DYNAMIC AND WEB-BASED 4D VISUALIZATION OF STREETSPACE ACTIVITIES DERIVED FROM TRAFFIC SIMULATIONS AND SEMANTIC 3D CITY MODELS

C. Beil[1,*], M. Kendir[1], R. Ruhdorfer[2], T. H. Kolbe[1]

[1] Chair of Geoinformatics, Technical University of Munich, Germany - (christof.beil, murat.kendir, thomas.kolbe)@tum.de
[2] virtualcitySYSTEMS GmbH, Tauentzienstraße 7b/c, 10789 Berlin, Germany - rruhdorfer@vc.systems

Commission IV, WG IV/9

KEY WORDS: Digital Twin, 4D visualization, 4D tiling, City Model, Traffic Simulation, CityGML, CZML, SUMO

ABSTRACT:

Semantic 3D city models can serve as anchor points for different components of urban digital twins. In addition to static 3D models such as buildings, transportation infrastructure, vegetation, or city furniture, this can also include dynamic processes such as traffic movement or changing traffic signals. Integrating these aspects into a dynamic, realistic, and accessible 4D visualization presents a number of requirements and challenges, which are discussed. While the City Geography Markup Language (CityGML) is a well established OGC standard for modeling and exchanging semantic 3D city models, the Cesium Language (CZML) provides capabilities for visualizing time-dependant properties that can be displayed in the Cesium virtual globe. Results of the open-source microscopic traffic simulation tool SUMO include information on locations and orientations of vehicles, bicycles, pedestrians and other traffic members as well as traffic signal information and can be exported as a CSV table. In order to allow a 4D visualization of these simulation results, a scheme for deriving 3D + t CZML documents from data given as 2D + t CSV files is presented. Additionally, dynamically changing traffic lights are integrated. Based on data available for the city of Munich, a CityGML-compliant streetspace model is generated. This enables the combination of traffic simulation results and semantic 3D city models within a common Cesium based web-visualization and allows a direct and platform independent access to realistic visualizations of streetspace activities. Since this often results in a huge amount of data, a spatio-temporal tiling strategy enabling the visualization of large CZML data is presented.

1. INTRODUCTION

The term ’digital twin’ is commonly used to describe virtual representations of real-word objects that can be used for simulations, analysis, and many other applications (Grieves and Vickers, 2017). Recently, this approach was transferred to the field of smart cities in the context of urban digital twins (Dembski et al., 2020). While there is no exact definition of this concept and its components yet, semantic 3D city models can serve as an anchor point for linking different elements of urban digital twins. In addition to static 3D representations of the environment such as buildings, vegetation, or transportation infrastructure, this also includes dynamic and time-dependant processes such as traffic flow and other streetspace activities (e.g. changing traffic lights). Changes in mobility concepts of cities (e.g. replacing car driving lanes with bicycle paths) will have an effect on traffic movement but also on aspects such as air quality. Similarly, temporary events such as closed roads have an impact on the traffic and thus on citizens (Amini et al., 2018). Simulating, visualizing, and publicly communicating different scenarios will be increasingly relevant. Digital traffic simulations can be done using available open-source applications such as the micro-traffic simulation tool ”Simulation of Urban MObility” (SUMO) (Behrisch et al., 2011). While these simulations provide insight into the behavior of individual vehicles and traffic participants, visualization capabilities are limited and often require additional software. The combined utilization of 3D vehicle models and detailed 3D representations of cities and landscapes within a common visualization offer the opportunity for a more realistic representation of simulation results. There are some examples for linking traffic simulation tools with game engines, which result in highly realistic visualizations (Artal-Villa et al., 2019). However, this also requires the usage of additional software in order to be able to access these visualizations. In contrast, web-based visualizations offer the advantage to explore these representations in any browser and independent from additional software or operating system. This might be especially important within the context of public participation, e.g. to illustrate different planning scenarios of traffic infrastructure to the public. Additionally, this provides the opportunity to include geo-referenced models of the environment containing semantic information, which can be directly retrieved by clicking on individual objects. While formats such as KML provide some capabilities for representing time-dependant activities, the Cesium Language (CZML) was specifically developed for this purpose and thus is used for visualizing traffic simulation results within a Cesium based interactive web-visualization. In a city-wide context, these kind of simulations (and respective visualizations) will include a huge number of traffic members. While there are mechanisms for tiling static 3D models such as buildings to enable performant streaming, this is currently not available for time-dependant data (Yao, 2020). After presenting some related work and discussing requirements of realistic 4D visualizations of traffic and streetspace activities, a scheme for deriving CZML data from traffic simulation results of a SUMO simulation is shown and implemented for a use-case in the city of Munich. Additionally, a spatio-temporal tiling mechanism for CZML data is proposed in order to allow for a web-based visualization of large time-dependant scenarios in combination with semantic 3D city models.

