MeteoExplorer: Visual data analysis software for atmospheric and geoscientists

Lianqing Yu

National Meteorological Center, Beijing, China

Correspondence
Lianqing Yu, National Meteorological Center, No. 46 Zhongguancun South Avenue, Beijing, 100081, China. Email: lianqing.yu@hotmail.com; yulq@cma.gov.cn

Abstract
Over recent years, several visualization tools that offer advanced three-dimensional (3D) rendering techniques, while still showing good performance for large data sets, have been developed. However, how to provide an intuitive way to combine traditional two-dimensional (2D) methods essential in operational weather forecasting and new 3D techniques welcomed in atmospheric research is not fully investigated, if not overlooked, in these tools. The paper presents MeteoExplorer: a software tool designed to analyse and visualize atmospheric and geoscience data. It consists of a developer-oriented cross-platform foundation framework and several applications built upon the framework. The framework is developed with C++ on the central processing unit side and shader languages on the graphics processing unit (GPU) side. Modules in the framework are loosely coupled with minimum interdependencies to facilitate code reuse and improve code efficiency. Compared with other software tools, the framework provides distinctive features, including a unified framework that integrates both 2D and 3D rendering context, comprehensive support of popular data formats and computing platforms, decent rendition quality via the export of vector graphics and page layout customization, and interactive composition of synoptic charts. Based on the foundation libraries, the software can develop web, desktop and mobile applications for meteorological purposes. These applications support the analysis and visualization of numerous data set types, as well as the interactive content creation. MeteoExplorer is an exploratory project that provides forecasters with new techniques for climate diagnosis and forecasts.

Keywords
atmospheric science, data visualization and analysis, geoscience, weather forecast

1 INTRODUCTION

With the rapid progress in numerical weather prediction research (Warner, 2011) and the ever-increasing processing capabilities of computing resources, the size of data sets generated by numerical models is growing unprecedentedly. Furthermore, spatial and temporal resolutions of deterministic numerical models are
The paper is organized as follows. A concise review of recent work in visualization and analysis software development is presented in Section 2. An overview of MeteoExplorer is detailed in Section 3, where the architecture and key technologies are discussed. Section 4 focuses on the applications of the developed software in weather forecasting. A feature comparison of MeteoExplorer with other visualization packages is provided in Section 5. Section 6 concludes.

2 | RELATED WORK

Numerous applications have been developed to meet the huge demand for analysing and visualizing large data sets. These applications can be classified into three categories based on potential users. The workstation-class systems are developed for operational forecast environments, and include MetView from the European Center for Medium-Range Weather Forecasts (ECMWF) (Russell et al., 2010), the National Oceanic and Atmospheric Administration (NOAA) Advanced Weather Interactive Processing System (AWIPS) (McCaslin et al., 2000), the Ninjo from Deutscher Wetterdienst (DWD) (Joe et al., 2005), Meteo-France’s Synergie-Next, and the Meteorological Information Comprehensive Analysis and Process System (MICAPS) (Yu and Hu, 2011) of the China Meteorological Administration (CMA). These systems are designed for ad hoc prescheduled operations. Thus, they generally provide a streamlined user interface and do not necessarily offer a wide range of features.

The second category includes software tools that target researchers and professionals and are either freely available in the public domain or can be purchased via a commercial license. The Grid Analysis and Display System (GrADS) (Doty and Kinter, 1995) and NCAR Command Language (NCL) are two script-programming-based tools whereby users can perform data calculations and plotting tasks by writing and executing codes. The Integrated Data Viewer (IDV) (Murray et al., 2009), developed by Unidata, is a Java-based cross-platform application for the geoscience community. The IDV supports a wide range of data sets and allows users to manipulate them within a unified interface. The Vis-5D (Hibbard et al., 1994) is an additional interesting tool that pioneered the 3D visualization of scientific data, dating from the 1990s. In particular, it was employed by the NOAA Forecast Systems Laboratory to develop the AWIPS 3D display capabilities. Wang (2014) developed a geographical information system (GIS)-based application denoted MeteoInfo for the visualization and analysis of meteorological data. MeteoInfo supports several commonly used meteorological data formats and offers a library framework and Python-based scripting capabilities for users to analyse their data.
Several graphics processing unit (GPU)-based systems that offer 3D rendering capabilities have been developed with the recent increase in the general availability of powerful consumer graphics hardware. Hollt et al. (2014) proposed a GPU-based interactive visualization system for the exploration and analysis of ensemble simulation data. They successfully applied the results of uncertainty analysis to plan the placement and operation of offshore structures, as well as the path planning of underwater gliders. In addition, Clyne et al. (2007) and Li et al. (2019) proposed the interactive visualization and analysis tool denoted as Visualization and Analysis Platform for Ocean, Atmosphere, and Solar Researchers (VAPOR) which can explore terabyte-sized domain science data sets using commodity personal computers (PCs). It also makes use of external tools (e.g. Interactive Data Language [IDL] and Python) for mathematical computations and scripting capabilities. This allows for progressive data access combined with the rapid extraction of region of interests. The Met.3D (Rautenhaus et al., 2015) visualizes cloud cover as an isosurface from ensemble prediction data sets in order to estimate the spatial occurrence probability of warm conveyor belts.

