

The Urban Toolkit: A Grammar-based Framework for Urban Visual Analytics

Gustavo Moreira, Maryam Hosseini, Md Nafiul Alam Nipu,
Marcos Lage, Nivan Ferreira, and Fabio Miranda

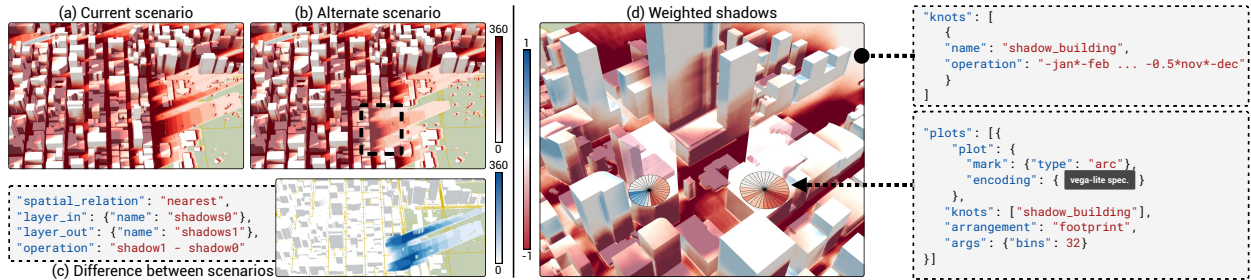


Fig. 1: The Urban Toolkit enables the creation of urban visualizations through a new grammar that uses *knots* to integrate thematic (e.g., sunlight access, demographic) and physical (e.g., buildings, street networks) data layers. Left: An example of *what-if* scenario planning for urban shadow analysis. The difference between current (a) and proposed (b) urban development scenarios can be specified through algebraic operations using the grammar (c). Right: Sunlight access analysis at the building level. Chained operations are specified using the grammar, with plots embedded in the 3D environment (d).

Abstract— While cities around the world are looking for smart ways to use new advances in data collection, management, and analysis to address their problems, the complex nature of urban issues and the overwhelming amount of available data have posed significant challenges in translating these efforts into actionable insights. In the past few years, urban visual analytics tools have significantly helped tackle these challenges. When analyzing a feature of interest, an urban expert must transform, integrate, and visualize different thematic (e.g., sunlight access, demographic) and physical (e.g., buildings, street networks) data layers, oftentimes across multiple spatial and temporal scales. However, integrating and analyzing these layers require expertise in different fields, increasing development time and effort. This makes the entire visual data exploration and system implementation difficult for programmers and also sets a high entry barrier for urban experts outside of computer science. With this in mind, in this paper, we present the Urban Toolkit (UTK), a flexible and extensible visualization framework that enables the easy authoring of web-based visualizations through a new high-level grammar specifically built with common urban use cases in mind. In order to facilitate the integration and visualization of different urban data, we also propose the concept of *knots* to merge thematic and physical urban layers. We evaluate our approach through use cases and a series of interviews with experts and practitioners from different domains, including urban accessibility, urban planning, architecture, and climate science. UTK is available at urbantk.org.

Index Terms—Urban visual analytics, Urban analytics, Urban data, Visualization toolkit.

1 INTRODUCTION

Cities around the world are looking for smart ways to make use of data to address their problems. By analyzing data from cities, urban scientists, practitioners, policymakers, and communities can gain a deeper understanding of complex urban problems, enabling better planning, policies, and urban resilience, making government more efficient and, ultimately, improving the lives of residents. But the complex nature of urban issues and the overwhelming amount of data have posed significant challenges in translating these efforts into actionable insights. Drawing a comprehensive picture often involves collaboration across

multiple disciplines, including computer science, engineering, climate sciences, architecture, urban planning, and public health, leading to a variety of analytical workflows and tasks to transform, integrate, and analyze different urban data layers, often across multiple spatial and temporal scales [25]. Urban visual analytics is considered a key component in enabling such analyses. It has been used to augment data mining solutions and leverage domain knowledge [7, 16, 27, 28], as well as public dissemination of findings and policies [30, 44, 45]. While there have been many successful urban visual analytics tools, tackling a variety of urban problems [24, 71], the area still suffers from a fragmented community and disconnected efforts [69]. Projects are often siloed and developed based on one-off collaborations between computer scientists and urban experts to tackle specific problems. In the end, new solutions are often built from scratch, leading to wasted efforts and results that are difficult to reuse and extend to different scenarios and geographical locations. Furthermore, creating visual analytics prototypes and tools is still challenging and time-consuming, requiring expertise in visualization, computer graphics, and data management. This is especially true in the increasingly important case of 3D urban analytics [47], in which the design and implementation of visualization systems are substantially more complex. This results in a considerable effort even for experienced visualization developers, and also in high entry barriers for urban experts outside of computer science and communities lacking the necessary resources.

- Gustavo Moreira, Md Nafiul Alam Nipu, and Fabio Miranda are with the University of Illinois Chicago.
E-mails: {gmorei3,mnipu2,fbiom}@uic.edu.
- Maryam Hosseini is with the Massachusetts Institute of Technology.
E-mail: maryamh@mit.edu.
- Marcos Lage is with the Universidade Federal Fluminense.
E-mail: mlage@ic.uff.br.
- Nivan Ferreira is with the Universidade Federal de Pernambuco.
E-mail: nivan@cin.ufpe.br.

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org.
Digital Object Identifier: xx.xxx/TVCG.201x.xxxxxx

The visualization community has made significant strides in facilitating the authoring and prototyping of interactive visualization interfaces through toolkits and high-level grammars. While these contributions have significantly lowered the entry barrier for information [11, 12, 56, 57, 74], scientific [38, 43, 52], and immersive visualization [13, 60], the same cannot be said about urban visual analytics. Urban data and urban analytical tasks impose specific requirements that must be met to drive real-world applications. To democratize urban visual analytics, we need new capabilities that empower a broad range of stakeholders to analyze urban data and foster new reproducible, flexible, and extensible tools.

As a step towards this direction, in this paper, we propose the Urban Toolkit (UTK), an open-source visualization toolkit for the easy authoring and prototyping of urban visual analytics applications. Considering such applications and tools, UTK proposes to enable the following: (1) extensibility so that new functionalities by visualization researchers can be incorporated, (2) reproducibility, ease of use, and deployment by urban experts and communities, and (3) flexibility so that outcomes can be translated into different urban domains and geographical locations. Our goal is to minimize the necessity of being familiar with low-level concepts (e.g., rendering) or coding (e.g., C++ or JavaScript) to enable prototyping, evaluation of visualization designs, and the creation of compelling and useful urban visual analytics tools, even for 3D analysis scenarios. To achieve this, at the core of UTK is a high-level visualization grammar for creating and sharing map and plot-based interactive urban visualizations. The grammar uses *knots* (grammar-defined links, or ties, between spatial layers) to easily integrate 2D and 3D data layers across multiple spatial resolutions. UTK also offers a suite of functionalities to facilitate the ingestion and parsing of data from different sources, supporting a broad range of use cases. A tight connection between the data handling and visualization capabilities enables the easy integration of UTK into existing urban workflows and the quick prototyping of visualization interfaces. UTK is publicly available at urbantk.org.

These functionalities can be seen from at least three different angles: first and foremost, as a unified framework aimed at urban scientists and experts that abstracts many low-level data management and visualization aspects involved in implementing urban visual analytics applications. Second, as a testbed for visualization researchers and practitioners interested in the implementation of new applications as well as the prototyping and user evaluation of design choices and techniques for urban visualization. And third, as an easy-to-use authoring tool for stakeholders and communities that enables the creation and sharing of interactive urban visualizations. To highlight the usefulness of our proposal across diverse domains and workflows, we present a set of use cases that implement complex analysis scenarios using UTK. We also report on a series of interviews with urban experts from urban planning, architecture, and climate sciences. The work reported in this paper can be summarized as follows:

- We introduce a high-level visualization grammar to facilitate the development and sharing of urban visual analytics applications.
- We conceptualize a flexible way to specify data integration across multiple spatial resolutions.
- We present the Urban Toolkit, a toolkit that facilitates the design, authoring, and deployment of urban visual analytics applications.
- We report on a set of use cases inspired by previous works demonstrating UTK’s ease of use, flexibility, and reproducibility.
- We present the feedback obtained from domain experts, reporting their perspectives on UTK.

2 BACKGROUND

Urban environments are complex ecosystems where various agents, events, and infrastructures dynamically and continuously interact. That, coupled with the fast-paced rhythm of life, creates an intricate net of relationships, correlations, events, and phenomena that can be sensed, simulated, or modeled, generating a large-scale stream of data. Following previous works [17, 47], in this paper, we categorize the urban data analysis targets into *physical* and *thematic layers*. Physical layers represent the city’s physical aspects (e.g., buildings, road networks,

park, water bodies) as geometries. Thematic layers store the urban data from simulations, machine learning models, sensing initiatives, or surveys. Figure 2 shows a thematic layer with sunlight access data integrated with four different types of physical layers. Multivariate datasets might require the use of abstract visualizations [39].

Traditionally, the decision-making process has relied on tools that mainly utilize certain urban datasets defined over 2D maps, using a flat city metaphor to represent the city environment [71]. More recently, however, several works have explored alternatives that integrate the built and the natural environment into the analysis process [47]. The reason is that a large portion of urban data either relates to natural phenomena (e.g., wind [9], noise [63]) or must be analyzed concurrently with physical layers (e.g., road topology [36], buildings [53], sidewalks [31]). If this physical context is not considered, relevant insights can be lost [10, 25]. For example, civil engineers analyzing landslides or floods must integrate data from simulations and data representing buildings, mountains, and street networks to assess the impact of new disasters [19]. Urban planners, architects and community boards analyzing the impact of new buildings on surrounding ones must integrate view, sky exposure, and building data to inform design and neighborhood character [27]. These analyses are often performed by teams with complementary objectives. In summary, at the core of a growing number of urban analysis workflows is the fundamental integration between physical (including 3D) and thematic data layers over multiple spatial scales.

