

## Visual analysis of three-dimensional flow field based on WebVR

Shilin Zhao, Sheng Jin, Congfang Ai and Nan Zhang

### ABSTRACT

With the rapid development of internet technologies, it is possible to provide computing services and visualize calculated results on the internet. A three-dimensional flow field visualization method based on WebVR is presented in this paper. We devised and built an immersive and interactive three-dimensional virtual reality scene employing web-standard technologies (i.e., HTML5, JavaScript, WebGL, and Ajax) and computing services provided by hydrodynamic software, using GPUs to accelerate the display of flow field in the browser, without the use of plug-ins. On the basis of three-dimensional topography and surface flow field, a three-dimensional flow field presentation method of superimposing multiple sections into the original computational domain was proposed. Furthermore, the description of tracer sphere and path line was adopted to describe the structure characteristics of the flow field. Replacing complete three-dimensional sphere models with textured stylized particles improved the frame rate of the browser greatly when rendering animations. This research enables developers and users of the hydrodynamic model to be immersed in their data of flow field using Google Cardboard. As far as we know, this is the first time that WebVR technology has been applied in three-dimensional hydrodynamic simulation.

**Key words** | hydrodynamic model, network simulation, visual analysis, WebGL, WebVR

**Shilin Zhao** (corresponding author)

**Sheng Jin**

**Congfang Ai**

State Key Laboratory of Coastal and Offshore Engineering,

Dalian University of Technology,

Dalian 116024,

China

E-mail: zhaoshilin@mail.dlut.edu.cn

**Nan Zhang**

Liaoning University of International Business and

Economics,

Dalian 116024,

China

### INTRODUCTION

More and more three-dimensional hydrodynamic models, like Delft3D, EFDC, MIKE 3, FLOW-3D, have been applied to engineering practice, which contributes to studying the spatial-temporal characteristics of water flow movement accurately (Freeman *et al.* 2015; Ashall *et al.* 2016; Gems *et al.* 2016; Yin & Seo 2016). However, the above-mentioned models are primarily desktop applications, and rarely used as public web services (Sun *et al.* 2015). It is necessary to install corresponding software packages on each computer to run these programs. Moreover, most of their post-processing modules are presented in several two-dimensional horizontal planes and vertical planes to exhibit the three-dimensional scenario indirectly (Kang & Jin 2010). The visualization of calculated results is limited in terms of the degree of user interaction and user control, including a

lack of support for the ability to zoom, pan, rotate, and perform any type of highly interactive visualization process and analysis (Yin *et al.* 2015).

We are far away from the conception of 'web page' and have accepted the view of 'web application'. Browsers are changing from simple display tools to highly available application platforms as web technologies improve. One of these improvements is the emergence of the HTML5 standard and its related APIs. HTML5 is the latest HTML standard, which is a language for structuring and presenting content for the World Wide Web (Boulos *et al.* 2010). It extends the characteristics of traditional HTML, such as two-dimensional graphics, network transmission, local data storage, and enables sophisticated and elegant web applications to run in many different systems. The WebGL is an extension of

HTML5 canvas element, which is a three-dimensional image API on web proposed by Khronos (2009). First, it uses JavaScript scripting language to implement web interactive three-dimensional animation without any browser plug-in support. Second, it achieves graphics rendering by taking advantage of the underlying graphics hardware (the GPU) acceleration capabilities with a unified, standard, cross-platform OpenGL interface (Lian *et al.* 2017). In the past, users could only rely on plug-ins or download local applications to achieve the three-dimensional experience. Now, three-dimensional display can be achieved using WebGL only to write simple code. WebGL is now fully supported in the latest versions of all major browsers (whether on desktop or mobile). WebGL and associated HTML5 break down the barriers for the development of browser-based interactive multimedia applications.

Several efforts have been carried out to explore the feasibility of employing web-based architectures to run virtual reality applications, aiming to get the same experience as the desktop version of three-dimensional applications. In 1994, Mark Pesce and Tony Parisi created a browser called Labyrinth, which was the early prototype of the three-dimensional browser on the World Wide Web. In December 1997, Virtual Reality Modeling Language (VRML 97), which is the original standard for Web3D, was released. In January 1998, the VRML97 was approved by the international standardization organization (ISO) (Carey *et al.* 1997). As an international standard of ISO/IEC, the stability of VRML is guaranteed, and the application of network three-dimensional technology is rapidly promoted. In 2002, the Web3D alliance issued a draft of the extensible three-dimensional (X3D) standard. In August 2004, the X3D specification was approved by ISO as the international standard of ISO/IEC 9775 (ISO 2008). VRML is still used, but is currently being replaced with X3D, which is an open standards XML-enabled three-dimensional file format to enable real-time communication of three-dimensional data across all applications and network applications. Subsequently, three-dimensional visualization technologies such as Cult3D, Viewpoint, Java3D, Unity3D, and Flash3D appeared successively.

An advantage of Web3D technologies is that they can also be better integrated with modern tools in the visualization ecosystem. In addition, the web can be the least

platform-related mode to structure and share visualizations (Roberts *et al.* 2014). However, all of the different types of Web3D technologies mentioned above are only applicable to specific browsers, or browsers usually need to install different plug-ins to render and display three-dimensional scenes (Christen *et al.* 2012). With the same Web3D technology, different plug-ins will also generate different quality rendering effects. Therefore, users need to consider the compatibility of browser plug-ins to different systems and browsers, and select the appropriate ones according to the user's requirements for the quality of the picture, which greatly affects the user experience.

With the development of virtual reality (VR)-related fields and internet technologies, the combination of VR and internet technology has become one of the developing trends. Gracanin & Wright (1999) first put forward the concept of web virtual reality (WebVR). WebVR is a web-based interactive VR system, which is based on the internet and online VR technology to display the omni-directional information of different scale scenes in the form of three-dimensional real scene simulation. WebVR is a booming frontier technology, which represents a direction for the development of VR and computer applications. Sometimes, picture crosstalk and a lack of camera lens distortion correction resulted in an uncomfortable viewing experience in the browser-based non-VR applications. These problems can be solved by taking advantages of WebVR technology. Several researchers have conducted studies in the application of VR-related technologies to the display of hydrodynamic model results. Liang & Molkenhain (2001) developed a virtual GIS-based hydrodynamic model system for Tamshui River. They converted the model predictions into the VRML format and showed the integration of the terrain model, fluid flow, and salinity field in the virtual environment. Wenisch *et al.* (2004) presented an application of an interactive CFD-based computational steering system, connecting a high performance computer with a VR environment. Zhang *et al.* (2004) created the three-dimensional view image of a ship using the MultiGen software of MultiGen Corporation. The texture of the ship was rooted in the photos taken from real ships. Kang & Jin (2010) employed the video anaglyph technology in three-dimensional hydrodynamic simulation. Thekkedan *et al.* (2015) used VRML to create the VR for a remotely operated vehicle during the inspection tasks.

However, most visualization tools mentioned above are standalone, running on certain operating systems and platforms, or browsers and usually need users to install different plug-ins to render and display three-dimensional scenes. These studies do not take advantage of the rapidly evolving internet technologies.

We devised and built an immersive and interactive three-dimensional VR scene employing modern web technologies, without the use of plug-ins. Through the WebVR system, users can browse scenes and objects displayed in a three-dimensional form at anytime and anywhere in the form of webpage, and can perform real-time interaction and roaming operation in order to take part in experiences that are either very difficult or impossible to take place in real life. Therefore, because of their immersion, interactivity, and involvement, WebVR applications are being increasingly deployed in many contexts (Finkelstein et al. 2011; Abbasi & Baroudi 2012; Muhanna 2015; Lv et al. 2016), for example, distance education, e-commerce, interactive entertainment, flight simulators, and geographical information systems. However, the authors have not found the application of this technology in the visualization of the three-dimensional hydrodynamic model.

To the best of the authors' knowledge, this is the first time that the calculated result of a three-dimensional hydrodynamic model using WebVR technology has been demonstrated and analyzed. This paper describes in detail how web-standard technologies can be used to implement fully immersive and interactive VR three-dimensional visualization system of flow field. The remainder of this paper is organized as follows: In 'Methods', we mainly present the process of establishing the three-dimensional VR scene based on network. The 'Results and discussion' section reveals the application of this method in practical engineering. In 'Conclusions', we give a conclusion and discuss the future work.

## METHODS

### Hydrodynamic model

HydroInfo was developed by Dalian University of Technology, and can provide hydrodynamic calculation services on

the network server (Jin 2013). Users no longer need to install any of the local software, they customize the software on demand, update it online, and complete the complex hydrodynamic simulation process using only a browser. HydroInfo can be applied to the flow and transport problems of flood, breakup and inundation, sediment and river evolution analysis, tidal current and wave numerical simulation, and so on.