In addition to the aforementioned freely available software, several commercial packages are commonly used, such as Mathwork’s MATLAB, Harris’ IDL and ESRI’s ArcGIS. These packages provide a rich set of features and are regarded as fourth-generation programming languages. However, such commercial toolkits generally target the general computing market and, therefore, may not meet some of the specific demands of the atmospheric and geoscience community.

Lastly, the third category includes several consumer-oriented applications, such as Google Earth, the NOAA SOS Explorer, the National Aeronautics and Space Administration’s (NASA) World Wind, and MeteoEarth. These applications are designed for the general public and provide real-time weather forecasting, as well as the rendering of atmospheric variables, including clouds, precipitation areas and wind trajectories.

For a comprehensive survey of visualization techniques and tools employed for meteorological data analysis, see Rautenhaus et al. (2018).

3 | OVERVIEW OF THE FOUNDATION FRAMEWORK

3.1 | System architecture

A modular and stratified approach is employed during the system design phase to achieve a high level of code reusability and portability. The outcome is a library framework that consists of several loosely coupled foundation modules upon which applications can be built.

A diagram representation of the system architecture is provided in Figure 1. A module is represented as a box with a solid border. Modules are enclosed within a layer drawn as a box with a dashed border. As demonstrated, the system is stratified into three layers. At the lowest level is the “infrastructural libraries” layer that consists of five modules. As the name suggests, the functionalities of the layer are application-agnostic and employed by modules or applications at higher levels. The “utility” module provides miscellaneous functionalities such as file system manipulation, networking data transfer, language localization and so on. The “map and projection” module implements GIS features including multi-resolution map tiles, cartographic projections and map clippings. The “math” module provides an array of mathematic algorithms. Data input and output are handled by the “I/O” module, which supports a wide variety of popular data formats. The “render” module is responsible for graphic object visualizations.

The “application libraries” layer lies on top of the “infrastructural libraries” layer, the features of which are implemented to meet the real-world demands of certain applications. The “meteorology” module manages to implement all the necessary meteorological algorithms such as objective analysis (Barnes, 1964; Cressman, 1959), isoline analysis (Hansen and Johnson, 2004), potential vorticity inversion calculation (Davis, 1992; McIntyre and Norton, 2000), tropical cyclone intensity estimation (Dvorak, 1975) and so on. Graphic representations of data sets are accomplished via the “graphics” module, which provides comprehensive visualization representations (Nocke et al., 2008), layer manipulations and page layout management. The “performance” module
is designed to improve the security, stability and performance of the applications. The last module of the layer, “GUI/Draw”, implements the logic behind the user interface. For example, it provides a complete mechanism to handle device input events, such as panning and zooming. It also recognizes drawing gestures such that users can draw predefined symbols in an interactive manner. Similar to the “infrastructure libraries” layer, the functionalities provided by the modules of the “application libraries” layer are generic.

The top layer, “applications”, develops real-world applications based on the functionalities provided by the underlying modules. They can be developed for desktops, workstations, mobile devices and HTML 5 websites and can include even non-GUI services running on servers.

In order to decrease coupling dependencies among modules and maintain the entire code base as compact and efficient as possible, the following design decisions are enforced. First, modules inside the same layer are independent of each other and function calling is prohibited between code segments of different modules. Second, applications or modules of a higher level layer can call the application programming interface (APIs) provided by modules of lower level layers, but not vice versa. This ensures a one-way dependency route.

3.2 | Technical advantages

In order to demonstrate the unique characteristics that MeteoExplorer brings to users, several key technologies are discussed in the following sections.

