ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume VI-4/W2-2020, 2020
5th International Conference on Smart Data and Smart Cities, 30 September – 2 October 2020, Nice, France

# EXPLORING SCHEMES FOR VISUALIZING URBAN WIND FIELDS BASED ON CFD SIMULATIONS BY EMPLOYING OGC STANDARDS

Sven Schneider*, Thunyathep Santhanavanich, Athanasios Koukofikis, Volker Coors

University of Applied Sciences Stuttgart
Schellingstraße 24
70174 Stuttgart, Germany
(sven.schneider, thunyathep.santhanavanich, athanasios.koukofikis, volker.coors)@hft-stuttgart.de

**Commission IV**

**KEY WORDS:** Computational Fluid Dynamics (CFD) Simulation, Environmental Simulation, Visualization, Web Service, 3D City Models, CityGML, Open Geospatial Consortium

**ABSTRACT:**

In this paper various schemes for visualizing geo-spatial data such as Computational Fluid Dynamics (CFD) data are explored. The architecture of a new Smart Cities Platform is presented and examples of the visualization capabilities are given. Results show that scalar and vectorial measurands, such as wind pressure and wind directions, may be presented using the same schemes, however, interpretation of the visualization varies between measurands. A *hex-grid* representation of the highly dense point cloud data yields easier interpretation of the scene as do streamlines for visualizing a path of flow over and around buildings. Results of performance evaluations suggest that the same visualisation scheme (e.g. *hex-grid* ) but different data formats, yields faster loading times when using 3D Tiles rather than GeoJSON and an overall smoother interaction within the application.

## 1. INTRODUCTION

Just like companies and organizations, modern cities are facing the age of digital transformation towards smarter cities. The transformation will take place (and to some extent has already started) in many different areas such as digital city-services, resource management, mobility concepts, and public transport, as well as climate protection and environmental monitoring. While sophisticated state-of-the-art web-technologies, particular at the back-end of city platforms are inevitable to gather, maintain and distribute a plethora of data, an intuitive and interactive way of visualizing these data are just as important in order to condense large amounts of data to an understandable level of detail (Albakour et al., 2014; Portmann et al., 2017).

The use of Computational Fluid Dynamics (CFD), a method for simulating wind flow or propagation of fluids through and around objects, has been an established method since the pre-1980s in engineering fields such as aircraft design. However, during the 1980s, CFD was then also used to obtain velocity and pressure fields around buildings (Blocken, 2014). CFD in urban areas tries to answer questions like pedestrian comfort, pollution dispersion, wind load on buildings, and urban microclimate (Franke et al., 2011a). Studies performing CFD simulation on smaller or larger parts of real cities began around 2000 (Toparlar et al., 2017). For CFD simulations in cities as well as visualization of buildings in 3D, CityGML models serve as a basis prior to conversion into appropriate CAD file formats (e.g. STEP) and streamable tile formats (e.g. 3DTiles, Cozzi et al. 2019), respectively. The use of CityGML data for CFD simulations is problematic because of errors and the complexity of building models leading to high computational demands, thus steps for simplifications need to be done; however, this is beyond the scope of this paper. The interested reader is, however, referred to (Piepereit et al., 2018) and references therein.

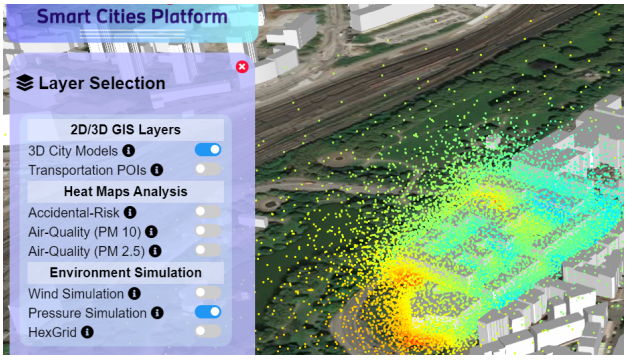


Figure 1. Screenshot of our Smart Cities Platform employing OGC standards at the back end.
(http://icity.hft-stuttgart.de/hftsmartplatform)

Visualization of complex environmental forces or phenomena such as wind (Figure 1), plays an important role in providing decision-makers a tool to understand and interpret large amounts of numerical data irrespective whether the data come from measurements or simulations. Having a set of diverse methods to show data in different contexts, scales, and details allows to maximize the amount of information that can be extracted from the data. The most apparent contexts are 2D vs. 3D, for example, heat maps vs. point clouds, respectively. However, with modern rendering capabilities and definition of OGC standards (Open Geospatial Consortium)[1] for defining rendering pipelines, streamable formats and web standards, the technical hurdles to implementing powerful data visualizations is *streamlined* as never before. This allows, to show highly heterogeneous data in a single view, render large data sets at high frame rates and even deliver relatively large and complex data sets at low bandwidth speeds due to streamable formats (e.g.

