GROUP 007

OP2 | ECE297

AYAN ALI
JET CHIANG
REYAB SALUJA

Goal: Make Urban Environments Easier to Understand and Explore

What we've been working on

Users can View City in **2D**

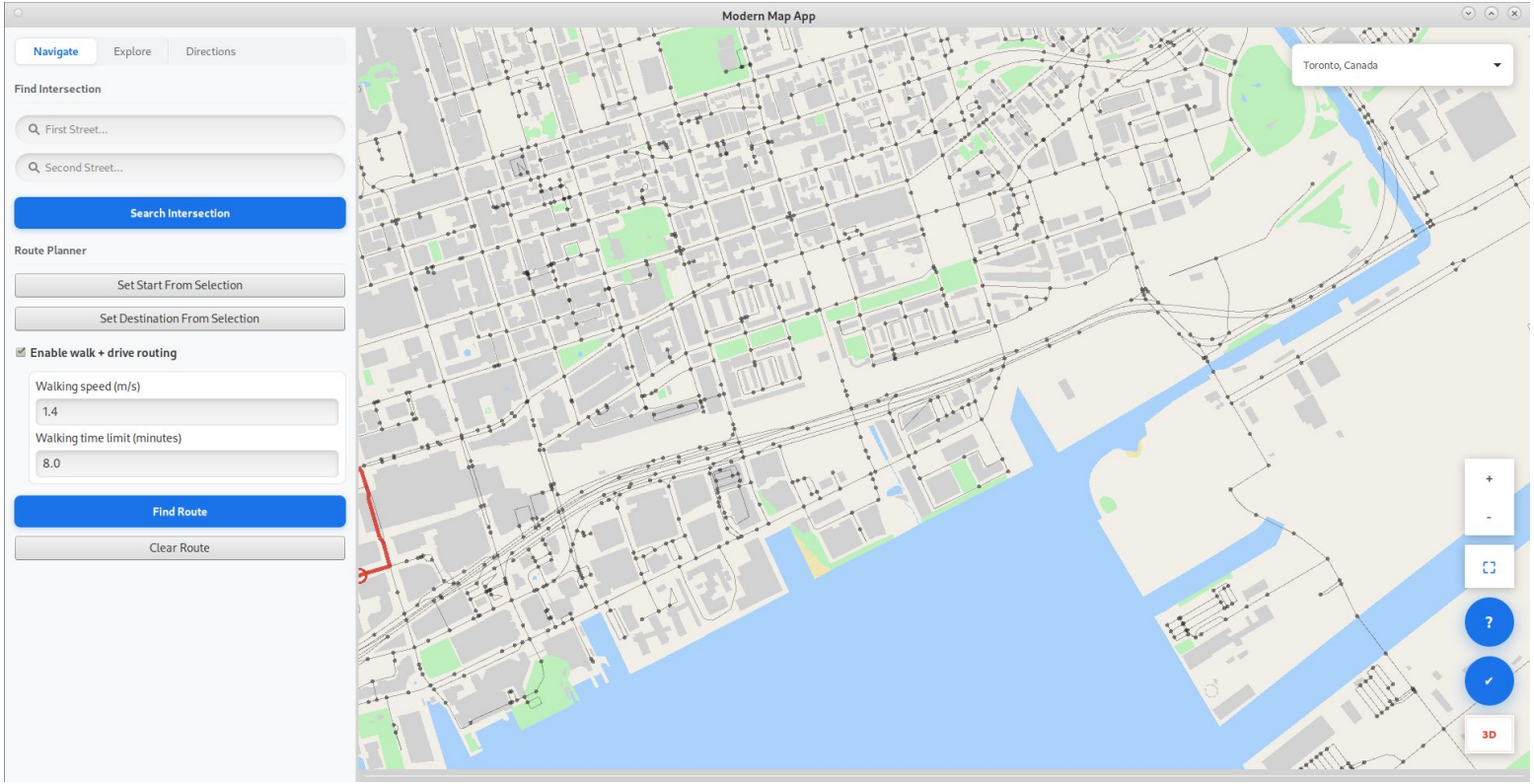


Figure 1: screenshot from ece297 application, shows the 2D map view.

Users can View City in **3D**

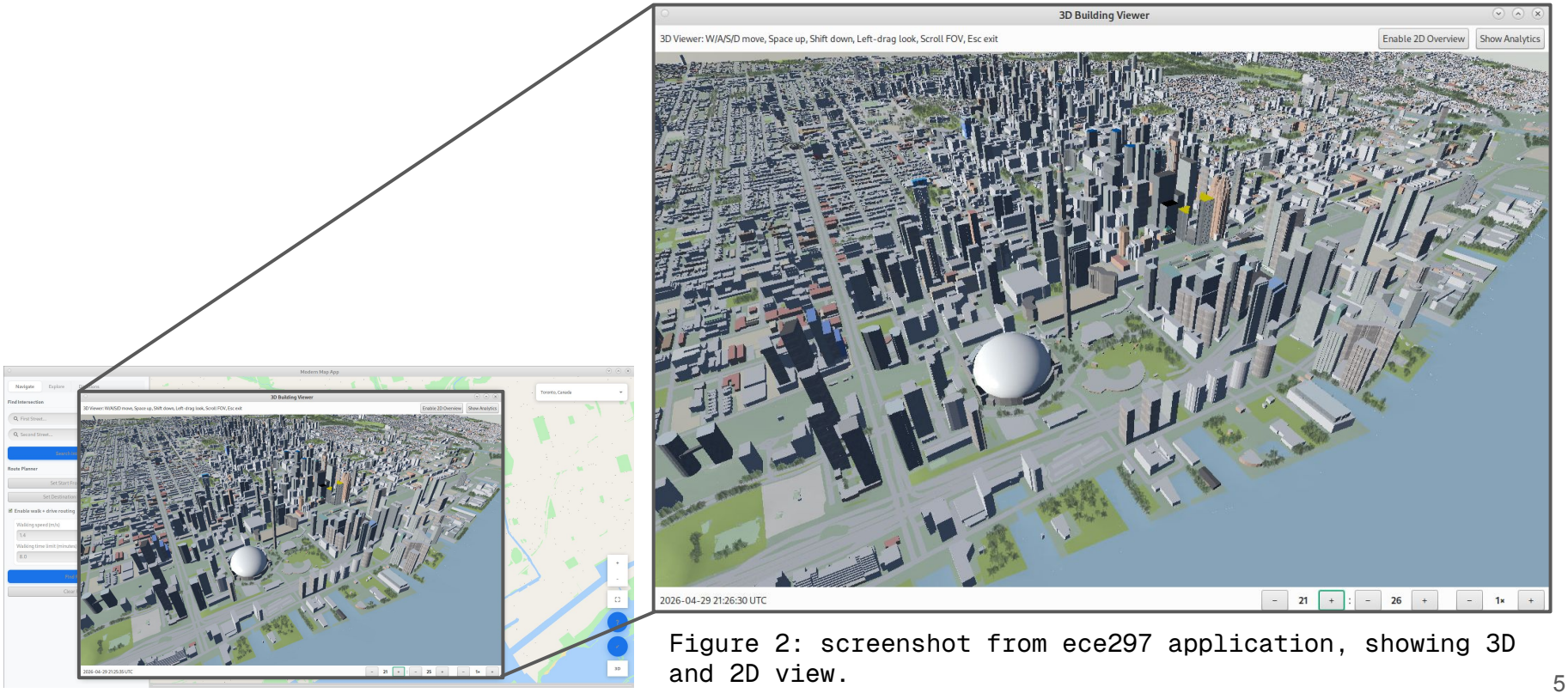


Figure 2: screenshot from ece297 application, showing 3D and 2D view.

Design Achievements

3D Viewer

2D Map
- Styling with CSS
- POI Navigator

Map Algorithms
- Intelligent Searching
- Courier Optimization

3D Viewer Highlights

1 Accurate Buildings, Roads, Areas
2 Terrain Generation
3 Landmarks & Complex Roofs
4 Time of Day
5 Performant Trees & Billboarding
6 Ground & Building Textures
7 Shader Design
8 Performance Considerations

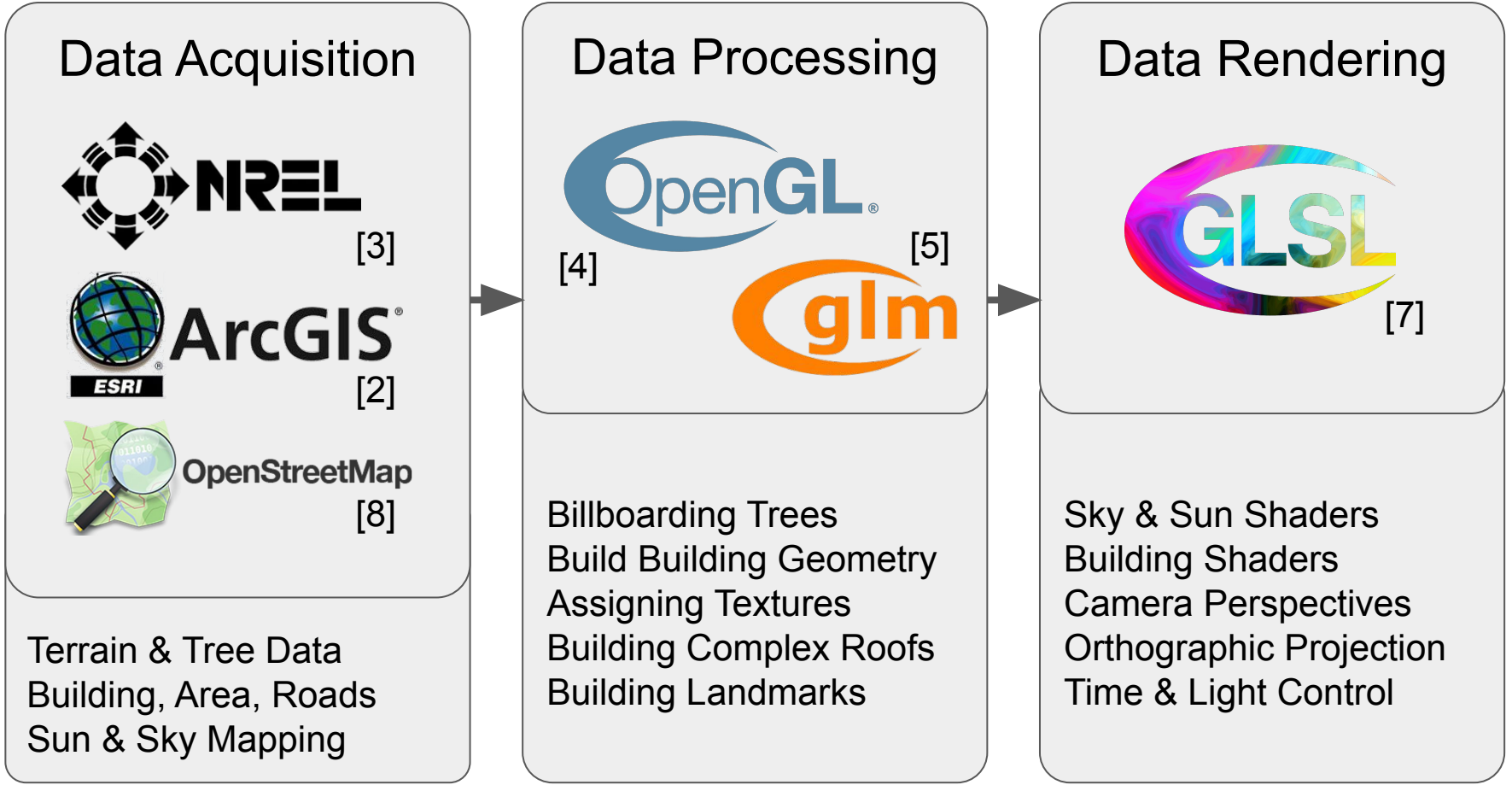
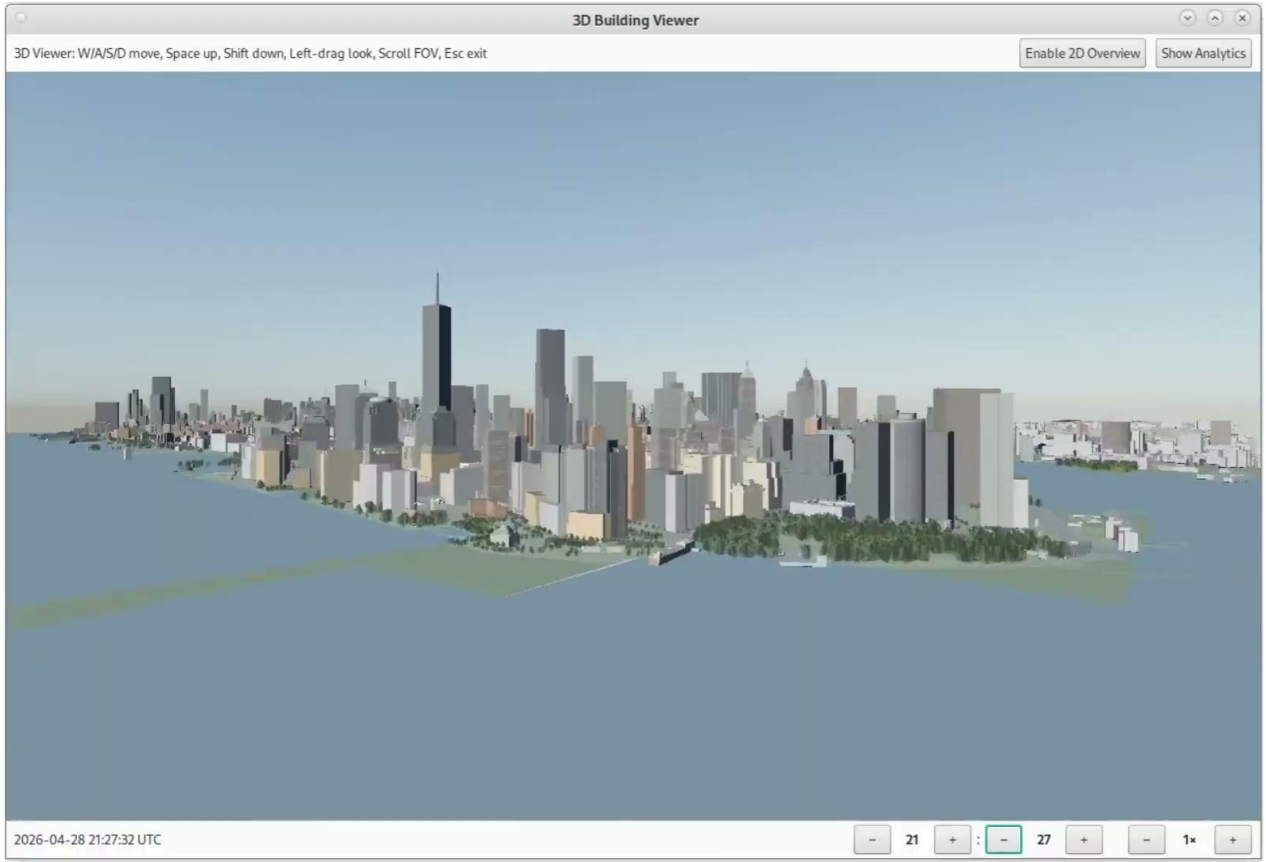
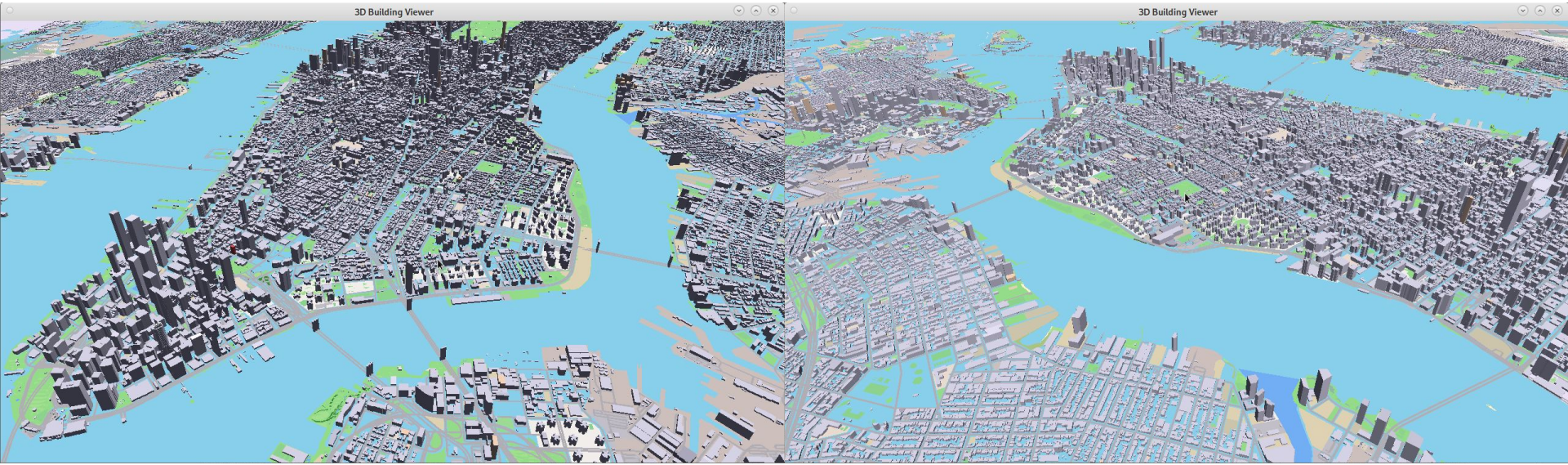


Figure 3: 3D rendering pipeline: terrain, data processing, and data rendering

1 Accurate Buildings, Roads, Areas



2 Terrain Generation

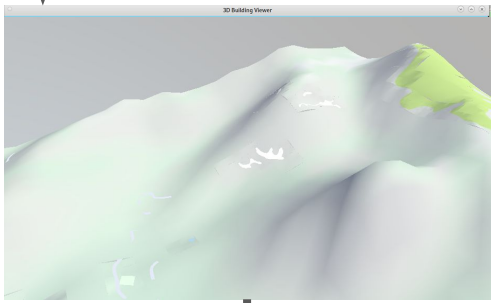


Buildings without terrain take away from our goal. We need a way of building terrain data whilst avoiding some common issues.

ArcGIS Elevation Data

ArcGIS [2]

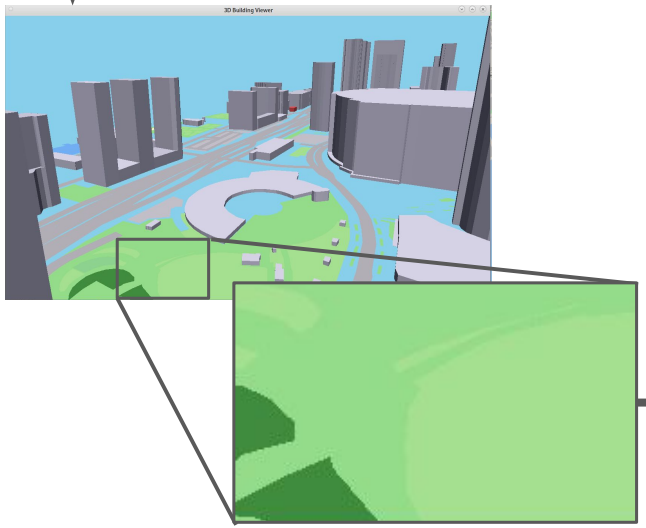
Per-vertex terrain data



OSM Area Data

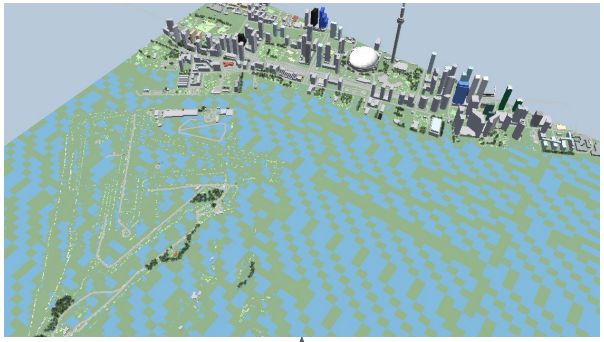
OpenStreetMap [8]

Quad-based Polygon data for areas such as parks



* this process has quite abig problem as areas and terrain meshes conflict with each other causing “z-fighting”

Fill the GPU Memory for Graphics Processing*



OSM Area Edge Data

OpenStreetMap

[8]

OSM Area Mesh Data

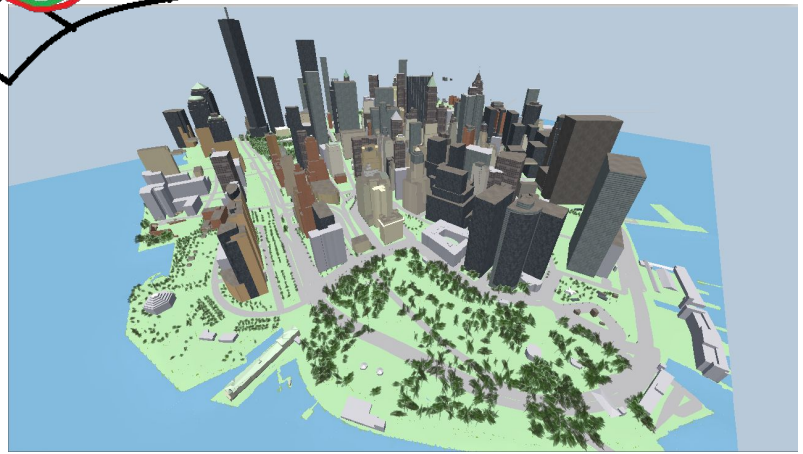
interpose the terrain that has area-shaped holes with original areas

ArcGIS Elevation Data

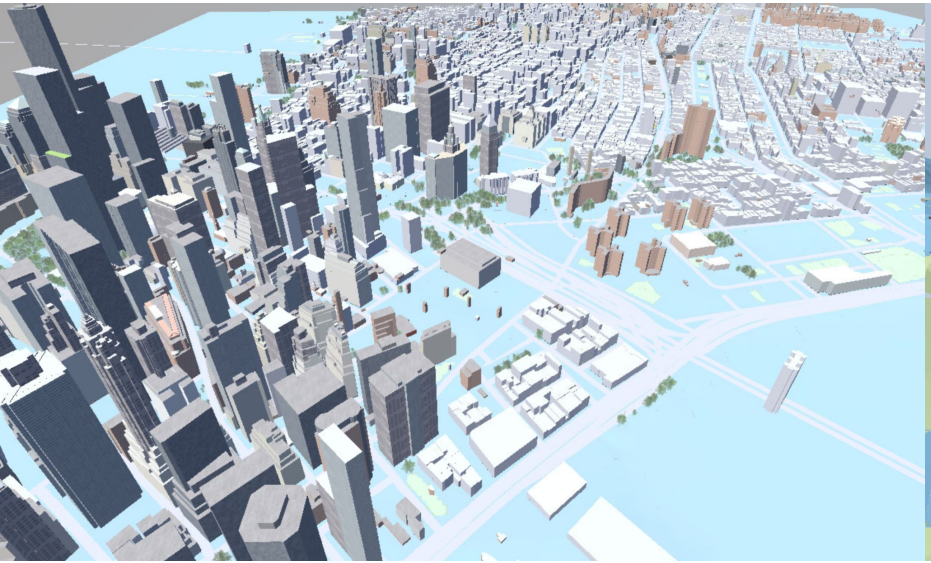
ArcGIS ESRI

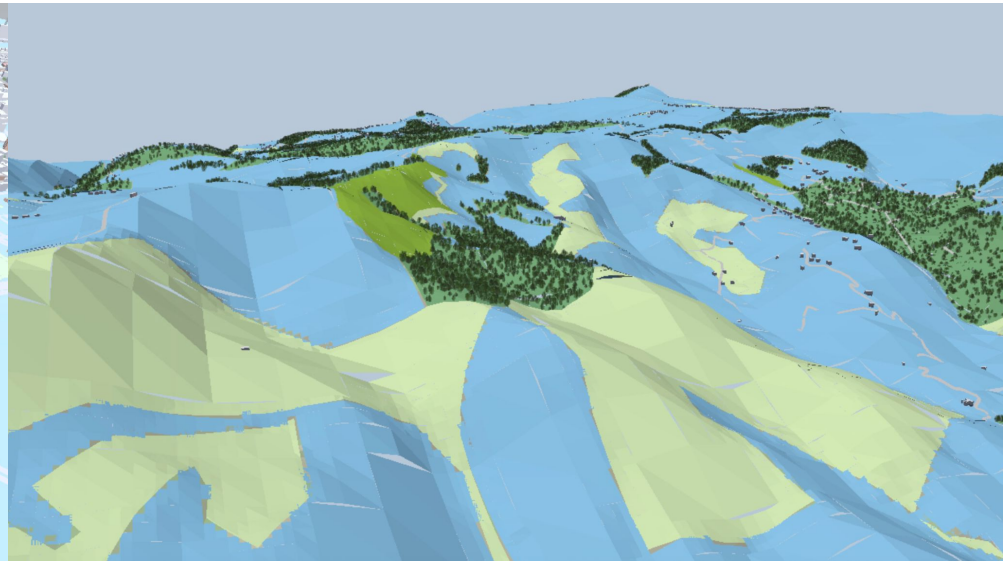
[2]

OSM Area Edges cut the Elevation Mesh

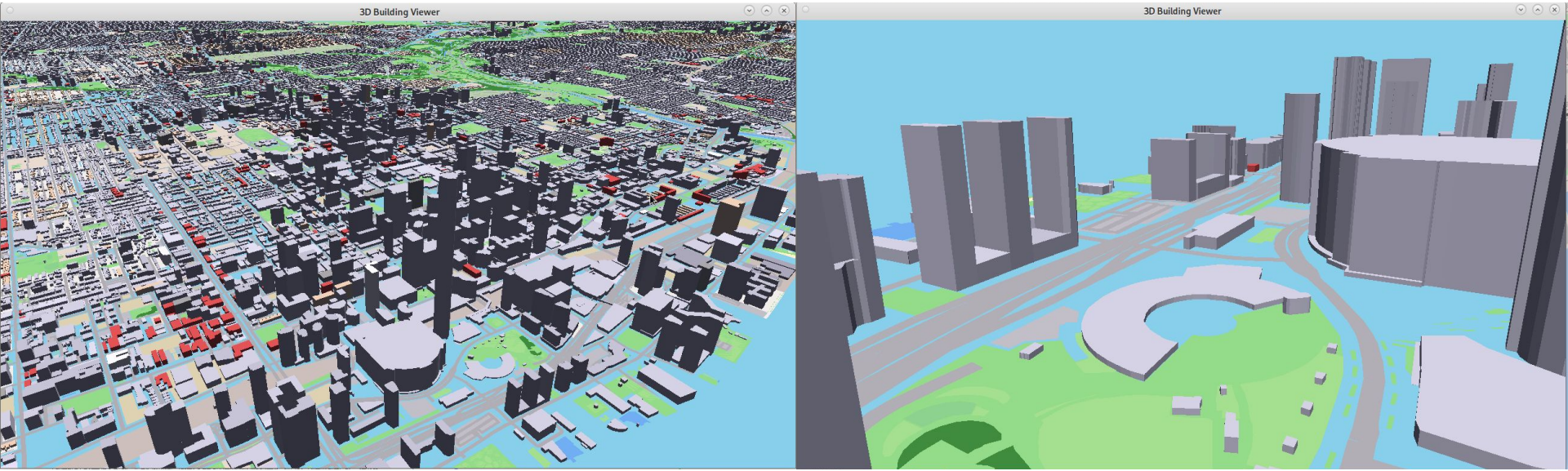


After solving the problem of z-fighting via meshcutting, we were left with the problem of water cutting into land. Here careful rules around OSM areas left a final mesh that was well-suited to pedestrian walking.

