Article

Particle System-Based Multi-Hierarchy Dynamic Visualization of Ocean Current Data

Qingtong Shi 1, Bo Ai 1,*, Yubo Wen 2, Wenjun Feng 3, Chenxi Yang 2 and Hongchun Zhu 1

1 College of Geodesy and Geomatics, Shandong University of Science and Technology, Qingdao 266590, China; alexamo2461@gmail.com (Q.S.); hongchun@sdust.edu.cn (H.Z.)
2 South China Sea Information Center of State Oceanic Administration, Guangzhou 510310, China; wenyb@sos.mnr.gov.cn (Y.W.); yangchenxi@sos.mnr.gov.cn (C.Y.)
3 Qingdao Yuehai Information Service Co., Ltd., Qingdao 266590, China; fwj@oceanread.com
* Correspondence: aibo@sdust.edu.cn

Abstract: In three-dimensional (3D) digital Earth environment, there are many problems when using the existing methods to express the ocean current, such as uneven distribution of seed points, density leap in scale change and messy visualization. In this paper, a new dynamic visualization method of multi-hierarchy flow field based on particle system is proposed; Specifically, three typical spherical uniform algorithms are studied and compared, and the streamline becoming denser from the equator to the poles on globe is eliminated by placing seed points using Marsaglia polar method as the most efficient. In addition, a viewport-adaptive adjustment algorithm is proposed, which realizes that the density of particles is always suitable to any viewing distance during continuous zooming. To solve the visual representation deficiency, we design a new dynamic pattern to enhance the expression and perception of current, which makes up for the shortcoming of the arrow glyph and streamline methods. Finally, a prototype of GPU parallel and viewport coherence is achieved, whose feasibility and effectiveness are verified by a series of experiments. The results show that our method can not only represent ocean current data clearly and efficiently, but also has outstanding uniformity and hierarchy effect.

Keywords: ocean current; flow visualization; streamlet; particle system; dynamic visualization

1. Introduction

In marine science the term ocean current refers to the large-scale and relatively regular flow of seawater, which has an important impact on global heat transport, weather and climate, sediment transport and marine transport. Therefore, oceanographers, coastal managers and ships need to obtain current information, analyse and master the temporal and spatial variation law of currents [1]. As a means of visual expression and an effective data analysis method, ocean current scientific visualization can intuitively extract and display ocean information from massive data and is of great significance for revealing the dynamic changes of global ocean currents.

Since feature-based methods focus more on extracting subsets with features from entire datasets, the methods for the flow field visualization are mainly divided into three categories in terms of drawing: direct visualization, geometric visualization and texture-based visualization [2]. Direct visualization uses repeatable icons such as arrow glyphs on a regular grid to represent the flow field at different positions [3]; it is more suitable for the static display of two-dimensional (2D) flow fields because of its simple expression and fewer calculation characteristics. However, this method has the disadvantage of visual complexity and lack of visual coherency. Geometric visualization uses streamlines, surfaces, pipes and other geometric elements to simulate flow information [4], which can better express the flow structure and spatial continuity. Nevertheless, its rendering effect depends heavily on the position of seed points, which is prone to causing streamline
occlusion and clusters. The texture-based visualization method convolved the flow field information with the filter on a random noise texture to represent the flow field by a continuous and consistent dense texture [5]. However, its processing result is a static image with no vector features and is poor for multi-scale expression. For the presentation of geographic environment vector fields, such as wind and wave, arrow has been the mainstream expression in early cartography for a long time [6]. With the development of computer graphics and animation technology, slender streamline geometry to express flow field information becomes more and more popular [7]. As the need to generate noise and cumbersome convolution operation, texture-based methods are mainly used to display generalized flow fields and are rarely used in cartography.