*Corresponding author

[1] https://www.ogc.org/

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-VI-4-W2-2020-157-2020 | © Authors 2020. CC BY 4.0 License. 

3D Tiles).

The aim of this study is to evaluate the following research questions: ❶ how to visualize wind field data with a focus on user interaction and ❷ how to stream such data in a web-based environment using OGC standards. Therefore, an interactive web-based platform to visualize and analyze a large amount of data is used as a case study to answer these questions. The front-end will explore different types of visualization schemes in Cesium digital globe, while the back-end is comprised of state-of-the-art OGC services to provide great flexibility to the users. One efficient way to allow viewing heterogeneous data to show diverse kinds of information is the interoperable 3D Portrayal Service (3DPS, Simon Thum et al. (2017)). The 3DPS standard describes how clients and services negotiate what is being delivered and in which manner. This enables interoperable 3D portrayal to the users so that they can view, analyze, and combine 3D geoinformation from diverse sources in a single view.

The paper will first provide a state of the art review on urban CFD and 3D visualization, describe data and methods as well as show and discuss the results of different visualization schemes, before finally concluding the paper.

## 2. STATE OF THE ART

### 2.1 Urban wind simulation

The research community so far utilized various methods to perform wind energy estimation in urban areas. Wang et al. (2018) compared and analyzed seven different urban tissues using CFD simulations in the area of Beijing (Figure 2). The numeric results of the simulations were correlated with urban morphology parameters such as floor area ratio, plot ratio, average building height, standard deviation of the building heights, mean building volume, relative rugosity, and porosity.

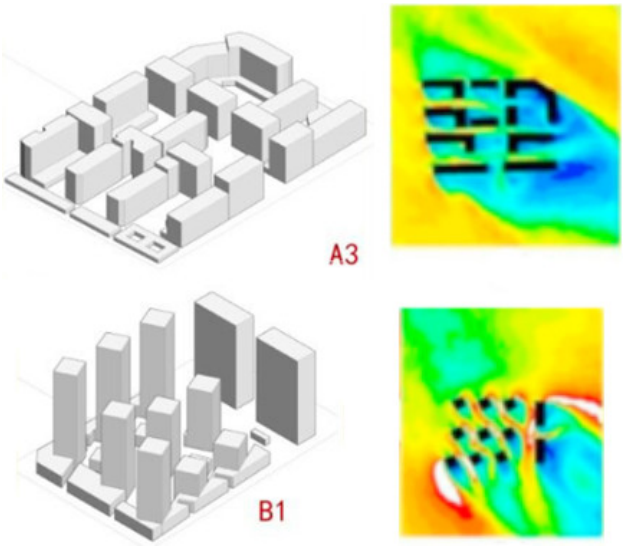


Figure 2. Urban tissues in Beijing and wind velocity contour lines (Wang et al., 2018)

The results showed that urban areas with high porosity, high average building height, and low floor area ratio could yield more energy from the wind.

In Helsinki's Kalasatama area, throughout the *digital twins* project (Suomisto et al., 2019), a high accuracy Digital Surface Model (DSM) model was generated for visualization purposes and a semantic city model based on CityGML. The CityGML model was used in a CFD simulation to investigate the wind impact on the microclimate, comfort, and safety of the streets and pedestrian areas (Figure 3).

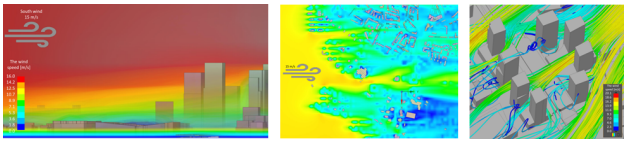

Figure 3. left: View of Kalasatama from the east. A south wind blows 15 meters per second, middle: A section cut of wind speeds about 31 meters above sea level, right: airflow coming from the right edge (south) of the image at a height of about 50 meters (Suomisto et al., 2019)

### 2.2 3D Streaming and visualization

Numerous studies in 3D web visualization are using emerging technologies by utilizing declarative and imperative graphics rendering pipelines. For example, a CityGML based view-dependent 3D city client-server architecture was implemented by Gaillard et al. (2015) as a source information model. *3DImpact* is an interactive web application developed by Patterson (2016) that accesses World Health Organization data, to visualize data using a virtual 3D globe based on Google Earth. In Döllner et al. (2012) a server-side rendering pipeline of 3D scenery, including client-side reconstruction based on geometry buffers, was presented. Another web visualization by Prieto (Izkara) was developed for 3D city models on mobile devices using X3DOM, driven by the use of CityGML as the reference data model. A novel method for rectangular selection of components in large 3D models on the web was introduced by Friston et al. (2019). A thorough study with a focus on the view conformance class of the 3DPS was done by Gutbell et al. (2016). In their work, they discussed the main aspects and features of the 3DPS standard using two prototype service implementations. A distributed data set was integrated into a web-based visualization showing high-resolution images and 3D city models based on CityGML. They show that geospatial data can be served by different independent data providers without data integration on the data model level.

