Digital Twins for Topology Density Map Analysis

Zhicheng Hu[1], Agus Hasan[2], Ricardo da Silva Torres[2,3]
[1]NTNU – Norwegian University of Science and Technology, Trondheim, Norway
[2]NTNU – Norwegian University of Science and Technology, Ålesund, Norway
[3]Wageningen University & Research, Wageningen, The Netherlands
{zhicheng.hu, agus.hasan, ricardo.torres}@ntnu.no, ricardo.dasilvatorres@wur.nl

KEYWORDS

Digital Twin, topology density map, spatiotemporal data, urban data visualization.

ABSTRACT

Properly analyzing spatiotemporal patterns is of paramount importance, especially in urban planning. In this paper, we introduce two digital twins to support the analysis of spatiotemporal data associated with urban topologies. In particular, both tools visually encode temporal changes in density maps constrained by a network. Moreover, we present the software architectures and discuss their use in urban planning usage scenarios.

INTRODUCTION

Digital twins have been established as a pivotal technology for supporting decision-making based on the analysis of different what-if scenarios [Shahat et al., 2021]. According to [Verdouw et al., 2021], a digital twin is defined as "*a dynamic representation of a real-life object that mirrors its states and behaviour across its lifecycle and that can be used to monitor, analyze and simulate current and future states of and interventions on these objects, using data integration, artificial intelligence, and machine learning.*" In the context of urban planning, digital twins have been employed successfully to model complex processes, leveraging the possibilities of definition of sustainable solutions for several relevant societal problems, including mobility assessment [Major et al., 2021], [Major et al., 2022], waste management [Nasar et al., 2020], light infrastructure design and implementation [Hassan et al., 2022], environmental pollution [Major et al., 2021], among others.

This paper is concerned with the development of urban digital twins to support the analysis of spatiotemporal changes over time associated with density maps restricted by a network. Density maps (often encoded as heatmaps) have been widely used to analyze the spatial distribution of vector fields [Hogräfer et al., 2020]. Examples of applications include the evaluation of different traffic conditions [Xie and Yan, 2008], pollution distribution [Ren et al., 2020], and demographic evolution [Feng et al., 2020]. Recently, the work by [Feng et al., 2020] introduced a very effective alternative for the computation of density maps constrained by a topology. Their formulation for computing Topology Density Maps (TDMs) encompasses three main steps: encoding of the network data (vertices, and edges and their associated weights), computation of the accessibility information (cost to connect to pre-defined points of interest), and construction of a visualization strategy based on surface mappings. Despite the demonstrated effective results, related to its use in traffic analysis applications, the method of [Feng et al., 2020] relies on only spatial analysis.

Using digital twins in urban planning is especially relevant considering monitoring and prediction analyses. Monitoring digital twins focus on digitally monitoring the state of the behavior of physical objects, while predictive digital twins relate to the projections of future states of such objects [Verdouw et al., 2021]. With this respect, the temporal evolution of density maps plays an important role, especially in supporting the understanding of different what-if simulation results associated with the future states of a city. In this paper, we investigate the visualization of the temporal changes related to density maps encoded in topologies. We introduce two prototypes recently proposed to support the analysis of Temporal Topology Density Maps (TTDMs), discussing their architecture, main features, and usage scenarios.

The remainder of this paper is organized as follows. Next section introduces the main concepts related to topology density maps and our proposal for encoding temporal changes. Next, we present the developed prototypes that implement the different algorithms for topology density computation and its assessment over time. The final section covers our conclusions and presents directions for future research.

BACKGROUND CONCEPTS

This section introduces the background concepts related to the computation of Topology Density Maps and Temporal Topology Density Maps.

Topology Density Maps (TDMs)

Topology Density Maps (TDMs) are defined as maps that encode non-linear scalar fields on a topology (e.g., road network). In a recent publication, the work by [Feng et al., 2020] proposed a three-step approach for computing TDMs. First, a network is employed to encode vertices and edges. The network is seen as a weighted direct graph in which weights

Communications of the ECMS, Volume 37, Issue 1,
Proceedings, ©ECMS Enrico Vicario, Romeo Bandinelli,
Virginia Fani, Michele Mastroianni (Editors) 2023
ISBN: 978-3-937436-80-7/978-3-937436-79-1 (CD) ISSN 2522-2414

represent the costs to move from one vertice to another. The second step refers to the computation of the accessibility of nodes given existing points of interest (POIs). This process is performed by means of the Dijkstra shortest path algorithm. The final step concerns the calculation of the influence zones for the whole 2D space.

Figure 1 provides an example related to the computation of a TDM. The input network data includes six nodes. We assume that $H1$ and $H2$ are POIs and the other four nodes (A, B, C, D) are non-POIs. After the encoding of network data, the graph G will be wrapped up to two directed acyclic graphs (DAGs) for POIs $H1$ and $H2$. The cost F_c is calculated from the POI to non-POI nodes and finally to any point in the 2D planar space. The density field is the complete 2D planar surface for this network. In this figure, the color of the nodes and the tapered edges reflect the accessibility data G_{cost}. The visualization of G_{TDM} concerns one density estimation field with colors that represent the propagation of the density values along the edges and the 2D planar surface. For example, the blue region with the nodes $H1$, A, and B has more variations in the intensities of density fields when compared to the red region with the nodes D, C, and $H2$.

For more details regarding the computation of TDMs, the reader may refer to [Feng et al., 2020].