With the increasing spatial resolution and data density of collected data and forecast models, the need for better visualization methods becomes more important [8]. The particle system can be regarded as a variant of the geometric method, where a set of particles are placed and advected over time. Each particle has a position, velocity, certain lifetime and other properties [9]; When a particle exceeds its lifetime, stops moving, or exits the dataset domain, a new particle is reseeded in a random position [10]. Due to attractive effects (creating high-contrast images and finite clutter), it has been widely used in the numerical simulation and expression of vector fields [11]. In the last decades, relevant research and visualization platforms using the particle system concept achieved good results in meteorological, oceanological and other vector fields, which were mainly based on two-dimensional map drawing [12–14]. Clearly, extending flow visualization from 2D to 3D has the advantage of providing more comprehensive phenomenon and perceptual information [15]. Displaying the flow field information under the 3D earth not only eliminates the deformation caused by the map projection (compared with the 2D), but also gives richer display ability with the depth. Regarding visualization in 3D environment, much prior research is only technology realization and application [16–19]. For example, Fu [17] built a particle management strategy to realize efficient visualization of global flow field with GPU parallelism. Zhang et al. [18] realized the visualization of meteorological and hydrological data, such as typhoon and flood, at the mesoscale using particle tracking method. Wang et al. [19] studied and realized the time-series expression of large-scale wind data in the web environment. The above studies all draw the surface flow field as 2D graticules, which made the flow lines gradually dense from the equator to the poles and damaged the dynamic visualization effect. Furthermore, continuous zooming as a fundamental function of digital earth, a fixed number of particles cannot satisfy the continuously display requirement, when the flow field is observed at different viewing height [20].

To improve the visualization effect and realize the multi-hierarchy expression of ocean flow field in the 3D digital earth environment, we propose a dynamic flow visualization method based on particle system. This paper introduces a seeds placement algorithm under the realization of 3D spherical particles evenly distributed. An adaptive adjustment algorithm based on viewport is designed to solve the density problem caused by scale changes during scaling. In addition, the properties of particles are improved based on visual perception as the animate pattern of flow field. The rest of the paper is organized as follows: materials and methods are introduced in Section 2, including ocean current data, framework details and methods introduction. Experiments and results are given in Section 3. Finally, the conclusions of this work are discussed in Section 4.

2. Materials and Methods

2.1. Ocean Current Data

Generally, current data include tidal current and circulation. Tidal current is the horizontal flow phenomenon of seawater under the influence of tidal force; circulation is the result of the superimposed influence of several factors, such as deflection force caused by the rotation of the Earth, wind, and land-sea distribution. At the global scale, the circulation plays a decisive role in the process of seawater flow. The circulation is
influenced by the topography near the shore and velocity, while the effect of tidal current near the shore is more prominent [21].

In this study, the data (Table 1) selected for the surface forecast of global circulation and tidal current were provided by the South China Marine Forecasting Centre of State Oceanic Administration. The original data format is NetCDF, which was developed by Unidata project scientists of the University Corporation for Atmospheric Research (UCARU) and was used as the standard input and output format of FVCOM numerical model. The above data are represented by a regular data grid, that is, the same number of points is generated to represent the data of different longitude and latitude. The flow vector of each data point is represented by two variables <u, v> that indicate the vector components in the horizontal (x, y), respectively. First, the two kinds of data are superimposed into comprehensive current forecast data on a global scale. Then, the NetCDF format is compressed into a texture format, because the amount is too large to data transmit and use in the web environment. The texture size of the flow field data is 1440 * 720, which is the same as the original data arrangement and the data size is only about 30 KB.

Table 1. Summary of the current data.

| Option                | Parameter                   |
|-----------------------|-----------------------------|
| Spatial resolution    | 0.25° × 0.25°               |
| Temporal resolution   | 3 h                         |
| Longitudinal range    | 180° W–180° E              |
| Latitudinal range     | 90° S–90° N                |
| Dimension             | 1440 × 720                 |
| Data size             | 1.13 GB                     |

2.2. Framework

An overview diagram of the multi-hierarchy dynamic visualization is shown in Figure 1. Our method can efficiently deal with the dynamic render of global ocean current with the help of frustum culling technology and GPU acceleration.

Data Preprocessing: The core of data preprocessing is to obtain and analyse the current data stored in NetCDF file, divide and re-output them to a new format for transmission. As discussed in Section 2.1, we develop a data preprocessing program to help us complete the work of parse, synthesis and format conversion. During the execution of
the program, the circulation and tidal current data at each moment are split from the original data according to the time dimension, and then they are combined. Moreover, the vector information of the original dataset is stored in the RGBA channels of texture space; u and v components of the flow field are stored in the R and G channels and stretched to (0.255). The B channel of texture stores the critical point of the flow field, and the A channel is used to judge the sea and land position.

- **Camera Control**: In our method, particles are only drawn in the current viewport. The display will be redrawn if the camera is moved, which improve efficiency and facilitate the adjustment of density. Camera and viewport information such as camera position, rotation angle and viewport size is obtained to calculate the appropriate number of particles using our proposed algorithm before rendering; this makes adaptive resolution representation possible to present multi-hierarchy ocean current information.