The two-dimensional hydrodynamic module solves the two-dimensional shallow water equation which is shown below:

$$\frac{\partial U}{\partial t} + \frac{\partial(F^I - F^V)}{\partial x} + \frac{\partial(G^I - G^V)}{\partial y} = S$$

In the above equation,

$$\begin{cases} U = \begin{pmatrix} h \\ uh \\ vh \end{pmatrix}, F^I = \begin{pmatrix} uh \\ u^2 + gh^2/2 \\ uvh \end{pmatrix}, G^I = \begin{pmatrix} vh \\ uvh \\ v^2h + gh^2/2 \end{pmatrix} \\ F^V = \begin{pmatrix} 0 \\ \mu_e \partial uh / \partial x \\ \mu_e \partial vh / \partial x \end{pmatrix}, G^V = \begin{pmatrix} 0 \\ \mu_e \partial uh / \partial y \\ \mu_e \partial vh / \partial y \end{pmatrix}, \\ S = S_0 + S_f = \begin{pmatrix} 0 \\ ghS_{0x} \\ ghS_{0y} \end{pmatrix} + \begin{pmatrix} 0 \\ -ghS_{fx} \\ -ghS_{fy} \end{pmatrix} \end{cases}$$

where  $U$  is the independent variable,  $F$  and  $G$  are the fluxes in the x-direction and y-direction, respectively;  $S$  is the source term, superscript  $I$  and  $V$  are the convective flux and viscous flux, respectively;  $h$  is water depth,  $u$ ,  $v$  is velocity in the x-direction and y-direction, respectively;  $g$  represents gravity acceleration,  $S_{0x}$ ,  $S_{0y}$  are bottom slope source terms in the x-direction and y-direction, respectively;  $S_{fx}$ ,  $S_{fy}$  are bottom friction source terms in the x-direction and y-direction, respectively;  $S_{0x}$ ,  $S_{0y}$ ,  $S_{fx}$ ,  $S_{fy}$  can be written as:

$$S_{0x} = -\frac{\partial Z_b}{\partial x}, S_{0y} = -\frac{\partial Z_b}{\partial y}, S_{fx} = \frac{n^2 u \sqrt{u^2 + v}}{h^{4/3}}, S_{fy} = \frac{n^2 v \sqrt{u^2 + v}}{h^{4/3}}$$

where  $n$  represents roughness, and  $Z_b$  represents bottom elevation.

Unstructured grids are generated by Delaunay method (Watson 1981). A vertex-centered finite volume method is

used for solution of the two-dimensional shallow water equation on unstructured grids. Based on the approximate Riemann solution of Roe scheme, a Project Limited Central Difference (PLCD) method based on Monotonic Upwind Scheme for Conservation Laws (MUSCL) scheme is constructed. By correcting the source term of the bottom slope to solve the hydrostatic problem under submerged condition, the harmony of the higher order scheme is guaranteed and the second-order accuracy of the scheme is guaranteed. The conservative flux limitation method is used to avoid negative water depth at the wet-dry interface, to ensure that there is no mass error in the calculation, and to better deal with the problems of dynamic boundary and total mass conservation.

### System architecture

For completeness, the overall framework of the online hydrodynamic model system based on B/S architecture is shown in Figure 1. It is different from traditional hydrodynamic applications that are based on desktop. Based on the browser/server structure, the hydrodynamic model Hydro-Info was used on the server to provide computing services, which is the core of the whole system. It consists mainly of discrete method, numerical model, file system, database engine, and so on. The data in the database can be divided into graphic data, modeling data, computing data, and basic information data, which provides data support for the entire system. On the browser side, there are two main parts: modeling and flow field demonstration. The main functions are numerical modeling, parameter setting, statistical inquiry of simulation results, and display of flow field. Asynchronous JavaScript and XML (AJAX) used asynchronous data transfer (HTTP request) between the browser and the web server, allowing web pages to exchange data with the web server without reloading whole pages. It promotes local renewable pages, which lightened the burden of the server and enhanced the user experience.

The focus of this paper, as shown in the top left corner of Figure 1, is to use WebVR technology to realize three-dimensional rendering of flow field in the browser. Through the WebVR technology, users can enhance their immersion experience by using virtual reality hardware. In this paper, iPhone 6S and Google Cardboard are assembled into a

simple HMD (head-mounted display). The screen is split into two parts that display a stereoscopic image angled slightly different for each eye respectively, which enables people to see the three-dimensional content of the screen based on binocular stereo vision. In addition, it relies on accelerometers, gyroscopes, and other sensors in the phone to interact with the scene. Users can access the visualization system through laptops, desktops, and smartphones with HMD.

### Establishment of three-dimensional virtual reality scene

This paper uses mature and open source WebGL three-dimensional engine graphics framework and some other JavaScript libraries, such as Three.js (Dirksen 2013), to build a three-dimensional virtual reality scene in the browser. WebGL allows access to graphics processing hardware (the GPU) directly from the browser (via JavaScript) to accelerate rendering. Programmers do not need to know the complex underlying details and coloring language of WebGL to build realistic three-dimensional scenes conveniently and quickly.

The three-dimensional virtual reality scene is constructed using the calculated results of the hydrodynamic model HydroInfo, which are stored on the server side in the form of files. The structure of the result file is shown in Figure 2. On the client side, users request data from the web server through the web browser. After receiving the data from the server, the browser parses it. With the support of data, the three-dimensional scene can begin to be built. First, a *THREE.Scene* object is created which is used to store all other different objects. Second, we add a camera to the scene. *Three.js*, that is the JavaScript three-dimensional library for WebGL, provides us with two different types of cameras: orthogonal projection camera and perspective projection camera. In this paper, we chose a perspective projection camera that could produce a more realistic world effect. The farther away from the camera, the smaller the object would be rendered. Third, a light source is added to the scene to make the rendered scene visible. *Three.js* contains a large number of light sources with different usages. The light sources in this article are *THREE.AmbientLight* and *THREE.Directionallight*. The color of *THREE.AmbientLight* will be superimposed on

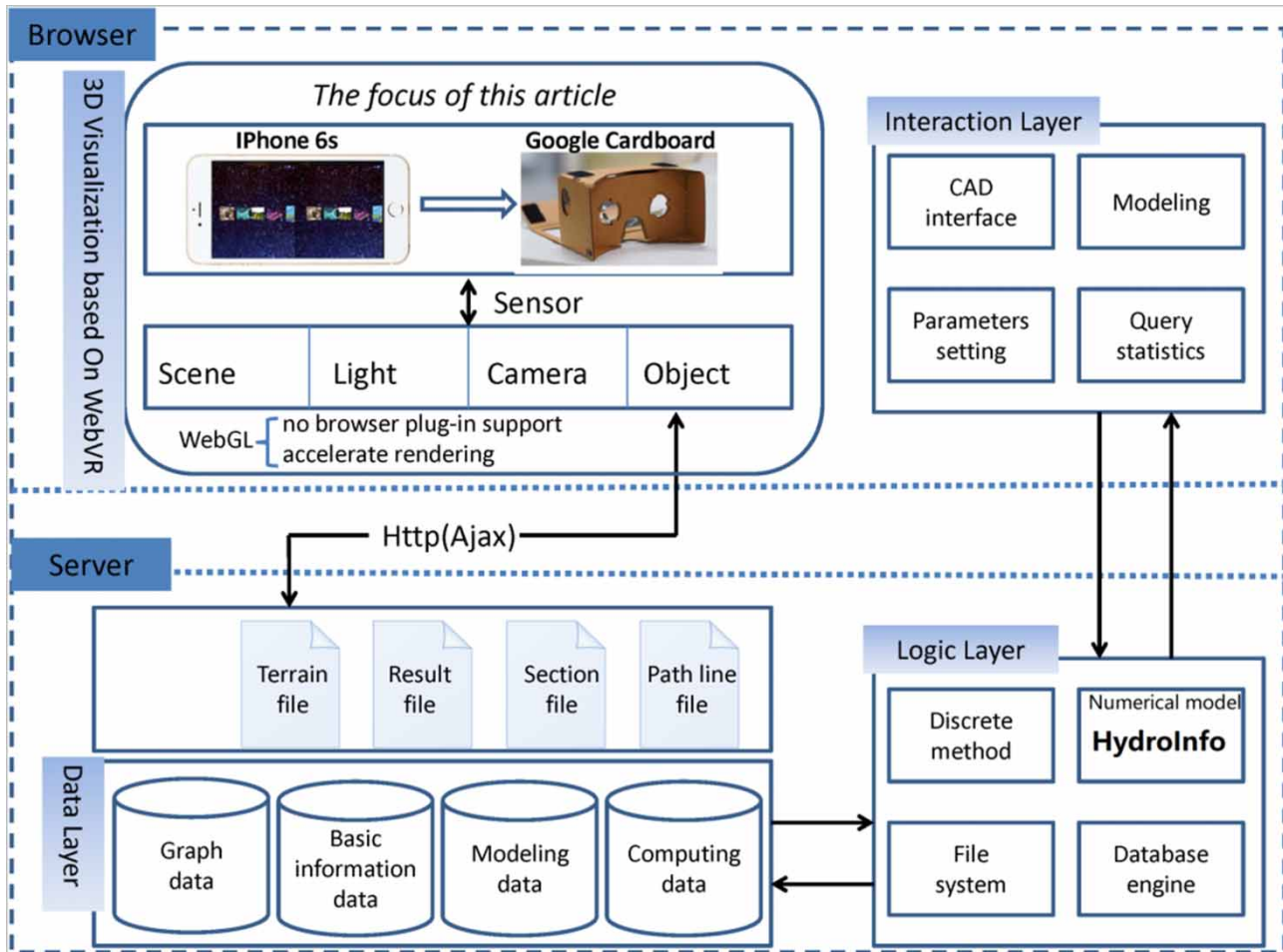


Figure 1 | System framework based on B/S structure.