Designing and implementing interactive visualization systems for exploring multiple layers of data poses several challenges. First, the design process of an urban visual analytics application is naturally an iterative one [58], but there have been few studies that guide the best practices in the field [47, 48]. As a consequence, multiple alternative designs have to be tried out before the final one is chosen. Second, implementing visualizations and interactions, especially 3D ones, is not trivial. In fact, even for developers with experience in computer graphics technologies, a substantial effort is often necessary to iterate over different design choices. Finally, while general GIS systems provide analytical capabilities alongside the possibility of defining dashboards, they are not extensible and have a long learning curve. Therefore, a visual analytics framework that can support the exploration of urban data by different stakeholders can greatly benefit from a highly flexible approach. Such a framework can allow the stakeholders to guide the analysis in any direction required without being constrained by the limitations of the system. That is especially relevant with urban data due to its intricacies and complex relationships. Moreover, it can serve as a test bed for evaluation studies, facilitating comparison among different visualization designs, integration schemes, and interactions. Our overall goal in the present paper is to propose such a framework.

3 RELATED WORK

In this section, we review prior work in two areas. First, we review prior work on urban visual analytics tools. Then, we review prior work on grammars, toolkits, and authoring tools to facilitate the creation of new visualizations.

3.1 Urban visual analytics

Visual analytics systems are important tools for urban data analysis [24]. However, these tools are often tailored to particular domains and designed to analyze specific data layers independently. For example, previous works focused on transportation and mobility [28], planning [46], environmental pollution [23], and public safety [29]. More closely related to our work are tools that enable the integration and exploration of multiple urban data layers. Chang et al. [15] proposed a system that juxtaposes a 3D view of the city with an information view that displays multidimensional urban data. Ferreira et al. [27] proposed a visual analytics tool that integrates physical and thematic data layers for urban development analysis. More recently, Ortner et al. [49] and Miranda et al. [45] proposed systems that integrate 3D spatial and non-spatial views for view impact and sunlight access analysis, respectively. Zeng and Ye [70] proposed a system that integrates physical entities and design

metrics to study urban vitality. Sun et al. [62] proposed to embed visualizations into the physical layer, while Speckmann and Verbeek [61] and Angelini et al. [8] proposed map distortion techniques, with the latter inspired by manipulations of volumetric data [21]. The integration of layers is also explored in tools for disaster management [20, 65].

While these tools have shown valuable contributions in tackling various urban problems, the vast majority is not publicly available. This is a serious limiting factor for urban experts interested in using these tools across different domains or regions. Moreover, even if they were available, many rely on complex datasets that require laborious and time-consuming steps to parse and clean. Popular commercial and open-source GIS tools, while providing spatial analysis capabilities and abstracting many of these challenges, do so to the detriment of flexibility. They also offer limited support for the integration of spatial and abstract visualization components, as well as the integration of multiple physical and thematic layers.

UTK shares many similarities with previous efforts in urban visual analytics. It offers a unified method to integrate diverse data layers and supports multiscale analysis, as well as spatial (including 3D) and abstract visualization components. By adopting a declarative visualization grammar and streamlining data integration computations, we hope to facilitate the use and deployment of applications by urban stakeholders. We also hope to offer flexibility to visualization researchers so that they can extend and adapt the system as needed while ensuring reproducibility of outcomes to enable sharing of results across different domains and regions.

3.2 Grammars and authoring tools for visualization

Urban visual analytics tools are systems specifically built to gain insight into datasets and facilitate domain use cases. Because of their complexity and design process, these tools are usually difficult to adapt to other scenarios, datasets, regions, or scales. To facilitate deployment, flexibility, and reproducibility, general authoring tools have also been adopted by urban experts, enabling the creation of visualization interfaces, sometimes even removing the requirement to know how to program. These works can be divided into three categories: template-based tools that expose their functionalities through GUIs, low-level libraries requiring programming knowledge, and high-level visualization grammars. Given the ubiquity of spatial data, many of these tools support the creation of maps-based visualizations.

In the first category, we have general template-based tools that follow a drag-and-drop metaphor for authoring visualization interfaces [5, 6, 68]. While enabling the easy creation of visualization interfaces, they are too general to support the complex data and analytical tasks of real-world urban applications [71]. Their mapping capabilities are also restricted to 2D spatial regions. In the second category, we have low-level visualization libraries [1, 3, 12]. These libraries offer some facilities for the creation of well-known charts, as well as the freedom to create new visualization designs. `deck.gl` [2], for example, offers functionalities to create layer-based spatial visualizations. However, it lacks support to integrate, interact and link multiple layers and visualizations.

In the third category, we have high-level visualization grammars [37, 42, 51, 56, 59, 60, 74], a compromise between the ease of use of template-based tools and the flexibility of visualization libraries. Rather than be constrained by templates or requiring the programming of individual visualization components, visualization grammars empower users to specify their visualizations through high-level abstractions. In doing so, visualization specifications and system components are clearly separated. Given their ease of use and flexibility, these grammars allow for quickly iterating over different visualization designs. For example, with Vega [57], Vega-Lite [56], and Animated Vega-Lite [74], users can author their own visualizations through JSON files following rules that specify marks, encodings, and interactions of the plots. DXR [60], VRIA [13], and Deimos [40] extend Vega-Lite's grammar to virtual and augmented reality, offering the ability to create immersive visualizations within arbitrary (including urban) environments, but without offering complex integration and interaction between different data layers.

Our work is inspired by and extends the aforementioned high-level visualization grammars. UTK, beyond offering a unified and streamlined framework to load and parse urban layers, also provides a conceptual model to integrate physical and thematic layers across multiple resolutions. This model is made available to the user through a declarative grammar, enabling data transformation while abstracting low-level implementation details.

4 UTK DESIGN GOALS

In this section, we synthesize the design goals that guided the development of UTK. These principles were motivated by our previous contributions and collaborations with urban experts [26, 27, 31, 45, 47] and also by our goal of making the general tasks of designing, prototyping and sharing interactive urban visualizations easier. Recurrent meetings with an urban expert (co-author of the paper) also influenced the design and implementation of the toolkit. These meetings can be divided into three phases. In the initial meetings, we focused on establishing a shared vocabulary and on critically reflecting on the limitations and tradeoffs of previous works. Then, we reflected on our experience as visualization researchers, tool builders, and urban experts and how our practices could be improved with a unified framework. This process led to a set of design goals and capabilities for UTK. In the final phase, as the framework took shape, we reflected on whether the design goals were being met. Each broad design goal is divided into fine-grained goals. In the follow-up sections, we refer back to these goals when describing UTK's grammar (Section 5) and overall framework (Section 6). Section 7.2 reflects on the design goals.

(D1) Extensibility. The extensibility to create new visualization designs and use these designs at different scales and aggregations is paramount to urban visual analytics tools. Visual analysis of data happens through an iterative process where insights are roughly derived through free exploration and formulation of hypotheses. This freedom is impacted by the system's range of features and execution possibilities and the amount of *friction* it generates during its use. The smaller the gap between what the user conceptualizes as the next exploratory step and the actualization of it, the smaller the friction [72]. Minimizing this gap is not trivial, especially considering complex data such as urban ones, and the relationship between multiple data layers. With this in mind, it is important to offer the user the **ability to create custom visualizations** that match, to the best extent possible, their exploration model (D1.1). These models might also **include complex operations, such as aggregations or comparisons** (e.g., computing the difference between two or more development scenarios [14, 22]) (D1.2). It is also important to **abstract low-level tasks** (loading data, rendering, interaction, navigation) so that the user can focus on the actual tasks that more directly impact the visual analytics process (D1.3).

(D2) Reproducibility. Reproducing an urban visual analytics system is neither an easy nor sufficiently discussed task. Most of the previous works remain closed-source projects, to the detriment not only of the broader visualization community but also of urban experts. These systems are often a complex amalgamation of features, techniques, models, and libraries. Therefore, even though the data is readily available in many cases, the development of the surrounding system is time-consuming and full of intricate implementation details. Rather, we should strive to enable easier collaboration between users and increase the verifiability of future research, especially when considering that many of these tools aim to influence decision-making in real-world settings [67]. We should increase reproducibility in two major ways: (1) **Standardizing formats and inputs** so that data can be easily packaged and shared (D2.1), and (2) **Treating the entire system as an artifact that is shaped around a high-level specification** defined by the user (D2.2). Since the system is built from a specification, reproducing it should be as simple as sharing a JSON file with supplementary datasets. **(D3) Flexibility.** The principle of how flexible a tool is to be adapted to other regions, scales, and data is also rarely considered. Most visual analytics contributions follow one-off collaborations with urban experts and result in highly specialized tools to a single scenario. This flexibility is seldom listed as a requirement for these tools. From a visualization perspective, the effort needed to re-engineer a tool to work in

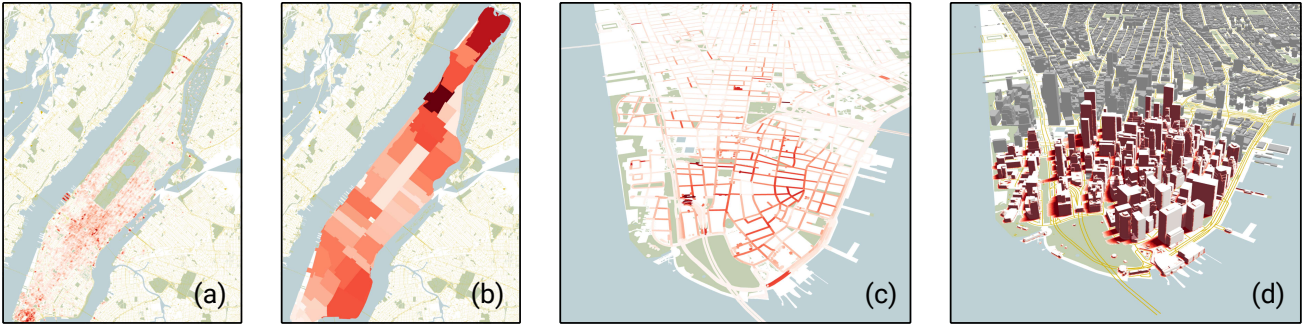


Fig. 2: UTK supports the integration of thematic and physical layers at multiple scales through grammar-defined knots. For example, sunlight access aggregated at a grid (a), ZIP code (b), and street level (c). It also supports thematic data over building surfaces (d).

another region or with other data might not be worthwhile if the proper incentives are not in place (e.g., collaborators to use the tool, perspectives towards a new publication). From an urban domain perspective, extending tools often requires expertise in highly specialized areas of computer science beyond what can be expected from an urban expert. The status quo is where urban visual analytics tools are rarely adopted by experts beyond the original collaborators that helped in the design process. A well-designed framework must offer **support to changing region, scale, or data (D3.1)** and offer solutions to **minimize friction and retooling while avoiding exposing low-level functionalities to the user (D3.2)**.

