A NEW GEOSPATIAL DATABASE FRAMEWORK

Advanced GIS Integration and Application-Development Tools for Bridging Geospatial and Attribute Data

Smart Buildings and Cities • *Mapping Software* • *Energy Infrastructure* • *Civil Engineering* • *Real Estate* • *Urban Design* • *Panoramic Photography*

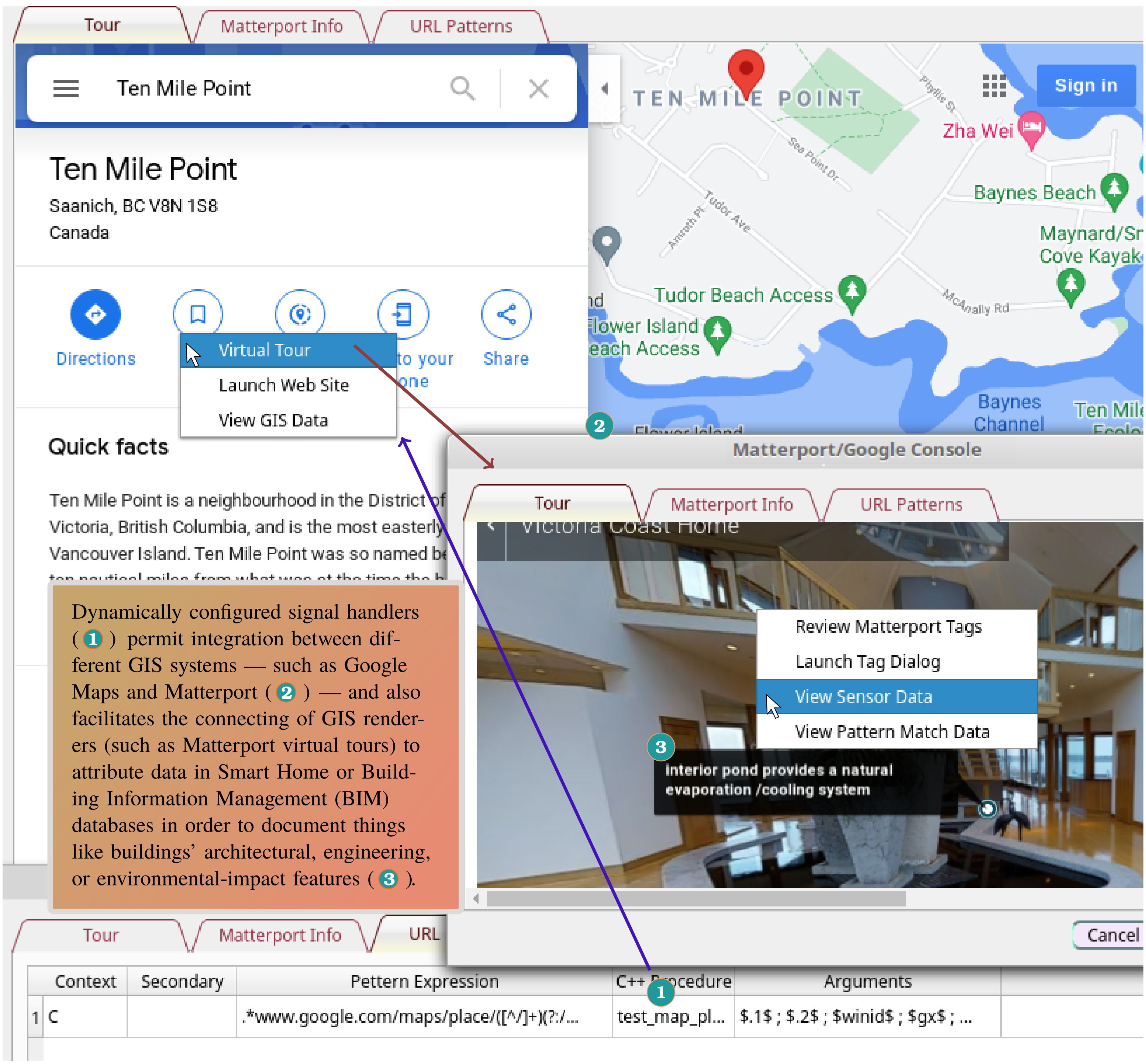

Figure 1: Using a Signal-Handling Framework to Launch a Virtual Tour from Google Maps

Product Description

Whereas traditional geospatial technology prioritizes a browser-based ecosystem, LTS has designed a new geospatial computing framework which is optimized for native desktop-style applications in place of web applications. Our framework offers novel application-development tools focused on seamlessly integrating host applications with embedded **GUI** components — allowing maps and virtual tours to be accessed *directly* within host applications, rather than through a web browser, so that visitors to maps and virtual tours can benefit from the full features of desktop-style software (such as dialog boxes, context menus, tooltips, and multiple-window displays) while engaging with maps and virtual tours.

By embedding geospatial displays in native software this opens up new possibilities for geospatial technology, allowing for fluid, responsive **GUI** design to benefit architects, engineers, and everyday users.