the color of the existing objects in the scene, and *THREE.DirectionalLight* can be used to create the shadow. Fourth, a three-dimensional mesh is created, which is the most difficult and important part of the whole process of scene building. It consisted of two parts, geometry and material. *THREE.Geometry* defines the shape of the object. The computational domain is discretized into many unstructured grids by the hydrodynamic model. The elevation value of each grid node is obtained by terrain elevation interpolation. The node coordinates with elevation are assigned to the vertices' attributes of the instance object of *THREE.Geometry*, and the connection relationships of the grid are assigned to the faces attributes of the instance object of *THREE.Geometry*. Then, the *computeFaceNormals* method is executed to determine the normal vector of each face. The

creation of the three-dimensional geometry is completed. This article uses *THREE.MeshLambertMaterial* to define the appearance and material of the object. It is a material that takes into account the effects of illumination and is used to create dim, non-bright objects. According to the height of the geometric node, the color of each vertex is determined to create a gradient effect for the material. Then, the geometry and material are merged into the grid that can be added to the scene to complete the establishment of the three-dimensional grid. Fifth, coordinate transformation: the objects that need to be rendered are properly displayed on the computer screen through the coordinate transformation of translation, zoom, rotation, etc. Sixth, *WebGLRenderer* renders the scene using the computer graphics.



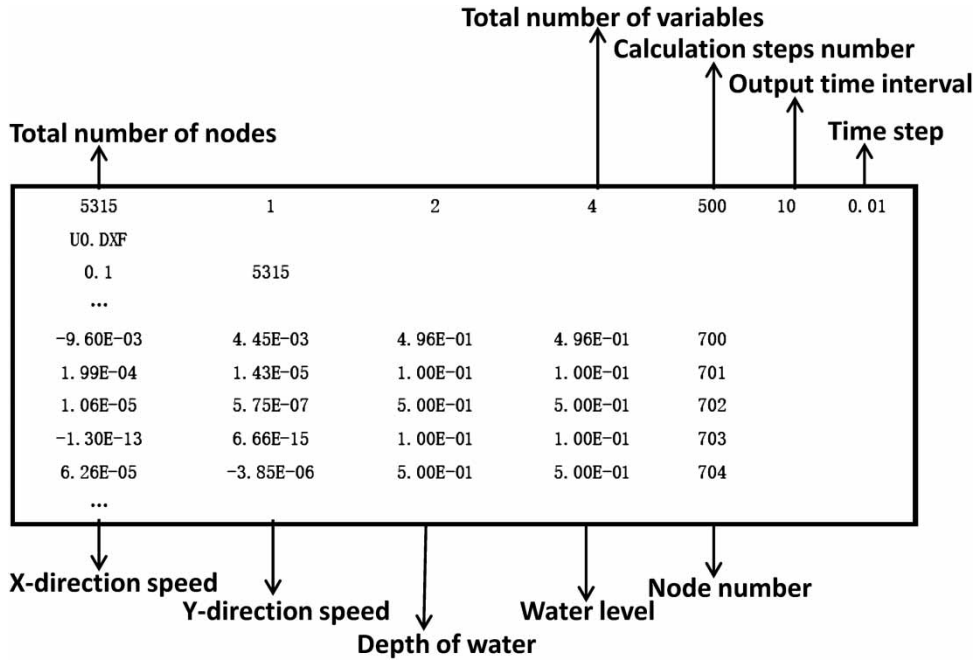


Figure 2 | Definitions of the result file structure.

Finally, the ordinary three-dimensional scene drawn above is converted into a VR scene, using the A-Frame framework. A-Frame is an open source three.js framework made by Mozilla, which uses the WebVR API. The framework utilizes an entity-component-system pattern and an asset management system (Mozilla 2018). The entities alone are basically empty containers, without any attributes, unless components are inserted into the entities' sockets. Components are what make the entities have visibility, color, animations, functionality (Hedelin 2017), etc. The objects described above in the scenario are added to the A-Frame components separately. Then, components are inserted into the different types of entities' sockets. The

three-dimensional VR scene is established. At this point, when the user clicks the 'Enter VR' button, the screen is cut into two parts, and the content displayed on the two parts has a certain difference. The user can get the immersive experience by using a mobile phone mounted in Google Cardboard. In addition, users can interact with VR scenes using a Gazed-Based cursor component. The entire visualization process of the three-dimensional WebVR scene is shown in Figure 3.

In the process of building a three-dimensional VR scene online, the following technical difficulties were encountered. First, *Morphtarget*, the built-in method of three.js, was used to animate the water surface. The memory cannot be freed, and

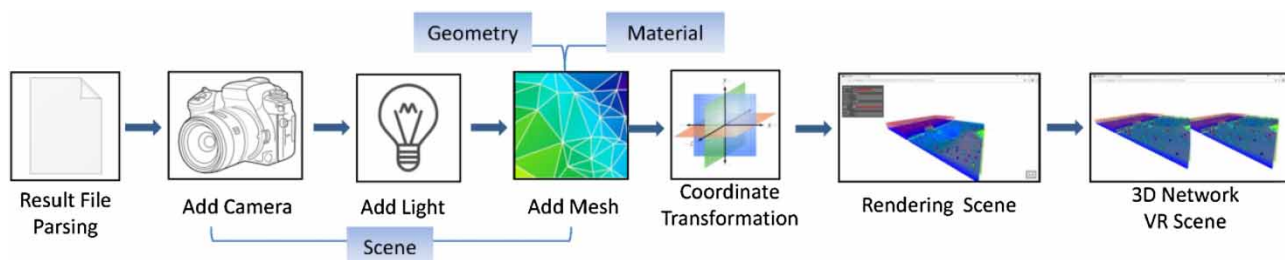


Figure 3 | Visualization process of three-dimensional WebVR scene.

the color of the water surface cannot change with time. In order to solve this problem, we adopted the method of destroying the model of the previous moment in the process of rendering the scene, and rebuilding the model of the new moment. Second, when rendering the velocity flow field, an oriented cone needs to be drawn at each mesh point. Due to the excessive number of grid points, the burden in the scene was too large, and which can easily cause rendering failure. We adopted a method named *Geometry.mergeMesh* to solve this technical difficulty.

### Cross-section generation

In order to visualize the analytical result of the cross-section, the mesh generation of the cross-section is the key. First, the section lines are reprocessed so that they fall entirely into the computational domain. Then, it is necessary to divide each section line into several equal parts in order to smooth the color transition of the cross-section. The endpoints of each equal part are used as the identification points of the section line. According to the coordinates of the identification points and the corresponding elevation values of the free surface at different moments, the nodes of the profile grid are obtained, and then the connection information between nodes is constructed to complete the gridding of the profile.

Suppose the identification point P locates in the mesh element E, and the three vertices of element E are A, B, and C, respectively. The values of variables at the point P are calculated according to the barycentric interpolation method. The values of the variables at the three vertices are  $Val_A$ ,  $Val_B$ , and  $Val_C$ , respectively, and the areas of triangle PBC, PAC, and PAB are  $S_{\Delta PBC}$ ,  $S_{\Delta PAC}$  and  $S_{\Delta PAB}$ . Then, the value of the variable at point P  $Val_P$  can be calculated as:

$$Val_P = \frac{Val_A * S_{\Delta PBC} + Val_B * S_{\Delta PAC} + Val_C * S_{\Delta PAB}}{S_{\Delta PBC} + S_{\Delta PAC} + S_{\Delta PAB}}$$

### Representation of flow field

The Lagrange method is often used to describe the flow field in order to better track the flow trajectory in accordance with people's intuitive understanding of the flow movement.

The flow field calculated by the hydrodynamic model is represented by the Euler method, which reflects the time-varying process of the variables at grid nodes or cell centers. The Lagrange field describes the motion track of the particle, which is usually characterized by path lines and tracer balls (Zhang et al. 2018).

