International Journal of
Geo-Information

MDPI

Article

Flow Modeling and Rendering to Support 3D River Shipping Based on Cross-Sectional Observation Data

Xuequan Zhang [1], Jin Liu [2,*], Zihe Hu [3] and Ming Zhong [4,5,6]

[1] School of Resource and Environmental Sciences, Wuhan University, Wuhan 430079, China; brucezhang@whu.edu.cn
[2] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University, Wuhan 430079, China
[3] GNSS Research Center, Wuhan University, Wuhan 430079, China; huzihe@whu.edu.cn
[4] Intelligent Transport Systems Research Center, Wuhan University of Technology, Wuhan 430063, China; mzhong@whut.edu.cn
[5] National Engineering Research Center for Water Transport Safety, Wuhan 430063, China
[6] Engineering Research Center for Transportation Safety, Ministry of Education, Wuhan 430063, China
* Correspondence: JLiu@sgg.whu.edu.cn; Tel.: +86-133-1695-6666

Received: 7 February 2020; Accepted: 7 March 2020; Published: 9 March 2020

Abstract: The flow in meandering rivers is characterized by rapid changes in flow velocity and water level, especially in flooded environments. Accurate cross-sectional observation data enable continuous monitoring of flow conditions, which is important for river navigation. In this paper, cross-sectional data based flow modeling and rendering methods are studied to build an interactive hybrid flow environment for three-dimensional river shipping. First, the sparse cross-sectional data are extrapolated and interpolated to provide dense sampling points. Then, the data are visualized separately by dynamic texture mapping, particle tracking, streamline rendering, and contour surface rendering. Finally, the rendering models are integrated with ship animation to build a comprehensive hybrid river navigation scenario. The proposed methods are tested by visualizing measured cross-sectional data in the Yangtze River using an open-source software, called World Wind. The experimental results demonstrate that the hybrid flow rendering achieves comprehensive visual effect and the rendering frame rate is greater than 30. The interactive hybrid flow visualization is beneficial to support river shipping analysis.

Keywords: cross-sectional data; data processing; flow modeling; flow rendering; river shipping

1. Introduction

According to the recent World Meteorological Organization (WMO) report [1], flood risks are expected to increase in the coming decades, along with higher intensities, and greater damages. The increasing flood risk in rivers is associated with many global-scale factors, such as climate changes and channel regulations in upper courses. River flooding leads to rapid changes in flow conditions, which greatly affects river shipping safety. Improved Earth observation techniques, such as satellite images [2], Unmanned Aerial Vehicle (UAV) images [3] and Light Detection and Ranging (LIDAR) point clouds [4], have greatly enhanced the ability to monitor the overall state of rivers. However, the resulting data are quite limited by the nature of the measuring equipment, which can make continuous observation difficult. In addition, they also have limitations in terms of measuring terrain under water [5]. Traditional bathymetric surveys and Sound Navigation And Ranging (SONAR) techniques [6] are alternative solutions for obtaining dynamic measured river data over time. These data are typically observed by ground-based surveying based on transverse profiles, also known as cross-sections, at locations selected to capture salient features of the water flow [7].

However, the cross-sectional data thus obtained are often sparse and discrete, so appropriate data processing must be implemented to satisfy the requirements of flow modeling and simulation. Lazaro et al. [8] processed river bed bathymetry based on ArcGIS, introducing different spatial interpolating methods to evaluate anisotropic effects [9]. Flanagin et al. [10] and Schappi et al. [11] adopted Kochanek–Bartels splines and a bilinear techniques, respectively, to reproduce river bed terrain from cross sections. Caviedes-Voullieme et al. [12] implemented the Cubic Hermite Spline (CHS) method to build continuous and smooth river channel topography from cross-sectional data. Zhang et al. [13] used the shortest temporal distance to interpolate the river topography, which achieved good performance. Chen et al. [14] studied and compared three cross-sectional interpolation methods, including linear interpolation, inverse distance weighting (IDW), and natural neighbor (NN), to model river stages. In the actual situation, the measuring equipment is not uniformly feasible everywhere, for example, the river channel terrain cannot be observed in very shallow water. In addition, the river trend is complex, and it may not be monotonically increasing but meandering. As a result, existing processing methods should be improved to calculate accurate dense sampling points for meandering rivers.

River ship navigation is associated with a variety of information about flow conditions, such as flow velocity, water depth, river boundaries, and vortices. Flow visualization is important to illustrate the dynamic flow distribution and elucidate the laws of water motion, which are very important for safe river shipping [15,16]. A variety of available visualization methods can express flow information comprehensively and help to build a hybrid flow environment. Flow rendering techniques can be divided into two categories: scientific flow visualization and visual flow effects. Scientific flow visualization is used to show the distribution of various data, such as water level, water depth, flow velocity, and pollutant concentration. Particle tracking, streamline rendering, and contour surface rendering are most commonly used to present flow conditions [17–21]. Visual flow effects are used to simulate water surface changes using computer graphics technology, enhancing the sense of reality in a virtual geographic environment [22–26]. Dynamic texture mapping is commonly used to simulate the water surface. However, most existing flow rendering methods are designed for large-scale water, such as marine environments. In contrast, river flow, especially in flooded environments, is characterized by meandering shape of the river way and quick variation of the flow state. As a result, the existing visualization methods should be improved for rivers. Meanwhile, most of the existing visualization models can only display flow information in one aspect [19], users need to constantly switch models to comprehensively analyze river status. To quickly display all the information in a scene, hybrid rendering needs to be studied and optimized for effective flow rendering of meandering rivers.

In this paper, the challenges are addressed to realize flow modeling and rendering for three-dimensional (3D) river shipping in a virtual globe. All the data processing, modeling, and rendering techniques are designed and optimized for meandering river flow. First, cross-sectional data are extrapolated and interpolated spatiotemporally to provide continuous dense sampling points. Then, these data are visualized by flow simulation and visualization methods. To improve rendering efficiency, the Levels of Detail (LOD) strategy and Graphics Processing Unit (GPU) acceleration are applied to the rendering models. The flow surface is visualized by mesh construction and dynamic texture mapping, which simulates the real flow scene. Particle tracking, streamline rendering, and contour surface rendering are optimized to present detailed flow variation and distribution. Finally, the visualization models are combined together to build an interactive hybrid flow environment for ship navigation. Using the methods presented in this paper, the cross-sectional data are processed and visualized to provide continuous comprehensive flow information for 3D river shipping.

2. System Framework

Figure 1 shows the framework developed in this study. It consists of three main components: data processing, flow modeling and rendering, and river shipping. The original data is the cross-sectional observation along the river, which is obtained by SONAR technology. In the data processing stage, the waterway cross-sectional data, in conjunction with Digital Elevation Model (DEM) data, are first extrapolated and then interpolated. The cross-sectional extrapolation extends the missing data in shallow water and fits the surrounding terrain. The water surface is spatially resampled both in the transverse and longitudinal directions for detailed modeling. Temporal interpolation is applied to smooth the animation between two key frames. Through data processing, accurate river boundaries and dense sampling points are acquired.

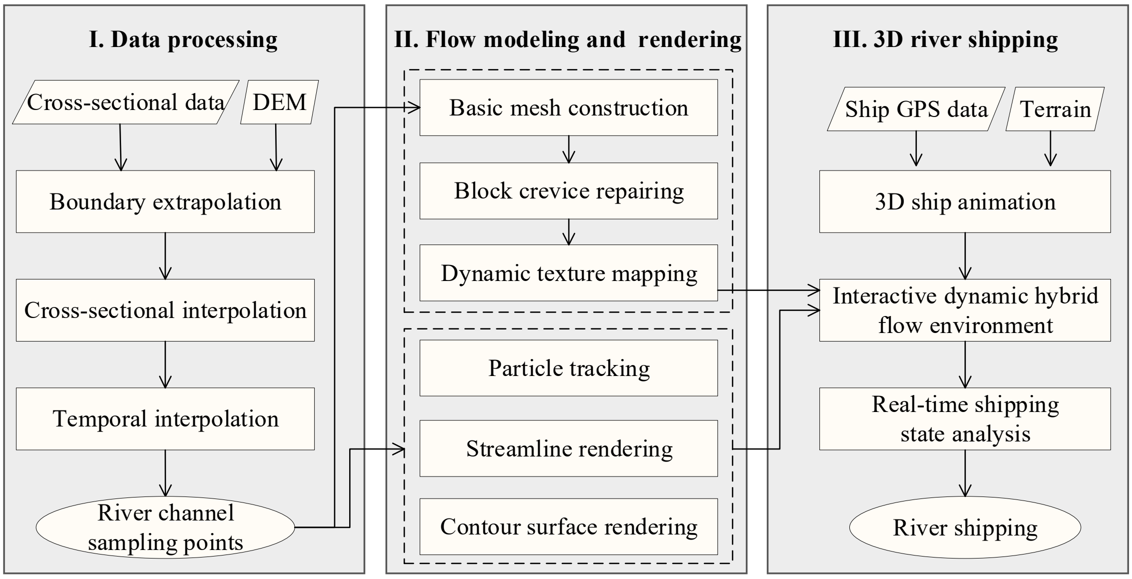


Figure 1. 3D river shipping framework.

