. The user can go back to any stage. For example, the one can move back to the stage of d) selecting another section on the user interface panel and conduct new demonstration.

## B. DISCUSSIONS

Several points were derived after implementing and experiencing a pipeline of the CFD work in this study. Firstly, the developed simulation ROM works so well since it felt like the process of CFD mesher, flow simulator, and I/O communications were conducted nearly in real time. Augmented simulation results upon a real object definitely help ones perform scientific analysis from the overall and intuitive point of view. This is why AR is useful in MRO field. In addition to this aspect, ones can handle the augmented objects by hand gestures in the MR environment. With the delicacy of hand and finger, arbitrary points in 3D space were pick up intuitively while it is difficult to pick up arbitrary points in the current 2D display. Therefore, a user experience of querying simulation results gave a feeling that analysis was conducted more immersively and interactively than in case of the current monitor display work. Besides, one can easily touch or grab the simulation object and move it to any location in 3D



**FIGURE 13.** Simulation result augmentation.

real space. This means that the whole 3D space becomes the MR user's work space. It came up with an idea that the wing design process would have been conducted much more efficiently if there existed MR authoring method of airfoils by hand gesture. Various hand gesture tools such as ruler, xy plot along a line, memo (write/copy/paste), and other augmenting reference information can be developed and help design and analysis works. By the way, it should be pointed out that using hand gesture was efficient in case that one should touch objects, but not so comfortable in case that the one should write many parameters due to the current MR UI/UX. It feels like that the touch gesture for entering characters in MR will not be able to reach current keyboard work due to its human and device interactivity.

Although narrow viewing angle and insufficient touching sense of MR device which are being improved in the research areas of computer science and engineering, the authors were able to realize the collaborative aspect of MR by



experiencing voice communications between ones on the server side and another on the device side nearby. It felt like that augmentation of simulation results in real physical space help ones perform cooperative analysis using hand gestures while overlay and occlusion have received technical attentions so far. The cooperative immersive analysis work will be conducted between users are located in the same space and look at the same simulation result object. One can move and zoom augmented objects even if it is simulation result, so it is good for customization for share them with users in the same space. After concerned objects lie in right place, users can discuss in various manners such as pointing, querying, measuring, et cetera. Communications of analysis should go significantly well with looking at other users' face, voices and body gestures.

While this study deals with a two-dimensional case, three-dimensional simulation cases will provide much more opportunities because ones are always annoyed in picking points in three-dimensional space on two-dimensional display whatever they use keyboard or mouse. On the other hand, it is expected that latency issue will arise if larger computational model such as a three-dimensional wing case is simulated. Therefore, data distribution and compression issues in the stage of I/O for such three-dimensional ROM development should be important in the future work.

## VI. CONCLUSION

This study tried to implement a digital twin pipeline which conducts a case of basic CFD work and to discuss the user experiences. An MR framework was developed containing modules of a real time CFD simulator, scientific visualization, and hand gesture for running the simulator and querying some vertex values. First, a reduced order model using CNN-based U-net algorithm was developed so that simulation results can be obtained in real time according to arbitrary shapes of airfoils and flow conditions. Next, pipelining the modules was done using several famous toolkits, and hand gestures were adequately designed and implemented for running and reviewing the simulation in an MR environment. Our user experience test demonstrated that picking up locations in 3D space, interacting with them, and even communicating between MR users in the same space would promote interactive CFD simulation works. Therefore, collaborative designing and analysis in MR should be a meaningful step in the future work.

## APPENDIX A

### ELLIPTIC PDE-BASED STRUCTURED MESH GENERATION

In this study, the simulation mesher employed elliptic PDE-based structured meshing method [22]. This method fits when the points are given along the boundaries. The feature of structured mesh is that the vertex (or node) points in the physical space are mapped onto a set of three integers  $i, j, k$  (coordinate directions in 3D). This set of integers indicates the computational space as shown in Figure 14. The coordinates  $\xi, \eta, \zeta$  in the computational space is directly related to

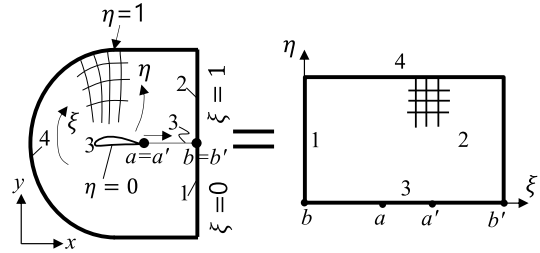


FIGURE 14. 2D C-mesh topology.