Temporal Topology Density Map (TTDM)

The algorithm proposed by [Feng et al., 2020] demonstrates great potential in the context of urban mobility analysis. Despite its promising results, the algorithm's efficiency in determining density maps for regions outside the input network is limited. Furthermore, the lack of support for representing changes in topology density maps over time is a significant drawback. In several applications, understanding trends and patterns over time associated with spatial data is a key element in supporting better-informed decision-making.

Temporal Topology Density Map (TTDM) is an optimized TDM algorithm, that supports urban data analysis considering the spatial distribution of the scalar field as well as the temporal variation. It combines the representation power of Change Frequency Heatmap (CFH) [Mariano et al., 2017] and the efficiency of Image-Foresting Transform (IFT) [Falcao et al., 2004].

Figure 2 provides one running example with four timestamps temporal changes (the size of the timestamps $T = 4$) on the introduced TDM example. This stack of graphs, including temporal changes of edge costs, is the input for the steps that compute a representative TDM and encode temporal changes. After the computation, it produces three outputs (the visualized 1D network, a 2D planar texture color map, and a 3D mesh with a height map) for the final visualization.

More details regarding the computation of TTDM can be found in the work by [Hu, 2022].

PROTOTYPES

Two prototypes were designed and implemented. The first one, a desktop software prototype, aims to support the performance and qualitative assessment of the TDM and TTDM. This prototype allows algorithm computation analysis according to different parameter settings. The second one, a web-based software prototype, aims to support the analyses of diverse visual layouts associated with different case studies.

This section overviews the main technologies employed in the implementation of the proposed algorithms, as well as the prototypes created.

Implementation Aspects

Figure 3 illustrates the main technologies employed in the implementation of these two software prototypes. The Python programming language and the OSMnx package [Boeing, 2017][1] were utilized to download and export geospatial data from OpenStreetMap.[2] The python module is responsible for encoding the target network obtained from OpenStreetMap (Label 1) to a comma-separated values (CSV) file. The alternative supporting format (Label 2) of network data is GeoJSON,[3] which is a geospatial data interchange format based on JavaScript Object Notation (JSON). The free Open Source software QGIS[4] and the commercial software ArcGIS (Pro, Online)[5] are the main popular tools to create and edit GeoJSON files. Similarly, there are also two additional formats (CSV and JSON) used for loading weather data from two Norwegian providers: Norwegian Climate Service Center[6] (3 in the figure) and Meteorologisk Institutt – Frost Application Programming Interface (API)[7] (4).

The TTDM computation analysis tool (Desktop) (Label 5) is a software prototype that contains an implementation of the TTDM algorithm in Unity[8] with C# script and Mapbox Software Development Kit (SDK).[9] The input graph network is encoded into two file formats (CSV, GeoJSON). This prototype computes edge costs based on weather data recorded in CSV and JSON. The prototype also integrates a TTDM Dynamic Link Library (DLL) implemented based on the IFT C source code package.[10] The software provides an approach to execute the TTDM

[1]`https://github.com/gboeing/osmnx` (As of Mar. 2023).
[2]`https://www.openstreetmap.org/` (As of Mar. 2023).
[3]`https://geojson.org/` (As of Mar. 2023).
[4]`https://qgis.org/en/site/` (As of Mar. 2023).
[5]`https://www.arcgis.com/` (As of Mar. 2023).
[6]`https://seklima.met.no/` (As of Mar. 2023).
[7]`https://frost.met.no/` (As of Mar. 2023).
[8]`https://unity.com/` (As of Mar. 2023).
[9]`https://www.mapbox.com/unity` (As of Mar. 2023)
[10]`https://github.com/tvspina/ift-demo` (As of Mar. 2023).

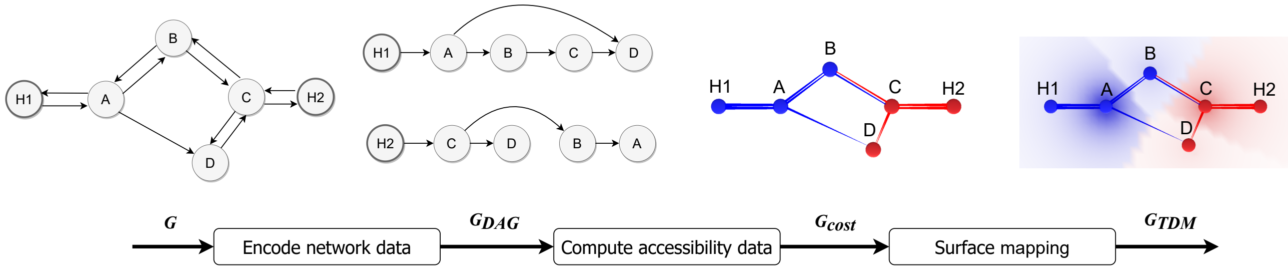


Fig. 1: A running example for Topology Density Map. The network data G includes two POI nodes ($H1, H2$) and four non-POI nodes (A, B, C, D). For each POI node, G_{DAG} is one DAG connecting non-POI nodes from it. G_{cost} is the accessibility data, which includes the path cost and related POI label. It can be directly used for the G_{TDM} for the final visualization.