Furthermore, the importance and usefulness of the 3DPS was shown in Koukofikis et al. (2018). 3D hierarchical formats can transmit arbitrary sized geospatial data; however, they are not interoperable with visualization technologies on the client. They implemented a series of prototypes focusing on rendering of hierarchical organized massive 3D data in various web client technologies employing 3DPS. Their focus was mainly to stream building geometry, and they found that a scene could be queried via the 3DPS by specifying a spatial region, rather than an URL, so that data was delivered either using I3S or 3D Tiles as a data delivery format.

A common rendering format for large 3D data is the glTF file format for 3D scenes and models using the JSON standard developed by the Khronos 3D Formats Working Group (2020). It can provide efficient transmissions and loading of 3D scenes and models in applications with a need for small 3D asset size and low run-time processing. However, building on top of glTF,

there is the 3D Tiles format, which is fully supported by the Cesium library and can stream 3D content similar to ESRI's i3d format. The 3D city model format of CityGML can be converted to streamable formats such as i3d and 3D tilesets and can be streamed and rendered as needed. It is an open specification for streaming 3D geo-spatial data sets optimized for streaming and rendering in interactive scenes as it is highly flexible, styleable, and able to handle heterogeneous data such as buildings, trees, point clouds, and vector data (Figure 5).

## 3. DATA AND METHODOLOGY

### 3.1 Wind Simulation using CFD

The data of the CFD simulation around a small building block in Stuttgart was taken from (Piepereit et al., 2018). The data basis was a ❶ CityGML model of a building block which was ❷ simplified using their algorithm. The simplified building model was then written as a STEP file so it could be read by the simulation software, ANSYS Fluent[2]. Based on the details retained in this simplified model, ❸ an irregular mesh was calculated. The wind in the simulation was triggered by a so called atmospheric boundary layer which can be characterized as a velocity profile with a parabolic shape (ABL, Tominaga et al. 2008). For each cell, the Navier-Stokes equations are solved by taking into account the neighboring cells (RANS; compare Franke et al. 2011b). In essence, the result of the CFD simulation ❹ is a table, that stores in each row, the coordinates of the cell center (in a local coordinate system), pressure and velocity values among other physical parameters.

**3.1.1 Study area** The study area is a district called "Stoeckach" which is situated in the heart of Stuttgart, Germany. It is an area that is subject to major restoration and reshaping. Under these terms fall refurbishing building facades to adhere to the latest local energy standards, improving the area by integrating parks, and demolition and construction of old and new buildings, respectively. A major undertaking is the integration and extension of the district heating network. To estimate the energy demand of buildings and districts, energy demand simulation software such as SimStadt [3] can be used. However, using CFD simulations wind pressure on facades and predictions about the comfort of new spaces (e.g. parks or "urban canyons") can be made. In this paper, we focus on the visualization of such CFD Simulation results within this district.

**3.1.2 CFD post-processing** The CFD results were ❺ georeferenced by selecting a distinct point from the visualized results and matching it with a building vertex at the same position. Then, data were centered and rotated and shifted to the coordinates of the EPSG 31467 coordinate system. The data for pressure and velocity are scaled to fit within an 8 bits unsigned integer. The data are ❻ mapped to a 'spectral' color table ranging from blue to red, representing low and high values, respectively. Finally, the modified data are written to a text file using geo-referenced coordinates and the pressure or velocity magnitude encoded as RGB values. This file is basically a point cloud and can be ❼ converted to Cesium 3D Tiles (OGC standard) using the CesiumPointCloudGenerator 2019 or using Cesium ION. After this step, the CFD simulation results are ready ❽ to be delivered and streamed for visualization in CesiumJS web globe.

[2]https://www.ansys.com/
[3]www.simstadt.eu

**3.1.3 Generation of a *hex-grid*** The large amount of points when viewing CFD simulation results as a point cloud (e.g. top of Figure 7), can be quite demanding on the hardware to render the data at high frames per seconds (fps). Also the amount of data to be downloaded by clients with low bandwidth internet connections may lessen and slow down the user experience. Albeit the point cloud data is relatively small when converted to 3D Tiles (around 3 MB), a simple layer containing a few polygons may be faster to render. Therefore, the point cloud was summarized using hexagonal bins with certain width (5 and 15 m). To perform the hex-binning and create a *hex-grid* the Whitebox GAT (Lindsay, 2016)[4] software packages was used. The tool is an open access software that allows to modify the code of every single algorithm within the software package. This option was used by extending the functionality of the binning algorithm to include an averaging of the pressure values within the CFD data set. Thus, the resulting *hex-grid* was a grid of hexagonal cells, where each cell represents the average pressure value at a certain height within a specified area. The area is defined by the width parameter $w$ by which the side length $s$ of the hexagon is defined. Thus, $s$ of the hexagon can be calculated as