5 A VISUALIZATION GRAMMAR FOR URBAN VISUAL ANALYTICS

With the aforementioned design principles in mind, we now present the key concept behind UTK: a visualization grammar specifically designed for urban visual analytics. The grammar allows for a detailed and precise specification of the entire visualization interface: views, camera position and direction (for spatial views), data layers (including operations on top of them), and plots. The former is the primary component of the grammar, and all other items are defined in the context of a view. That being said, a view can be manipulated as any other item and can be used to compose multiple or embedded views. Next, we detail our grammar’s rules, grouping them into three categories: composing views, data layers (including thematic, physical, layer integration, and operation), and plotting data. For our abstract code description, we follow Ren et al.’s notation [54], with `:=` representing assignments, `|` representing “or”, `?` representing optional elements, `+` representing one or more elements, and `*` zero or more.

5.1 Composing views

At the highest level of our visualization grammar, we have the element `views` containing one or more (`view`, `camera`) elements of the visualization. Each individual `view` element is defined by: a `map`; `knots`, defining links between data layers (Section 5.2); and optional `plots`, defining 2D plots (Section 5.3). `knots` store the integrated data that will be visualized in both maps and plots. Considering Figure 1 (d), `map` will define the overall 3D map, `knots` will define the integration between physical (i.e., buildings) and thematic (i.e., shadow) layers, and `plot` will define the embedded radial plot.

```
views := (view+, camera+)
view := (map, knots, plot*)
map := (camera_id, (knot_id, interaction)+)
camera := (camera_id, position, direction)
interaction := brush | pick
```

A combination of multiple `view` elements should enable the creation of flexible juxtaposed visualizations [47]. If a `camera` is shared across `view`, then the same camera transformations are used in the views. Conversely, if different `camera` elements are used, we have a multi-view interface. Finally, a `map` is defined as reference to a camera and a list of tuples with a reference to a `knot` to be rendered and an `interaction`. The brushing interaction allows users to select a subset

of a layer. On the other hand, the picking interaction allows the user to select entire objects at once (considering the definition of an object at the geometry level).

5.2 Specifying data layers

The capacity to load, integrate, visualize, and perform operations on top of data layers sets UTK’s grammar apart. As previously mentioned, UTK organizes layers into two types: *thematic* and *physical* layers. Thematic data layers correspond to discrete measurements over the 2D or 3D space. For example, noise complaints and crime occurrences over 2D regions or shadow values over 3D building surfaces. Physical layers correspond to the built or natural environment, such as buildings, road networks, mountains, or regions of interest in a city, such as neighborhoods and parks. The physical layer will be responsible for defining all the geometric information that will ultimately be rendered on the map, while the thematic layer carries the data attributes that will be linked to the physical layer. Thematic layers must be defined through a set of discrete points (coordinates). Physical layers, however, can only be defined through a set of objects and their coordinates:

```
layerthematic := (layer_name, coordinates, color_scale)
layerphysical := (layer_name, objects, coordinates)
```

A `color_scale` will map the thematic data domain to a color range. Both thematic and physical layers are composed of points. For the physical layers, these coordinates can be grouped together to form a 2D or 3D polygon (object), such as ZIP area. A key aspect in many urban analytical workflows is the necessity to navigate and visualize data at multiple scales (Figure 2). The separation of data through layers allows us to cover these scales by simply specifying aggregations of thematic layers over physical layers. For example, a macro-scale analysis can be performed by joining a thematic layer with a physical layer describing neighborhoods. Similarly, a meso-scale analysis can be performed with a join between the thematic layer and a physical layer describing lots and a micro-scale analysis with a physical layer describing buildings.

5.2.1 Integrating layers

UTK’s grammar allows for spatial joins to be performed through *knots* – i.e., an explicit integration between thematic and physical layers. `knots` are high-level data integration operations that link thematic and physical layers, and each `view` is composed of a series of `knots`.

```
knots := (knot ⊗ filter)+
knot := (knot_name, (integration_scheme)+)
filter := bounding_box | address
```

Each `knot` is defined as a series of chained `integration_scheme` elements. An `integration_scheme` is defined as a `spatial_relation` (e.g., nearest, contains, within) between either two knots or two layers (specified by their names):

```
integration_scheme := ((layerin | knotin), (layerout | knotout),
    spatial_relation?, operation?)
spatial_relation := nearest | contains ...
```

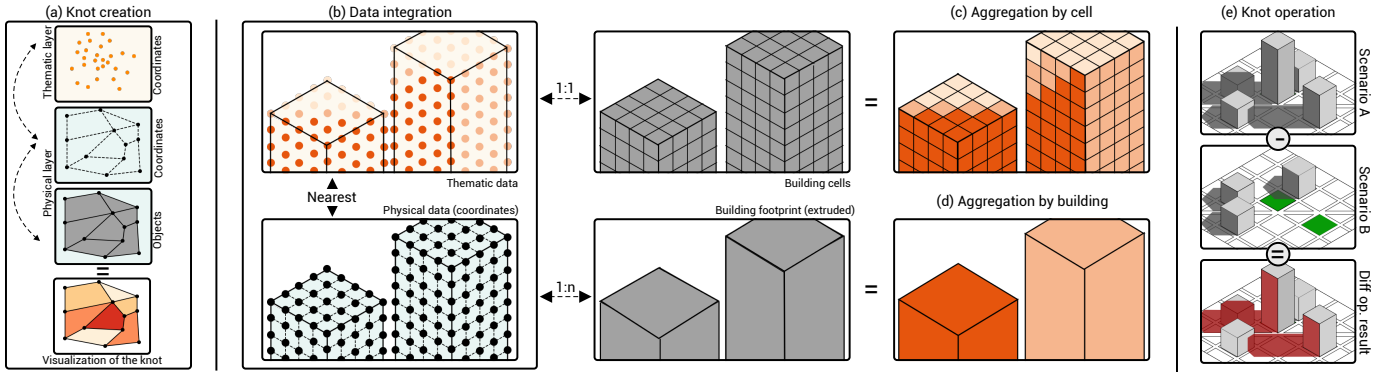


Fig. 3: Integration between thematic and physical layers. (a) A knot is defined as the combination of a thematic and a physical layer through a spatial relation (e.g., nearest, contains) and an aggregation operator (e.g., sum, mean) if the relationship is $1 : n$. Each thematic data point will be mapped to an individual physical layer coordinate (e.g., grid) and/or object. (b) For example, thematic data that is defined at a regular grid can be linked with a grid defined at the building level. Aggregations can then be performed at a building cell level (c) or building footprint level (d). (e) The concept of knots also allows for more complex data operations, such as the comparison between alternate planning scenarios.

A filter \otimes operation performs a selection of items from knots taking into account a bounding box or specified address. If the spatial relation is $1 : n$ (i.e., physical object contains n thematic data points), we can specify an aggregation operation (e.g., mean), or a user-defined JavaScript function (`custom_function`) that aggregates the values.

```
operation := aggregation | custom_function
aggregation := min | max | sum | mean ...
```

Knots provide a flexible and intuitive way to chain integration schemes over multiple physical scales. For example, thematic point data (e.g., taxi pickups) can be aggregated over physical object data (e.g., ZIP codes) by specifying a knot with `contains` (i.e., a link between point and object) followed by an `aggregation`. This concept also allows for more complex integration, such as the ones considering sunlight access values defined at the surface of a building. For example, consider the case shown in Figure 3 (b, c). In (b, top), an input thematic layer contains sunlight access values defined at a coordinate level. A nearest `spatial_relation` creates a link between each thematic coordinate and the nearest coordinate in the output building physical layer (b, bottom). Aggregations can then be performed at building cell level (c) or footprint level (d). Figure 4 shows the results of these aggregations. Through these operations, UTK supports multi-scale exploration (D3.1).



Fig. 4: UTK abstracts spatial joins through the use of knots. Top: Knot integrating thematic and physical layers. Bottom: Knot integrating a thematic layer with the average per building.

5.2.2 Operations with knots

Knots also facilitate the specification of layer operations, supporting **D1.2**. This is particularly important for *what-if* analyses in which an urban expert must compare different built environment configurations and evaluate the one that best satisfies certain criteria (e.g., building designs that minimize the amount of shadow on neighboring parks). Figure 1 (left) presents an example of the shadow in two scenarios: with and without two buildings near a park in Chicago. The difference (Figure 1 (c)) is shown in shades of blue. UTK's grammar enables such operations to be specified through the `operation` element. For example, the knot (`difference, (nearest, shadow0, shadow1, shadow0-shadow1)`) computes the difference between two knots (`shadow0` and `shadow1`). Note that since both knots are defined over the same physical layer (i.e., buildings), the `nearest` element is redundant. Operations can only be performed between knots with the same physical layer, but their thematic layers can have different resolutions. Since knots will map each thematic data point to an individual physical layer coordinate (through the `integration_scheme`), this guarantees that there will not be any mismatch between knots.