$i, j, k$  as follows:

$$\begin{aligned} \xi &= i/i_{max}, & i &= 0, 1, 2, \dots, i_{max} \\ \eta &= j/j_{max}, & j &= 0, 1, 2, \dots, j_{max} \\ \zeta &= k/k_{max}, & k &= 0, 1, 2, \dots, k_{max} \end{aligned} \quad (1)$$

This mapping implies that  $0 < \xi < 1, 0 < \eta < 1$  and  $0 < \zeta < 1$ . Neighboring mesh points can be connected to form cubes (rectangles if it is 2D) in the computational and hexahedra (quadrilaterals in 2D) in the physical space. Mesh generation systems discretize the boundary surfaces of the flow domain using quadrilateral mesh, termed the surface mesh, and fill the interior with hexahedral mesh. The mesh inside the domain can be called the volume mesh. The generation of the structured mesh starts by distributing vertex points along boundary curves or surfaces. The procedure is to place the vertex denser in regions with high curvature. Using the point distribution on boundary curves, the surface mesh can be generated. One can finally construct the volume mesh based on the surface meshes which enclose the physical domain. Therefore, the common problem is to generate a mesh inside the domain based on known boundary discretization which is performed by elliptic PDE-based method. Methods based on elliptic PDE's are known to generate meshes with smoothly varying mesh sizes and slopes of the mesh lines. Furthermore, the elliptic methods offer the possibility to control the orthogonality and the spacing near boundaries, which is important particularly for the viscous flows.

In 3D, the Poisson equations for the unknown Cartesian coordinates  $\vec{r} = [x, y, z]^T$  of the mesh points can be written as

$$\begin{aligned} &\alpha_{11}\partial^2\vec{r}/\partial\xi^2 + \alpha_{22}\partial^2\vec{r}/\partial\eta^2 + \alpha_{33}\partial^2\vec{r}/\partial\zeta^2 \\ &+ 2(\alpha_{12}\partial^2\vec{r}/\partial\xi\partial\eta + \alpha_{13}\partial^2\vec{r}/\partial\xi\partial\zeta + \alpha_{23}\partial^2\vec{r}/\partial\eta\partial\zeta) \\ &= -(P\partial\vec{r}/\partial\xi + Q\partial\vec{r}/\partial\eta + R\partial\vec{r}/\partial\zeta)/J^2 \end{aligned} \quad (2)$$

In 2D, the formula reduced to the following form.

$$\begin{aligned} &\alpha_{11}(\partial^2x/\partial\xi^2 + P\partial x/\partial\xi) + \alpha_{22}(\partial^2x/\partial\eta^2 + Q\partial x/\partial\eta) \\ &- 2(\alpha_{12}\partial^2x/\partial\xi\partial\eta) = 0 \\ &\alpha_{11}(\partial^2y/\partial\xi^2 + P\partial y/\partial\xi) + \alpha_{22}(\partial^2y/\partial\eta^2 + Q\partial y/\partial\eta) \\ &- 2(\alpha_{12}\partial^2y/\partial\xi\partial\eta) = 0 \end{aligned} \quad (3)$$

where  $(P, Q, R)$  and  $J$  indicate the control functions and Jacobian respectively. The coefficients  $\alpha$  are given by the relations:

$$\begin{aligned}
 \alpha_{11} &= (\partial\bar{r}/\partial\eta \cdot \partial\bar{r}/\partial\eta) (\partial\bar{r}/\partial\eta \cdot \partial\bar{r}/\partial\eta) \\
 &\quad - (\partial\bar{r}/\partial\eta \cdot \partial\bar{r}/\partial\xi)^2 \\
 \alpha_{22} &= (\partial\bar{r}/\partial\xi \cdot \partial\bar{r}/\partial\xi) (\partial\bar{r}/\partial\xi \cdot \partial\bar{r}/\partial\xi) \\
 &\quad - (\partial\bar{r}/\partial\xi \cdot \partial\bar{r}/\partial\zeta)^2 \\
 \alpha_{33} &= (\partial\bar{r}/\partial\xi \cdot \partial\bar{r}/\partial\xi) (\partial\bar{r}/\partial\eta \cdot \partial\bar{r}/\partial\eta) \\
 &\quad - (\partial\bar{r}/\partial\xi \cdot \partial\bar{r}/\partial\eta)^2 \\
 \alpha_{12} &= (\partial\bar{r}/\partial\xi \cdot \partial\bar{r}/\partial\zeta) (\partial\bar{r}/\partial\eta \cdot \partial\bar{r}/\partial\zeta) \\
 &\quad - (\partial\bar{r}/\partial\xi \cdot \partial\bar{r}/\partial\eta) (\partial\bar{r}/\partial\zeta \cdot \partial\bar{r}/\partial\zeta) \\
 \alpha_{13} &= (\partial\bar{r}/\partial\xi \cdot \partial\bar{r}/\partial\eta) (\partial\bar{r}/\partial\eta \cdot \partial\bar{r}/\partial\zeta) \\
 &\quad - (\partial\bar{r}/\partial\xi \cdot \partial\bar{r}/\partial\zeta) (\partial\bar{r}/\partial\eta \cdot \partial\bar{r}/\partial\eta) \\
 \alpha_{23} &= (\partial\bar{r}/\partial\xi \cdot \partial\bar{r}/\partial\zeta) (\partial\bar{r}/\partial\xi \cdot \partial\bar{r}/\partial\eta) \\
 &\quad - (\partial\bar{r}/\partial\eta \cdot \partial\bar{r}/\partial\zeta) (\partial\bar{r}/\partial\xi \cdot \partial\bar{r}/\partial\xi)
 \end{aligned} \tag{4}$$

In 2D, again, formula reduced to the following form.

$$\begin{aligned}
 \alpha_{11} &= (\partial x/\partial\eta)^2 + (\partial y/\partial\eta)^2 \\
 \alpha_{22} &= (\partial x/\partial\xi)^2 + (\partial y/\partial\xi)^2 \\
 \alpha_{12} &= (\partial x/\partial\xi) (\partial x/\partial\eta) + (\partial y/\partial\xi) (\partial y/\partial\eta)
 \end{aligned} \tag{5}$$

The elliptic equations (2) or (3) are usually discretized using second-order central finite differences. The resulting set of linear algebraic equations can be solved by any standard technique, such as by the Gauss-Seidel relaxation scheme accelerated by multigrid in numerical algorithms [38]. The control functions can be updated in an outer iteration. It is advisable to carry out several iterations with the Laplace equations ( $P = Q = R = 0$ ), in order to smooth the initial mesh and so increase the robustness of the procedure.

## APPENDIX B CONVOLUTIONAL NEURAL NETWORK

### A. AUTOENCODER

Roughly speaking, there are two functions that exist in details in deep neural network: classification and regression. The latter includes image reproducing or resolution improving. Figure A1 depicted two representative image regression algorithms which perform image reproduction work (color theme transformation). Photo images to which CNN is applied to can be regarded as two-dimensional spatial data as well as three dimensional if red/green/blue color channel is included. Since the input of widely used fully connected (or artificial) neural network which is comprised in fully connected form is limited to one dimensional array form as shown in Figure 15(b), in case that fully connected neural network is implemented in the image data, vectorizing (or flattening) of two-dimensional image data is required. But the vectorization of the image data leads to spatial information loss. Consequently, there is inefficiency of artificial neural network for two dimensional problems due to the lacking information

which occurs by loss of spatial information. CNN is a representative AI model that enables learning and inferring in state of maintaining spatial information of image.

As shown in Figure 15(a), there are two network parts. One is called convolution network that extracts a single vector as a feature and the other is called deconvolution network which generates new image from the feature vector.

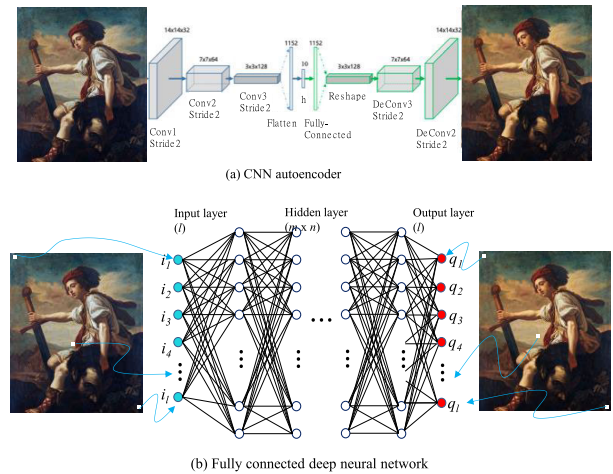


FIGURE 15. Two neural network processes of image reproduction.

