

# SPATIOTEMPORAL VISUALIZATION OF TIME-SERIES SATELLITE-DERIVED CO<sub>2</sub> FLUX DATA USING VOLUME RENDERING AND GPU-BASED INTERPOLATION ON A CLOUD-DRIVEN DIGITAL EARTH

Sensen Wu<sup>a, b</sup>, Yiming Yan<sup>a, b</sup>, Zhenhong Du<sup>a, b, \*</sup>, Feng Zhang<sup>a, b</sup>, Renyi Liu<sup>a, b</sup>

<sup>a</sup> Zhejiang Provincial Key Laboratory of Geographic Information System, Zhejiang University, Hangzhou 310028, China – wusensegis@zju.edu.cn; duzhenhong@zju.edu.cn

<sup>b</sup> Institute of Geographic Information Science, School of Earth Sciences, Zhejiang University, Hangzhou 310027, China

**KEY WORDS:** Cloud-driven Digital Earth, Volume Rendering, GPU-based Interpolation, Time-series Satellite Data, Ocean Carbon Cycle

## ABSTRACT:

The ocean carbon cycle has a significant influence on global climate, and is commonly evaluated using time-series satellite-derived CO<sub>2</sub> flux data. Location-aware and globe-based visualization is an important technique for analyzing and presenting the evolution of climate change. To achieve realistic simulation of the spatiotemporal dynamics of ocean carbon, a cloud-driven digital earth platform is developed to support the interactive analysis and display of multi-geospatial data, and an original visualization method based on our digital earth is proposed to demonstrate the spatiotemporal variations of carbon sinks and sources using time-series satellite data. Specifically, a volume rendering technique using half-angle slicing and particle system is implemented to dynamically display the released or absorbed CO<sub>2</sub> gas. To enable location-aware visualization within the virtual globe, we present a 3D particle-mapping algorithm to render particle-slicing textures onto geospace. In addition, a GPU-based interpolation framework using CUDA during real-time rendering is designed to obtain smooth effects in both spatial and temporal dimensions. To demonstrate the capabilities of the proposed method, a series of satellite data is applied to simulate the air-sea carbon cycle in the China Sea. The results show that the suggested strategies provide realistic simulation effects and acceptable interactive performance on the digital earth.

## 1. INTRODUCTION

Over the past two or three centuries, greenhouse gases produced by human activities, e.g., carbon dioxide (CO<sub>2</sub>), have become sufficiently widespread and been regarded as the main cause of global warming (Canadell, 2003). The oceans, which play a vital role in the global climatic change, can annually take in approximately 30% of the anthropogenic CO<sub>2</sub> from the atmosphere (Feely, 2004; Millero, 1995). In addition, due to the lack of efficient estimation techniques of the terrestrial carbon, the land-sourced carbon is generally calculated through the estimation results of ocean and atmosphere (Pan, 2014). Therefore, an accurate comprehension of the distribution and variability of ocean carbon is quite important for understanding the basic structure of the global carbon cycle and providing diagnostic insight into climate changes.

Satellite remote sensing, with its large-scale and real-time advantages, has been widely used to evaluate the ocean carbon flux and budget (Liu, 2014; Vasco, 2009). Time-series satellite-derived CO<sub>2</sub> (SateCO<sub>2</sub>) flux products are therefore produced and applied to reveal the patterns of ocean carbon cycle. Visualization techniques are regarded as a powerful manner to analyze and present climate changes and environmental simulations (Liu, 2014; Ziolkowska and Reyes, 2016). In previous studies, the spatial distributions of ocean carbon were usually demonstrated by thematic maps while the temporal variations were displayed by diagrams (Benallal, 2017; Padhy, 2016; Parard, 2016; Schuster, 2013).

The ocean carbon changes through time and space simultaneously. Traditional methods, however, can hardly express its spatiotemporal dynamic process at multi-scales since its characteristics of time and space are clarified separately.

Volume rendering technology, which explores the inner structure of three-dimensional (3D) data and displays the spatiotemporal dynamics of four-dimensional (4D) data by adding time attribute, is an important branch of computer graphics. It has been widely applied in various fields to facilitate visual analytics and simulations. The SateCO<sub>2</sub> flux data that we focus on in this paper is essentially a time-space dataset, and volume rendering techniques can be utilized for its 4D visualization to dynamically demonstrate the evolution of CO<sub>2</sub> gas as was done by Du et.al (2015).

However, the behaviors of the ocean carbon cycle are controlled by various processes in global ocean environment (Canadell, 2003). Although the method suggested by Du (2015) has achieved visually realistic simulation of the ocean carbon fluxes, their study mainly focused on regions. The role of these regions in the global carbon cycle system has yet to be discussed.

After the concept of ‘Digital Earth’ was presented by Gore in 1998, different commercial virtual globe platforms, e.g., Google Earth and World Wind, were developed and have continued to evolve over recent years (Gore, 1998; Liu, 2015). Virtual globes provide a global coordinating framework to visualize large-scale spatial datasets and to simulate geographic phenomena. These globe platforms have been proved to be

\* Corresponding author

useful for climate and marine researches (Brovelli, 2016; Duarte, 2016; Li, 2011; Li, 2011; Sarthou, 2015; Shupeng and van Genderen, 2008).

Considering the regional and global characteristics of the ocean carbon cycle, utilizing volume rendering techniques to visualize SateCO<sub>2</sub> flux data on the virtual globe would be an improved and much powerful solution for exploring the spatiotemporal dynamics of ocean carbon sources and sinks. Due to the fact that current commercial digital earth platforms do not directly support volume visualization, attempts were carried out to realize volume rendering by extending these platforms (Li, 2011; Liang, 2014; Liu, 2014; Liu, 2015) or developing their own digital earth (Chen, 2012; Li, 2011; Sarthou, 2015). However, studies on presenting the spatiotemporal variations of ocean carbon on the globe using volume rendering are still rare and needed further research. Also, volume rendering using time-series SateCO<sub>2</sub> flux data concerns GPU-based real-time data processing, which requires great efforts to be integrated into commercial virtual globes.

