WEB BASED VOLUME RENDERING OF AIR QUALITY 3D DATASETS AT THE CITY SCALE

Aitor Moreno[(1)], John Congote[(1, 2)], Oscar Ruiz [(2)], Juan Luis Pérez[(3)], Roberto San José [(3)]
[(1)] Vicomtech Research Center, Donostia - San Sebastian, Spain
[(2)] CAD/CAM/CAE Laboratory, EAFIT University, Medellin, Colombia
[(3)] Environmental Software and Modelling Group, Computer Science School, Technical University of Madrid (UPM), Spain
amoreno@vicomtech.org

Abstract. Air quality is listed as one of the world's worst pollution problem, mainly in urban areas where most of the population lived. Air quality Models (AQMs) can give an analysis of the causes which have led to these concentrations of pollutants. The data produced by urban AQM are 3D. Thus, the atmosphere around the studied urban environment is sampled in a volumetric grid, enabling the utilisation of well known volume rendering techniques.
This work presents the visualization of 3D datasets resulting of AQM models at city scale, loaded and rendered interactively in a Web application, where different UI controls are used to configure the parameters of the visualization and to navigate through the datasets.

1. INTRODUCTION

Nowadays the big cities face important problems related to the air quality of the surrounding atmosphere. The increase of the number of vehicles and the industry in general emit lot of pollutants to the air and they have to be monitored continuously to check that their levels are in the safe zone. Almost all big municipalities have installed a monitoring network to measure meteorological variables and pollutant levels and similarly to the weather predictions models, the Air Quality Models can be used to provide useful and important information to the citizens. AQMs are the only method which can quantify the relation between emissions and concentrations, including the consequences of future scenarios and the determination of the effectiveness of abatement strategies.
The air quality models produce a considerable amount of data. Raw data can be hard to conceptualize, particularly when the size of the data sets can be terabytes. In order for this data to provide useful insight into the workings of the atmosphere it must be visualized in a form that users can readily interpret.
Volume rendering is a facto standard model of scientific visualization. The methodology is used in several areas like medical imaging, geo-visualization, engineering, and almost any other field, which deals with n-Dimensional datasets. The method renders an image of a volume using the metaphor of passing light through semi-transparent objects. This grants the identification of internal structures, which are occluded by the external levels of the volume. The colour of the object is normally arbitrary and generated by a false colour model that is user assigned for each dataset and the region of interest of the volume.

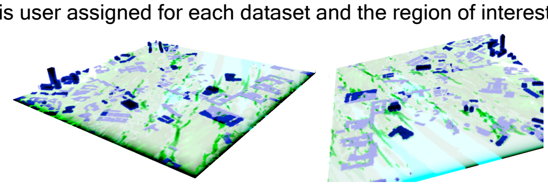

Figure 1. Screenshot of the volume rendering Web application for AQM datasets.

The computer graphics algorithms improve thanks to the increasing power of the GPU (Graphics Processing Units). This hardware represents a new model of parallel computation used normally in the game and entertainment applications. However this computational power now available in personal

computers and handheld devices is recently exploited by the new standard of WebGL. This standard allows the use of these devices with a full hardware acceleration mode, which was no possible in the past. Volume rendering is a costly algorithm, which uses this new available power allowing the use of this methodology en the new devices.
This paper presents the related work for the visualization the result of air quality models, and a brief description of the volume rendering techniques. Then, the followed methodology to load AQM datasets into a volume rendering Web application is explained, including the found differences between classical volumetric datasets (mainly medical dataset) and the AQM dataset. Finally, some discussion issues are addressed including the conclusions and the future work (see
Figure 1).

2. RELATED WORK

For this work, we have divide our research on two fields, one for the visualization techniques used nowadays to display information coming from Air Quality Models (AQM) and the other for the technical background of the Volume Rendering methods for scientific purposes.

2.1 Visualization of Air Quality information

2D maps and temporal series are used to the visualization of the air quality data produces by the AQMs. (see Figure 2).