To maximally accommodate how users interact with geospatial applications (such as Google Maps, Matterport, and Open Street Map), our framework introduces the following: (1) novel representations for geospatial data; (2) novel protocols for *attribute* data which provides information about buildings and locations visible on geospatial displays such as maps or virtual tours; and (3) a new way of describing user actions in the context of software components where maps and/or tours are rendered.

Our New Geospatial Signaling Protocol

Using existing technology, it is difficult for applications to obtain information about user actions so as to properly respond to them. This is because as users interact with maps or virtual tours, they will often request actions that are specific to geospatial data and, therefore, require a special vocabulary distinct from other user-interaction models.[1] Although data about users' actions can be extracted from **URL**s (for example, by examining how Google Maps encodes location data in **URL** strings) as well as by registering callback procedures with application signals (such as navigation and context-menu requests), neither web **URL**s nor application signal protocols were designed for **GIS**, making it difficult to extract the relevant data needed to respond to users' requests.

Our framework addresses this problem by providing a signal layer and runtime-reflection mechanism which abstracts from low-level **URL** encodings and application signals, generating signals according to a new protocol which combines features of native reactive programming and of web resource models. Applications can then respond to our framework's signals instead of having to process low-level application events. Developers can use our framework to register procedures which will be executed in response to user actions, with the runtime-reflection system ensuring that proper functions are called with all necessary data.

In order to correctly supply data to geospatial event-handlers, our framework provides a new representation system for geospatial data which is configured to work natively with the framework's signals and runtime-reflection. This new data model uses graph-database technology to integrate data structures associated with geospatial values and events (maps/**GIS** coordinates, locations and regions, movement vectors, and user-action data) and those associated with objects or features situated at or within geospatial locations/regions: street locations, urban environment features, Building Information Management (**BIM**) objects, **IoT** devices, **IoT** and **GIS** digital twins, and so forth.

[1] Such actions include ♦ zooming in or out on a map/tour; ♦ clicking or requesting a context menu for tags or popups made visible in the proximity of a corresponding geospatial landmark; ♦ scrolling a map or satellite image in a specific geographical direction; ♦ selecting locations of interest via mouse actions; ♦ right- or left-clicking links inside text areas that provide information about selected locations; ♦ manipulating path overlays for distances or routes between destinations; and ♦ advancing along coordinate vectors in paroramic-photography or street-view tours.

Bridging Geospatial and Non-Spatial/Attribute Data

In view of the fact that a geospatial information system must be flexible enough to work with so many different varieties of non-spatial/attribute data — which can include Cyber-Physical reports from sensors measuring temperature or humidity to commercial site information, such as the names and street address of business properties, to geoengineering and civic-utilities data, such as energy sources, evacuation routes, or security-camera locations, and so forth — we have developed a variety of tools, as part of this overall framework, for integrating geospatial data with general-purpose databases.[2]

All geospatial informatics requires that designations of geographic and/or spatial points and regions be paired with non-spatial attribute data representing objects or features which are located at, around, or inside a given point/region. Our framework supports attribute data by prioritizing library-based data-integration methods, where non-spatial data is exposed to a **GIS** application via code libraries that supply a procedural interface through which this external data can be queried.

In fact, attribute data can derive from many domains, including geography, energy, hydrology, biodiversity/ecosystem research, urban design, civil engineering, demographics, epidemiology, and real estate. In order to create a bridge between geospatial and external non-spatial/attribute data from such variegated sources, our framework employs a novel property-hypergraph data model which incorporates features of both property graph and hypergraph database engines. This data-representation paradigm is flexible enough for both geospatial and non-spatial data to be represented via a common protocol but, nevertheless, expressive enough to identify data structures with enough detail as necessary for the purpose of aligning procedures between geospatial and non-spatial application components.[3]

Integrating Smart Buildings, Smart Cities, Sensor Data, and Digital Twins

Our framework provides special tools for working with code libraries dedicated to architecture, civil engineering, energy grids, sensor networks, and other domains which rely on computations where geographical coordinates and/or coordinate systems (such as direction vectors) are needed, which demonstrates the integration of geospatial and non-spatial/attribute data: allowing data and simulations of industrial parts and/or building materials and architectural designs to be integrated with maps and virtual tours as **GIS** attribute data and/or Panoramic Photography annotations.[4]

Because our framework is designed for native software it can bridge, via a common protocol, geospatial and non-spatial/attribute data, thereby eliminating the need for complex data-translations between web-based rendering engines (built around **HTML**/JavaScript resources) and engineering/Cyber-Physical code libraries (which are usually **C** or **C++** based).[5] In short, our framework enables software developers to uniquely integrate **GIS** front-ends (such as maps and street views) with analytics and visualization for individual locations on a geospatial grid, such as virtual tours of buildings or industrial sites (such as seen in Figure 1).