3.2.1 | Integrated 2D and 3D rendering engine

Numerical simulation data sets are 3D and generally change over time. As a result, they are inherently four-dimensional (4D). However, most of the GIS and meteorology tools use only 2D visualization methods and, therefore, provide 2D plots, including weather maps, vertical cross-sections and various types of diagrams. The missing dimensions are compensated by changing rendering frames along the vertical and time dimensions. Meteorologists use these products to explore the data and organize their forecast thoughts.

Contrary to the widespread use of 2D methods, the application of 3D techniques is still limited. Forecasters are trained and familiar with 2D plots and, thus, they tend to be reluctant to accept 3D graphics (Szoke et al., 2003). In addition, the disadvantages of 3D visualization prevent its adoption. First, the scene tends to be illegible when a large number of objects that occlude each other are rendered and users may lose spatial perception in a 3D context. Second, the high requirement for dedicated graphics hardware results in performance issues. Third, 3D methods may not be more intuitive for meteorologists to understand their data compared with traditional 2D methods, particularly if inappropriate presentation methods are employed.

Nonetheless, there has been significant progress in consumer graphics processor capabilities and several interesting GPU-based systems (Clyne et al., 2007; Rautenhaus et al., 2015; Li et al., 2019) proposed the application of 3D methods to the visualization of scientific data. Such software systems offer innovative visualizing features that help users analyse their data in a way that is impossible with conventional 2D methods. In particular, these two systems are 3D-only, in that both the rendering of graphics objects and the manipulation of viewpoints are performed in a 3D context.

As forecasters are already familiar with the 2D products essential in operational settings, an integrated approach that incorporates both a 2D and a 3D context into one framework is employed in the proposed system. The outcome is a rendering engine that provides the following innovative features.

First, users can switch between a 2D and a 3D context during the application runtime. The 3D context consists of two views: planar and globe. Rendition results are consistent during the context switch. This allows users to work comfortably in the 2D context and try 3D methods when required. Second, several 3D visualization methods, including direct volume rendering, isosurface and polygon tessellation, are implemented. Third, the geographical location and elevation corresponding to the mouse pointer position are provided in order to give users a spatial perception. Fourth, alpha composition is used to draw 3D objects semi-transparently to alleviate the occlusion problem. Fifth, the engine makes use of low-level graphics toolkits, such as Direct3D and OpenGL, that provide hardware acceleration capabilities to increase the interactive performance.

The last three features manage to overcome the previously detailed shortcomings of current 3D methods.

Figure 2 presents the visualization results of the ECMWF forecast data valid at 1200 UTC, September 13, 2016, when Typhoon Maranti was at its peak intensity. The 3D planar view is used here, and three variables are depicted. The world map is shown horizontally to aid users with the spatial perception of their data. Specific cloud liquid water content is visualized as filled contours in grey scale, while the 700 hPa wind field is denoted as brown barbs. Thus, the location of Typhoon Maranti can be easily identified to the southeast of Taiwan.
3.2.2 Comprehensive support of data formats

MeteoExplorer supports a wide range of data formats commonly used in the atmospheric and geoscience community, including Network Common Data Form (NetCDF), World Meteorological Organization (WMO) GRIB editions 1 and 2, GrADS (both station and gridded), CMA MICAPS, ESRI shapefile, Google Keyhold Markup Language (KML), High performance Data Format (HDF4, and HDF5).

The variety of data formats poses a challenge for developers. Furthermore, data formats usually consist of several sub-formats during their evolution. For example, NetCDF has three sub-formats: classic, 64-bit offset and NetCDF-4/HDF5. In order to overcome this challenge, the concept of a universal data model is proposed for an abstract description of the gridded data. The model consists of a data structure that represents all variables in a data set, as well as a set of helper functions to manipulate the data. The model can also calculate derived variables based on available values in the data set. For example, isentropic potential vorticity can be derived if both the temperature and wind of the isobaric layers are available.

Thus, the MeteoExplorer desktop application provides an intuitive UI (Figure 3) for users to create layers from a gridded data file. All variables are listed in the tree view control on the left and organized based on their dimensionalities. To create a layer, users have to select a variable, time and level, and then click the “Create Layer” button.