5.3 Plotting data

UTK's grammar also facilitates the creation of 2D plots, both linked to the view or embedded in the 3D environment. Both types have been shown to be useful in our previous evaluations [47], and support **D1.1**. For the specification of these plots, we enable the insertion of Vega-Lite's specifications into one defined according to UTK's grammar. In doing so, we leverage a wealth of easy-to-use visualizations that have been proposed by the Vega community at large (**D1**). A UTK plot is defined as:

```
plot := (vega_spec, (knot_name, arrangement)+,
        interaction?, args?)
arrangement := linked | embedded
embedded := surface | footprint ...
```

where `vega_spec` is a Vega-Lite specification. Two types of arrangements are supported. In a `linked` arrangement, plots are displayed on top of the 3D map. In an `embedded` arrangement, plots are embedded directly into the 3D map. Two different embedding arrangements are supported: `surface`, where a 2D plot is displayed on the surface of a physical layer, and `footprint`, where a 2D plot is displayed on the horizontal cross-section of a physical layer. Plot parameters (e.g., number of segments in a radial plot) are also defined in the grammar (`args`) and passed to both Vega-Lite and to the frontend (to compute the intersection between building and horizontal cross-section). Figure 5 shows examples of these possibilities.

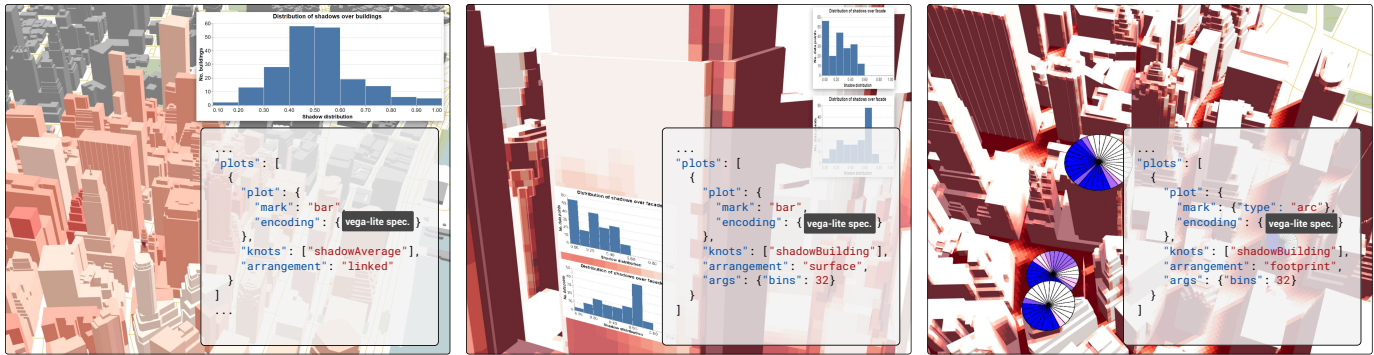


Fig. 5: Different types of physical and thematic layers integration. Left: **linked view** showing the distribution of sunlight access. Middle: **embedded surface plots** showing the distribution of sunlight access over building surfaces. Right: **embedded footprint plots** showing horizontal cross-section distribution of sunlight access.

6 UTK: A TOOLKIT FOR URBAN ANALYTICS

UTK is a general toolkit for urban analytics that uses the previously introduced visualization grammar to create new map-based visualizations. However, given the complexity of urban data and workflow, UTK also offers data loading and parsing capabilities. Our framework can then be divided into two modules, detailed next.

6.1 Backend module

UTK’s backend is a collection of data management functionalities for loading, saving, transforming, and aggregating the data into formats appropriate to the frontend. Its functionalities are exposed through a Python API that can be accessed in computational notebooks. In line with **D1.3**, many of the time-consuming aspects of urban analytics are abstracted by this component in such a way that, in order to have the data for an initial urban interface, the user only needs to:

```
import utk
uc = utk.OSM.load('Chicago,USA',layers=['buildings'])
uc.save('chi')
uc.view()
```

This will download and parse OpenStreetMap’s (OSM’s) building data for the city of Chicago and open a web browser with a default grammar specification to visualize the data. UTK supports a wide range of data that model physical and thematic aspects of a city. To ensure consistency, the physical layer is immutable in the system, meaning that all operations done using knots only transform the thematic layer. The immutability of the physical layers adds stability to the system since the users do not need to be concerned about ensuring that the operations maintain the mapping of the physical/geometric representations. The layers can be loaded from multiple data formats, including Pandas’ DataFrames, GeoJSONs, shapefiles, and CSV files, supporting **D2.1**.

Physical layers. We support four types of physical layers:

OpenStreetMap data. OSM provides spatial data for a large number of cities. The data, however, is specified as tags and elements that represent a number of urban features, such as parks, water bodies, and buildings. Such data must be parsed to be used by rendering and visualization pipelines. With UTK, a user can specify a region of interest through a bounding box, polygon, or address (similar to the example above). UTK will automatically download and parse the data into UTK-ready layers and formats, such as 3D triangle meshes for OSM buildings, 2D polygons for OSM parks, and network data for street networks. The framework also supports the Protocolbuffer Binary Format – a data format used to store OSM data locally.

Polygons and triangle meshes. UTK supports both 2D polygons as well as 3D triangle meshes representing physical features, such as neighborhoods, ZIP codes, and buildings.

Network data. Our framework also supports street and footpath sidewalk data that usually describes a graph where vertices correspond to street corners and edges to road or sidewalk segments.

Grid. UTK supports a fine-grained grid that covers the city and/or the surface of buildings. These grids can be used to aggregate and overlay thematic data over surfaces.

Thematic layers. UTK can load data where each point is defined by a latitude, longitude, height, and value tuple. In practice, any value attributed to a coordinate in the 3D space can be loaded.

As an example to showcase UTK’s flexibility in using data from previous efforts, we have incorporated Miranda et al. [45] shadow accumulation algorithm and also made it available via our Python API. For each surface point in the urban environment, the computed dataset contains the accumulated shadow over a user-specified period of time – i.e., how much of the view to the sun was occluded considering the accumulation period. That being said, all data used in the grammar specification must follow a two-step loading process. The first step involves preprocessing the supported data so that they conform to a format accepted by UTK. The second step establishes links between the different layers to create knots, as specified in the grammar. The .utk files are the result of this two-step process.

6.2 Frontend module

UTK’s web interface is composed of two components: A JSON editor and a map view (Figure 6). The JSON editor allows for the editing, saving, and loading of visualization specifications through an embedded JSON editor. We use JSON for its conciseness and portability (**D2.2**). As the user adds new views, layers, knots, plots, or changes parameters in the JSON file, the map view is updated. If the user navigates the 2D or 3D environment and changes camera parameters, the new information will be saved in the JSON file. The map view will also display the linked and embedded plots specified using Vega-Lite’s grammar.

When specifying **layer** elements, the user defines a path to the .utk file as the name of the layer. These references can only be made inside the `layer_in` element of the `integration_scheme`. This is the only place in the specification where the user can make direct references to locally stored data. Otherwise, the data is only manipulated as **knots**. This allows for isolation between data and interface so that the interpreter can guarantee consistency of operations.

Data interaction. Interactions are supported on the map and on the 2D plots by specifying an `interaction` for `map` and `plot` elements, respectively. Selected entities in the 2D plots are highlighted on the map and vice versa. This link between plots and physical elements is facilitated by the knots. A reference to a knot is required when creating a new plot – hence the data ingested to Vega-Lite is going to be dictated by how the knots are created. For example, if a knot is aggregating data by ZIP area, the data passed to the plot will be at the same resolution. And if a bar chart is displayed, each bar element will have a 1 : 1 mapping to a particular ZIP code. A selection in the chart will then highlight the appropriate area.

6.3 Implementation & analysis workflow

Implementation. The UTK framework has been developed using a client-server architecture. UTK’s frontend uses React for user interfaces and WebGL for the 3D map component. UTK’s backend was implemented using Python. For reverse geocoding and to access OSM

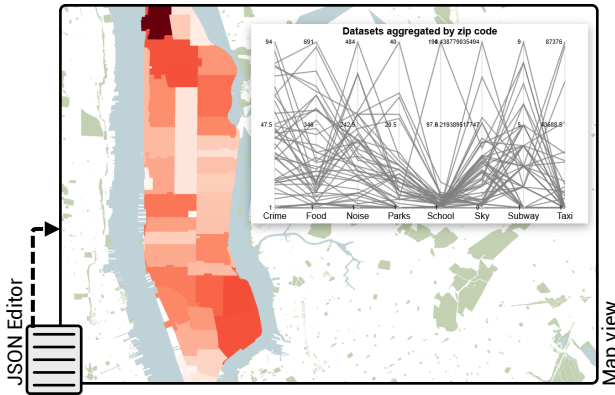


Fig. 6: UTK’s frontend has a built-in JSON editor, and modifications to the JSON file are visualized in the map view. Here, we visualize the spatial distribution of noise complaints in Manhattan, with a linked parallel coordinate plot showing other urban datasets.

data, we use GeoPy and Overpass, respectively. Spatial joins and aggregations are performed using Geopandas’ functionalities.

