A New City Map

David Chen davidchen14855@gmail.com

Hongyang Yang hyang981@student.ubc.ca

Madison Lore mlore@student.ubc.ca

Niels Semb Nielssolesemb@gmail.com

1. Introduction

Maps have been made for thousands of years. The makers, the cartographers had fascinated people with their making and the usage has been plenty. As much as it is science, it is art. It depends a lot on who is observing, but even more on how the cartographers are framing it. As Marcel Duchamp so very nicely put it, “Anything can be art, but everything is not already art. What makes something art is the act of framing it as art”.

The maps have historically often given a distorted view since the maps present the world through the lenses of the cartographer. It could be their lack of knowledge, his or her biases, needs for the audience or how they felt about their neighborhoods.

The integration of non-spatial data with the base spatial data has brought along a new branch of maps. They are now not just conveying geographical information, but also some other piece of data. In cartography literature, it is known as thematic cartography [6].

Street maps give us a picture of the day to day flow we humans have through the landscape. The streets differ from wide to narrow, short to long, straight to curved. Bind together with intersections we get a network as we know as street maps, and they can indeed be an ingredient to art. It’s like we are living in an art world.

In this paper, we want to look at how street maps can be used for making abstract art. Could an end user utilize this tool to make art out of their own neighborhood? What kind of non-spatial data can be encoded into the street map to enrich the street map?

2. Related Work

Today there are many different cartographers who love to make maps with non-spatial data. One of them is the graphical designer Paula Shcer [1]. She is making maps with lots of different data visualizations. One is a map of the US that includes counties and zip codes. Illustrates how some little counties have many zip codes, but big counties have few. Another is a demographic map that shows the age of people and racial and ethnic background (fig. 1). The painting for Paula is not the most practical way to encode the information, but most artful and expressive.

Figure 1. The US with states, and zipcodes

The next project is not per se art, and isn't made for it either, but shows one of the ways OSMNX [2] has been utilized. The road curvature map [10] is made by a curvature algorithm. It determines the curviness of the roads and encodes it with a multi-color gradient as shown underneath.

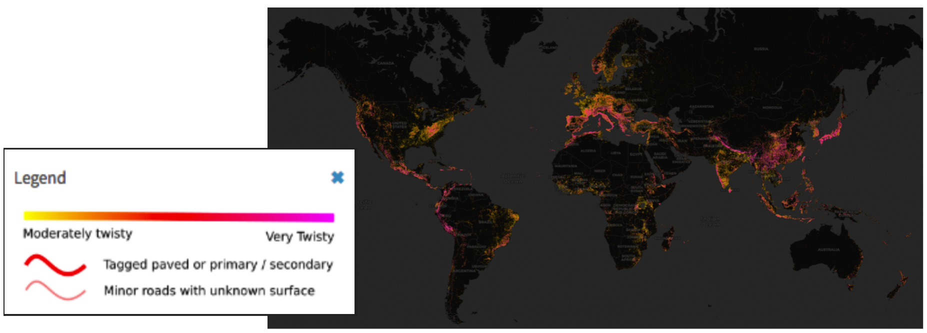

Figure 2. Road Curvature Map

3. Data and Task Abstraction

3.1 Datasets and Data Abstraction

Primarily, the dataset used is derived from OpenStreetMaps (OSM) [7]. In OSM, any combination or all of the roads, walkways, bike paths, or trails can be returned for a given area. The data are thus divisible for driving, cycling, and/or walking. The structure is that of a graph, where, for example, in the driving network, the edges represent roads and the nodes represent connection points or intersections between roads. In the case where an intersection is a roundabout or holds some structure, that intersection then consists of edges with nodes connected to each of the incoming roads. In the case where roads are separated in direction by a median of reasonable size, each direction is represented as an edge. In most cases, a road with lanes going either in one or two directions without a median is represented as one edge.

Each edge holds attributes that describe the road. These attributes are shown in Table 1. Because OSM is curated by the open-source community, not every attribute is available for each edge. All edges have at minimum: the origin node and destination node. Each node holds attributes that describe the connection or intersection. These attributes are shown in Table 2. In this case, these attributes are always present.