It is assumed that the location of the marker point on the path line or tracer ball is  $X(t)$  at time  $t$ , and the velocity of movement is:

$$V(X) = \frac{dX(t)}{dt}$$

After  $\Delta t$  moves to the new location

$$X(t + \Delta t) = X(t) + \int V(X(t))dt$$

The second-order Runge-Kutta method is adopted to solve the problem in order to obtain higher time precision, guarantee a certain computational efficiency, and meet the engineering requirements. The new position can be calculated using the following formula:

$$\begin{cases} X(t + \Delta t) = X(t) + \frac{1}{2}(K_1 + K_2) \\ K_1 = \Delta t * V(t, X(t)) \\ K_2 = \Delta t * V\left(t + \frac{\Delta t}{2}, X(t) + \frac{K_1}{2}\right) \end{cases}$$

The locations of the marker points are connected in chronological order to generate path lines, and the positions of the tracer balls change with time to form animation. The particles are checked to determine whether their positions exceed the calculated domain at each moment, and if so, they are deleted.

### Case study simulation

In order to test the feasibility of the method used in this paper, a case study of three-dimensional dam break visualization was taken as an example. This example is an asymmetric dam break test with an obstacle downstream. There are dry and wet interfaces in the process of dam

break, and complex flow states such as reflected flow and backflow are formed in many places. As shown in Figure 4, the pool with solid walls around it is divided into two parts, the middle of which is controlled by a sluice gate, and a pyramid-shaped obstacle is arranged in the downstream submerged area. The initial water level of the reservoir is 0.5 m and that of the flood plain is 0.1 m.

Our current network visualization system for presenting the case study was built using the Three.js and A-Frame, an open source three.js framework uses the WebVR API. The data used for visualization were calculated by HydroInfo.

The case study results have been measured on a personal laptop that was equipped with an Intel(R) Core(TM) i7-6700HQ 2.60GHZ CPU, 8.00 GB of memory, and an NVIDIA GeForce 960M graphic with 2.0 GB of graphics memory. The operating system was the Windows 10 64-bit Professional Edition, the development environment was JetBrains WebStorm 2018.1.1 x64, and the browser used was Google Chrome 67, although the system had been successfully tested to work with all major browsers, including mobile ones such as Mobile Safari, Firefox 25+, and Android Browser. The hydrodynamic model HydroInfo is running on the server to provide computing services. On the browser side, the user sends a computing request to the server. After the server receives the request, the

background program calls the model for hydrodynamic calculation and saves the calculated results on the server. The calculation progress is constantly returned to the front end. When the browser receives the instruction that the calculation has been completed, the user can send a visualization request to the server to load the results of the server-side calculation. The calculated results include different hydraulic factor values of different nodes at different moments. After receiving the data from the server, the browser parsed it. Then, the case study calculated results are shown in our three-dimensional VR scene. Different from the traditional way of redefining a new canvas to analyze the profile data in a two-dimensional pattern, a three-dimensional flow field with multiple profiles shows the position of profiles in the original computational domain and is conducive to the comparative analysis of multiple cross-section data. When the Lagrange method is used to describe the three-dimensional flow distribution, path lines and tracer particles are used to track the trajectories of mass particles to make us better understand the flow field.

## RESULTS AND DISCUSSION

### Case study results

The three-dimensional scene can be rendered on all the main brands of web browsers in a computer, and viewed with the naked eye. As shown in Figures 5 and 6, the three-dimensional visualization scenes of the plain Three.js version are displayed on regular browsers, which can be interacted with mouse controls. Figure 5(a) shows the three-dimensional topography whose height is represented by different rainbow colors. Figure 5(b) reveals the grid, which consists of 5,269 nodes and 10,176 elements. The free surface and local flow fields of the three-dimensional dam break are shown in Figure 5(c) and 5(d). The direction of the cone represents the velocity direction of the flow field, and the length of the cone indicates the magnitude of the flow field velocity. The animation vividly shows the violent change process of surface flow velocity and water surface when dam break occurs. The velocity at the dam break is large and the change of water surface is intense. Due to the action of plunging flow, there are two major circulations

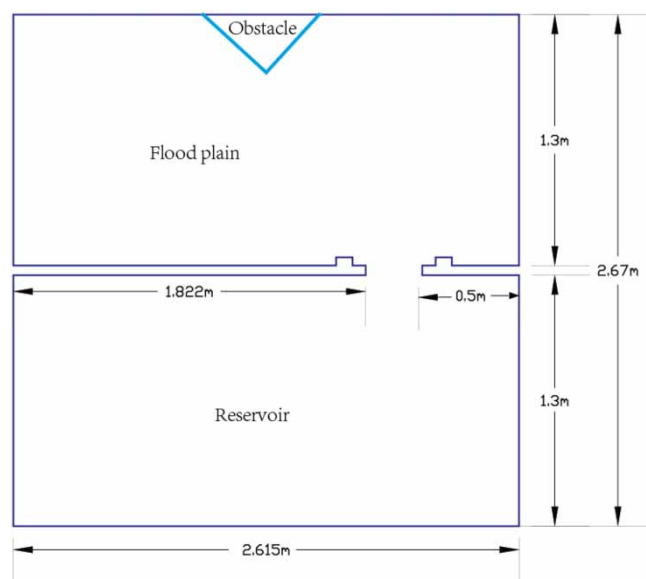
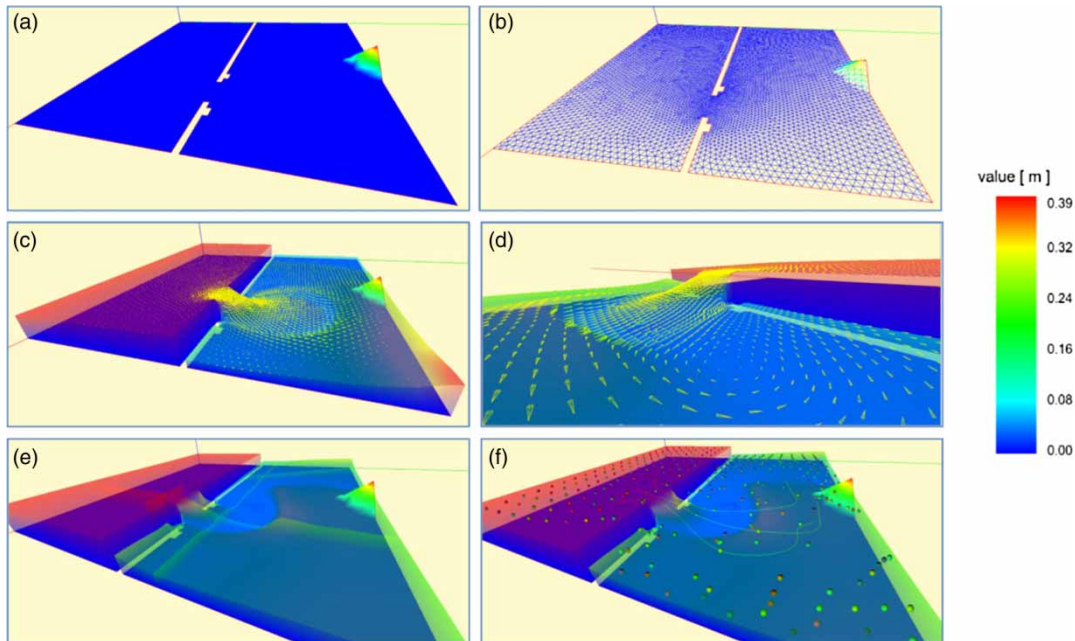


Figure 4 | Plane of the case study area.