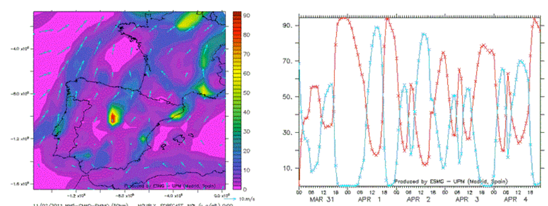

Figure 2. Air quality forecast by UPM AQMs models. Left: Map of NO2 and winds over Iberian Peninsula. Right: Time series of O3 (blue) and NO2 (red) over Madrid.

Air quality visualization has special requirements respect to other kind of information. Large masses of spatial data have to be managed by the visualization system. Visualization systems have been used in the air quality modelling community for some time to make sense of the output from computational mode. However, problems can arise when trying to interpret the results of the simulation. The visualization tools were designed as general visualization systems, to be used for everything from air quality modelling to the design of jet engines. Since they were designed as general purpose systems, only a small subset of the functionality was ever used for Air Quality Modelling.
New visualization tools have been develop to specific air quality modelling, with only the specific set of functionality needed for air quality analysis Visual tools like Vis5D provide a large number of complex methods to show AQM and meteorological information (see Figure 3, right).
A decisive criterion for the usability of visualization systems is often found in the implemented strategy for human computer interaction, above all concerning 3D-navigation [3]. During the mapping of 3D spaces to 2D screens, problems occur similar to those in traditional mapping. Additionally, interaction and navigation play an important role. Navigation is conventionally done by mouse movement or keyboard input.

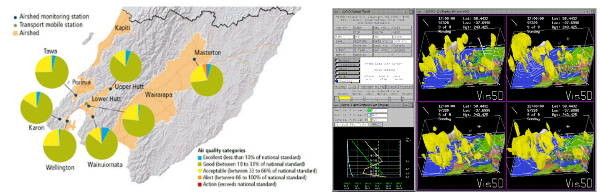

Figure 3. Left: Regional air quality in NewZeland data showing different pollutant information georeferenced[1]. Right: Vis5D sample screenshot showing different climate dataset[2].

In the last years, the Web has provided a great public space to publish weather and AQM data. In the same way that people check if it will rain tomorrow or not, there are several places around the world where people check the expected air quality. Although there is specific information in the weather websites for some air quality related variables, normally, they are at national or regional scale and in 2D. However, some national authorities are opening their databases, providing access to historic and realtime AQM datasets, enabling the innovation in several fields, including the visualization of such datasets.

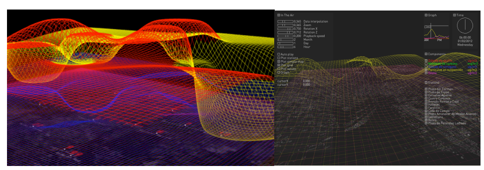

Figure 4. InTheAir[3] visualizes different agents measured in realtime in the Madrid's air quality monitoring network

In the Air is a Java based Web application, aimed to visualize the microscopic and invisible agents of Madrid´s air (gases, particles, pollen, diseases, etc). It uses realtime public data to create coloured grids per agent, combined with a Madrid simple map (orthophoto) for visual georeferenciation (see Figure 4).

2.2 Volume Rendering of scientific datasets

In 3D scalar field interactive visualization, two solutions prevail: Surface Rendering and Direct Volume Rendering. Surface Rendering has the advantage of being easy to compute due to its low geometric complexity. The main disadvantages are: *i)* A surface must be synthesized first, which is not a trivial task as it depends on the quality of the sample; *ii)* Since it must be precalculated, the result is static and cannot be easily adjusted in real time.

Recent advances in Direct Volume Rendering and graphic card capabilities allow the representation of volumes with good quality by projecting volumetric data into a 2D image, depending on the position of a virtual camera. The main advantage of this technique is the visualization of all inner characteristics at once.