Analysis workflow. Figure 7 presents a high-level overview of UTK’s architecture and the expected ways that the user can interact with both UTK’s frontend and backend. If no data was previously created, in the first step, the user should set up the UTK environment (Figure 7 (right)). This involves creating a new Jupyter Notebook and specifying an area of analysis using UTK’s Python API. After that, the physical data layers will be created and persisted as .utk files. Even though the framework already offers a number of data functionalities, users can also make use of other libraries and load their output into the framework as long as the data adheres to UTK’s formats. Having a separation between the frontend and the backend enables users to go back and forth between UTK’s Python API and the grammar visualization functionalities. After the data is created, the user can visualize it using the frontend and use UTK’s grammar to create the visualizations. When deploying to the web, users can optionally hide the JSON editor and only expose the map view. A tight connection between visualization and data through knots ensures that the user has enough flexibility to iteratively modify the data in Jupyter Notebooks and easily visualize the new data without having to worry about setting up the environment from scratch (D3.2).

7 EVALUATION

To demonstrate UTK’s capabilities, flexibility, and ease of use, we present examples motivated by real-world problems and inspired by previous collaborations with architects and urban planners. Then, we report on a series of one-hour semi-structured interviews with five different urban experts, where we asked them their perspectives on UTK’s usability, limitations, and potential extra features to be added.

7.1 Example gallery

Next, we highlight examples showcasing how UTK can be used in different analysis workflows, motivated by the needs of urban domains.

7.1.1 Example 1: Building energy efficiency

In this use case, we are exploring buildings in Chicago’s Near North Side area to identify locations in each building that are the most energy efficient in terms of receiving more shadows during summer than winter. As noted in previous studies [35,66], more shadows during summer can lead to cooler internal temperatures, and more sunlight during winter has the opposite effect. In both cases, shadows can contribute to a lower dependency on AC. We start this analysis by using UTK’s backend to create a physical layer with Chicago’s buildings and twelve thematic layers with the accumulated shadows for a day in each month of the year. Creating these layers is simple, requiring less than 10 lines of Python code. We then move to UTK’s frontend and, using the JSON editor, we create twelve knots, each linking the physical layer with a thematic layer. To model the trade-off between summer and winter shadows, we create a new knot that averages the accumulated shadows,

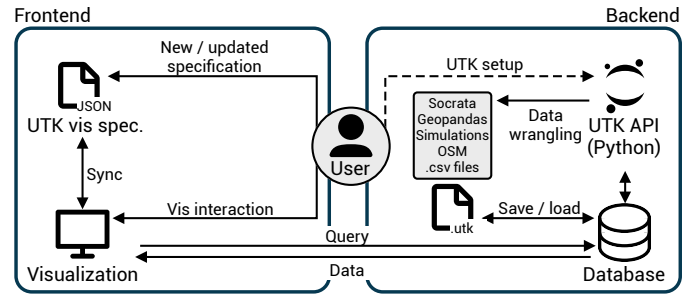


Fig. 7: Overview of UTK’s architecture and how the user interacts with the frontend and backend components of the framework.

with weights depending on the month (see Miranda et al. [45] for the weights). Again, the concept of knots provides an easy abstraction for averaging the accumulated shadows. Note that UTK’s expressivity allows us to chain aggregations. We define a separate knot for each month’s shadow and perform an operation over knots to get the final result, as shown in Figure 1 (d, top).

UTK also allows us to easily integrate 2D and 3D visualization to minimize occlusion problems. Figure 1 (d, bottom) highlights the specification of an **embedded** radial plot that displays the data distribution along the horizontal cross-section (**footprint**), with shades of blue denoting facades that have more positive than a negative shadow, according to our weighted average. One could also use iterate over different types of plots, as shown in Figure 5. With minimal effort in terms of navigation, it is possible to assess that the selected buildings have very different cross-section profiles. By analyzing data at the building level, experts can gain insights into the unique characteristics of individual buildings and their surrounding environments, allowing them to develop targeted energy consumption strategies. Moreover, the analysis can reveal which floors and sides of each building have been exposed to more summer shadows on average. Through the data analysis of building and land use, experts can also identify candidate areas for urban farming and green infrastructure [50], promoting more sustainable and resilient urban development.

7.1.2 Example 2: Historic preservation

Preserving historic landmarks is key to the preservation of cultural heritage. One of the important but often overlooked causes of building deterioration is the shadow cast by neighboring buildings [41]. In Boston, the shadow cast by new developments has long been the subject of public debate. In this case, we will use UTK for *what-if* or alternate scenario testing. We have chosen the John Hancock Tower in Back Bay, Boston – a 60-story skyscraper. We would like to measure the accumulated shadows cast by Hancock Tower on the 146-year-old Trinity Church and the Public Garden. We start by using UTK’s backend to download OSM data for Boston and perform the shadow simulation for winter and summer. Next, using UTK’s frontend, we create four knots, integrating physical and thematic layers for summer (with and without the tower) and winter (with and without the tower). Here, UTK’s capabilities in computing differences between knots can help compute the amount of accumulated shadows cast by the tower. Considering two knots with the computed shadow for the current and alternate states, we can simply create a new one and compute their difference: $knot_{diff}=(diff,(knot_{s0},knot_{s1}, nearest, knot_{s0}-knot_{s1}))$

Figure 8 (a) shows the current state of accumulated shadows during summer and winter (the number of accumulated minutes is shown near the color scale). The right panel depicts the amount of shadow that would disappear with the removal of the tower; in other words, UTK enables us to easily extract the amount of accumulated shadow cast by this specific building. We can see from the figure that during summer, the orientation of the building resulted in a low amount of accumulated shadow cast over the church compared to the total accumulated shadow cast by other buildings in its vicinity. During winter, the eastern side of Trinity is more severely shadowed by the tower. As illustrated in Figure 8 (b, top), during winter, the shadow cast by the Hancock Tower also spreads over the Public Garden, located miles away.

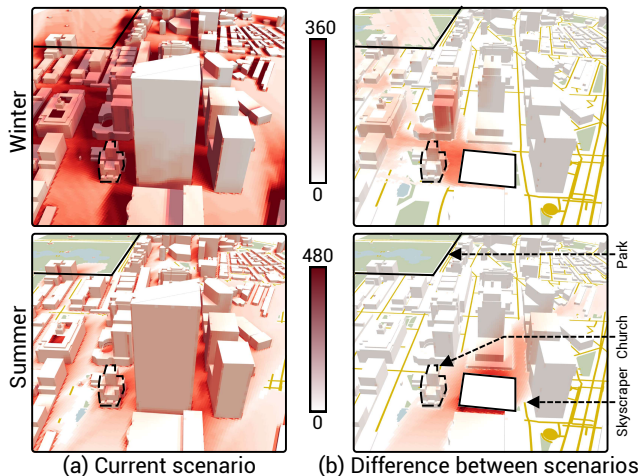


Fig. 8: Evaluating shadows cast by a skyscraper in Boston. (a) The current shadow situation during winter and summer. (b) Amount of shadow cast solely by the skyscraper – i.e., shadow that would disappear with the removal of the building. The color scales show the accumulation period (360 minutes during winter and 480 minutes during summer).

7.1.3 Example 3: Neighborhood signatures

Cities are often described by their most prominent features, such as cultural diversity and economic activities. Recognizing the unique characteristics of each neighborhood is crucial in planning and resource allocation. Given that, identifying patterns and understanding the dynamics between livability measures require analyzing datasets across scales [27].

In this example, we will use eight datasets from the NYC Open Data portal [4] for neighborhood characterization: crime and noise reports; restaurant, parks, and subway locations; sky exposure; school quality reports and taxi pickups. We start by using the backend to parse the CSV files from the identified datasets and create the thematic layers. In this step, we also use the backend to download OSM data for NYC and parse shapefiles with the neighborhoods. Using the frontend, we then specify knots that aggregate thematic layers over physical ones. For example, to create thematic and physical layers:

```
layer_noise=(noise,coords), layer_zip=(zip,objs,coords)
```

Aggregations and joins can be performed through knots that link physical and thematic layers. We then create a knot with a single integration scheme between `layer_noise` and `layer_zip`, specifying that noise complaints within ZIP areas will be added together:

```
knot_noise=(noise2zip,(layer_noise,layer_zip,contains,sum))
```

We can then specify a Vega-Lite parallel coordinate plot that visualizes the data from knots created for each of the previously mentioned dataset. This creates the interface shown in Figure 6 (from a UTK specification with fewer than 200 lines). The map shows a neighborhood in Manhattan with the most noise complaints. Looking at other indicators, we can see that it has a low sky obstruction rate, signaling a lower density neighborhood with relatively low access to public transit. Considering that it does not have a large number of taxi pickups and it is not a destination for food services, this elevated number of noise complaints seems alarming. Further investigation into the types of complaints is required to address the concerns of its residents. This insight was easily extracted from simple interactions with UTK, enabling users to spend more time on analysis and exploration.

7.1.4 Example 4: Tripping risk in Downtown Boston

Sidewalk surface is among the most important factors in determining the risk of outdoor falls [64]. Slippery surfaces pose a major challenge to pedestrians navigating the outdoor environment in cold and snowy weather [18]. In the absence of sunlight, and when the temperature suddenly drops, a transparent and slippery form of thick coating of ice, known as *black ice*, can form on top of pavements. In general, the temperature of the surface and the type of surface material are key contributing factors to black ice formation [33].

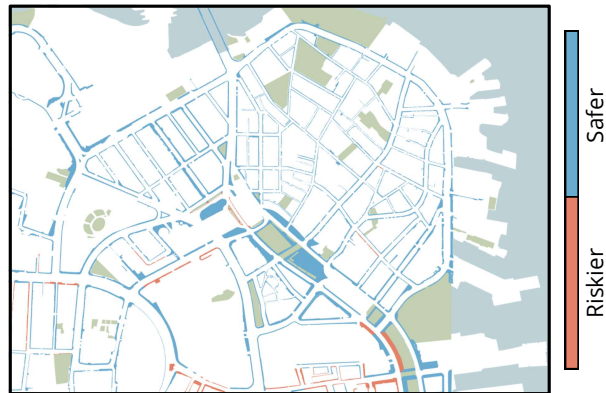


Fig. 9: Knot's custom operator specifies whether a sidewalk is dangerous or not based on its surface material and shadow.