Attribute	Attribute Type
origin_node	int, ordinal
destination_node	int, ordinal
oneway	bool, categorical
lanes	int, quantitative
name	str, categorical
highway	str, categorical
maxspeed	int, quantitative
length	int, quantitative
geometry	LineString

Table 1. Edge attributes presented in OSM data.

Attribute	Attribute Type
latitude	float, quantitative
longitude	float, quantitative
street_count	int, quantitative

Table 2. Node attributes presented in OSM data.

Additionally, OpenStreetMaps holds a large corpus of amenity points of interest. These amenities range in use from sustenance, to education, to healthcare, and more. A full list of amenity categories is listed in Table 3. Within each category, there are a number of subcategories, saved as a dictionary of category keys and more specific subcategory values. A representation of these subcategories is shown for the broader category of "Education" in Figure 3. Most of these amenities are represented as **point** or **area** marks.

All cities are different, but in the bounds of Vancouver, BC, Canada, there are 22,668 edges, 7,606 nodes, and 9,965 amenities. In a neighborhood within Vancouver, for example, Kitsilano, defined as 1km radius from a center point in the neighborhood, there are 522 edges, 163 nodes, and 613 amenities.

Amenities
Sustenance Education Transportation Financial Healthcare Entertainment, Arts & Culture Public Service Facilities Waste Management

Table 3. Amenities present in OSM data. All amenities are string categorical values.

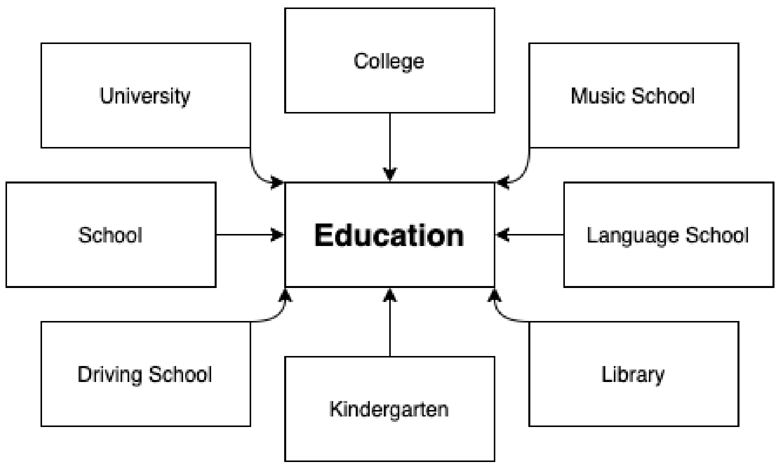

Figure 3. Subcategories of amenities for the “Education” category in OSM.

3.3 Tasks and Task Abstraction

An end user would use this visualization to create art from their neighborhood in order to foster a personal connection. Through this visualization tool, the end user would be able to define their home neighborhood and create abstract art pieces from the streets and intersections that comprise this place. The abstract art is created by separating the edges of the specified neighborhood and rearranging them randomly or according to predefined styles.

Outside of the domain of road networks, this visualization implementation could be extrapolated to any domain with line or shape structures that wish to allow their users to connect with the underlying dataset. At its core, the task follows the abstraction action definition of Analyze > Consume > Enjoy. An end-user may not seek out art created through the rearrangement of roads but could be satisfied by the creativity of the visualization and create art for many important locations in their life. Alternatively, the same visualization may follow the abstraction action definition of Query > Compare in which a user is converted from simply an enjoyer into someone interested in the comparison of art produced for different neighborhoods around the world.

4. Proposed Solution

4.1 Implementation

Our proposed implementation strategy for this visual interface is currently in the process of being defined. To enable users to interact with our visualization, we plan on creating a single page web application using the React framework leveraging Typescript as our language of choice. Typescript is effective in that it should allow us to keep our application type-safe and adherent to current mainstream application development best practices. On this application, we plan on using Mapbox[5] along with the react-map-gl[9] wrapper which helps to facilitate access to Mapbox functionality such as capturing user input and displaying the output art. Mapbox will also allow users to actively engage with a cartographical interface which should provide them with the context of what we are trying to achieve.