$$s = w/[2 \cdot \cos(\pi/6)] \,, \tag{1}$$

and the area $A_{\bigcirc}$ of each hexagon is calculated with

$$A_{\bigcirc} = 3 \cdot s \cdot (w/2) \,. \tag{2}$$

The area containing the building block of interest (AOI) is around 47,300 $m^2$ large, however, the extend simulated with CFD is approximately 235,187$m^2$. Most of the cells of *hex-grid* were within the AOI, however, some were dispersed to a larger area, albeit there were gabs further away from the AOI. The width of the hexagon $w$ was chosen to be 5 m and 15 m equalling to $A_{\bigcirc}$ of 21.7 $m^2$ and 194.9 $m^2$, respectively. These values have lead to suitable generalization of values from the point cloud by dividing the data into around 2,890 and 800 hexagons, respectively. The pressure values within the area of the hexagon at a given height (approx. at building top level) were averaged and represented by a single value for the entire hexagon.

**3.1.4 Streamlines** are another well known and often used means of visualization of CFD simulation results. We explored this option as well and found, that they delivered good results and provided an easy way to understand the CFD results. In particular, the wind speed (norm of the vector) was visualized using the same method as described in Section 3.1.2, i.e. by transforming the data to a point cloud and then converting it to 3D Tiles. Here, the streamlines are not connected and only

[4]https://jblindsay.github.io/ghrg/index.html

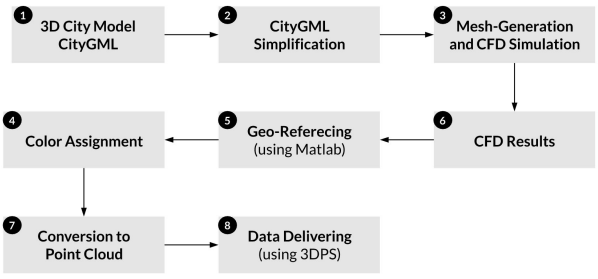


Figure 4. Implementation Step. Numbers correspond to the numbers in the text of Section 3.1.

visualized as individual 3D points. However, the conversion to 3D Tiles yielded a small data size (around 1.1 MB) that was loaded quickly from the server.

### 3.2 Managing 3D content using OGC 3D Portrayal Service

To manage the availability of the 3D content in a standard way, the OGC had introduced the OGC 3D Portrayal Service (3DPS) as a standard for 3D geospatial content delivery implementation specification. The 3D Portrayal Service (Hagedorn et al., 2017) is a OGC standard that abstracts the access of 3D geospatial datasets in various client platforms via the web for visualization purposes. Information can be accesed via three methods: *GetCapabilities*, *AbstractGetPortrayal* and *AbstractGetFeatureInfo*. A *GetCapabilities* request returns the available request methods, spatial extents, data layers, layer styles and streaming formats supported. To retrieve a scene the *AbstractGetPortrayal* operator is realized in two methods. The *GetScene* method is used for client side rendering and the *GetView* method for server side rendering. The *AbstractGetFeatureInfo* implementations are used for requesting metadata of scene features. The supported methods are *GetFeatureInfoByObjectId*, *GetFeatureInfoByPosition* and *GetFeatureInfoByRay*. The 3D Portrayal Service specification does not indicate a content delivery format. The 3D content can be retrieved as a `HTTP/GET Key Value Pair (KVP)-based` request. Table 1 shows examples of important KVP encodings used to request the specified 3D contents on our visualization platform.

| Key example | Value example | Definition |
|---|---|---|
| service | 3DPS | Service type identifier |
| request | GetScene | Operation name |
| version | 1.0 | standard version |
| bounding box | 0.0,0.0,0.0,10.0, 10.0,50.0 | Bounding box of the dataset |
| layers | building, pointcloud | Layer identifiers |
| lods | citygml:3 | LODs of requested layers |
| format | 3dtiles | Formats of requested layers |

Table 1. Example of 3DPS HTTP/GET KVP-based encoding

### 3.3 Data Visualization

The data visualization in 3D is done using the 3D web-based globe application. Currently, several providers of open-source frameworks are available such as NASA WebWorldWind[5] and Cesium[6]. These frameworks help the software developers to quickly develop interactive visualizations of the GIS layers in both 2D and 3D. In this paper, we showcase the development of the application based on the Cesium library as it supports most GIS standard protocols such as GeoJSON, OGC 3D Tiles, OGC Web Map Services (WMS), and OGC Web Feature Services (WFS).