Therefore, a digital earth platform is developed in this article to support the interactive analysis and display of multi-geospatial data, and a framework of interactive volume rendering on the virtual globe of 4D volume data generated from SateCO<sub>2</sub> flux data is designed and realized. In conclusion, we have four main objectives in this paper: (1) to design a cloud-based data storage pattern for managing multi-type data and providing distributed computing capability for data analysis; (2) to develop a digital earth platform to support multivariate analysis and visualization in a cloud-based environment; (3) to dynamically create smooth CO<sub>2</sub> flux particles among space and time with a GPU-based spatiotemporal interpolation approach; and (4) to realize interactive volume rendering for the dynamic visualization of the ocean carbon flux and budget on the virtual globe.

The remainder is organized as follows. Section 2 introduces the spatiotemporal visualization framework of our research. In Section 3, we provide a detailed implementation of our approach, which includes a multi-source data store based on the cloud, a digital earth to support multivariate visualization, a technique to dynamically generate smooth 4D particles, and a rendering method of volumetric data on the globe. Section 4 demonstrates the proposed approach using time-series SateCO<sub>2</sub> flux data and discusses the visualization results over conventional methods. Conclusions and future works are presented in Section 5.

## 2. SPATIOTEMPORAL VISUALIZATION FRAMEWORK ON THE DIGITAL EARTH

As is shown in Figure 1, the framework of our spatiotemporal visualization on the digital earth consists of three parts based on the theory of modularized design. At the top is a data cloud that manages multi-type data. In the middle layer is a digital earth, where data analysis and visualization resides. The bottom layer is a GPU-based interactive volume rendering model for spatiotemporal visualization of time-series SateCO<sub>2</sub> flux on the virtual globe.

Specifically, the data cloud is implemented by combining PostgreSQL-PostGIS (PPGIS) with Hadoop distributed file system (HDFS), along with a distributed computing engine developed to improve the data analysis efficiency. The digital earth composes of an earth engine that constructs the fundamental terrain globe, a viewer controller for processing the handles of graphical user interface (GUI) and the earth engine, and a renderer engine that supports various rendering methods for spatiotemporal analysis and visualization. Notably, the renderer engine is able to link and execute external GPU

fragment code, such as compute unified device architecture (CUDA) kernels and volume rendering shaders. Therefore, GPU-based interactive volume rendering models using texture slicing and particle system can be implemented on the renderer engine. The detailed implementation of the framework is given in Section 3.

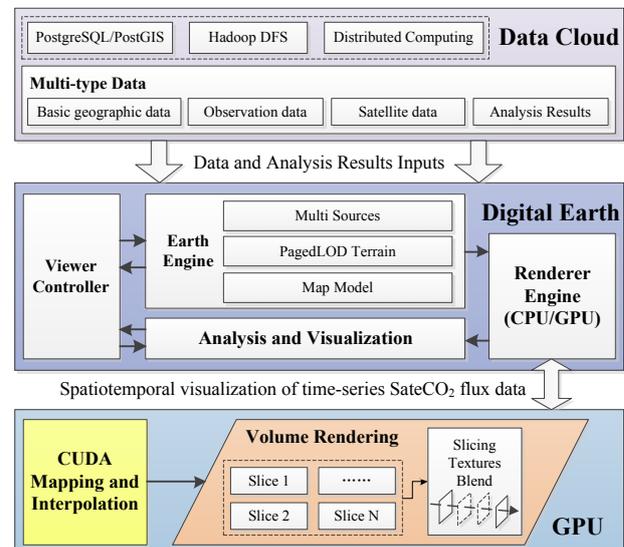


Figure 1. Spatiotemporal visualization framework on the digital earth

## 3. IMPLEMENTATION

### 3.1 Cloud-based data management

In order to organize and manage multi-source data of various types in our study, a hybrid data store is designed and implemented by combining PPGIS with HDFS (Figure 2). PostgreSQL, along with PostGIS, which provides spatial objects for the PostgreSQL database, is employed to address the structured data while HDFS, with its advantage of high scalability and fault-tolerance, is applied to manage the volume semi-structured or unstructured data.

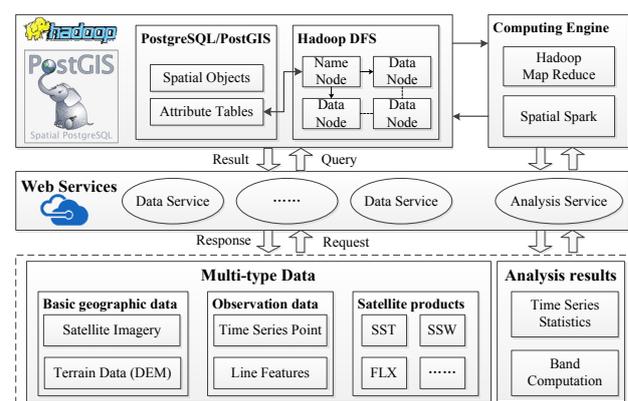


Figure 2. Cloud-based data management

The diverse data in our research can be divided into three categories: marine observation data, marine satellite-derived data, and basic geographic data. Marine observation data includes time-series fixed point attribute data, (e.g., real-time ecological buoy recording nutrient parameters at a fixed location) and line feature data representing measurement attributes along a linear feature (e.g., ship observation data

along monitoring sections). Given that marine observation data is mainly made up of records with spatial properties, these data are handled in PPGIS. Marine satellite-derived data includes various product types, such as sea surface wind (SSW), sea surface temperature (SST), chlorophyll-a (CHL) and air-sea CO<sub>2</sub> flux (FLX). It is commonly organized in data grids and is known as a classic type of unstructured data, and is therefore distributedly stored in HDFS. Basic geographic data, as the foundational spatial information to assist users to cognize geographic locations and spatial correlations such as satellite imagery, terrain data and place mark data, is structured in custom image tile format and also managed by HDFS for distributed searching and loading. Moreover, a computing engine using Hadoop MapReduce and Spatial Spark based on the data cloud is developed to enhance

the performance of complex spatiotemporal calculations, such as long time-series statistics and complicated band computation algorithm. Multiply data requests and analysis are done through web services.