3.2.3 Export of vector graphics and page layout customization

Users expect a decent rendering performance as well as high-quality exported imagery, including vector graphics that may be used as figures in publications or presentations. Thus, MeteoExplorer offers the exporting of vector graphics and page layout customization.
More specifically, MeteoExplorer can export the scene as a vector graphics file. To achieve this, an abstract model denoted as the model of graphics rendering abstract layer (MGRAL) is proposed (Yu, 2019). Here, all the required functionalities are declared as interfaces (C++ pure virtual functions in the abstract base class), and are defined in concrete subclasses and called by the client. The model consists of four subclasses implemented using three graphics toolkits: Direct3D (Luna, 2012), the OpenGL fixed function pipeline and OpenGL programmable pipeline (Shreiner et al., 2013), and Windows Graphics Device Interface (GDI; Petzold, 1998). At runtime, only one instance of the subclass exists, and the choice of toolkit depends on the user’s preference or system configuration. With this design, the proposed engine can change the internal graphics toolkit to satisfy a particular requirement of the user. For example, the user may start the application with the Direct3D powered engine. When the user wants to export the rendition content as a vector graphics file, the toolkit is switched automatically to GDI during the image export operation and back to Direct3D afterwards. Two techniques are developed to support this feature. First, toolkits can be switched from one to another on the fly. Second, rendition results are preserved during the switch as the display properties of the graphics objects are almost identical for all the toolkits.

Next, in order to produce graphics suitable for publication, the page layout customization feature can be implemented on all visual elements, such as titles, coordinate axes, legend and insets. MeteoExplorer provides two predefined layout views: browse and image export, and offers the “page layout settings” dialogue to allow for a user to configure the layout. In browse view, the base map takes the entire window estate, while elements including titles, legend and inset are overlaid on top of the base map. In image export view, the rendering surface of the layers (or canvas) is restricted within a rectangular area and other elements have their own estate. The rendering result of the station plot, together with filled contours via the objective analysis of the precipitation values in the image export view, is illustrated in Figure 4.
The observation data set used is the 24 hr-accumulated precipitation valid at 0800 GMT + 8, March 31, 2017. The legend is positioned to the right of canvas. The South China Sea is drawn in the bottom-right of the canvas as the inset.

The canvas mask is also supported, such that users can have their layers drawn inside a specified region, typically defined by a country or a state border. Taking Figure 4 as an example, the station plot and contours outside the Chinese administration border are masked out. From a technical point of view, the canvas mask is implemented using the stencil buffer from Direct3D and OpenGL, and the GDI clip region.

3.2.4 | Cross-platform support

MeteoExplorer is designed to support as many mainstream platforms as possible. Currently, MeteoExplorer supports three primary desktop operating systems, including Microsoft Windows, Apple MacOS X and GNU Linux. Furthermore, efforts were carried out to port the code to mobile operating systems including iOS, Android and Windows Phone. A JavaScript port is under way for HTML5 web applications.

To support multiple computing platforms, several decisions related to the corresponding technology have to be made. First, the programming language must be considered. In order to maximize the runtime performance, MeteoExplorer is implemented as a native application using C++ instead of as an interpreted application coded in JAVA or C#. Next, code reusability is key for the ability of a software system to support multiple platforms. To achieve this goal, the MeteoExplorer framework is decomposed into several loosely coupled modules (Figure 1). As these modules are platform agnostic, MeteoExplorer can be applied to a new platform as long as the C++ compiler is available. In order to do this, developers have to create a user interface using the native toolkit of the platform and call the APIs provided by the modules to implement the requested features.

FIGURE 4  Rendering result of the station plot and filled contours via the objective analysis of the precipitation values in image export view
4 | METEOROLOGICAL APPLICATIONS

With the support of foundation libraries introduced in Section 3, the MeteoExplorer applications developed for the data analysis and visualization in weather forecast operations are now described.

4.1 | Desktop application

4.1.1 | Introduction

The MeteoExplorer desktop application was developed for professionals and researchers to analyse and visualize multi-source atmospheric and geoscience data in an interactive manner. The software is implemented in C++, while the Win32 API is employed to develop the GUI for Windows, and Qt is used for Linux and MacOS. Four graphics toolkits are incorporated into the rendering engine (see Section 3.2.3), with the OpenGL 3 programmable pipeline implementation performed by OpenGL shading language (GLSL) as the default driver. For systems with low-end GPU chips or a remote desktop connection scenario, the software falls back to the OpenGL fixed function pipeline implementation once the absence of the driver support is detected.