* Corresponding author

2. RELATED WORK

There is some related work on web-based visualizations of time-dependant and city related processes in general. (Murshed et al., 2018) present a web-based application for visualizing dynamic processes in smart cities such as changing irradiation data on buildings or changing cooling energy needs. Methods for representing time-evolving 3D city models using 3DTiles, including different building states such as creation, modification or demolition, are described in (Jaillot et al., 2020). (Mao et al., 2020) present dynamic style animations of energy simulations using 3DTiles. The integration of dynamic sensor data with city models plays a central role in the context of smart cities. (Macura, 2019) describes a method for visualizing time-dependant sensor data of buildings within a Cesium based 3D GIS application by coloring volumetric geometries representing individual rooms with colors according to temperature measurements at certain times. (Chaturvedi, 2021) presents dynamic visualizations of building energy demands in Cesium derived from live sensor data. (Kurkcu et al., 2017) visualize bus time data, including color-coded speed-per-section representations and bus stop times. However, this visualization does not include individually moving objects. (Schwab et al., 2020) show how pedestrian and vehicle simulation results can be coupled and visualized within simulation environments such as Virtual Test Drive (VTD). Furthermore, there are some publications focusing specifically on the visualization of microscopic traffic simulations. (Chen et al., 2015) give an overview and categorization of traffic data visualizations and describe different typical processing stages of generating visualizations from traffic data. (Chao et al., 2020) present a survey on different approaches to animated visualizations of traffic simulations. (Xu et al., 2022) present an application for managing and visualizing traffic simulation results. The application can work with both Vissim and SUMO simulation results and visualizes moving vehicles and static signals using colored dots overlayed with a 2D map, whereby the color of individual points depends on their current speed. Additionally, traffic densities can be represented using a heat-map visualization. Most of the traffic simulation visualizations mentioned so far mainly focus on representing vehicle movements using point geometries or more abstract representations of general traffic movement or traffic volume. However, a more realistic visualization of individual traffic participants as well as other streetspace activities could be achieved using 3D models. (Keler et al., 2018) present a Virtual Reality (VR)-based, ego-perspective bicyclist visualization coupled with traffic simulation results. (Artal-Villa et al., 2019) present methods for coupling traffic simulations with game engines such as Unity. Simulation tools such as 'Car Learning to Act' (CARLA) are coupled with game engines (the Unreal Engine in this case) and thus (in addition to realistic simulations) provide highly detailed visualizations (Dosovitskiy et al., 2017). While this results in realistic and visually appealing visualizations, this also requires specific apps or software for viewing these representations. Furthermore, game engines usually use a local reference system and do not support geo-referenced data. Thus, it is difficult to integrate other data sources (e.g. point clouds or city models) ad-hoc within a common visualization. A more accessible approach for visualizing traffic simulation results can be done by coupling these with semantic 3D city models and creating web-based visualizations using virtual globes, which also support geo-referencing. (Ruhdorfer et al., 2018) derived input data for the micro-traffic simulation tool Vissim from semantic 3D streetspace models and then visualized traffic simulation results in GoogleEarth using KML. (Yao, 2020) demonstrates methods for extending the Cesium virtual globe in order to visualize large city models and also briefly mentions the possibility to include data available in the Cesium Language (CZML) format. (Chaturvedi et al., 2019) present how the concept of Dynamizers can be used for integrating and visualizing dynamic data within semantic 3D city models. Several publications present how CZML can be used for visualizing dynamic processes in the Cesium virtual globe (Zhu et al., 2018a).

3. CONCEPTUAL CONSIDERATIONS ON WEB-BASED 4D TRAFFIC SIMULATION VISUALIZATIONS

Traffic simulations can be done in different levels of detail. (Chao et al., 2020) give an overview on visualization methods for macroscopic, mesoscopic and microscopic traffic simulations. While macroscopic simulations describe traffic flows on a large scale without representing individual vehicles, microscopic simulations contain information on the movement of discrete agents. Mesoscopic simulations can be placed between those two levels of detail. Sub-microscopic (driving) simulations are considered to have the highest level of detail and additionally include more precise simulations of vehicle behaviors and driving maneuvers. In this paper the focus is on visualizing microscopic traffic simulations for two main reasons: (1) simulation networks can be created using widely (and openly) available data (such as OpenStreetMap) that allow simulations on a city scale, and (2) these simulations can be done using open-source simulation tools such as SUMO (Behrisch et al., 2011).

3.1 Requirements for and challenges of web-based traffic simulation visualizations

There are a number of requirements for a realistic time-dependant and web-based representation of microscopic traffic simulation results, which impose some challenges upon the creation of such visualizations. Visualizing results of a traffic simulation requires time series data. Usually those trajectories of traffic participants are given as discrete points over time. Additional properties describe the orientation of each agent. A first requirement for an accurate visualization is a sufficient resolution of the time series data in position and time. This means balancing between number of points necessary for describing an accurate trajectory and an acceptable amount of data. These discrete points then need to be interpolated in order to generate a smooth visualization of moving objects. Visualizing the results of a traffic simulation in a city context imposes additional requirements. Road traffic is commonly simulated based on street networks in a 2D linear graph representation. The time series points of an agent are then derived from this network and are therefore only as accurate and smooth as the input geometry. A second requirement for accurate visualizations are matching models of the environment. Corresponding street space models are often created from polygonal road geometries originating from different data sources than the simulation network and thus may diverge slightly. For an integrated visualization, data sources for simulation and visualization must match within centimeter accuracy. If elevation is neglected within the traffic simulation, the z-coordinates must be added to the positions from a corresponding elevation model. Special consideration is required concerning ramps, bridges or tunnels, which distinguishes a 2.5D from a true 3D visualization. Additional 3D city furniture and street space objects increase the realism of visualizations. For dynamically changing traffic signals information on the exact 3D location of traffic lights and controller series from the simulation are needed. Microscopic traffic simulations can be done on a city scale and potentially over a long period of time. A large number of 3D models for the static environment representation together with suitable models for each type of traffic participant soon result in a huge amount of data. A third requirement for web-based traffic simulation representations therefore is a good streaming performance of the visualization. This requires concepts for tiling and a smart representation of relevant data. There are strategies for tiling large static 3D city models in order to allow a performant web-based visualization. While dynamic data can be streamed incrementally, a similar concept for spatio-temporal tiling of dynamic data is necessary. Size and level of detail of used 3D models also play a role. Animations on dynamic traffic members such as rotating tires increase the degree of realism. Here, again, a balance between realistic and performant visualizations needs to be found.