- **Dynamic Rendering**: This part mainly consists of four subsections: (1) seeding, (2) symbol design, (3) particle tracking and (4) rendering. Drawing process is also shown on the right side of Figure 1. The spherical seed point distribution algorithm is introduced to place a fixed number of particles randomly and evenly in the viewport. A texture is created using a shader program, and pixels are equal to the number of particles in implementation. In addition to saving the position information of particles, each pixel also saves other properties such as age. The position and other properties are constantly updated at each frame. Initially, the particle age is set to zero to record its state.

2.3. Methodology

In this section, the spherical seed placement algorithm is discussed. Then the adaptive density adjustment strategy is introduced, and the dynamic streamlet generation process is proposed finally.

2.3.1. Spherical Seed Placement Algorithm

At present, the streamline visualization method based on virtual earth takes the points as a 2D flow field when seeding, divides the plane into several sub-regions by grid and places the points randomly or uniformly, and finally converts the plane to the sphere through spherical coordinate transformation. It is an angular uniformity on the sphere, and the “bunching” will occur near the pole when there are many seed points (Figure 2). A seed point placement method with uniform distribution on 3D spheres needs to be used to make the particle points effectively cover the flow field information on the virtual earth and improve the expression effect. We summarize and compare typical uniform spherical algorithms and use the optimal method as the seed point generation algorithm, which can avoid the particles being too dense in the polar region and too sparse in the equatorial region.

In probability statistics, there are three major spherical random uniform distribution algorithms that are proposed by Muller, Cook and Marsaglia [22].

(1) **Muller’s method**. Generate $V_1$, $V_2$, $V_3$ from an independent uniform distribution on $(-1, 1)$ until $S = V_1^2 + V_2^2 + V_3^2 < 1$, then the coordinates are:

$$ (x, y, z) = \left( \frac{V_1}{\sqrt{S}}, \frac{V_2}{\sqrt{S}}, \frac{V_3}{\sqrt{S}} \right) $$

(1)

This idea is to select a point in the cube randomly, reject it unless it is inside the tangent sphere, then project the point onto the sphere’s surface.

(2) **Cook’s method**. Generate $V_1$, $V_2$, $V_3$, $V_4$ from an independent uniform distribution on $(-1, 1)$ until $S = V_1^2 + V_2^2 + V_3^2 + V_4^2 < 1$, then the coordinate is:

$$ (x, y, z) = \left( \frac{2(V_2V_4 + V_1V_3)}{S}, \frac{2(V_3V_4 - V_1V_2)}{S}, \frac{(V_1^2 + V_2^2 - V_3^2 - V_4^2)}{S} \right) $$

(2)
It is an extension of the random distribution of points in the plane circle in 3D space. Meanwhile, the method can extend more generally to the uniform distribution of points in N-dimensional space.

(3) Marsaglia’s method. Generate $V_1, V_2$, from an independent uniform distribution on $(-1, 1)$ until $S = V_1^2 + V_2^2 < 1$, then the coordinate is:

$$ (x, y, z) = \left( 2V_1 \sqrt{1-S}, 2V_2 \sqrt{1-S}, 1 - 2S \right) $$

\[ (3) \]

Figure 2. Incorrectly distributed points.

Marsaglia’s method is a pseudo-random number sampling method used to generate a pair of independent standard normal random variables [23]. In fact, it is a variant of Muller’s algorithm based on transformation sampling, which avoids calculating trigonometric functions by transforming the generating function to polar coordinates.

2.3.2. Adaptive Density Adjustment Strategy

Besides the uniform distribution of particles in the entire mapping area, the dynamic change of density when zooming is also the requirement for the characteristic of vector field. The overall trend of the current wants to be observed when the camera is far away from the earth, the information should be simplified, the number of particles should be reduced, and the particle symbols should be highlighted. Local details want to be observed when the viewing distance is closer, the resolution of the focus area needs to be improved, and the particles should be encrypted to highlight the detailed information. In order to achieve reasonable density at different scales, this paper proposes a density control method based on viewport. Particles are adjusted according to certain rules when the viewing distance changes to keep the particles in the screen within an appropriate number range and improve the effect of flow field visualization.