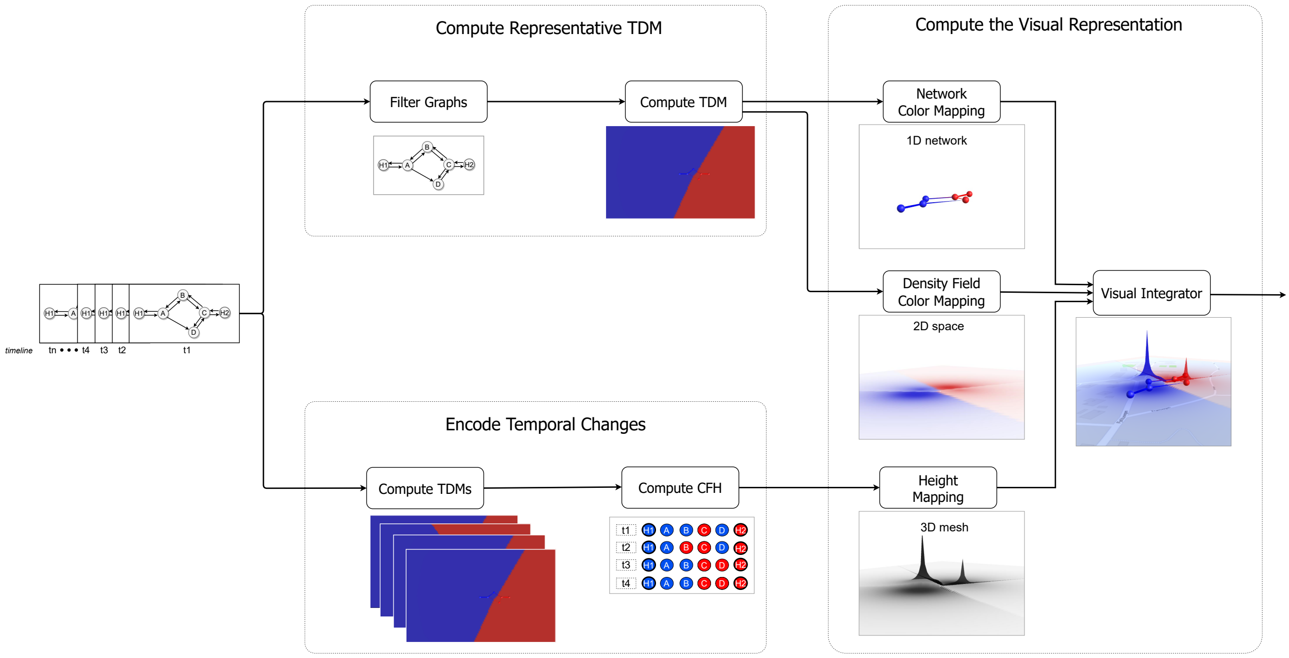


Fig. 2: A running example for Temporal Topology Density Map.

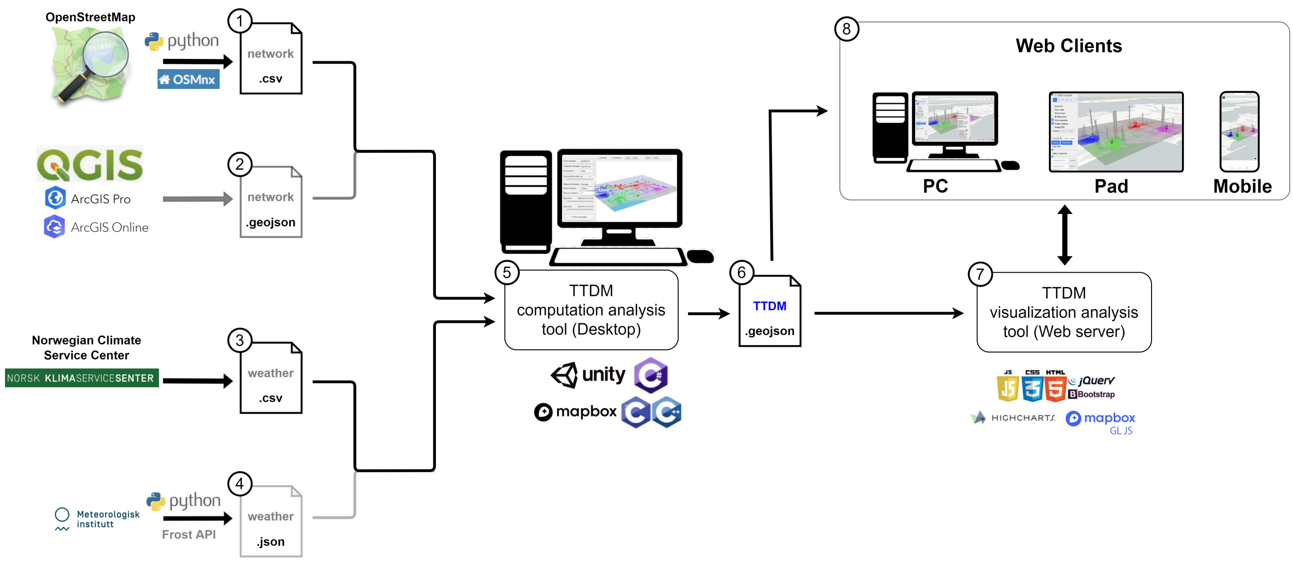


Fig. 3: Overview of the different technologies used in the implementation of the prototypes.

algorithm on selected datasets with customized parameters. It supports 3D visualization as well as saving the algorithm computation result as a GeoJSON format file (Label 6).

TTDM visualization analysis tool (Web server) (Label 7) is a software prototype to visualize the TTDM computation results on the web. It is mainly implemented using Javascript, Mapbox GLJS library,[11] and Bootstrap.[12] This prototype supports the assessment of generated visual structures with a more user-friendly user interface (UI). Different kinds of web clients (Label 8) are expected to access this web server. Those clients allow users to upload a TTDM GeoJSON file (Label 6) and compare associated visual results.