Preprocessing of images does not intervene in the images since there is no part of the DVR of the computations even when the camera is displaced. In order to project the volumetric data, several methods

---

[1] Air Quality for Greater Wellington region (New Zealand). Website: http://www.gw.govt.nz/air-quality-2009-1/

[2] Vis5D: OpenGL-based volumetric visualization program for scientific multidimensional datasets. Website: http://vis5d.sourceforge.net/

[3] InTheAir. Madrid pollutants visualization Web Application. Website: http://intheair.es/

exist [Meisner2000]. Westover [Westover1992] discusses Volume Splatting and represents each scalar value by a simple geometrical shape that will face the camera, allowing fast rendering. The loss of quality is one of its major drawbacks. A technique called Shear Warping [Lacroute1994], consists of applying shear warp transformations to the volume slices to imitate the real orientation of the camera. Since the technique is based on simple transformations, the method is quite fast, but its main drawback is a low sampling power. With the constant improvement in graphic card capabilities, the Texture Mapping method has been popularized in video-games. It consists of re-slicing the volume depending on the orientation of the camera viewpoint, and representing all of the slices at once taking advantage of eventual occlusion optimizations [Hibbard1989], but the lack of specialized visualization methods in this algorithm has made it unusable for professional applications such as medical imaging.

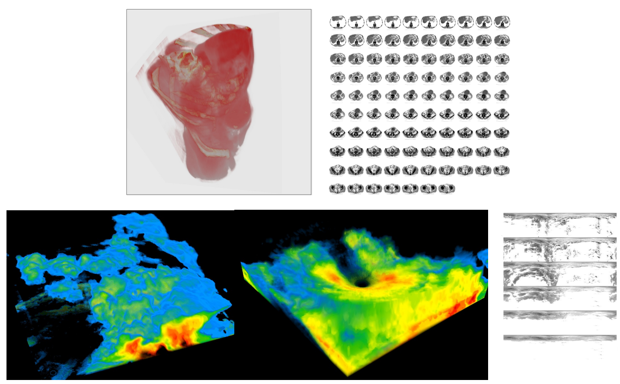

Figure 5. Volumetric visualization examples with the greyscale input dataset. Top: Medical dataset. Bottom: Weather radar dataset

Volume Ray Casting was initially presented by Kajiya [Kajiya1984] as an extension of the Ray Tracing algorithm for volumetric shapes, being later formalized by Levoy [Levoy1988]. Since then, Volume Ray Casting has become one of the most common methods for volume rendering. The set of rays from the camera reach the 3D scene and hit the objects, generating parametric (scalar) landmark values. By defining a blending function it is possible to give priorities to the different values encountered along the ray, allowing the visualization of different internal structures. Additional modifications to the algorithm, such as transfer functions, and Phong illumination [Phong1975] were developed in order to improve the perception and make the volume look realistic. Compared to the other techniques, this one is older and more accurate in sampling. However, the computational power required makes its usage initially difficult in real-time interactive representations, allowing other approximations to establish. Nowadays, the increasing computational power of graphic cards allows fast calculations [Kruger2003] which give new interest to Volume Ray Casting. Markus Hadwiger [Markus2009] presents a tutorial with all the basic explanation on Volume Ray Casting.

3. METHODOLOGY

In this work, our goal is to experiment with the AQM dataset and check if they can be loaded and rendered using volume rendering techniques. For such goals, we have selected some dataset provided by the Madrid Technical Univesity (UPM).

In the following sections, we will describe the process to convert the AQM dataset into the required images for the volume rendering engine, enhancing the significant differences of such datasets to the classical volumetric datasets.

In the last sections, the HTML interface for the Web application will be described, including the hardware and software platform used for the tests.

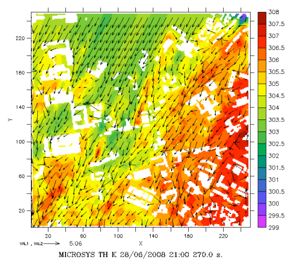

Figure 6: Potential temperature (K) on June, 28, 2010 at 21h00 GMT after 270 s simulation time using EULAG model.