To access the OpenStreetMap data which serves as the basis of the artistic visualizations we are going to display, we are using OSMnx[2], a library developed to allow users to easily access and manipulate OpenStreetMap data. This library will allow us to fetch, curate and store the data we are going to require to generate artistic visualizations for each neighborhood. To complement this library, we are planning to use FastAPI[4] to serve as the data access layer. FastAPI will act to facilitate data exchange via REST endpoints to the UI layer of the visualization. Since FastAPI is Python-based, it should seamlessly integrate with OSMnx.

We are currently exploring multiple options when it comes to creating the artistic visualization of the neighborhood. Leveraging D3.js[3] to create the output artistic visualization and directly render it in-app by providing the library with our curated data is one option currently being considered. Another option would be to render visualization in the backend layer via Plotly[8] and deliver the output via image format to the user interface. This approach would separate the UI layer and the data processing layer completely and may result in better performance at the cost of additional overhead in formatting the output on the UI layer.

4.2 Scenario of Use

Our end user is looking for a gift for a friend. She knows that her friend loves art, so she is looking for something different and interesting. Our end user and her friend used to be roommates and loved walking around their neighborhood of Kitsilano, so she would love to find something that brings in this personal connection. She finds this visualization tool and creates art through the following steps:

1. The end user enters the address she would like to center her art around or places a pin to choose it on the map. She also can optionally choose the style of art for her neighborhood piece from a predetermined list of styles.
2. The system draws a 1km radius around this point to define the road edges and nodes that will be incorporated into the art piece.
3. Within the 1km radius, the areas of interest and unique characteristics of the neighborhood are retrieved by the system to influence the position of the edges as well as other channels such as color, width, or smoothness.
4. The edges and nodes are separated into distinct pieces and rearranged according to the chosen style, or randomly by default.

5. If random, the system assigns line weights, colors, and/or size augmentations to each edge along with a chosen starting location for one end of the edge. If in a particular style, the line weights, colors, and/or size augmentations are predetermined in that style.
6. The end user is able to download their art as an image, create a new piece from the same location, or create a new piece from a different location.

4.3 Illustrations

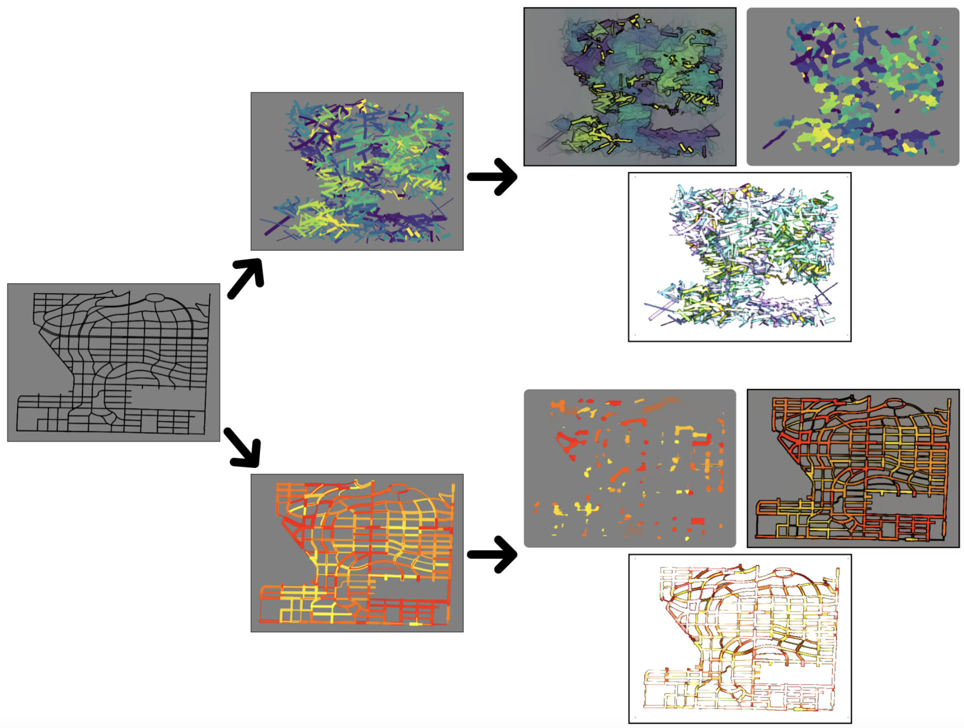

Figure 4. A first illustration draft of how a neighborhood layout may be transformed into art.