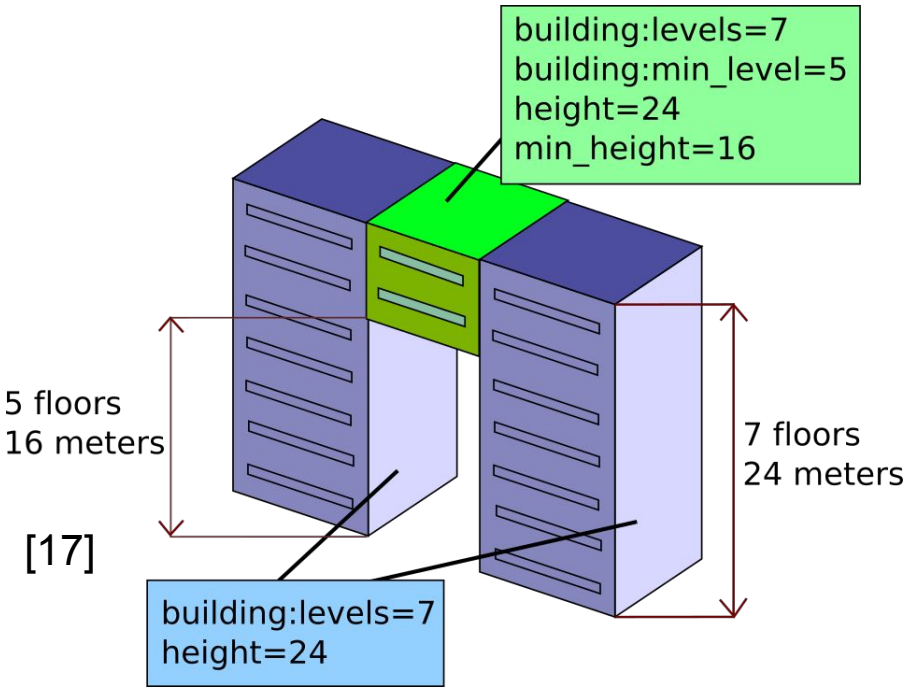




3 Landmarks & Complex Roofs



Landmarks help the the urban explorer find their bearings and find viewing angles not accessible in 2D.



OSM Represents buildings via the use of parts, these parts can then be combined to form complex shapes such as landmarks.

[5]

OSM Building Data

OpenStreetMap

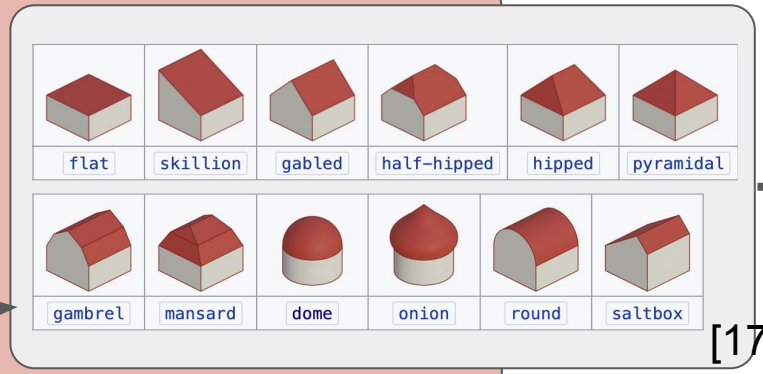
[8]

Roof Type

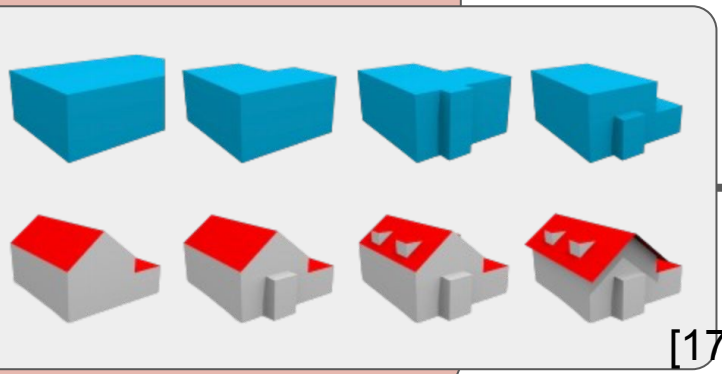
Camera Position in Data Representation

Level of Detail Value

Building Buildings from Parts



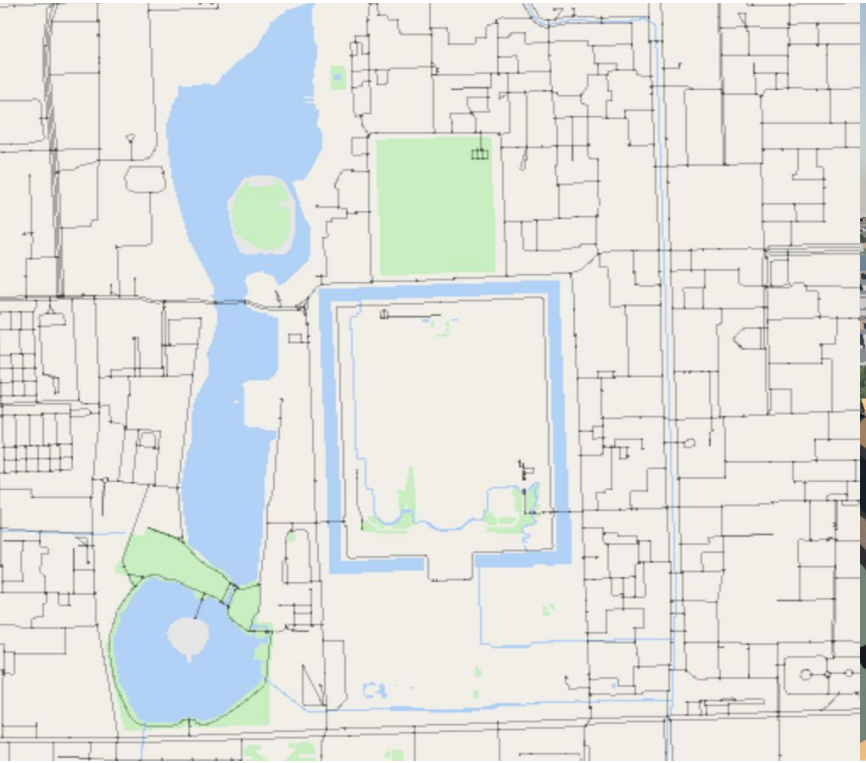
[17]



[17]

Building Mesh Data from initial building parsing

This yields landmarks that help users identify regions of interest and views that cannot be seen in 2D maps alone.



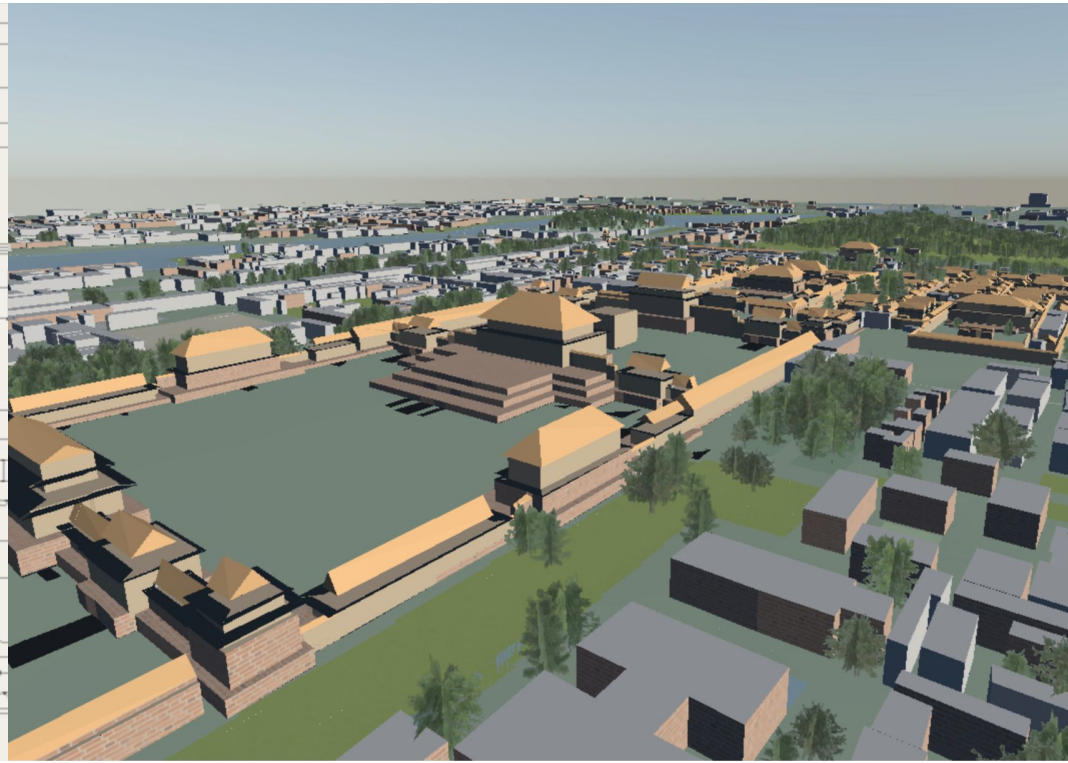


Figure 4: screenshot from ece297 application, showing Forbidden City, Beijing

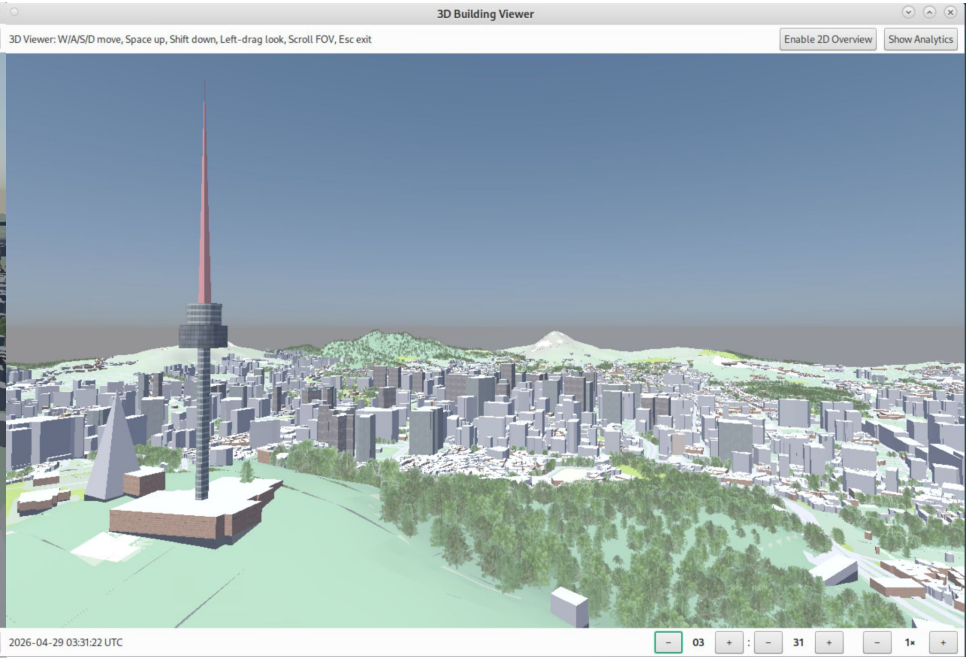


Figure 5: screenshot from ece297 application, showing Seoul Tower and Namsan Park.

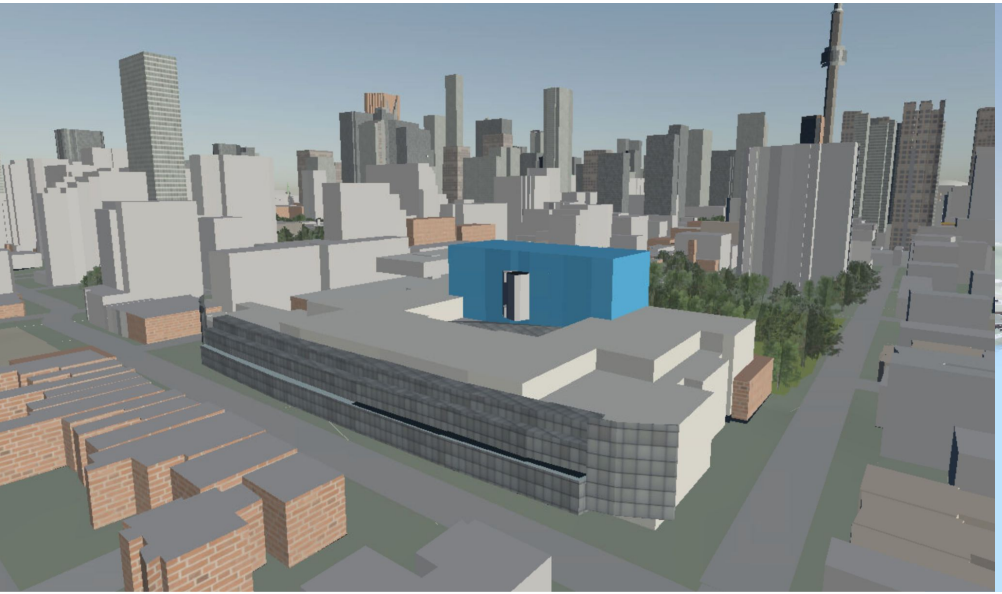


Figure 6: screenshot from ece297 application, showing Art Gallery of Ontario, Toronto.

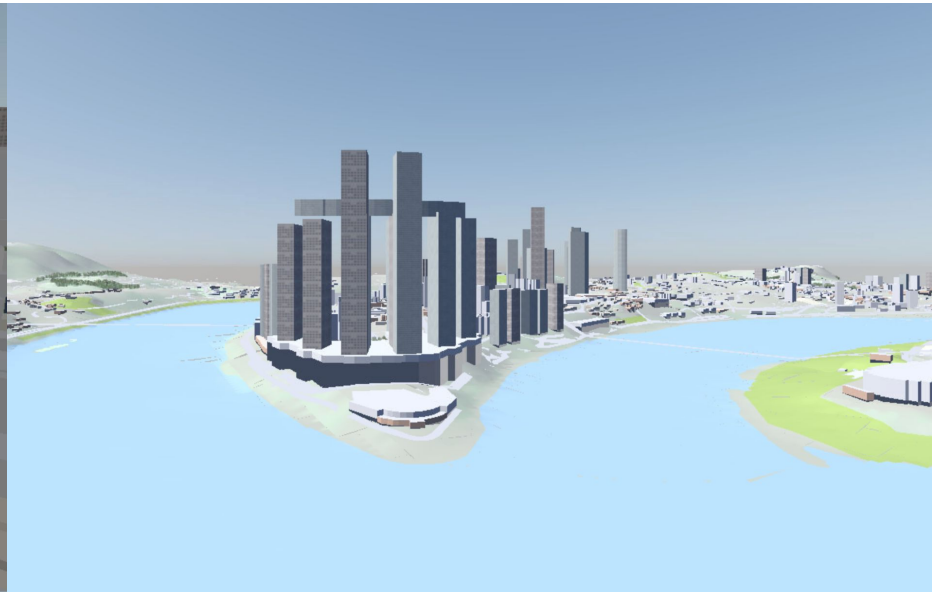
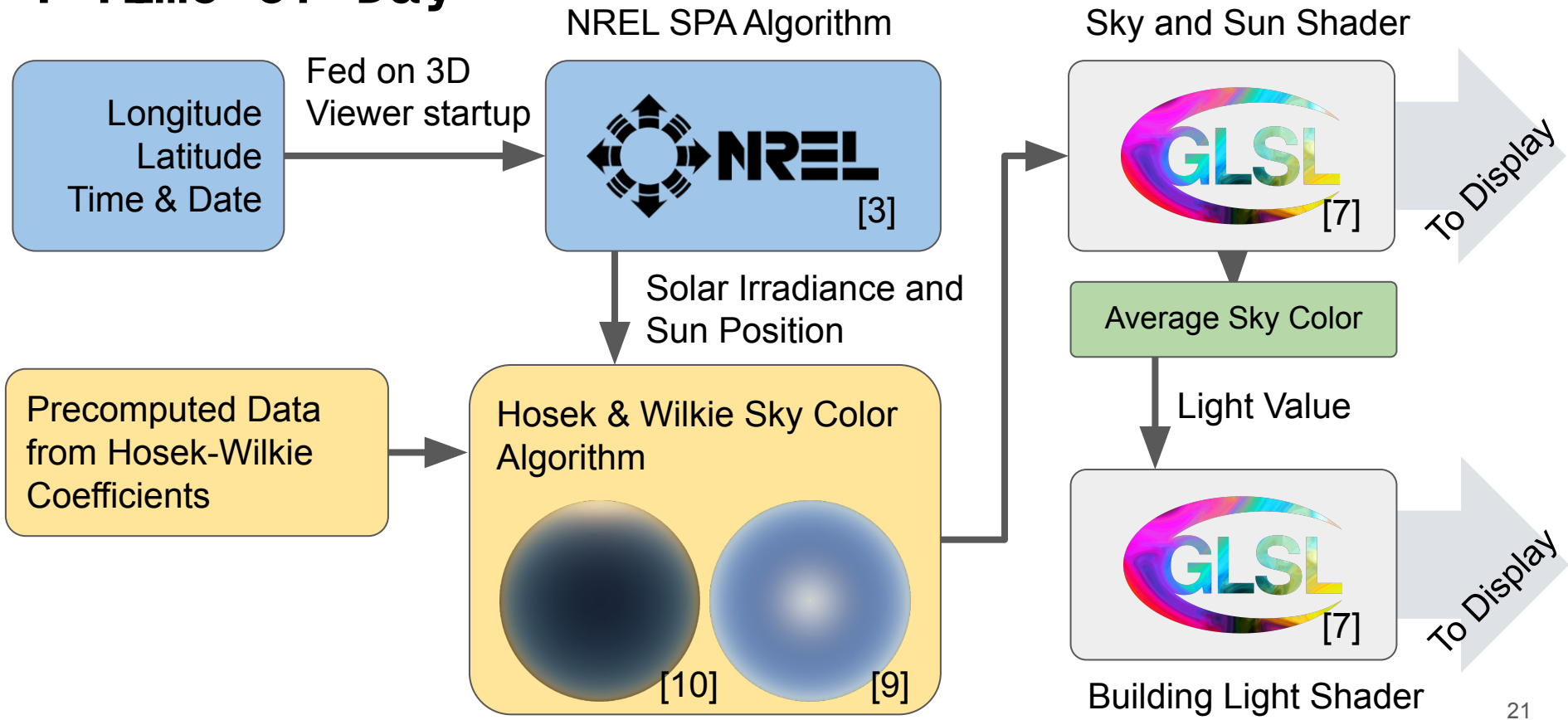
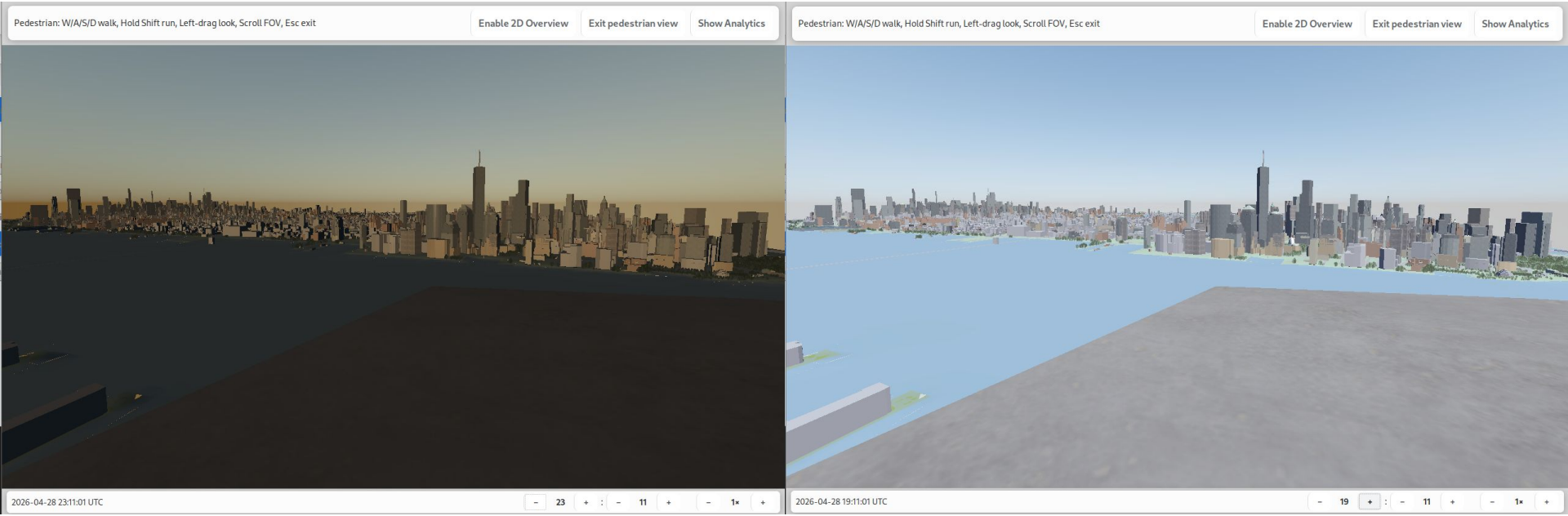


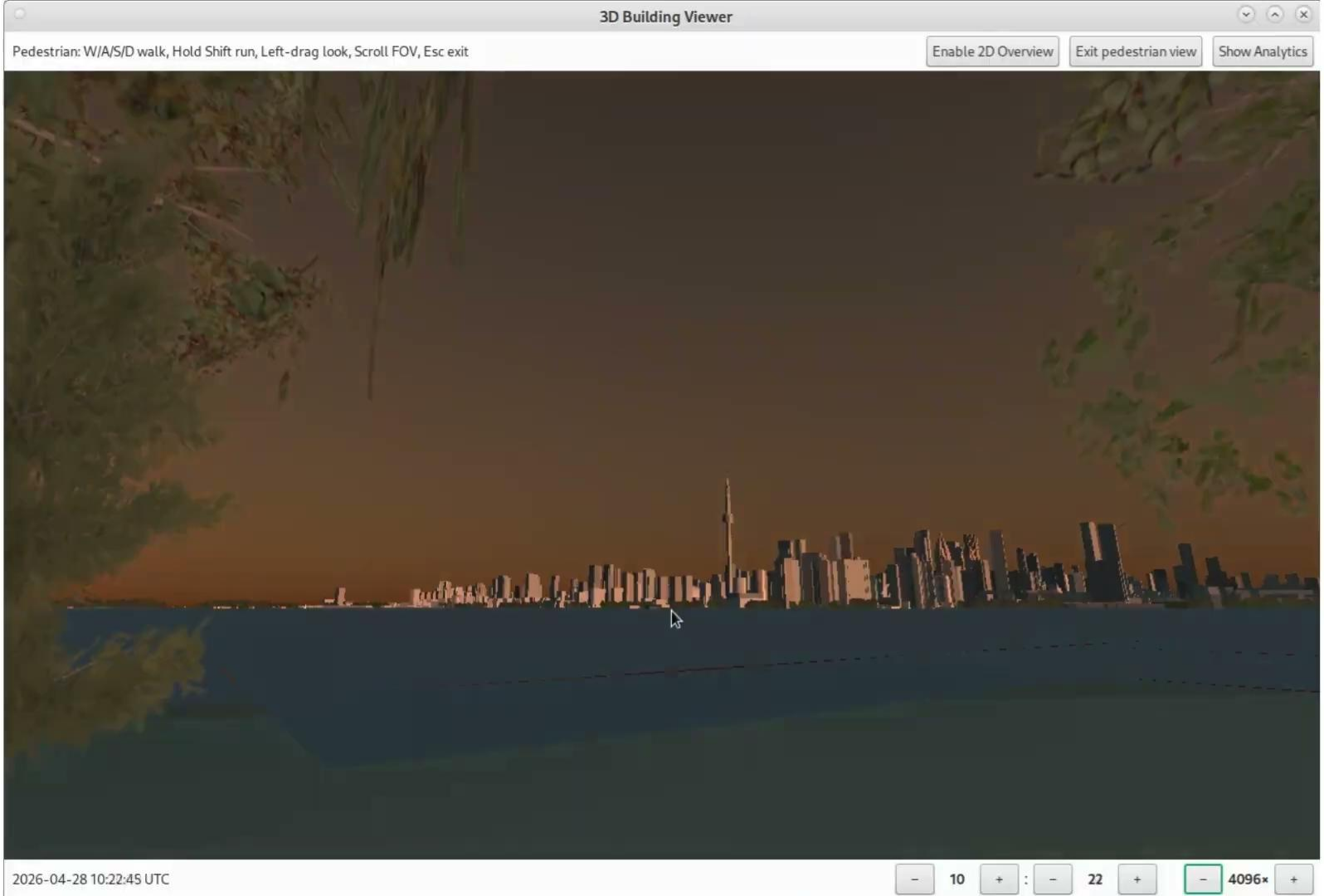
Figure 7: screenshot from ece297 application, showing Chaotianmen Sqaure, Chongqing.