3.2 Standards and data formats relevant for web-based visualizations of time-dependant processes

Web-based visualizations can be done using virtual globes such as Cesium or GoogleEarth. This limits available data formats that can be used to display spatio-temporal information in such environments to the Keyhole Markup Language (KML) and the Cesium Language (CZML). While it is possible to reference data represented in the 3DTiles format within CZML, 3DTiles natively does not support dynamic 4D representations. With KML it is possible to represent time-dependant data either by specifying a 'TimeStamp' or 'TimeSpan' for which an object is visible or by defining a 'Track' on which a point geometry or a 3D model moves over a given time period. In contrast to CZML, KML does not support dynamic updates of semantic information or real-time data transmission (Zhu et al., 2018b). Thus, CZML is considered to be the most suitable data format for visualizing traffic simulation results within a web-based visualization. A detailed explanation of the structure and contents of CZML files is given in section 4.1. A standard worth mentioning in this context is the 'OGC - Moving Features Access' (Hayashi et al., 2017). Data available according to this standard could be transferred to visualization formats such as CZML using methods similar to the process described in section 4.4.

4. VISUALIZING TRAFFIC SIMULATION RESULTS IN COMBINATION WITH 3D CITY MODELS

4.1 Introduction to the Cesium Language (CZML)

The current version 1.0 of CZML was developed by the company AGI and can be used for visualizing time-dependant data in the Cesium virtual globe. A documentation of the standard and its concepts is provided in the CZML Guide[1]. (Zhu et al., 2018b) give a detailed explanation of the structure and content properties of CZML documents. In the following example the most relevant parts of the structure and properties of CZML documents with respect to visualizing streetspace activities are described using an example derived from traffic simulation results. CZML documents (.czml) are based on the JavaScript Object Notation (JSON) data format and consist of multiple sequential JSON objects called packets. The first packet in any CZML file is a document object defining an 'id', 'name' and 'version' of the CZML file (blue in Figure 1). Information on the time interval (start and end time of the simulation) as well as the desired current (starting) time of the visualization can be contained within a 'clock' property. All properties are stored as a 'name : value' pair. Individual packets for each object are created (green in Figure 1). First, the 'id' of each traffic member and an optional 'name' property is stored. Then a 'model' property containing a relative path to a folder referencing different .gltf models depending on the type of traffic member (e.g. pedestrians, bicycles or trams) is included. A 'heightReference' property is set to 'NONE' since this allows the vertical position of models to correspond to their z-values. In order to hide models at a certain zoom-level, a 'distanceDisplayCondition' is set to a range of 1 to 800 meters indicating the visibility of objects based on camera distance. Within the position property the geographic coordinates and elevation values of objects are given for time steps (one second in this case) relative to the time defined within the respective 'epoch' property. Similarly, the orientation of objects is specified with a unit quaternion representation of angles for time steps relative to the 'epoch' property. CZML offers the opportunity to specify an interpolation algorithm, which is used to interpolate data between time-tagged values. If not further specified, a linear interpolation is used.

```
[
 {
  "id" : "document",
  "name" : "TrafficSimResults",
  "version" : "1.0",
  "clock" : {
   "interval" :
   "2021-10-18T09:03:49.449Z/2021-10-18T10:10:49.0489Z",
   "currentTime" : "2021-10-18T09:05:40.449Z",
   "multiplier" : 1
  }
 },
 {
  "id" : "car01",
  "name" : "car01",
  "model" : {
   "gltf" : "./Modelle/1.gltf",
   "heightReference" : "NONE",
   "distanceDisplayCondition" : {
   "distanceDisplayCondition" : [ 1, 800 ]
   }
  },
  "position" : {
   "epoch" : "2021-10-18T09:03:49.449Z",
   "cartographicDegrees" :
   [ 0.0, "11.568003", "48.132709", 517.89,
     1.0, "11.567893", "48.132711", 518.00,
     ...
   ]
   },
  "orientation" : {
   "epoch" : "2021-10-18T09:03:49.449Z",
   "unitQuaternion" :
   [ 0.0, -0.21071101232792602, -0.3013600122118491,
          -0.6864276868642915,  -0.6273755199476252,
     1.0  -0.21621932000480018, -0.2974324100898577,
          -0.6978521288546177,  -0.6146426387242255,
     ...
   ]
   }
 }
]
```

Figure 1. Example CZML file derived from SUMO traffic simulation results

[1] https://github.com/AnalyticalGraphicsInc/czml-writer/wiki/CZML-Guide

4.2 Introduction to the City Geography Markup Language (CityGML)

The international standard CityGML published by the Open Geospatial Consortium (OGC) is well established for modeling, storing and managing semantic 3D city models (Kutzner et al., 2020). CityGML allows the modeling of urban objects with their 3D geometry and 3D topology, semantics and appearance in four different levels of detail. Additionally, time-dependant properties can be represented. The conceptual model of the newest version CityGML 3.0 was published in September 2021 (Kolbe et al., 2021). The modular structure of the standard provides multiple thematic modules for representing different parts of cities and landscapes. The most relevant module in the context of visualizing traffic simulation results is the Transportation Module used for representing streetspace and other transportation infrastructure. (Beil et al., 2020) present several examples for detailed semantic 3D streetspace models according to CityGML. So far, CityGML is most commonly used for representing buildings. However, there are also examples for modelling city furniture or vegetation using the standard (Floros and Dimopoulou, 2016). Tools such as the open-source 3DCityDatabase are capable of transforming CityGML data into visualization formats such as COLLADA or GLTF, which then can be streamed in Cesium-based web-visualizations such as the 3DCityDB Web-Map Client (Yao et al., 2018). This also allows the integration and interactive availability of semantic information.