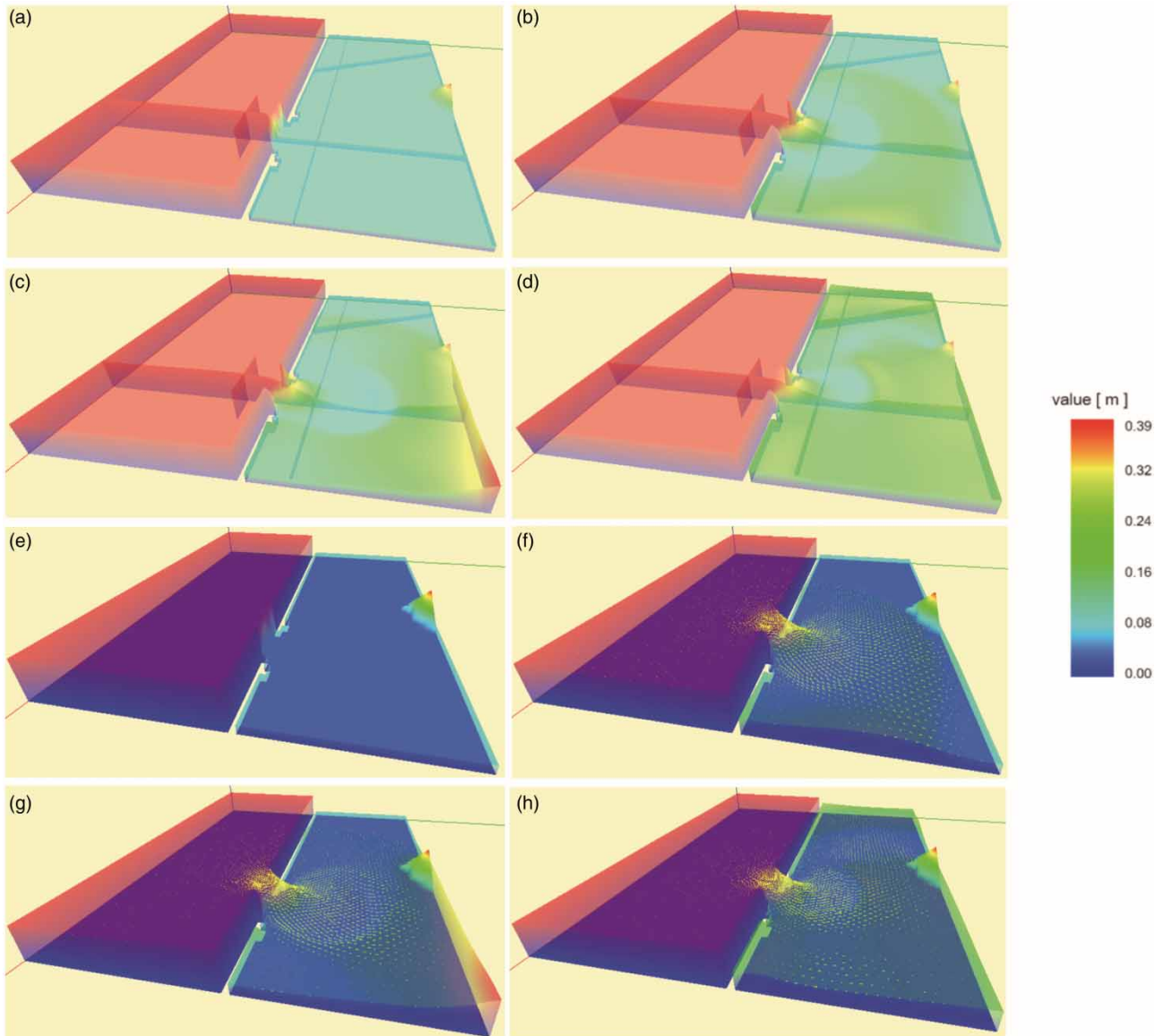
**Figure 5** | Three-dimensional visualization results of the plain Three.js version shown in a desktop browser: (a) three-dimensional topography, (b) grid, (c) free surface, (d) local flow fields, (e) flow field with multiple profiles, (f) flow field expressed by path lines and trackballs.

on both sides of the dam break, the left side of the dam break is a counterclockwise circulation, and the right side is a clockwise circulation, as shown in [Figure 5\(d\)](#). Meanwhile, with the superposition of reflected water waves, the phenomenon of local backwater is obvious. [Figure 5\(e\)](#) exhibits a three-dimensional flow field with multiple profiles. The three-dimensional visualization of path lines and tracer particles is realized, as shown in [Figure 5\(f\)](#). [Figure 6\(a\)–6\(d\)](#) show cross-section changes at different times. [Figure 6\(e\)–6\(h\)](#) display flow velocity and water level time series of the dam break. The three-dimensional network-based visualization system can achieve the same rendering effect as similar studies done by [Kang & Jin \(2010\)](#) and [Zhang et al. \(2016\)](#). They realized the three-dimensional display of dam break on the client side using Windows Presentation Foundation (WPF) based on client/server (C/S) architecture.

The three-dimensional scenes in [Figures 5](#) and [6](#) are implemented using pure Three.js. They can be rendered at a very high frame rate, even with frequent mouse interactions. However, sometimes picture crosstalk and a lack of camera lens distortion correction result in an uncomfortable viewing experience.

In order to solve these problems, we also built a three-dimensional virtual reality visualization system using WebVR technology, on top of Three.js, on the basis of the above work. The problem of insufficient camera lens distortion correction was solved by taking advantage of Google Cardboard. The three-dimensional virtual reality visualization system using WebVR technology improved the image clarity greatly compared with the pure Three.js implementation. While using WebVR technology could affect the frame rate, to some extent, when too many objects were rendered in the scene, the overall experience of the WebVR version was considered to be better than the ordinary Three.js version.

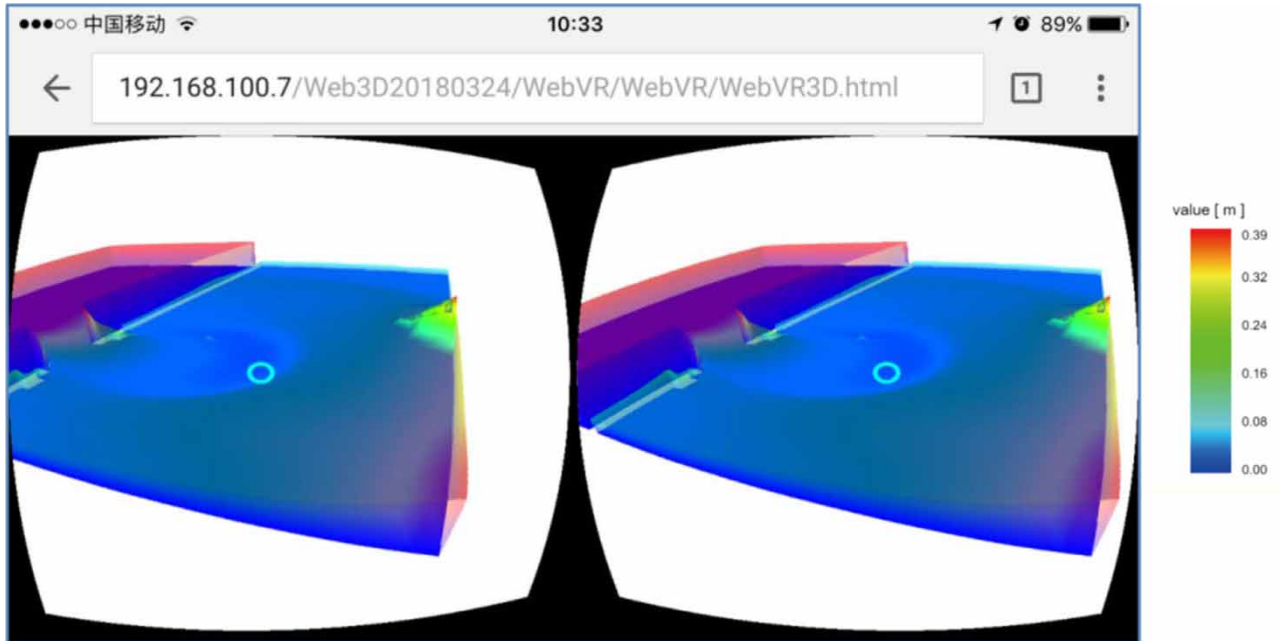
The three-dimensional virtual reality visualization system based on WebVR was built, and the expected results were achieved. The three-dimensional visualization results of flow fields in Google Cardboard mode on a mobile device are shown in [Figures 7–9](#). [Figure 7](#) shows the surface flow fields of three-dimensional dam break at  $t = 11$  s and [Figure 8](#) is the presentation of flow fields with multi-cross-sections. [Figure 9](#) reveals the three-dimensional flow field at  $t = 20$  s expressed by path lines and trackballs. It renders a split screen, stereo version of a three-dimensional scene.