In the CNN for image feature extraction, convolution filter shifts all around the input data, and feature map resulted through the convolution calculation. As shown in Figure 16, feature extraction is conducted by piling and multiplying given map and convolution kernel which strides step by step. The magnitude of the step is called ‘stride’. The stride is 1 in the case of Figure 16. By this way, feature can be extracted into two-dimensional shape. This is why the feature is called map. Figure 17 demonstrates an example of whole CNN process for three channel image input. Convolutional layer is a computing layer of multiplying activation function after applying filters to input data. Next to it, a pooling layer computation shown Figure 17(b) is optional often used for

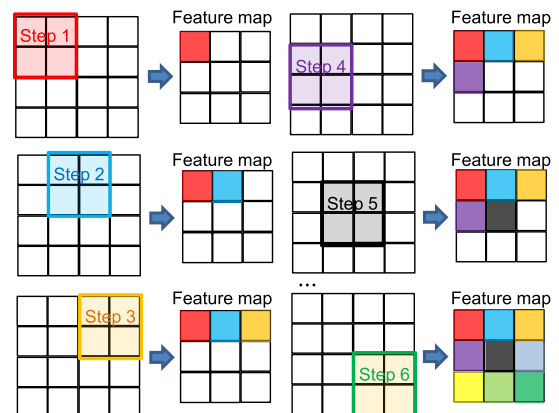


FIGURE 16. Process of extracting feature map.

improving learning efficiency. Deconvolution layer is just conducted by the reverse way. One had better add padding area shown in Figure 17(c) for the performance along the boundary. Size and shape of output data varies depending on following factors in the convolutional layer: filter size, stride, padding, and maximum pooling size.

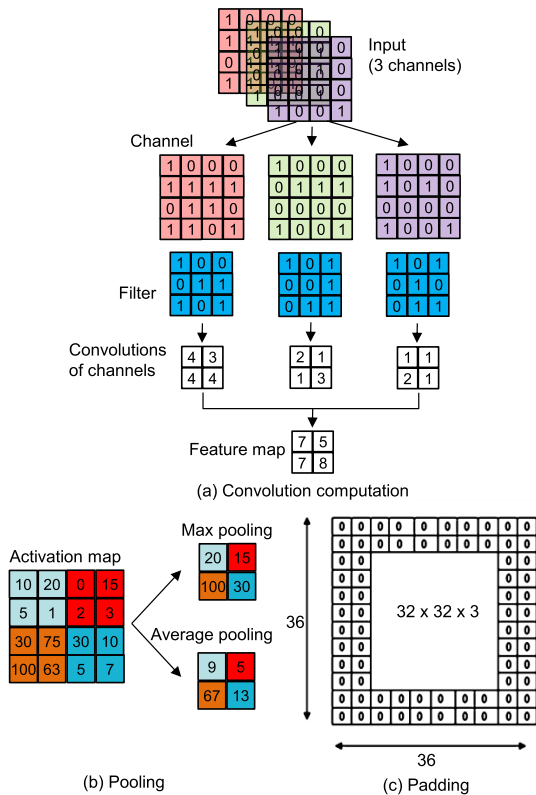


FIGURE 17. Typical implementation of CNN computation.

