Web based Interactive Visualization of Weather Radar Data

Ankit Sharma   Armand Girier   Yeonsoo Yang   Nobuyasu Nakajima

Toshiba Corporate Software Engineering Center *

Abstract

We present a novel web browser based interactive visualization application for exploring 3D weather radar data sets delivered by next generation phased array radars. The purpose is to aid in rapid observation of localized weather phenomenon, short term weather forecasting and disaster prevention. Our application allows remote users to access multiple data sets via a web browser application, and the ability to view ray casted weather data volumes atop 3D maps from different viewpoints, animated in time. The interface allows updating intensity threshold and color scale range for comprehending 3D precipitation shape at different intensity levels, vertical cross-sectional views for viewing the internal structure and orthographic projection onto 3D map to clearly locate the affected areas. A demo collated using Osaka city torrential rain data set was presented to domain experts and received encouraging remarks on usability and performance.

Keywords: Ray casting, torrential rain, volume visualization, weather radar, WebGL.

Index Terms: I.3.6 [Computer Graphics]: Methodologies and Techniques—interaction techniques; I.4.10 [Image Processing and computer vision]: Image Representation—Volumetric

1 Introduction

In meteorology, study of localized extreme weather cases is of utmost importance for heavily populated cities. Researchers in this area are interested in learning the mechanism behind phenomenon such as the initiation, development and dissipation of convective precipitation that results in torrential rain and tornadoes. Next generation phased array radar systems [1] allow rapid 3D observations. Data from such systems is usually available as a 3D scalar field of echo intensity values. Conventional techniques for visualizing radar data focus on conical scan surfaces (plan position indicator, PPI) or 2D cross-sectional cuts of the data, which might be vertical (range height indicators, RHI) or parallel to the surface of the earth (constant altitude plan position indicators, CAPPI). 3D visualization techniques such as ray casting based volume rendering, on the other hand, accumulate information from the entire data set and project a 2D image of the 3D data. It enables us to visualize the data set as a whole, offering better comprehension of salient data features along with the ability to view the volume from different points in space. 3D visualization of radar data for study of localized weather phenomenon has been reported in [1] wherein the authors used commercial visualization software, best suited to people with a background of computer visualization techniques.

We introduce an easy to use web browser application and framework for exploration of meteorological datasets.

* e-mail: <first-name.last-name>@toshiba.co.jp

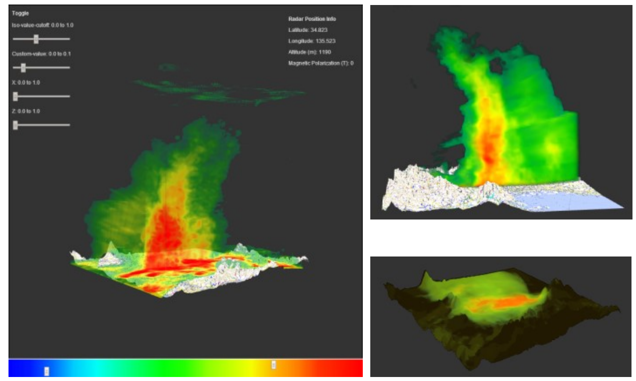

Figure 1 *left,* screenshot of the web application, radar information is shown in the upper right corner, controls in upper left and color map at bottom; *top right,* cross-sectional view of the data; and *bottom right,* projection using orthographic ray caster

1.1 Related Work

First attempts at web based volume rendering of weather data involved a thin-client framework, wherein, the rendering is done on the server [2]. This hindered client side interactivity rendering it unsuitable as a tool for studying spatially localized data sets.

Solutions based on client side rendering mostly depend on specialized browser plugins such as VRML plugins as in [3] with the notable exception of [4]. The latter addresses web based volume rendering using WebGL.

1.2 Our Approach

Our prototype application uses WebGL and provides the user an easy to use interface to inspect time varying weather radar data sets. The rendering is animated in time and the user is free to rotate the rendered 3D volume or move around to a different viewpoint. Different projection modes, maximum intensity projection, average intensity projection or the standard alpha composting based volume rendering projection can be chosen. To reveal regions with specific intensity levels, sliders for intensity thresholds and selecting cross-sections have been implemented. A digital elevation model for the underlying terrain is also rendered, height map for the given geographical location being fetched from The Geospatial Information Authority of Japan (GSI). Orthographic projection of data onto the map is rendered to clearly identify affected areas. Figure 1 shows screenshots of the client side interface. By integrating various interactivity options, we hope to provide valuable insights to the users.

2 System Design

Figure 2 shows the main components of the framework and the data flow. At the server side of the framework, normalized data from archives is interpolated to texture atlas images, which are then encoded as a video that is streamed to the client and used as render input. At the client side, GPU ray casting and UI handling are done using WebGL and JavaScript.

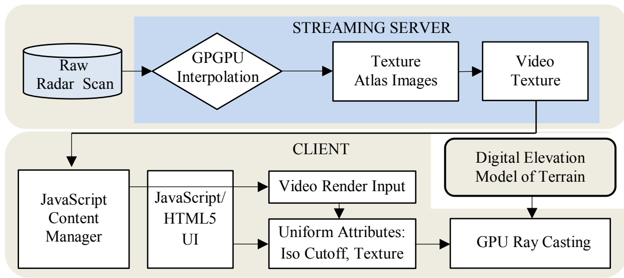

Figure 2: Components and data flow

2.1 Volume Texture Atlas Construction

Weather radars capture data in spherical coordinates. To compute texture atlas images, we fit a uniform 3D grid over the spherical data. Trilinear interpolation is used to calculate the intensities at the grid vertices. After this, the grid is sliced and arranged as a 2D texture atlas image.