Figure 5 shows an application demo of the Cesium application visualizing the New York City 3D city models in 3D Tiles format. All geographic layers, including wind simulations and 3D city models, are converted to 3D Tiles to be visualized with the Cesium application. The resulting 3D vector fields will be stored in a relational database, and will be accessed by the application layer.


Figure 5. 3D Visualization of New York City model on Cesium

The development of the depicted architecture (Figure 6) allows visualizing the outcomes of the wind simulations to a greater audience. 3DPS would play an essential role in the access to the web interface of 3D meshes and wind field information. On the client-side, the user will be able to navigate and visually analyze the simulation data utilizing a multi-layer scheme.

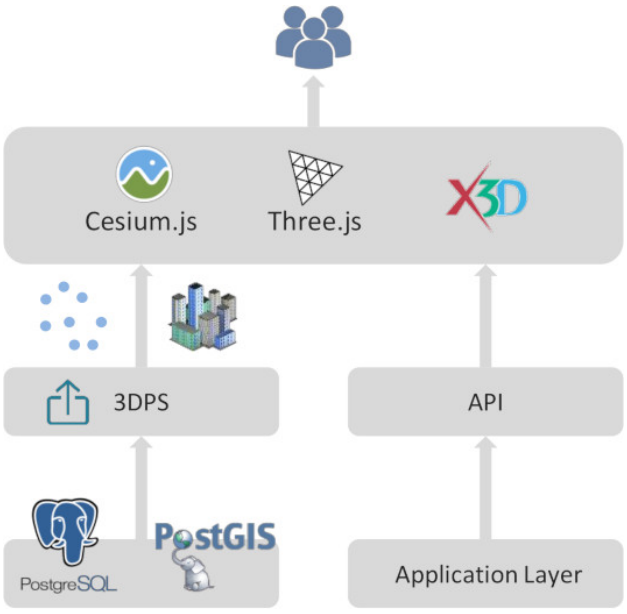


Figure 6. Overall architecture of the application. In this study, we use Cesium.js, but other rendering frameworks are also possible.

### 3.4 Measuring loading times

To provide quantitative performance measures of different visualization schemes and data formats, we assessed data loading

[5]https://worldwind.arc.nasa.gov/web/
[6]https://cesium.com/

times using the Google Chrome Developer Tools. Here the Cesium application was loaded without the CFD results in an *incognito tab*, so that no cache files were used. We zoomed closely into the area of interest and only then activated the switch to load the CFD results. This methodology ensured that the 3D Tiles were loaded completely upon activating the switch in our web application, as it usually would only download the high-level features and load or stream more detailed data as the user would zoom into an AOI.

The Chrome Developer Tools were then used to measure the times to load the different CFD results and formats. The tab and browser were closed completely, and the same procedure was repeated ten times to calculate an average loading time for a specific CFD scheme and format. The CFD data were requested from our 3DPS server with a high bandwidth internet connection; therefore, only relative evaluation against each of the formats and schemes is possible.

## 4. RESULTS AND DISCUSSION

The results of our architecture are visualized within our Smart Cities Platform (Figure 1), where different visualizations of the CFD results are presented (Figures 7 - 10) based on the layer selection of the user.

Results of the wind pressure (a scalar measurand) for the defined city block, can be visualized easily, and interpretation of the results is also straight forward because the color information provides direct intuition of the intensity of the pressure at each location. However, the visualization of the wind speed directions, which is a vectorial measurand, is not so straight forward to interpret. Of course, the norm of the wind speed vector is again a scalar and thus also easier to interpret; however, depending on the application, the wind direction is also of great importance. Our aim here was to show the difficulty of visualizing vector information and especially point out the difficulty of interpreting the results. To help to interpret the visualization of vectorial data, other means of visualization need to be applied, e.g., using arrows or line segments.

The use of the *hex-grid* method (Figure 9), allows for a faster and easier interpretation of results. It provides a quick summary of the highly-dense point cloud data (e.g. showing pressure). Also, it helps to simplify vectorial data by binning and averaging the components of the vector. For example, using a *hex-grid* , a single hexagon is colored according to the major wind direction (i.e., performing a majority vote of all the components) within the defined hexagon area. It can be envisioned that this method may be implemented into the Smart Cities Platform so that a user can choose the appropriate width of the hexagon to vary the level of detail (i.e., smaller hexagons for more detail).

