The Journal's name will be set by the publisher
DOI: will be set by the publisher

Web Based Hybrid Volumetric Visualisation of Urban GIS data

Integration of 4D Temperature and Wind Fields with LoD-2 CityGML models

John Congote[1,3,a], Aitor Moreno[1,b], Luis Kabongo[1], Juan-Luis Pérez[2], Roberto San-José[2], and Oscar Ruiz[3]

[1] *Vicomtech-IK4 Research Centre, Spain*
[2] *Environmental Software and Modelling Group, Technical University of Madrid (UPM), Spain*
[3] *EAFIT University, Colombia*

Abstract. City models visualisation, buildings, structures and volumetric information, is an important task in Computer Graphics and Urban Planning. The different formats and data sources involved in the visualisation make the development of applications a big challenge. We present a homogeneous web visualisation framework using X3DOM and M3DX3DOM for the visualisation of these urban objects. We present an integration of different declarative data sources, enabling the utilization of advanced visualisation algorithms to render the models. Its tested with a city model composed by buildings from the Madrid University Campus. Volumetric datasets coming from Air Quality Models and 2D layers wind. Results of visualisation of all the urban models in real time on the Web, achieving interactive rates. An HTML5 web interface is presented to the user, enabling real time modifications of visualisation parameters. This work also addresses usability issues of the Framework when time based dataset are presented to the user.

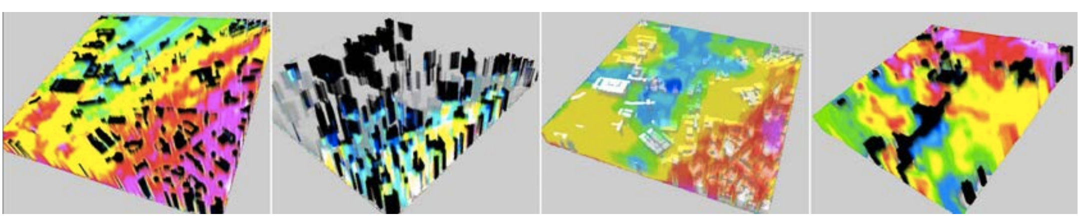

Figure 1. Different results of the proposed hybrid visualization web application, showing a temperature scalar field combined with georeferenced buildings.

1 Introduction

Scientific visualisation of city scale models, such as geometry, terrain, volumetric data and others are a big challenge in computer graphics. These models are represented in several markup languages

[a] e-mail: jcongote@vicomtech.org
[b] e-mail: amoreno@vicomtech.org

such as CityGML, X3D, COLLADA, DEM for geometry. NetCDF, Raw data for volumetric datasets or special compressed as JPEG, GeoTIFF formats for images. Some of this formats are rarely used and only some specialized software can work with these formats. The Web with the new standards as XML or HTML5 allows the inclusion of different models into a common framework, but the restrictions presented in the Web architecture difficult the implementation of the algorithms necessary for the correct visualisation of this models.

Volume rendering is a model of scientific visualisation. The methodology is used in several areas like medical imaging, geo-visualisation, 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.

Our contribution is the visualisation of air quality models, wind fields, buildings in a common framework using Declarative 3D technologies such as X3DOM and MEDX3DOM. Modification over the frameworks are proposed to work with these datasets. Also brief description of the volume rendering techniques. Then, the followed methodology to load AQM datasets into a volume rendering Web application and integrated with georeferenced geometry.

2 Related Work

2.1 Air Quality Information visualisation

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. 2D maps and temporal series are usually used for the visualization of the air quality data produced by the AQMs. These models generate spatial and temporal information, so most of the vector and scalar fields we wish to visualize vary with time. Animation of atmospheric data has been used to examine changes in the data over time. In case of wind data, they are visualized through vectors with a magnitude and a direction.

Air pollution and meteorology modeling are disciplines where 3D visualization is beginning to become very important subjects. The scientifics need tools to visualize and animate the data produced by their simulations. New visualization techniques allow to improve the level of understanding of the chemical and physical atmospheric process. Also, the employ of 3D displays and fast visualization tools help to navigate through the data.