### 3.2 A digital earth to support multivariate visualization

Our digital earth is primarily developed in C++ and OpenSceneGraph (OSG). The osgEarth library is integrated to address geographic features while the GUI widgets are designed by Qt. Libraries are adopted for specific usages, e.g., GDAL for the read-write of spatial data. The structure of the digital earth is shown in Figure 3.

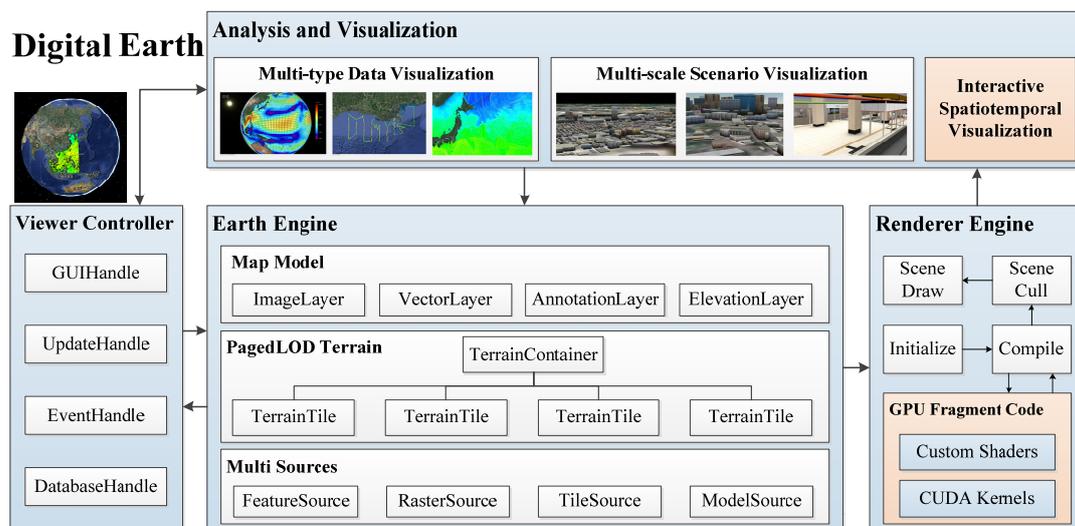


Figure 3. Digital earth implementation architecture

The core module is an earth engine that organizes multiple data and establishes the foundational terrain globe based on a paged level of detail (LOD) map model. The viewer controller is in charge of responding to the GUI and event handles, and managing the earth engine accordingly, such as loading or deleting data, updating display settings etc. The graphic shaders for rendering the data map of the earth engine are addressed by a renderer engine. Based on GLSL (OpenGL shading language) and CUDA, custom shaders and GPU-parallel kernels can be compiled with the renderer engine to realize customized volume rendering for visualizing ocean carbon cycle. This will be discussed detailedly in Section 3.3.

The MP terrain engine of osgEarth is used to construct the terrain globe, and GDAL and feature\_geom plugins are adopted to display various spatial data, including the digital elevation model (DEM), satellite imagery, satellite-derived products, spatial vectors etc. These plugins are extended to support multiple symbolic rendering of geospatial data, and the visualization effects on the globe are shown in Figure 4.

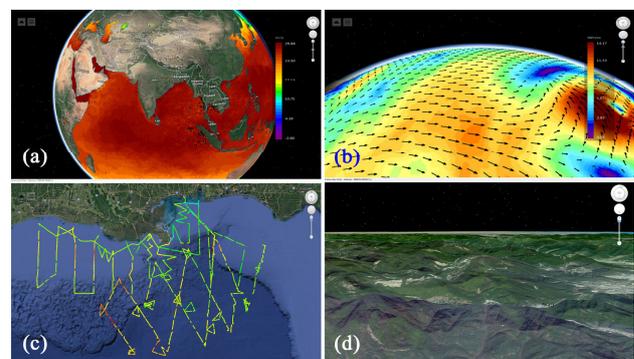


Figure 4. Various spatial data are displayed on the virtual globe, (a) satellite-derived SST product; (b) satellite-derived SSW product; (c) *in situ* data of salinity; (d) satellite imagery and DEM data

Using the distributed computing engine in the data cloud, computing-intensive spatiotemporal analysis functions are also provided in the digital earth. For example, the statistical results of long time-series point analysis of daily CHL products from 2010 to 2015 can be obtained promptly within an acceptable time (Figure 5).

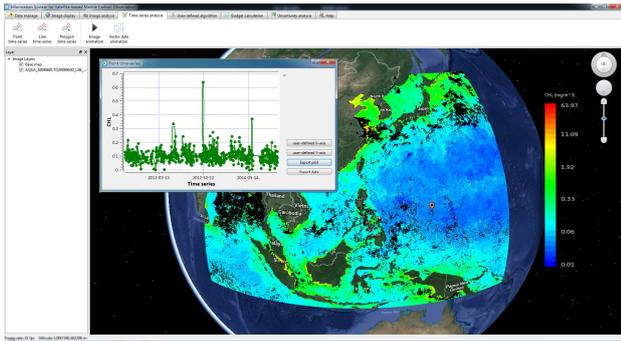


Figure 5. Time-series point statistics of the daily CHL products from 2010 to 2015

To evaluate the capacity of the distributed computing engine, performance tests of the time-series statistics were conducted. The computing engine was deployed on a data cloud implemented by Hadoop 2.7.0 and Spark 2.0.1 with 8 Dell R720 servers. As we can see in Figure 6, compared to the conventional one single server solution, our cloud solution responded much more promptly with less execution time during the increase of data amount. In addition, when the time length of data reaches 16 years, the efficiencies of time-series point, line, and polygon statistics have improved by almost 4, 5, and 7 times respectively, which further demonstrates the performance of the computing engine.