The flow modeling and rendering stage includes flow simulation and flow visualization. The water surface is dynamically constructed by basic mesh and fusion zones, which avoids cracks between quadrangle parts. Then, the flow surface is dynamically mapped by water texture with real flow velocity. The particle tracking, streamline rendering, and contour surface rendering displays the specific flow conditions in points, lines, and surfaces separately.

Finally, the different flow rendering models are integrated with ship Global Positioning System (GPS) animation to support 3D river navigation. The hybrid rendering strategy enhances the information displayed in a single scene, which is important for safe shipping. In addition, real-time ship route and flow state information can be analyzed to support decision-making.

3. Methodology

3.1. Study Area and Data

The Yangtze River, which is the longest river in Asia, is a crucial waterway for river transportation in China [13]. It links 11 provinces from the East China Sea to Sichuan and has a complex navigational environment. Because the water stage of the Yangtze River varies greatly between the dry season and the rainy season, a flood scenario is chosen to demonstrate the method. The experimental data are real cross-sectional observation data of a river flood in 2009 (Figure 2), which comes from the Changjiang Water Resources Commission. It is caused by the continuous heavy rain and the river stage varies greatly, which is suitable for studying the flow field. The experimental area is a segment of the main channel about 35 km in length extending from Shaba Bay to Songzikou, Zhijiang City. The specific data parameters are shown in Table 1. The surveyed dataset forms 248 × 233 × 30 dynamic surfaces along the river, and the data variable type includes flow velocity and water depth. The time range is from 8:00 6 September to 14:00 7 September, the time step is 1 h, so the time series can be obtained by the start time and time step. In addition, as cross-sectional measurement is not feasible to record the data in very shallow water, the real water boundaries of the cross-sections should be extrapolated based on

DEM data with a higher resolution. DEM data with a resolution of 5 m are adopted to process the cross-sectional data.

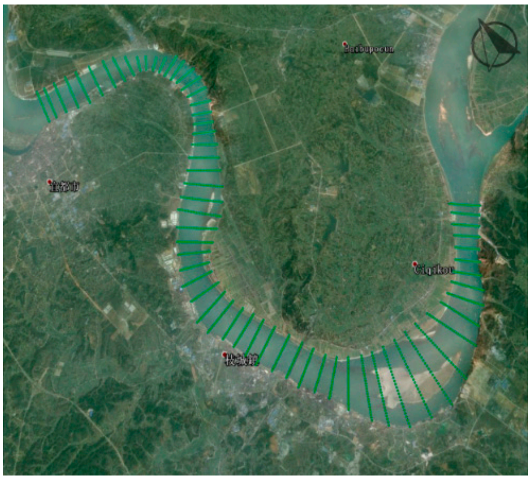


Figure 2. Cross-sectional data distribution.

Table 1. Summary of the river flood data.

Name	Data
Dimensions	248 × 233 × 1 × 30
Latitudinal range	30.2468 °N–30.4284 °N
Longitudinal range	111.4626 °W–111.6344 °W
Time range	8:00 6 September–14:00 7 September
Time step	1 h
Variable type	Flow velocity, water depth
Data volume	2 GB

3.2. Data Processing

Surveyed cross-sectional data are usually sparse in the transverse direction at each cross-section as well as in the longitudinal direction along the river channel [14]. As river flow simulation and visualization require a large amount of topographic data, resampling of the original cross-sectional data is necessary. Processing of the cross-sectional data includes boundary extrapolation, cross-sectional interpolation, and temporal interpolation.

3.2.1. Boundary Extrapolation

With the change of seasons, the water level of the river also changes. Cross-sectional measurement is not feasible to record the data in very shallow water. Observation data are usually inconsistent with the real river boundaries. As a result, the real water boundaries of the cross-sections should be extrapolated based on DEM data. The boundary extrapolation of cross-sectional data is shown in Figure 3. $\{Ci|i = 1, 2, \cdots, m\}$ denotes the waterway cross-sectional observation data, and $\left\{P_{i,\, j}\middle|j = 1, 2, \cdots, n\right\}$ denotes the observational points in section C_i.

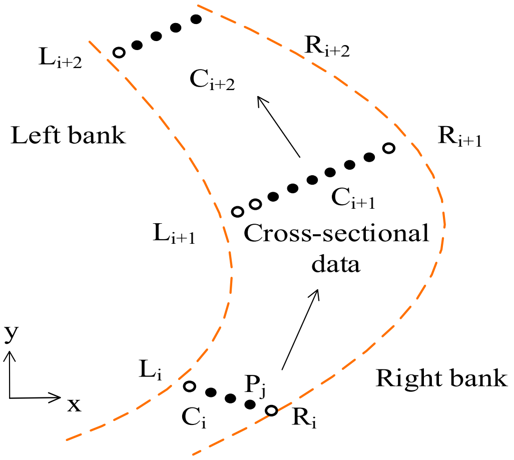


Figure 3. Boundary extrapolation of cross-sectional data.

The position and value of each added point in the cross-section are calculated as follows:

(1) The center point of the cross-section $P_{i,\,k}$ is determined, and a search is performed along $\overrightarrow{P_{i,\,k}P_{i,\,1}}$ with step d.

(2) The terrain elevation of the point is obtained, and the difference value Δh between the water level and the terrain altitude is computed.

(3) If Δh is less than a threshold H (for example 0.1 m), the searching point is taken as the boundary point. For the Δh values of two adjacent sampling points, if one is positive and the other is negative, and both the Δh values are bigger than H, the dichotomy algorithm [27] is adopted to determine the boundary point; otherwise, the point is added as a new sampling point, and the process is repeated from step (2) to sample the next point.

(4) Similarly, another endpoint along $\overrightarrow{P_{i,\,k}P_{i,\,\mathrm{n}}}$ is searched with step d. $\{(Li, Ri)\big| i = 1, 2, \cdots, m\}$ denotes the boundary points of the cross-section data.

(5) The positions of the added sampling points are calculated. Once the plane coordinates are acquired, the Bursa–Wolf seven parameters transformation [28] is applied to compute the coordinates in the Cartesian coordinate system. Then, the positions in the spherical coordinate system is:

$$\begin{cases} longitude = arctan(\frac{y}{x}) \\ latitude = arcsin(\frac{z}{\sqrt{x^2+y^2+z^2}}) \\ altitude = \sqrt{x^2+y^2+z^2} - R \end{cases} \tag{1}$$

(6) The flow velocity and water depth of the added sampling point are calculated. The velocity of the boundary point is regarded as zero, and the water levels of the sampling points in a cross-section are the same. Linear interpolation is adopted for the calculation:

$$f(x, y) = f(x_i, y_i) + \frac{f(x_{i+1}, y_{i+1}) - f(x_i, y_i)}{d_t} d_i \tag{2}$$

where $f(x, y)$ is the estimated value of the added point, $f(x_i, y_i)$ and $f(x_{i+1}, y_{i+1})$ are the adjacent sampling points of the added point, d_i is the distance between (x, y) and (x_i, y_i), and d_t is the distance between (x_i, y_i) and (x_{i+1}, y_{i+1}). The reliability can be evaluated by interpolation error [29].

3.2.2. Cross-Sectional Interpolation

As the measured cross-sectional data are usually sparse along the flow direction, they must be interpolated for detailed visualization. For a meandering river, the adjacent cross-sections are usually not parallel. Consequently, the sampling cross-sections are computed by equal angle distribution (Figure 4).

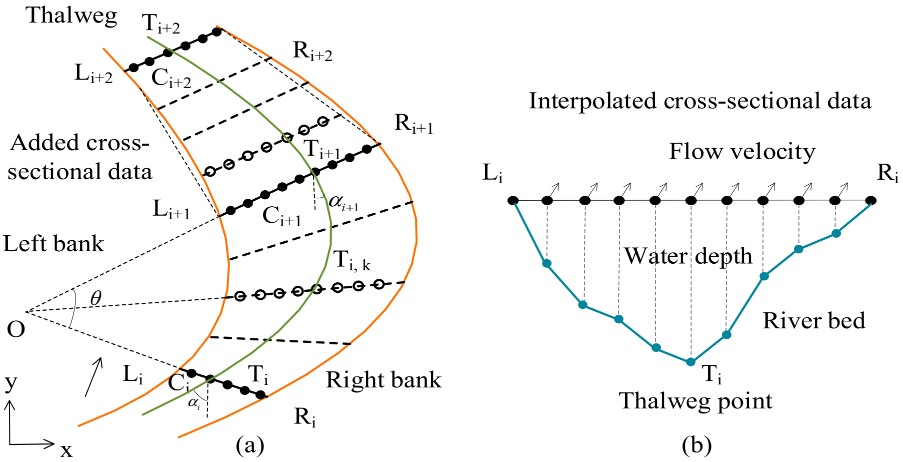


Figure 4. Cross-sectional interpolation: (**a**) position calculation; (**b**) value calculation (flow velocity and water depth).