3.1 Dataset description and loading stage

Numerical techniques for simulation of the atmospheric dynamics have been improved substantially and the progress of the computer power allows performing very high resolution simulations. Dataset has been obtained from large scale numerical experiments to simulate turbulent fluxes for urban areas. EULAG (UCAR, US) (Smolarkiewicz, P. K., and L. G. Margolin, 1997) micro scale model (CFD) which was modified to include an energy balance equation to obtain the urban atmosphere/biosphere energy exchange. In the Figure 6 we can see a 2D map of the Potential Temperature (K) simulated by the EULAG model.

The dataset represents a temperature field in a region of Madrid, which includes some buildings. The coverage of the data is around 1 km. × 1 km., with a spatial resolution of 4m, giving a 250 × 250 grid. Also, the dataset covers 100 m. in altitude with the same spatial resolution, resulting in a volumetric dataset of 250 × 250 × 25 scalar values.

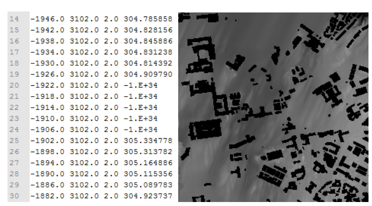

Figure 7. The first step involves reading and creating the individual images

The file of the dataset is around 46 MiB in size in ASCII, with the complete sampling of the scalar volumetric field, a 250 × 250 × 25 grid. Each line of the file contains the X, Y and Z coordinates and the scalar value, using a specific value (-1.E+34) for the "non existing values" (see Figure 7), normally associated to the places where the buildings are located. Once the data has been loaded, individual images per Z level can be created and saved for further use. In this case, we have mapped the range 298K - 313 K to the 256 possible values of a byte. The colour mapping is applied by matching greyscale levels to colours, using custom or predefined transfer functions (see Figure 8).

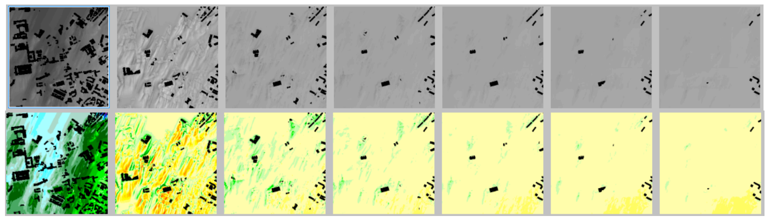

Figure 8. A subset of slices from an AQM dataset. Up: in greyscale. Down: applying a colouring mapping.

3.2 Volumetric ray casting for AQM

Once the dataset has been converted to a set of Z-ordered images (representing the scalar 3D field), some extra steps have to be done to be successfully loaded by the volumetric ray casting techniques.

In [Congote2011], the reader can find a fully detailed technical description of how the volume ray casting technique is designed and implemented. Essentially, the AQM dataset are very similar to the common medical datasets (see Figure 5), but there are some differences. First, the typical AQM dataset are not exactly a cubic grid, finding that the length in the Z axis (altitude) is less than in XY space. Furthermore, the typology of geodata in the urban areas are biased, i.e., the most interesting data structures are near the ground level, leading to a quite limited number of Z-slices with interesting information. This effect can be seen in the Figure 8, where the first four Z-slices contains almost all the information of the dataset (composed of 25 levels), a 16% of the whole volume.

It is worth to mention that the transfer functions are used to assign optical properties (colour and opacity) to the original volume data values in order to improve the visualization of the internal parts of the volume data. In volume ray casting, the transfer functions are used to obtain the optical properties of the volume data at each ray step. These values are blended using the composition function. In general, two transfer functions are defined, the colour transfer function, which obtains an RGB value and the opacity transfer function, which obtains a transparency value.