In the 3D environment, digital Earth is projected onto the flat plane through perspective transformation [24], and the final rendering rectangle display area is called viewport. However, the process has a noticeable distortion; the geographical distances of equal intervals (such as one pixel) on the screen are different. [25]. The farther the viewing distance is, the more pronounced deformation of the earth image, especially in the surrounding area far from the viewpoint. Therefore, we believe that the key to density adjustment is controlling the number of seed points according to the proportion of spherical deformation under different viewing distances.

Specifically, we propose to regard the digital earth as a standard sphere. The ratio between the spherical crown surface area and the corresponding secant area at the current
viewing distance is calculated as the density adjustment factor, and the number of particles is set and multiplied by the adjustment coefficient. Firstly, it is reasonable to regard it as a standard sphere. Taking WGS84 reference ellipsoid as an example, its semimajor axis is 6378.137 km, and the semiminor axis is 6356.752 km. The flattening is only $1/(298.257 223) \approx 0.003352$. There is no need to be so precise because it is just to get a reasonable-looking density in the viewport, though they are essential for surveying.

The principle is shown in Figure 3, where $R$ is the radius of the virtual earth (regard it as a sphere here), $S_1$ is the surface area of the sphere crown at height $h$, and $S_2$ is the area of the cut circle. The mathematical expression of the proper number of particles is given by Equation (4):

$$\text{num} = N \cdot f = N \cdot \frac{S_1}{S_2}$$

$$S_1 = 2\pi Rh, \quad S_2 = \pi r^2$$

Figure 3. Diagram of particle number adjustment.

In Equation (4), $f$ is the adjustment factor. $N$ is the basic number that can be customized because different people have different habits of visual experience and sensitivity regarding density.

The calculation of $h$ in Figure 3 is shown in Figure 4 and Equation (6), where $d$ is the viewing distance, $O$ represents the center of the globe and $A$ represents the viewing position.

$$h = \frac{\overrightarrow{AB} \cdot \overrightarrow{AC}}{d} - d$$

The projection of $\overrightarrow{AB}$ onto $\overrightarrow{AC}$ is $\overrightarrow{AD}$, where $D$ is perpendicular foot, $B$ is the furthest point from the camera within the viewport, while $C$ is the closest. According to Equation (4), the number of particles is about twice that of $N$ on global scale, and it is approximately the same as $N$ on local scale. When camera zooms, $f$ is a continuous function with the adjustment of viewing distance, and the change of density in the viewport is smooth without jumping sensation. Particles in the viewport always maintain a reasonable density and have good visual stability whenever the visual field is enlarged or narrowed.
2.3.3. Streamlets Generation Based on Visual Perception

The effectiveness of a data display depends on how well critical patterns can be perceived. A static pattern or symbol generally has shape, size, direction, color, and texture properties from visual perception. As the sign moves, it also includes such dynamic properties as time, order, rate, and frequency [26]. Streamline attributes are often associated with the flow information when streamlines are used to display the flow field. For example, the length or color of line segment represents the velocity, and the tangent direction of motion represents the direction. However, outlines without width can not only weaken the expression of flow field information but are still ambiguous with respect to direction, especially when static streamline is used for expression, it is difficult to distinguish the direction of movement of the flow field. Ware et al. [8] suggests that some form of asymmetry along the contour is needed to resolve the directional ambiguity, and the teardrop-shaped design provides more robust asymmetry than arrows. Meanwhile, much evidence suggests that animated representations of flow fields are more effective than common static representations, even for steady flow [27–29]. Inspired by visual perception and literature [8], we design a dynamic streamlet pattern and express flow information through iterative drawing.

The dynamic streamlet pattern (Figure 5) makes full use of various visual variables to express orientation and direction signals: the speed is indicated with color and step, and the direction and orientation are indicated with transparency and shape. In addition, the head-to-tail asymmetries can be mapped to direction in the expression of steady flow and change of time in unsteady flow. As shown in Figure 5b, each streamlet is drawn by several iterations, and this process consists of three stages: growth, activity, and extinction, divided by $l_{\text{birth}}, l_{\text{young}}, l_{\text{mature}}$ and $l_{\text{extinct}}$. Check the particles before tracking, only active particles may be rendered in viewports, and once the particles are completely dead, they are no longer painted.