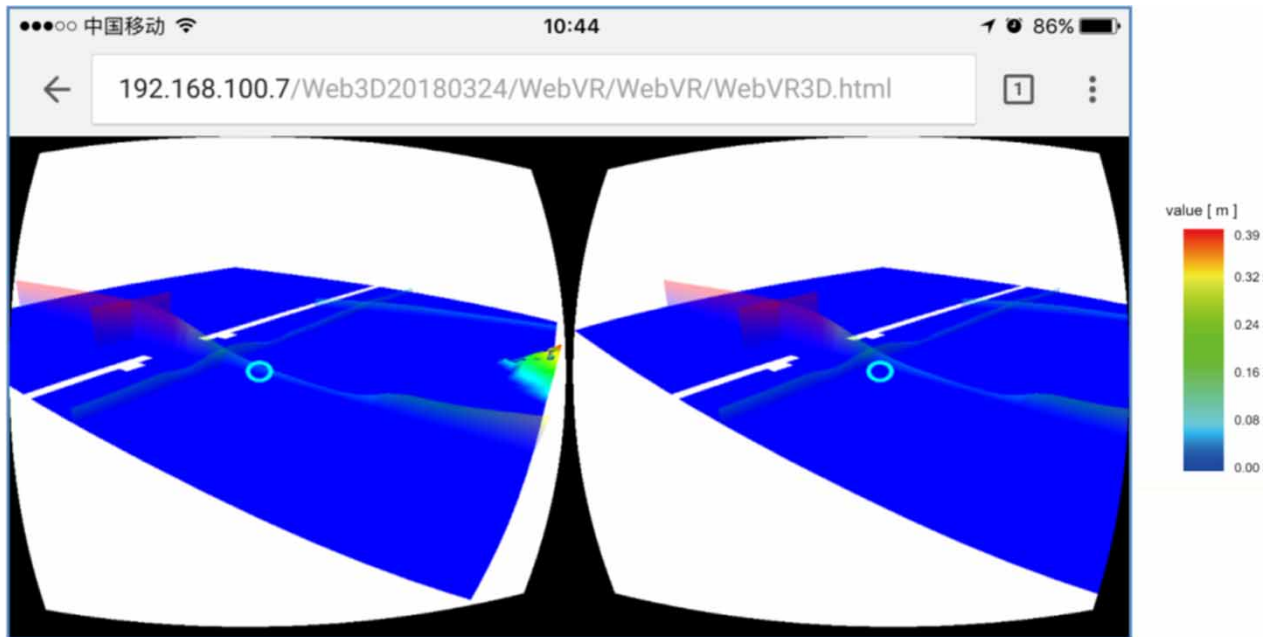
**Figure 6** | Three-dimensional visualization results of the plain Three.js version shown in a desktop browser: (a) cross-sections at  $t = 0$  s, (b) cross-sections at  $t = 8$  s, (c) cross-sections at  $t = 11$  s, (d) cross-sections at  $t = 24$  s, (e) flow velocity and water level at  $t = 0$  s, (f) flow velocity and water level at  $t = 8$  s, (g) flow velocity and water level at  $t = 11$  s, (h) flow velocity and water level at  $t = 24$  s.

When viewed with the naked eye, the two parts displayed on the screen had a certain angle difference. One can see the stereo effect and get the immersive experience with the help of a HMD (such as a mobile phone mounted in Google Cardboard). Our virtual reality system utilized head tracking with 3 degrees of freedom (roll, pitch, and yaw) for mobile browsers with Google Cardboard, relying on a device's accelerometer, gyroscope, and other sensors.

By shaking your head, you can see the local details of the flow field you are focusing on. The WebVR environment enables us to observe first hand, and helps in providing better insights about the complicated flow phenomena in the virtual three-dimensional world. It vividly shows the violent change process of surface flow velocity and water surface when dam break occurs. The velocity at the dam break is large and the change of water surface is intense.



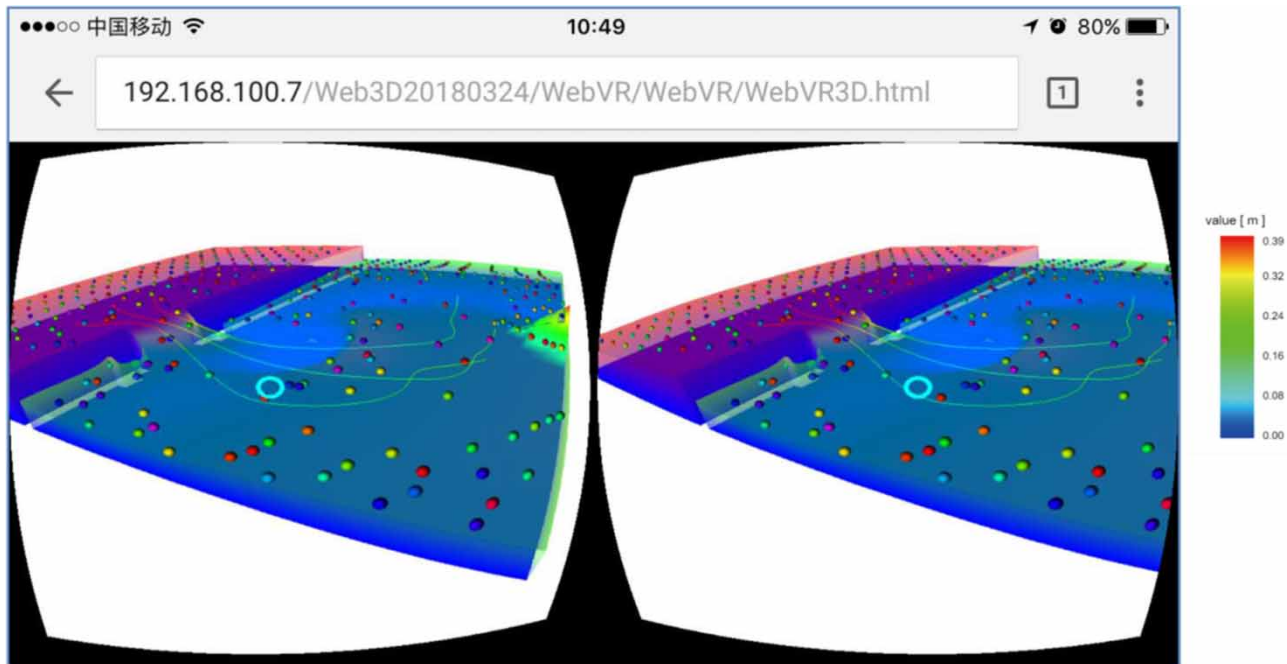
**Figure 7** | Surface flow fields displayed in Google Cardboard mode on a mobile device at  $t = 11$  s.



**Figure 8** | Flow fields with multi-cross-sections in Google Cardboard mode on a mobile device at  $t = 11$  s.

Due to the action of plunging flow, there are two major circulations on both sides of the dam break. Meanwhile, with the superposition of reflected water waves, the phenomenon of local backwater is obvious. One can perform real-time

interaction and roaming operation with the flow field. With the aid of a Gazed-Based cursor component, users can interact with VR scenes on the internet. These new capabilities in terms of visualization were not available



**Figure 9** | Flow fields with path lines and trackballs displayed in Google Cardboard mode on a mobile device at  $t = 20$  s.

previously. It is worth mentioning that WebVR mode and non-WebVR mode can be freely switched.

### Performance improvement

An important metric to consider in web applications is the frame rate, which is mainly the number of frames processed per second (Marcelo *et al.* 2016). Therefore, it is important to measure the frame rate performance that can be procured when using web technologies. In fact, the frame rate is supposed to be as high as possible to augment the sense of immersion, interaction, and participation. In this section, we have tested the performance of our visualization system using the stats component which displayed a UI with performance-related metrics. Several performance metrics were considered, including fps (frames per second), requestAnimationFrame (latency), textures (number of three.js textures in the scene), programs (number of GLSL shaders in the scene), geometries (number of three.js geometries in the scene), vertices (number of vertices in the scene), faces (number of faces in the scene), calls (number of draw calls on each frame), load time (how long it took for the scene to

start rendering, in MS) and entities (number of A-Frame entities). However, the performance of frame rate can be affected by many factors. The slow frame rate is mainly caused by a complex scene that is beyond the rendering ability of the graphics card. When the three-dimensional flow field is described by the Lagrange method, it is often necessary to show the movement of the flow more intuitively by tracing the trajectory of the particles. In this paper, the shape of tracer particles was selected as three-dimensional spheres. The rendering of a three-dimensional sphere is mainly controlled by three attributes: the radius, the number of segments in the horizontal direction, and the number of segments in the vertical direction. The radius determines the size of the final mesh of the sphere. The more segments in both directions, the smoother the surface of the sphere will be. Experiments showed that when the number of segments was 8 and the radius of the sphere was 2, the smooth degree of the sphere could be satisfied. However, when the number of spheres increased greatly, the number of vertices and faces in the whole scene would increase significantly and the FPS of the browser would decrease obviously, thus the rendering of the scene would not seem smooth.



Some papers have discussed issues related to frame rate performance when using web technologies. [Anthes \*et al.\* \(2005\)](#) have tried to deal with jitter problems with buffering techniques to improve frame rate performance. [Steinbach \*et al.\* \(2010\)](#) discussed how package loss impacted the frame rate. [Marcelo \*et al.\* \(2016\)](#) ensured that their transmission of data through WebRTC took little time to reach the other nodes. Nevertheless, most of their methods focus on computer internet technologies, and they are not applicable for this study. Combined with our practical needs, this paper proposed a method of replacing three-dimensional spheres with textured styled particles to solve the above issue of network performance. First, the particle objects were created, then the sphere image was loaded into the texture by the built-function, and finally, the texture was assigned to the map attribute of the particle material. This completed the styling of the particles.