When comparing *hex-grid* results using different formats, such as geoJSON and 3D Tiles, the overall loading times for each data set and scheme are also of great interest (Table 2). There are a couple of interesting differences between visualizations and loading times of geoJSON and 3D Tiles. ❶ The amount of data used for a geoJSON (albeit formatted) is significantly larger than the corresponding 3D tileset. For the *hex-grid* shown in Figure 8, a hexagon width of 5 m was chosen. This yields to around 2,890 hexagons (polygons as geoJSON *geometry type*), leading to a file size of just over 3 MB. On the other hand,

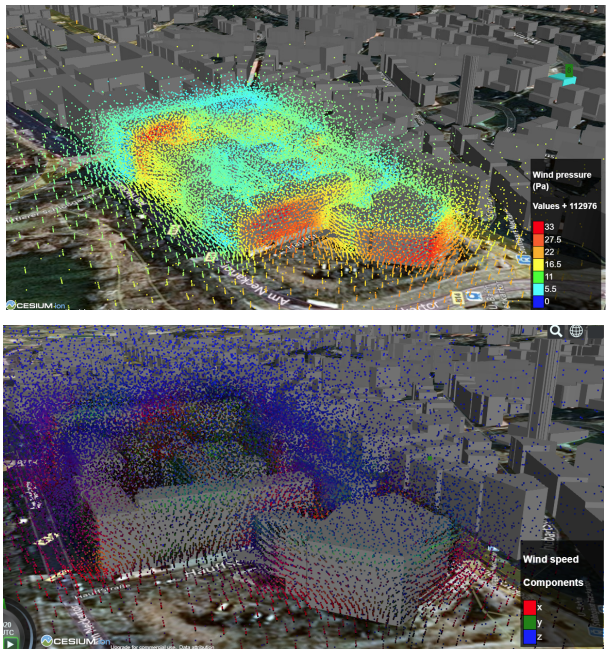


Figure 7. Visualization of a wind field and 3D buildings as 3DTiles (derived from CityGML). Top images shows visualization of wind pressure as points in different colours ranging from low (blue) to high (red). Bottom images shows wind speed encoded as RGB, where the R,G,B corresponds to components of wind speed in x,y,z, respectively.

the 3D tileset generated using FME (Feature Manipulation Engine[7]) was only around 1.45 MB small. ❷ In another "real-world" performance test, where the subjective loading times of the web application were assessed, the entire application loaded faster using the 3D tileset. This was not just only because the file size was smaller, but also because only a small portion of the entire tileset was loaded at the beginning (i.e. when entering the web application) and only on zoom or pan within the Cesium view, more tiles were streamed seamlessly. This subjectively was a much smoother experience than when using the geoJSON format where the entire file needs to be downloaded at once. ❸ Furthermore, what was even more noticeable was the performance when rendering the elements and moving through the scene. There was absolutely no lag or drop in performance when adding the *hex-grid* as 3D Tiles; however, the performance dropped noticeably when all the polygons were added using the geoJSON format. ❹ There was however, an advantage when using the geoJSON, namely that the polygons were clamped to the ground and buildings, which gave a much better look and feel than with a floating plane (3D Tiles) at building height. Of course, this is a highly application-specific demand, and adaptions or manipulation within Cesium for the tileset could lead to similar results.

## 5. CONCLUSIONS

In this paper, we have examined a number of user interactive visualization schemes for highly dense data. State of the art and established OGC standards were employed to develop our Smart Cities Platform based on the Cesium digital globe library and OGC web standards. We have demonstrated several different possibilities for visualizing CFD urban wind data, which

[7] https://www.safe.com/

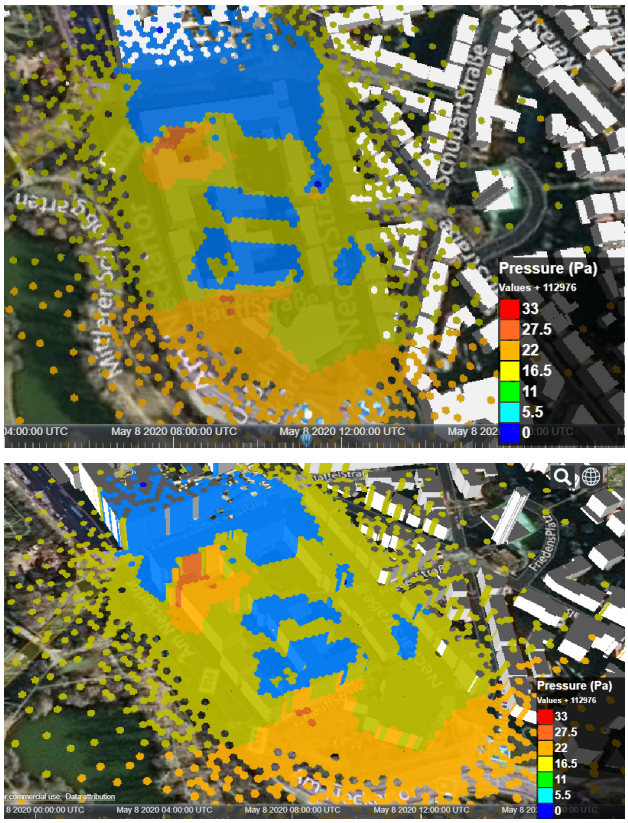


Figure 8. Visualization of pressure at roof top level (slice height = 5 m) showing the same building block as in Figure 7. Wind pressure layer was requested from the 3DPS as a *hex-grid* (hexagon width of 5 m) using 3D Tiles (top) and *geoJSON* (bottom).