The above illustration (Figure 4) shows an example of creating art from only the road network of a user's neighborhood (excluding the points of interest). In this example, the neighborhood of Shaughnessy in Vancouver, BC, Canada is shown transformed in two ways. The top example shows a random rearrangement of the street layout and a distribution of color and line widths for each road. The bottom example shows the same neighborhood only with the distribution of colors and line widths, but the shape of the neighborhood is preserved. Each of these examples are then transformed into different styles which could be determined by the user's preference for art style.

5. Milestones

We plan to spend about 320 hours together towards the project. Table 4. provides a rough estimate of the project's tasks. Application development will be handled by Hongyang, David, and Niels. Data science work will be handled by Madison. Research, development, and literary responsibilities will be shared by all of the members of the group.

Task	Hours per person(Estimated)	Deadline	Description
Pitch	3	Sep 29	Raise ideas, search for the dataset, create slides, and rehearse pitch(ALL)
Pre-proposal meetings	4.5	Oct 13	Finalize group, get the dataset(Madison), brainstorm project topic and define the topic(All), meet with Tamara(All), update the topic(ALL)
Proposal	3	Oct 21	Write the project proposal(ALL)
Implementation	10	Ongoing	Learn selected tool(ALL)
	3.5	Oct 23	Clean and analyze data(Madison)
	6	Oct 25	Determine image generating methods(ALL)
	3.5	Oct 27	Finish the requirement analysis of the display layer project(Niels, David, Hongyang)
	10	Nov 3	Design and implement baseline single page application(Niels, Hongyang) Implementation of baseline data service layer Python backend (David) Process the data and image generation in Python (Madison) **Will be collaborative**
	6	Nov 7	Finish the first draft of the project(ALL)
Project Report Update	3	Nov 16	Update project report(ALL)
Peer Project Review	4	Nov 17	Prepare the project demo, review peer project, and get suggestions(ALL)
Refined implementation	10	Nov 28	Implement image generating methods(Python: Madison, React: Niels, Hongyang, David)
	3	Dec 10	Polish implementation(add features, improve feasibility, etc.)(Niels, David, Hongyang), prepare demo(Madison)
Final presentation	6	Dec 15	Prepare and rehearse presentation(ALL)
Final paper	8	Dec 17	Finalize paper(ALL)

Table 4. A rough estimate of the project's tasks

6. Bibliography

1. “A la Carte: Paula Scher’s American maps chart more than just territory”. Wallpaper*. Available: https://www.wallpaper.com/art/paula-schers-american-maps-chart-more-than-just-territory.[Accessed: 20-Oct-2021].
2. G. Boeing, “OSMNX: New methods for acquiring, constructing, analyzing, and Visualizing Complex Street Networks,” *Computers, Environment and Urban Systems*, vol. 65, pp. 126–139, 2017.
3. D3: Data-Driven Documents. Michael Bostock, Vadim Ogievetsky, Jeffrey Heer. IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis), 2011.
4. “FastAPI,” *FastAPI* [Online]. Available: https://fastapi.tiangolo.com/. [Accessed: 20-Oct-2021].
5. “Mapbox,” *Mapbox*. [Online]. Available: https://www.mapbox.com/. [Accessed: 20-Oct-2021].
6. Tamara Munzner. Visualization analysis and design. CRC press, 2014.
7. OpenStreetMap contributors. (2017). Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org .
8. “Plotly,” *Plotly* [Online]. Available: https://plotly.com/python/ . [Accessed: 20-Oct-2021].
9. “React-map-GL - github pages,” *React-Map-Gl*. [Online]. Available: https://visgl.github.io/react-map-gl/. [Accessed: 20-Oct-2021].
10. “Road Curvature Map”. Road Curvature. [Online]. Available: https://roadcurvature.com/2020/10/04/new-curvature-map/ [Accessed: 20-Oct-2021].