The specific cross-sectional interpolation steps include:

(1) The thalweg of the river is constructed based on cross-sectional data $\{Ci|i = 1, 2, \cdots, m\}$. The thalweg point in each cross-section is defined by calculating the lowest point, described as $\{Ti|i = 1, 2, \cdots, m\}$. (x_i, y_i) denotes the position of point T_i, (x_{i+1}, y_{i+1}) denotes the position of point T_{i+1}. The *k-th* interpolated thalweg point $T_{i,k}$ is calculated using Quartic Hermite Spline with Parameter (QHSP) [30]:

$$y_{i,k} = a_0 y_i + a_1 y_{i+1} + b_0 (x_{i+1} - x_i) cot\alpha_i + b_1 (x_{i+1} - x_i) cot\alpha_{i+1} \tag{3}$$

where $\cot \alpha_i$ and $\cot \alpha_{i+1}$ are the tangents of the flow direction at the cross sections C_i and C_{i+1}. The coefficients can be computed as follows:

$$\begin{cases} a_0 = \mu\xi^4 - 2(\mu - 1)\xi^3 + (\mu - 3)\xi^2 + 1 \\ a_1 = -\mu\xi^4 + 2(\mu - 1)\xi^3 - (\mu - 3)\xi^2 \\ b_0 = \mu\xi^4 - (2\mu - 1)\xi^3 + (\mu - 2)\xi^2 + \xi \\ b_1 = -\mu\xi^4 - (2\mu + 1)\xi^3 - (\mu - 3)\xi^2 \end{cases} \tag{4}$$

where the parameter ξ is determined by the change in the x-coordinate:

$$\xi = \frac{x_{i,k} - x_i}{x_{i+1} - x_i} \tag{5}$$

(2) The angle θ between cross-section C_i and C_{i+1} is calculated as follows:

$$\theta = \arccos\left(\frac{\overrightarrow{C_i} * \overrightarrow{C_{i+1}}}{\left|\overrightarrow{C_i}\right| * \left|\overrightarrow{C_{i+1}}\right|}\right) \tag{6}$$

(3) As shown in Figure 4a, if the angle θ is smaller than a given threshold (for example 5°), the cross-sections are regarded as paralleled lines, and resampling is performed with equal distance distribution. For example, for sections C_{i+1} and C_{i+2}, lines $L_{i+1}L_{i+2}$ and $R_{i+1}R_{i+2}$ are connected and divided into m segments according to the distance $L_{i+1}L_{i+2}$. The corresponding division points are connected and the intersection points between those points and the thalweg are computed.

(4) Otherwise, if the angle θ is bigger than the threshold, for example, for sections C_i and C_{i+1}, they are resampled by equal angle distribution. The direction of the *k-th* sampled cross-section can be

calculated. The intersection points between the thalweg and the cross-section vector are computed by equiangular interpolation [31]:

$$\overrightarrow{OT_i} * \overrightarrow{OT_{i,k}} = \left|\overrightarrow{OT_i}\right| * \left|\overrightarrow{OT_{i,k}}\right| * cos(\frac{k}{s} * \theta)\ k \in [1,\ s] \tag{7}$$

where s is the number of the added cross-section between C_i and C_{i+1}.

(5)The boundaries of the added cross-section are extrapolated based on Section 3.2.1, and riverbank lines are constructed based on the QHSP method defined in Formula (3).

(6) The flow velocity and water depth of the added sampling points (Figure 4b) are calculated. The linear interpolation method defined in Formula (2) is applied.

3.2.3. Temporal Interpolation

The cross-sectional observation data are recorded at equal time-intervals, so key frame animation techniques is applied to simulate the dynamic flow [32]. To smooth the flow animation, the linear interpolation is adopted to calculate data in two frames. The values of a point on the river surface are denoted as $value_n$ and $value_{n+1}$ at the time of t_n and t_{n+1}, respectively. In addition, t_x is a specific time between t_n and t_{n+1}. The value of the point at time t_x is $value_x$, which can be calculated as:

$$value_x = value_n + (value_{n+1} - value_n) * \frac{t_x - t_n}{t_{n+1} - t_n} \tag{8}$$

3.3. Flow Simulation

The flow regime of the water in a river is different from the static water in an ocean or lake. As a river is long and narrow, and its elevation varies along the channel, the water flow differs at different locations. To simulate the dynamic change of water flow in the river, dynamic texture mapping with real flow velocity is applied.

3.3.1. Basic Mesh Construction

Because a long river may span several kilometers, rendering the entire flow dataset directly inevitably leads to low efficiency. Therefore, subsection and LOD technologies are applied to build river surface with multi-resolution, which improve the rendering efficiency. The key point of constructing LOD based mesh is to determine subsection size based on the camera state. As the quad-tree tile pyramid method for terrain scheduling in the virtual globe has considered the influence of viewing distance and vertical scaling for tile subdivision [33], adopting the terrain tile computing result is beneficial for reducing the amount of LOD calculations.

As shown in Figure 5, the river is first divided into many sections of equal length. As the visual range can be calculated based on the camera pose, the river sections in viewing extent are filtered and loaded. Corresponding to the terrain pyramid level, the cross-sectional data are filtered to construct the mesh surface. When the terrain pyramid level is small, the selected cross-sectional data are sparse; otherwise, the selected cross-sectional data are dense. Finally, the river surface mesh is built by the Delaunay algorithm [34].

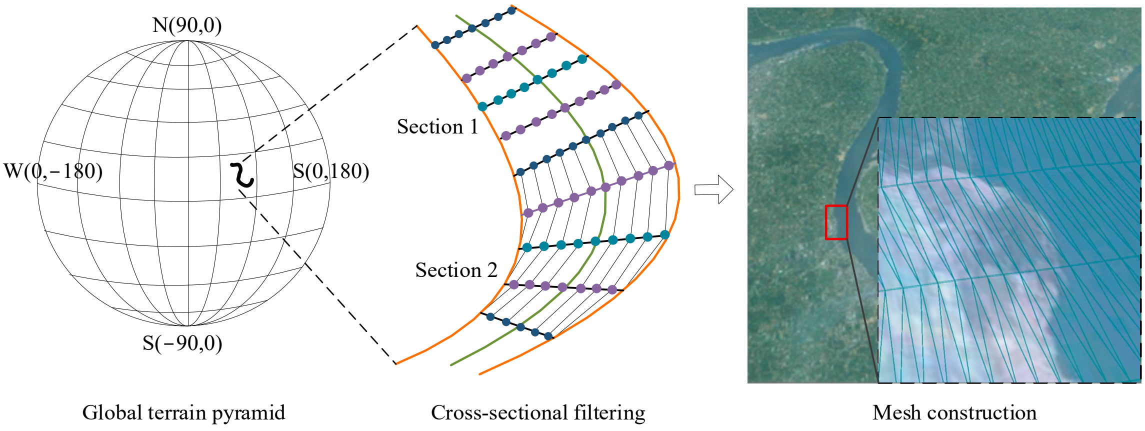


Figure 5. LOD-based surface mesh construction.

3.3.2. Block Crevice Repairing

Because the water flow velocity is different at the river surface, dynamic texture mapping on a single-layer mesh leads to crevices or unnatural connections. To achieve smooth flow effects, a three-layer water surface mesh construction method is proposed. The key to repair the crevices between different blocks is to establish vertical and horizontal fusion zones, which compensate for cracks at the edges of blocks with different speeds and opacities, and achieve smooth transitions.

As shown in Figure 6a, in the first-layer, the water flow velocity in the block *ABCD* is treated as the same. As the water flow velocities in the blocks *ABCD* and *ACGH* are different, a vertical fusion zone ACP_1P_2 is created to smooth the connection (Figure 6b). Similarly, a horizontal fusion zone CDQ_1Q_2 is created to smooth the connection between blocks *ABCD* and *CDEF* (Figure 6c). The point P_1 and Q_1 can be calculated as:

$$\begin{cases} \overrightarrow{AP_1} = r * \overrightarrow{BA}\ \mathrm{r} \in [0,1] \\ \overrightarrow{CQ_1} = t * \overrightarrow{AC}\ \mathrm{t} \in [0,1] \end{cases} \tag{9}$$

where the coefficients *r* and *t* can be adjusted according to the visual effects.