2.2 Volume visualisation

In computer graphics, Ray-casting is a well known direct volume rendering technique that was designed by Kajiya and Herzen [1] as one of the initial developments in this area. Traditionally, three dimensional objects have been created by using surface representations, drawing geometric primitives that create polygonal meshes [2], hence provoking the loss of information from one dimension. Ray casting suitably fits GPUs' operating mode [3], because of the independence of each ray that is launched to the scene, making this algorithm highly parallelisable and allowing the exploitation of GPUs' parallel architecture. For GPU ray casting, the volume element is stored in the GPU memory as a 3D texture and a fragment shader program is used in order to implement the ray casting algorithm. visualisation of volumes on the web had been implemented and by Congote et al. [4].

2.3 Flow visualisation

A great amount of methodologies to visualize vector fields (flow fields) has been developed among the last decades. Geometric-based approaches draw icons on the screen whose characteristics represent the behavior of the flow (as velocity magnitude, vorticity, etc). Examples of these methodologies are arrow grids [5], streamlines [6] and streaklines [7]. However, as these are discrete approaches, the placement of each object is critical to detect the flow's anomalies (such as vortexes or eddies), and therefore, data preprocessing is needed to perform an illustrative flow visualisation. An up-to-date survey on geometric-based approaches is presented by [8]. However, in terms of calculating those trajectories for determined points in the field, the procedures usually compute for each point the integrals, and, as a result, the procedures are computationally expensive for highly dense data sets. Visualisation of flows in the web had been implemented by Aristizabal et al. [9]

2.4 Hybrid visualisation

There are several models for combining polygonal geometry and volume rendering, normally this methods use a blending between the image results, making these approaches unfeasible for city models, where large sections of the model could be occluded, this problem can not be solved with these techniques. Some methods identify the intersection between rays launched in the volume rendering process and geometry [10]. Ray casting is a flexible algorithm that allows the implementation of acceleration methods, such as Empty Space Skipping [11] or *Early Ray Termination*. Early ray termination is an optimization process that establishes certain limitations in the volume, samples encountered after them do not contribute to the value of the pixel [12]. Geometry is rendered in the first place to correctly look at the intersections of the geometry and the volume. Besides, parts that are occluded by the geometry should not contribute to the final image, not performing any ray casting at all. In order to achieve this feature, rays should terminate when they hit a polygonal object, accordingly modifying the ray length image if a polygonal object is closer to the view point than the initial ray length. Hybrid visualisation on web had been proposed by Ginsburg et al. [13]

2.5 Web 3D Rendering and MEDX3D

The use of the recently released WebGL standard [14] leads to new methods for web 3D visualisation, where most part of the computational processes are performed in vertex and fragment shaders that run on the GPU hardware. WebGL is a software library that enables HTML5-based browsers to identify clients' graphics hardware. HTML5, the latest Internet standard propose, provides native elements for audio and video. WebGL consists of a ow-level imperative graphic programming API based on OpenGLES 2.0 for Javascript that enables flexibility and exploits the characteristics of advanced graphics cards. Due to the constant improvement of the performance of Javascript interpreters, the management of scene elements behaves similarly to the ones obtained by using natively compiled languages. Moreover, some WebGL extensions have been implemented in order to achieve a friendly interaction, such as SpiderGL [15].Several standards and proprietary solutions are currently being developed in order to fulfill the necessity of moving 3D visualisation into the web [16], such as X3D, a standard derived from VRML that stores 3D information in a scenegraph format using XML (Extensible Markup Language). This model has been implemented in a declarative form, as an extension of HTML; X3DOM presents a framework for integrating X3D nodes into HTML5 DOM content [17] and other alternatives have also been developed, e.g. XML3D [18]. Finally, there is a standardization for X3D in the MedX3D volume rendering model [19, 20]. The MEDX3DOM proposal is presented by Congote [21] to visualize volumetric datasets in a declarative model.