Specifically, one of the following three situations are considered dead: (1) The speed at the position is 0. (2) The particle is beyond the viewport area. (3) The particle reaches its life span. The present life of particles is set to $l$, when $l_{\text{birth}} < l < l_{\text{young}}$, segment is constantly superimposed, and the length of streamlet is getting longer and longer to achieve the effect of fading in. When $l_{\text{young}} < l < l_{\text{mature}}$, the total number of segments of the streamlet in the live phase remains the same, with each segment drawn forward, a segment at the tail disappears in screen. Accordingly, the segment is continuously cleared in the extinction stage, and the streamlet becomes shorter and shorter until it disappears, which has the effect of fading out. The most important thing in this process is that no matter how long the streamlet is, the streamlet from head to tail is always from entirely opaque to entirely transparent.

Figure 4. The relation between viewing distance and $h$ in Figure 3.
Figure 5. (a) The head-to-tail asymmetry along the streamline can clearly provide direction and orientation signals, it can be accomplished by shape and transparency; the color and step size of primitive can indicate magnitude. (b) Particle dynamic rendering diagram with continuously updated age.

In the drawing, particles move in 3D space iteration by iteration; only a tiny segment is drawn in each iteration, and which is tangent to the local vector field. If particle’s position in the flow field at time $t$ is $y_t$ and velocity is $v(y_t)$, then the equation for the position and time of the particle is:

$$y_{t+\Delta t} = y_t + \int v(y_t) dt$$  \hspace{1cm} (7)

When particles are tracking, the solution of this differential equation needs to be obtained by numerical integration to plot the trajectory [30]. Here we use the Runge-Kutta 4 method to solve the problem:

$$\begin{cases} y_{t+\Delta t} = y_t + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 = v(y_t)\Delta t \\ k_2 = v(y_t + \frac{k_1}{2})\Delta t \\ k_3 = v(y_t + \frac{k_2}{2})\Delta t \\ k_4 = v(y_t + k_3)\Delta t \end{cases}$$ \hspace{1cm} (8)

We use bilinear interpolation to complete the data sampling because the next step of the particles can not necessarily move to another point due to the original data are discrete grids in space. Another key step is geographical transformation to ensure ocean data value is accurately placed in their actual spatial positions above or below the sphere. It mainly includes converting original data coordinates, unit coordinates, geographic coordinates, and 3D cartesian coordinates.

3. Experiments and Results

In this section, we conduct a series of experiments to evaluate the performance of the visualization method. The client machine has 16 GB memory, an Intel Core i5-8300H octacore processor CPU operating at 2.3 GHz and an NVIDIA GeForce GTX 1060 GPU. Our dynamic visualization prototype of ocean current is developed based on Cesium and written in WebGL. The rendering parameters are shown in Table 2. As described in Section 2.1, the data includes circulation and tidal current, and the time is 9:00 on 25 July 2021.
Table 2. Parameters of digital Earth rendering.

| Option            | Parameter            |
|-------------------|----------------------|
| Camera projection | perspective projection |
| Viewing frustum   | 60°                  |
| Reference ellipsoid | WGS84               |
| Viewport          | 1920*1080            |

3.1. Experiments on Validity of the Seed Placement Methods

The above three algorithms generate random points in the three-dimensional space \( R^3 = [-1, 1]^3 \) to evaluate the uniformity of the global distribution, and the results are presented in Figure 6. The methods from top to bottom are Muller’s, Cook’s and Marsaglia’s. It is observed that the uniformity of the algorithms is similar when the number of sphere points is large as if all the methods can guarantee the uniform distribution of streamline on the Earth sphere.

![Figure 6. Distribution of points created by three algorithms. (a) is Muller’s, (b) is Cook’s and (c) is Marsaglia’s. The number of points from left to right is 1000, 10,000 and 100,000.](image)

However, the efficiency of the algorithm is critical because the number of seed points required often needs hundreds of thousands. Firstly, we compare the space occupation of three algorithms to identify the proper method, as shown in Table 3.
Table 3. Comparison of space occupation of three algorithms.

| Method      | Space Efficiency | Uniform Variates |
|-------------|------------------|------------------|
| Muller’s    | $\frac{\pi}{6}$ | $\frac{18}{\pi}$ |
| Cook’s      | $\frac{\pi^2}{32}$ | $\frac{128}{\pi^2}$ |
| Marsaglia’s | $\frac{\pi}{4}$ | $\frac{8}{\pi}$ |

Then the generation time of different numbers of points under the same hardware and coding environment is compared. In Figure 7, the x-axis indicates the number of generated particles, and the y-axis represents the average time cost for rendering.

Figure 7. Points generation time of different methods.

The results show that Marsaglia’s method is more efficient in space occupation and time cost when it is applied to the seed point placement. The rendering result of the flow field in the Arctic region is shown in Figure 8, in which Figure 8a uses angle random distribution and Figure 8b uses spherical uniform algorithm.