4 Time of Day

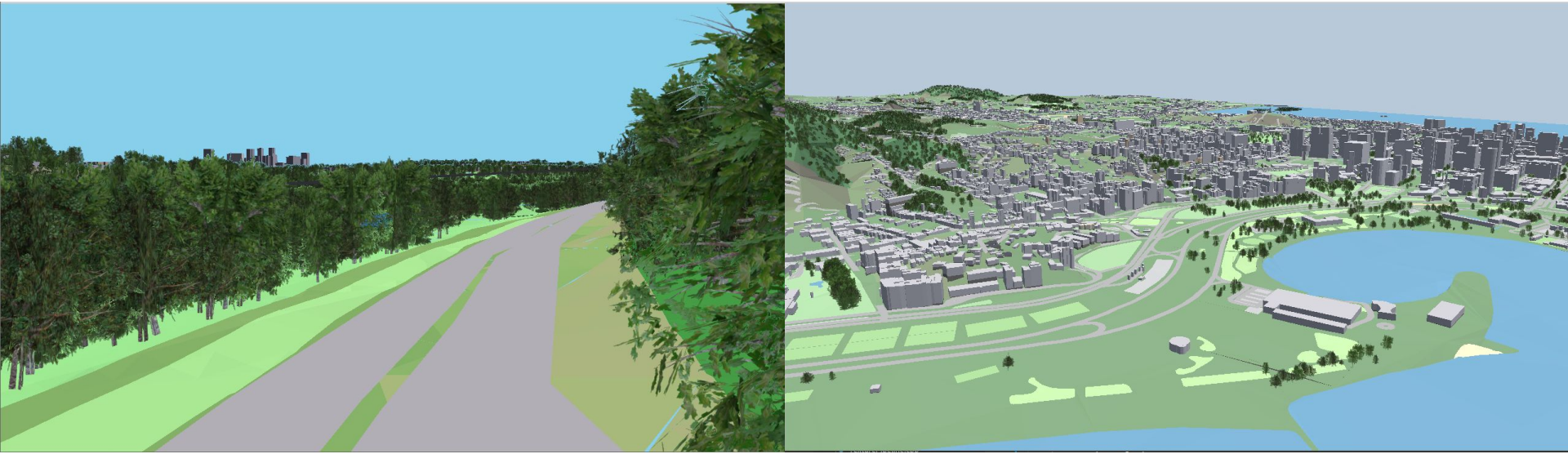

NREL SPA Algorithm
Sky and Sun Shader
Longitude
Latitude
Time & Date
Fed on 3D
Viewer startup
NREL
[3]
GLSL
[7]
To Display
Solar Irradiance and
Sun Position
Average Sky Color
Precomputed Data
from Hosek-Wilkie
Coefficients
Hosek & Wilkie Sky Color
Algorithm
Light Value
[10]
[9]
GLSL
[7]
To Display
Building Light Shader

The Hosek-Wilkie Algorithm yields cheap sky and sun shaders for calculating light incident on buildings.




3D Building Viewer
Pedestrian: W/A/S/D walk, Hold Shift run, Left-drag look, Scroll FOV, Esc exit
Enable 2D Overview
Exit pedestrian view
Show Analytics
2026-04-28 10:22:45 UTC
10
22
4096×

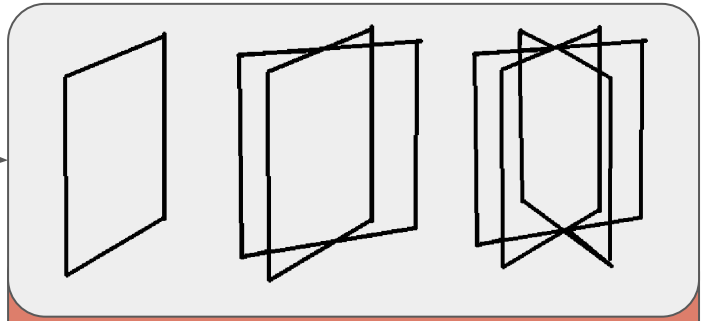
5 Performant Trees and Billboarding



How do we place thousands of trees whilst maintaining performance and keeping trees as true-to-life as possible?

Tree Image Textures

Create 3-quad mesh and place images on mesh

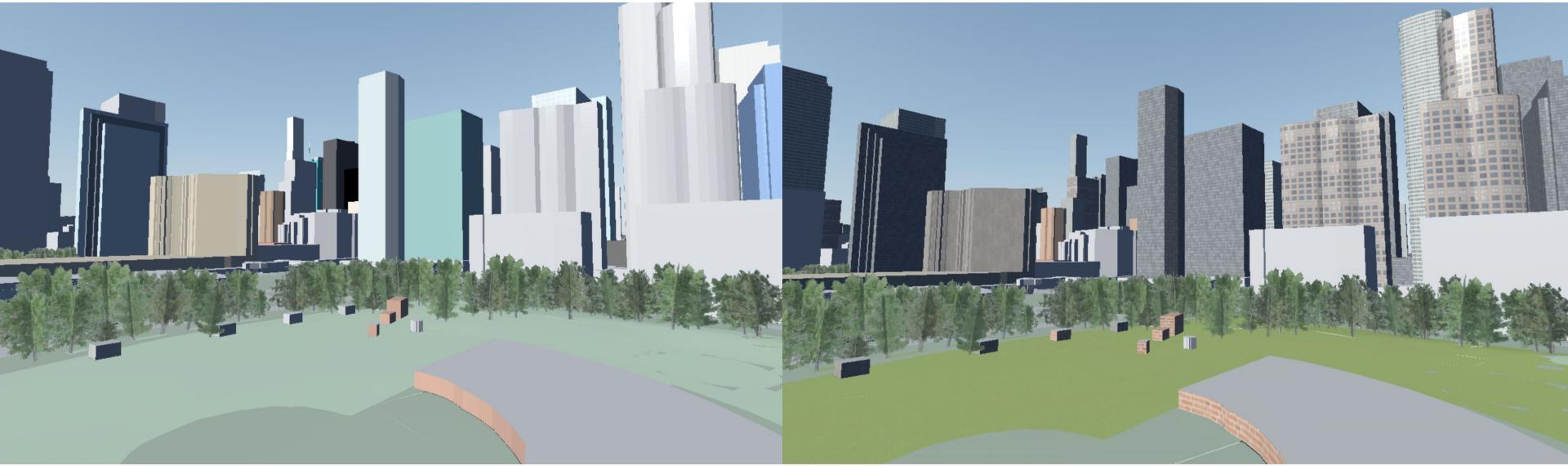


Randomize size* rotation, and tilt.

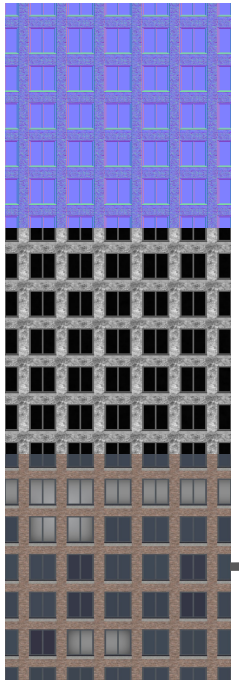
* we only randomize size of the OSM data does not contain the age of the tree.

Place Trees at OSM Tree ways or distribute over landuse:wood and landuse:natural areas

6 Ground & Building Textures



Solid colors allow for building differentiation, but how do we make the cities look more realistic?



[12]

[5]

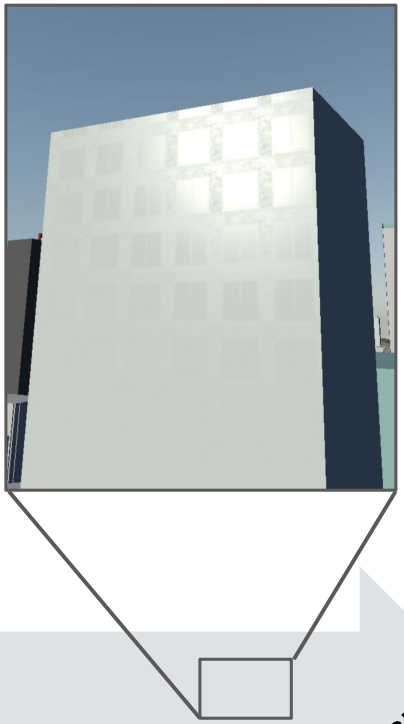
[5]

Building Mesh Data from Data Processing (has added parts from Complex Roofs)

Terrain Mesh Data from Mesh Processing

Apply Textures to Walls of buildings with a given material tag.

Apply textures to OSM areas that are interposed on terrain mesh.



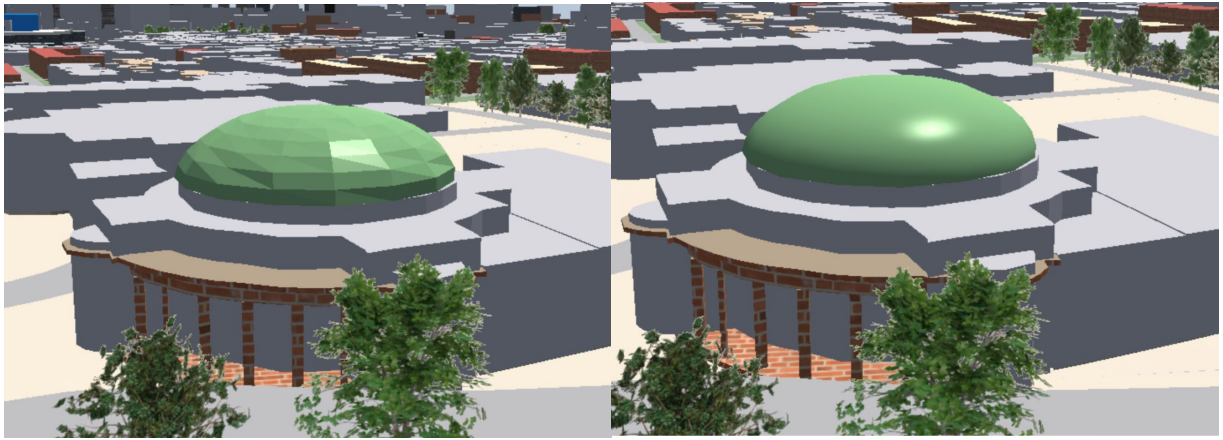
To Display

GLSL Pipeline

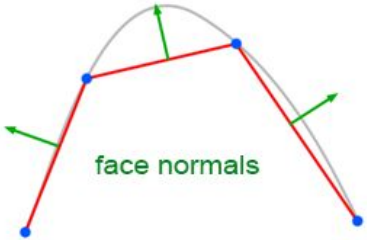
[7]

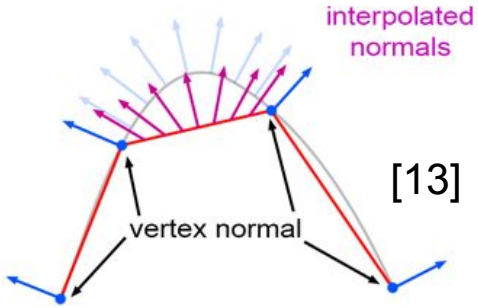
7 Shader Design Highlights

Mesh and Terrain Smooth Shading



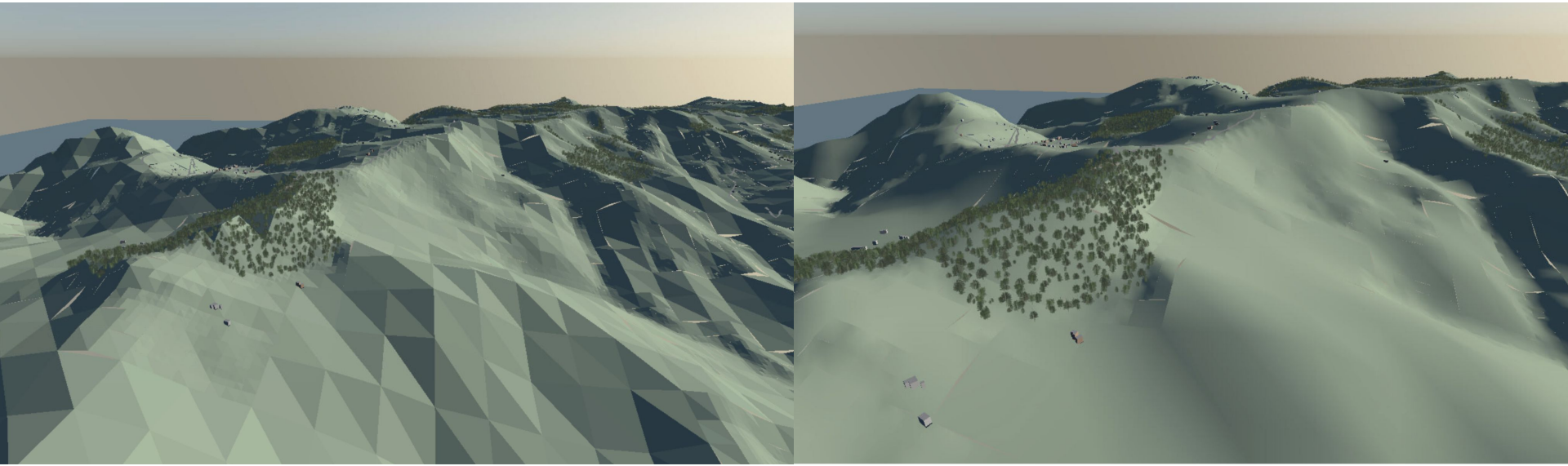
Smooth shading allows us to cheat by reducing tri count and placing computation in the fast GPU, making scenes cheaper.



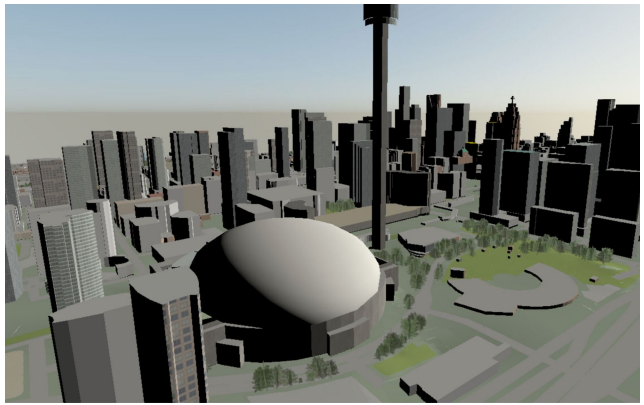


[13]

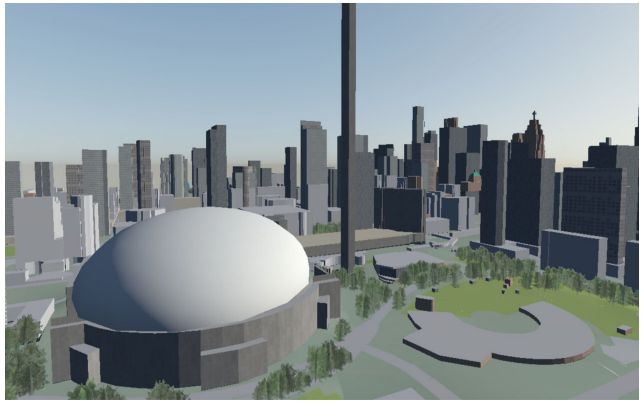
St. Helena looks much *smoother* with smooth shading!



Blinn-Phong Lighting Model (1/2)

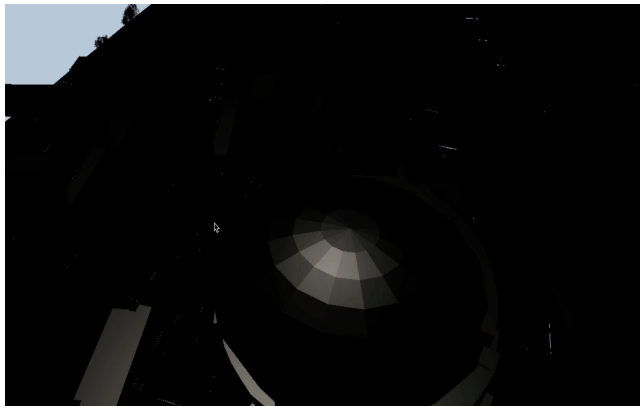


This is light from the sun, it is very harsh, the atmosphere should reflect the light around.



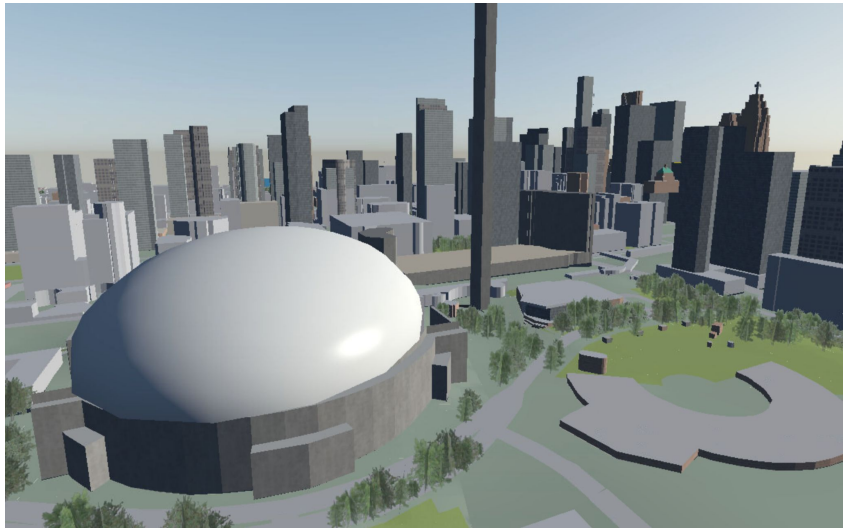
We can cheat this result by adding a simple ambient light to the opposing side of the sunlight.

Blinn-Phong Lighting Model (2/2)



The surface appears to be very Matte. We can cheat reflection calculations by adding a very sharp point light labelled a specular light.

This is the combined result.



8 Performance Considerations

To maintain performance we follow the process:

- Balancing detail at the micro and macro level
- Cheating by faking real-world systems and effects via approximation and precomputation
- Cull data that is not pertinent to the current camera position or view angle

Nielsen's Usability Engineering [2, pp. 135] gives a hard limit of 0.1s before the user finds a program unresponsive. Our renderer runs at a **minimum 100ms**. [1]

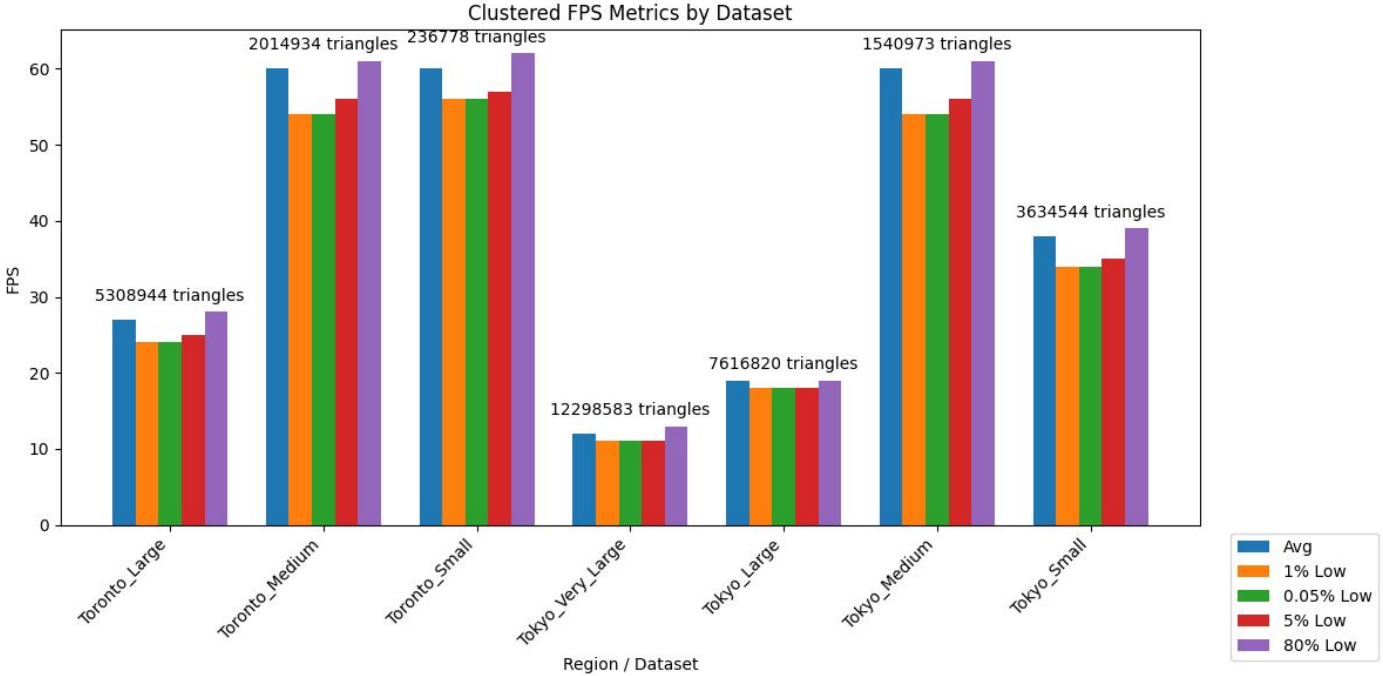
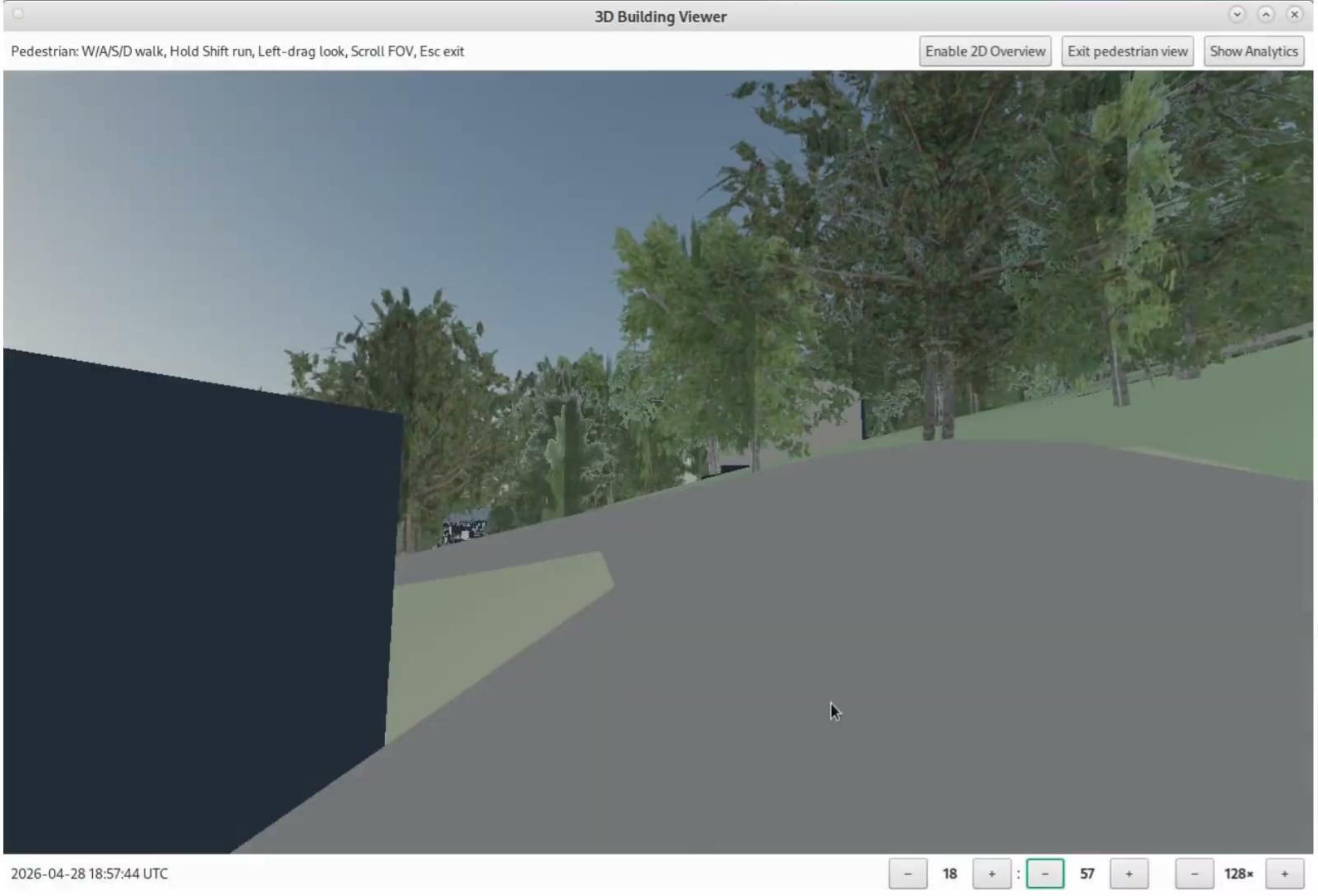
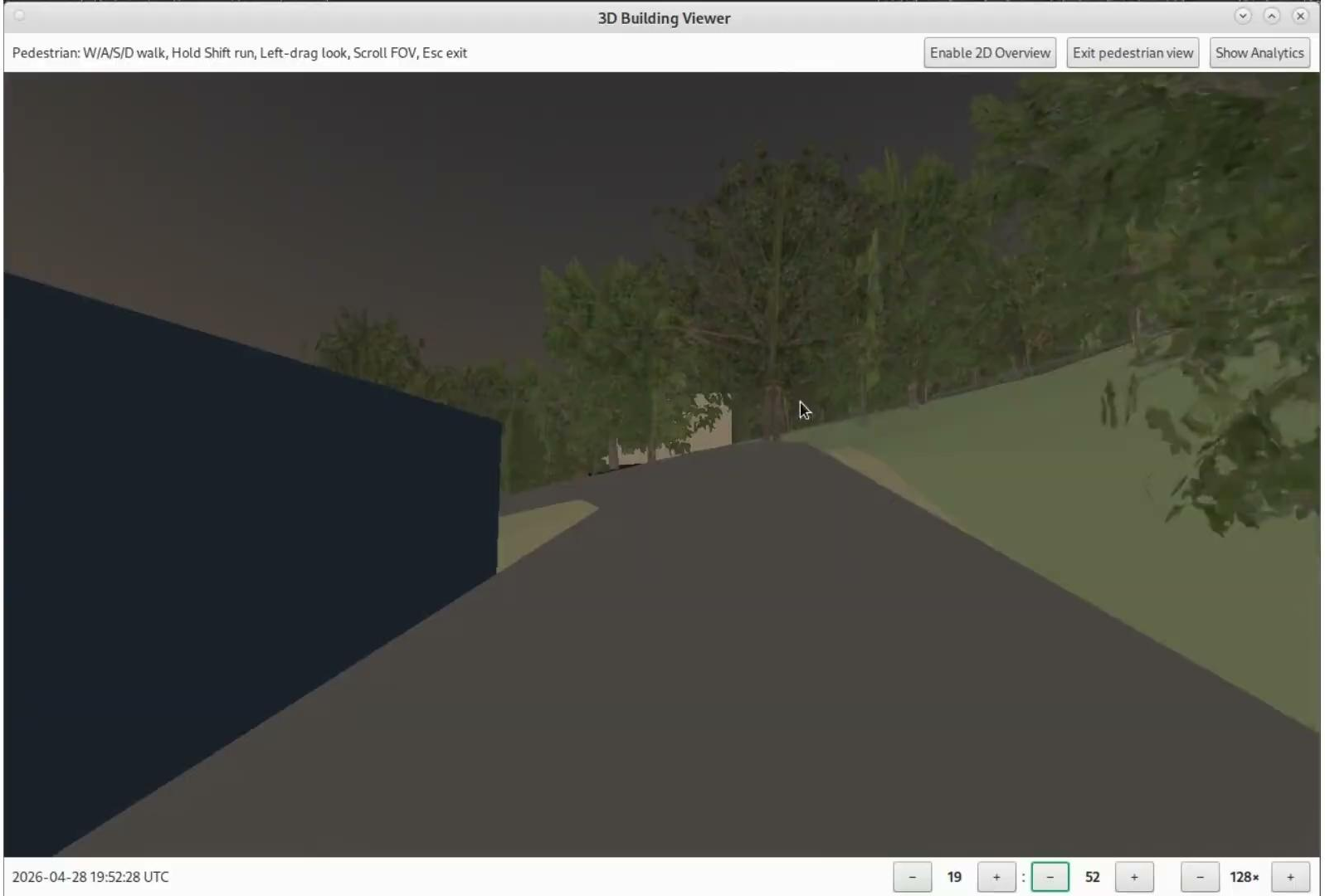