In this example, we use surface material data [31] together with shadow accumulation data aggregated at the sidewalk level to calculate a risk measure for identifying places that can pose a higher risk of falls in cold seasons [32]. Our risk measure is calculated using three factors of interest to urban accessibility experts: the percentage of bricks, the percentage of granite, and the accumulated shadow data for a day in December (winter solstice). UTK's backend was used to create a physical layer from a shapefile with sidewalk geometry and thematic layers from our shadow simulation and from a CSV file with the sidewalk material data. We classified dangerous sidewalks as the combination of two knots, one with accumulated shadow and another one with the sidewalk material. We considered dangerous sidewalks as the ones with bricks and concrete, and more than half of the time under shadow during the accumulation period. This can be easily represented through the operation:

```
knot_danger = (danger, (knot_shadow, knot_mat, nearest, op))
op=((mat=='brick' || mat=='conc') && shadow>0.5?0:1)
```

We chose Boston's downtown and North End neighborhoods for their historic fabric and because they are considered popular destinations. Figure 9 shows our tripping risk map. UTK allows for easy iteration over multiple parameters through a simple combination of knots. The operation can be adjusted based on user's preferences and city-scale fall risk models by simply editing the JSON specification.

7.2 Reflection on design goals

In terms of extensibility, UTK provides the ability to create custom plots and integrate them into map-based visualizations. Our examples show two different possibilities (radial plot, parallel coordinates) of juxtaposed and embedded plots, satisfying **D1.1**. Examples 1 and 4 highlight the use of knots to perform operations across layers (**D1.2**). Example 4, specifically, goes toward tackling a design consideration recently highlighted by Saha et al. [55], i.e., facilitating the blending of datasets for multivariate analysis. Throughout the examples, UTK abstracted many low-level steps involved in urban data analytics, namely downloading and parsing OSM data, joining data for visualization purposes, rendering, and navigation, satisfying **D1.3**. From an implementation perspective, UTK's backend and frontend follow a modular approach, ensuring that new functionalities can be added without the need to modify the core code.

For reproducibility, the examples highlight UTK's support for standard formats, namely CSV files in example 3 and shapefiles in example 4, supporting **D2.1**. Also, rather than relying on cumbersome deployments, visualizations can be shared through small and concise JSON files (**D2.2**). Considering its flexibility, UTK also provides a suite of data functionalities that facilitate porting use cases for other regions (**D3.1**). Examples 1 and 2 could be easily replicated in other cities just by changing the layer creation step. Similarly, new datasets could be added to example 3 or other assessment factors could be considered in example 4. The examples also highlight analyses at different scales (sidewalk, building, and neighborhood). With UTK, visualizing the basic layers of a city requires only a few lines of Python code to import

the data, along with a JSON file specifying the desired visualization. UTK abstracts all necessary data transformations, enabling users to focus on the analytical aspects of their work without worrying about the technical details (D3.2).

7.3 Experts' feedback

To gain domain perspectives into several aspects of UTK, we interviewed five urban experts from different specialties, roles, and levels of seniority. None of the experts are co-authors in this paper. We asked the experts to asynchronously interact with UTK by reading a quick start document and three tutorials highlighting different use cases. Then, we proceeded to a semi-structured interview where the expert was asked questions regarding UTK's features, limitations, and usability. We interviewed: two climate scientists with experience in climate and urban morphology based in North America (P1 and P2), an urban planner with experience in urban micro-climate based in Asia (P3), an architect and urban planner interested in the analysis of urban morphology (P4), a geographer and GIS expert that worked as the deputy secretary of urban planning for a large city in South America (P5). P5 has an MSC degree, and the others have PhDs.

Use cases. P1 and P2 were primarily interested in using UTK for heat vulnerability analysis. They mentioned it would be interesting to *"integrate temperature, humidity, and shading for city-scale analysis and compare seasons."* They were also receptive to how UTK moves *"away from static maps and enables alternate scenario planning, ideal for climate change resilience."* Finally, P1 mentioned how UTK *"facilitates engagement not only across disciplines, but also across urban communities"*, as certain outcomes can *"target users with different levels of expertise."* P1 mentioned that the visual analysis could be *"authored by an expert, but delivered to a community through a web portal, hiding the grammar from them."* One concrete example was the possibility to create *"hourly temperature shadow maps for communities to minimize heat exposure."*

P3 was responsive regarding the multi-scale capabilities offered by UTK, enabling *"switching contexts and scales in a fast way"* and the *"exploration of how different sides of buildings are exposed to sunlight."* In their practice, P3 uses temperature and sky view factor simulations to compute the pedestrian level of comfort. They mentioned that UTK's *"capacity to aggregate data and perform a difference of the results is especially useful to assess the impact of different parameters and conditions of the simulations"*. P4 pointed out the potential for collaborative analysis using the framework: *"researchers could easily share their visualizations instead of cumbersome GIS files."* They mentioned that making the fast shadow accumulation computation available could lead to major use cases given that it's *"easy to use and integrate into existing workflows."* Moreover, noise and property management were brought up as potential use cases. P5 mentioned that the tool is ideal for analyzing urban morphology and that *"3D makes it more attractive to users and a great tool for communication."* They mentioned how, for urban land use decisions, *"3D is important to assess possible new developments and communicate that to stakeholders."*

Usability & adoption. We asked the urban experts how easy it would be to integrate UTK into their current practices. They all mentioned that they see value in the tool. P3 mentioned that, by adopting 2D and 3D metaphors, the framework can *"cater to a broader audience, as opposed to just supporting one or the other."* Regarding UTK's usability, P4 mentioned the need for more intuitive ways to interact with the physical layer: *"it would be helpful to click on a building and interactively change its height."* P4 was also ambivalent regarding having access to the grammar, pointing out that *"learning what JSON I can write takes time"*, but appreciated that they *"don't have to work with too many tools and menus."* P5 was also more cautious, saying that *"it would require training to educate people on the grammar, with examples to showcase its use."* But on the other hand, *"GIS tools are complex and cumbersome, lightweight solutions would be very welcomed."* They mentioned that the *"usability of current GIS tools is a problem and experts sometimes resort to limited toolboxes such as Google Maps / Earth for their vis."* And that *"even though data is being made available, visualizations are usually very simple, whatever*

is supported by Google Maps." P5 highlighted that *"UTK could easily plug into a data lake to visualize bus data, with operational indicators for the city."* Moreover, it provides a *"lightweight approach to visualize cities in 3D, good for both city management and communication."*

7.4 Comparison with existing tools

In this section, we provide a comparison between UTK and popular GIS tools, namely ArcGIS and QGIS. One of the drawbacks of these tools is the overwhelming number of operations and toolboxes. For instance, ArcGIS Pro has more than 200 different operations in only one of its 41 toolboxes, each having its specific requirements for data and processing, and according to previous work, users often struggle to find the right one for their task [73]. UTK, on the other hand, offers a concise specification that is easy to learn and modify. The grammar-based framework makes it possible to perform a wide range of visual analytic tasks with our grammar elements, offering a compelling alternative to traditional GIS tools. The ability to define custom links between layers summarizes a range of geospatial operators into a single functionality.

Another limitation of existing tools is the lack of support for reproducible and shareable formats in many of their toolboxes. To reproduce an analysis done with those operations, a user needs access to the tool, which might require costly licensing and a cumbersome installation process. In contrast, UTK saves the visualization and processing specification in human-readable JSON files. By building their own chain of operations, users can easily iterate designs, ensuring more transparency and reproducibility of their work. The self-contained nature of grammar-based visualizations allows for easier collaboration, sharing, and replication of visualizations. ArcGIS and QGIS also have limited capabilities in creating plots and embedding them into physical layers [34]. UTK provides customizable plotting functionalities (through Vega-Lite), as well as the ability to manipulate plots with flexible integration options (e.g., embedded on building surfaces [47]). Beyond embedding plots, UTK directly links them to the underlying data source, enhancing interactivity and contextualization.

8 CONCLUSION AND FUTURE WORK

This paper presents the Urban Toolkit, a flexible grammar-based declarative framework for urban visual analytics. UTK streamlines many of the usual data tasks common in urban analyses and, as shown in our use cases, makes it possible to prototype complex analysis scenarios relatively easily, in line with what has been recently highlighted by Saha et al. [55] and Ziegler and Chasins [73]. We envision that our toolkit can popularize the process of authoring urban visual analytics systems, as well as facilitate user studies for a better understanding of the benefits of 2D and 3D visualizations in urban environments [47].

While promising, our current implementation presents some limitations. As shown in the examples throughout the paper and the accompanying video, UTK can leverage data concerning different layers from reasonably large portions of a high-density urban area (e.g., Manhattan with approximately 3 million triangles). However, high-resolution geometric data can also make matching operations between physical and thematic layers computationally expensive. As of now, our implementation uses Geopandas to perform spatial joins, a library that was not optimized for interactivity. We intend to investigate how to integrate more efficient algorithms and data structures to enable higher resolution physical models and thematic data, as well as faster joins [26]. Furthermore, UTK's support for temporal data is limited. In our current implementation, different knots can be used to aggregate data referring to different time ranges, but that is not scalable. We intend to investigate extensions to the grammar and the incorporation of timeseries data structures to better handle temporal data.