Using the sphere image as the particle texture, the number of vertices and faces in the scene was greatly reduced, and the problem of slow frame rates was solved. The number of vertices and faces in the scene is 32,724 and 10,908, respectively, when the terrain is rendered without adding the tracer particle effect, and the browser's FPS (frames per second) is 60.06. In this paper, 361 tracer particles were used, and the radius of each sphere was 2. In order to make the surface of the sphere smooth, there were eight segments in vertical and horizontal directions, respectively. When the three-dimensional spheres are added as tracer particles, as shown in the third column of [Table 1](#), the vertex number increases to 153,012 and face number increases to 51,004, which is nearly five times larger than the number of vertices and faces when the trackballs are not added. As a result, the FPS dropped to 24.78, causing the animation rendering to be unsmooth. The fourth column of [Table 1](#) shows the performance of the browser after replacing the three-dimensional spheres with textured styled particles, and one can see that the number of points has increased by 361 and the number of faces has not increased compared with the previous period. At this time, the FPS increased to 54.74, which greatly improved the fluency of the animation and solved the problem of network performance when the number of tracer particles was large. The FPS was slightly lower than 60. In

**Table 1** | Performance-related metrics in different cases

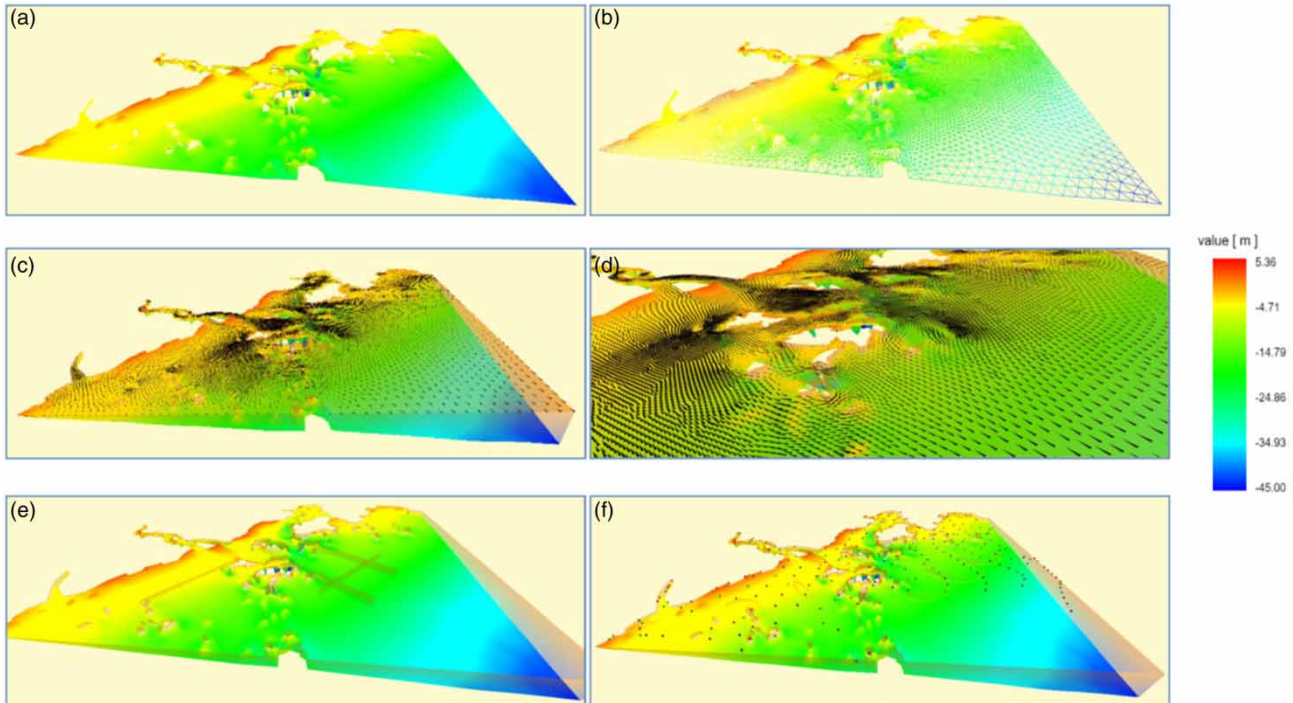
Scene rendering	FPS	Vertices	Faces	Points
Only the terrain	60.06	32,724	10,908	0
Terrain and spheres	24.78	153,012	51,004	0
Terrain and texture styled particles	54.74	33,085	10,908	361

different cases, a detailed summary of browser performance-related metrics is provided in [Table 1](#).

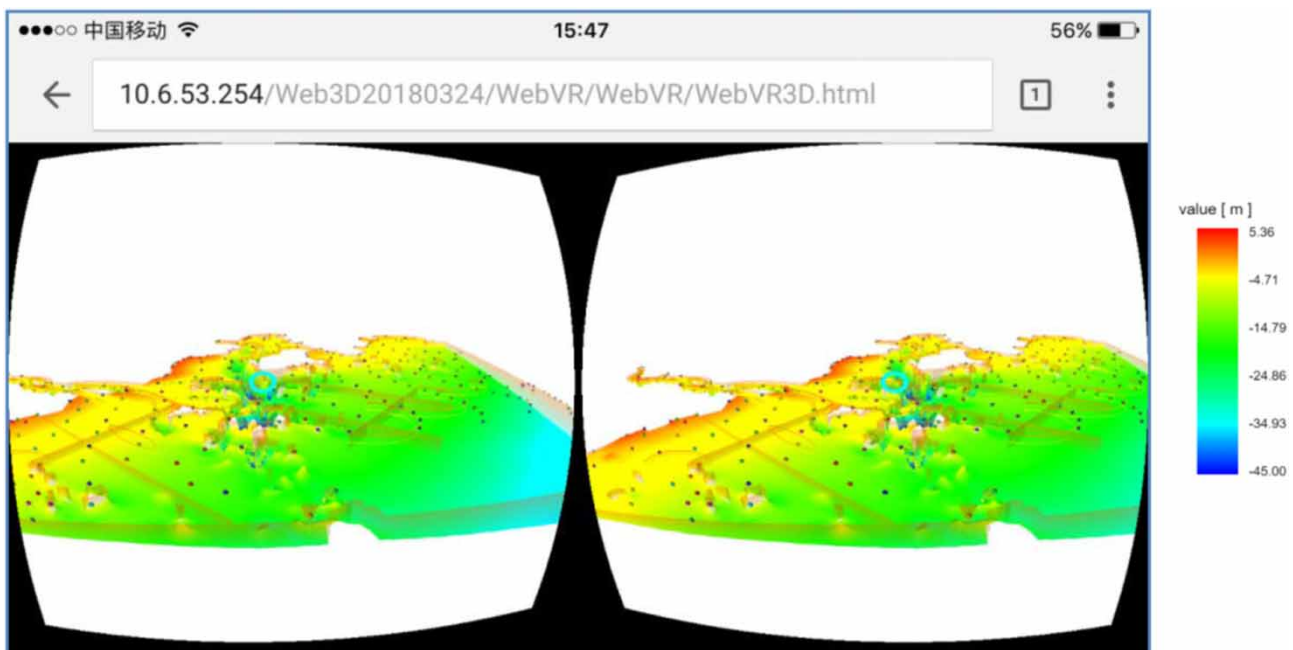
### Engineering application

In order to better verify the feasibility of the method used in this paper, the VR visualization system based on WebVR was applied to the flow field visualization analysis of Oujiang river estuary. The whole computational domain is about 95.3 km from east to west, and 83.5 km from north to south. The scope of numerical simulation is large and the terrain is complex. The three-dimensional topographic map of the Oujiang river estuary is shown in [Figure 10\(a\)](#). The unstructured triangular grid is adopted in the horizontal direction of the model, which makes it possible to better fit the numerous islands in the calculation domain and the irregular shoreline boundaries with complicated twists and turns. The triangular mesh is shown in [Figure 10\(b\)](#). There are 17,241 grid nodes and 32,834 triangular elements. The maximum grid size is 2,000 m, and the minimum grid size is 300 m. The performance of the browser after replacing the three-dimensional spheres with textured styled particles meets practical engineering requirements and the FPS is 53.88. [Figure 10\(c\)](#) shows the height of water surface and the velocity vector field at a given moment. [Figure 10\(d\)](#) reveals the details of the local velocity vector field. [Figure 10\(e\)](#) exhibits a three-dimensional flow field with multiple profiles. It visually shows the position of profiles in the original computational domain and is conducive to the comparative analysis of multiple cross-section data. The three-dimensional visualization of path lines and tracer particles is realized, as shown in [Figure 10\(f\)](#). The three-dimensional visualization results of the Oujiang river estuary in Google Cardboard mode on a mobile device are shown in [Figure 11](#).





**Figure 10** | Three-dimensional visualization results of the Oujiang river estuary in plain Three.js mode shown in a desktop browser: (a) three-dimensional topography, (b) grid, (c) free surface, (d) local flow fields, (e) flow field with multiple profiles, (f) flow field expressed by path lines and trackballs.



**Figure 11** | Three-dimensional visualization results of the Oujiang river estuary in Google Cardboard mode shown on a mobile device.