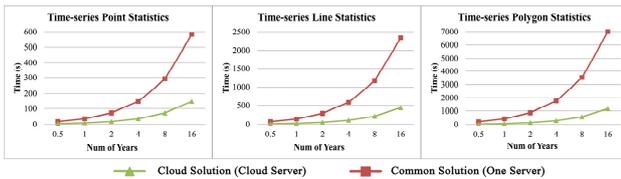


Figure 6. Comparison of time-series statistics between cloud solution and common solution

### 3.3 Location-aware interactive volume rendering of CO<sub>2</sub> flux

By utilizing the renderer engine that we put forward in the digital earth from last chapter, we proposed an interactive volume rendering method implemented on the globe for the time-series SateCO<sub>2</sub> flux data (Figure 7).

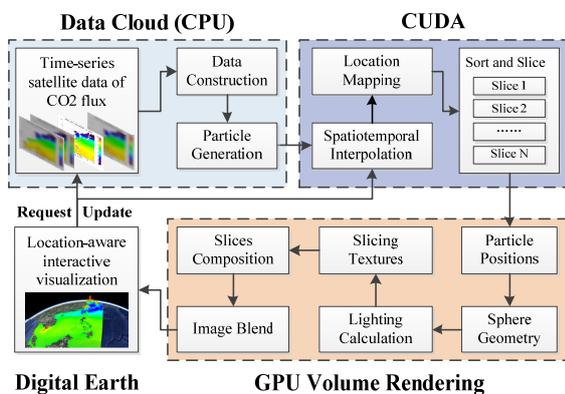


Figure 7. Location-aware interactive volume rendering of CO<sub>2</sub> flux

In order to render CO<sub>2</sub> gas, we first design a data model to generate particle systems from the CO<sub>2</sub> flux products. A spatiotemporal interpolation technique using CUDA kernels is

then implemented to dynamically obtain spatiotemporal smooth flux particles. The particles are mapped through parallel coordinate transformation in GPU to be location-aware on the digital earth. Rather than directly rendering polygon slices from the 3D volume, a half-angle rendering technique proposed by Green (2008) is adopted and extended in our research. Finally, the slicing textures are composed and blended into the virtual globe.

#### 3.3.1 Modeling of time-series SateCO<sub>2</sub> flux data

Generally, the sea-water CO<sub>2</sub> consists of riverine fluxes and air-sea interface fluxes. In this paper, the SateCO<sub>2</sub> flux data denotes the air-sea CO<sub>2</sub> fluxes. When the flux value is positive, CO<sub>2</sub> is emitted from water to air; otherwise, the water absorbs CO<sub>2</sub>. The amount of stored or released carbon is quantified by carbon budget, which can be calculated by integrating carbon flux among a specific time and space. The positive or negative budget value indicated a carbon source or a carbon sink. According to the definition, the carbon budget (B) within an area S over time T is:

$$B = \int_0^T \int_0^S Flux(x, y, t) ds dt \quad (1)$$

where (x, y) is the position in S, and t is the time in T. To render CO<sub>2</sub> flux, a data model proposed by Du (2015) is adopted to construct the air-sea SateCO<sub>2</sub> flux data to a particle system. The initial position of each particle is generated from a random function and is located around its raster grid cell center. The velocity of each particle is proportional to the CO<sub>2</sub> flux value. The motion direction (up or down) of particles is determined by the flux value (positive or negative). Since the cumulative gas volume is distinguished by particle velocity, the carbon budgets can be represented by the depth or height of particles. In addition, the variations of the fluxes are demonstrated by color and size changes of particles. In our study, a simple linear mapping transfer function is adopted to convert the flux to particle color and size (Equation 2).

$$Particle (color, size) = transfer(Flux) \quad (2)$$

#### 3.3.2 Spatiotemporal interpolation using CUDA

In this research, the current SateCO<sub>2</sub> fluxes are usually monthly average products, the time span of which would result in temporal discontinuity during image presentation. Temporal interpolation between time-adjacent images is hence required. In the meantime, due to cloud covering or observational constraints, some regions of satellite products are missing. To achieve a better visualization result, spatial interpolation techniques are applied to fill the missing parts. However, both spatial and temporal interpolations require intensive computation and substantial memory, which is unpractical for large-scale geospatial data. Moreover, dynamic spatiotemporal interpolation in real-time rendering is not yet supported by most volume rendering algorithms. Therefore, to acquire a smooth rendering result, we present an on-the-fly spatiotemporal interpolation method using CUDA kernels.

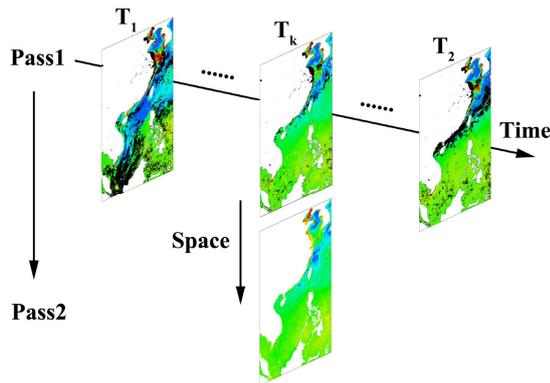


Figure 8. Spatiotemporal interpolation using CUDA

In the rendering stage, the flux value of each frame is calculated by its two time-adjacent satellite-derived products. As displayed in Figure 8, the flux values of the same point in the two satellite images are  $V_1$  and  $V_2$  at time  $T_1$  and  $T_2$ , so the flux value  $V_k$  at time  $T_k$  can be obtained using a linear mapping method (Equation 3):

$$V_k = V_1 + \frac{V_2 - V_1}{T_2 - T_1} \times (T_k - T_1) \quad (3)$$

To fill the missing region during real-time rendering, a GPU-based inverse distance weighting (IDW) algorithm is adopted in our study using CUDA parallelism. IDW algorithm was first proposed by Chaboissier (1968) and is widely used in geosciences for spatial interpolation. It computes the predicted value of an interpolated location via weighting average of sample values.