Overview of Prototypes

This section describes the main features of the developed prototypes.

TTDM Computation Analysis

The software prototype TTDM computation analysis tool (Desktop) is designed to support computation analysis. Its main functionalities are:

- Integration of a TTDM C source code package in the TTDM computation.
- Execution of algorithms to support performance and qualitative assessment and download of results.
- Execution of the TTDM algorithm on a selected graph dataset with different parameters and visualization of results in a 3D view.
- Saving of TTDM computation results as a GeoJSON file for visualization assessment.

Figure 4 provides an overview of the implementation components. The external resources include files or interfaces needed for the implementation. The functions are programmed in C# script codes, and visual structures are created by means of visual objects in Unity. Mapbox API (Label 1) is called by the Mapbox SDK (Label 4) to construct a Mapbox street map Layer (Label 9). The network and weather data files (Label 2) are the data source for the core function in the TTDM computation (Label 5). This algorithm needs to call the IFT algorithm available in the created TTDM DLL file. The TTDM computation module uses the configuration defined in the user interface (Label 11) to update the TTDM Visual Layers (Label 10). Other features refer to exporting the TTDM .geojson file (Label 6) and performance assessment (Label 7). Finally, label and density maps created during the computation can be exported (Label 8).

Figure 5 presents a screenshot of the User Interface of the TTDM computation analysis tool (Desktop). On the left region, there is a menu composed of five panels (Labels 1-5). This menu allows the

[11]https://www.mapbox.com/mapbox-gljs (As of Mar. 2023).
[12]https://getbootstrap.com/ (As of Mar. 2023).

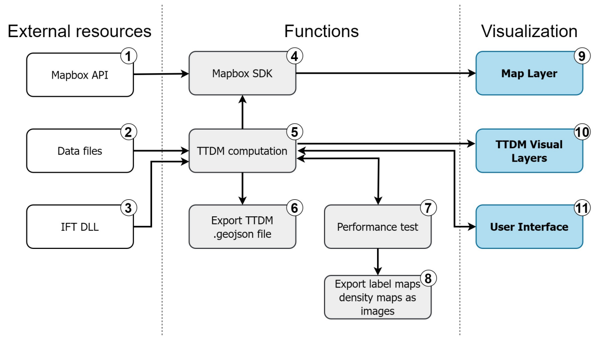


Fig. 4: The implementation architecture of the TTDM computation analysis tool (Desktop).

definition of the configuration of the parameters before the TTDM computation. The users may select the graph dataset, the method for encoding temporal changes, the weather dataset, and the daily data filter (e.g., "all days," "weekdays," and "weekends") in the first Panel (Label 1). There are three choices in the drop-down menu for the temporal change encoding methods. Available options include importing predefined temporal changes, creating random temporal changes, and computing the simulated temporal changes based on real weather data (e.g., in the current version, snow data). The second one (Label 2) provides three choices to estimate the density field on 2D planar surface TTDM with running Topology Density Maps (TDM), IFT-based Topology Density Map (ITDM) option 1, and option 2. It is also possible to use a slider to select any timestamp ITDM as a representative. Another available option refers to the definition of the representative based on an average function on the third panel (Label 3). The fourth panel (Label 4) includes the parameter configuration relating to the computation of a change frequency heatmap. It allows the users to choose the density or label maps as the input of the CFH algorithm, the change binary pattern, and the time range (by the sliders). The TTDM computation (Label 5) will be started after the choice of the interpolation scale. The top-right menu (Label 6) includes the display control of the visualization components: loading and saving of the TTDM computation result, saving density maps and label maps as figures, and executing the performance test. In the center region (Label 7), there is an area to display the 3D visualized results with the fly control camera by the mouse. At the bottom (Label 8), it shows some hotkey information and one help button.

TTDM Visualization Analysis

The software prototype TTDM visualization analysis tool (Web server) is designed for visualization analysis. Its main functionalities are::

- Decoding of the TTDM computation result (GeoJSON file) for the visualization analysis.
- Filtering the temporal changes with customized