Figure 8: Running at the above given FPS for Toronto and Tokyo scenes rendered at 1920x1080 resolution. [1]


3D Building Viewer
Pedestrian: W/A/S/D walk, Hold Shift run, Left-drag look, Scroll FOV, Esc exit
Enable 2D Overview
Exit pedestrian view
Show Analytics
2026-04-28 18:57:44 UTC
18
57
128×

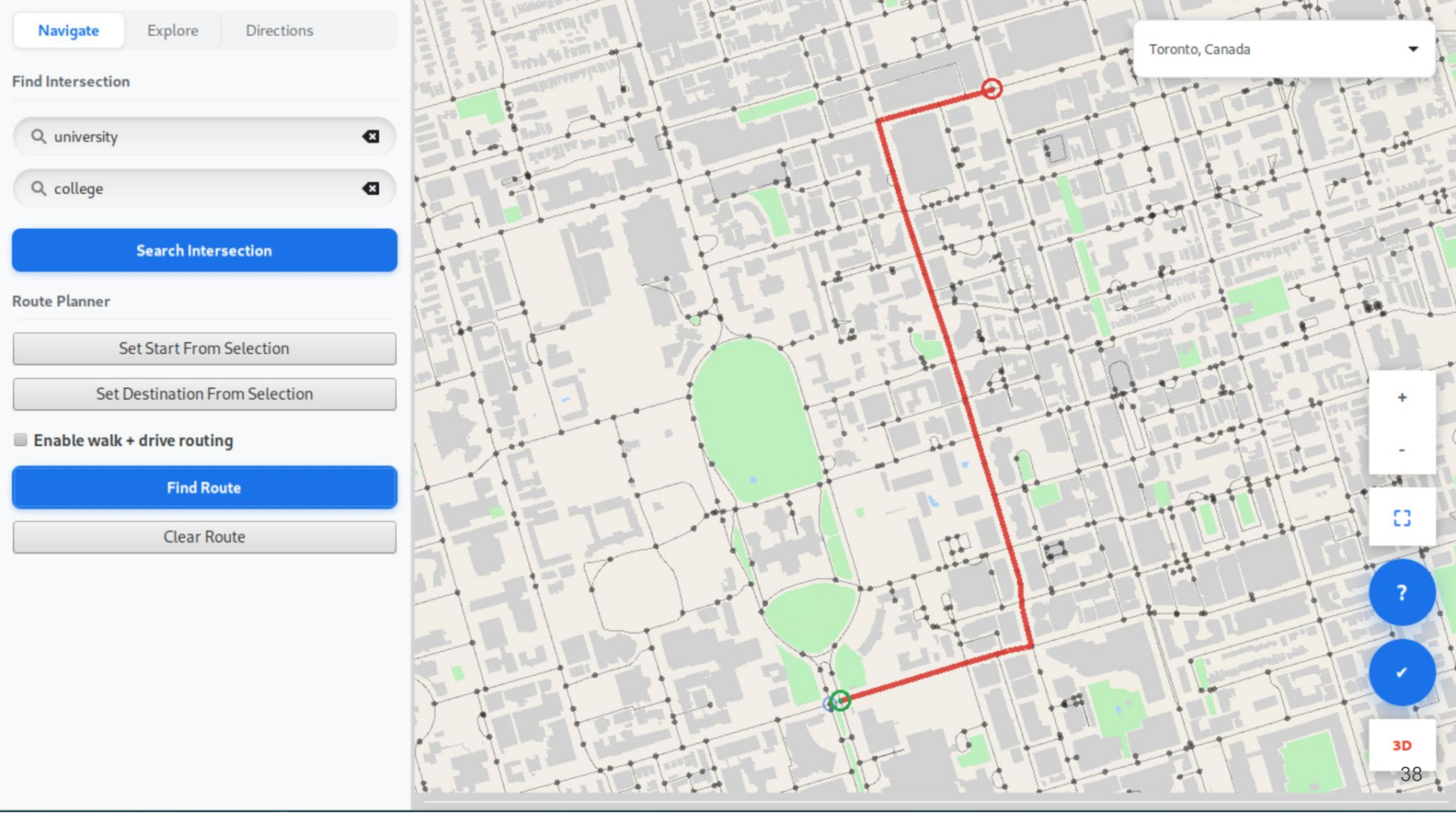

3D Building Viewer
Pedestrian: W/A/S/D walk, Hold Shift run, Left-drag look, Scroll FOV, Esc exit
Enable 2D Overview
Exit pedestrian view
Show Analytics
2026-04-28 19:52:28 UTC
19
52
128×

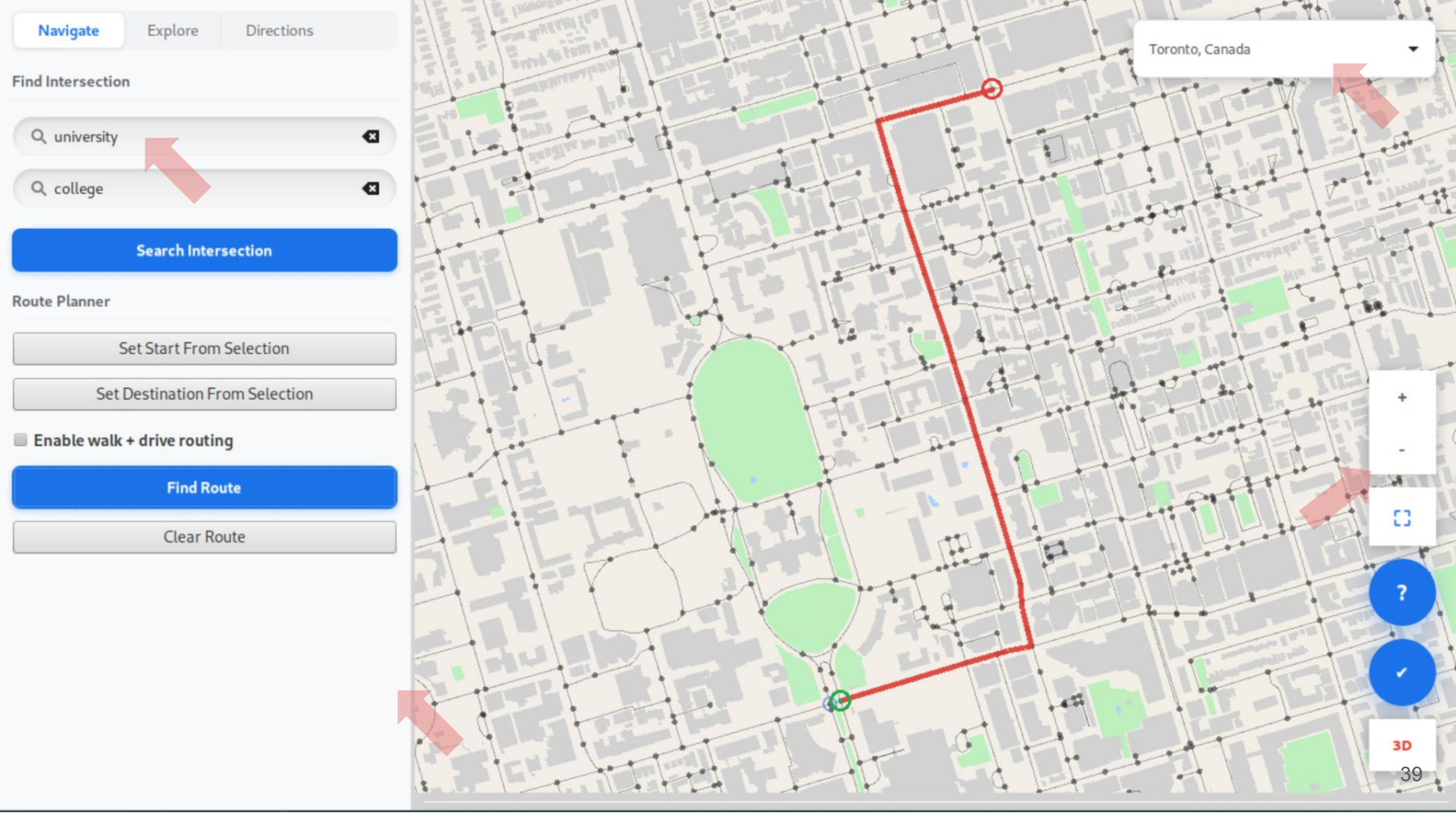
2D Viewer Highlights

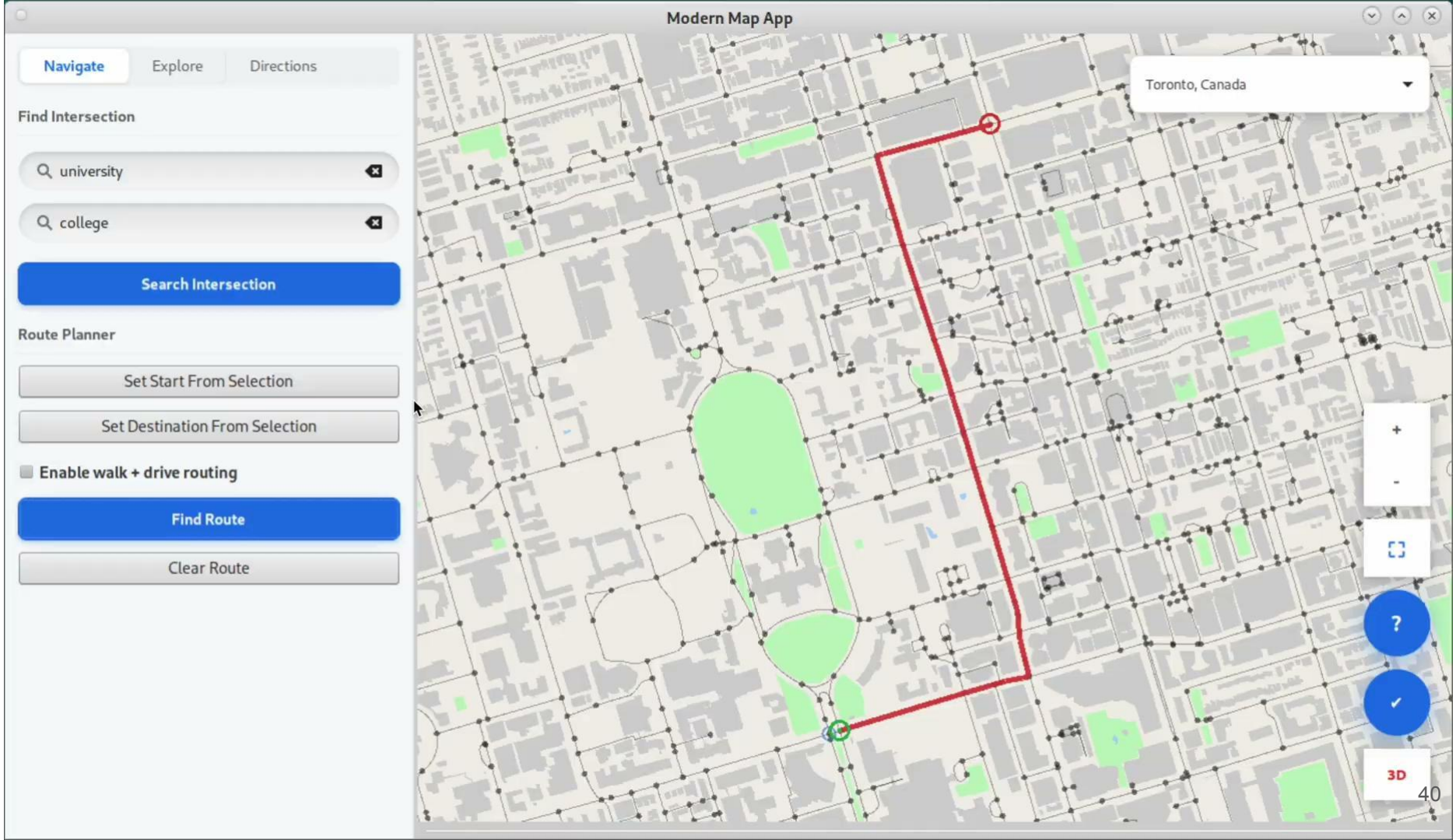
1 Styling with CSS and MVVM
2 POI Navigator

3 Milestone 1: Spatial Quadtree
4 Milestone 3
5 Intelligent Directions
6 Travelling Courier Optimization

1 Custom Styling with CSS

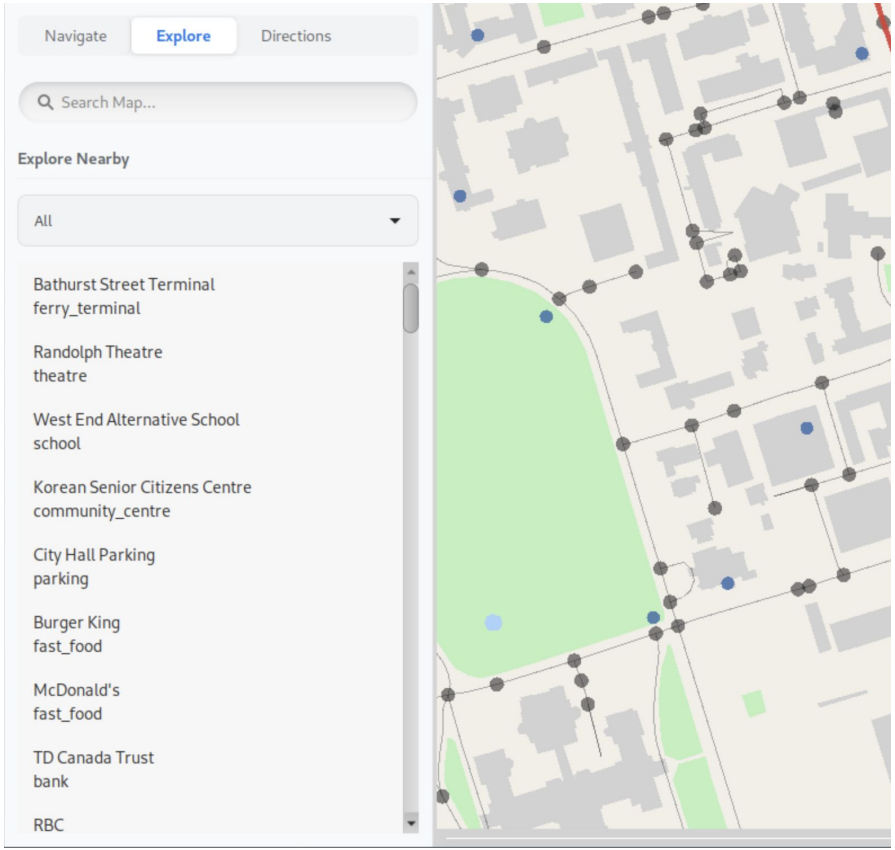

Navigate
Explore
Directions
Find Intersection
university
college
Search Intersection
Route Planner
Set Start From Selection
Set Destination From Selection
Enable walk + drive routing
Find Route
Clear Route
Toronto, Canada
+
-
?
3D

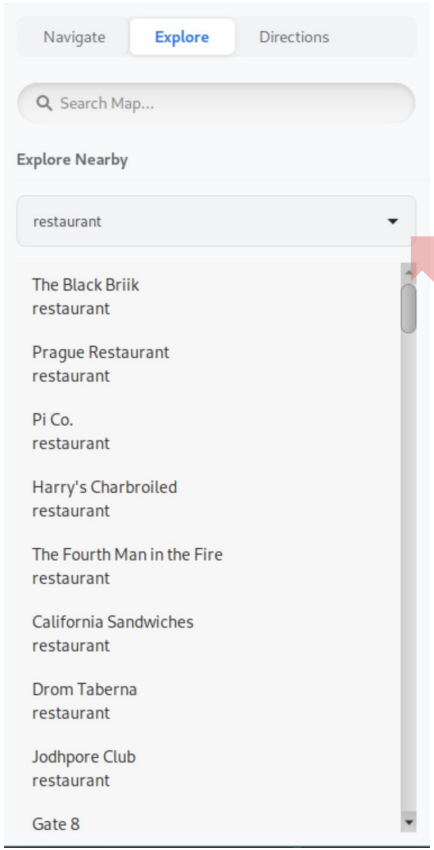

Navigate
Explore
Directions
Find Intersection
university
college
Search Intersection
Route Planner
Set Start From Selection
Set Destination From Selection
Enable walk + drive routing
Find Route
Clear Route
Toronto, Canada
+
-
?
3D

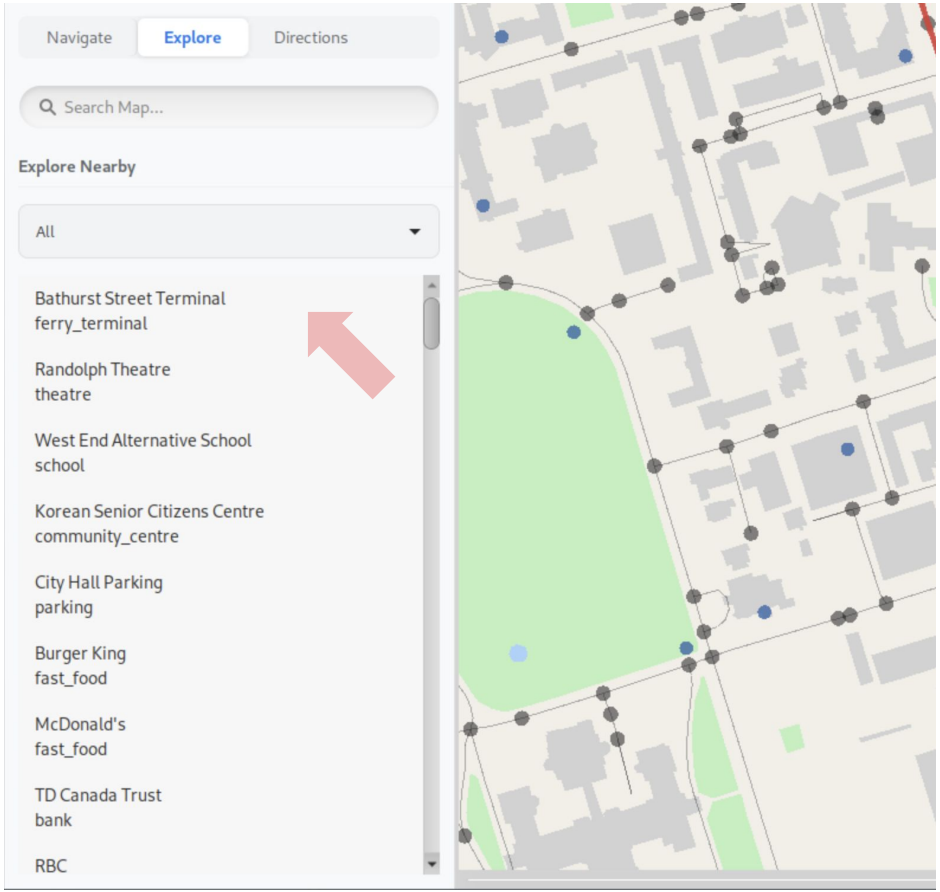

Modern Map App
Navigate
Explore
Directions
Find Intersection
university
college
Search Intersection
Route Planner
Set Start From Selection
Set Destination From Selection
Enable walk + drive routing
Find Route
Clear Route
Toronto, Canada
3D

2 POI Navigator

Nearby POI's are Displayed on Explore Tab

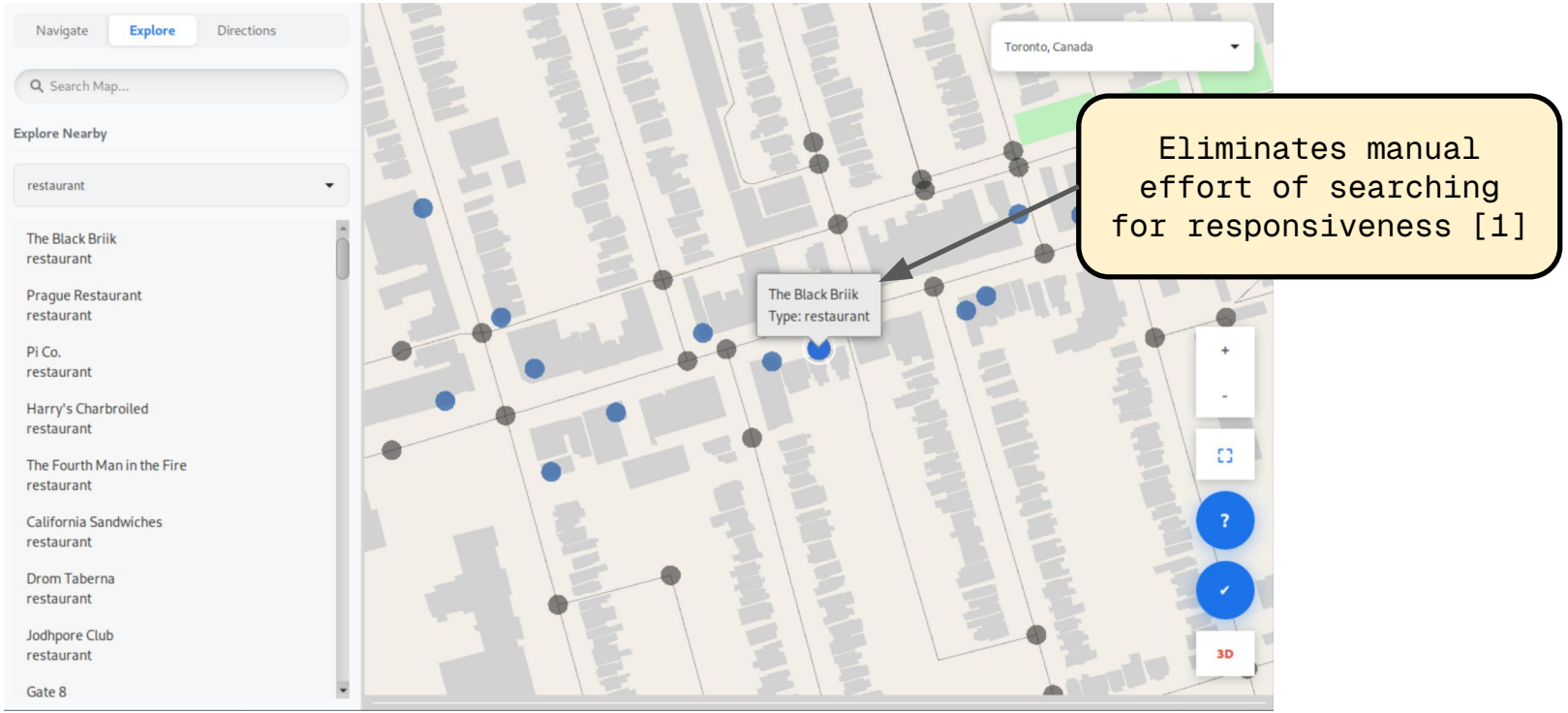






Fitts's Law: having search in close proximity to the map reduces the time needed to interact with results [1].