Since shared memory is much faster than global memory in the CUDA architecture, an optimization strategy that uses shared memory is applied in our research to implement the IDW interpolation (Mei, 2014). The detailed procedure is as follows:

- (1) The coordinates of flux particles in global memory are transported to shared memory.
- (2) All threads from one block load the coordinates of each particle and compute the distances and inverse weights to other particles that are stored in current shared memory.
- (3) The partial weights and weighted values of all particles are utilized to calculate the prediction value of unknown particles.
- (4) Predicted values are written back into global memory.

By making full use of the hardware's capability for operating both temporal and spatial mapping in CUDA, the particle velocity can be recalculated at runtime when the view direction changes.

### 3.3.3 Location-aware mapping on the virtual globe

Two types of coordinate systems are used in our digital earth platform, i.e., geographic coordinate system (GCS) and Cartesian coordinates system (CCS). A CCS represents coordinates in 3D space through a tuple set  $(X, Y, Z)$  (Figure 9). A GCS is a widely-used spherical coordinate system in geosciences, which depicts spatial locations by latitude, longitude, and altitude (Figure 9).

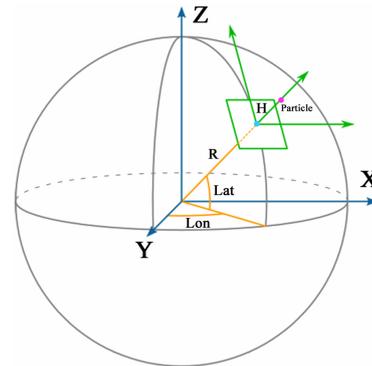


Figure 9. Cartesian coordinate system and geographic coordinate system

Due to the fact that GPU rendering pipeline does not support spherical coordinates, the flux particles, which are originally organized by GCS, require real-time transformation from GCS to CCS in the rendering stage (Equation 4).

$$\begin{cases} X = (R + H) \times \cos(lat) \times \sin(lon) \\ Y = (R + H) \times \cos(lat) \times \cos(lon) \\ Z = (R + H) \times \sin(lat) \end{cases} \quad (4)$$

where  $(X, Y, Z)$  is the coordinates of CCS,  $R$  is the earth's radius, and  $H, lon, lat$  respectively denote the altitude (height), longitude, and latitude of GCS. Since the transformations of different particles are independent, parallel processing in CUDA can be easily implemented to improve the efficiency.

### 3.3.4 Interactive volume rendering

Volume rendering using particle system is employed in this research, which decouples rendering from simulation and enables high quality rendering even from lower resolution grids (Cohen, 2010). Rather than directly rendering polygon slices from the 3D volume, a half-angle rendering technique proposed by Green (2008) is adopted and extended to be applicable on the digital earth. We first project the particles onto the virtual globe and sort them along a half-angle axis, and render them in slicing batches with shadow effects. The slicing textures are then composed to form the final image, which is blended with the virtual globe and is displayed on the screen in the end.

This technique obtains translucent and stereoscopic  $CO_2$  gas with color and shadow effects on the globe, yet it still manages to accelerate the rendering process significantly by using a single 2D shadow buffer and avoiding repeated particle information visiting.

In the implementation, the key idea is to compute the axis which is half way from the light direction to view direction. The rendering direction is determined by the relative orientation between the viewer and the light.

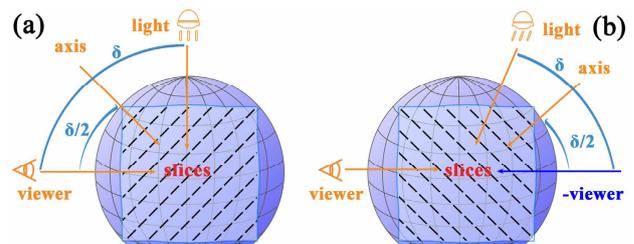


Figure 10. Half-angle axis calculation based on the viewer and light directions on the virtual globe

When the angle of the viewer and the light is less than 90° (Figure 10a), the slices are rendered from front-to-back through the viewer point; otherwise (Figure 10b), the slices are rendered from back-to-front through the negative viewer direction.

The rendering process can be represented as follows:

- (1) Generate or update the location and color of the flux particles using spatiotemporal interpolation in CUDA.
- (2) Map the coordinates of the particles to the digital earth coordinate system.
- (3) Calculate the half-angle vector and sort the particles along the vector.
- (4) Project the particles of each slice onto the slice plane and render them from the viewpoint of both the viewer and the light.
- (5) Compose slicing textures to form the final image and blend it with the virtual globe.
- (6) Repeat the above rendering steps to achieve a dynamic simulation of the ocean carbon cycle.

#### 4. RESULTS AND DISCUSSION

The framework proposed in Section 2 is implemented in C++. OSG 3.2.0, osgEarth 2.7.0, and GDAL 1.10.1 library are used for rendering and loading data. The spatiotemporal interpolation of the flux particles and coordinate transformation are speeded up by CUDA 8.0. The air-sea SateCO<sub>2</sub> flux products organized in HDF5 format are provided by the Second Institute of Oceanography (SIO), State Oceanic Administration of China (SOA). The monthly average products of the air-sea SateCO<sub>2</sub> flux in 2009 covering the whole China Sea are chosen for the simulation in this research. Detailed information of the data is given in Table 1.