3 Methodology

MEDX3DOM implements the standard *MEDX3D* in *X3DOM*. The implementation is based on the generation of the nodes for two components: *Texturing3D* and *VolumeRendering*. The actual status of the implementation declares all nodes for *X3DOM* architecture. Texturing3D nodeset define the tags to work with the information of 3D data, specific readers for file formats such as *DICOM*, *NRRD* and raw data are part of the standard. Other kind of data sources could be implemented like *WADO*, which is an specification for a WebService to obtain *DICOM* data or *NetCDF* to read scalar data. 3D texture data is normally stored in a **Texture3D** structured which is part of common 3D APIs like *OpenGL*. This structure is not available in WebGL and a mapping between a Texture3D and a Texture2D was define in order to overcome this limitation.

Volume data is storage in atlas format. An example for the volume dataset represented in this format is shown in figure 3. This model of representation also allows the storage of preprocessed data such as vector fields, where each color magnitude represent the direction of the gradient vector for each axis. Given the flexibility of the *X3DOM* architecture the transfer functions can be store as images or generated by the application.

The implementation of the MEDX3D standard into X3DOM allows the declaration of volume visualisation in X3D and visualized the render in browser without the use of plugins. The files follow the standard MEDX3D but the image texture is defined as a **Texture2D** structure. The file format is presented in the Figure 2, were an extraction of and HTML5 file is presented were a volume is rendered in a web page.

3.1 Flow Visualisation Style

A new style is proposed for the visualisation of vector field, the *StreamLineFlowStyle* allow the visualization of vector fields such as winds by transforming the vectors to a geometry representation know as stream lines. This lines are modeled as geometry internally for the MEDX3DOM extension. The visualisation of geometry and volumetric datasets in a common render pass is know as Hybrid visualisation. Other flow visualisation styles could be implemented where no geometry proxy is generated. Composition methods for visualisation of several volumes simultaneously should be implemented.

3.2 Hybrid Visualisation

The ray casting process has been improved to allow the integration of geometry. Each ray traverses the volume, where the volume dataset are represented. But the ray traversal will take into account any geometry inside the volume, so the ray casting termination condition has been modified. The termination condition is based on the alpha value for each sample of the transparency colour of the ray.

If the value to 1, it means that the ray has reached the final colour, and therefore, the remaining steps of the ray are not evaluated. Our approach to take into account the geometry of the buildings is to provoke an early termination of the ray casting, detecting when the ray collides with a geometric element. In that case, the alpha value is set to 1, and the finalisation of the ray casting is triggered.

4 Results

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

```
<html>
<head> ... </head>
<body>
<h1>Volume Rendering</h1>
<X3D xmlns='http://www.web3d.org/specifications/x3d-namespace'
  showStat='true' showLog='true' width='500px' height='500px'>
  <Scene>
    <Background skyColor='0 0 0'/>
    <Viewpoint description='Default' zNear='0.0001' zFar='100'/>
    <Transform>
     <Shape>
         <Inline url="buildings.x3d"> </Inline>
     </Shape>
     <VolumeData id='volume' dimensions='4.0 4.0 4.0'>
        <ImageTexture containerField='voxels' url='aqm/aqm_frame0.png'/>
<OpacityMapVolumeStyle>
<ImageTexture url='aqm/transferFunction.png'/>
</OpacityMapVolumeStyle>
      </VolumeData>
     <VolumeData id='volume' dimensions='4.0 4.0 4.0'>
        <ImageTexture containerField='voxels' url='aqm/aqm_wind_frame0.png'/>
<StreamLineFlowStyle >
</StreamLineFlowStyle>
      </VolumeData>
    </Transform>
  </Scene>
</X3D>
</body>
</html>
```

Figure 2. HTML/X3DOM proposed file fragment to visualize a the first frame of a AQM volumetric dataset using *MEDX3DOM*. The texture image is defined in atlas format (Figure 3).

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.