Since urban planning is tightly linked to simulations for scenario planning, we plan to incorporate other simulation capabilities, such as wind and noise, and make them available for researchers and practitioners through our API. In addition, we plan to conduct a controlled user study to evaluate ease-of-use, as well as investigate the deployment of applications in different platforms, such as immersive environments [13, 60]. By making UTK public, we hope to pave the way for a more robust ecosystem of open tools for urban visual analytics.

ACKNOWLEDGMENTS

We would like to thank the reviewers for their constructive comments and feedback. This study was partly funded by the University of Illinois' Discovery Partners Institute (DPI), CNPq (316963/2021-6), and FAPERJ (E-26/202.915/2019, E-26/211.134/2019).

REFERENCES

- [1] Bokeh. <https://bokeh.org/>. Accessed: March 31, 2023. 3
- [2] deck.gl. <https://deck.gl/>. Accessed: March 31, 2023. 3
- [3] Matplotlib. <https://matplotlib.org/>. Accessed: March 31, 2023. 3
- [4] NYC OpenData. <https://opendata.cityofnewyork.us/>. Accessed: March 31, 2023. 8
- [5] Tableau. <https://www.tableau.com/>. Accessed: March 31, 2023. 3
- [6] Trifacta. <https://www.trifacta.com/>. Accessed: March 31, 2023. 3
- [7] G. Andrienko, N. Andrienko, S. Rinzivillo, M. Nanni, D. Pedreschi, and F. Giannotti. Interactive visual clustering of large collections of trajectories. In *2009 IEEE Symposium on Visual Analytics Science and Technology*, pp. 3–10. IEEE, Oct. 2009. doi: 10.1109/VAST.2009.5332584 1
- [8] M. Angelini, J. Buchmüller, D. A. Keim, P. Meschenmoser, and G. Santucci. SurgeryCuts: Embedding additional information in maps without occluding features. *Computer Graphics Forum*, 38(3):237–247, 2019. doi: 10.1111/cgf.13685 3
- [9] R. Avini, P. Kumar, and S. Hughes. Wind loading on high-rise buildings and the comfort effects on the occupants. *Sustainable Cities and Society*, 45:378–394, Feb. 2019. doi: 10.1016/j.scs.2018.10.026 2
- [10] M. Batty and A. Hudson-Smith. Visual analytics for urban design. *Urban Design*, 132:38–41, 2014. 2
- [11] M. Bostock and J. Heer. Protovis: A graphical toolkit for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1121–1128, Nov. 2009. doi: 10.1109/TVCG.2009.174 2
- [12] M. Bostock, V. Ogievetsky, and J. Heer. D³ data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, Dec. 2011. doi: 10.1109/TVCG.2011.185 2, 3
- [13] P. Butcher, N. John, and P. Ritsos. VRIA: A web-based framework for creating immersive analytics experiences. *IEEE Transactions on Visualization and Computer Graphics*, 27(7):3213–3225, July 2021. doi: 10.1109/TVCG.2020.2965109 2, 3, 9
- [14] A. Chakraborty and A. McMillan. Scenario planning for urban planners: Toward a practitioner's guide. *Journal of the American Planning Association*, 81(1):18–29, Jan. 2015. doi: 10.1080/01944363.2015.1038576 3
- [15] R. Chang, G. Wessel, R. Kosara, E. Sauda, and W. Ribarsky. Legible Cities: Focus-dependent multi-resolution visualization of urban relationships. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1169–1175, Nov. 2007. doi: 10.1109/TVCG.2007.70574 2
- [16] S. Chen, X. Yuan, Z. Wang, C. Guo, J. Liang, Z. Wang, X. Zhang, and J. Zhang. Interactive visual discovering of movement patterns from sparsely sampled geo-tagged social media data. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):270–279, Jan. 2016. doi: 10.1109/TVCG.2015.2467619 1
- [17] Z. Chen, Y. Wang, T. Sun, X. Gao, W. Chen, Z. Pan, H. Qu, and Y. Wu. Exploring the design space of immersive urban analytics. *Visual Informatics*, 1(2):132–142, 2017. doi: 10.1016/j.visinf.2017.11.002 2
- [18] T. Chippendale and M. Boltz. The neighborhood environment: Perceived fall risk, resources, and strategies for fall prevention. *The Gerontologist*, 55(4):575–583, Aug. 2015. doi: 10.1093/geront/gnu019 8
- [19] D. Cornel, A. Buttlinger-Kreuzhuber, A. Konev, Z. Horváth, M. Wimmer, R. Heidrich, and J. Waser. Interactive visualization of flood and heavy rain simulations. *Computer Graphics Forum*, 38(3):25–39, June 2019. doi: 10.1111/cgf.13669 2
- [20] D. Cornel, A. Konev, B. Sadransky, Z. Horváth, E. Gröller, and J. Waser. Visualization of object-centered vulnerability to possible flood hazards. *Computer Graphics Forum*, 34(3):331–340, June 2015. doi: 10.1111/cgf.12645 3
- [21] C. Correa, D. Silver, and M. Chen. Feature aligned volume manipulation for illustration and visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1069–1076, Sept. 2006. doi: 10.1109/TVCG.2006.144 3
- [22] B. Delaney. Visualization in urban planning: they didn't build LA in a day. *IEEE Computer Graphics and Applications*, 20(3):10–16, June 2000. doi: 10.1109/38.844365 3
- [23] Z. Deng, D. Weng, J. Chen, R. Liu, Z. Wang, J. Bao, Y. Zheng, and Y. Wu. AirVis: Visual analytics of air pollution propagation. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):800–810, Jan. 2020. doi: 10.1109/TVCG.2019.2934670 2
- [24] Z. Deng, D. Weng, S. Liu, Y. Tian, M. Xu, and Y. Wu. A survey of urban visual analytics: Advances and future directions. *Computational Visual Media*, 9(1):3–39, Mar. 2023. doi: 10.1007/s41095-022-0275-7 1, 2
- [25] H. Doraiswamy, J. Freire, M. Lage, F. Miranda, and C. Silva. Spatio-temporal urban data analysis: A visual analytics perspective. *IEEE Computer Graphics and Applications*, 38(5):26–35, Sept. 2018. doi: 10.1109/MCG.2018.053491728 1, 2
- [26] H. Doraiswamy, E. Zacharatos, F. Miranda, M. Lage, A. Ailamaki, C. Silva, and J. Freire. Interactive visual exploration of spatio-temporal urban data sets using Urbane. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD '18*, pp. 1693–1696. Association for Computing Machinery, May 2018. doi: 10.1145/3183713.3193559 3, 9
- [27] N. Ferreira, M. Lage, H. Doraiswamy, H. Vo, L. Wilson, H. Werner, M. Park, and C. Silva. Urbane: A 3D framework to support data driven decision making in urban development. In *2015 IEEE Conference on Visual Analytics Science and Technology, VAST '15*, pp. 97–104. IEEE, Oct. 2015. doi: 10.1109/VAST.2015.7347636 1, 2, 3, 8
- [28] N. Ferreira, J. Poco, H. Vo, J. Freire, and C. Silva. Visual exploration of big spatio-temporal urban data: A study of New York City taxi trips. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2149–2158, Dec. 2013. doi: 10.1109/TVCG.2013.226 1, 2
- [29] G. Garcia-Zanabria, M. Raimundo, J. Poco, M. Nery, C. Silva, S. Adorno, and L. Nonato. CriPAV: Street-level crime patterns analysis and visualization. *IEEE Transactions on Visualization and Computer Graphics*, 28(12):4000–4015, Dec. 2022. doi: 10.1109/TVCG.2021.3111146 2
- [30] P. Hemmersam, N. Martin, E. Westvang, J. Aspen, and A. Morrison. Exploring urban data visualization and public participation in planning. *Journal of Urban Technology*, 22(4):45–64, Oct. 2015. doi: 10.1080/10630732.2015.1073898 1
- [31] M. Hosseini, F. Miranda, J. Lin, and C. Silva. CitySurfaces: City-scale semantic segmentation of sidewalk materials. *Sustainable Cities and Society*, 79:103630, Apr. 2022. doi: 10.1016/j.scs.2021.103630 2, 3, 8
- [32] M. Hosseini, M. Saugstad, F. Miranda, A. Sevtsuk, C. Silva, and J. Froehlich. Towards global-scale crowd+AI techniques to map and assess sidewalks for people with disabilities. In *CVPR 2022 AVA (Accessibility, Vision, and Autonomy Meet) Workshop*, 2022. doi: 10.48550/ARXIV.2206.13677 8
- [33] K. Houle. Winter performance assessment of permeable pavements: A comparative study of porous asphalt, pervious concrete, and conventional asphalt in a northern climate. Master's thesis, University of New Hampshire, 2008. 8
- [34] K. Hu. A usability study of ArcGIS Pro: Scaffolding an intuitive and fluid geovisualization workflow. Master's thesis, Purdue University, 2017. 9
- [35] T. Ichinose, L. Lei, and Y. Lin. Impacts of shading effect from nearby buildings on heating and cooling energy consumption in hot summer and cold winter zone of China. *Energy and Buildings*, 136:199–210, Feb. 2017. doi: 10.1016/j.enbuild.2016.11.064 7
- [36] F. Jiang, L. Ma, T. Broyd, W. Chen, and H. Luo. Digital twin enabled sustainable urban road planning. *Sustainable Cities and Society*, 78:103645, Mar. 2022. doi: 10.1016/j.scs.2021.103645 2
- [37] H. Kim, R. Rossi, F. Du, E. Koh, S. Guo, J. Hullman, and J. Hoffswell. Cicero: A declarative grammar for responsive visualization. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, CHI '22*, pp. 1–15. Association for Computing Machinery, 2022. doi: 10.1145/3491102.3517455 3
- [38] G. Kindlmann, C. Chiw, N. Seltzer, L. Samuels, and J. Reppy. Diderot: a domain-specific language for portable parallel scientific visualization and image analysis. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):867–876, Jan. 2016. doi: 10.1109/TVCG.2015.2467449 2
- [39] M. Kraus, J. Fuchs, B. Sommer, K. Klein, U. Engelke, D. Keim, and F. Schreiber. Immersive analytics with abstract 3D visualizations: A survey. *Computer Graphics Forum*, 41(1):201–229, Feb. 2022. doi: 10.1111/cgf.14430 2
- [40] B. Lee, A. Satyanarayan, M. Cordeil, A. Prouzeau, B. Jenny, and T. Dwyer. Deimos: A grammar of dynamic embodied immersive visualisation morphs and transitions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, CHI '23*, pp. 1–18. Association