Figure 8. Comparison of Rendering Effect of Flow Field in Arctic Region. (a) is the result of using traditional placement method, and (b) is the result of using Marsaglia’s algorithm.

Therefore, we can see that Marsaglia’s method can solve the problem of high-density distribution in the two poles in the global flow field environment.
3.2. Experiments on Visual Effect of the Flow Field Representation

Figure 9 is a comparison of the visual effects of the direct method, the streamline method, and the dynamic streamlet method based on the particle system. This comparison investigates the visualization of the Kuroshio, located near 34 °N.

![Figure 9](image)

Figure 9. Current data representation is generated by different visualization methods at a viewing distance of 2000 km. (a) is the rendering for the direct visualization method (b) is the rendering for the geometric visualization method, and (c) is the rendering for the dynamic streamlet method based on the particle system.

The disadvantages are obvious when the direct method uses opaque arrow symbols with different lengths to represent the flow field: (1) the rationality of the sampling density of the icon is difficult to control; (2) as a discrete display method, it destroys the internal continuity of the data. In Figure 9b, the streamline method uses vector lines to simulate the motion of ocean current with different color standards for different ranges of velocity. There are still deficiencies in expressing the direction of flow, although it can retain spatial continuity better than the direct method. Our method makes up for this shortcoming very well. In fact, the visual variables affecting dynamic symbols are abstracted into some parameters in our method, and the range of parameters is set according to experience, allowing users to adjust the variable values like width, transparency, change rate and frequency, to quickly change the overall effect to meet the user-defined needs.

When drawing the ocean flow field, it is crucial to reveal the current feature precisely. The critical point refers to the position where each component in the flow field is zero and generally divided into the saddle, repelling node, attracting node, attracting focus, repelling focus, and center according to the topology [31]. Figure 10 shows the result of depicting the global current using our method, reflecting the character of seawater movement around the world; Only the five structures are shown in Figure 10 since there are very few repulsive nodes in the actual ocean.

![Figure 10](image)

Figure 10. Flow feature representation diagram.
3.3. Experiments on Feasibility of the Density Adjustment Strategy

The comparison results are shown in Figure 11 when viewing the globe at 2000 km and set the basic number to 10,000, 20,000 and 50,000. In this case, the display becomes very dense and disturbs observing the current movement process when \( N \) is high. One can hardly observe the distribution and feature of current when \( N \) is small, and the feature and evolution are clear and easy to inspect when \( N \) is set to a medium number. As a matter of fact, different people have different visual perceptions of particle density; when \( N \) is set to about 15,000–35,000 (viewing distance is 2000 km), the results accord with most people’s visual habits. Focusing on the process of camera movement is a notable advance of the proposed adjustment method; With the adjustment, the change of density has been smooth without visual jump occurring.

**Figure 11.** Current data representation generated by different visualization methods at the viewing distance of 2000 km. (a) is the rendering for the direct visualization method; (b) is the rendering for the geometric visualization method; and (c) is the rendering for the dynamic streamlet method based on the particle system.

Setting the basic number to 20,000, the results under different viewing distances are shown in Figure 12 using the proposed adaptive density adjustment method. The number of streamlets is about 40,000 (Figure 12a) when the distance is far from the globe \( (d = 10,000 \text{ km}) \), the flow characteristics can be intuitively distinguished, and the details can be fully retained, such as the distribution of vortices in the global perspective. The number of particles decreases when the distance becomes smaller, but the density within the unit geographical range increases. Zooming in the marked area (Figure 12b), the viewing distance is 3500 km, the characteristics and motion process of the Equatorial Current at the mesoscale can be clearly observed. Even in the static screenshot, the North Equatorial Current and the South Equatorial Current near the equator moving from east to west can still be seen using the dynamic streamlet.

Figure 12c shows the flow field effect through close observation of the white box, and the viewing distance is 1500 km. It illustrates that there is still a reasonable streamline density at small and medium scales, and the proposed method well shows the topological structure like vortexes and saddle points.
Figure 12. Rendering results at different viewing distances. (a) is the rendering result at 10,000 km; (b) is the rendering result at 3500 km; (c) is the rendering result at 1500 km; and (d) is the rendering result at 500 km.