**B. U-NET**

Autoencoder, a representative method applying CNN to image reproduction, is an AI model possible of multiple application by giving asymmetric variation to the network

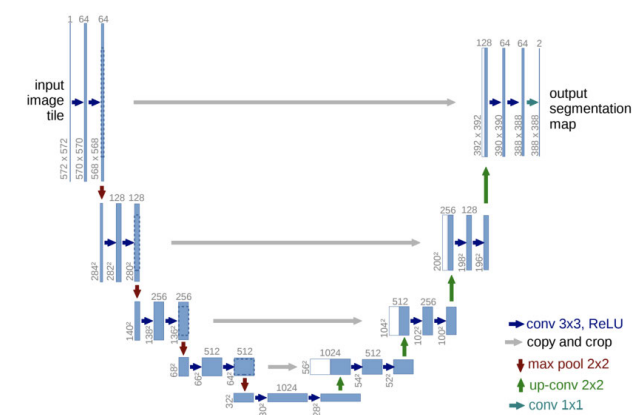


FIGURE 18. Example of U-net model (courtesy of Olaf Ronneberger et al. (2015)).

in addition to having simple functions such as reproducing and restoring images. U-net (because it is U-shaped), specifically developed for the segmentation of medical images and videos, is a kind of Autoencoder network which equalizes the shapes of two corresponding layers which lies symmetrically between convolution and deconvolution processes. However, the only difference is that the parameters calculated in the convolutional stage are used in the deconvolutional process again. That is why U-net needs additional memory consumption while is widely used due to its high accuracy. Likewise input and output fields' correlation in CFD can be learned by appropriately adding variables of interest to latent feature vector in the middle.

**REFERENCES**

- [1] M. Huzaifa, R. Desai, S. Grayson, X. Jiang, Y. Jing, J. Lee, and S. V. Adve, "ILLIXR: Enabling end-to-end extended reality research," in *Proc. IEEE Int. Symp. Workload Characterization (IISWC)*, Nov. 2021, pp. 24–38.
- [2] T. M. Peters, C. A. Linte, Z. Yaniv, and J. Williams, *Mixed and Augmented Reality in Medicine*. Boca Raton, FL, USA: CRC Press, 2018.
- [3] V. R. Curran, X. Xu, M. Y. Aydin, and O. Meruvia-Pastor, "Use of extended reality in medical education: An integrative review," *Med. Sci. Educ.*, vol. 33, no. 1, pp. 275–286, 2023.
- [4] S. Alizadehsalehi, A. Hadavi, and J. C. Huang, "From BIM to extended reality in AEC industry," *Autom. Construct.*, vol. 116, Aug. 2020, Art. no. 103254, doi: 10.1016/j.autcon.2020.103254.
- [5] M. D. O. Carrasco and P.-H. Chen, "Application of mixed reality for improving architectural design comprehension effectiveness," *Autom. Construct.*, vol. 126, Jun. 2021, Art. no. 103677, doi: 10.1016/j.autcon.2021.103677.
- [6] A. Cöltekin, A. L. Griffin, A. Slingsby, A. C. Robinson, S. Christophe, V. Rautenbach, and A. Klippel, "Geospatial information visualization and extended reality displays," in *Manual of Digital Earth*. Singapore: Springer, 2020, pp. 229–277.
- [7] S. Doolani, C. Wessels, V. Kanal, C. Sevastopoulos, A. Jaiswal, H. Nambiappan, and F. Makedon, "A review of extended reality (XR) technologies for manufacturing training," *Technologies*, vol. 8, no. 4, p. 77, Dec. 2020, doi: 10.3390/technologies8040077.
- [8] Y. Lee and B. Yoo, "XR collaboration beyond virtual reality: Work in the real world," *J. Comput. Design Eng.*, vol. 8, no. 2, pp. 756–772, Apr. 2021.
- [9] P. Milgram and F. Kishino, "A taxonomy of mixed reality visual displays," *IEICE Trans. Inf. Syst.*, vol. 77, no. 12, pp. 1321–1329, Dec. 1994.
- [10] S. Solmaz, L. Kester, and T. Van Gerven, "An immersive virtual reality learning environment with CFD simulations: Unveiling the virtual garage concept," *Educ. Inf. Technol.*, Apr. 2023, doi: 10.1007/s10639-023-11747-z.
- [11] J. Venn, C. E. Larkee, G. J. Garcia, V. L. Rayz, and J. F. LaDisa Jr., "A workflow for viewing biomedical computational fluid dynamics results and corresponding data within virtual and augmented reality environments," *Frontiers Med. Technol.*, vol. 5, Feb. 2023, Art. no. 1096289, doi: 10.3389/fmedt.2023.1096289.
- [12] J.-R. Lin, J. Cao, J.-P. Zhang, C. van Treeck, and J. Frisch, "Visualization of indoor thermal environment on mobile devices based on augmented reality and computational fluid dynamics," *Autom. Construction*, vol. 103, pp. 26–40, Jul. 2019.
- [13] J. Stam, "Real-time fluid dynamics for games," in *Proc. Game Developer Conf.*, vol. 18, Mar. 2003, p. 25.
- [14] H. Nguyen, *GPU Gems 3*. Boston, MA, USA: Addison-Wesley Professional, 2007.
- [15] R. Bridson, *Fluid Simulation for Computer Graphics*. Boca Raton, FL, USA: CRC Press, 2015.
- [16] J. N. Kutz, *Data-Driven Modeling & Scientific Computation: Methods for Complex Systems & Big Data*. Oxford, U.K.: Oxford Univ. Press, 2013.
- [17] L. Ladický, S. Jeong, B. Solenthaler, M. Pollefeys, and M. Gross, "Data-driven fluid simulations using regression forests," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 1–9, Nov. 2015.
- [18] J. Tompson, K. Schlachter, P. Sprechmann, and K. Perlin, "Accelerating Eulerian fluid simulation with convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2017, pp. 3424–3433.