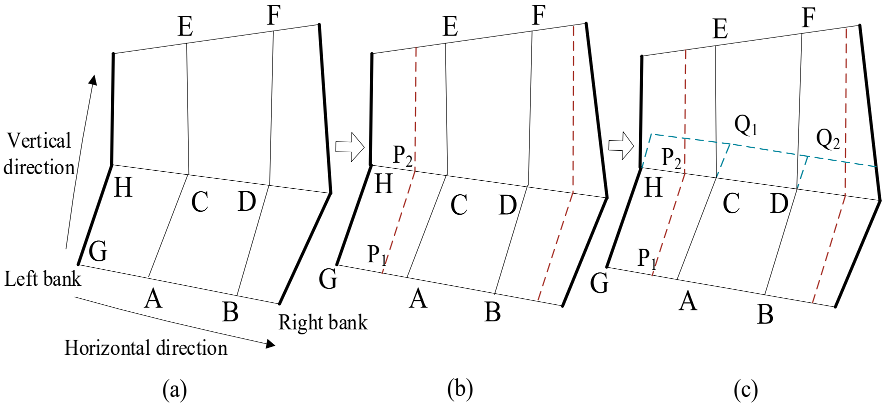
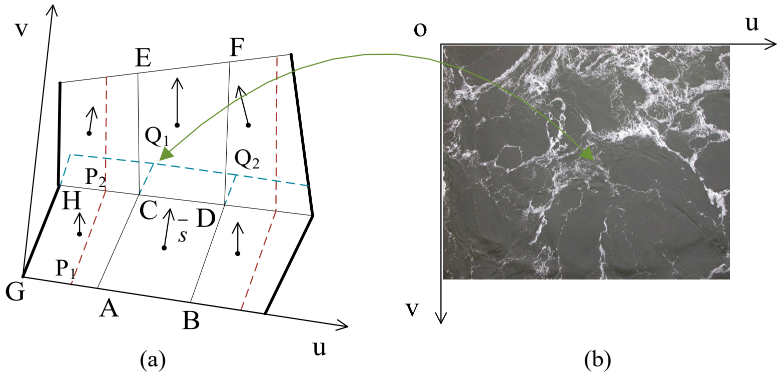


Figure 6. Three-layer water surface construction: (**a**) basic surface division; (**b**) vertical fusion zone; (**c**) horizontal fusion zone.

3.3.3. Dynamic Texture Mapping

To simulate real flow effects, the water material is applied for dynamic texture mapping based on the surface mesh (Figure 7). As mentioned above, the adjacent block textures should be fused by the velocity interpolation.



(a) (b)

Figure 7. Dynamic texture mapping: (**a**) water surface mesh; (**b**) texture mapping.

The specific texture coordinate calculation includes four parts:

(1) Static texture coordinate calculation. The texture coordinates of each vertex are calculated to map water texture on the river surface mesh. Here, *(U, V)* denotes the texture coordinates, where the *U* axis is along the cross-section, and the *V* axis is along the main flow direction and perpendicular to the cross-section. The texture coordinates of the vertex P_j in cross-section C_i can be calculated based on the cross-section and the thalweg:

$$\begin{cases} U_{(i,\,j)} = \frac{\left|\overrightarrow{P_{j-1}P_j}\right|}{d} * j \\ V_{(i,\,j)} = V_{(i-1,\,j)} + \frac{\left|\overrightarrow{T_{i-1}T_i}\right|}{d} \end{cases} \tag{10}$$

where d is the basic tile length between two sampling points, $\left|\overrightarrow{P_{j-1}P_j}\right|$ is the length of the cross-sectional points $P_{j\text{-}1}$ and P_j and $\left|\overrightarrow{T_{i-1}T_i}\right|$ is the curve length of the thalweg points $T_{i\text{-}1}$ and T_i.

For the vertex P_1 in the vertical fusion zone ACP_1P_2, the *V* coordinate has the same value as vertex *A*. For the vertex Q_1 in the horizontal fusion zone *CDQ1Q2*, the *U* coordinate has the same value as vertex *A*. Thus, the *U* coordinate of vertex P_1 and the *V* coordinate of vertex Q_1 can be acquired:

$$\begin{cases} U_{P_1} = U_A - \frac{\left|\overrightarrow{AP_1}\right|}{d} \\ V_{Q_1} = V_A + \frac{\left|\overrightarrow{CQ_1}\right|}{d} \end{cases} \tag{11}$$

(2) Dynamic texture movement. To achieve effective flow simulation, the moving speed of the water texture is calculated based on the measured flow velocity for each block separately. The *U* coordinate stays the same and the *V* coordinate is adjusted by adding an offset value. The moving speed of each block is calculated based on the average speed of the four vertices. The *V* coordinate offset value can be computed as:

$$\Delta V = \frac{\bar{s}}{d} * m \tag{12}$$

where $\bar{s}$ is the moving speed of the block, and m is the flow velocity exaggeration factor.

(3) Texture fusion. As the water flows, each block moves at an inconsistent speed, causing the textures to be pulled and misaligned. To achieve smooth transitions between blocks, the texture coordinates of the horizontal and vertical zones must be updated adequately. For the block *ABCD*, the moving speeds of fusion zones ACP_1P_2 and CDQ_1Q_2 are separately set to be average speeds of *ABCD* and adjacent blocks separately. Meanwhile, the opacities of the vertices in the fusion zone need to be varied gradually to blend them together. The opacities of vertices P_1 and P_2 are set to be 0, and the opacities of vertex *A* and *C* are set to be 1. In this case, the shading program automatically implements linear gradient processing from opaque to completely transparent between *AC* and P_1P_2, such that the basic block and the fusion zones are smoothly merged into a large texture block. Similarly, the block *ABCD* and fusion zone CDQ_1Q_2 are merged together.

(4) Texture rendering. To simulate a real water flow scene, light must be considered [26]. The water surface color is calculated by High Level Shader Language (HLSL) programming as follows:

$$C_{water} = \alpha * C_{picture} + \beta * C_{ambient} + \gamma * C_{diffuse} + \varphi * C_{specular} + \delta * C_{reflect} \tag{13}$$

where $C_{picture}$ is the texture color value, $C_{ambient}$ is the ambient color, $C_{diffuse}$ is the diffuse color, $C_{specular}$ is the specular color and $C_{reflect}$ is the reflected color. The values of the coefficients α, β, γ, φ and δ lie between 0 and 1, and they can be interactively modified according to the weather conditions.

3.4. Flow Visualization

For flow visualization, particle tracking, streamline rendering, and contour surface rendering are adopted to show the water flow information in detail.

3.4.1. Particle Tracking

Particle tracking is a well-known technique for the determination of velocity vectors according to the local velocity observations in a flow field [20]. Cellular Automaton (CA) is adopted to simulate the spatio-temporal interactions between cells [35]. The CA-based particle tracking mainly involves two objects. One is the particle, the speed and direction of each particle is updated. The other is the cell; the state of the cell is characterized by the number of the particles. If a particle in the neighborhood moves into the cell, the number of elements in the cell is increased by 1. If the particle leaves the cell or its velocity becomes 0, the number of elements in the cell is decreased by 1. The integral equation of particle motion is described as:

$$\begin{cases} x(t+\Delta t) = x(t) + \int_t^{t+\Delta t} v_x dt \\ y(t+\Delta t) = y(t) + \int_t^{t+\Delta t} v_y dt \end{cases} \tag{14}$$

The trajectory of a particle is usually represented as an arrow marker or ribbon trail. Fixed-sized particle visualization leads to crossover for a large-scale view (Figure 8a), whereas for small-scale view, particles are too small to be observed. As a result, LOD technology is applied to compute the cell size and the trajectory length (Figure 8b). Both the cell size and the trajectory length are inversely proportional to the pyramid level of the virtual globe.

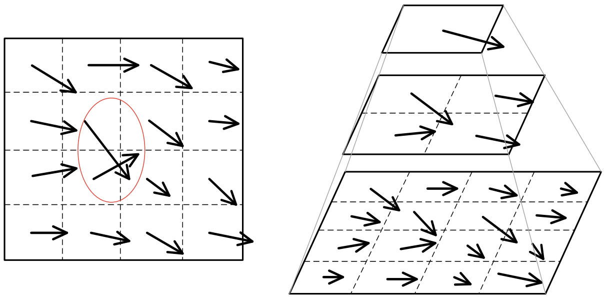

(a) (b)

Figure 8. Particle tracking visualization: (**a**) fixed-size visual effect; (**b**) LOD-based visual effect.

3.4.2. Streamline Rendering

Streamlines are water flow curves, which provide a dense visualization result with many flow characteristics, such as vortices, sources, and sinks [36]. In streamline visualization, a certain number of seeds are placed on the canvas, and the curves are drawn from each seeding location. Every point along a curve is tangent to the instantaneous local velocity vector. To improve the rendering efficiency and local details, an adaptive streamline construction method is applied.