3.4. Experiment on Rendering Efficiency

To verify the performance of the proposed method, the rendering efficiency is evaluated using frames per second (FPS) as the benchmark [32]. During iterative rendering, the configuration of width, change rate and frequency will affect the rendering results. Therefore, we set these values as appropriate fixed values according to experience. Rendering rate experiments are conducted from different perspectives, and the basic number is 20,000. Figure 13 shows the stable and interactive states' result, illustrating the method provides a smooth animation efficiently due to the parallel rendering power of GPU and the view frustum culling strategy. The result also indicates that the rendering FPS tends to drop as the camera zooms out; the farther the viewing distance is, the more particles are generated, and more computation will be involved.
4. Conclusions and Future Work

This paper proposes a new method based on particle system to perform multi-hierarchy dynamic visualization of the global ocean current in digital earth. The contribution of this study are as follows:

(1) Firstly, three spherical uniform distribution algorithms are studied and compared. The experimental results show that Marsaglia polar method has more advantages than the other two methods. The probability model is applied to the seed process of particle system, which effectively solves the problem of uneven streamline in the traditional method.

(2) Secondly, we propose a viewport-adaptive density adjustment algorithm. The streamline density changes smoothly to avoid visual jumps and make the multi-resolution representation possible when the camera is zoomed in or out.

(3) Thirdly, we also design a dynamic streamlet pattern. The asymmetry of the head and tail realized by the transparency and width, compared with the conventional streamline method, highlights the flow direction and intensity distribution of the flow field, and improves the occlusion and clutter.

In future work, we will study the multi-scale feature extraction based on global ocean current and build a 3D dynamic streamlet model with lighting and shadows. More than that, we will explore a multi-hierarchy 3D visualization method based on real-time features to depict and simulate ocean current field.

Author Contributions: Conceptualization, Bo Ai; methodology, Qingtong Shi; software, Wenjun Feng; validation, Wenjun Feng and Chenxi Yang; data curation, Yubo Wen; writing—original draft preparation, Qingtong Shi; writing—review and editing, Hongchun Zhu; visualization, Wenjun Feng; All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China [Grant No. 62071279, 41930535]; and the SDUST Research Fund [Grant No. 2019TDJH103].

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to it is provided by the South China Marine Forecasting Centre of State Oceanic Administration. It is only used for experiments due to institutional policy reasons and not disclosed to the public.

Acknowledgments: The authors would like to thank the editors and anonymous reviewers for their meaningful comments and helpful suggestions.

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

1. Klemas, V. Remote Sensing of Coastal and Ocean Currents: An Overview. *J. Coast. Res.* 2012, 28, 576. [CrossRef]

2. Laidlaw, D.H.; Kirby, R.M.; Jackson, C.D.; Davidson, J.S.; Miller, T.S. Comparing 2D Vector Field Visualization Methods: A User Study. *IEEE Trans. Vis. Comput. Graph.* 2005, 11, 12. [CrossRef] [PubMed]

3. Ai, B.; Sun, D.; Liu, Y.; Li, C.; Yang, F.; Yin, Y.; Tian, H. Multi-Scale Representation of Ocean Flow Fields Based on Feature Analysis. *IJGI* 2020, 9, 307. [CrossRef]

4. Sane, S.; Bujack, R.; Garth, C.; Childs, H. A Survey of Seed Placement and Streamline Selection Techniques. *Comput. Graph. Forum* 2020, 39, 785–809. [CrossRef]

5. Huang, J.; Pan, Z.; Chen, G.; Chen, W.; Bao, H. Image-Space Texture-Based Output-Coherent Surface Flow Visualization. *IEEE Trans. Vis. Comput. Graph.* 2013, 19, 1476–1487. [CrossRef] [PubMed]

6. Baker, M.P.; Bushell, C. After the Storm: Considerations for Information Visualization. *IEEE Comput. Grap. Appl.* 1995, 15, 12–15. [CrossRef]

7. Liu, Z. A Prototype Framework for Parallel Visualization of Large Flow Data. *Adv. Eng. Softw.* 2019, 130, 14–23. [CrossRef]

8. Ware, C.; Kelley, J.G.W.; Pilar, D. Improving the Display of Wind Patterns and Ocean Currents. *Bull. Amer. Meteor. Soc.* 2014, 95, 1573–1581. [CrossRef]

9. Reeves, W.T. Particle Systems—A Technique for Modeling a Class of Fuzzy Objects. *ACM Trans. Graph.* 1983, 2, 91–108. [CrossRef]