4.3 Microscopic traffic simulations using SUMO

"Simulation of Urban MObility" (SUMO)[2] is an open-source software package for microscopic traffic simulations developed by the German Aerospace Center (DLR) (Behrisch et al., 2011) capable of handling networks on a large scale. Simulations can include different traffic types such as vehicles, bicycles or pedestrians. A SUMO-Network is a directed graph, with junctions of a road network represented by nodes and linear connections represented by edges (Lopez et al., 2018). To create a SUMO-Network, different input files are needed. One file should contain the nodes, one should specify the edges and one should specify the connections between the edges. The connections define which lane of an origin edge is connected to the lane of a destination edge. The file format is based on XML. While it is possible to include elevation data into the simulation, the SUMO network data in this example is given in cartesian 2D coordinates. OpenStreetMap (OSM) data is used as the foundation from which the SUMO network is built. Based on aerial images and other reference data, the OSM-based SUMO network is manually adjusted in order to be as accurate and up to date as possible. The SUMO network is calibrated using information on traffic signals as well as traffic counts and detector data for the relevant area. The accuracy of SUMO simulation results is limited by the accuracy of the underlying input data from which the simulation network is generated. Output lon/lat geo-coordinates are given with a precision of six decimal places. There are a range of output files available such as raw vehicle positions or emissions. Another file called 'Floating Car Data Output' (FCDOutput) contains information on location (longitude / latitude coordinates), orientation (angle), speed, vehicle types and other information for certain time steps. The angle describing the orientation of traffic members is given according to the navigational standard (0-360 degrees, going clockwise with 0 at the 12 o'clock position). The position given with lon/lat coordinates corresponds to the front of a vehicle. While different time-steps are possible, experiments showed, that a one second step between locations is sufficient to achieve a smooth visualization later on. It is also possible to include elevation data within the SUMO simulation directly. In this case slope values can also be part of the output file. However, for OSM data this is still experimental. The area of interest (approximately 0.5 km x 1 km) will be visualized later. In order to get realistic simulation results, the actual traffic simulation is conducted for a larger area in the center of Munich. Multiple simulations with different traffic types and time-steps are conducted.

[2] https://sumo.dlr.de/docs/index.html

4.4 Converting SUMO simulation results to CZML

The SUMO simulation results are then converted to the CZML format with a conversion workflow created using the Feature Manipulation Engine (FME). While FME does not support CZML natively, it is possible to create and write text documents with a corresponding file structure. Input data for this conversion process are the CSV data of the SUMO FCDOutput as well as a digital elevation model with a resolution of one meter provided as GeoTIFF. The workflow for generating a CZML document from SUMO simulation results is illustrated in Figure 2. First, point geometries are created from the posi-

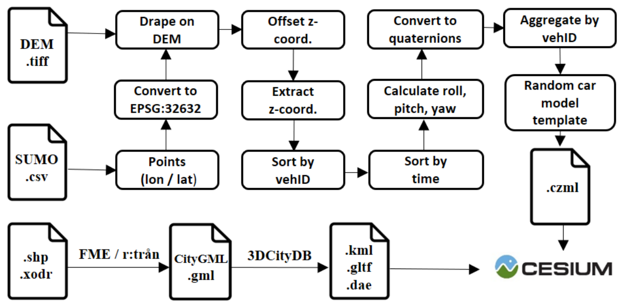

Figure 2. Workflow for generating and visualizing CZML documents from SUMO traffic simulation results

tional information in order to adapt the z-values of these points to the underlying terrain, the point geometries are transformed into the 'WGS 84 / UTM zone 32 N' (EPSG:32632) coordinate system and then draped on the digital elevation data. A vertical offset to each position can be included in order to avoid potential rendering problems due to multiple layers at the same height level. After storing the final z-coordinate of each point as a new attribute, all contents of the CSV file are sorted by their 'vehId' and 'time' attribute. Optionally, it is possible to filter for specific vehicle types, locations (bounding box) or time spans. Then, the orientation of traffic members given as roll, pitch and yaw values is calculated. 3D models are placed in their 'body frame' as shown in Figure 3. The placement of models is offset from the center of gravity, so that reference points represent the front of vehicles. According to the CZML Guide, the orientation property in CZML is defined as a vector that represents the 'body axes' of an object transformed to the 'Earth fixed axes'. The transformation between the 'Earth Centered Earth Fixed' (ECEF) frame and the 'East-North-Up' (ENU) local tangent plane coordinate frame is depending on the location of models on the Earth's surface (longitude (λ) / latitude (φ) coordinates). Some rotations need to be performed in order to align the vehicle roll, pitch, yaw axes with the East-North-Up axes. First, 3D models are rotated around the z-axis in order to point towards the North-axis. This depends on the local longitude as well as the orientation of the 3D model in its 'body frame'. Individual orientations of 3D models at a certain time then correspond to the angle value derived from the SUMO simulation. Then, the pitch and roll angles are adapted based on the local latitude and longitude coordinates, so that models are placed correctly on the local tangent plane.

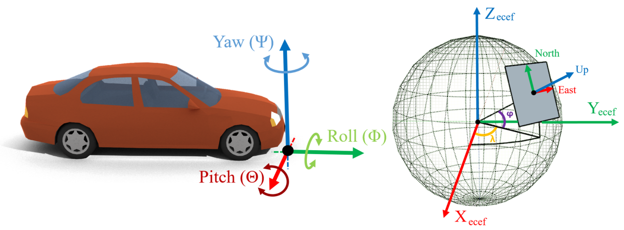

Figure 3. Left: Roll, pitch and yaw angles in the 'body frame' of the 3D model. Right: ECEF and ENU frames

After a conversion from degrees to radians, the angles then need to be converted to unit quaternions using the following formula.

$$\begin{bmatrix} qx \\ qy \\ qz \\ qw \end{bmatrix} = \begin{bmatrix} cos(\phi/2)cos(\theta/2)cos(\psi/2) + sin(\phi/2)sin(\theta/2)sin(\psi/2) \\ sin(\phi/2)cos(\theta/2)cos(\psi/2) - cos(\phi/2)sin(\theta/2)sin(\psi/2) \\ cos(\phi/2)sin(\theta/2)cos(\psi/2) + sin(\phi/2)cos(\theta/2)sin(\psi/2) \\ cos(\phi/2)cos(\theta/2)sin(\psi/2) - sin(\phi/2)sin(\theta/2)cos(\psi/2) \end{bmatrix}$$