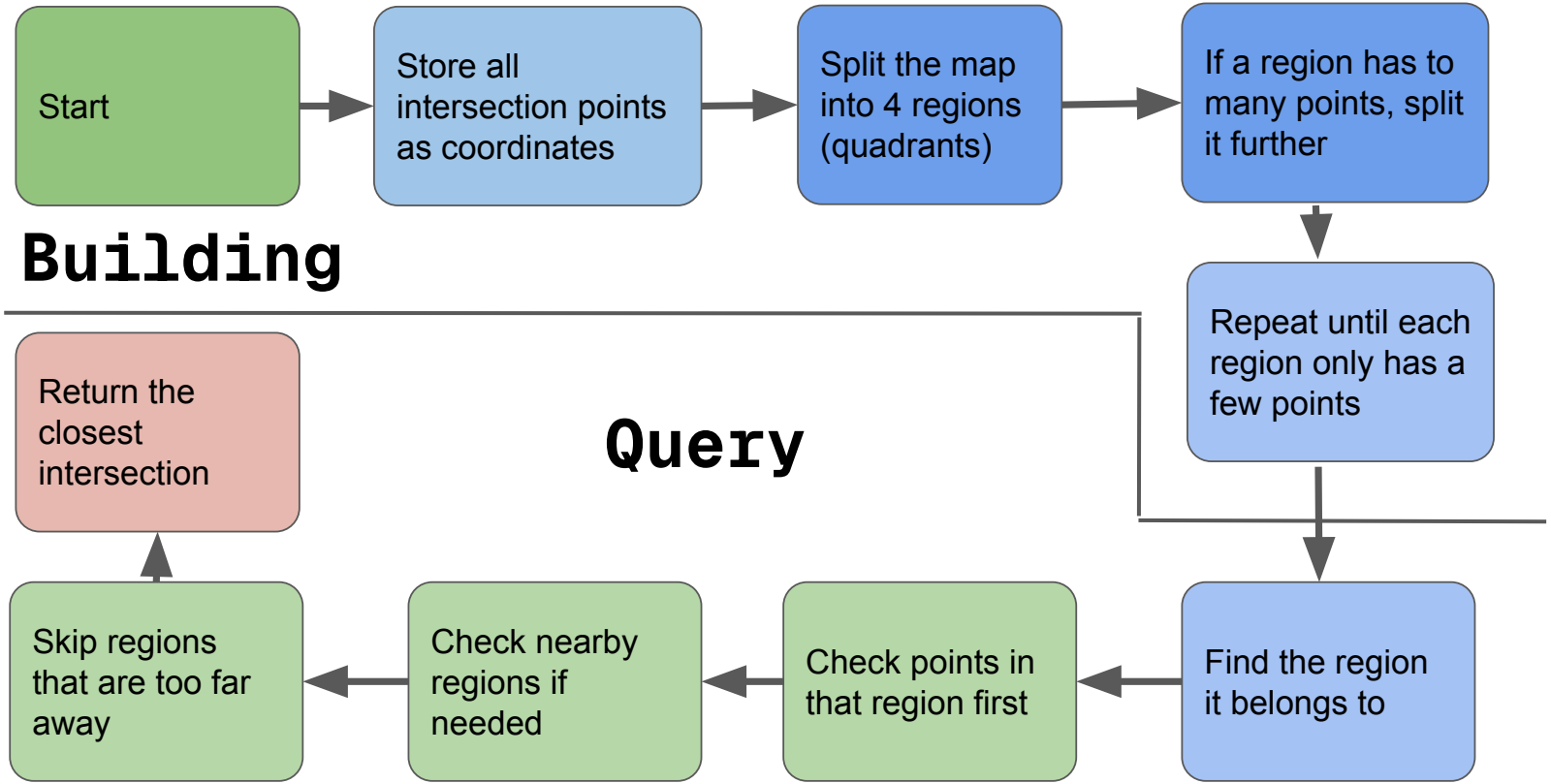
When you click on a POI, it takes you there!



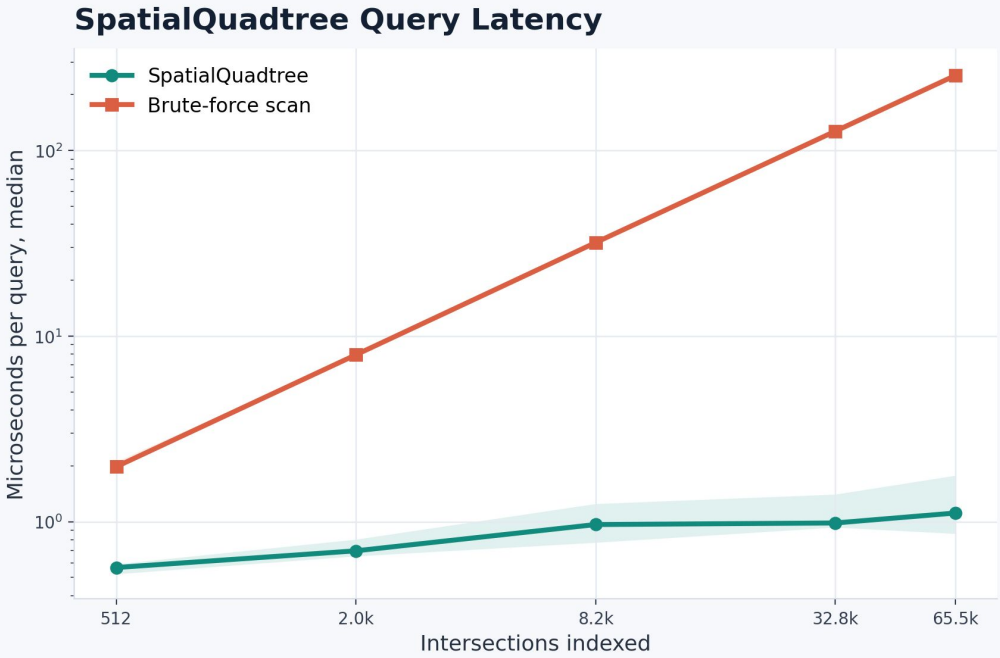
3 M1: Spatial Quadtree

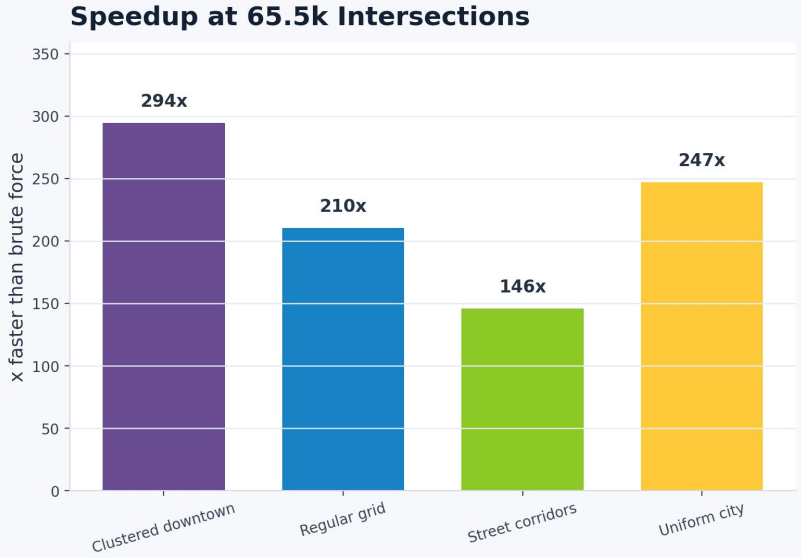
Visualization: How it Works

How it varies from the standard approach

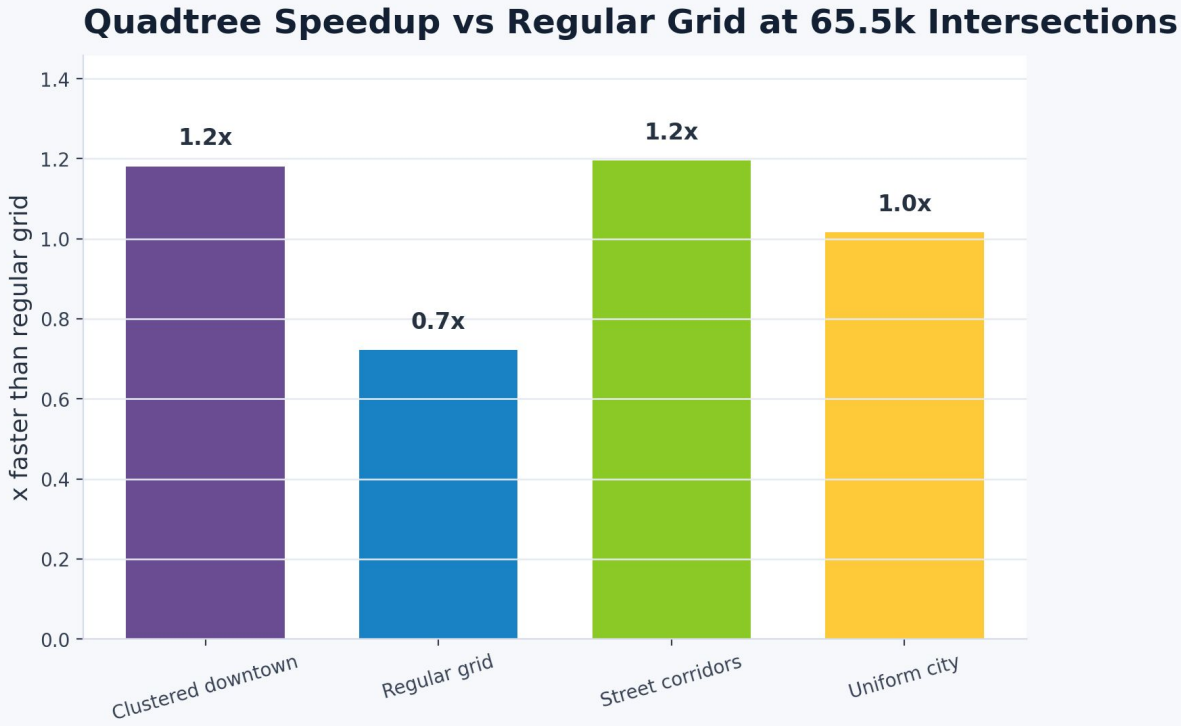


Results After Running Benchmarks



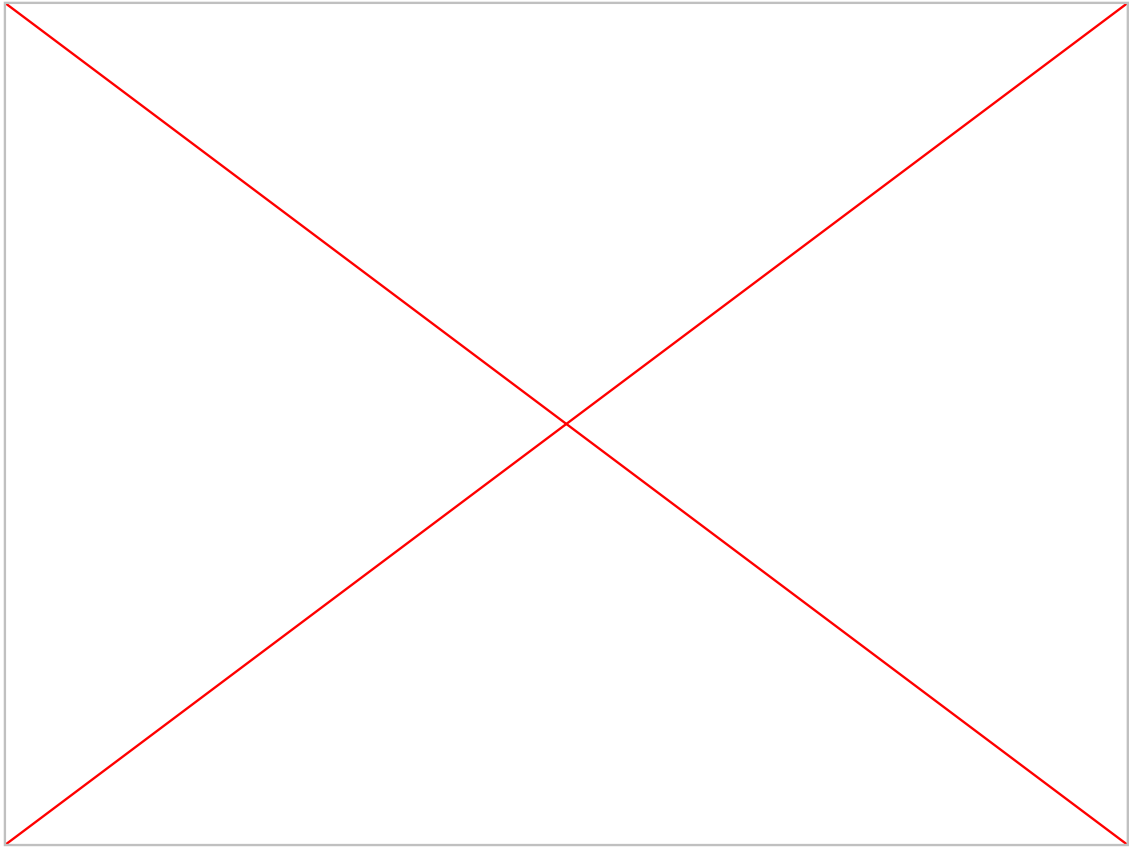


Benchmarking Analysis

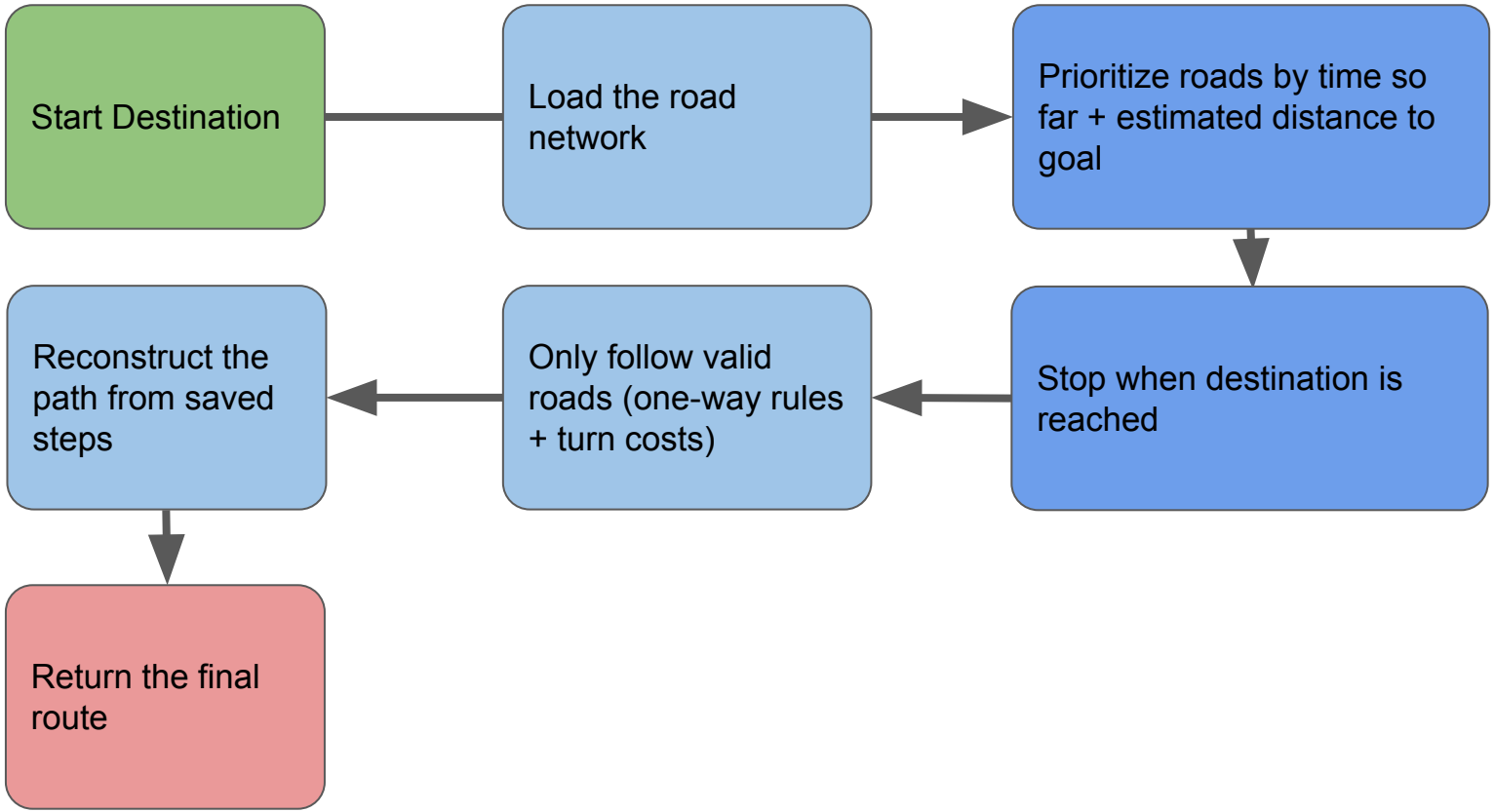


4 Milestone 3

We used the A* pathfinding algorithm



How our implementation works



Implementation Procedure

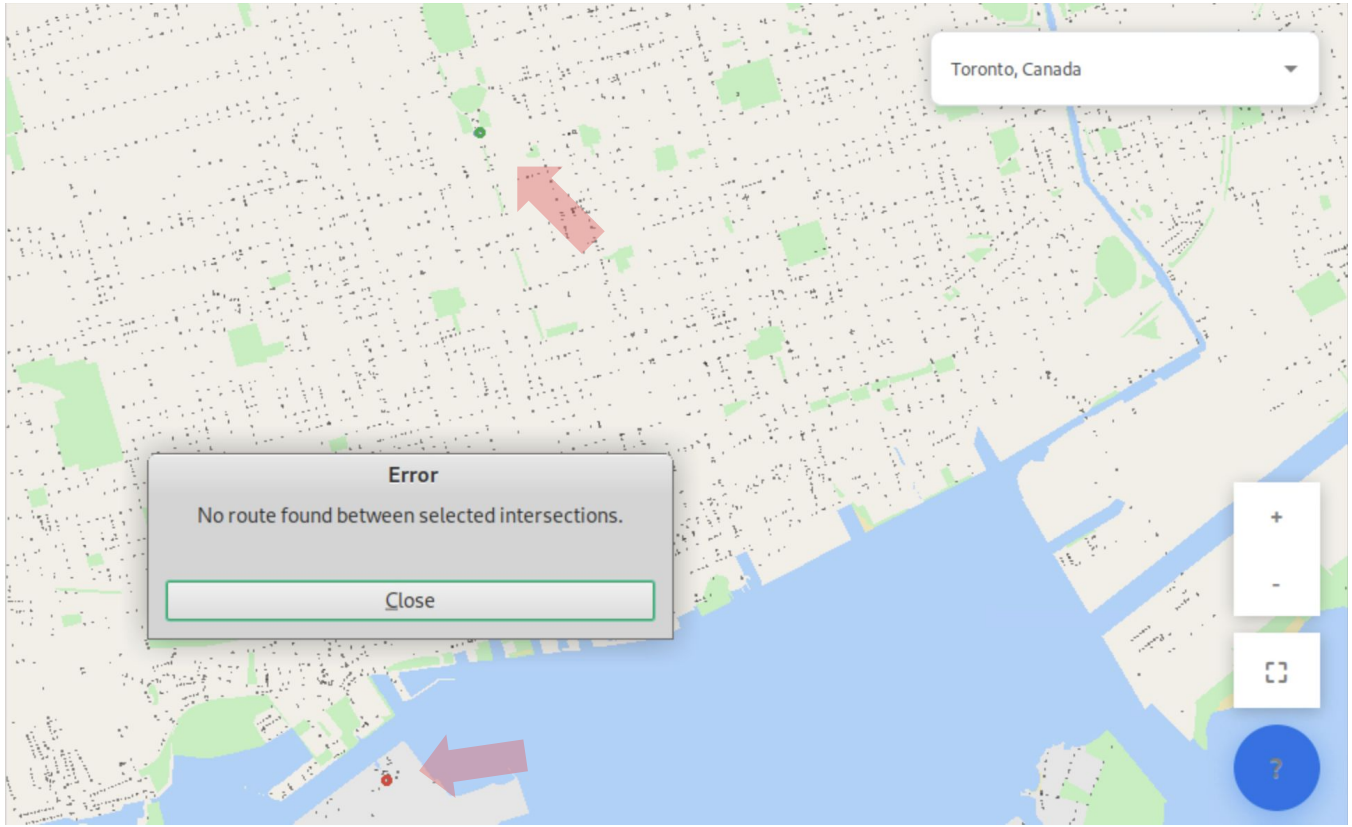
To improve performance we followed the process:

- **Heuristic:**
  straight-line distance / fastest speed

- **Key optimization:**
  cached segment times + cached adjacency lists

  Walking Mode:
- find reachable pickup intersections, then run one A* search from all of them

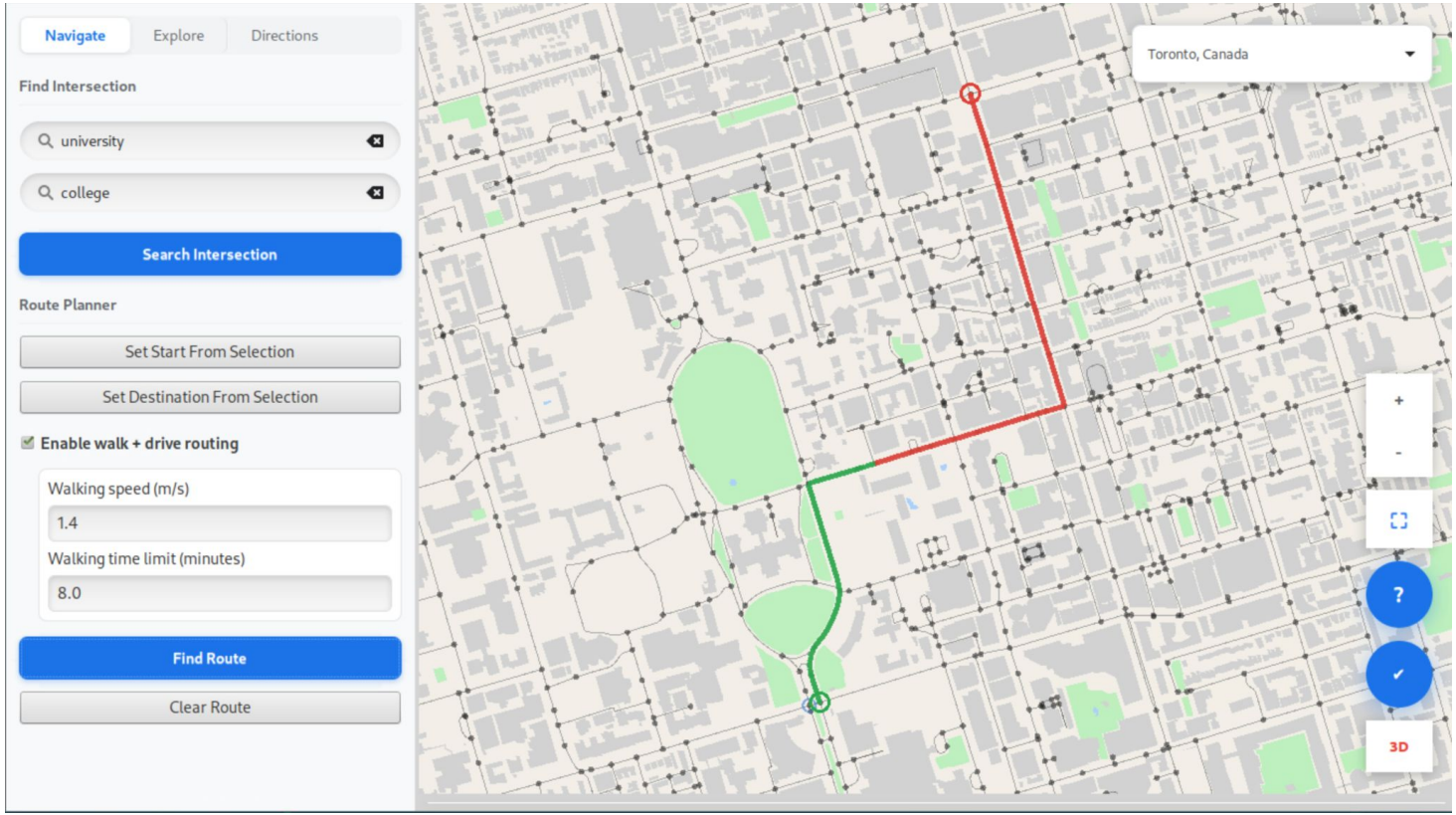
Handling Edge Cases

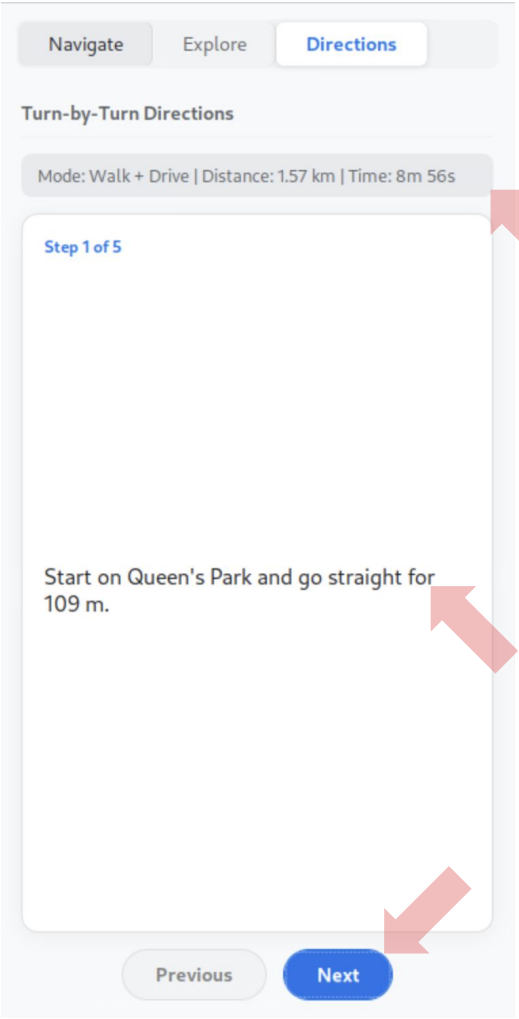


Uh oh! We selected an impossible route.