4.1 Dataset description and loading stage

Turbulent urban flow simulations require Computational Fluid Dynamics models. In this research the datasets have been obtained from large scale numerical experiments to simulate turbulent fluxes for

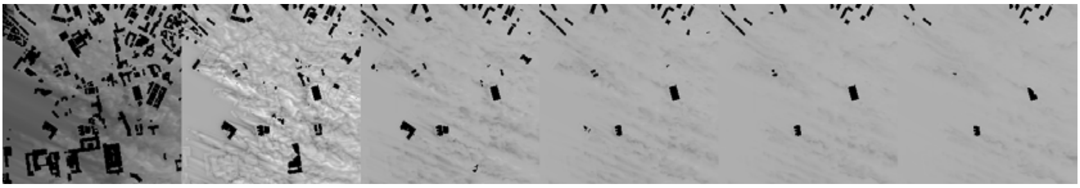

Figure 3. Texture atlas with the volume data for the AQM dataset

urban areas with the EULAG (UCAR, US) [22] CFD model, which was modified to include an energy balance equation to obtain the urban fluxes. In the Figure 3 we can see a sequence of 2D map of the Potential Temperature (K) simulated by the EULAG model.

The dataset represents the temperature and winds fields over a region of Madrid, which includes some buildings. The coverage of the data is around 1 km $\times$ 1 km., with a spatial resolution of 4m, giving a 250 $\times$ 250 regular grid. Also, the dataset covers 100 m. in altitude with the same spatial resolution, resulting in a volumetric dataset of 250 $\times$ 250 $\times$ 25 values for each variable. Temperature is a scalar value and wind is a vector field with two components horizontal (u) and vertical (v). The simulation time period is 5.5 minutes with outputs every 30 seconds. The computational time of this experiment is about 72 hours of CPU time with 64 processors.

Each line of the data file contains the X, Y and Z coordinates and the value, using a specific value (-1.E+34) for the "non existing values" (black pixels in Figure 3, 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 to 313 K to the 256 possible values of a byte.

4.2 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. Essentially, the AQM dataset are very similar to the common medical datasets (see Figure 3), 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 visualisation 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.

4.3 HTML Interface description

We implemented a HTML5 user interface using jQuery (see Figure 4) to interact with the AQM datasets, providing standard functionality to navigate the datasets at different level. First, we provide

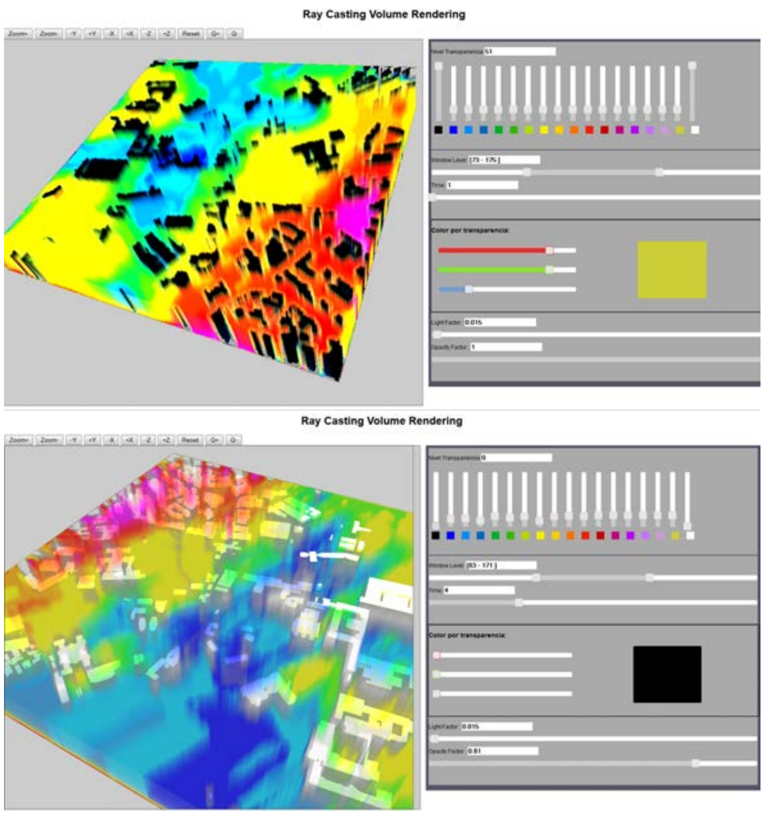

Figure 4. Different configuration of the HTML5 user interface and the achieved visual output.

mouse interaction in the HTML5 canvas through X3Dom custom navigation techniques. Also, some buttons in the upper part of the web page can be used to navigate inside the volume (zooming and panning). The time navigation is provided through a slider, used to select the frame of the volume datasets (see Figures 5 and 6). The animation can be set to manual or automatic, iterating continously between the frames of the datasets.