As shown in Figure 5, a canonical approach is employed in the user interface (UI) design. Common UI elements, including the menu, toolbar, a canvas view for graphics objects rendering, status bar and a layer management window, are available. Sub-windows for particular plots, such as the vertical cross-section, ensemble forecast and T-logP are uninitialized and hidden by default, but are generated and popped up on demand.

The Earth is drawn as a virtual globe as opposed to a map in 2D view (Figure 6) and 3D planar view (Figure 2). MeteoExplorer uses layers as the basic manipulation unit.

FIGURE 5  User interface of the MeteoExplorer desktop application. The plot shows a contour layer created from National Centers for Environmental Prediction (NCEP) global forecast system (GFS) mean sea level pressure (MSLP) data and a station layer from surface observations, both at 0000 UTC, April 2, 2017
to organize graphics objects. To manipulate layers, users may first select one or more layers from the tree view list on the left and then click a button on the right for the desired function. Figure 5 plots two layers: the contour layer is created from MSLP data from the National Centers for Environmental Prediction (NCEP) global forecast system (GFS) model, with a valid forecast time at 0000 UTC, April 2, 2017; and the station plot is created from surface observations at the same time.

Upon start-up, MeteoExplorer obtains the region information from the host operating system, thus indicating the user’s country at the centre of the application window with an appropriate zoom scale. Users may navigate to a particular region of interest by zooming and rotating (panning) in the 3D (2D) view with common gestures for a variety of input devices (mouse, keyboard, pen and fingers).

Loading a data file at runtime is supported. Users may open a file simply by dragging it into the application window, or use the relevant UI elements. The rendering content may be either copied to the system clipboard or exported to an image file in both raster and vector graphics. Popular image-encoding formats including BMP, JPEG, PNG and Windows WMF are supported.

4.1.2 Synoptic chart composition

MeteoExplorer is not merely a data viewer, but also provides interactive drawing capabilities. Forecasters may compose a synoptic chart by drawing or modifying isolines, fronts, trough and so on, and adding or removing weather symbols. This requires selecting a desired element from the toolbox window (Figure 6), and subsequently drawing the corresponding shapes according to the predefined gestures. Drawing operations may be cancelled or re-enabled using undo/redo commands from the UI. In order to provide a content recovery feature upon application crashing, the content of a drawn weather chart is automatically saved to a temporary file at a fixed interval or immediately as soon as the chart is modified.

A synoptic chart composed based on the surface station data at 0000 UTC April 1, 2017, is shown in Figure 6. Contours are drawn as black lines, while “G” (“D”) denotes a high (low) MSLP centre. Surface station data are also plotted, where MSLP (temperature) values are drawn in cyan (red) and wind barbs are in brown.
Visualization of ensemble data

The past decade has witnessed the ever-increasing applications of ensemble forecast methods. Compared with deterministic forecasting, ensemble prediction provides forecast uncertainties and plays an indispensable role in extreme weather and catastrophe forecasts. Owing to this, several ensemble visualization tools (such as those by Potter et al., 2009; Sanyal et al., 2010; and Rautenhaus et al., 2015) have been developed.

MeteoExplorer provides techniques covering all the steps of the ensemble data visualization pipeline introduced by Wang et al. (2019), and is detailed as follows:

- **Ensemble data.** MeteoExplorer supports four ensemble-prediction models, including the ECMWF Ensemble Prediction System (ENS), the NCEP Global Ensemble Forecast System (GEFS), the Canadian Meteorological Center (CMC) Global Ensemble Prediction System (GEPS) and the CMA GEPS.

- **Aggregation.** The system has managed to provide several aggregation methods, including statistical summaries and probability distributions of selected members to characterize the uncertainty information. To facilitate aggregation calculation, ensemble data exported by numerical models are reorganized by variable and level. That is, it is convenient to put data of all lead times and perturbations for one combination of variable and level (e.g. 500 mb geopotential height) into one file. A data pre-processing program that performs this conversion was developed and the total processing time for the four models was found to be within 1.5 hr.

- **Visualization.** Commonly used visualization techniques are supported. For statistics variables or probability distributions in space domain, pseudo-colouring, iso-contours (line and filled contours; Figure 8) and streamlines are provided. In order for users to investigate the region of interest, time-series plots revealing temporal trends, such as box–whisker plots (Figure 8), plume plots (Figure 7a) and wind roses plots...
(Figure 7b), are offered. In order to provide visualization for users in an interactive manner, parallel computing techniques are employed to reduce visualization time.