As shown in Figure 9, the velocity of the seed point P_i is $\overrightarrow{V(P_i)}$ and the velocity is $\overrightarrow{V(P_{i+1})}$ in the next frame, θ is the deflection angle, λ is the step factor associated with the camera pose, μ is the angle factor ($\mu > 1$), $\overline{v}$ is the ratio of the velocity to the maximum velocity in the field ($\overline{v} \in [0,1]$), D_0 is the basic fixed step, k is the terrain pyramid level corresponding to step D_0, $level \in [L_i, L_j]$ is current terrain pyramid level corresponding to the tile of the streamlines, and m is the viewing distance factor, so the adaptive step D_i can be calculated as follows:

$$\begin{cases} \lambda = (m*2)^{k-level} \\ \theta = arccos\left(\frac{\overrightarrow{V(P_i)}*\overrightarrow{V(P_{i+1})}}{\left|\overrightarrow{V(P_i)}\right|*\left|\overrightarrow{V(P_{i+1})}\right|}\right) \\ D_i = \lambda * D_0 * \left(1 + \overline{v} * \mu^{1-(1-cos(\theta))^2}\right) \end{cases} \tag{15}$$

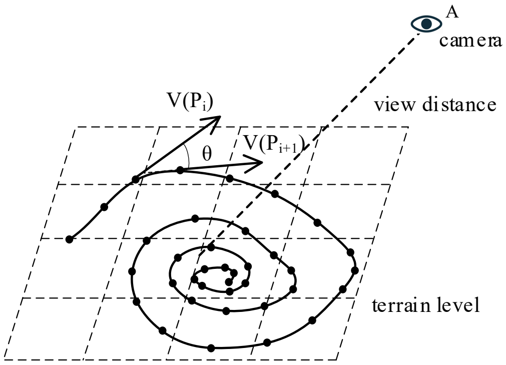


Figure 9. Adaptive streamline rendering.

The algorithm considers both the local flow condition and the global viewing distance. It is proportional to the current velocity and the viewing distance, but inversely proportional to the flow deflection angle. As the camera zooms in, the factor λ is reduced, resulting in the emergence of more visual details. When *level* is smaller than L_i or bigger than L_j, λ keeps the same. As the deflection angle increases, the sampling step decreases exponentially. At the same time, the algorithm avoids the situation whereby the velocity is small, but the deflection angle is large, which causes the step length to be excessive. LOD technology is also applied to control the streamline density.

3.4.3. Contour Surface Rendering

Contour surfaces are based on contour lines, which connect nodes with the same value. Contour surfaces are useful to help users to understand the scalar spatial variation and distribution of values, such as in the display of water depth and level, water surface slope, and pollutant concentrations [16]. The contour surface rendering includes contour line construction and classification coloring.

Assuming that two vertices of a cell are denoted P_1 and P_2 with values $P_1.value$ and $P_2.value$, the current contour line value is *value*. Therefore, the equivalent value can be judged by $f = (P_1.value - value) * (\ P_2.value - value)$. If $f > 0$, no equivalent values between the two points exist; if f < 0, for example, as shown in Figure 10a, the point P with the equivalent value can be calculated by Formula (16); if $f = 0$, the contour line crosses one of the vertices (Figure 10b). In this case, a very small offset value, for example 0.0001, is given to the vertices to apply the algorithm of the second case, which reduces the calculation difficulty.

$$\begin{cases} P.x = P_1.x + (value - P_1.value) * \frac{P_2.x - P_1.x}{P_2.value - P_1.value} \\ P.y = P_1.y + (value - P_1.value) * \frac{P_2.y - P_1.y}{P_2.value - P_1.value} \end{cases} \tag{16}$$

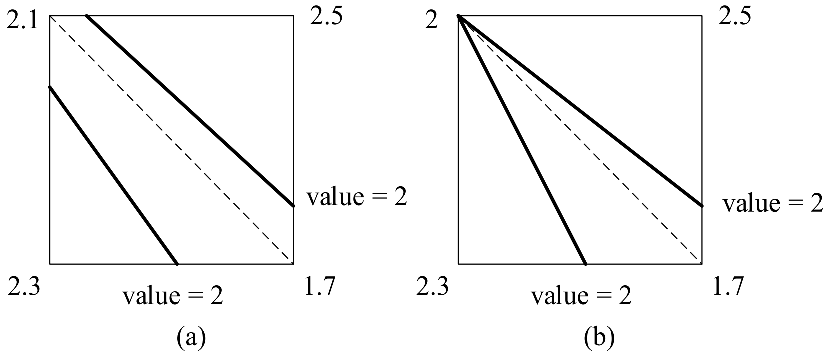


Figure 10. Equivalent point calculation: (**a**) the contour line passes through the two vertices; (**b**) the contour line crosses one of the vertices.

4. Implementation and Discussion

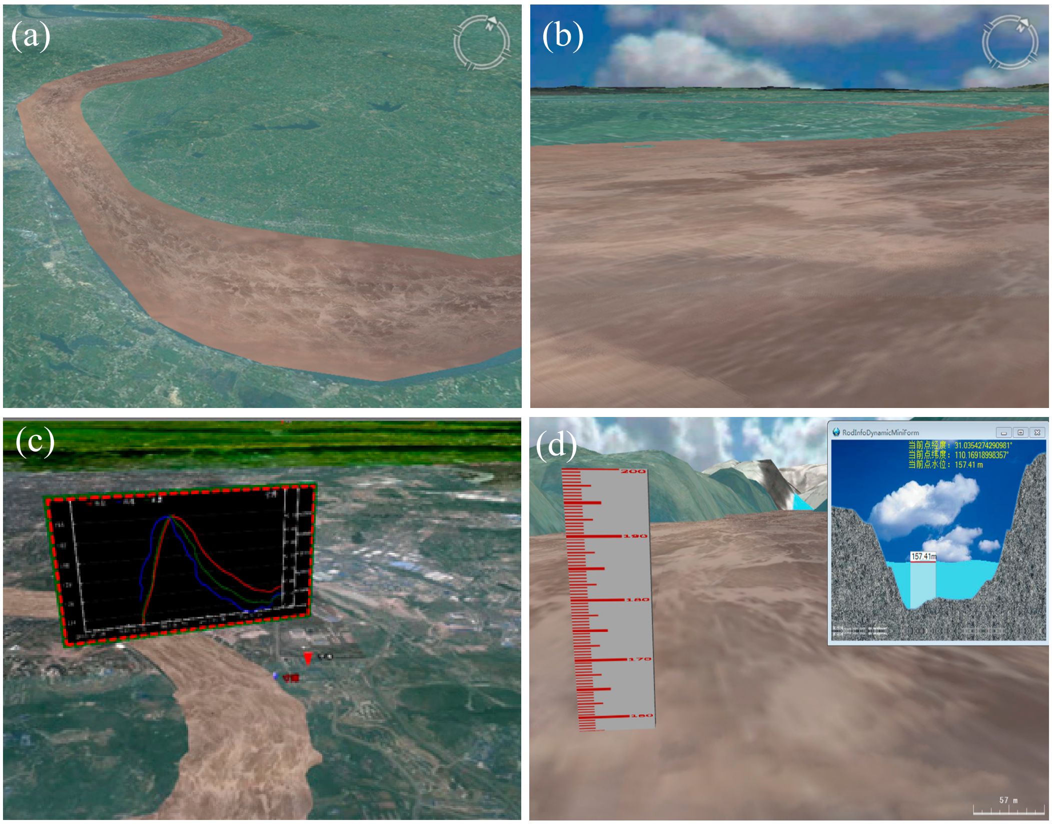
The experimental environment is in the 64-bit Windows 10 operating system. The CPU is a 2.7 GHz quad-core Intel Core i5-6400 CPU with 4 GB of RAM, and the graphics card is NVIDIA GeForce GT 730 with 2 GB of RAM. The flow simulation and visualization methods presented in this paper were implemented in World Wind (.NET version) [37] using Direct3D, and the rendering algorithms were written in HLSL 3.0.

4.1. Performance

Based on the method proposed in this paper, the cross-sectional observation data are first processed by boundary extrapolation, and then interpolated in both spatial and temporal dimensions. Through data processing, this method provides high-resolution and continuous sampling points for flow simulation and visualization. The water surface of the river is simulated by mesh construction and dynamic texture mapping. The flow visualization includes particle tracking, streamline rendering and contour surface rendering.

Figure 11 shows the visual effects of the flow simulation. It demonstrates the flow surface through the mixed migration of water texture (Figure 11a) and presents the flow differences in various parts of the river using real flow velocity (Figure 11b). The flow velocity is represented by the moving speed of the water texture. The flow velocity and water level can be interactively queried by users. Figure 11c shows the flow velocity variation of a selected position on the river surface. Billboard technology is applied to display the velocity curve, and the maximum and minimum velocities are counted in real time. Figure 11d shows the cross-section and water level of the selected position. The 3D water gauge model created by 3Dmax software is placed in the river to simulate the real water level measurement. Through flow simulation, users can intuitively determine the overall flow state of the river.