Table 2. Average loading times (out of 10 iterations), measured using Google Chrome Developer Tools.

| **Layer (data format)** | **Resource Sizes** | **Loading Time** |
|---|---|---|
| *Hex-grid* (3D Tiles) | 1.45 MB | 673 ms |
| *Hex-grid* (geoJSON) | 3.07 MB | 952 ms |
| Streamline (3D Tiles) | 1.05 MB | 452 ms |

may be a vital aspect for developing and administering smart cities in the future. Drawbacks of the proposed methods were discussed, such as the limitation of point cloud-based data visualization of vectorial data or rather the way such representations can be interpreted by a user.

Future work will focus on integrating a user-controllable *hex binning* capability to select the appropriate level of detail the user wants to see. Furthermore, integrating fully connected streamlines using Cesium specific formats such as CZML polylines might provide a better interpretation of results. User acceptance studies should also be conducted to assess how well the different formats and schemes allow users to navigate, understand and interpret urban CFD result visualizations.

## ACKNOWLEDGEMENTS

This work has been developed in the project *i_city* (Funding number: 03FH9I01IA). *i_city* is supported by the German Federal Ministry of Education and Research (BMBF). The authors are responsible for the content of this publication. The authors gratefully acknowledge the Stadtmessungsamt Stuttgart for the data of 3D citymodel of Stuttgart.
Further gratitude goes to M. Deininger, M. von der Gruen and U. Voss, for providing the CFD data set.


Figure 9. Pressure showing same building block as Figure 7. Wind pressure was averaged in the z-domain across the entire building height (slice height 31 m). Data were requested from the 3DPS as a *hex-grid* - layer (hexagon width of 15 m) .

## References

Albakour, M.-D., Macdonald, C., Ounis, I., Clarke, C. L. A., Bicer, V., 2014. Information Access in Smart Cities (i-ASC). M. de Rijke, T. Kenter, A. P. de Vries, C. Zhai, F. de Jong, K. Radinsky, K. Hofmann (eds), *Advances in Information Retrieval*, Springer International Publishing, Cham, 810–814.

Blocken, B., 2014. 50 years of Computational Wind Engineering: Past, present and future. *Journal of Wind Engineering and Industrial Aerodynamics*, 129, 69 - 102.

Cesium 3d Tiles Creator, 2019. https://github.com/tum-gis/cesium-point-cloud-generator. Accessed: 2019-12-05.

Cozzi, P., Lilley, S., Getz, G., 2019. 3d tiles specification 1.0. Technical report, Open Geospatial Consortium.

Döllner, J., Hagedorn, B., Klimke, J., 2012. Server-based rendering of large 3d scenes for mobile devices using g-buffer cube maps.

Franke, J., Hellsten, A., Schlunzen, K. H., Carissimo, B., 2011a. The COST 732 Best Practice Guideline for CFD simulation of flows in the urban environment: a summary. *International Journal of Environment and Pollution*, 44(1-4), 419-427.

Franke, J., Hellsten, A., Schlunzen, K. H., Carissimo, B., 2011b. The COST 732 Best Practice Guideline for CFD simulation of flows in the urban environment: a summary. *International Journal of Environment and Pollution*, 44(1-4), 419-427.

Friston, S., Doboš, J., Wong, C., Fan, C., Montero, S., Steed, A., 2019. Rectangular selection of components in large 3d models on the web. *The 24th International Conference on 3D Web Technology*, Web3D '19, Association for Computing Machinery, New York, NY, USA, 1–9.