The Transfer Function configuration has been implemented combining a set of sliders, colours and the window level range configuration. There are 2 special sliders, fixed to the 0 and 255 values. The remaining 16 sliders maps the value range [1, 254] but it can be modified with the Window Level range editor, implemented as a range slider in jQuery. In the default settings, each sliders modifies the color and transparency of 16 values (256/16). Reducing the window level to the range [64, 128], the 16 regular sliders will adjust the transfer function to that specific range, so each slider will map just 4 values ((128 − 64)/16). In the AQM dataset sample (Figure 5), the scalar values are defined in the 298 K to 313 K range (15 K), so each channel range is nearly 1 K with a full range window level. Choosing the [64, 128] window level will provide a visualisation of the range [302 K, 306 K], with a resolution of 0.25 K for each channel or slider.

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

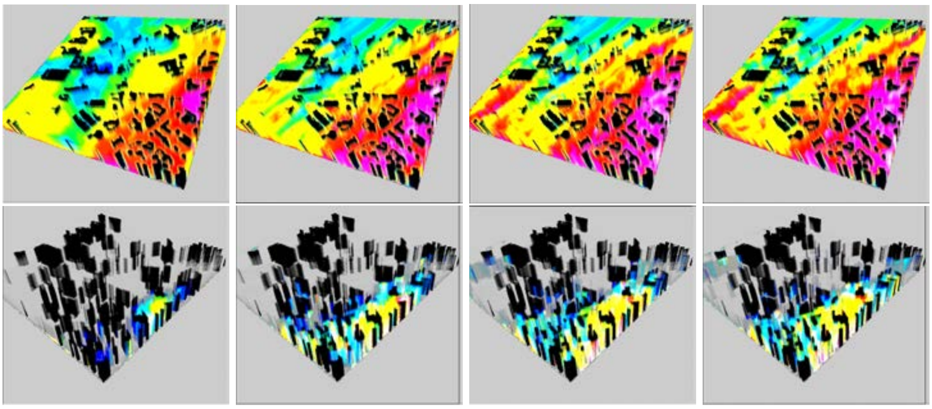

Figure 5. Two temporal series of the same datasets with different transfer function and window level configuration.

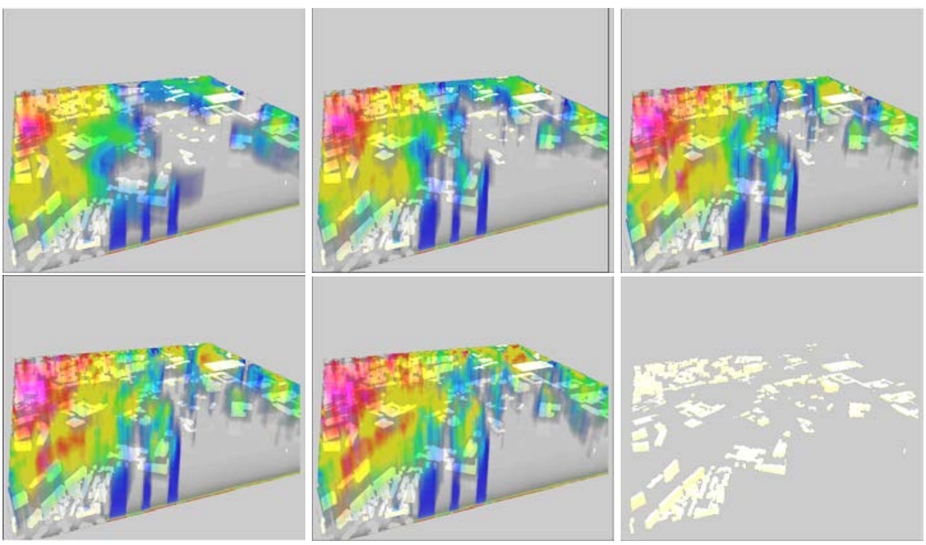

Figure 6. A set of frames of the hybrid visualisation of the volume and the buildings geometry (shown independently in the last subimage).

in all the channels. The light factor enhances the contrast of the colours, making more visible and noticeable the inner structures of the volume datasets.