- Composition. Superimposition and juxtaposition are supported. By default, the system uses the superimposition method to overlay all layers on a map such that domain scientists can compare various aspects of different data sets without spatial registration. The spaghetti plot (Figure 7c) is such an example. In addition, the system supports the juxtaposition method with postage stamp maps (Figure 7d) in which summary views of all ensemble members are displayed.

4.2 | Web application

Recent decades have seen an obvious trend in the gradual transition from desktop to web applications, and almost all the forecast systems introduced in Section 2 have web variants. Web applications do not require installation and information can be conveniently accessed via the internet. Advanced web technologies, including HTML5, CSS, JavaScript and WebGL, make it possible to develop a web application that can handle intense graphics rendering and complicated user interactions.

The MeteoExplorer web application employs browser/server (B/S) architecture. The web client is implemented in JavaScript and WebGL is used for graphics rendering. In addition, the web servlet is implemented in Java and runs inside the Tomcat web container. The web client is responsible for user interactions and data visualization. For computation-intensive tasks, it sends requests to the server, where data I/O and calculations are performed, and subsequently uses the results returned from the server.

Figure 8 demonstrates the web client of the medium-range ensemble forecast system powered by the MeteoExplorer web libraries, together with several layers representing ensemble products calculated from ECMWF-ENS model prediction data valid at 1200 UTC, September 26, 2019. Line (filled) contours denote the mean (spread) of MSLP, while the box–whisker plots represent the precipitation of Beijing Haidian station. Results are calculated via a bilinear interpolation of the gridded data to the geographical location of (116.37 ° N, 40.08 ° E) with a forecast period of 10 days.

4.3 | Mobile application

Over recent years, mobile computing devices such as smart phones, tablet computers, and wearables have
formed a new computing platform that plays an increasing role in both personal and enterprise computing.

In the field of atmospheric and geosciences, software support is key for the effective performance of these all-new computing devices. Therefore, it is necessary to port existing desktop-oriented frameworks to mobile platforms. During the porting process, several issues must be considered. First, the source code must be optimized as mobile devices are generally underpowered in their processing capabilities and have limited memory capacity. For example, single precision is preferred over double precision for floating-point variables at the compile time to reduce memory consumption. Furthermore, the parameters of several algorithm implementations are adjusted so as to reduce computational complexity while at the same time maintaining accuracy. Second, as there is no C++ toolkit for these mobile platforms for the development of a UI, an interface is designed on top of the foundation modules to provide UI logic, the implementation of which is developed with Objective-C (iOS), Java (Android) and C# (Windows Phone). Third, several factors have to be accounted for in GPU programming to accommodate lower end devices that do not support complete OpenGL 3.3 and Direct3D 11 feature sets. For example, 3D texture is not supported in Direct3D 9 and OpenGL ES 2.0. Fourth, features designed especially for mobile devices, such as touch gestures and application life circle, are implemented.

To demonstrate the progress of the porting work, a reference application denoted as MeteoExplorer touch is developed. This application employs Direct3D 11 via a high-level shading language (HLSL) as the graphics driver of the rendering engine. A screen shot of the application running on the Windows 8 operating system is shown in Figure 9. The black (green) contours represent the MSLP (500 hPa geopotential height) and brown wind barbs denote the 700 hPa wind field. All layers are created from the NCEP FNL (Final) operational global analysis data with a valid forecast time at 0000 UTC, October 25, 2012, the time at which Hurricane Sandy reached its climax.

5 | COMPARISON WITH CURRENT VISUALIZATION PACKAGES

Compared with current visualization packages such as VAPOR and Met.3D that offer advanced 3D rendering capabilities, MeteoExplorer aims to achieve a balance between providing traditionally well-tested methods and...
exploring new innovative techniques. As a result, it employs a unified approach that incorporates both a 2D and a 3D rendering context into one framework and provides three typical view settings: 2D, 3D plane and 3D globe. The 2D view allows users to accomplish routine tasks using familiar and trusted methods. This is important in operational scenarios where analysis and visualization tasks are predefined. Extra efforts (e.g. the export of vector graphics and page layout management) have been made to improve rendition quality. That said, 2D visualization can still be obtained in VAPOR and Met.3D, yet they require users explicitly to manipulate the viewpoint or manually set camera parameters. And their 2D view is volatile because of 3D-oriented gestures. In MeteoExplorer, changing view settings is just one mouse click away and gestures are defined separately for 2D and 3D views. Interactively composing synoptic charts is another feature that is absent in VAPOR and Met.3D, yet welcomed in operational weather forecasting.