2.2 Volume Rendering

Practical ray casting using WebGL is described in [4]. The paper computes ray coordinates using a two-pass approach wherein the front and back faces of the cube bounding the volume data are rendered into off-screen buffers. The interpolated position values for a pixel in the front and back faces gives the start and end positions for the ray through that pixel.

We depart from the two-pass approach in that we compute rays in camera space coordinates. Since only one of the faces needs to be rendered for computing ray directions, we shave off a rendering pass from the previous algorithm. The ray tracer loops till the ray exits the bounding cube or encounters an obstacle such as a mountain. The latter is found out by comparing the depth texture value of polygons rendered with the ray depth computed in camera space. The code was implemented as a new node in X3DOM [5] for rapid prototyping.

2.3 Interaction Design

Since users are mainly interested in tracking down movements of high intensity regions, maximum intensity projection is implemented. A 1D texture that maps echo intensities to color values is used as the transfer function. Average intensity projection for an X-ray view of the volume and an alpha composting based approach were also implemented. The latter provides better comprehension of the 3D structure. To isolate high intensity regions, a threshold value (iso-cutoff) is specified and the ray ignores regions with lower intensities during the ray casting loop, resulting in a pseudo-isosurface projection. The threshold value could be set in real-time using a slider. Also, by sliding the ray start point back and forward, cross sections of the data along a particular axis can be visualized. An orthographic ray caster is used to cast the data down onto the map. These controls are exposed to the user via a web interface.

3 Evaluation and Discussions

We ran our tool on Osaka city torrential rain data sets. The tool was run on both PC (Chrome, Intel HD 4000 GPU) and smartphone device (Firefox, Adreno 320 GPU) and gave acceptable frame rate per second of 26 and 4 respectively. Results of 3D precipitation observation are shown in Figure 3. We were mainly interested in finding out the usability of our tool. Preliminary evaluations by domain experts from Toshiba Meteorological Systems Group were favorable. Since these users regularly use various existing radar visualization products, their feedback on the tool was solicited. Interestingly, at first, they confirmed the validity of the projection by citing expected features of the torrential rain. They were also quick to point out artifacts introduced by radar beams and voids in regions obstructed by mountains. Users were particularly intrigued by the fact that they could visualize animated 3D weather data from different angles and change intensity iso-cutoff values in real-time, a radical departure from the 2D glyphs they were used to seeing, enabling easier navigation and better data perception. Though, questions about errors and uncertainties introduced due to sampling during ray casting were there, the overall response was encouraging.

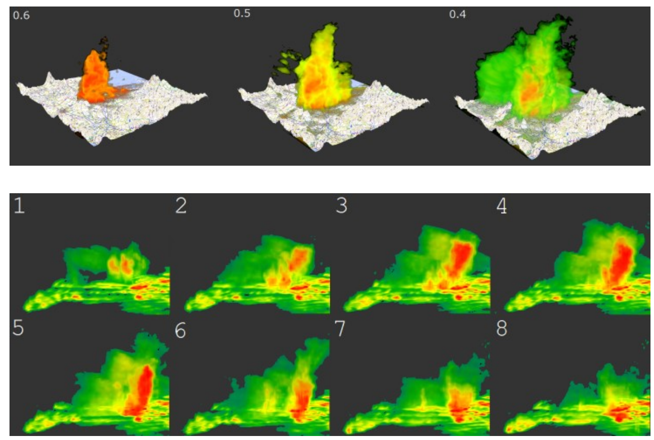

Figure 3: top, observation of cumulonimbus shape by varying iso-cutoff values (shown in upper left corner of the figures), bottom, animated ray casting, lifecycle of cumulonimbus (precipitation marked as red)

The web based nature of the framework allows for extension to real-time weather data analysis and deployment on multiple platforms like tablets and personal computers. Furthermore, we plan on implementing a wider range of visualization techniques for data sets like wind and temperature fields.

4 Acknowledgements

The authors thank the Toshiba Meteorological Systems group, Osaka University and National Institute of Information and Communications Technology of Japan (NICT) for providing the data sets used in this report.

References

[1] S. Satoh, T. Ushio, S. Shimamura, K. Maruo, F. Mizutani, M. Wada, H. Hanado, S. Kawamura, S. Uratsuka, and T. Iguchi. Three-dimensional Fine Structure of Localized Heavy Rainfalls Measured by Phased Array Weather Radar, 36th Conference on Radar Meteorology, September 2013.

[2] Sundaram, V., Zhao, L., Song, C., Benes, B., Veeramacheneni, R., Kristof, P. 2008. Real-time Data Delivery and Remote Visualization through Multi-layer Interfaces. In Grid Computing Environments Workshop, 2008. GCE'08, 1–10.

[3] Behr, J., Alexa, M. 2001. Volume visualization in vrml. In Proceedings of the sixth international conference on 3D Web technology, ACM New York, NY, USA, 23–27.

[4] Congote, J., Kabongo, L., Moreno, A., Seguro, A., Posada, J., Ruiz, O. 2011. Interactive visualization of volumetric data with WebGL in real-time. In Proceedings of the 16th International Conference on 3D Web Technology (Web3D '11). ACM, New York, USA, 137-146.

[5] X3DOM. http://www.x3dom.org/