Every conversion and calculation step so far is done for each point individually. Now, information is aggregated based on corresponding 'vehId' attributes. In this process individual CZML packets are created for each 'vehId'. Interval and epoch times are specified according to the time span of the simulated scenario. Time-step, position and orientation values are added to each packet iteratively. Relative paths to a folder containing a bibliography of different 3D models are created based on 'vehType' attributes. The 3D models used were downloaded from the SketchUp Warehouse and converted to .gltf using FME. The origin of 3D models is placed as shown in Figure 3. This corresponds to the front of a vehicle, which is also the reference point of positions used in SUMO. This method also takes into account different heights of the used 3D models and places the models so that tires align with the ground (this is another reason, why the 'heightReference' is set to 'NONE'). Additionally, optional point geometry representations visible from a certain zoom-level can be added. Cesium offers an animated model called 'Cesium Man', which is used to represent pedestrians. In case the 'vehType' property is 'tram', 'person' or 'truck' corresponding model paths are set. For cars, seven different 3D models are assigned semi-randomly by setting the relative path to a 3D model called [1-7].gltf. Models are assigned to vehicles sorted by 'vehId'. This is relevant in order to assign the same 3D model to vehicles each time the workflow is run. If the exact information on specific vehicle types were available, corresponding 3D models could be referenced. Finally, the resulting document is formatted and written using the FME 'Text File' writer with an output file extension set to '.czml'. Low-poly 3D models were used in order to keep the number of geometries to a minimum. More detailed models could easily be integrated by substituting models in the folder referenced by CZML objects, while sticking to the described naming convention of 3D models. The number of different 3D models used could also be changed easily. In general this workflow is not limited to SUMO outputs but could also be applied to results of traffic simulation tools such as Vissim or other moving objects. Time-dependant objects represented according to the OGC standard 'Moving Features Access' (Hayashi et al., 2017) for example could also be processed to CZML in a similar way.

4.5 Dynamic visualization of traffic lights

In order to consider additional dynamic streetspace activities, a QGIS plugin[3] capable of generating 3D models of dynamically changing traffic lights has been implemented. The plugin is able to read point based vector layers, to specify CZML object properties, to set time intervals in which objects should be visible (availability), to specify 3D models and to export results as CZML files. Figure 4 shows the graphical user interface of the plugin. Two types of poles (i-type (without arm) and l-type (with arm)) are created. Azimuth angles of poles and traffic lights are calculated based on their position relative to a corresponding street centerline. Azimuth angles of objects should be calculated on a projected (and conformal) coordinate system. In this example, the coordinate system 'ETRS89 / UTM zone 32 N' (EPSG:25832) is used before transforming positions to the 'WGS84' (EPSG:4326) geographic coordinate system. While

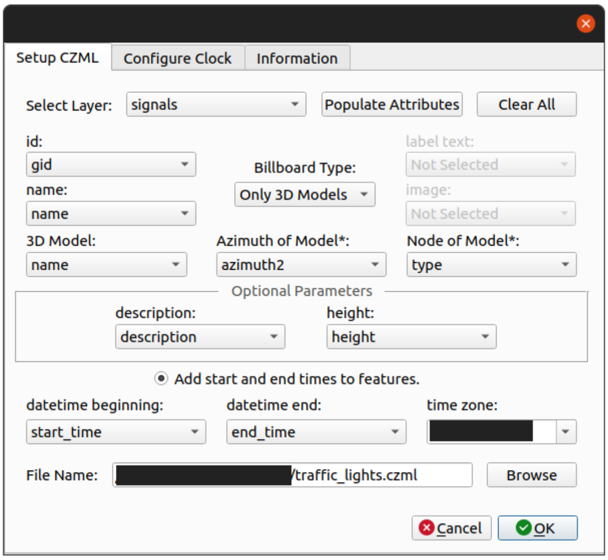

Figure 4. CZML export parameters in a QGIS plugin for creating poles and dynamic traffic lights

poles are represented as static 3D models, three types of vehicle traffic lights (red, yellow, green) and two types of pedestrian traffic lights (red and green) are represented using corresponding 3D GLTF models. Due to the fact that multiple traffic lights can be mounted on the same pole on different heights and directions, all traffic lights are related to poles by using local ID attributes. Since traffic lights should be visualized as assembled objects to the poles, origin points of all 3D models need to be moved carefully to the reference point of rotation using a 3D editor. In addition, 3D models should share only one single node, because node names will be used as reference point of rotations in the CZML file. Traffic signals can be exported to CZML for specified time sequences each referencing different 3D models. In the QGIS plugin, these sequences can be set using the 'beginning time' and 'end time' fields. For instance, if traffic light phases loop three times in a certain time period with three different color phases each, that means 9 rows with disjoint time sequences are created for each object. In the QGIS software, the 'Dynamic Temporal Control' option helps filtering features by

[3] https://plugins.qgis.org/plugins/CZMLBillboardMaker/

considering date time attributes and also a toolbar called 'Temporal Controller' can be used to animate features while feature categories are visualized with different styles. In this way, time sequences of traffic lights can be tested on a 2D map canvas before exporting 3D models to a CZML file. Unlike the export process of poles, traffic signals are exported as animated objects between given time intervals. Since SUMO simulation results also include information on time-dependant traffic light status, this output can be used as input information for this plugin.