5 Intelligent Directions

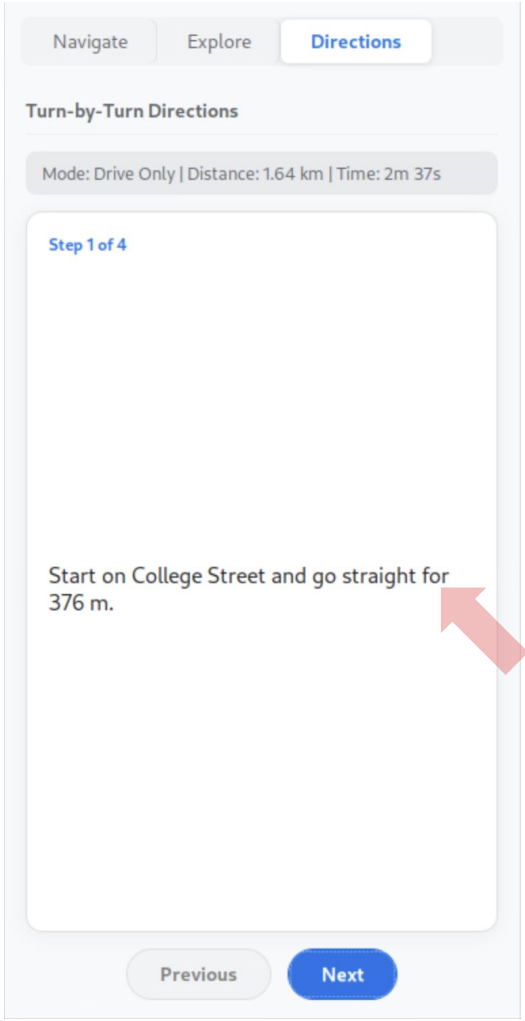
Example: Walk + Drive Mode Directions





Walking is the **Green** Path

Driving is the **Red** Path



Different directions based on walking vs driving

6 Milestone 4

The Problem at Hand

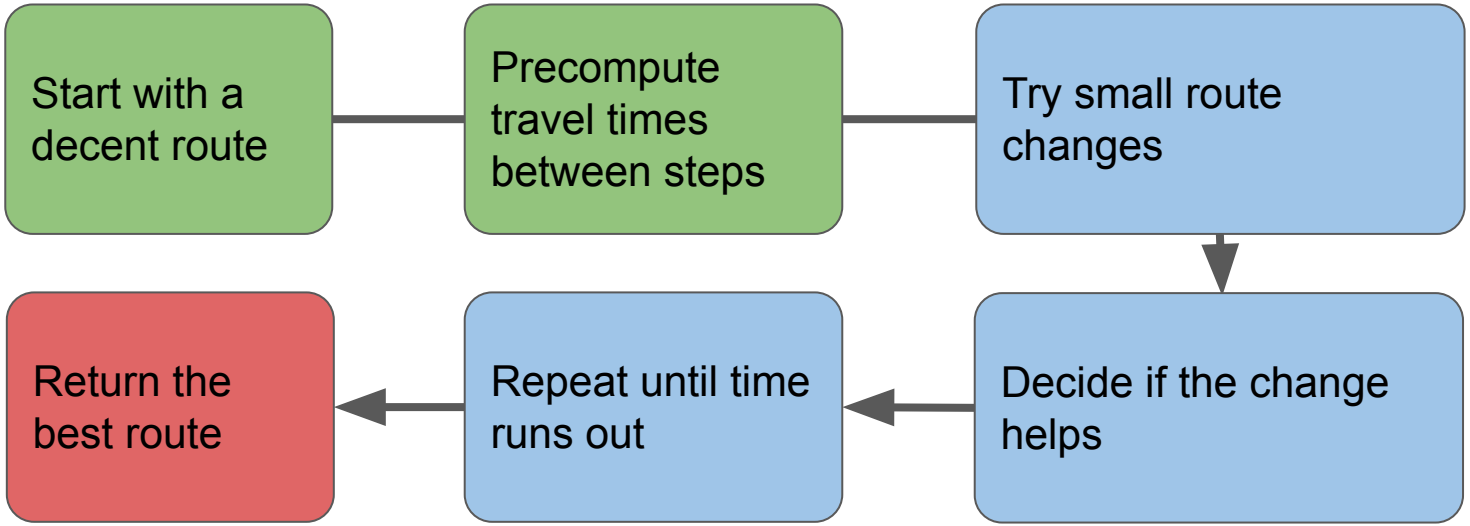
Travelling Courier = TSP with more rules

Our strategy

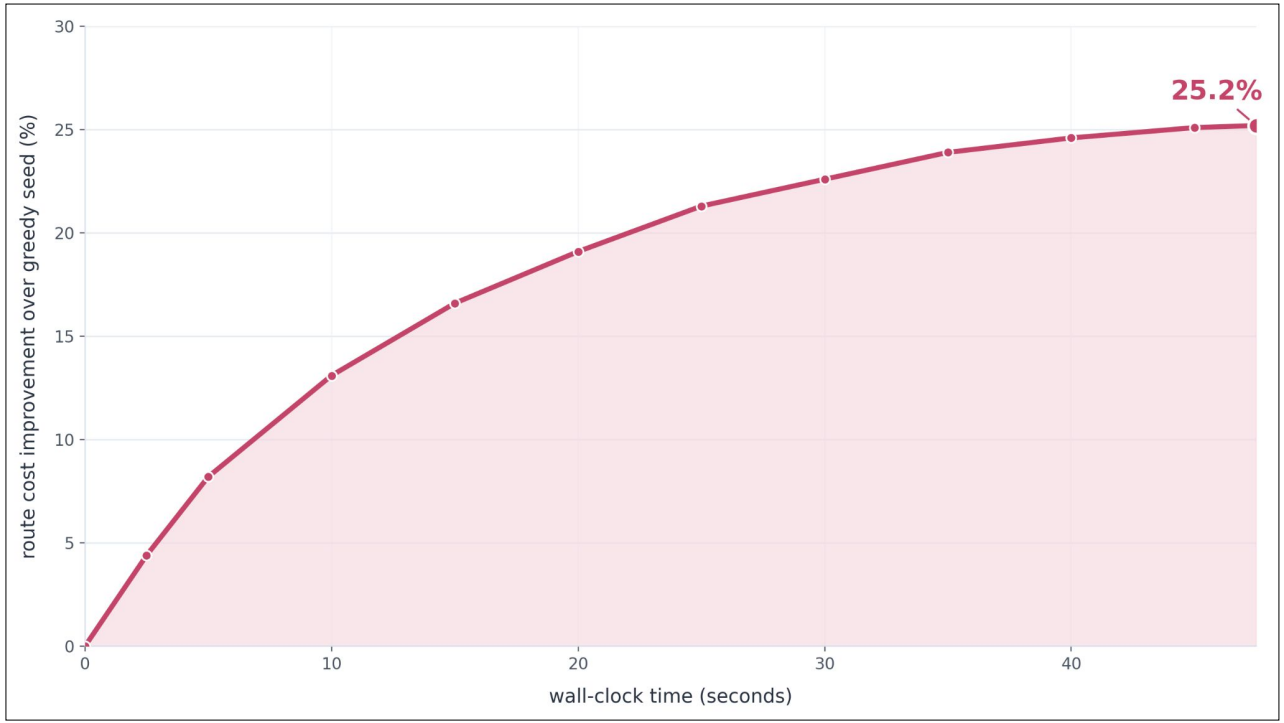
- **First:** build a valid route using a greedy nearest-neighbour strategy
- **Then:** continuously improve that route *instead* of trying every possible route (too slow) [23]

Johnson & McGeoch describe TSP as a classic hard routing problem where heuristics/local search are used to find good routes quickly. Johnson & McGeoch, 1997. See [23]

Complete Implementation

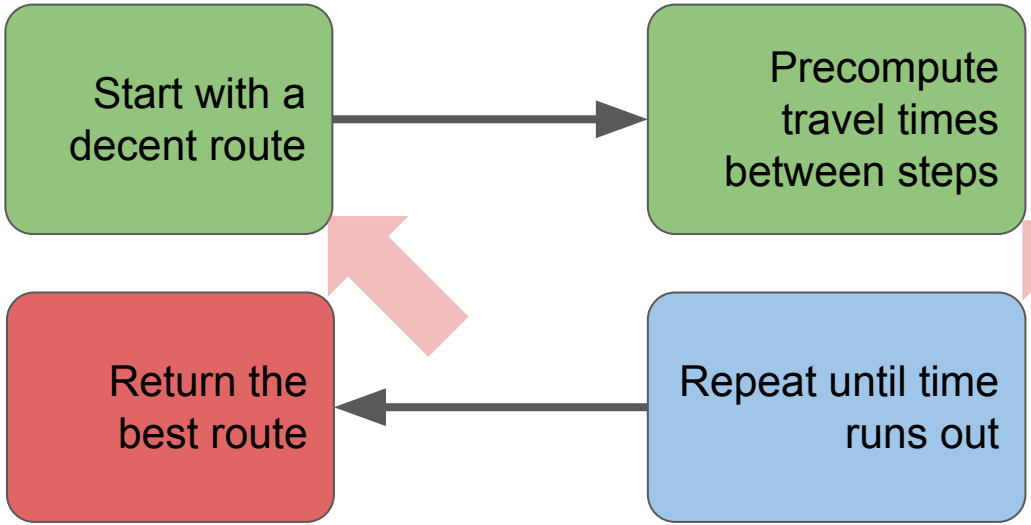


Our results: Simulated annealing improves the greedy route over time



Starting from a feasible nearest-neighbor route, simulated annealing reduced total route cost by 25.2% within the 47.5s time budget.

Main Optimizations of the Courier Route



Greedy Start improves the QOR (Quality of Result) [24]

Precomputation of travel times improves the QOR [25]

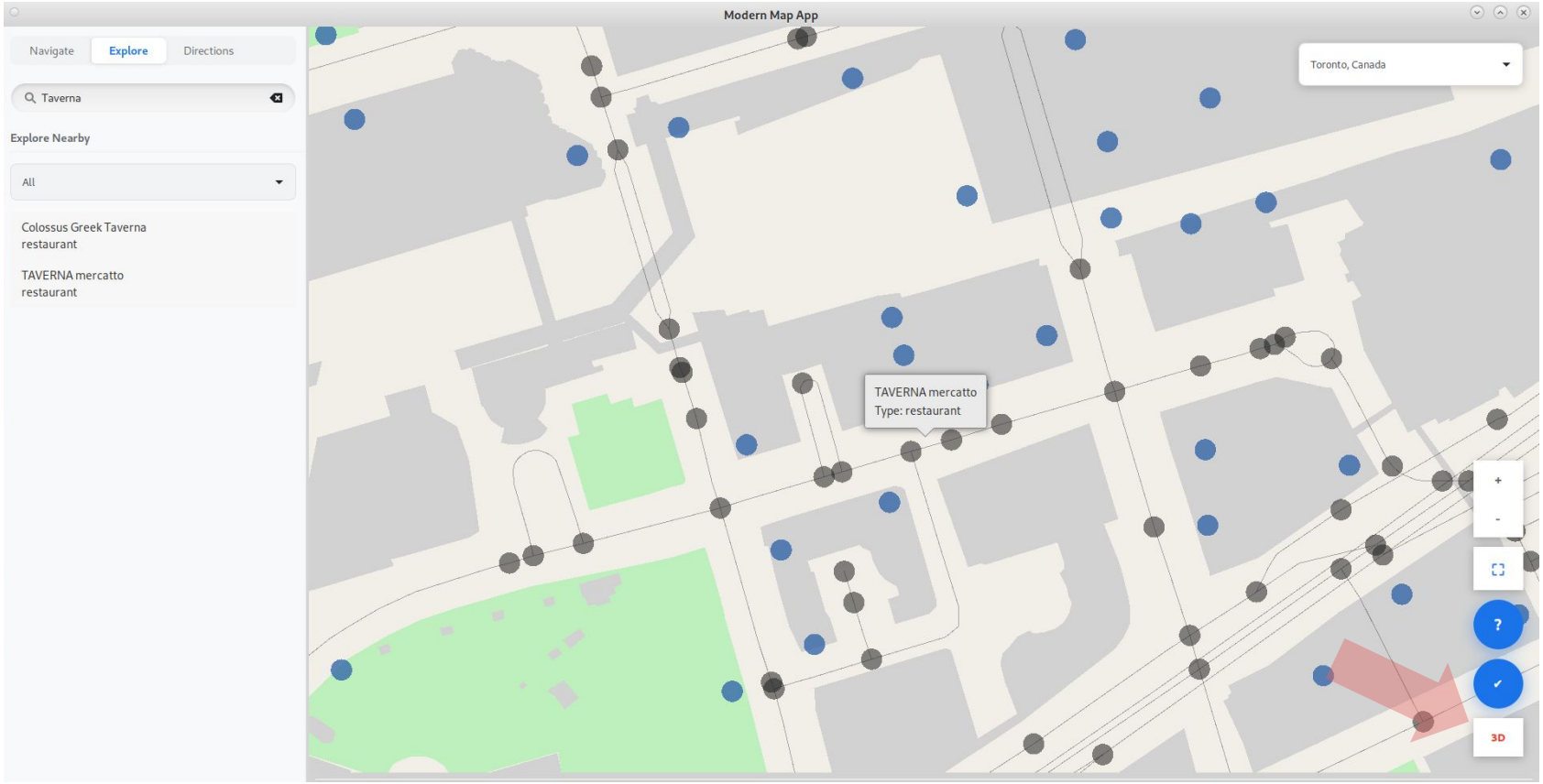
Parallelizing (Multithreading) improves QOR

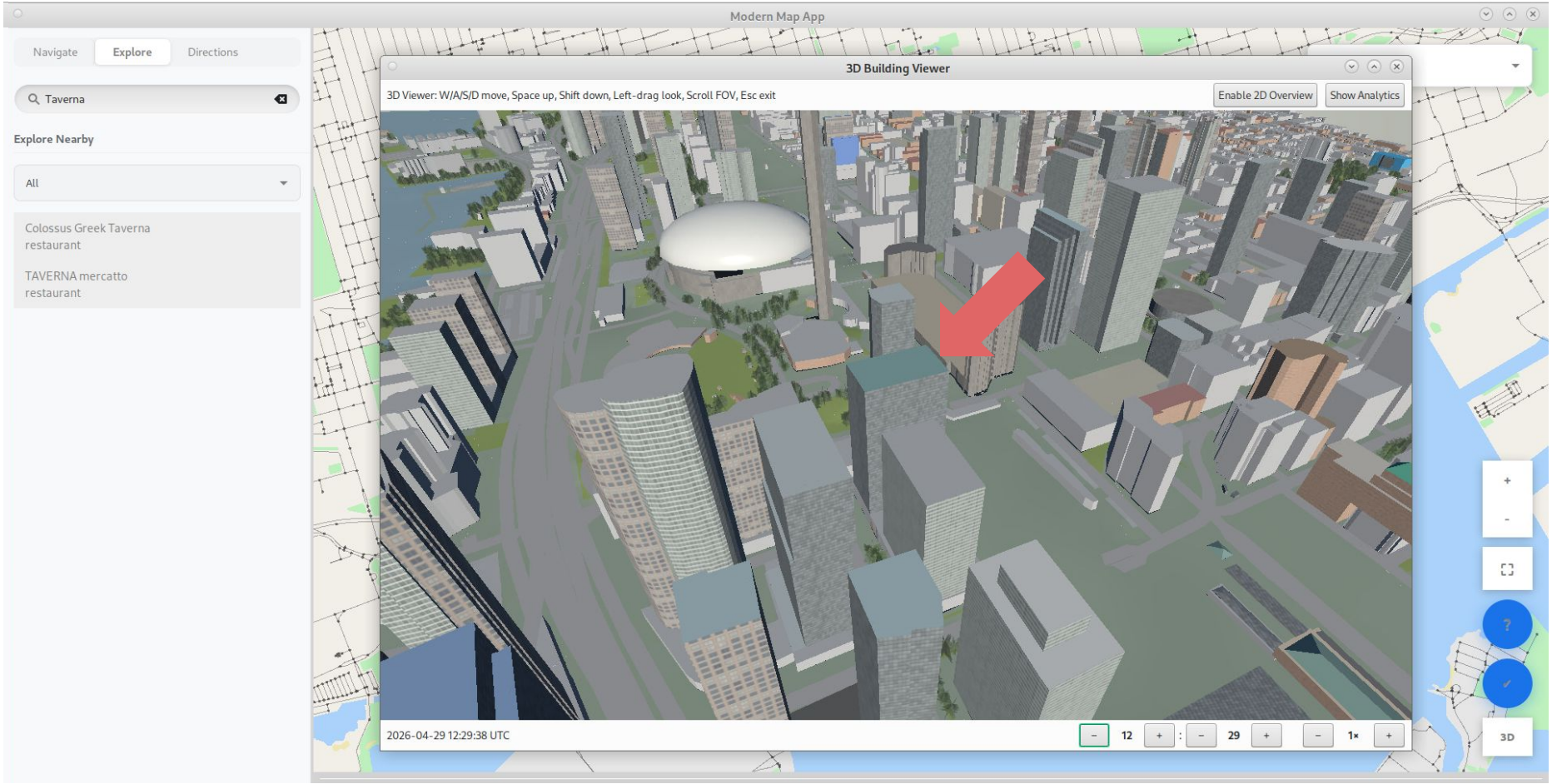
Inspired by simulated annealing and TSP local-search methods: Kirkpatrick et al. (1983), Johnson & McGeoch (1997), Applegate et al. (1999). See [24] and [25]

With these features,

let’s see it all in action.

Let's say you're finding a restaurant in Toronto …




Modern Map App
Navigate
Explore
Directions
Taverna
Explore Nearby
All
Colossus Greek Taverna
restaurant
TAVERNA mercatto
restaurant
3D Building Viewer
3D Viewer: W/A/S/D move, Space up, Shift down, Left-drag look, Scroll FOV, Esc exit
Enable 2D Overview
Show Analytics
2026-04-29 12:29:38 UTC
12
29
1×
3D

I double check my route to see how to cross the tracks …

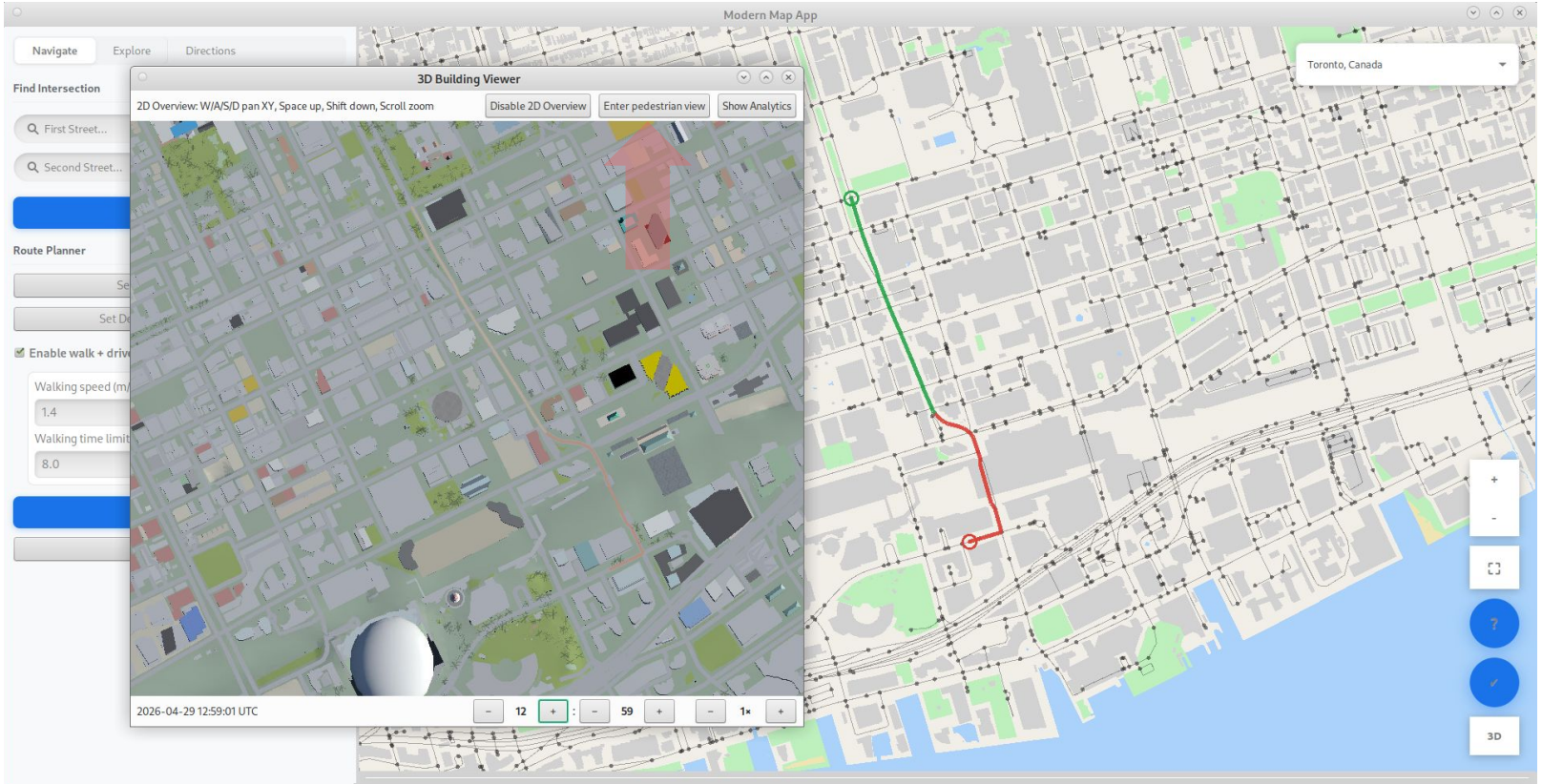
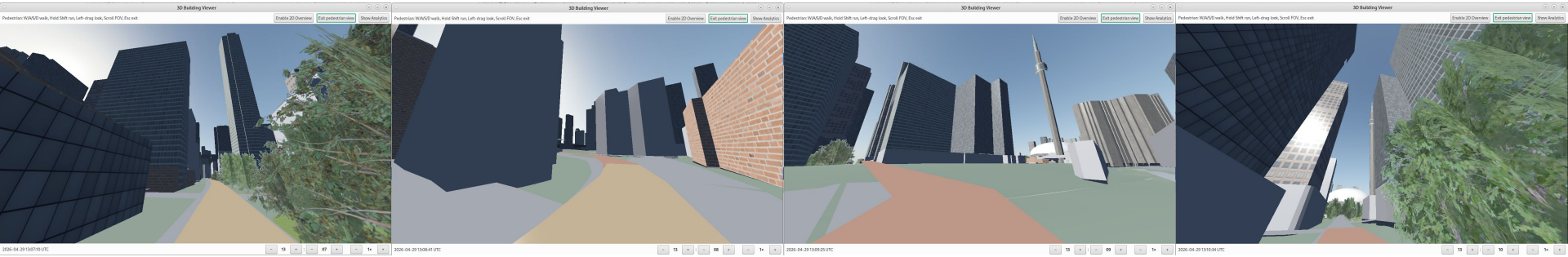
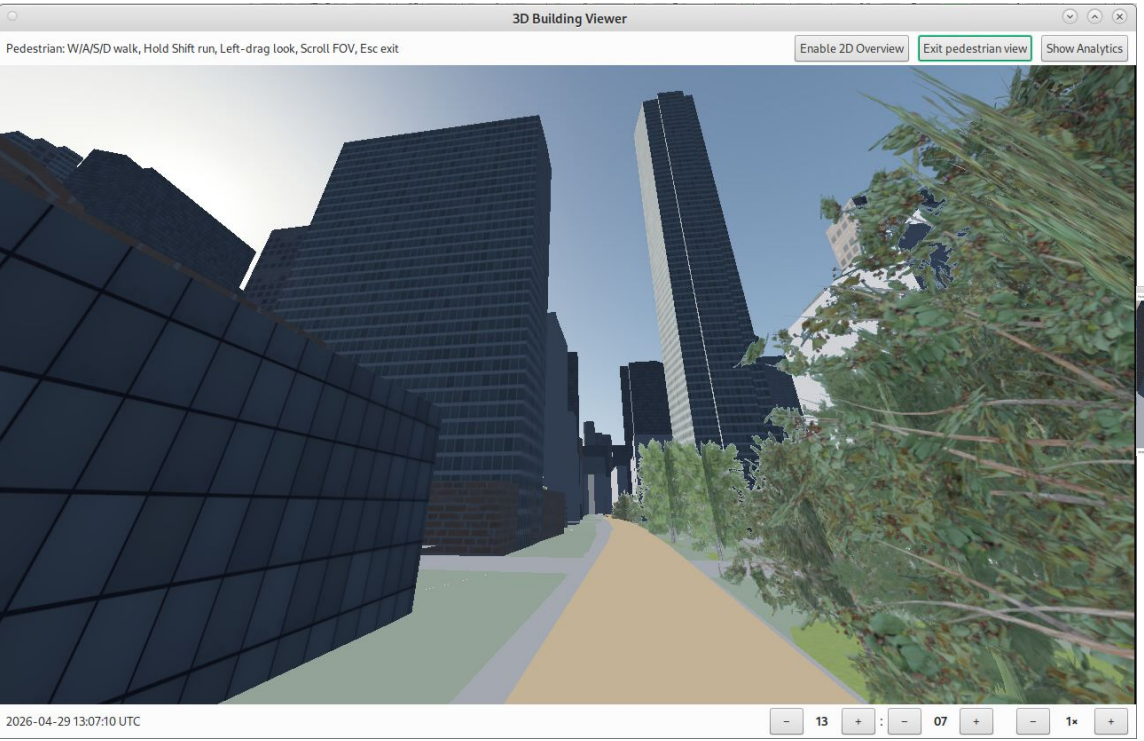


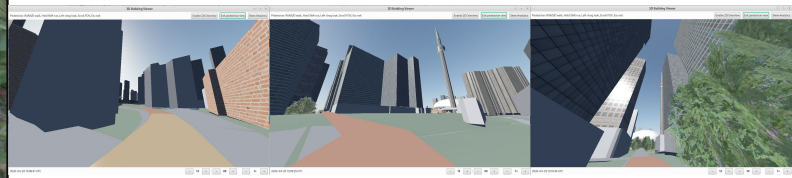
Figure 9: 3D Viewer is now in 2D Overivew mode.