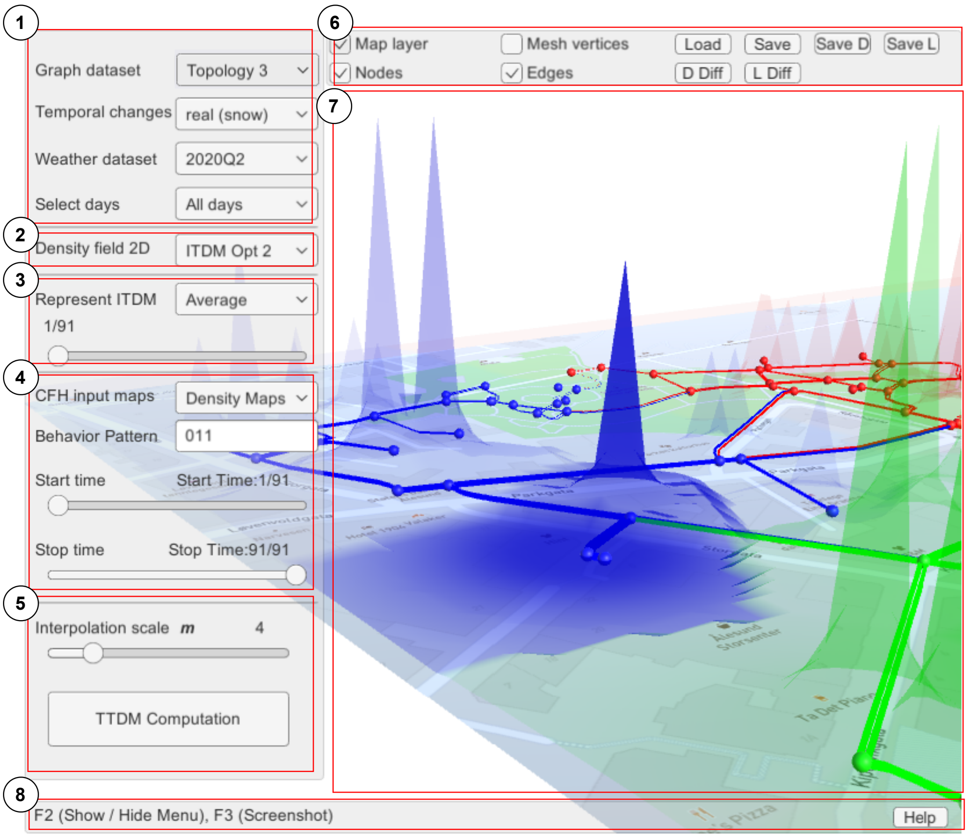


Fig. 5: Screenshot of the TTDM computation analysis tool (Desktop) created for TTDM computation analysis.

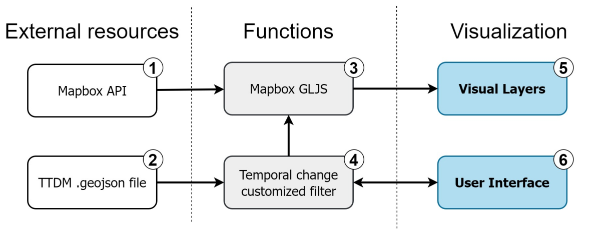


Fig. 6: The implementation architecture of the TTDM visualization analysis tool (Web server).

parameters.
• Visualizing all data in the geographic coordinates system.
• Providing a more user-friendly interface with some features, such as the object track function by a mouse hover and click, the playback function of the ITDM for each timestamp, the camera positions synchronization, etc.

Figure 6 shows an overview of its implementation. Mapbox API (Label 1) is accessed by the Mapbox GLJS (Label 3) to create all visual layers (Label 5). The TTDM GeoJSON file (Label 2) is uploaded for the temporal change customized filter (Label 4). This filter implements the main functions based on the settings defined in the user interface (Label 6) and exports the data to visual layers by Mapbox GLJS.

Figure 7 presents a screenshot of the user interface of the TTDM visualization analysis tool (Web server) accessed by a web browser. There is one toggle menu (Label 1) on the top left. It includes two main tab pages (Label 2) “Load” and “Layers.” The “Load” page (Label 3a) allows the users to choose a remote TTDM GeoJSON file on the list directly or one local file to upload before the visualization analysis. It also supports the playback function with the sliders for the selected timestamp data and the waiting time for a load of each timestamp. Different options are available for encoding temporal changes, such as the selection of CFH input data (density map or label map), the optional metric functions (change, increase, decrease), the threshold for defining binary maps, and the change binary pattern. The “Layers” page (Label 3b) provides more options for advanced visualization analysis. It includes layers to control the different map layer views provided by Mapbox GLJS, components of the visual integrator.In addition, some features like the camera position synchronization by clicking copy and paste button and the transparency alpha adjustment are also provided on this page. The information on the object information panel (Label 4) is updated when the mouse moves on the map in the center main region (Label 5). This center main region displays the final visualization result on the map, supporting pan and zoom operations with the mouse. The object screenshot panel

(Label 6) is activated after the left-click of the mouse. This panel makes it easier to compare the data of one marked position with another one shown in the object information panel (Label 4). The snow depth data may also be shown with various chart styles if the user activates the option "Graph XY (snow depth)" on the "Layers" page (Label 3b).

Figure 8 illustrates the visual effects of one example with the default configuration. For instance, the map style options include light (L), satellite (S), dark (D), street (E), and outdoor (O). All of them are provided by Mapbox GLJS. The default configuration is the light map style. It also supports the display of the terrain and building data as additional features with selected map styles. Next, three data layers (POIs layer, intersections layer, and roads layer) will responsively show points-of-interest nodes, intersections (nodes), and roads (edges). Here, we used labels in the interface of the mobility applications instead of the algorithm to support future user studies with domain experts. At last, the three main modules (network color mapping, density color mapping, and height mapping) of the visual integrator can be configured. Also, it is possible to choose alternatives to display the CFH result using height information. The center region of the figure contains the visual result related to the integration of these selected multiple visual layers.

Figure 9 is one example of the object track function. It tracks and displays the object hovered by the mouse. The priority from high to low is node, edge, and point. In TTDM GeoJSON file, all information is only saved in the geographic coordinates. TTDM visualization analysis tool (Web server) is a prototype of visualizing and analyzing the TTDM computation result using GeoJSON format. It means it is also feasible to visualize the TTDM computation result by the programming script in any software that supports GeoJSON.

For most usage scenarios, the users are more interested in distinguishing the temporal changes in 3D views. Figure 10 provides five typical configurations for the usage. Here, we used the extrusion of polygon[13](Figure 10e) to create the 3D bars array for demonstration. It is easiest to recognize the elevations accurately in our designed aspect.