4.6 A tiling strategy for CZML documents

Even though contents of CZML documents can be streamed incrementally (using individual packets), performance within a browser-based visualization may decrease depending on the number of objects, the extent of the area of interest, the detail of referenced 3D models (number of geometries) and the time-span covered by the simulation. This is especially relevant for visualizing streetspace activities on a city-wide scale and over a longer period of time. While there are tools and strategies available for tiling large static 3D city models (e.g. using the 3DCityDB Exporter), it would be beneficial to have similar processes for dynamic and time-dependant data provided in the CZML format in order to improve loading within web-based viewing applications. (Yao, 2020) describes a lightweight tiling schema for a grid-based tiling of geo-spatial data. A similar mechanism is created and prototypically implemented for tiled CZML data. Objects (positions of traffic members) lying within predefined 2D bounding boxes are assigned to corresponding tiles organized within a hierarchical directory structure. Additionally, a JSON file containing meta-information (number of rows and columns, bounding box) on the created tiles are provided. Based on the current camera view within a web application, only visible tiles are loaded. Tiles are labeled according to their column and row within the grid as illustrated in the left image of Figure 5. Tiles are stored in such a way, that all tiles of a certain row are located within a corresponding sub-folder and named accordingly (numbering starting with 0). Within each row sub-folder additional sub-folders corresponding to respective columns are available. All sub-folders are contained within a root-folder called 'tiles'. Each CZML file is called 'tile.czml' and the 'name' property of each CZML document must correspond to the respective row / column (e.g. 0/0). In order to prevent the 'clock' to be reset every time a new CZML tile is loaded, 'epoch' properties are defined for each object individually, indicating the time at which the objects appear. An 'availability' property is set in the header packet at the beginning of the document. This mechanism can be extended further by considering a time-dependant tiling strategy as illustrated in Figure 6. Here, tiles for four consecutive time-spans are created. This means, that additionally to the 2D location based tiling, a time-based tiling is used. This can be achieved either by creating individual CZML files covering different time periods or by splitting CZML packets of one object into consecutive packets (with the same ID) for the respective time spans making use of the incremental streaming ability of CZML.

4.7 Generating a 3D streetspace model

In order to achieve a combined visualization of traffic simulation results with a semantic 3D city and streetspace model, available data sources are evaluated with regard to their suitability to be transformed to CityGML. Polygonal 2D layers on land use as well as transportation areas are converted to CityGML using FME. Streetspace objects are assigned to TrafficAreas

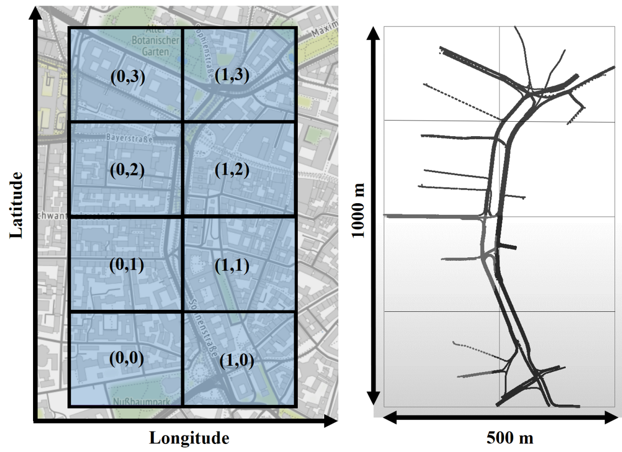

Figure 5. Left: Tile structure and corresponding labeling; Right: Points of SUMO simulation results containing time and orientation attributes

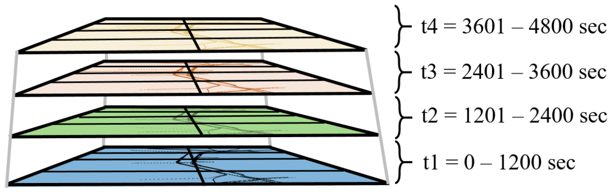

Figure 6. 2D + t tiling: Area of interest segmented into 8 tiles for 4 consecutive time spans

or AuxiliaryTrafficAreas based on function attributes and textured appropriately. The granularity of the resulting streetspace model corresponds to 'way' (according to CityGML 3.0), since traffic surfaces are representing individual carriageways, sidewalks and bicycle paths but do not show individual lanes. Sidewalks and traffic islands are extruded by 15 cm. Additionally, a layer containing information on markings is available, which is also transformed to CityGML. Depending on their function, TrafficAreas and AuxiliaryTrafficAreas are assigned to Road, Track or Railway top-level features. The area of interest also contains a fountain and small ponds, which are assigned to Waterbody features. All objects are then adapted to the digital elevation model. CityGML building models in Level of Detail (LoD) 2 are included into the visualization. SolitaryVegetationObjects are created from 2D point information on tree locations, which also contain attributes on tree height and crown diameter. Low-poly tree models are placed at the corresponding locations using implicit geometries and scaled according to height and diameter information. Building, streetspace, vegetation objects, and other thematic layers are converted to GLTF/COLLADA using the 3DCityDB Exporter and visualized in the '3DCityDB Web-Map-Client'. While CityGML 3.0 streetspace models may not be widely available yet, it is possible to derive such models from polygonal street information or from OpenDRIVE data using the open-source converter r:trån[4].

5. RESULTS

The workflow described in section 4.4 was used for deriving CZML data from a SUMO traffic simulation conducted for an area of interest in central Munich and combined with the pole and traffic light models presented in section 4.5 as well as with

[4] https://github.com/tum-gis/rtron

the 3D streetspace model described in section 4.7. Contents of the resulting CZML files are explained in more detail in section 4.1. The available traffic simulation covered a time period of 1 hour and 20 minutes with over 300,000 individual data points corresponding to 1,359 vehicles, 21 trams, 245 pedestrians and 63 bicyclists. Additionally, an alternative simulation for another time period containing 2,623 vehicles and 682 trucks is conducted (approx. 1,500 polygons per car model). Individual CZML files are created per traffic type and integrated within web-based visualizations using the '3DCityDB Web-Map Client'[5]. Traffic lights generated according to the process described in chapter 4.5 and the generated 3D streetspace model are integrated into the visualization. 36 traffic signals and 23 poles were created for a selected intersection within the area of interest by considering their real-world coordinates and types. Each city object contains semantic information, which can be viewed interactively by clicking on individual features. Several CZML layers are integrated within the visualization. Figure 7 illustrates two points in time, with car models stopping at a red light (top) and moving after the light has changed to green (middle) as well as a panoramic view (bottom) of the visualization including trams and pedestrians. The number of individual traffic members that can be included within a (performantly running) visualization mainly depends on the geometric complexity of individual models, the number of relevant objects and the total time-span of the simulation. The tiling mechanism described in section 4.6 is implemented for a generic example dataset by creating individual CZML files, each corresponding to a tile within a pre-defined grid. In principle, the number of grid cells is not limited, however this needs further testing.