4.4 Buildings Visualisation

The buildings model has been reconstructed from tagged SHP files, where each building corresponds with a 2D polygon, having their height specified as a feature in the DBF file. The reconstructed 3D model is equivalent to the LoD 2 proposed in the CityGML specification, although a X3D file was created to be later inlined in the X3D scene (see Figure 2).

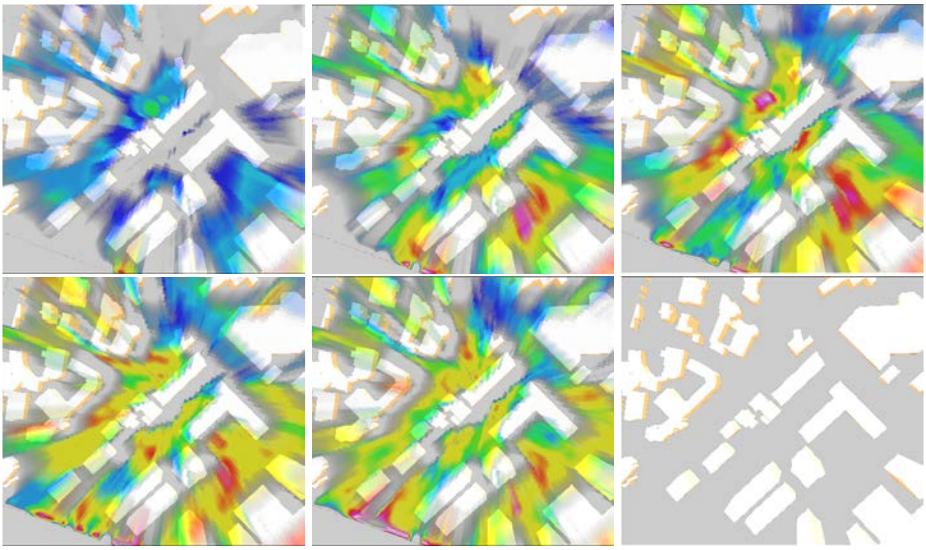

Figure 7. Top view of the volume dataset combined with the buildings geometry. The set of frames shows the temperature evolution of a small part of the urban area (shown independently in the last subimage).

Using the hybrid integration method for volume rendering allows us to visualise the buildings in the volume, as it can be seen in the Figures 6 and 7.

4.5 Wind Flow Visualisation

The wind information, provided as u and v components for each voxel in the dataset, were processed with the methods presented by Aristizabal et al. [9]. The result is a set of geometric streamlines, converted to X3D file format. The final step involved the integration of such X3D model as an inline node into the X3D scene (see Figure 2).

The integration of the streamlines with the volume dataset has been performed in the same way as the buildings, as they are not conceptually different, but some precautions had to be taken into account. A full reconstruction of the complete vector field is so dense that the visualisation turns to be not usable and therefore, not practical. A simplification was performed to produce a smaller set of streamlines. Even with such reduction, we have reduced it even more to achieve better graphical results for the Figure 8.

4.6 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 12.0 for the tests, although other browsers are known to work with the implementation like Chrome 19 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.