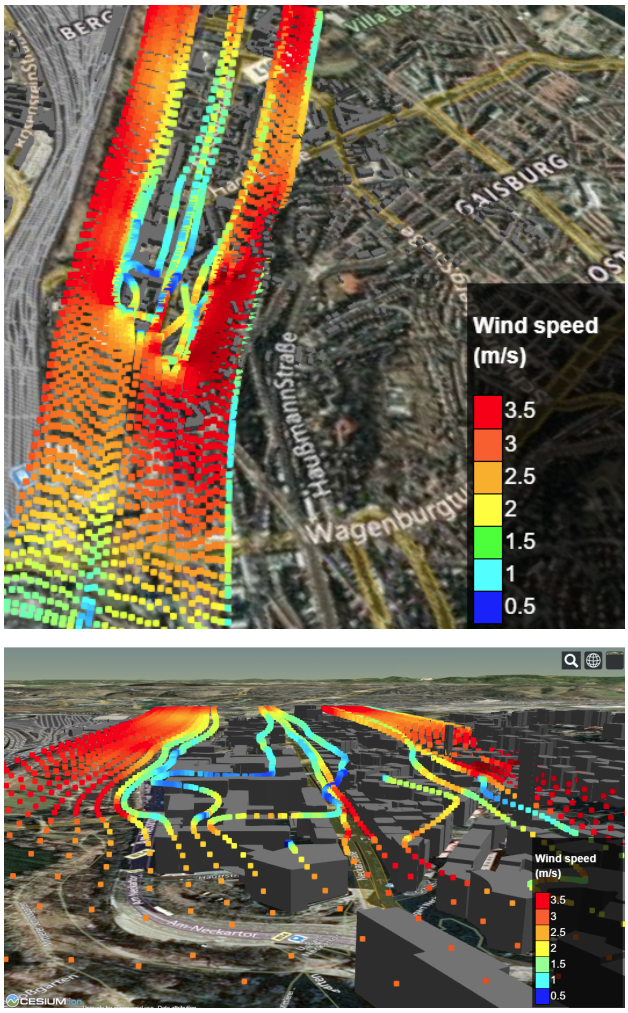


Figure 10. Visualization of streamlines in Cesium. Top: Overview of a city quarter. Bottom: zoomed view on building block. Streamlines are at building height and flow over and around roofs and buildings.

Gaillard, J., Vienne, A., Baume, R., Pedrinis, F., Peytavie, A., Gesquière, G., 2015. Urban data visualisation in a web browser. *Proceedings of the 20th International Conference on 3D Web Technology*, Web3D '15, Association for Computing Machinery, New York, NY, USA, 81–88.

Gutbell, R., Pandikow, L., Coors, V., Kammeyer, Y., 2016. A framework for server side rendering using ogc's 3d portrayal service. *Proceedings of the 21st International Conference on Web3D Technology*, Web3D '16, Association for Computing Machinery, New York, NY, USA, 137–146.

Hagedorn, B., Thum, S., Reitz, T., Coors, V., Gutbell, R., 2017. OGC 3D Portrayal Service. Version:1.0. http://www.opengis.net/doc/IS/3dps/1.0.

KhronosGroup, 2020. Khronos finalizes gltf 1.0 specification. https://www.khronos.org/news/press/khronos-finalizes-gltf-1.0-specification. Accessed: 2020-04-04.

Koukofikis, A., Coors, V., Gutbell, R., 2018. INTEROPERABLE VISUALIZATION OF 3D CITY MODELS USING OGCâ€™S STANDARD 3D PORTRAYAL SERVICE. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-4, 113–118.

Lindsay, J., 2016. Whitebox GAT. *Comput. Geosci.*, 95(C), 75–84.

Patterson, D., 2016. Interactive 3d web applications for visualization of world health organization data. *Proceedings of the Australasian Computer Science Week Multiconference*, ACSW '16, Association for Computing Machinery, New York, NY, USA.

Piepereit, R., Deininger, M., Kada, M., Pries, M., Voß, U., 2018. A SWEEP-PLANE ALGORITHM FOR THE SIMPLIFICATION OF 3D BUILDING MODELS IN THE APPLICATION SCENARIO OF WIND SIMULATIONS. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-4/W10, 151–156.

Portmann, E., Finger, M., Engesser, H., 2017. Smart Cities. *Informatik-Spektrum*, 40(1), 1–5.

Prieto, I. n., Izkara, J. L., 2012. Visualization of 3d city models on mobile devices. *Proceedings of the 17th International Conference on 3D Web Technology*, Web3D '12, Association for Computing Machinery, New York, NY, USA, 101–104.

Simon Thum, B., Reitz, T., Coors, V., Gutbell, R., 2017. OGC 3D Portrayal Service Standard.

Suomisto, J., Airaksinen, E., Bergstrom, M., Heinonnen, H., Lahti, K., Kaisla, K., 2019. The kalasatama digital twins project. Technical report, Helsinki 3D+.

Tominaga, Y., Mochida, A., Yoshie, R., Kataoka, H., Nozu, T., Yoshikawa, M., Shirasawa, T., 2008. AIJ guidelines for practical applications of CFD to pedestrian wind environment around buildings. *Journal of Wind Engineering and Industrial Aerodynamics*, 96(10), 1749 - 1761. 4th International Symposium on Computational Wind Engineering (CWE2006).

Toparlar, Y., Blocken, B., Maiheu, B., van Heijst, G., 2017. A review on the CFD analysis of urban microclimate. *Renewable and Sustainable Energy Reviews*, 80, 1613 - 1640.

Wang, B., Sun, S., Duan, M., 2018. Wind potential evaluation with urban morphology - A case study in Beijing. *Energy Procedia*, 153, 62 - 67. 5th International Conference on Energy and Environment Research, ICEER 2018, 23-27 July 2018, Prague, Czech Republic.