Figure 12 presents the visual effects of flow visualization, including particle tracking, streamline rendering, and contour surface rendering. Figure 12a shows the arrow marker-based particle tracking, which draws the trajectories of particles over a period of time. It is useful to show the flow velocity and direction of the water. LOD technology is applied to control the density of the arrow particles and avoid confusing visual effects. Ribbon trails can also be applied to display the flow direction and continuity (Figure 12b). Figure 12c shows the adaptive streamline rendering effects. The streamlines demonstrate the overall flow direction of the river. The flow vortices are visualized by adaptive sampling step, which achieve high accuracy in the curved parts. Red arrows are integrated to demonstrate the streamline direction. The water depth distribution is displayed by the contour surface model in Figure 12d. Contour surfaces are classified based on the water depth and plotted as hypsometric layers.



当前点经度：31.0354274290981°
当前点纬度：110.16918998357°
当前点水位：157.41 m

Figure 11. Flow simulation: (**a**) dynamic texture mapping (large scale); (**b**) dynamic texture mapping (small scale); (**c**) flow velocity analysis (velocity curve on the billboard); (**d**) profile analysis and water level query (3D water gauge model).

Based on the flow simulation and visualization models, free combination can be applied to build an interactive hybrid flow environment for river shipping. Depth peeling technology [38] is applied to blend the multi flow rendering models based on GPU programming. According to the rendered shape, the hybrid drawing depth is from surface to line to point in descending order. The multi modes are rendered based on sorted depth and blended in a scenario. Figure 13 shows the interactive hybrid flow visualization by combining particle, streamline, and contour surface rendering. Particle tracking presents the river flow velocity and direction in detail, streamline rendering demonstrate the overall flow direction, and contour surface rendering displays the overall water depth distribution. The parameters of the rendering models can be interactively adjusted, resulting in desired effects. All these information are important to monitor the flow conditions of the river. Using hybrid rendering, the flow velocity, flow path, water area, and water depth can be observed from a single scene, which is important for users to acquire different information directly to support river shipping. Figure 14 shows the real-time shipping state analysis. Particle tracking, streamline rendering, and texture mapping are combined to build the hybrid flow environment. Dynamic texture mapping simulates the sense of river flow in the real world, which helps to present the flow situation intuitively. Both the flow conditions and ship state in the ship position are analyzed and displayed in real time. The routing path of the ship is evaluated based on the current flow environment. At the same time, potentially dangerous regions (such as vortices) are analyzed to provide warnings for ship navigation.

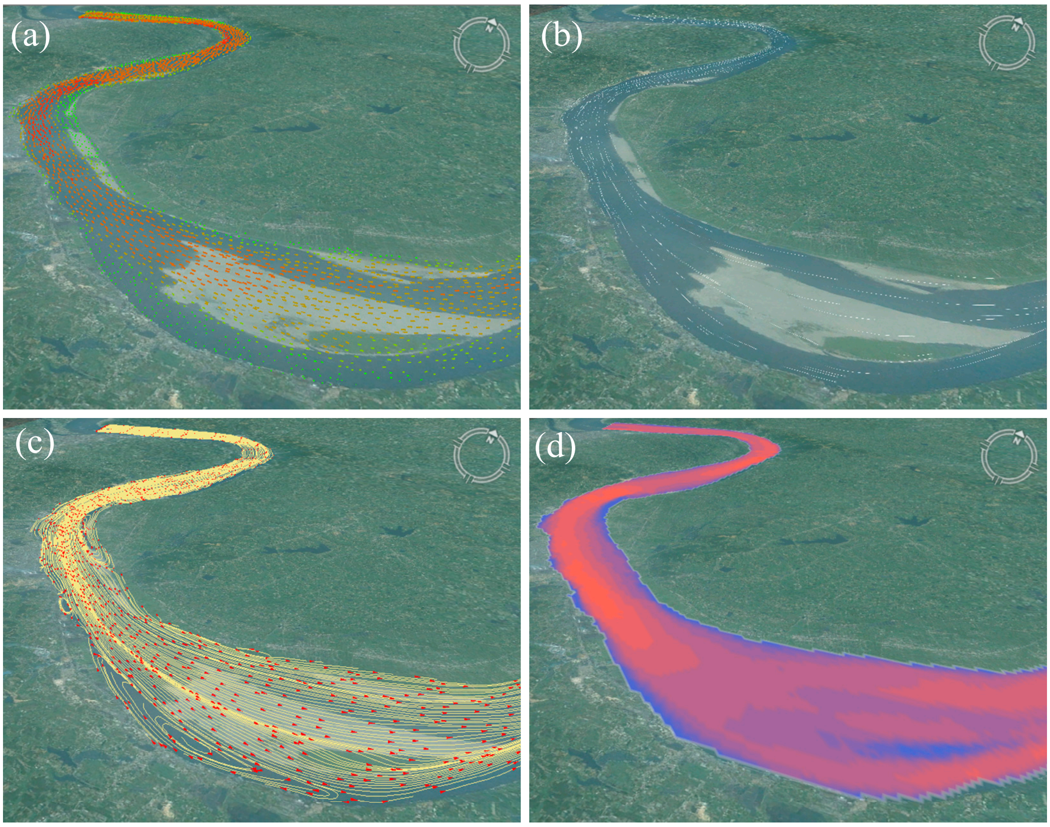
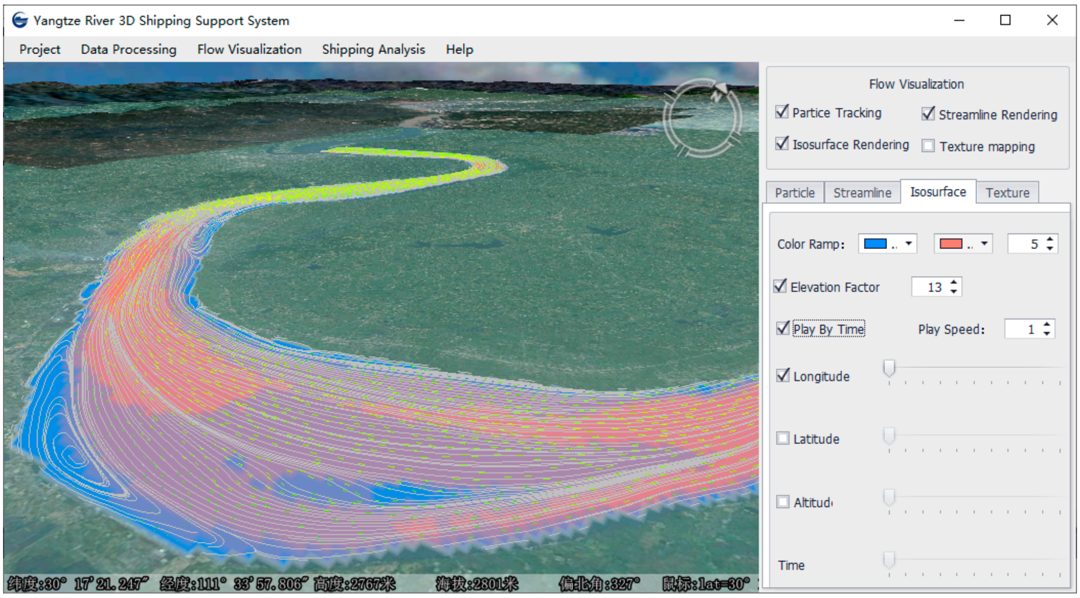


Figure 12. Flow visualization: (**a**) particle tracking (arrow marker); (**b**) particle tracking (ribbon trail); (**c**) streamline rendering; (**d**) contour surface rendering.