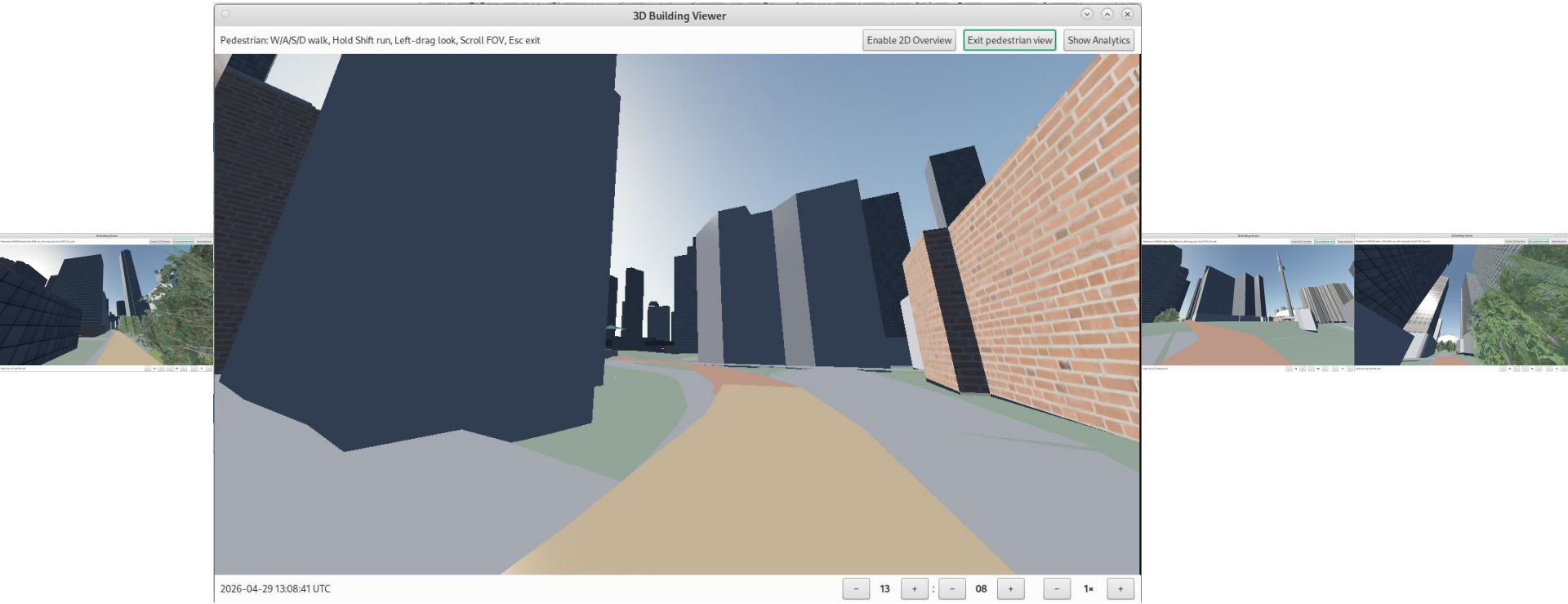
Let’s go on the walk!

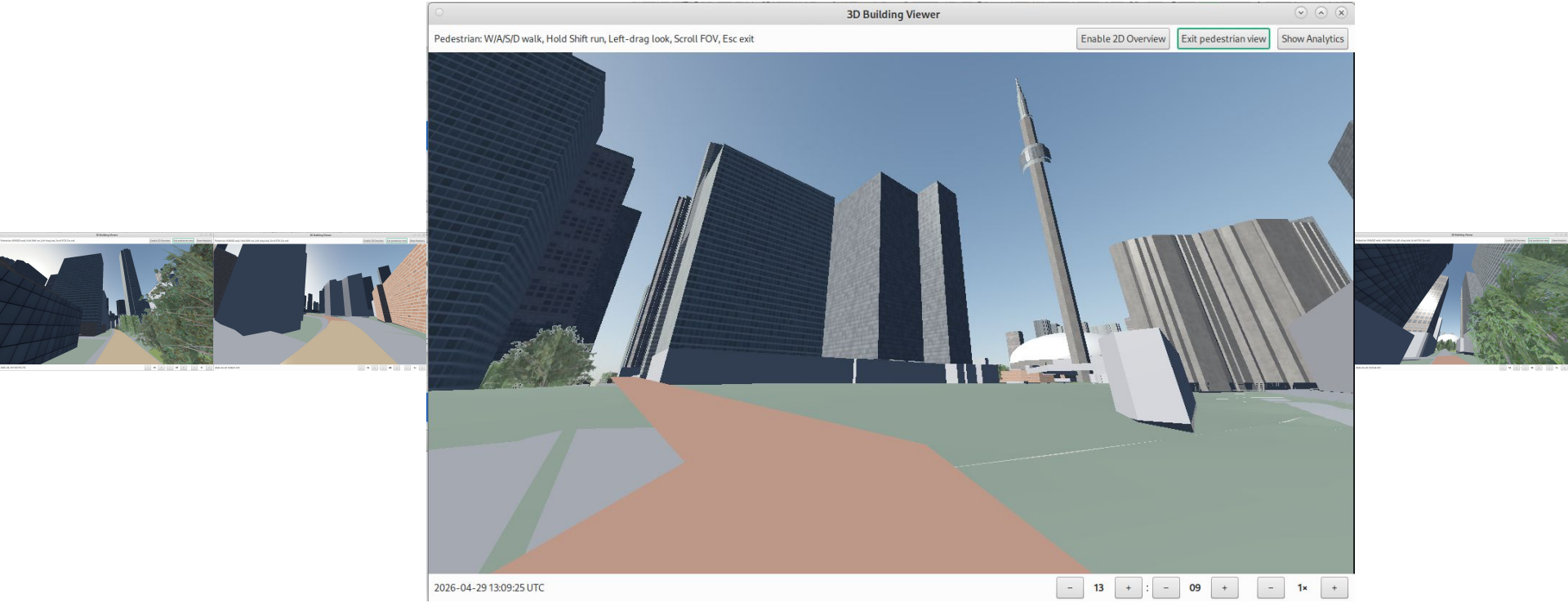


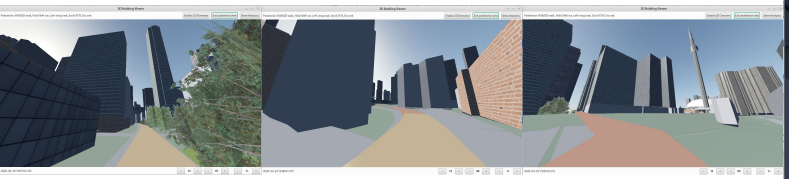
The views are quite nice … CN tower visible as I cross the train tracks!

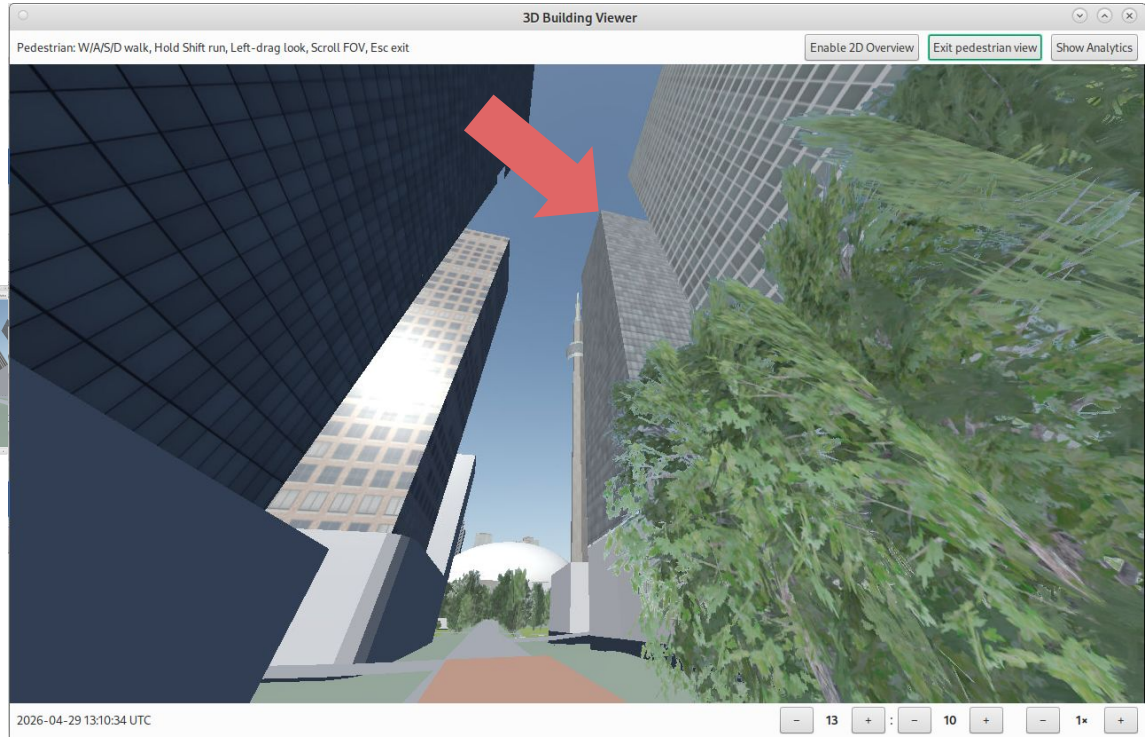

3D Building Viewer
Pedestrian: W/A/S/D walk, Hold Shift run, Left-drag look, Scroll FOV, Esc exit
Enable 2D Overview
Exit pedestrian view
Show Analytics
2026-04-29 13:07:10 UTC
13
07
1×



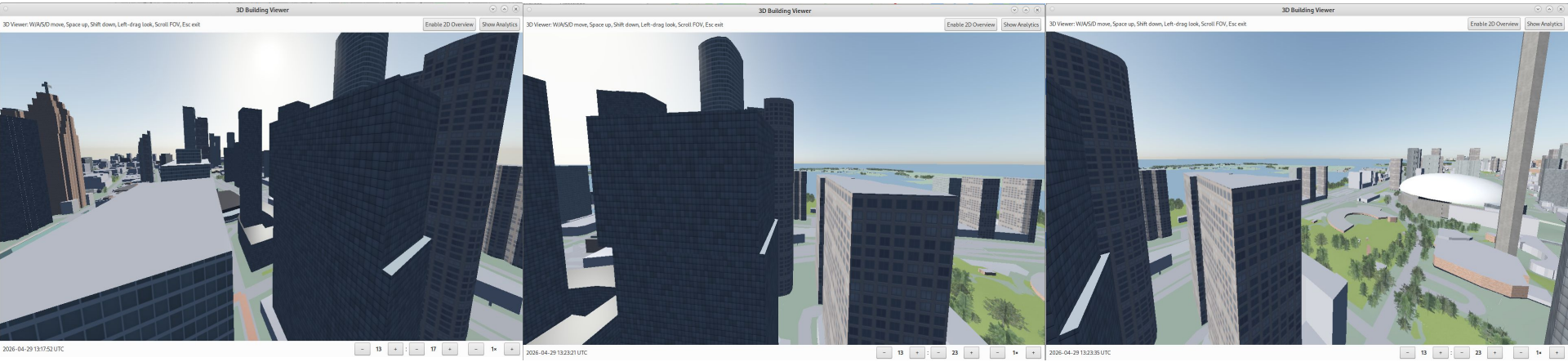

3D Building Viewer
Pedestrian: W/A/S/D walk, Hold Shift run, Left-drag look, Scroll FOV, Esc exit
Enable 2D Overview
Exit pedestrian view
Show Analytics
2026-04-29 13:08:41 UTC
13
08
1×


3D Building Viewer
Pedestrian: W/A/S/D walk, Hold Shift run, Left-drag look, Scroll FOV, Esc exit
Enable 2D Overview
Exit pedestrian view
Show Analytics
2026-04-29 13:09:25 UTC
13
09
1×

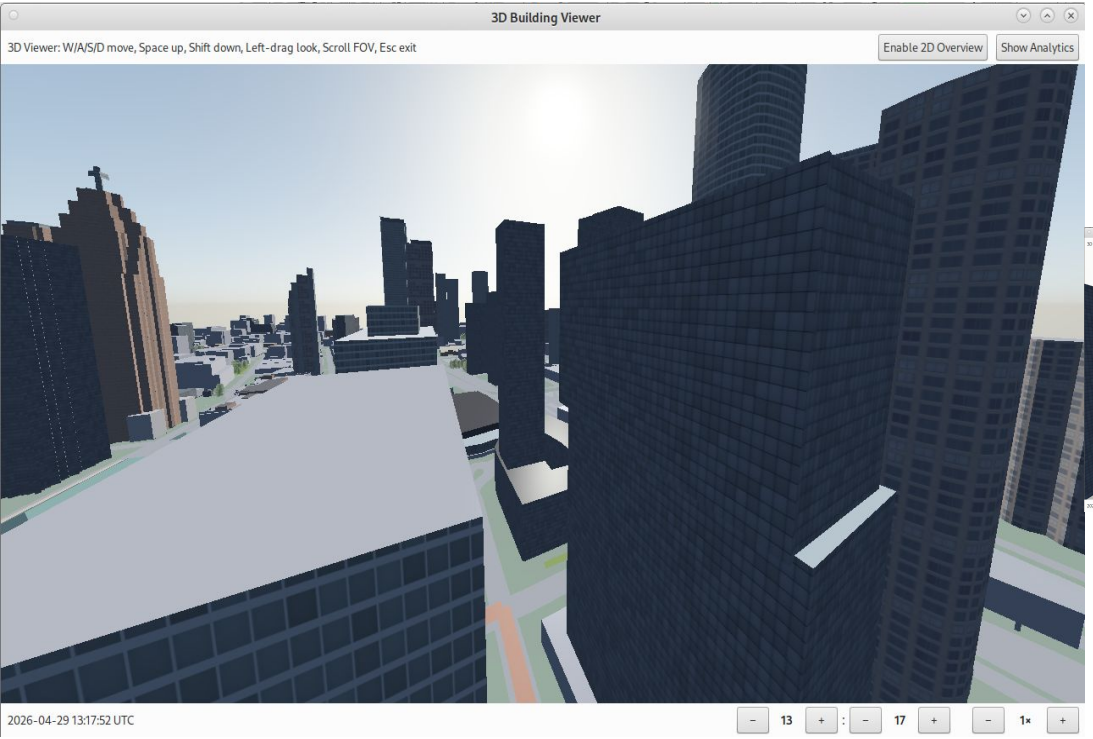


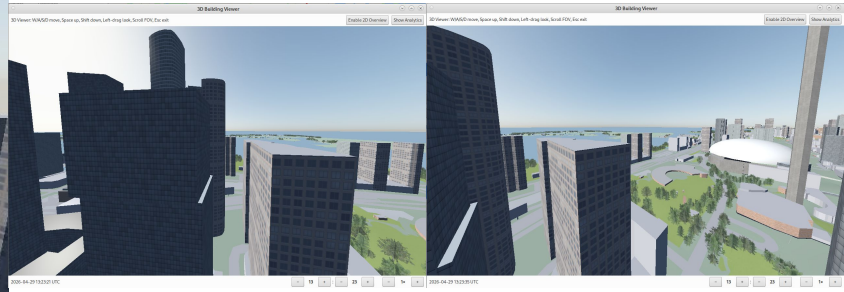

3D Building Viewer
Pedestrian: W/A/S/D walk, Hold Shift run, Left-drag look, Scroll FOV, Esc exit
Enable 2D Overview
Exit pedestrian view
Show Analytics
2026-04-29 13:10:34 UTC
13
10
1×

The walk looks quite nice. I forgot the check the views! Let's see how the views are?

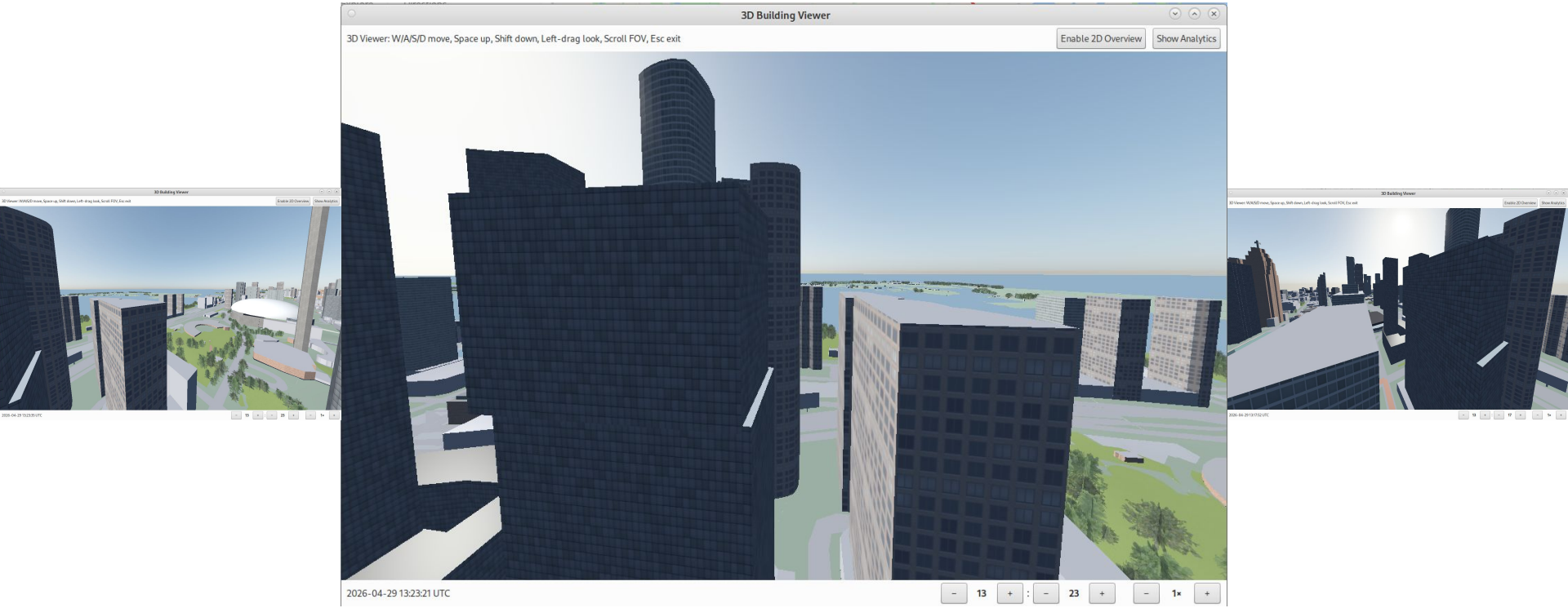


… here are the views from the rooftop restaurant!

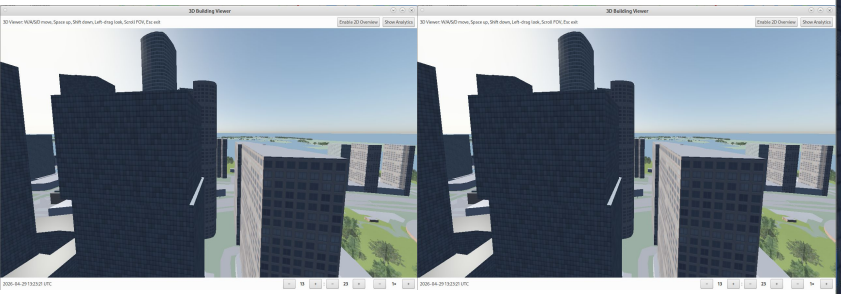


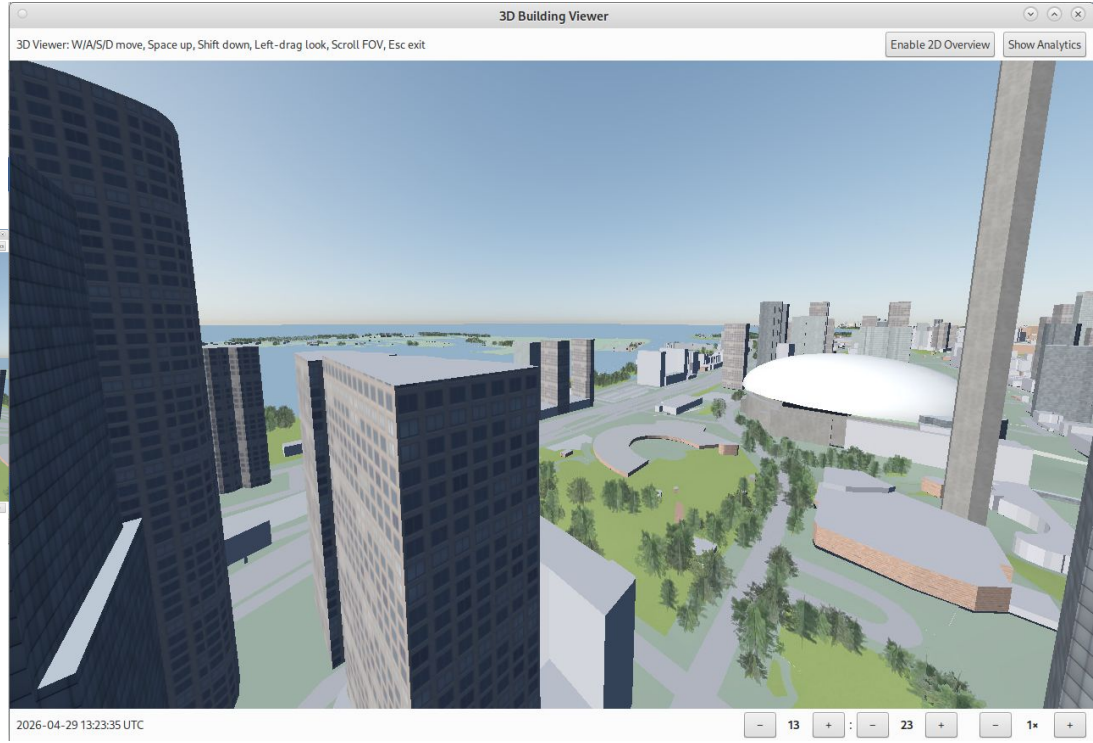


… oh dear … quite blocked in the SE direction!