We also explored the use of the proposed prototypes in a visualization lab at NTNU Ålesund with multiple projectors [14] as illustrated in Figure 11. It has the capability to support several clients running concurrently while simultaneously providing diverse analytical perspectives. Moreover, it offers multiple camera views of the same scenario, which can be configured differently to facilitate improved analysis. [15]

[13]`https://docs.mapbox.com/mapbox-gl-js/example/3d-extrusion-floorplan/` (As of Mar 2023).

[14]`https://www.ntnu.edu/iir/department-of-ict-and-natural-sciences` (As of Mar. 2023).

[15]More related videos and materials can be downloaded at `https://drive.google.com/drive/folders/1BUkvNtwth4Iv7QRjfBmBgGG2v8xMA3MT?usp=sharing/` (As of Mar. 2023).

CONCLUSIONS

In this paper, we have introduced two prototypes to support the assessment of Topology Density Maps and their changes, referred to as Temporal Topology Density Maps (TTDMs). The prototypes have the potential to be employed by citizens and professionals (e.g., urban planners, politicians) to analyse the temporal features of urban data with pre-defined data patterns, such as those encoded into binary strings.

The developed algorithms were embedded in two software prototypes. One prototype focuses on the Temporal Topology Density Map computation analysis, including the support for the assessment of different configuration settings. The other addresses the visualization analysis itself. Both of them served as tools for the conducted validation, involving performance and qualitative assessments. The source code of both prototypes can be found here[16,17] The first prototype focuses on the customization and analysis of the impact of different parameters of the algorithms. The goal, therefore, is to support the identification of the best configurations for a particular usage scenario. The second prototype, in turn, supports the assessment of generated visual structures by target users.

Future work will focus on integrating the TDM and TTDM visual structures into the NORDARK-DT, a digital twin has been developed to support lighting infrastructure planning for green urban areas.[18] We also plan to conduct user studies with relevant stakeholders who are potentially interested in identifying patterns and trends related to changes in density maps that are associated with decision-making processes in urban planning.

ACKNOWLEDGEMENTS

This work has been conducted in the context of the NORDARK project, funded by NordForsk. This work was also partially funded by the NFR SMART-PLAN (310056) and Twin Fjord (320627) projects.

References

[Boeing, 2017] Boeing, G. (2017). Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65:126–139.

[Falcao et al., 2004] Falcao, A. X., Stolfi, J., and de Alencar Lotufo, R. (2004). The image foresting transform: Theory, algorithms, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 26(1):19–29.

[Feng et al., 2020] Feng, Z., Li, H., Zeng, W., Yang, S.-H., and Qu, H. (2020). Topology density map for urban data visualization and analysis. *IEEE transactions on visualization and computer graphics*, 27(2):828–838.

[16]`https://github.com/felando1984/WebTTDM` (As of Mar. 2023).

[17]`https://github.com/felando1984/TTDM` (As of Mar. 2023).

[18]`https://nordark.org/` (As of Mar. 2023).

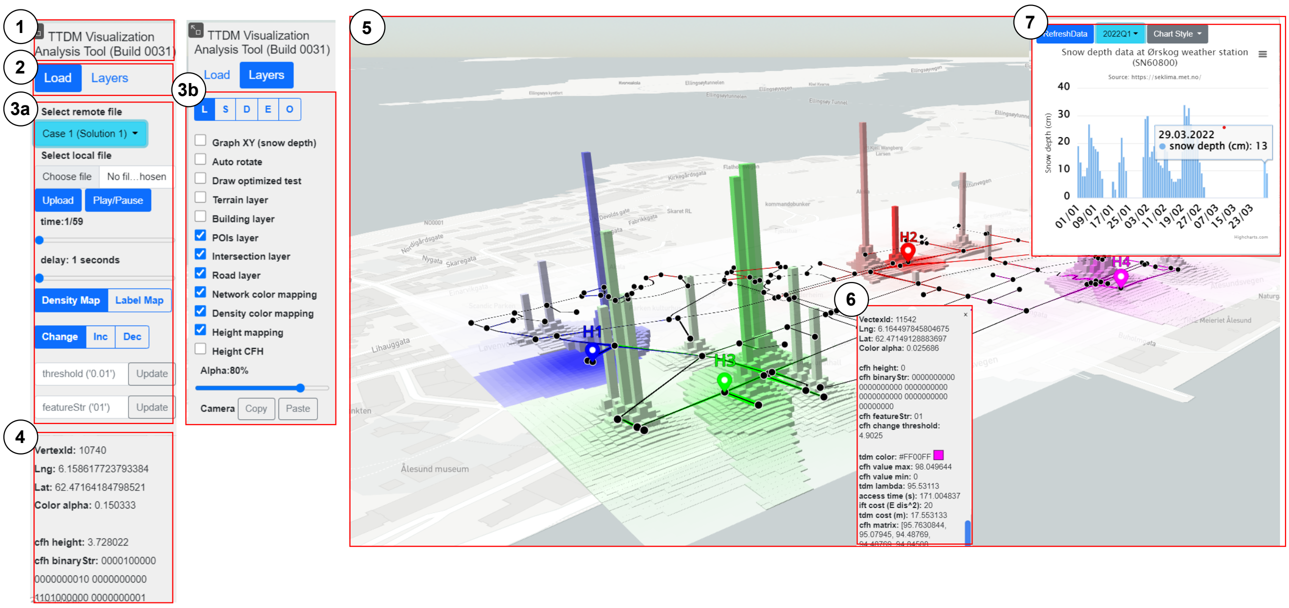


Fig. 7: Screenshot of the TTDM visualization analysis tool (Web server) created for TTDM visualization analysis.