Flow Visualization

☑ Partice Tracking ☑ Streamline Rendering

☑ Isosurface Rendering □ Texture mapping

Color Ramp: 🟦 .. 🟥 .. 5

☑ Elevation Factor 13

☑ Play By Time Play Speed: 1

☑ Longitude

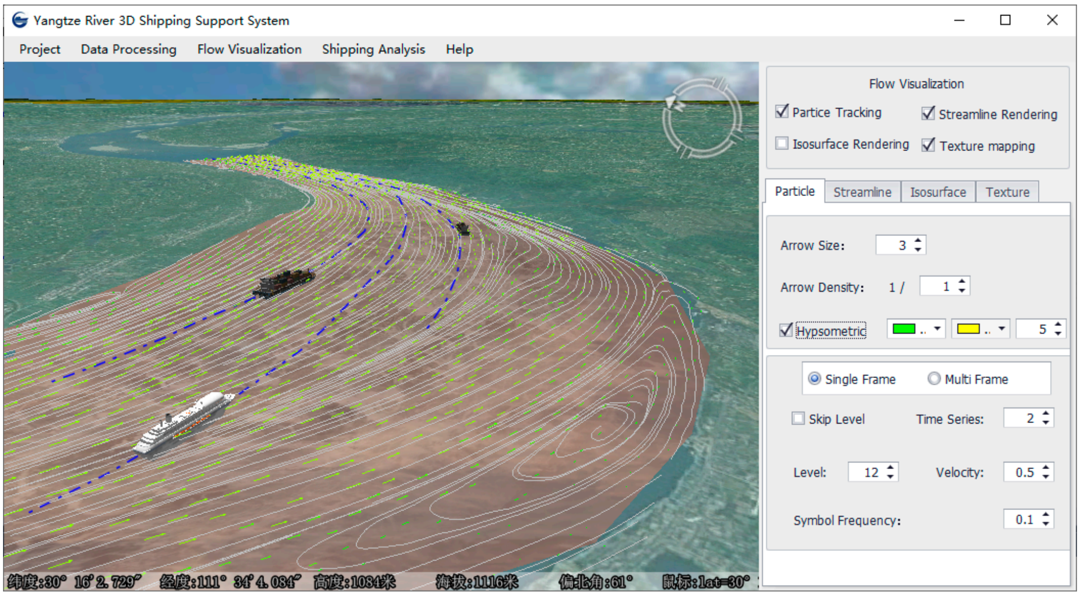
□ Latitude

□ Altitud

Time

纬度:30° 17' 21.247" 经度:111° 33' 57.806" 高度:2767米 海拔:2801米 偏北角:327° 鼠标:lat=30°

Figure 13. Interactive hybrid flow visualization (combining particle tracking, streamline rendering, and contour surface rendering).



Flow Visualization

☑ Partice Tracking ☑ Streamline Rendering
☐ Isosurface Rendering ☑ Texture mapping

Arrow Size: 3
Arrow Density: 1 / 1
☑ Hypsometric 🟩 🟨 5

● Single Frame ○ Multi Frame
☐ Skip Level Time Series: 2
Level: 12 Velocity: 0.5
Symbol Frequency: 0.1

纬度:30° 16′2.729″ 经度:111° 34′4.084″ 高度:1084米 海拔:1116米 偏北角:61° 鼠标:lat=30°

Figure 14. Real-time shipping state analysis (combining particle tracking, streamline rendering, and texture mapping).

Considering the multi-dimensional visualization models, Root Mean Square Error (RMSE) or r^2 are not suitable criteria, so Frame Per Second (FPS) and memory amount are used as the benchmark to evaluate the rendering efficiency quantitatively [39,40]. Table 2 shows the experimental results of the different rendering models and combinations. The performance is evaluated at a 1286×900 screen resolution using perspective projection. As the pyramid terrain structure in the virtual globe considered the influence of viewing distance and vertical scaling, the LOD strategy is applied to the rendering models based on the virtual globe. HLSL programming accelerates the real-time rendering of the mass flow data. The memory amount of the system is below 400 MB. The average FPS of the hybrid rendering 1 (corresponding to Figure 13) is 31, and the average FPS of the hybrid rendering 2 (corresponding to Figure 14) is 36. It should be noted that the rendering cost of the virtual globe is not excluded from the experiment, the actual rendering FPS of the rendering models should be better than the tested numbers. As a result, the cross-sectional data can be interactively visualized to support river shipping.

Table 2. The rendering efficiencies of the flow models.

Rendering Models	Memory Amount	Average FPS
Particle tracking	245 MB	38
Streamline rendering	282 MB	42
Contour surface rendering	291 MB	35
Dynamic texture mapping	250 MB	58
Hybrid rendering 1	353 MB	31
Hybrid rendering 2	329 MB	36

4.2. Discussion

Flow simulation and visualization are used to show the distribution of flow field characteristics and dynamic processes, as well as to provide practical warnings for river shipping navigation [15]. Ships can locate their positions via GPS in the 3D virtual geographical environment and obtain the flow conditions around them. Through the flow modeling and rendering methods presented in this paper, the following results are achieved in the experiments:

(1) Comprehensive flow visualization. Because different visualization methods have different characteristics and advantages, the combination of available rendering methods can express flow

information comprehensively [41]. The flow surface simulation realizes a sense of reality for the river flow environment. The flow field rendering demonstrates the distribution of flow direction, flow velocity, and vortices, supporting early warning to aid navigation. The contour surface rendering shows the water depth distribution, which is important for ship route planning. Based on these rendering methods, hybrid flow visualization achieves comprehensive information display in a single scenario.

(2) Smooth animation effects. The cross-sectional data are extrapolated and interpolated, providing dense sampling points. LOD technology is applied to retrieve appropriate precision data for efficient visualization. GPU programming improves the efficiency of the flow rendering models. According to the experiment results, the hybrid flow rendering FPS is greater than 30.

(3) Interactive rendering and analysis. The cross-sectional data can be interactively visualized by the template configuration and analyzed at a specific location and time. Meanwhile, the framework shows early warning (e.g., vortex) and route planning, which are beneficial to reduce river shipping accidents.

5. Conclusions and Future Work

Cross-sectional data measured by bathymetric surveys or SONAR techniques are important for monitoring river flow conditions continuously. Implementation of multi-dimensional dynamic visualization of cross-sectional data provides a real-time, dynamic, and interactive hydrological environment for ship navigation. This paper investigates data processing, flow modeling, and flow rendering methods for a meandering river. The cross-sectional data are extrapolated and interpolated to provide dense sampling points for flow modeling and rendering. The flow surface visualization simulates a real sense of the flow, the flow rendering models display different flow information. As a result, the interactive hybrid flow environment provides comprehensive flow information and real-time shipping analysis, which are important for river navigation.

To improve flow rendering efficiency, the LOD strategy and GPU acceleration are applied. Although the hybrid rendering is achieved through deep peeling technology, the flow models are actually drawn separately, the rendering effect and efficiency could be further improved. For example, Line Integral Convolution (LIC) [42–44] integrates streamlines and a noise texture for vector field visualization. However, the noise texture is the same over the river surface, and therefore cannot simulate the real water flow [45]. Flow visualization methods that integrate more information and achieve better effects need to be studied. In addition, because flow rendering models involve mass data calculation, the latest parallel computing technologies, such as High Performance Computing (MPI), Spark and Compute Unified Device Architecture (CUDA) [46], can be adopted to further improve rendering efficiency. In future work, parallel hybrid flow visualization algorithms should be studied to support river shipping.

Author Contributions: All authors contributed extensively to the work presented in this paper. Xuequan Zhang formed the research idea, designed and conducted the experiment, and drafted the manuscript. Jin Liu analyzed the results and revised the manuscript. Zihe Hu processed the ship GPS data and analyzed ship navigation. Ming Zhong contributed to the methodology design. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by research funds from Major State Research Development Program of China (No. 2017YFB0504103), National Science and Technology Fund of China (No. 2013ZX07503001-04, No. 2013ZX07503001-06).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. WMO. *The Global Climate in 2015–2019*; WHO: Geneva, Switzerland, 2019.
2. Bizzi, S.; Demarchi, L.; Grabowski, R.; Weissteiner, C.; Van de Bund, W. The use of remote sensing to characterise hydromorphological properties of European rivers. *Aquat. Sci.* **2016**, *78*, 57–70. [CrossRef]