- [19] B. Kim, V. C. Azevedo, N. Thuerey, T. Kim, M. Gross, and B. Solenthaler, "Deep fluids: A generative network for parameterized fluid simulations," *Comput. Graph. Forum*, vol. 38, no. 2, pp. 59–70, May 2019.
- [20] Z. J. Grey, O. A. Doronina, and A. Glaws, "Separable shape tensors for aerodynamic design," *J. Comput. Design Eng.*, vol. 10, no. 1, pp. 468–487, Jan. 2023.
- [21] Y. Wang, K. Shimada, and A. B. Farimani, "Airfoil GAN: Encoding and synthesizing airfoils for aerodynamic shape optimization," *J. Comput. Design Eng.*, vol. 10, no. 4, pp. 1350–1362, Jul. 2023.
- [22] J. Blazek, *Computational Fluid Dynamics: Principles and Applications*. Oxford, U.K.: Butterworth-Heinemann, 2015.
- [23] J. Sa, S. H. Park, K. W. Cho, and K. J. Jung, "Modeling and prediction of the crossflow transition using transition transport equations," in *Proc. 33rd AIAA App. Aerodyna. Conf.*, 2015, p. 3160.
- [24] J. H. Shin and J. H. Sa. (2022). *UIUC Airfoil Dataset, Grid, Aerodynamics, Simulation. Dataset*. [Online]. Available: <https://dataon.kisti.re.kr/search/view.do?mode=view&svclId=6946e8f70074543011c7a8ca383215f4>
- [25] J. H. Shin, W. T. Hwang, S. I. Choi, and K. W. Cho, *Introduction to AI-CFD Techniques: Problems of Flows Around Airfoils*. Daejeon, South Korea: KISTI Research Report, 2023.
- [26] D. Chen, X. Gao, C. Xu, S. Chen, J. Fang, Z. Wang, and Z. Wang, "Flow-GAN: A conditional generative adversarial network for flow prediction in various conditions," in *Proc. IEEE 32nd Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2020, pp. 315–322.
- [27] P. Pant, R. Doshi, P. Bahl, and A. B. Farimani, "Deep learning for reduced order modelling and efficient temporal evolution of fluid simulations," *Phys. Fluids*, vol. 33, no. 10, Oct. 2021, Art. no. 107101.
- [28] L. Wright and S. Davidson, "How to tell the difference between a model and a digital twin," *Adv. Model. Simul. Eng. Sci.*, vol. 7, no. 1, pp. 1–13, Dec. 2020.
- [29] F. Tao, B. Xiao, Q. Qi, J. Cheng, and P. Ji, "Digital twin modeling," *J. Manuf. Syst.*, vol. 64, pp. 372–389, Jul. 2022.
- [30] M. Berger and V. Cristie, "CFD post-processing in Unity3D," *Proc. Comput. Sci.*, vol. 51, pp. 2913–2922, Sep. 2015.
- [31] S. Ong, V. K. Siddharaju, S. Ong, and V. K. Siddharaju, "Beginning windows mixed reality programming: For HoloLens and mixed reality headsets," in *Introduction to the Mixed Reality Toolkit*. India: APress, 2021, pp. 85–110.
- [32] M. S. Selig. (1996). *UIUC Airfoil Data Site*. [Online]. Available: [https://m-selig.ae.illinois.edu/ads/coord\\_database.html](https://m-selig.ae.illinois.edu/ads/coord_database.html)
- [33] M. Dreha, "XFOIL: An analysis and design system for low Reynolds number airfoils," in *Proc. Low Reynolds Number Aerodynamics Conf.*, Notre Dame, IN, USA, Jun. 1989, pp. 1–12.
- [34] F. J. J. Joseph, S. Nonsiri, and A. Monsakul, "Keras and TensorFlow: A hands-on experience," in *Advanced Deep Learning for Engineers and Scientists: A Practical Approach*. Cham, Switzerland: Springer, 2021, pp. 85–111, doi: [10.1007/978-3-030-66519-7](https://doi.org/10.1007/978-3-030-66519-7).
- [35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural. Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.
- [36] K. Relan, *Building REST APIs With Flask*. Berlin, Germany: APress, 2019.
- [37] R. A. McDonald and J. R. Gloude-mans, "Open vehicle sketch pad: An open source parametric geometry and analysis tool for conceptual aircraft design," in *Proc. AIAA SciTech Forum*, 2022, p. 4.
- [38] B. P. Flannery, W. H. Press, S. A. Teukolsky, and W. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*. New York, NY, USA: Syndicate of the University of Cambridge, 1992.