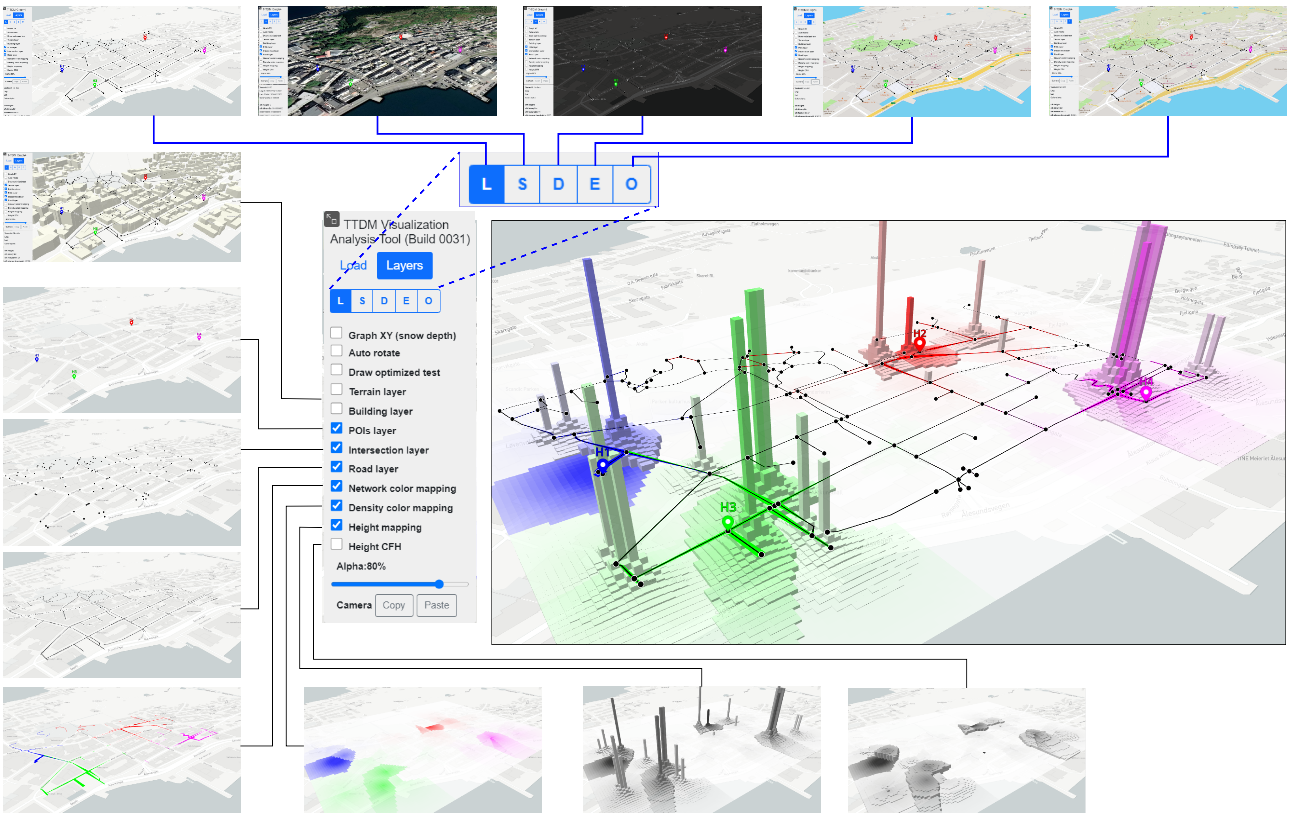


Fig. 8: Visual effects of the default example after the initialization.

[Hassan et al., 2022] Hassan, M. U., Angelaki, S., Alfaro, C. V. L., Major, P., Styve, A., Alaliyat, S. A.-A., Hameed, I. A., Besenecker, U., and da Silva Torres, R. (2022). Digital twins for lighting analysis: Literature review, challenges, and research opportunities. In *Proceedings of the 36th ECMS International Conference on Modelling and Simulation, ECMS 2022, Ålesund, Norway, May 30 - June 3, 2022*, pages 226–235. European Council for Modeling and Simulation.

[Hogräfer et al., 2020] Hogräfer, M., Heitzler, M., and Schulz, H.-J. (2020). The state of the art in map-like visualization. *Computer Graphics Forum*, 39(3):647–674.

[Hu, 2022] Hu, Z. (2022). Encoding and visualizing temporal changes of topology density maps. Master's thesis, Norwegian University of Science and Technology (NTNU).

[Major et al., 2022] Major, P., da Silva Torres, R., Amundsen, A., Stadsnes, P., and Mikalsen, E. T. (2022). On the use of graphical digital twins for urban planning of