10. Lobo, M.; Telea, A.C.; Hurter, C. Feature Driven Combination of Animated Vector Field Visualizations. *Comput. Graph. Forum* 2020, 39, 429–441. [CrossRef]

11. Bujack, R.; Middel, A. State of the Art in Flow Visualization in the Environmental Sciences. *Environ. Earth Sci.* 2020, 79, 65. [CrossRef]

12. Rautenhaus, M.; Bottinger, M.; Siemens, S.; Hoffman, R.; Kirby, R.M.; Mirzargar, M.; Rober, N.; Westermann, R. Visualization in Meteorology—A Survey of Techniques and Tools for Data Analysis Tasks. *IEEE Trans. Vis. Comput. Graph.* 2018, 24, 3268–3296. [CrossRef]

13. 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]

14. Wang, S.B.; Pan, Z.G. Comparison, Analysis and Review of 2D Flow Visualization. *J. Syst. Simul.* 2013, 28, 1685–1881.

15. Samsel, F.; Petersen, M.; Abram, G.; Turton, T.L.; Ahrens, J. Visualization of Ocean Currents and Eddies in a High-Resolution Global Ocean-Climate Model. In *Supercomputing 2015*; Association for Computing Machinery: New York, NY, USA, 2015.

16. Zhang, F.; Mao, R.; Du, Z.; Liu, R. Spatial and Temporal Processes Visualization for Marine Environmental Data Using Particle System. *Comput. Geosci.* 2019, 667, 1–8. [CrossRef] [PubMed]

17. Fu, S. Visualization of Global Ocean Flows Based on Particle System. Master’s Thesis, Shandong University of Science and Technology, Qingdao, China, 2015.

18. 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* 2020, 13, 1258–1278. [CrossRef]

19. 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]

20. Ba, Z.; Shan, G.; Liu, J.; Xia, S. A Feature-Based Seeding Method for Multi-Level Flow Visualization. *J. Comput. -Aided Des. Comput. Graph.* 2016, 28, 32–40.

21. Bahaj, A.S. Generating Electricity from the Oceans. *Renew. Sustain. Energy Rev.* 2011, 15, 3399–3416. [CrossRef]

22. Marsaglia, G. Choosing a Point from the Surface of a Sphere. *Ann. Math. Statist.* 1972, 43, 645–646. [CrossRef]

23. Thomas, D.B.; Luk, W.; Leong, P.H.W.; Villasenor, J.D. Gaussian Random Number Generators. *ACM Comput. Surv.* 2007, 39, 11-es. [CrossRef]

24. Sherlock, M.J.; Hasan, M.; Samavati, F.F. Interactive Data Styling and Multifocal Visualization for a Multigrid Web-Based Digital Earth. *Int. J. Digit. Earth* 2021, 14, 288–310. [CrossRef]

25. Li, Z.; Yan, H. Transformation in Scale for Continuous Zooming. In *Manual of Digital Earth*; Guo, H., Goodchild, M.F., Annoni, A., Eds.; Springer Singapore: Singapore, 2020; pp. 279–324. [CrossRef]

26. Ai, T. Dynamic Symbol and Dynamic Map. *J. Wuhan Tech. Univ. Surv. Mapp.* 1998, 01, 49–53.

27. Kettunen, P.; Oksanen, J. Motion of Animated Streamlets Appears to Surpass Their Graphical Alterations in Human Visual Detection of Vector Field Maxima. *Cartogr. Geogr. Inf. Sci.* 2019, 46, 489–501. [CrossRef]

28. Mashima, D.; Kobourov, S.; Hu, Y. Visualizing Dynamic Data with Maps. *IEEE Trans. Vis. Comput. Graph.* 2012, 18, 1424–1437. [CrossRef] [PubMed]

29. Cybulski, P.; Wielebski, L. Effectiveness of Dynamic Point Symbols in Quantitative Mapping. *Cartogr. J.* 2019, 56, 146–160. [CrossRef]

30. Fang, Y.; Ai, B.; Fang, J.; Xin, W.; Zhao, X.; Lv, G. Multi-Scale Flow Field Mapping Method Based on Real-Time Feature Streamlines. *IJGI* 2019, 8, 335. [CrossRef]

31. Keqin, W.; Zhanping, L.; Song, Z.; Moorhead, R.J. Topology-Aware Evenly Spaced Streamline Placement. *IEEE Trans. Vis. Comput. Graph.* 2010, 16, 791–801. [CrossRef] [PubMed]

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]