- for Computing Machinery, 2023. doi: 10.1145/3544548.3580754 3
- [41] C. Lee. Building boom in Boston casts shadows on history and public space. <https://www.nytimes.com/2017/07/11/us/boston-development-shadow-building-boom.html>, 2017. Accessed: March 31, 2023. 7
- [42] Z. Liu, C. Chen, F. Morales, and Y. Zhao. Atlas: Grammar-based procedural generation of data visualizations. In *2021 IEEE Visualization Conference (VIS)*, pp. 171–175. IEEE, Oct. 2021. doi: 10.1109/VIS49827.2021.9623315 3
- [43] S. L'Yi, Q. Wang, F. Lekschas, and N. Gehlenborg. Gosling: A grammar-based toolkit for scalable and interactive genomics data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):140–150, Jan. 2022. doi: 10.1109/TVCG.2021.3114876 2
- [44] A. Malik, R. Maciejewski, S. Towers, S. McCullough, and D. Ebert. Proactive spatiotemporal resource allocation and predictive visual analytics for community policing and law enforcement. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1863–1872, Dec. 2014. doi: 10.1109/TVCG.2014.2346926 1
- [45] F. Miranda, H. Doraiswamy, M. Lage, L. Wilson, M. Hsieh, and C. Silva. Shadow Accrual Maps: Efficient accumulation of city-scale shadows over time. *IEEE Transactions on Visualization and Computer Graphics*, 25(3):1559–1574, Mar. 2019. doi: 10.1109/TVCG.2018.2802945 1, 2, 3, 6, 7
- [46] F. Miranda, H. Doraiswamy, M. Lage, K. Zhao, B. Gonçalves, L. Wilson, M. Hsieh, and C. T. Silva. Urban Pulse: Capturing the rhythm of cities. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):791–800, 2017. doi: 10.1109/TVCG.2016.2598585 2
- [47] R. Mota, N. Ferreira, J. Silva, M. Horga, M. Lage, L. Ceferino, U. Alim, E. Sharlin, and F. Miranda. A comparison of spatiotemporal visualizations for 3D urban analytics. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):1277–1287, Jan. 2023. doi: 10.1109/TVCG.2022.3209474 1, 2, 3, 4, 5, 9
- [48] R. Neuville, J. Pouliot, F. Poux, and R. Billen. 3D viewpoint management and navigation in urban planning: Application to the exploratory phase. *Remote Sensing*, 11(3), 2019. doi: 10.3390/rs11030236 2
- [49] T. Ortner, J. Sorger, H. Steinlechner, G. Hesina, H. Piringer, and E. Gröller. Vis-A-Ware: Integrating spatial and non-spatial visualization for visibility-aware urban planning. *IEEE Transactions on Visualization and Computer Graphics*, 23(2):1139–1151, Feb. 2017. doi: 10.1109/TVCG.2016.2520920 2
- [50] A. Palliwal, S. Song, H. Tan, and F. Biljecki. 3D city models for urban farming site identification in buildings. *Computers, Environment and Urban Systems*, 86:101584, Mar. 2021. doi: 10.1016/j.compenvurbsys.2020.101584 7
- [51] D. Park, S. Drucker, R. Fernandez, and N. Elmquist. Atom: A grammar for unit visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 24(12):3032–3043, Dec. 2018. doi: 10.1109/TVCG.2017.2785807 3
- [52] P. Rautek, S. Bruckner, E. Gröller, and M. Hadwiger. ViSlang: A system for interpreted domain-specific languages for scientific visualization. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2388–2396, Dec. 2014. doi: 10.1109/TVCG.2014.2346318 2
- [53] P. Redweik, P. Teves-Costa, I. Vilas-Boas, and T. Santos. 3D city models as a visual support tool for the analysis of buildings seismic vulnerability: The case of Lisbon. *International Journal of Disaster Risk Science*, 8(3):308–325, Sept. 2017. doi: 10.1007/s13753-017-0141-x 2
- [54] D. Ren, B. Lee, and M. Brehmer. Charticulator: Interactive construction of bespoke chart layouts. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):789–799, 2019. doi: 10.1109/TVCG.2018.2865158 4
- [55] M. Saha, S. Patil, E. Cho, E. Y.-Y. Cheng, C. Horng, D. Chauhan, R. Kangas, R. McGovern, A. Li, J. Heer, and J. E. Froehlich. Visualizing urban accessibility: Investigating multi-stakeholder perspectives through a map-based design probe study. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, pp. 1–14. Association for Computing Machinery, 2022. doi: 10.1145/3491102.3517460 8, 9
- [56] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-Lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):341–350, Jan. 2017. doi: 10.1109/TVCG.2016.2599030 2, 3
- [57] A. Satyanarayan, R. Russell, J. Hoffswell, and J. Heer. Reactive Vega: A streaming dataflow architecture for declarative interactive visualization. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):659–668, Jan. 2016. doi: 10.1109/TVCG.2015.2467091 2, 3
- [58] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, Dec. 2012. doi: 10.1109/TVCG.2012.213 2
- [59] M. Shih, C. Rozhon, and K.-L. Ma. A declarative grammar of flexible volume visualization pipelines. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):1050–1059, Jan. 2019. doi: 10.1109/TVCG.2018.2864841 3
- [60] R. Sicat, J. Li, J. Choi, M. Cordeil, W.-K. Jeong, B. Bach, and H. Pfister. DXR: A toolkit for building immersive data visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):715–725, Jan. 2019. doi: 10.1109/TVCG.2018.2865152 2, 3, 9
- [61] B. Speckmann and K. Verbeek. Necklace maps. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):881–889, Nov. 2010. doi: 10.1109/TVCG.2010.180 3
- [62] G. Sun, R. Liang, H. Qu, and Y. Wu. Embedding spatio-temporal information into maps by route-zooming. *IEEE Transactions on Visualization and Computer Graphics*, 23(5):1506–1519, 2017. doi: 10.1109/TVCG.2016.2535234 3
- [63] J.-H. Tang, B.-C. Lin, J.-S. Hwang, L.-J. Chen, B.-S. Wu, H.-L. Jian, Y.-T. Lee, and T.-C. Chan. Dynamic modeling for noise mapping in urban areas. *Environmental Impact Assessment Review*, 97:106864, Nov. 2022. doi: 10.1016/j.eiar.2022.106864 2
- [64] E. Twardzik, K. Duchowny, A. Gallagher, N. Alexander, D. Strasburg, N. Colabianchi, and P. Clarke. What features of the built environment matter most for mobility? Using wearable sensors to capture real-time outdoor environment demand on gait performance. *Gait & Posture*, 68:437–442, Feb. 2019. doi: 10.1016/j.gaitpost.2018.12.028 8
- [65] M. Vuckovic, J. Schmidt, T. Ortner, and D. Cornel. Combining 2D and 3D visualization with visual analytics in the environmental domain. *Information*, 13(1):7, Dec. 2021. doi: 10.3390/info13010007 3
- [66] P. Wang, Z. Liu, and L. Zhang. Sustainability of compact cities: A review of inter-building effect on building energy and solar energy use. *Sustainable Cities and Society*, 72:103035, 2021. doi: 10.1016/j.scs.2021.103035 7
- [67] S. Wang, F. Lyu, S. Wang, C. Catlett, A. Padmanabhan, and K. Soltani. Integrating CyberGIS and urban sensing for reproducible streaming analytics. In *Urban Informatics*, pp. 663–681. Springer Singapore, 2021. doi: 10.1007/978-981-15-8983-6_36 3
- [68] K. Wongsuphasawat, Z. Qu, D. Moritz, R. Chang, F. Ouk, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager 2: Augmenting visual analysis with partial view specifications. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pp. 2648–2659. Association for Computing Machinery, May 2017. doi: 10.1145/3025453.3025768 3
- [69] W. Yap, P. Janssen, and F. Biljecki. Free and open source urbanism: Software for urban planning practice. *Computers, Environment and Urban Systems*, 96:101825, 2022. doi: 10.1016/j.compenvurbsys.2022.101825 1
- [70] W. Zeng and Y. Ye. VitalVizor: A visual analytics system for studying urban vitality. *IEEE Computer Graphics and Applications*, 38(5):38–53, Sept. 2018. doi: 10.1109/MCG.2018.053491730 2
- [71] Y. Zheng, W. Wu, Y. Chen, H. Qu, and L. Ni. Visual analytics in urban computing: An overview. *IEEE Transactions on Big Data*, 2(3):276–296, Sept. 2016. doi: 10.1109/TBDATA.2016.2586447 1, 2, 3
- [72] L. Zhicheng and J. Stasko. Mental models, visual reasoning and interaction in information visualization: A top-down perspective. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):999–1008, Nov. 2010. doi: 10.1109/TVCG.2010.177 3
- [73] P. Ziegler and S. E. Chasins. A need-finding study with users of geospatial data. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, pp. 1–16. Association for Computing Machinery, 2023. doi: 10.1145/3544548.3581370 9
- [74] J. Zong, J. Pollock, D. Wootton, and A. Satyanarayan. Animated Vega-Lite: Unifying animation with a grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):149–159, Jan. 2023. doi: 10.1109/TVCG.2022.3209369 2, 3