Table 1. Summary of the simulated air-sea SateCO<sub>2</sub> flux data

Data Property	Value
Data Dimension	1800 × 2460 × 12
Latitude Extent	0~41° N
Longitude Extent	100~130° E
Time Extent	2009/01/01~2009/12/31
Spatial Resolution	0.0167°
Temporal Resolution	Monthly
Uncompressed Data Size	480 MB

#### 4.1 Visualizations

Based on the spatiotemporal visualization framework, the monthly average products of air-sea SateCO<sub>2</sub> flux in 2009 are utilized to generate the 4D volume particle data. The interactive volume rendering of the air-sea CO<sub>2</sub> flux and carbon budget on the virtual globe is realized. In addition, both CPU and GPU-based spatiotemporal interpolations of flux particles are implemented in the virtual globe environment, and the performances of the two methods are compared as shown in Figure 11. When the 4D smooth flux particles are created within CPU, the average frame rate achieves at about 15 frames per second (FPS) and varies violently during the rendering. While, the average frame rate of the GPU-based implementation acquires relatively higher FPS and maintains at about 30 FPS benefited from the parallel generation of the flux particles. It demonstrates that the spatiotemporal interpolation strategy using GPU yields considerable improvements in the rendering performance.

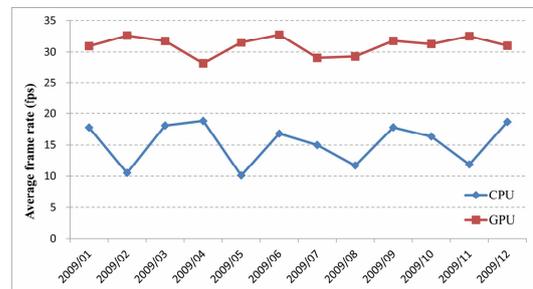


Figure 11. Performance comparisons of CPU and GPU-based spatiotemporal interpolations in rendering

Figure 12 shows snapshots of different visualization results of air-sea SateCO<sub>2</sub> flux products. The original method projects and renders satellite data on the globe by a fixed colorbar from blue to red with the missing parts marked in black, as shown in Figure 12a. Figure 12b, 12c and 12d show three new rendering modes of flux particles, i.e., points, point sprites, and half-angle volume rendering. The rendering of point sprites is implemented by a geometry shader, and volume rendering is performed based on the results of point sprites.

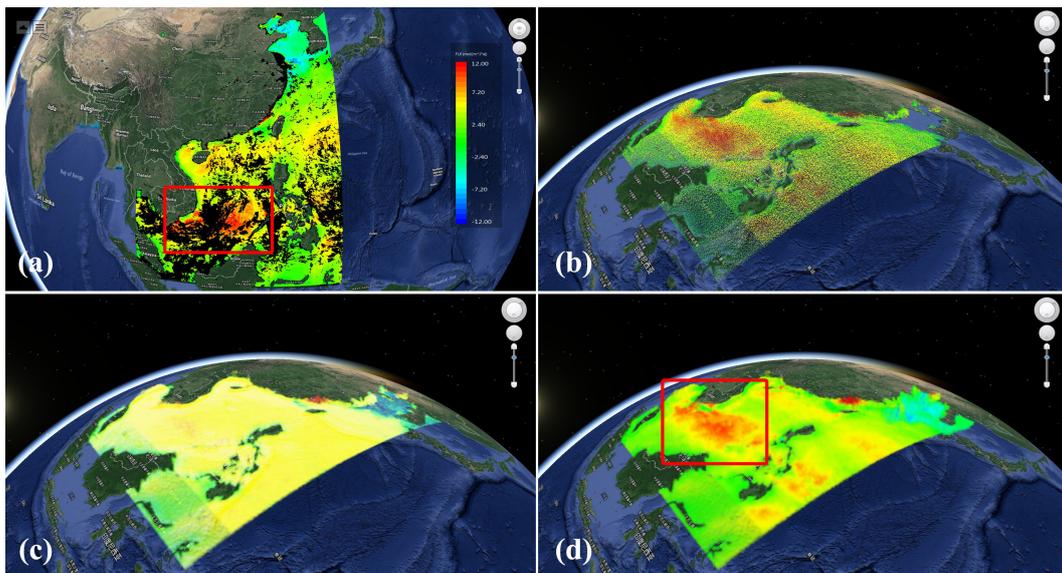


Figure 12. The original satellite data (a) and CO<sub>2</sub> flux particles rendering with points (b), point sprites (c) and volume rendering (d) on the globe

Figure 13 and Figure 14 show the dynamic simulation of air-sea CO<sub>2</sub> flux in the China Sea from August to December in 2009 using different rendering methods. In Figure 13, the evolution of CO<sub>2</sub> flux over time is shown by the color difference among multiple images. Meanwhile, the spatial locations and correlations are easily acquired from the GCS of the virtual globe. Nevertheless, it can be seen that Figure 14 provides more characteristics and presents information in greater details. In addition to displaying the instant air-sea CO<sub>2</sub> flux with color change, the carbon budget accumulated since January is represented by the altitude or depth of the particles. The variations of ocean carbon sources or sinks are therefore easily recognized from the continuous visualization results. For example, the carbon source in the South China Sea (Region 1 in Figure 14) releases more CO<sub>2</sub> from August to September, and the release drops down slowly on October and turns to absorb CO<sub>2</sub> on November. Region 2 of the Yellow Sea and the Bohai Sea, however, shows as a large carbon sink consistently from October to December. Besides, some novel phenomena are observed from the results. For instance, although main areas of the Bohai Sea act as carbon sinks in winter, its northern part

(Region 3 in Figure 14) presents as strong carbon sources and releases vast amounts of CO<sub>2</sub> with considerably high altitude (carbon budget).

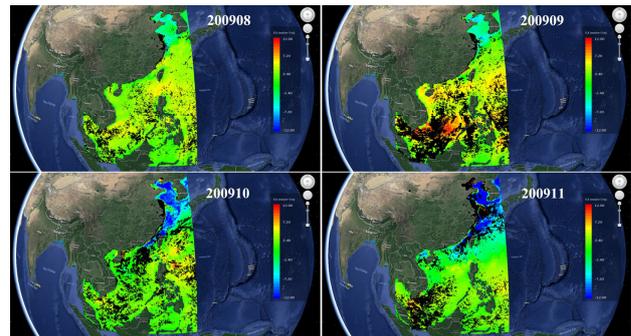


Figure 13. Air-sea CO<sub>2</sub> flux dynamic simulation over time represented by time-series satellite data.