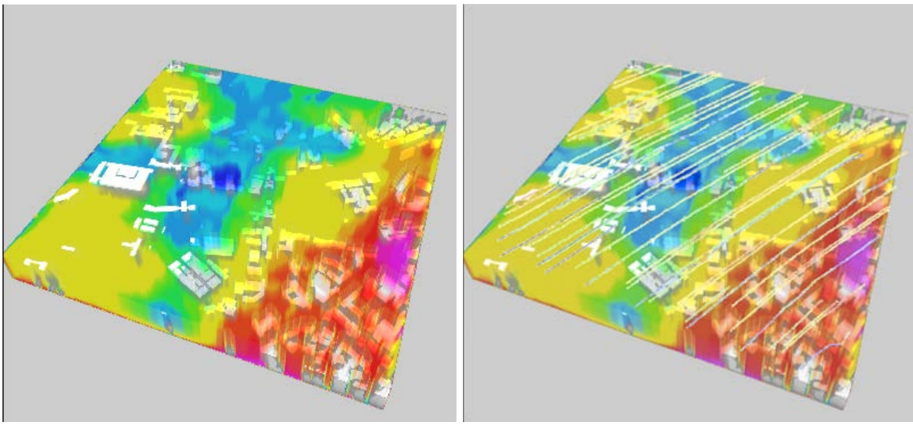

Figure 8. Hybrid visualisation of temperatures dataset (on the left, just the volume rendering and the buildings are shown) and wind information, reconstructed as colored geometric lines. For clarity purposes, only a small subset of wind flow lines have been added to the image (right).

Firefox also works if Angle is disabled; setting the native OpenGL subsystem as the rendering back-end. The Web application is served through a LightTPD Web server, installed and configured in the same computer for this experiment.

5 Conclusions and Future Work

This work has presented a Volume Rendering Web application to visualize 3D scalar and vector fields coming from AQM. An HTML5 interface has been provided to enable the users to configure the parameters of the visualisation and to navigate through the datasets (spatial and temporal navigation).

The integration of geometrical models (like buildings or wind streamlines) and the volume dataset helps to contextualize the data, helping to analyze visually the results of the AQM simulations.

In this paper a hybrid visualization has been presented, combining temperature (scalar field), wind flow (vector field) and the LoD 2 buildings existing in the zone. The results show that such integration of georeferenced datasets, provided through an interactive and high-performance Web visualization module, enhances the global understanding of the datasets.

This work has also presented how visual analysis of temporal datasets can be performed interactively. Using the same settings for the transfer function and window level, the fast loading times and the visualization performance of the models allows the creation of smooth animations, presented in this work as sequential frames.

This work has been focused on the rendering of temperature datasets, but the AQM can produce simulation for a wide variety of pollutants. For this 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.

For such operations, the key point is how users can deal with the heterogeneity of the multivariate scenarios. Transfer functions are a fundamental tool for the correct visualisation of such volumes. Some of the future work can be oriented to the generation of transfer functions by visual clues, know as Visual Volume Attention Maps (*VVAM*) [23]. Some preliminary results has been already researched and these methodologies are going to be tested in *MEDX3DOM*.

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 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.

References

[1] J.T. Kajiya, B.P. Von Herzen, SIGGRAPH Comput. Graph. **18**, 165 (1984)

[2] M. Levoy, IEEE Comput. Graph. Appl. **8**, 29 (1988)

[3] H. Scharsach, Proceedings of CESCG **5**, 67 (2005)

[4] J. Congote, A. Segura, L. Kabongo, A. Moreno, J. Posada, O. Ruiz, *Interactive visualization of volumetric data with WebGL in real-time*, in *Proceedings of the 16th International Conference on 3D Web Technology* (ACM, New York, NY, USA, 2011), Web3D '11, pp. 137–146, ISBN 978-1-4503-0774-1, `http://doi.acm.org/10.1145/2010425.2010449`

[5] R. Klassen, S. Harrington, *Shadowed hedgehogs: A technique for visualizing 2D slices of 3D vector fields*, in *Proceedings of the 2nd conference on Visualization'91* (IEEE Computer Society Press, 1991), pp. 148–153

[6] D. Kenwright, G. Mallinson, *A 3-D streamline tracking algorithm using dual stream functions*, in *Proceedings of the 3rd conference on Visualization'92* (IEEE Computer Society Press, 1992), pp. 62–68

[7] D. Lane, *UFAT: a particle tracer for time-dependent flow fields*, in *Proceedings of the conference on Visualization'94* (IEEE Computer Society Press, 1994), pp. 257–264