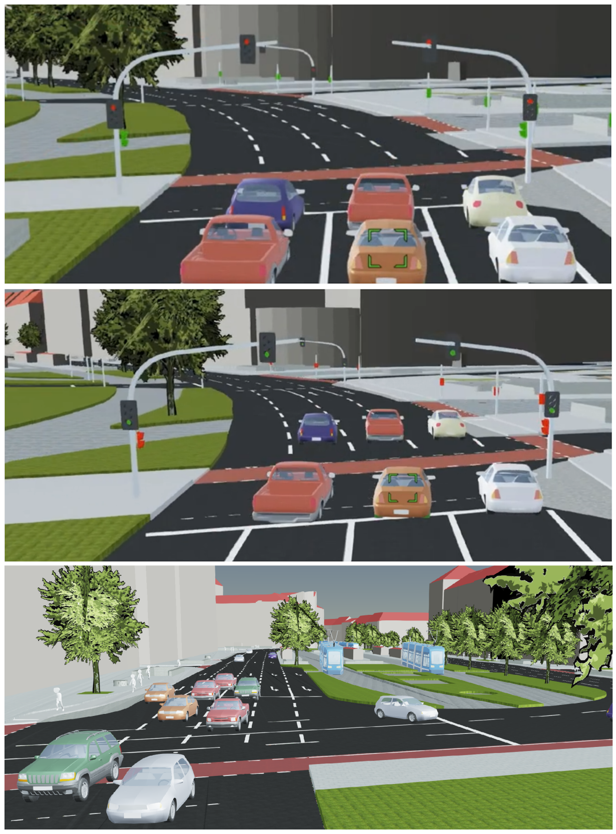

Figure 7. Vehicles and changing traffic signals with time. Top: Stopping cars and red traffic light; Middle: Moving cars and green traffic light; Bottom: Vehicles, trams and pedestrians

6. DISCUSSION AND OUTLOOK

In this paper requirements and challenges of generating realistic, web-based visualizations from microscopic traffic simulations are presented. Then, based on SUMO simulation results, a method for deriving CZML documents from this information is shown. Additionally, dynamically changing traffic lights are generated and visualized in combination with a CityGML streetspace model using the Cesium-based '3DCityDB Web-Map Client'. Furthermore, a tiling mechanism for time-dependant data represented in the CZML format is presented. Due to the nature and level of detail of microscopic traffic simulations (as performed by tools such as SUMO), some 'unrealistic' lateral vehicle movements are visible within the visualization of their results. While sub-microscopic driving simulations also contain highly detailed vehicle behaviors, this is not considered within microscopic traffic simulations. However, more detailed simulation results could also be transferred to CZML using the process described in this paper. While the method for deriving CZML documents from traffic simulations is demonstrated using SUMO, results from other tools (e.g. Vissim) could be processed similarly as long as relevant simulation outputs (position, orientation, vehicle type etc.) are available. The 3D models of vehicles used in this scenario do not contain additional animations such as rotating tires. Animated models could be easily included if available. While the used models of pedestrians do provide a walking animation, this animation also continues when pedestrians are stopping at a certain location. CZML provides a property called 'runAnimation'. This can be used to create different packets for moving and standing pedestrians with this property set to 'true' or 'false' respectively. The used tram model is relatively long, but does not bend when driving around corners. Within the web-based visualization it is possible to lock the current view to a specific traffic member, which then is followed in the continuing visualization. This could be further developed by creating an ego-perspective from the inside of vehicles. The z-values of models are derived from a digital elevation model resulting in 2.5D data. While this is sufficient in most cases, adaptions of height values of vehicles driving over bridges or into tunnels in order to derive a true 3D visualization could be considered. This is also relevant for visualizing pedestrian simulations on multiple floors within buildings. In this case, the tiling strategy presented in section 4.6 would need to be extended to the third dimension (e.g. voxel based). Additionally, the slope at a specific location (e.g due to the terrain or due to ramps and bridges) should be considered when calculating corresponding pitch angles of 3D models. In the future aerial mobility may play an increasing role within urban transportation. The visualization of drone or UAV flights could be integrated within 3D city model environments, too. In this context, the CityGML concept of TrafficSpaces representing the space where traffic actually takes place are relevant for aerial spaces as well. Traffic members currently not included (due to no information contained in the SUMO simulation) such as taxis, buses or cargo bikes could easily be integrated by extending the bibliography of referenced 3D models accordingly. While a scheme for creating tiled CZML is presented and implemented prototypically, it could be beneficial to be able to convert existing CZML data into the structure needed for the tiling mechanism to work. The streetspace model created from different data sources can be extended and improved by integrating other features such as city furniture or traffic signs using 3D models. So far the generation of CZML data is done after

[5] http://go.tum.de/054180

simulation results are available. In the future it might be possible to couple simulation and visualization at run-time. This might also be relevant when including dynamic real-time data such as sensor information. Based on work presented in this paper, the integration of static as well as time-dependant aspects of urban digital twins within a common representation can be improved.

ACKNOWLEDGEMENTS

We would like to thank the 'GeodataService' of Munich for providing the data used in this publication, Sasan Amini for conducting the SUMO traffic simulation and 'Safe Software' for providing the academic licence of the software FME.