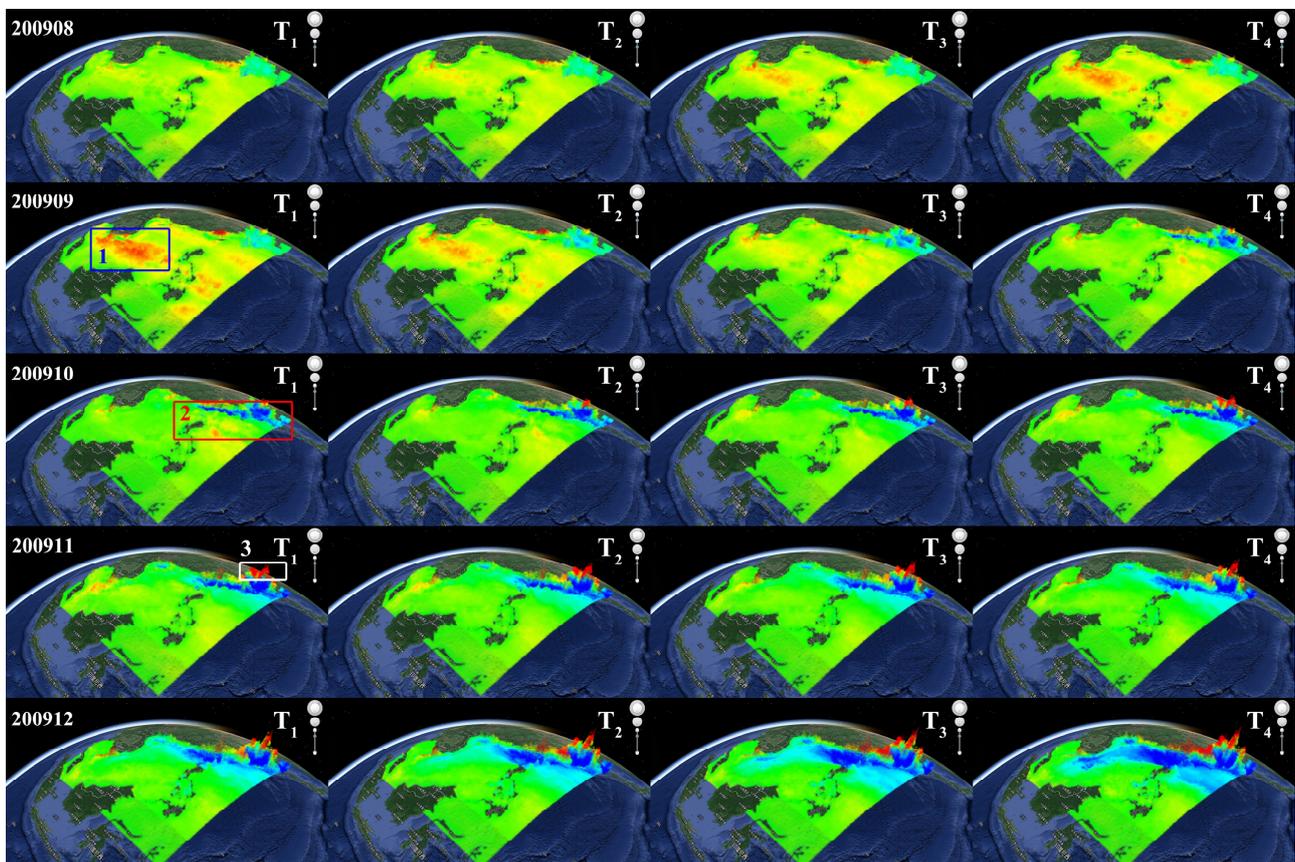


Figure 14. Air-sea CO<sub>2</sub> flux dynamic simulation over time represented by interactive volume rendering with real-time spatiotemporal interpolation

#### 4.2 Comparison and analysis

Compared to previous approaches, our method has several advantages. First, time-series SateCO<sub>2</sub> products are fully utilized by transforming to 4D dataset, and projecting onto a virtual globe so that the mechanisms of ocean carbon cycle are better revealed. Second, the simulation is realized in a virtual globe environment, thus, the locations and spatial relationships are recognizable. In addition, numerous auxiliary geographic data provided on the globe, such as DEM, satellite imagery,

place-markers etc., assist the user to better understand the results. Furthermore, besides expressing the CO<sub>2</sub> flux information by color changes, the carbon budget is also displayed in the altitude dimension.

Moreover, our dynamic temporal interpolation strategy overcomes the restriction of not able to continuously express the revolution over time (Figure 13), and achieves smooth visualization effects on the virtual globe (Figure 14, T<sub>1</sub> to T<sub>4</sub>). Similarly, the missing regions of satellite data are parallelly mapped in the GPU such that the entirely spatial characteristics

of the ocean carbon cycle are visualized. For example, the missing regions of the red frame in Figure 12a are filled smoothly in Figure 12d.

## 5. CONCLUSIONS

The objective of our study is to propose a spatiotemporal visualization framework for dynamic simulation of the ocean carbon cycle using time-series SateCO<sub>2</sub> flux data in a virtual globe environment. First, a cloud-based data store is designed to organize and manage multi-source data of various types. Then, a digital earth platform is developed to support the interactive analysis and display of multi-geospatial data on the cloud. After that, an interactive volume rendering method implemented by half-angle slicing and particle system is adopted to achieve spatiotemporal visualization of air-sea CO<sub>2</sub> flux on the globe. The spatial distributions of CO<sub>2</sub> flux are shown by color changes while the variability and patterns of the carbon sinks and sources are indicated in the altitude dimension. In addition, a spatiotemporal interpolation method using CUDA during real-time rendering is designed to obtain smooth effects on both spatial and temporal scales. Our experiments illustrate that the suggested framework achieves considerably high FPS rates (approximately 30 FPS) and visually realistic rendering effects. Strategies that allow efficient volume rendering and dynamical spatiotemporal interpolation are applicable and extensible to the simulation of similar phenomena in geosciences. Still, there are several issues that require consideration in future works. Although our digital earth is developed on the cloud computing architecture, the distributed computing ability is only applied to data analysis. The effectiveness of rendering methods using cloud computing capacity needs further study. In addition, the ocean carbon cycle is a rather complicated process, and its simulation and mechanism exploration need the establishment of real marine environment (e.g., sea surface and undersea environment) and the integration of more environment parameters (e.g., SSW and ocean current).