[2] This data model's central premise is to employ data structures similar to other spatial-programming contexts such as Image Annotations (as in the **AIM** Annotation and Image Markup system for bioinformatics) and Computational Geometry (e.g. kernel objects and morphology operators). Our framework describes geospatial locations, coordinates, values (such as distances), and regions using image and geometry-based representations which allow for general-purpose algorithms (not only those specific to **GIS** data) to be integrated with **GIS** code.

[3] More detailed data structures can then be generated via a procedural interface where a **GIS** component invokes non-spatial procedures specific to the domain of data points currently being situated at **GIS** locations. Our framework provides tools to manage the procedural workflows across distinct components (e.g. managing deferred responses, input validation, response decoding, and related remote-computing tasks, which are relevant to the **GIS**/non-spatial interoperation even if the requested procedures are not technically remotely-executed).

[4] Typical analyses programmed for such domains include validating the spatial arrangement of sensors; estimating the fault-tolerance of buildings and bridges given seismic and wind conditions; calculating buildings' carbon footprint by modeling heating and cooling requirements (which is affected by construction materials, placement and construction/type of windows, interior airflow, ventilation system, and so forth); or predicting the longevity of industrial parts which are exposed to atmospheric conditions.

[5] A discussion of hypergraph data modeling for Cyber-Physical systems, analyzing the software engineering benefits of the data models and programming paradigms employed by the framework outlined in the current document, can be found in Neustein (ed.) *Advances in Ubiquitous Computing: Cyber-Physical Systems, Smart Cities and Ecological Monitoring* (Elsevier, 2020) https://www.elsevier.com/books/advances-in-ubiquitous-computing/neustein/978-0-12-816801-1

Use Case: Next-Generation Integration of Virtual Tours and Street Maps with Attribute Data to Make Buildings Safe and Eco-Friendly

Given that panoramic photography is closely integrated with **GIS** map rendering within virtual "street view" tours (because locations visible in these virtual tours can be georeferenced and thus associated with coordinates on maps that are visible to the user) street view **GUI**s would need the capability to switch back-and-forth between **3D** tours and **2D** maps. Moreover, **GIS** data can be relevant for Panoramic Photography applied to indoor sites or fixed outdoor locations in addition to its use-cases for navigable street scenes.[6]

For example, architectural, engineering, or sensor data in formats such as **BIM**, **CoAP**, or **MQTT** can be rendered within Matterport tours via tags and embedded hyperlinks. However, a well-functioning integration of maps and virtual tours is only possible with the use of a signal-processing engine such as that provided by our framework, because detailed signal-handling is needed to connect signals originating from the map-based and tour-based parts of the integrated application.

The screenshots on Figure 1 furnish one example of this integration, where a dialog box hosting a Googe Maps view is programmed to identify sites having a Matterport virtual tour, rendering that tour in a dialog box of its own. Our framework intercepts signals within the Google Maps view to identify the relevant Matterport location, providing a more convenient and user-friendly bridge than the existing Google-Mapperport integration, which at present is only available for a small number of locations and in general is rather difficult for the end-user to follow.[7]

By installing signal-handlers for Matterport virtual tours, our framework would enable a Matterport engine to supply information about a building's architectural, engineering, safety, and environmental-impact features. For example, many contemporary "smart" homes or office spaces have specially-designed fixtures/equipment engineered to reduce a building's carbon footprint, and they may employ specific eco-friendly architectural and design choices to do so. Such features could on a Matterport tour be readily highlighted by tagging locations of eco-friendly equipment/actuators or, similarly, by identifying sensors installed to measure such factors as temperature, humidity, and indoor wind speed, which document the building's energy-efficiency.

For instance, a building which maintains comfortable temperatures while reducing cooling/heating energy consumption is more likely to achieve carbon-neutral status, critical attribute data which when effectively integrated with maps and virtual tours can prove very useful to engineers and urban planners. As demonstrated in Figure 1 (see the Matterport tag at 3), intercepting Matterport signals allows the application which embeds a virtual tour engine to connect Matterport tags on virtual tours with architectural, environmental, or Cyber-Physical databases tracking a building's design features and energy consumption. Thus, native design which enables this high-level integration of attribute data with **GIS** heralds the future of smart building design, clean energy, and smart cities with much greater efficiency.

Linguistic Technology Systems (LTS)
Amy Neustein, Ph.D.,
CEO and Founder
Phone: (917-817-2184)
Email: amy.neustein@verizon.net

[6] Geographic direction, sun exposure, wind patterns, topography, and hydrological/seismic conditions may all be factors influencing designs and engineering of buildings and installations, yielding data visualized by incorporating summaries or links into panoramic engines or virtual tours.

[7] Some businesses have recently employed a Matterport feature in their Google Maps listings which allow users to launch virtual visits to their locations while viewing information about their business on Google Maps, but this design forces users to select a Matterport tour among many other options via icons which are often invisible without being scrolled. In short, the existing Google-Matterport integration can be significantly improved by embedding both a Google Maps renderer and a Matterport virtual tour engine, each in their own respective native-application window and implementing signal-handlers so that the two windows may interoperate.