**JUNG-HUN SHIN** (Member, IEEE) received the M.Eng. and D.Eng. degrees from the Korea Advanced Institute of Science and Technology (KAIST). His academic degrees were related to computational science and engineering and the industrial concerns used to be robust and reliable design in various machines, such as precision and mechatronic, and medical ones. Since 2015, he has been devoting himself to computational and data-driven science and engineering using advanced computing infrastructure with the Korea Institute of Science and Technology Information (KISTI). His research interest includes exploring application of scientific simulation in extended reality.



applied services utilizing these technologies.



is currently a Project Leader of the Open XR Platform Research Center, Korea Institute of Science Technology Information, Daejeon, South Korea. Her research interests include the development of open XR collaboration platform and CFD visualization using AI for digital twin.



**SANG-JIN PARK** (Member, IEEE) received the M.S. and Ph.D. degrees in industrial engineering from Chosun University, Gwangju, South Korea, in 2011 and 2017, respectively.

He is currently with the Korea Institute of Science and Technology Information (KISTI). His research interests include techniques for mesh processing, extended reality (XR) technologies, which encompass both virtual reality (VR) and augmented reality (AR), and the development of

**MIN-AH KIM** (Member, IEEE) received the B.S. degree in computer science and the M.S. degree in computer graphics from Busan National University, Busan, South Korea, in 1996 and 1998, respectively. From 1998 to 2001, she developed the Wireless Application (WAP) Gateway and Short Message Service Center (SMSC) for wireless internet in LG Electronics, Seoul, South Korea. From 2005 to 2018, she developed a real time visualization system for massive CFD data. She

**MYEONG-JIN LEE** (Member, IEEE) received the B.S. degree in atmospheric science and industrial engineering from Yonsei University, Seoul, South Korea, in 2020, and the M.S. degree from the Graduate School of Culture Technology, KAIST, Daejeon, South Korea, in 2023, where she is currently pursuing the Ph.D. degree in metaverse with the Graduate School of Convergence. Her research interest includes deep learning based cloth simulation in VR/AR.



**SEUNG-CHAN LIM** (Member, IEEE) received the B.S. degree in ICT and communication arts from Handong Global University, in 2020, and the M.S. degree in culture technology from the Korea Advanced Institute of Science and Technology (KAIST), in 2022, where he is currently pursuing the Ph.D. degree in culture technology. His current research interest includes deep learning-based cloth simulation in VR/AR.



**KUM-WON CHO** (Member, IEEE) received the M.Eng. and D.Eng. degrees from the Aerospace Engineering, Korea Advanced Institute of Science and Technology (KAIST).

He was a Principal Researcher with the Korea Institute of Science and Technology Information (KISTI). He is currently a Professor of IT convergence with the Kumoh National Institute of Technology. He has conducted large-scale computational fluid dynamics works in high performance infrastructure technology and played a leading role in establishing a national Research and Development for computational science and engineering.

...