3.3 HTML Interface description

We implemented a simple HTML user interface using jQuery (see Figure 9) to interact with the volume rendering. It allows the modification of parameters such as the zoom and offset in the dataset, the Transfer Function colouring and some global values for the opacity and the light factor. The zoom and pan controls allow users to navigate through the AQM data. As previously stated, the useful information is found in the bottom of the volume, near the ground level, so these controls are useful to focus on specific regions of the volume. These navigation controls for zooming and panning have been implemented in the Web application as regular buttons, but more sophisticated interaction method could have implemented.

A very simple HTML-based editor to configure the Transfer Functions has been implemented, allowing customized inspection and segmentation of the AQM data by changing the values and colours with the provided sliders. Figure 10 shows different visualization of the same dataset, obtained by changing the transfer function (affecting colours and opacity of the scalar values). The editor is composed of 18 sliders, being the first one and the last one associated to the 0 and 255 values (minimum and maximum of the numeric representation of a byte). The 16 remaining sliders control the transparency and colour of each channel (16 values), except the second and the second to last sliders, which controls only 15 values (the 0 and 255 values are excluded). In the AQM dataset sample, the scalar values are defined in the 298K - 313 K range (15 K), so each channel range is nearly 1 K. If more precision is required, more sliders could

be added to the interface, but the UI will lose some usability. Other option is to map a narrower range of the input scalar values, like the 300K – 308K.
The interface also displays two extra sliders, supporting global modifications of the light factor and the transparency. The global transparency is a handy way to control the overall transparency in all the channels. The light factor enhances the contrast of the colours, making more visible and noticeable the inner structures of the volume.

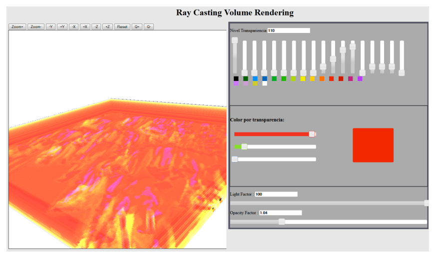

Figure 9. Screenshot of the Web Volume Rendering application

3.4 Testing platform

The tests have been performed in an Intel Quad Core Q9400 processor, 4GB of RAM and a GeForce GTX 275, Windows 7 PRO 64 bits with the latest stable graphics drivers. Amongst all of the Web browsers with full implementation of WebGL standard, we selected Firefox 10.0 for the tests, although other browsers are known to work with the implementation like Chrome 17 and Opera 12.00 alpha. Both Chrome and Firefox, in default configuration, use Google's Angle library to translate WebGL's native GLSL shaders to Microsoft's HLSL language and compile and run them through the DirectX subsystem. This procedure improves compatibility with lower-end hardware or older graphics drivers. Firefox also works if Angle is disabled; setting the native OpenGL subsystem as the rendering back-end. Finally, the Web application is served through a LightTPD Web server, installed and configured in the same computer for this experiment.

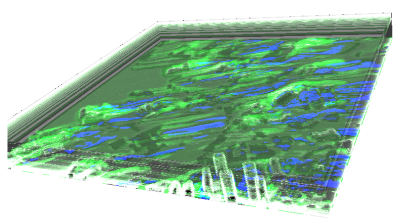

Figure 10. Different visualization results of the same volumetric AQM dataset.

4. CONCLUSIONS AND FUTURE WORK

This work has presented a Volume Rendering Web application to visualize 3D datasets coming from AQM. The HTML controls are used to configure the parameters of the visualization and to navigate through the datasets.
This preliminary approach has been focused in the differences between classical volumetric dataset and the AQM datasets, like the asymmetry in the Z direction and the bias of the data, commonly found at the bottom of the volume. Further research should be done to improve how the application handles this fact and provide specific functionality for these cases. Being limited to 1 byte for the greyscale (without going

to specialized hardware), a more intuitive window-level editor could be interesting. Right now, changing the range from 298K – 313K to 298K – 303K requires to export the images again, but losing the rest of the data. A seamless window-level automatic editor could be handy to interactively map the desired data range without exiting the application.