### 6 CONCLUSIONS AND FUTURE WORK

The MeteoExplorer software was developed for the analysis and visualization of atmospheric and geosciences data. In order to ensure the portability and reusability of the software, a modular and stratified design approach is employed. Compared with current visualization tools, several innovative features were developed to bring unique values to users. First, the rendering engine provides a unified framework that incorporates both two-dimensional (2D) and three-dimensional (3D) contexts that may be switched at application runtime. By extending 2D visualization into a 3D context, not only are proven 2D methods preserved but also new 3D methods are also introduced to be used when required. The engine also incorporates several graphics toolkits to provide both performance and rendition quality. In addition, MeteoExplorer supports a wide range of data formats commonly used in the meteorology and geographical information system (GIS) community. For gridded data sets generated by numerical prediction models in particular, a data representation model is proposed to handle...
different types of data formats in a unified and efficient manner. Third, the framework is cross-platform and can be ported to emerging computing platforms. Several software tools targeting desktop, mobile and web platforms were developed using the functionalities provided by the foundation libraries for real-world applications and, in particular, the data analysis and visualization in weather forecasts. Fourth, MeteoExplorer is not just a data viewer, but a content-creation tool that offers the feature of interactively composing synoptic charts, which is absent in most of the current visualization packages.

As its name suggests, MeteoExplorer is an exploratory project that aims to develop analysis and visualization methods useful for weather forecasting. The software is under active development and is being continuously improved to meet the demands of users. The author would like to explore the possibilities to deploy the framework to a broader range of application scenarios. In order to do this, the 3D rendering capabilities must first be improved in order to help professionals gain deeper insights into their data sets. In particular, the advanced technologies introduced in Met.3D and VAPOR such as flow-line rendering, 3D shadows will definitely benefit and inspire the future development of MeteoExplorer. Second, the implementation of a sophisticated lighting model will be developed in order to aid in identifying borders and relative positions of graphics objects. Third, the data-import capabilities of MeteoExplorer are not comprehensive as some data-encoding schemes, such as spherical grids, are not supported. Hence, the improvement of data support is an additional feature that will be developed in future. Fourth, calculating derived variables from numerical model data sets is currently statically built into the software and cannot be customized by users. To make this feature more flexible, script-based workflows will be implemented such that users may perform their own calculations.

ACKNOWLEDGMENTS
The work was supported by the seventh topic, medium-range weather forecast key technology integration and operational system development (2015BAC03B07), of the project key technology research on medium-range weather forecast (2015BAC03B00).

ORCID
Lianqing Yu https://orcid.org/0000-0003-1909-701X

REFERENCES
Barnes, S. (1964) A technique for maximizing details in numerical weather map analysis. Journal of Applied Meteorology, 3(4), 396–409.

Clyne, J., Mininni, P., Norton, A. and Rast, M. (2007) Interactive desktop analysis of high resolution simulations: application to turbulent plume dynamics and current sheet formation. New Journal of Physics, 9(8), 301–329.

Cressman, G.P. (1959) An operational objective analysis system. Monthly Weather Review, 87(10), 367–374.

Davis, C. (1992) Piecewise potential vorticity inversion. Journal of the Atmospheric Sciences, 49(16), 1397–1411.

Doty, B. E. & Kinter, J. L. I., 1995. Geophysical Data Analysis and Visualization using GrADS. In: E. P. Sussczewicz & J. H. Bredekamp, eds. Visualization Techniques in Space and Atmospheric Sciences. s.l.:s.n., pp. 209–219.

Dvorak, V. (1975) Tropical cyclone intensity analysis and forecasting from satellite imagery. Monthly Weather Review, 103, 420–430.

Gneiting, T. and Raftery, A. (2005) Weather forecasting with ensemble methods. Science, 310(5746), 248–249.

Goodchild, M.F. (2012) Next-generation digital earth. Proceedings of the National Academy of Sciences of the United States of America, 109(28), 11088–11094.

Hansen, C. & Johnson, C., 2004. Visualization Handbook. S.l.: Academic Press.

Hibbard, W.L., et al. (1994) Interactive visualization of earth and space science computations. Computer, 27(7), 65–72.