## REFERENCES

- Benallal, M.A. et al., 2017. Satellite-derived CO<sub>2</sub> flux in the surface seawater of the Austral Ocean south of Australia. *International Journal of Remote Sensing*, 38(6): 1600-1625.
- Brovelli, M.A., Hogan, P., Prestifilippo, G. and Zamboni, G., 2016. NASA Webworldwind: Multidimensional Virtual Globe for Geo Big Data Visualization. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLI-B2: 563-566.
- Canadell, J.G., Dickinson, R., Hibbard, K., Raupach, M. and Young, O., 2003. Global Carbon Project: The Science Framework and Implementation. *Environmental Policy Collection*, E86D(11): 2380-2389.
- Chaboissier, M.C. et al., 1968. A two-dimensional interpolation function for irregularly-spaced data, *ACM National Conference*, pp. 517-524.
- Chen, G., Li, B., Tian, F., Ji, P. and Li, W., 2012. Design and implementation of a 3D ocean virtual reality and visualization engine. *Journal of Ocean University of China*, 11(4): 481-487.
- Du, Z., Fang, L., Bai, Y., Zhang, F. and Liu, R., 2015. Spatio-temporal visualization of air-sea CO<sub>2</sub> flux and carbon budget using volume rendering. *Computers & Geosciences*, 77(C): 77-86.
- Duarte, E., Bordonhos, P., Dias, P. and Santos, B.S., 2016. *Living Globe: Tridimensional Interactive Visualization of World Demographic Data*. Springer International Publishing.
- Feely, R.A. et al., 2004. Impact of anthropogenic CO<sub>2</sub> on the CaCO<sub>3</sub> system in the oceans. *Science*, 305(5682): 362-6.
- Gore, A., 1998. *The Digital Earth : Understanding our planet in the 21th Century*. *Photogrammetric Engineering & Remote Sensing*, 65(2): 5-5.
- Green, S., 2008. Volumetric particle shadows. *Nvidia Developer Zone* <http://developer.download.nvidia.com>.
- Li, J., Wu, H., Yang, C., Wong, D.W. and Xie, J., 2011. Visualizing dynamic geosciences phenomena using an octree-based view-dependent LOD strategy within virtual globes. *Computers & Geosciences*, 37(9): 1295-1302.
- Li, W., Chen, G., Kong, Q., Wang, Z. and Qian, C., 2011. A VR-Ocean system for interactive geospatial analysis and 4D visualization of the marine environment around Antarctica. *Computers & Geosciences*, 37(11): 1743-1751.
- Liang, J., Gong, J., Li, W. and Ibrahim, A.N., 2014. Visualizing 3D atmospheric data with spherical volume texture on virtual globes. *Computers & Geosciences*, 68: 81-91.
- Liu, P., Gong, J. and Yu, M., 2014. Graphics processing unit-based dynamic volume rendering for typhoons on a virtual globe. *International Journal of Digital Earth*, 8(6): 1-20.
- Liu, P., Gong, J. and Yu, M., 2015. Visualizing and analyzing dynamic meteorological data with virtual globes: A case study of tropical cyclones. *Environmental Modelling & Software*, 64(C): 80-93.
- Liu, Q. et al., 2014. Estimating dissolved organic carbon inventories in the East China Sea using remote - sensing data. *JOURNAL OF GEOPHYSICAL RESEARCH-OCEANS*, 119(10): 6557-6574.
- Mei, G., 2014. Evaluating the Power of GPU Acceleration for IDW Interpolation Algorithm. *Scientific World Journal*, 2014(8): 171574.
- Millero, F.J., 1995. Thermodynamics of the carbon dioxide system in the oceans. *Geochimica Et Cosmochimica Acta*, 59(4): 661-677.
- Padhy, P.C., Nayak, R.K. and Dadhwal, V.K., 2016. Estimation of Partial Pressure of Carbon Dioxide and Air-Sea Fluxes in Hooghly Estuary Based on In Situ and Satellite Observations. *Journal of the Indian Society of Remote Sensing*, 44(1): 1-9.
- Pan, D., Liu, Q. and Bai, Y., 2014. Review and suggestions for estimating particulate organic carbon and dissolved organic carbon inventories in the ocean using remote sensing data. *Acta oceanologica sinica*, 33(1): 1-10.

Parard, G., Charantonis, A.A. and Rutgersson, A., 2016. Using satellite data to estimate partial pressure of CO<sub>2</sub> in the Baltic Sea. *Journal of Geophysical Research Biogeosciences*, 121(3): 1002-1015.

Sarthou, A., Mas, S., Jacquin, M., Moreno, N. and Salamon, A., 2015. Earthscape, a Multi-Purpose Interactive 3d Globe Viewer for Hybrid Data Visualization and Analysis. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-3/W3: 487-493.

Schuster, U., Mckinley, G.A., Bates, N. and Chevallier, F., 2013. An assessment of the Atlantic and Arctic sea-air CO<sub>2</sub> fluxes, 1990-2009. *Biogeosciences*, 10(1): 607-627.

Shupeng, C. and van Genderen, J., 2008. Digital Earth in support of global change research. *International Journal of Digital Earth*, 1(1): 43-65.

Vasco, D.W. et al., 2009. Satellite-based measurements of surface deformation reveal fluid flow associated with the geological storage of carbon dioxide. *Geophysical Research Letters*, 37(3): 174-180.

Ziolkowska, J.R. and Reyes, R., 2016. Geological and hydrological visualization models for Digital Earth representation. *Computers & Geosciences*, 94: 31-39.