The georeferenced characteristic of the data enables to add traditional geographical data merged to the volumetric visualization. The inclusion of a real ortophoto in the bottom of the visualization helps to understand where the data has been taken. In the images shown in this work, the shape of the buildings are clearly distinguishable, as the voxels which represent them contains a null value, mapped in an independent channel in the Volume Rendering. Unfortunately, not all the datasets show this handy feature, and the voxels representing the buildings are interpolated and, therefore, invisible following this approach to map the null value to a specific value.

The solution would be to merge the volume rendering with real building information, providing better understanding of the location of the building, the streets, the road network, and therefore, more valuable information could be deduced from the original dataset.

In this work we have focused on the rendering of individual datasets of a specific variable (temperature), but the AQM provides temporal series of various pollutants. The next steps should be oriented to the extension of the volume rendering Web application to the 5D (timed series with different variables). For the multivariate scenario, it will be important to address the mechanisms to allow the selection and the merging criteria of the different pollutants, as it would depend on the users to give more priority to one pollutant than others.

5. ACKNOWLEDGEMENTS

This work was supported by COST Action TU0801 "Semantic Enrichment of 3D City Models for Sustainable Urban Development", the Basque Government's ETORTEK research programme, the CAD/CAM/CAE Laboratory at EAFIT University and the Colombian Council for Science and Technology –COLCIENCIAS–. The authors thankfully acknowledge the computer resources, technical expertise and assistance provided by the Centro de Supercomputación y Visualización de Madrid (CESVIMA) and the Spanish Supercomputing Network. The authors also acknowledge the ESPAÑA VIRTUAL CENIT project funded by the Spanish Ministry of Science and Technology.

6. REFERENCES

Congote, J., Kabongo, L., Moreno, A., Segura, Á., Posada, J. and Ruiz, Ó (2011) "Interactive visualization of volumetric data with WebGL in real-time". Web3D 2011, Paris (France), 20 June 2011, pp. 137-146

Hibbard, W. and Santek, D. (1989) "Interactivity is the key" Proceedings of the 1989 Chapel Hill workshop on Volume visualization VVS '89, pp. 39-43. doi: http://doi.acm.org/10.1145/329129.329356

Kajiya, J. T. and Von Herzen, B. P. (1984): "Ray tracing volume densities" SIGGRAPH Comput. Graph, Vol. 18, pp. 165-174. ISSN: 0097-8930

Kruger, J. and Westermann, R. (2003) "Acceleration Techniques for GPU-based Volume Rendering" VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03). ISBN: 0-7695-2030-8

Lacroute, P. and Levoy, M. (1994) "Fast volume rendering using a shear-warp factorization of the viewing transformation" Proceedings of the 21st annual conference on Computer graphics and interactive techniques (SIGGRAPH '94), pp. 451-458- ISBN: 0-89791-667-0

Levoy, M. (1988) "Display of Surfaces from Volume Data" IEEE Comput. Graph. Appl., Vol. 8, pp. 29-37.

Markus H., Patric L., Christof R. S., and Timo R. (2009) "Advanced illumination techniques for GPU-based volume raycasting" ACM SIGGRAPH 2009 Courses, pp. 1-166

Meisner, M., Huang, J., Bartz, D., Mueller, K. and Crawfis, R. (2000) "A practical evaluation of popular volume rendering algorithms" Proceedings of the 2000 IEEE symposium on Volume visualization, pp. 81-90

Smolarkiewicz, P. K., and L. G. Margolin, (1997) "On forward-in-time differencing for fluids: An Eulerian/semi Lagrangian nonhydrostatic model for stratified flows". *Atmos-Ocean Special*,35,127-157.

Phong, B. T. (1975) "Illumination for computer generated pictures" Commun. ACM, Vol. 18, pp. 311-317.

Westover, L. A. (1991) “Splatting: a parallel, feed-forward volume rendering algorithm” University of North Carolina at Chapel Hill.