3. Coveney, S.; Roberts, K. Lightweight UAV digital elevation models and orthoimagery for environmental applications: data accuracy evaluation and potential for river flood risk modelling. *Int. J. Remote Sens.* **2017**, *38*, 3159–3180. [CrossRef]
4. Costabile, P.; Macchione, F.; Natale, L.; Petaccia, G. Flood mapping using LIDAR DEM. Limitations of the 1-D modeling highlighted by the 2-D approach. *Nat. Hazards* **2015**, *77*, 181–204. [CrossRef]
5. Lin, Y.-T.; Chen, W.-B.; Su, Y.-F.; Han, J.-Y.; Jang, J.-H. Improving river stage forecast by bed reconstruction in sinuous bends. *J. Hydroinformatics* **2018**, *20*, 960–974. [CrossRef]
6. Dal Cin, C.; Moens, L.; Dierickx, P.; Bastin, G.; Zech, Y. An integrated approach for realtime floodmap forecasting on the Belgian Meuse River. *Nat. Hazards* **2005**, *36*, 237–256.
7. Legleiter, C.J.; Kyriakidis, P.C. Spatial prediction of river channel topography by kriging. *Earth Surf. Process. Landf. J. Br. Geomorphol. Res. Group* **2008**, *33*, 841–867. [CrossRef]
8. Lázaro, J.M.; Navarro, J.Á.S.; Gil, A.G.; Romero, V.E. Developing and programming a watershed traversal algorithm (WTA) in GRID-DEM and adapting it to hydrological processes. *Comput. Geosci.* **2013**, *51*, 418–429. [CrossRef]
9. Schäuble, H.; Marinoni, O.; Hinderer, M. A GIS-based method to calculate flow accumulation by considering dams and their specific operation time. *Comput. Geosci.* **2008**, *34*, 635–646. [CrossRef]
10. Flanagin, M.; Grenotton, A.; Ratcliff, J.; Shaw, K.B.; Sample, J.; Abdelguerfi, M. Hydraulic splines: A hybrid approach to modeling river channel geometries. *Comput. Sci. Eng.* **2007**, *9*, 4–15. [CrossRef]
11. Schäppi, B.; Perona, P.; Schneider, P.; Burlando, P. Integrating river cross section measurements with digital terrain models for improved flow modelling applications. *Comput. Geosci.* **2010**, *36*, 707–716. [CrossRef]
12. Caviedes-Voullième, D.; Morales-Hernández, M.; López-Marijuan, I.; García-Navarro, P. Reconstruction of 2D river beds by appropriate interpolation of 1D cross-sectional information for flood simulation. *Environ. Model. Softw.* **2014**, *61*, 206–228. [CrossRef]
13. Zhang, Y.; Xian, C.; Chen, H.; Grieneisen, M.L.; Liu, J.; Zhang, M. Spatial interpolation of river channel topography using the shortest temporal distance. *J. Hydrol.* **2016**, *542*, 450–462. [CrossRef]
14. Chen, W.-B.; Liu, W.-C. Modeling the influence of river cross-section data on a river stage using a two-dimensional/three-dimensional hydrodynamic model. *Water* **2017**, *9*, 203. [CrossRef]
15. Zhang, S.; Zhang, T.; Wu, Y.; Yi, Y. Flow simulation and visualization in a three-dimensional shipping information system. *Adv. Eng. Softw.* **2016**, *96*, 29–37. [CrossRef]
16. Zhang, S.; Li, W.; Lei, X.; Ding, X.; Zhang, T. Implementation methods and applications of flow visualization in a watershed simulation platform. *Adv. Eng. Softw.* **2017**, *112*, 66–75. [CrossRef]
17. Wu, Y.; Peng, F.; Peng, Y.; Kong, X.; Liang, H.; Li, Q. Dynamic 3D Simulation of Flood Risk Based on the Integration of Spatio-Temporal GIS and Hydrodynamic Models. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 520. [CrossRef]
18. Liu, S.; Chen, G.; Yao, S.; Tian, F.; Liu, W. A framework for interactive visual analysis of heterogeneous marine data in an integrated problem solving environment. *Comput. Geosci.* **2017**, *104*, 20–28. [CrossRef]
19. Su, T.; Cao, Z.; Lv, Z.; Liu, C.; Li, X. Multi-dimensional visualization of large-scale marine hydrological environmental data. *Adv. Eng. Softw.* **2016**, *95*, 7–15. [CrossRef]
20. Zhang, F.; Mao, R.; Du, Z.; Liu, R. Spatial and temporal processes visualization for marine environmental data using particle system. *Comput. Geosci.* **2019**, *127*, 53–64. [CrossRef]
21. Tian, F.; Cheng, L.; Chen, G. Transfer function-based 2D/3D interactive spatiotemporal visualizations of mesoscale eddies. *Int. J. Digit. Earth* **2018**. [CrossRef]
22. Wang, S.; Kang, F. Automatic optimization for large-scale real-time coastal water simulation. *Math. Probl. Eng.* **2016**, *2016*. [CrossRef]
23. Nugjgar, P.; Fujimoto, T.; Chiba, N. Markov-type velocity field for efficiently animating water stream. *Vis. Comput.* **2012**, *28*, 219–229. [CrossRef]
24. Nugjgar, P.; Chiba, N. Markov-Type Vector Field for endless surface animation of water stream. *Vis. Comput.* **2013**, *29*, 959–968. [CrossRef]
25. Chentanez, N.; Müller, M. Real-time simulation of large bodies of water with small scale details. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*; Eurographics Association: Goslar, Germany, 2010; pp. 197–206.
26. Burrell, T.; Arnold, D.; Brooks, S. Advected river textures. *Comput. Animat. Virtual Worlds* **2009**, *20*, 163–173. [CrossRef]

27. Boultif, A.; Louër, D. Powder pattern indexing with the dichotomy method. *J. Appl. Crystallogr.* **2004**, *37*, 724–731. [CrossRef]
28. Harvey, B.R. Transformation of 3D co-ordinates. *Aust. Surv.* **1986**, *33*, 105–125. [CrossRef]
29. Blu, T.; Thévenaz, P.; Unser, M. Linear interpolation revitalized. *IEEE Trans. Image Process.* **2004**, *13*, 710–719. [CrossRef]
30. Hu, P.; Hou, J.; Zhi, Z.; Li, B.; Guo, K. An Improved Method Constructing 3D River Channel for Flood Modeling. *Water* **2019**, *11*, 403. [CrossRef]
31. Sibson, R. Locally equiangular triangulations. *Comput. J.* **1978**, *21*, 243–245. [CrossRef]
32. Zhang, X.; Yue, P.; Chen, Y.; Hu, L. An efficient dynamic volume rendering for large-scale meteorological data in a virtual globe. *Comput. Geosci.* **2019**, *126*, 1–8. [CrossRef]
33. Zhang, X.; Zhong, M.; Liu, S.; Zheng, L.; Chen, Y. Template-Based 3D Road Modeling for Generating Large-Scale Virtual Road Network Environment. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 364. [CrossRef]
34. Shewchuk, J.R. Delaunay refinement algorithms for triangular mesh generation. *Comput. Geom.* **2002**, *22*, 21–74. [CrossRef]
35. Hawick, K.A. Modelling flood incursion and coastal erosion using cellular automata simulations. In Proceedings of the IASTED International Conference on Environmental Management and Engineering, Banff, AB, Canada, 16–18 July 2014; 2014; pp. 16–17.
36. Wang, S.; Li, W. Capturing the dance of the earth: PolarGlobe: Real-time scientific visualization of vector field data to support climate science. *Comput. Environ. Urban Syst.* **2019**, *77*, 101352. [CrossRef]
37. Bell, D.G.; Kuehnel, F.; Maxwell, C.; Kim, R.; Kasraie, K.; Gaskins, T.; Hogan, P.; Coughlan, J. NASA World Wind: Opensource GIS for mission operations. In Proceedings of the Aerospace Conference, Big Sky, MT, USA, 3–10 March 2007; pp. 1–9.
38. Bavoil, L.; Myers, K. *Order Independent Transparency with Dual Depth Peeling*; NVIDIA OpenGL SDK: Santa Clara, CA, USA, 2008; pp. 1–12.
39. Helbig, C.; Bilke, L.; Bauer, H.-S.; Böttinger, M.; Kolditz, O. Meva-an interactive visualization application for validation of multifaceted meteorological data with multiple 3d devices. *PLoS ONE* **2015**, *10*. [CrossRef] [PubMed]
40. Liu, P.; Gong, J.; Yu, M. Visualizing and analyzing dynamic meteorological data with virtual globes: A case study of tropical cyclones. *Environ. Model. Softw.* **2015**, *64*, 80–93. [CrossRef]
41. Zhang, X.; Zhang, M.; Jiang, L.; Yue, P. An interactive 4D spatio-temporal visualization system for hydrometeorological data in natural disasters. *Int. J. Digit. Earth* **2019**, 1–21. [CrossRef]
42. Forssell, L.K.; Cohen, S.D. Using line integral convolution for flow visualization: Curvilinear grids, variable-speed animation, and unsteady flows. *IEEE Trans. Vis. Comput. Graph.* **1995**, *1*, 133–141. [CrossRef]
43. Ma, Y.-Y.; Guo, Y.-F. Visualization of Vector Field using Line Integral Convolution based on Visual Perception. In *Proceedings of the 2nd International Symposium on Computer Science and Intelligent Control*; ACM: Stockholm, Sweden, 2018; p. 58.
44. Min, H.; Haichao, Z.; Maosong, B.; Danchen, Z. Flow Visualization Based on Enhanced Streamline Line Integral Convolution. *J. Syst. Simul.* **2016**, *28*, 2933–2938.
45. Tang, B.; Shi, H. Intelligent vector field visualization based on line integral convolution. *Cogn. Syst. Res.* **2018**, *52*, 828–842. [CrossRef]
46. Reyes-Ortiz, J.L.; Oneto, L.; Anguita, D. Big data analytics in the cloud: Spark on hadoop vs mpi/openmp on beowulf. *Procedia Comput. Sci.* **2015**, *53*, 121–130. [CrossRef]