[8] T. McLoughlin, R. Laramee, R. Peikert, F. Post, M. Chen, *Over Two Decades of Integration-Based, Geometric Flow Visualization*, in *Computer Graphics Forum* (Wiley Online Library, 2010), Vol. 29-6, pp. 1807–1829

[9] M. Aristizabal, J. Congote, A. Segura, A. Moreno, H. Arriegui, O. Ruiz, *Hardware-accelerated Web Visualization of Vector Fields. Case Study in Oceanic Currents*, in *IVAPP-2012. International Conference on Computer Vision Theory and Applications*, edited by R.S.L. Paul Richard, Martin Kraus, J. Braz, INSTICC (SciTePress, Rome, Italy, 2012), pp. 759–763, ISBN 978-989-8565-02-0

[10] M.R.M. Silva, I.H. Manssour, C.M.D.S. Freitas, *Optimizing Combined Volume and Surface Data Ray Casting*, in *WSCG* (2000)

[11] J. Kruger, R. Westermann, *Acceleration Techniques for GPU-based Volume Rendering*, in *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)* (IEEE Computer Society, Washington, DC, USA, 2003), p. 38, ISBN 0-7695-2030-8

[12] M. Hadwiger, P. Ljung, C.R. Salama, T. Ropinski, *Advanced illumination techniques for GPU-based volume raycasting*, in *ACM SIGGRAPH 2009 Courses* (ACM, 2009), pp. 1–166

[13] D. Ginsburg, S. Gerhard, J.E. Congote, R. Pienaar, Frontiers in Neuroinformatics (2011)

[14] C. Marrin, *WebGL Specification*, Khronos WebGL Working Group (2011), `http://www.khronos.org/webgl/`

[15] M. Di Benedetto, F. Ponchio, F. Ganovelli, R. Scopigno, *SpiderGL: a JavaScript 3D graphics library for next-generation WWW*, in *Proceedings of the 15th International Conference on Web*

3D Technology (ACM, New York, NY, USA, 2010), Web3D '10, pp. 165–174, ISBN 978-1-4503-0209-8, http://doi.acm.org/10.1145/1836049.1836075

[16] J. Behr, M. Alexa, *Volume visualization in VRML*, in *Proceedings of the sixth international conference on 3D Web technology* (ACM New York, NY, USA, 2001), pp. 23–27

[17] J. Behr, P. Eschler, Y. Jung, M. Zöllner, *X3DOM: a DOM-based HTML5/X3D integration model*, in *Proceedings of the 14th International Conference on 3D Web Technology* (ACM, New York, NY, USA, 2009), Web3D '09, pp. 127–135, ISBN 978-1-60558-432-4, http://doi.acm.org/10.1145/1559764.1559784

[18] K. Sons, F. Klein, D. Rubinstein, S. Byelozyorov, P. Slusallek, *XML3D: interactive 3D graphics for the web*, in *Proceedings of the 15th International Conference on Web 3D Technology* (ACM, New York, NY, USA, 2010), Web3D '10, pp. 175–184, ISBN 978-1-4503-0209-8, http://doi.acm.org/10.1145/1836049.1836076

[19] N.W. John, M. Aratow, J. Couch, D. Evestedt, A.D. Hudson, N. Polys, R.F. Puk, A. Ray, K. Victor, Q. Wang, Studies In Health Technology And Informatics **132**, 189 (2008)

[20] N. Polys, A. Wood, P. Shinpaugh, Departmental Technical Report 1177, Virginia Tech, Advanced Research Computing (2011)

[21] J. Congote, *MEDX3DOM: MEDX3D for X3DOM*, in *To be pusblish in Proceedings of the 17th International Conference on 3D Web Technology* (ACM, New York, NY, USA, 2012), Web3D '12

[22] P.K. Smolarkiewicz, L.G. Margolin, *On forward-in-time differencing for fluids An Eulerian semi Lagrangian nonhydrostatic model for stratified flows.* (1997), Vol. 35, pp. 127–152

[23] A. Beristain, J. Congote, O. Ruiz, Studies in Health Technology and Informatics **173**, 53 (2012)