REFERENCES

Amini, S., Tilg, G., Busch, F., 2018. Evaluating the impact of real-time traffic control measures on the resilience of urban road networks. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 519–524.

Artal-Villa, L., Hussein, A., Olaverri-Monreal, C., 2019. Extension of the 3DCoAutoSim to simulate vehicle and pedestrian interaction based on SUMO and unity 3D. *2019 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 885–890.

Behrisch, M., Bieker, L., Erdmann, J., Krajzewicz, D., 2011. SUMO – Simulation of Urban MObility: An Overview. *SIMUL 2011*, ThinkMind.

Beil, C., Ruhdorfer, R., Coduro, T., Kolbe, T. H., 2020. Detailed Streetspace Modelling for Multiple Applications: Discussions on the Proposed CityGML 3.0 Transportation Model. *ISPRS International Journal of Geo-Information*, 9(10), 603.

Chao, Q., Bi, H., Li, W., Mao, T., Wang, Z., Lin, M. C., Deng, Z., 2020. A survey on visual traffic simulation: Models, evaluations, and applications in autonomous driving. *Computer Graphics Forum*, 39(1), 287–308.

Chaturvedi, K., 2021. Integration and Management of Time-dependent Properties with Semantic 3D City Models. Phd thesis, Chair of Geoinformatics, Technical University of Munich.

Chaturvedi, K., Yao, Z., Kolbe, T. H., 2019. Integrated management and visualization of static and dynamic properties of semantic 3D city models. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-4/W17, 7–14.

Chen, W., Guo, F., Wang, F.-Y., 2015. A survey of traffic data visualization. *IEEE Transactions on Intelligent Transportation Systems*, 16(6), 2970–2984.

Dembski, F., Wössner, U., Letzgus, M., Ruddat, M., Yamu, C., 2020. Urban digital twins for smart cities and citizens: The case study of Herrenberg, Germany. *Sustainability*, 12(6), 2307.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V., 2017. Carla: An open urban driving simulator. *Proceedings of the 1st Annual Conference on Robot Learning*, 78, PMLR, 1–16.

Floros, G., Dimopoulou, E., 2016. Investigating the enrichment of a 3D City Model with various CityGML Modules. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, XLII-2/W2, 3–9.

Grieves, M., Vickers, J., 2017. *Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems*. Springer International Publishing, Cham, 85–113.

Hayashi, H., Asahara, A., Kim, K.-S., Shibasaki, R., Ishimaru, N., 2017. OGC Moving Features Access. Version 1.0. `https://portal.ogc.org/files/16-120r3`.

Jaillot, V., Servigne, S., Gesquière, G., 2020. Delivering time-evolving 3D city models for web visualization. *International Journal of Geographical Information Science*, 34(10), 2030–2052.

Keler, A., Kaths, J., Chucholowski, F., Chucholowski, M., Grigoropoulos, G., Spangler, M., Kaths, H., Busch, F., 2018. A bicycle simulator for experiencing microscopic traffic flow simulation in urban environments. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 3020–3023.

Kolbe, T. H., Kutzner, T., Smyth, C. S., Nagel, C., Roensdorf, C., Heazel, C., 2021. OGC City Geography Markup Language (CityGML) Part 1: Conceptual Model Standard. `http://www.opengis.net/doc/IS/CityGML-1/3.0`.

Kurkcu, A., Miranda, F., Ozbay, K., Silva, C. T., 2017. Data visualization tool for monitoring transit operation and performance. *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, IEEE, 598–603.

Kutzner, T., Chaturvedi, K., Kolbe, T. H., 2020. CityGML 3.0: New functions open up new applications. *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 88(1), 43–61.

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., Wießner, E., 2018. Microscopic traffic simulation using SUMO. *2018 21st international conference on intelligent transportation systems (ITSC)*, IEEE, 2575–2582.

Macura, J., 2019. Time-variable Visualization from Sensor Data Inside Building in a 3D GIS Environment. Master's thesis, Department of Geomatics, University of West Bohemia.

Mao, B., Ban, Y., Laumert, B., 2020. Dynamic online 3D visualization framework for real-time energy simulation based on 3D tiles. *ISPRS International Journal of Geo-Information*, 9(3), 166.

Murshed, S. M., Al-Hyari, A. M., Wendel, J., Ansart, L., 2018. Design and implementation of a 4D web application for analytical visualization of smart city applications. *ISPRS International Journal of Geo-Information*, 7(7), 276.

Ruhdorfer, R., Willenborg, B., Sindram, M., 2018. Coupling of Traffic Simulations and semantic 3D City Models. *gis.Science*, 2018(3), 101-109.

Schwab, B., Beil, C., Kolbe, T. H., 2020. Spatio-semantic road space modeling for vehicle–pedestrian simulation to test automated driving systems. *Sustainability*, 12(9), 3799.

Xu, H., Wang, C. R., Berres, A., LaClair, T., Sanyal, J., 2022. Interactive Web Application for Traffic Simulation Data Management and Visualization. *Transportation Research Record*, 2676(1), 274-292.

Yao, Z., 2020. Domain Extendable 3D City Models – Management, Visualization, and Interaction. PhD thesis, Chair of Geoinformatics, Technical University of Munich.

Yao, Z., Nagel, C., Kunde, F., Hudra, G., Willkomm, P., Donaubauer, A., Adolphi, T., Kolbe, T. H., 2018. 3DCityDB – a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML. *Open Geospatial Data, Software and Standards*, 3(1), 1–26.

Zhu, L., Li, Z., Wang, Z., 2018a. Representing time-dynamic geospatial objects on virtual globes using CZML Part II: Impact, comparison, and future developments. *ISPRS International Journal of Geo-Information*, 7(3), 102.

Zhu, L., Wang, Z., Li, Z., 2018b. Representing time-dynamic geospatial objects on virtual globes using CZML Part I: Overview and key issues. *ISPRS International Journal of Geo-Information*, 7(3), 97.