Hollt, T., Magdy, A., Zhan, P., Chen, G., Gopalakrishnan, G., Hotteit, I., Hansen, C.D. and Hadwiger, M. (2014) Ovis: a framework for visual analysis of ocean forecast ensembles. IEEE Transactions on Visualization and Computer Graphics, 20(8), 1114–1126.

Joe, P. et al., 2005. Severe weather forecasting tools in NinJo workstation. In: Proceedings of World Weather Research Program Symposium on Nowcasting and Very Short Range Forecasting. Toulouse: s.n., pp. 5–9.

Leutbecher, M. and Palmer, T. (2008) Ensemble forecasting. Journal of Computational Physics, 227(2008), 3515–3539.

Li, S., Jaroszynski, S., Pearse, S., Orf, L. and Clyne, J. (2019) VAPOR: a visualization package tailored to analyze simulation data in earth system science. Atmosphere, 10(9), 488–509.

Luna, F., 2012. Introduction to 3D Game Programming with DirectX 11. S.l.: mercury learning and information LLC.

McCaslin, P., McDonald, P. and Szoke, E. (2000) 3D visualization development at NOAA forecast systems laboratory. ACM SIGGRAPH Computer Graphics, 34(1), 41–44.

McIntyre, M. and Norton, W. (2000) Potential vorticity inversion on a hemisphere. Journal of the Atmospheric Sciences, 57(9), 1214–1235.

Murray, D., McWhirter, J. & Ho, Y., 2009. The IDV at 5: new features and future plans. In: Proceedings of 25th Conference on International Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology. S.l.:s.n., p. 7B.5.

Nocke, T., Sterzel, T., Böttinger, M. & Wrobel, M., 2008. Visualization of Climate and Climate Change Data: An Overview. In: Ehlers, ed. Digital Earth Summit on Geoinformatics 2008: Tools for Global Change Research. s.l.:s.n., pp. 226–232.

Petzold, C., 1998. Programming Windows. 5 ed. s.l.: Microsoft Press.

Potter, K. et al., 2009. Ensemble-Vis: a framework for the statistical visualization of ensemble data. International Conference on Data Mining Workshops, pp. 233–240.

Rautenhaus, M., Böttinger, M., Siemen, S., Hoffman, R., Kirby, R. M., Mirzargar, M., Rober, N. and Westermann, R. (2018)
Visualization in meteorology—A survey of techniques and tools for data analysis tasks. *IEEE Transactions on Visualization and Computer Graphics*, 24(12), 3268–3296.

Rautenhaus, M., Kern, M., Schafler, A. and Westermann, R. (2015) Three-dimensional visualization of ensemble weather forecasts—Part 1: the visualization tool met3D (version 1.0). *Geoscientific Model Development*, 8(7), 2329–2353.

Russell, I., et al. (2010) Metview 4—ECMWF’s latest generation meteorological workstation. *ECMWF Newsletter*, 126, 23–27.

Sanyal, J., Song, Z., Dyer, J., Mercer, A., Amburn, P. and Moorhead, R. J. (2010) Noodles: a tool for visualization of numerical weather model ensemble uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 16(6), 1421–1430.

Shreiner, D., Sellers, G., Kessennich, J. & Licea-Kane, B., 2013. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Versions 4.3*. 8 ed. s.l.: Addison-Wesley Professional.

Szoke, E., Grote, H., McCaslin, P. & McDonald, P., 2003. D3D update: is it being used? In Proceedings of 19th International Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology, p. P1.10.

Wang, J., Hazarika, S., Li, C. and Shen, H.-W. (2019) Visualization and visual analysis of ensemble data: a survey. *IEEE Transactions on Visualization and Computer Graphics*, 25(9), 2853–2872.

Wang, Y. (2014) MeteoInfo: GIS software for meteorological data visualization and analysis. *Meteorological Applications*, 21, 360–368.

Warner, T., 2011. *Numerical Weather and Climate Prediction*. First ed. s.l.: Cambridge University Press.

Yu, L. (2019) Research and implementation of a versatile rendering engine for meteorological data visualization. *Journal of Marine Meteorology*, 39(3), 114–123.

Yu, L. and Hu, Z. (2011) Subsystem for interactively drawing synoptic chart in MICAPS. *Journal of Applied Meteorological Science*, 22(3), 375–384.

**How to cite this article:** Yu L. MeteoExplorer: Visual data analysis software for atmospheric and geoscientists. *Meteorol Appl*. 2020;27:e1916. [https://doi.org/10.1002/met.1916](https://doi.org/10.1002/met.1916)