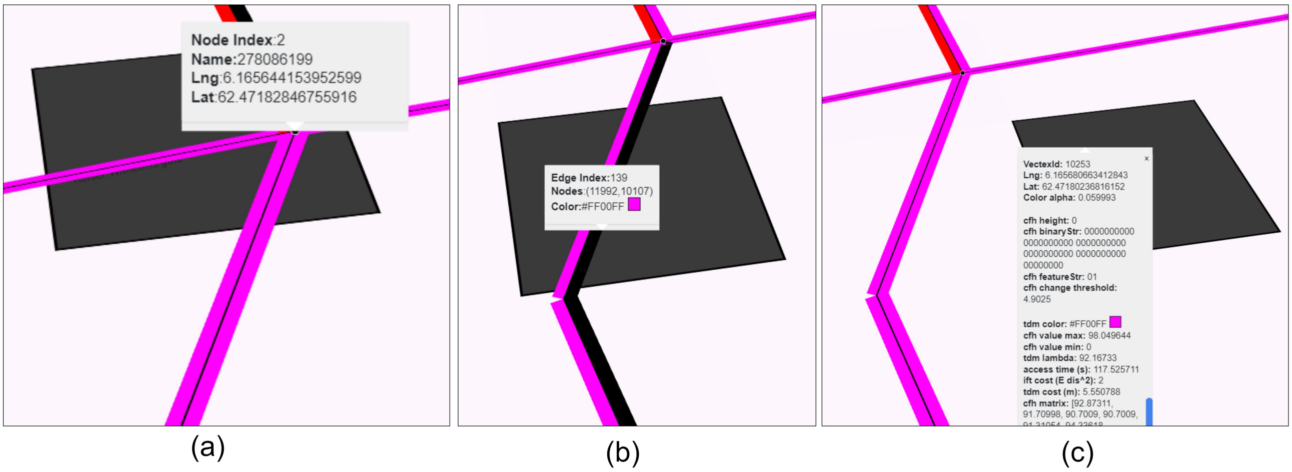
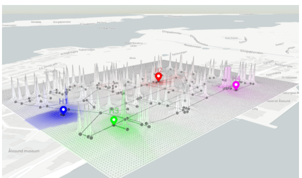


Fig. 9: Visual effects of the object track function for node (a), edge (b), and point (c).

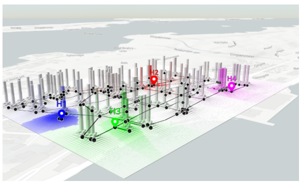
(a) Colored mesh without Geoinformation

(b) Texture mesh with Geoinformation

(c) Texture mesh with colored wireframe

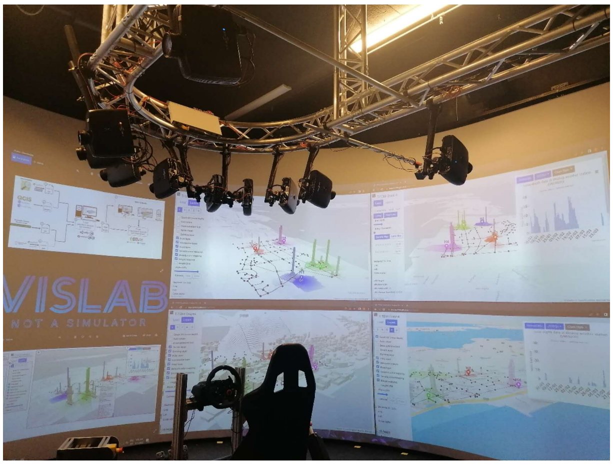


(d) Texture mesh with grey wireframe

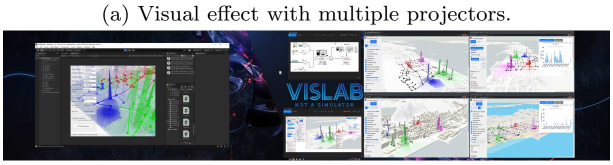


(e) 3D bar

Fig. 10: Visual effects of the 3D visualization.



(a) Visual effect with multiple projectors.



(b) Screenshot with two digital twins: Computation Analysis (left) and Visualization Analysis (right).

Fig. 11: Digital twins running in the visualization lab at NTNU, Ålesund.

mobility projects: A case study from A new district in åesund, Norway. In *Proceedings of the 36th ECMS International Conference on Modelling and Simulation, ECMS 2022, Ålesund, Norway, May 30 - June 3, 2022*, pages 236–242. European Council for Modeling and Simulation.

[Major et al., 2021] Major, P., Li, G., Hildre, H. P., and Zhang, H. (2021). The use of a data-driven digital twin of a smart city: A case study of Ålesund, norway. *IEEE Instrumentation Measurement Magazine*, 24(7):39–49.

[Mariano et al., 2017] Mariano, G., Soares, N. C., Morellato, L. P., and da Silva Torres, R. (2017). Change frequency heatmaps for temporal multivariate phenological data analysis. In *IEEE 13th International Conference on e-Science*, pages 305–314.

[Nasar et al., 2020] Nasar, W., Karlsen, A. T., and Hameed, I. A. (2020). A conceptual model of an iot-based smart and sustainable solid waste management system: A case study of A norwegian municipality. In *Proceedings of the 34th International ECMS Conference on Modelling and Simulation, ECMS 2020, Wildau, Germany, June 9-12, 2020*, pages 19–24.

[Ren et al., 2020] Ren, K., Wu, Y., Zhang, H., Fu, J., Qu, D., and Lin, X. (2020). Visual analytics of air pollution propagation through dynamic network analysis. *IEEE Access*, 8:205289–205306.

[Shahat et al., 2021] Shahat, E., Hyun, C. T., and Yeom, C. (2021). City digital twin potentials: A review and research agenda. *Sustainability*, 13(6).

[Verdouw et al., 2021] Verdouw, C., Tekinerdogan, B., Beulens, A., and Wolfert, S. (2021). Digital twins in smart farming. *Agricultural Systems*, 189:103046.

[Xie and Yan, 2008] Xie, Z. and Yan, J. (2008). Kernel density estimation of traffic accidents in a network space. *Computers, environment and urban systems*, 32(5):396–406.