## CONCLUSIONS

With the development of hardware and web technologies, web browsers can load and render complex three-dimensional scenes in real time. We proposed a new method to display the results of the hydrodynamic model on the internet by using WebVR technology. An immersive and interactive virtual reality three-dimensional scene of flow fields employing HTML5, JavaScript, WebGL, WebVR, and other web-standard technologies was devised and built. It will become an important part of the network computing platform of the hydrodynamic model. When the Lagrange method was used to describe the flow field, the method of replacing three-dimensional spheres with textured styled particles was proposed, which solved the problem that the frame rate of the browser was too low when the number of tracer particles was too large.

Experimental results demonstrate that the new approach achieves the desired effect. Users can browse scenes and objects displayed in a three-dimensional form at anytime and anywhere in the form of webpage, and can perform real-time interaction and roaming operation with the help of the simple HMD, which is assembled by a mobile phone and Google Cardboard. It will be very helpful for developers and users of hydrodynamic models to understand the calculated results more directly and conveniently, and it shows the value of the proposed method in hydrodynamic research.

As future work, we plan to do more research on our web-based visualization system to increase the sense of existence of people with the help of HDM with location tracking, which allows people to walk and approach objects in the scene. Furthermore, more research will also be done to improve the speed of browser loading data and rendering three-dimensional animation.

## ACKNOWLEDGEMENTS

This research is partially supported by the National Natural Science Foundation of China (Grant No. 51739011), the National Key Research and Development Program of China (Grant No. 2016YFC0402707), and the Fundamental Research Funds for the Central Universities (DUT19LAB35). The first author is very grateful to all the

members of the Institute of Hydraulics, Dalian University of Technology, for their selfless help and companionship.

## REFERENCES

- Abbasi, A. & Baroudi, U. 2012 *Immersive environment: an emerging future of telecommunications*. *IEEE MultiMedia* **19** (1), 80–80.
- Anthes, C., Haffeege, A., Heinzlreiter, P. & Volkert, J. 2005 A scalable network architecture for closely coupled collaboration. *Computing and Informatics* **24** (1), 31–51.
- Ashall, L. M., Mulligan, R. P., van Proosdij, D. & Poirier, E. 2016 *Application and validation of a three-dimensional hydrodynamic model of a macrotidal salt marsh*. *Coastal Engineering* **114**, 35–46.
- Boulos, M. N. K., Warren, J., Gong, J. & Yue, P. 2010 *Web GIS in practice VIII: hTML5 and the canvas element for interactive online mapping*. *International Journal of Health Geographics* **9** (1), 14.
- Carey, R., Bell, G. & Marrin, C. 1997 Iso/iec 14772-1: 1997 virtual reality modeling language (vrml97). *The VRML Consortium Incorporated* 34.
- Christen, M., Nebiker, S. & Loesch, B. 2012 *Web-based large-scale 3D-geovisualisation using WebGL: the openWebGlobe project*. *International Journal of 3-D Information Modeling (IJ3DIM)* **1** (3), 16–25.
- Dirksen, J. 2013 *Learning Three.js: the JavaScript 3D Library for WebGL*. Packt Publishing Ltd, Birmingham, UK.
- Finkelstein, S., Nickel, A., Lipps, Z., Barnes, T., Wartell, Z. & Suma, E. A. 2011 *Astrojumper: motivating exercise with an immersive virtual reality exergame*. *Presence: Teleoperators and Virtual Environments* **20** (1), 78–92.
- Freeman, A. M., Jose, F., Roberts, H. H. & Stone, G. W. 2015 *Storm induced hydrodynamics and sediment transport in a coastal Louisiana lake*. *Estuarine, Coastal and Shelf Science* **161**, 65–75.
- Gems, B., Mazzorana, B., Hofer, T., Sturm, M., Gabl, R. & Aufleger, M. 2016 *3D-hydrodynamic modelling of flood impacts on a building and indoor flooding processes*. *Natural Hazards and Earth System Sciences Discussions* **16** (6), 1–32.
- Gracanin, D. & Wright, K. E. 1999 Virtual reality interface for the World Wide Web. In: *Proceedings of the 15th TWENTE Workshop on Language Interactions in Virtual Worlds*, Enschede, The Netherlands, pp. 59–68.
- Hedelin, H. 2017 *Design and Evaluation of A User Interface for A WebVR TV Platform Developed with A-Frame*. Master thesis, Department of Computer Science, Linköping University, Sweden.
- ISO I. IEC 19775-1: 2008 *Information Technology – Computer Graphic, and Image Processing and Environmental Data Representation – Extensible 31)(X3D) – Part 1: Architecture and Base Components*. International Organization for Standardization, Geneva, Switzerland.

- Jin, S. 2013 *HYDROINFO Hydrological Information System*. [http://www.hydroinfo.com.cn/main\\_newForEnglish.html](http://www.hydroinfo.com.cn/main_newForEnglish.html) (accessed 20 July 2018).
- Kang, S.-H. & Jin, S. 2010 *Application of video anaglyph maker for 3-D flow simulation*. *Journal of Hydrodynamics, Ser. B* **22** (2), 289–294.
- Khronos 2009 *WebGL Overview*. <https://www.khronos.org/webgl/> (accessed 7 March 2019).
- Lian, Y., He, L., Ping, J., Zhang, H., Zeng, X., Wang, C. & Chen, L. 2017 Research and implementation on the WEB3D visualization of digital moon based on WebGL. In: *Paper Presented at the Geoscience and Remote Sensing Symposium (IGARSS), 2017 IEEE International*, Fort Worth, TX, USA.
- Liang, S. J. & Molkenhain, F. 2001 *A virtual GIS-based hydrodynamic model system for Tamshui river*. *Journal of Hydroinformatics* **3** (4), 195–202.
- Lv, Z., Li, X., Zhang, B., Wang, W., Zhu, Y., Hu, J. & Feng, S. 2016 *Managing big city information based on WebVRGIS*. *IEEE Access* **4**, 407–415.
- Marcelo, D. P. G., Dias, D. R. C., Mota, J. H., Gnecco, B. B., Durelli, V. H. S. & Trevelin, L. C. 2016 Immersive and interactive virtual reality applications based on 3D web browsers. *Multimedia Tools and Applications* **77** (1), 347–361.
- Mozilla 2018 *A-Frame-Introduction*. <https://aframe.io/docs/0.8.0/introduction/> (accessed 20 July 2018).
- Muhanna, M. A. 2015 *Virtual reality and the CAVE: taxonomy, interaction challenges and research directions*. *Journal of King Saud University-Computer and Information Sciences* **27** (3), 344–361.
- Roberts, J. C., Ritsos, P. D., Badam, S. K., Brodbeck, D., Kennedy, J. & Elmquist, N. 2014 *Visualization beyond the desktop – the next big thing*. *IEEE Computer Graphics and Applications* **34** (6), 26–34.
- Steinbach, E., Hirche, S., Kammerl, J., Vittorias, I. & Chaudhari, R. 2010 *Haptic data compression and communication*. *IEEE Signal Processing Magazine* **28** (1), 87–96.
- Sun, A. Y., Miranda, R. M. & Xu, X. 2015 *Development of multi-metamodels to support surface water quality management and decision making*. *Environmental Earth Sciences* **73** (1), 423–434.
- Thekkedan, M. D., Chin, C. S. & Woo, W. L. 2015 *Virtual reality simulation of fuzzy-logic control during underwater dynamic positioning*. *Journal of Marine Science and Application* **14** (1), 14–24.
- Watson, D. 1981 *Computing the N-dimensional Delaunay Tessellation with application to Voronoy polytopes*. *The Computer Journal* **24**, 167–172.
- Wenisch, P., Van Treeck, C. & Rank, E. 2004 *Interactive indoor air flow analysis using high performance computing and virtual reality techniques*. In: *ROOMVENT 9th International Conference of Air Distribution in Rooms*, Coimbra, Portugal.
- Yin, Z. & Seo, D. 2016 *Analysis of optimum grid determination of water quality model with 3-D hydrodynamic model using environmental fluid dynamics code (EFDC)*. *Environmental Engineering Research* **21** (2), 171–179.
- Yin, L., Zhu, J., Zhang, X., Li, Y., Wang, J., Zhang, H. & Yang, X. 2015 *Visual analysis and simulation of dam-break flood spatiotemporal process in a network environment*. *Environmental Earth Sciences* **74** (10), 7133–7146.
- Zhang, X. F., Yicheng, J., Yong, Y. & Zhihua, L. 2004 *Ship simulation using virtual reality technique*. In: *Proceedings of the 2004 ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Applications in Industry*, Singapore.
- Zhang, N., Jin, S. & Ai, C. F. 2016 *Study of three dimensional flow field visualization based on WPF*. *Journal of Waterway and Harbor* **37** (4), 455–460.
- Zhang, N., Jin, S., Ai, C. F. & Ding, W. Y. 2018 *An integrated water-conveyance system based on Web GIS*. *Journal of Hydroinformatics* **20**(3), 668–686.

First received 14 September 2018; accepted in revised form 17 May 2019. Available online 13 July 2019