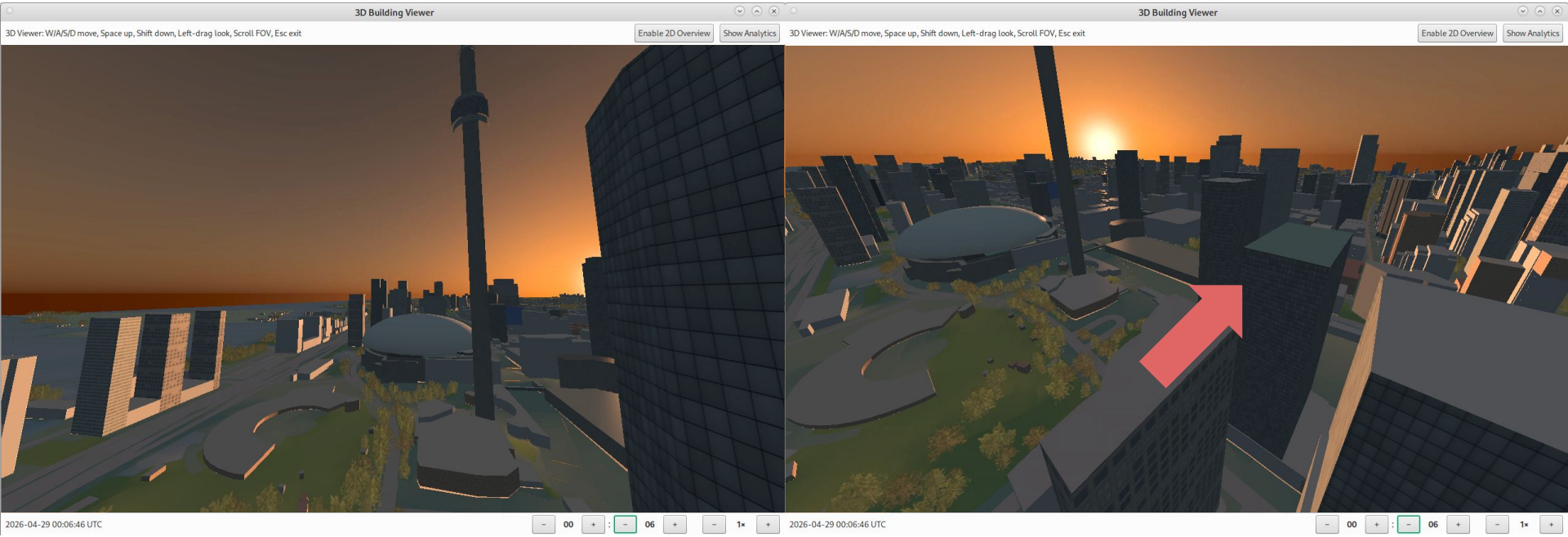


… uh oh … quite blocked in the S direction!





.. good views in the SW direction! I'll sit in the SW seating area! **But what about those sunset views?**



.. that’s unfortunate, the adjacent building blocks my view to the sunset! I’ll go back to search!

So you decide to look at another one! This one seems nice …

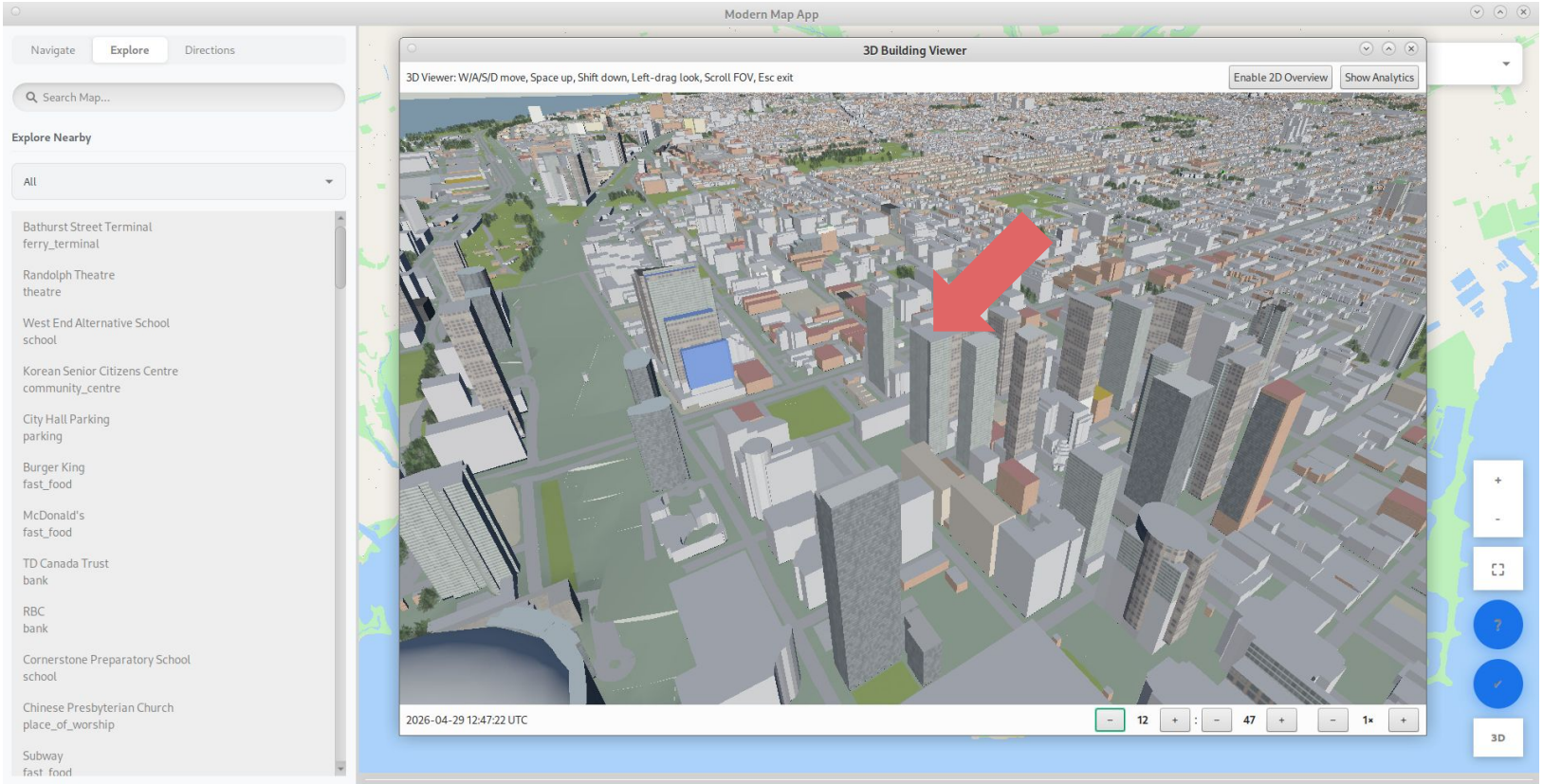
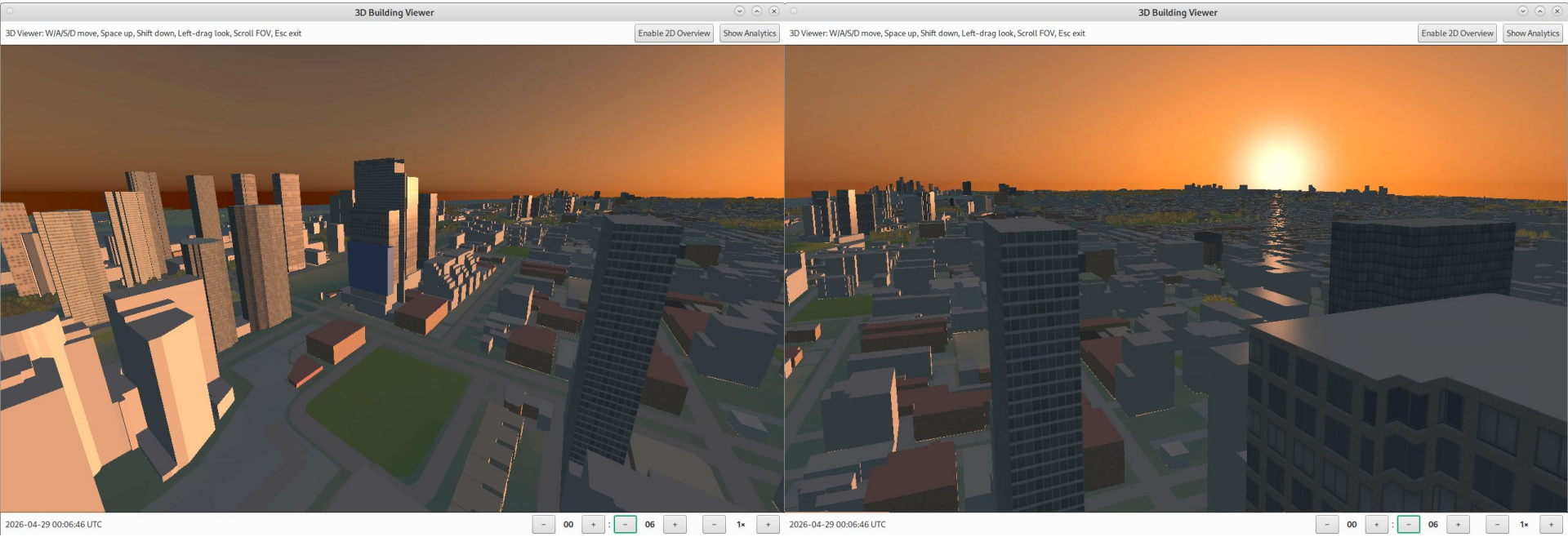


Figure 10: KOST resturant building under red arrow.



Ahh … much better!

Pitch & Vision

Goal: Make Urban Environments Easier to Understand and Explore

Extended Goal: *Making Spatial Intelligence more Accesible*

… realistic rendering, any city, any time.

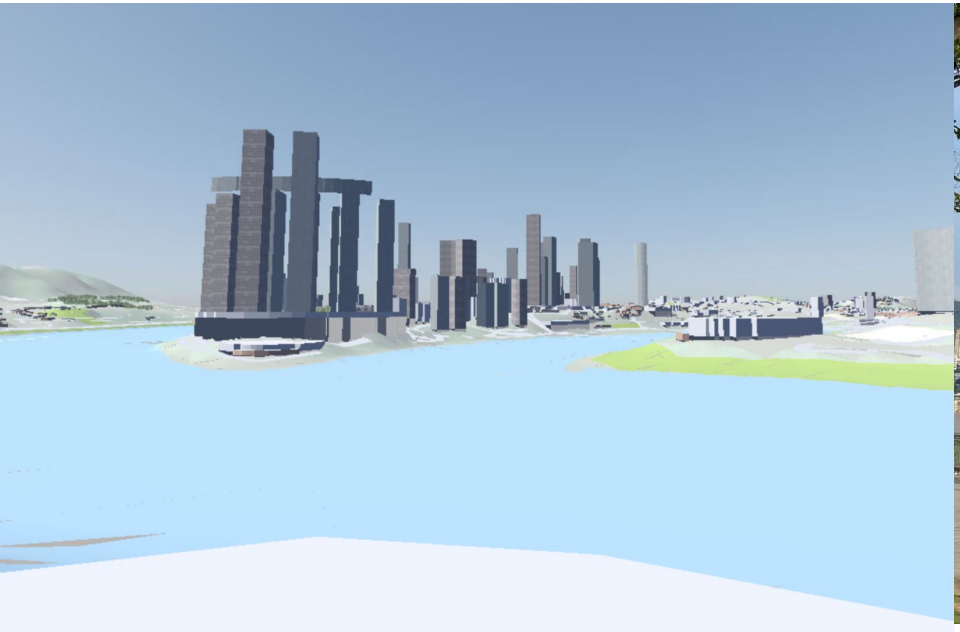


Figure 11: 3D viewer rendering (left) compared to Jet's photos taken last summer at Chongqing (right).

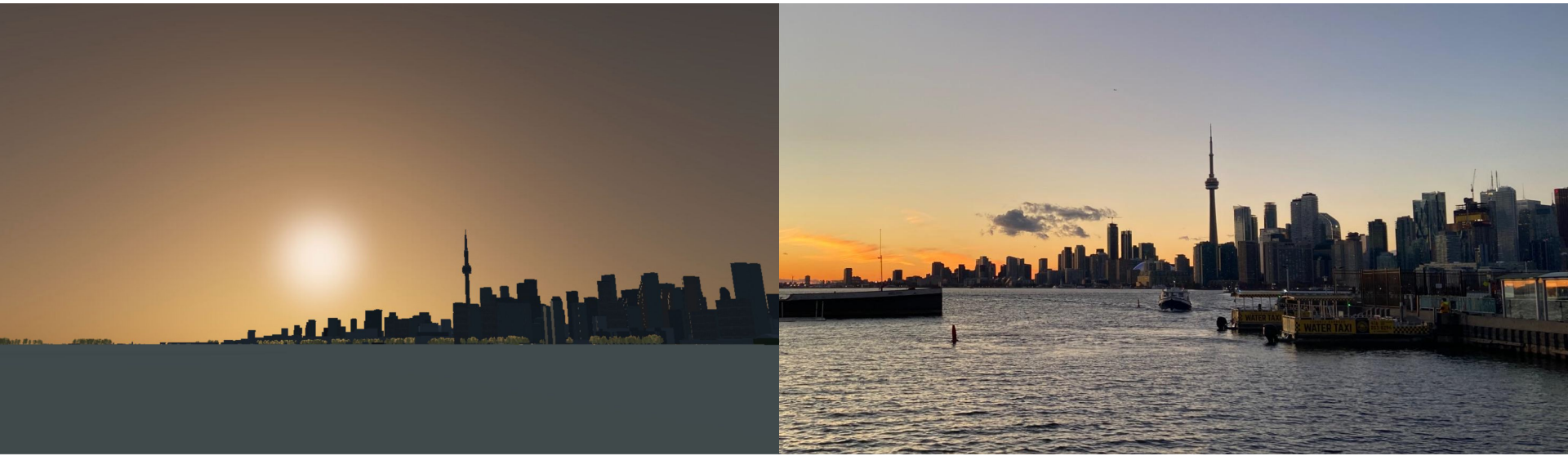
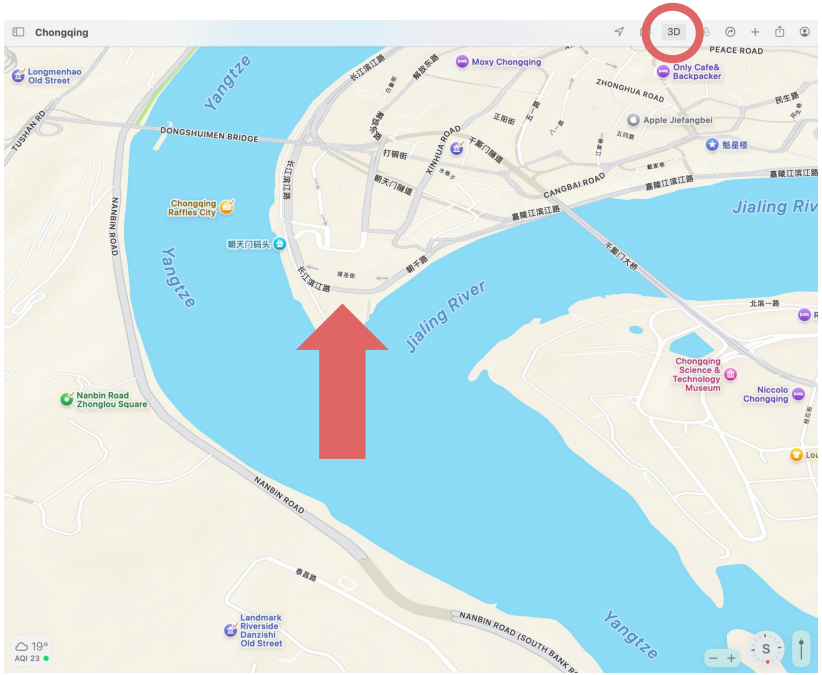


Figure 12: 3D viewer live view (left) compared to Jet’s photo last year on Toronto Island Pier (right)

Comparison with Apple Maps



There's nothing in 3D Mode!

What if I want to see another time?

Figure 13: Apple Maps 3D and "Look around" mode compared to our 3D mode and Pedestrian mode

Improvements

1 Data Sources
2 Visual Fidelity
3 Interactivity

Vertical Expansion

1a) Loading Model Meshes

Certain non-building landmarks are missing from OSM!

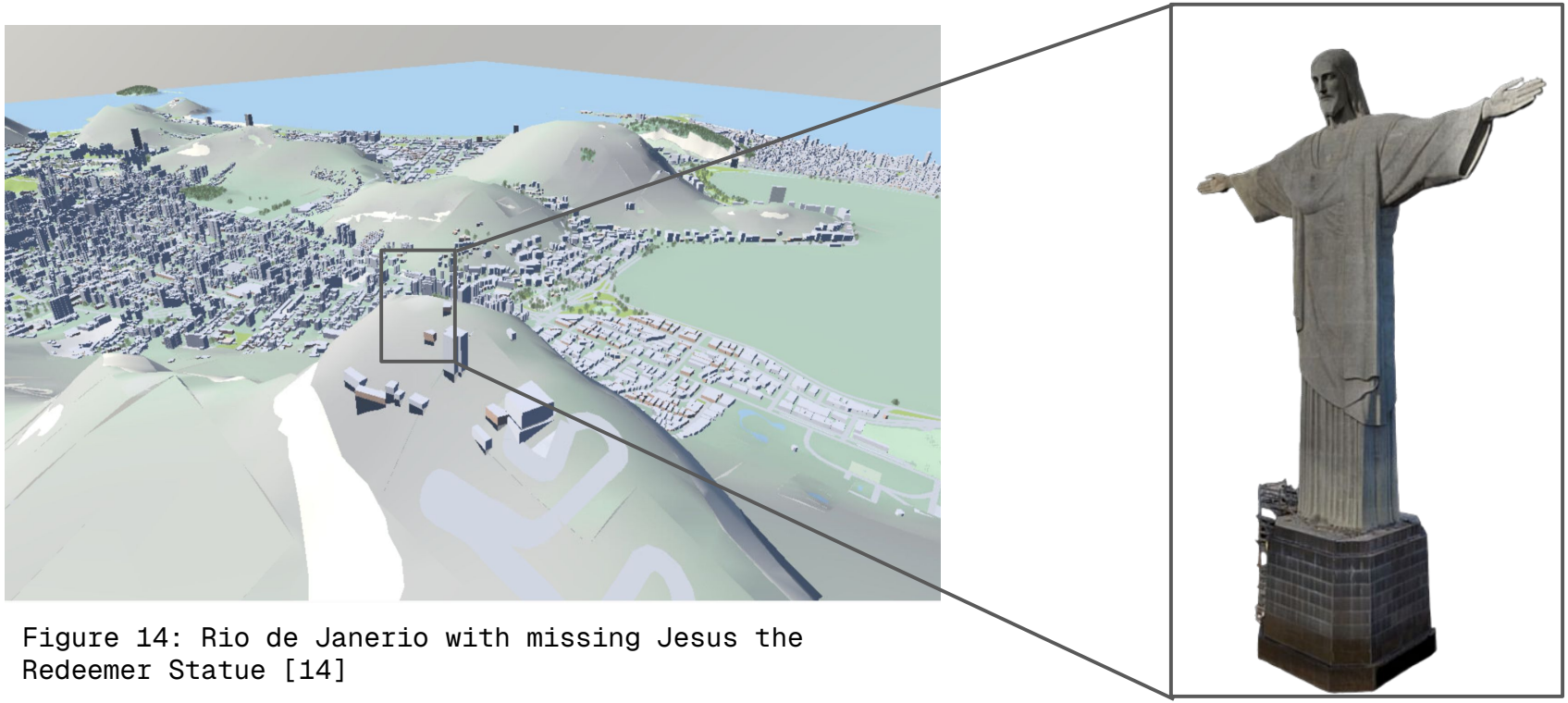


Figure 14: Rio de Janerio with missing Jesus the Redeemer Statue [14]

1b) Obtaining public transit data

- Streetcar Rails and Schedule?
- VIA/GO Schedule and Train tracks?

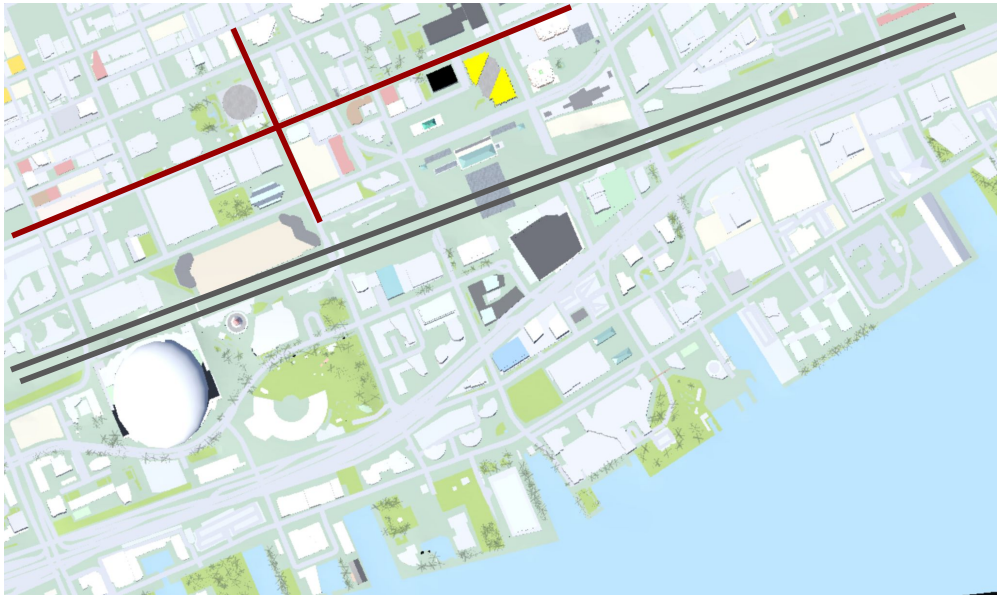


Figure 15: 2D overview mode showing missing transit tracks

2a) Physically Based Rendering

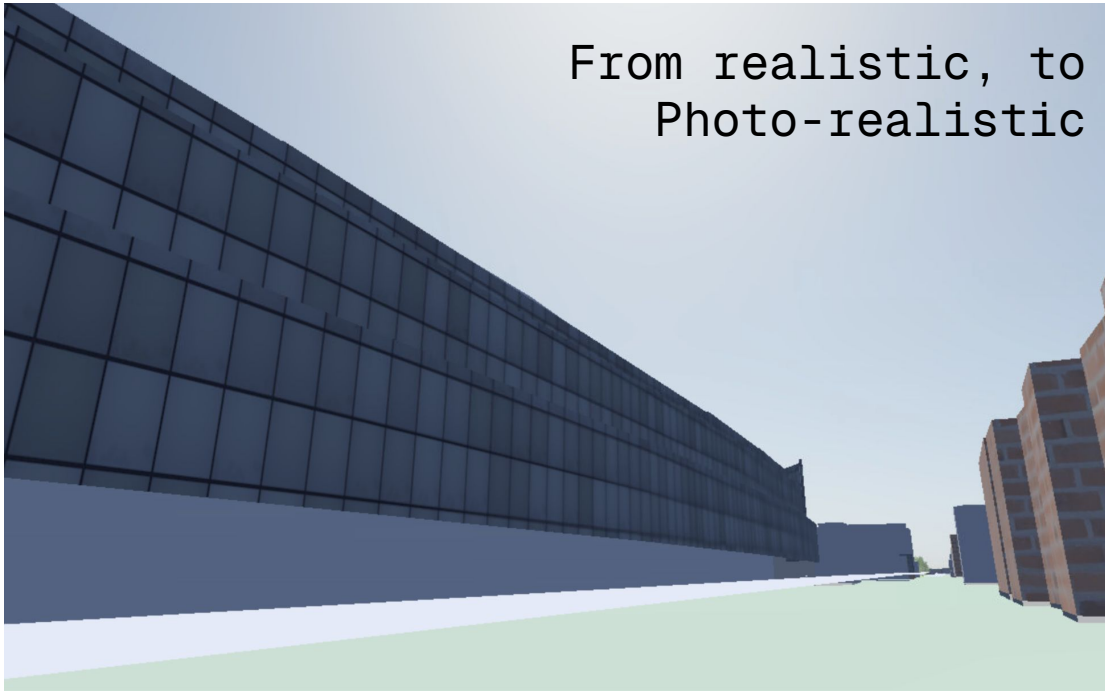
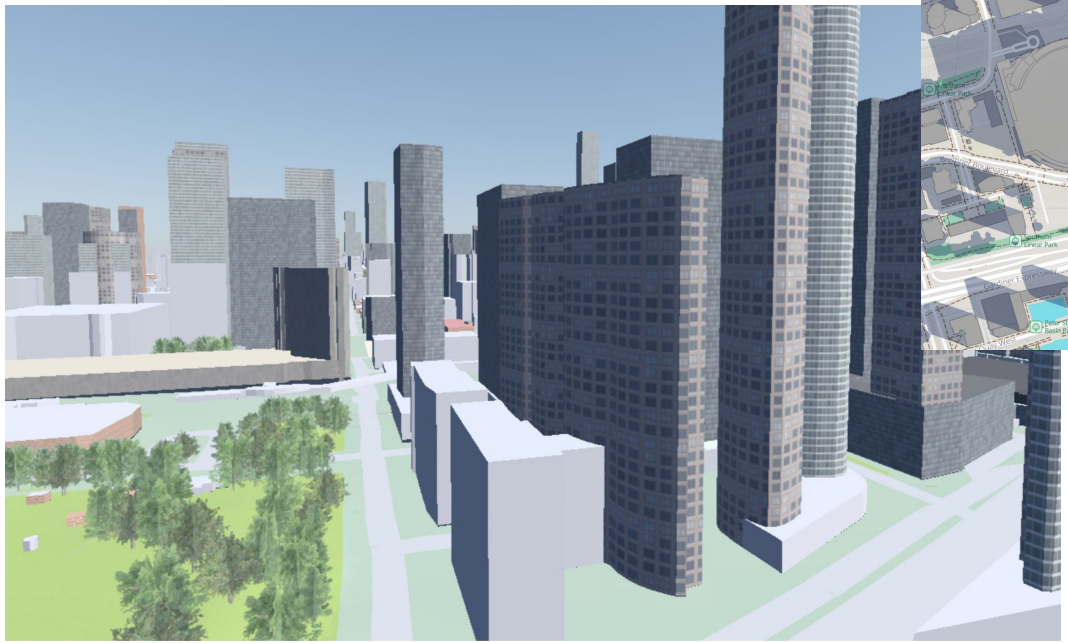


Figure 16: Comparison of glass facade of Art Gallery of Ontario in 3d renderer (left) and real life photo from Jet (right).

2b) Shadow maps



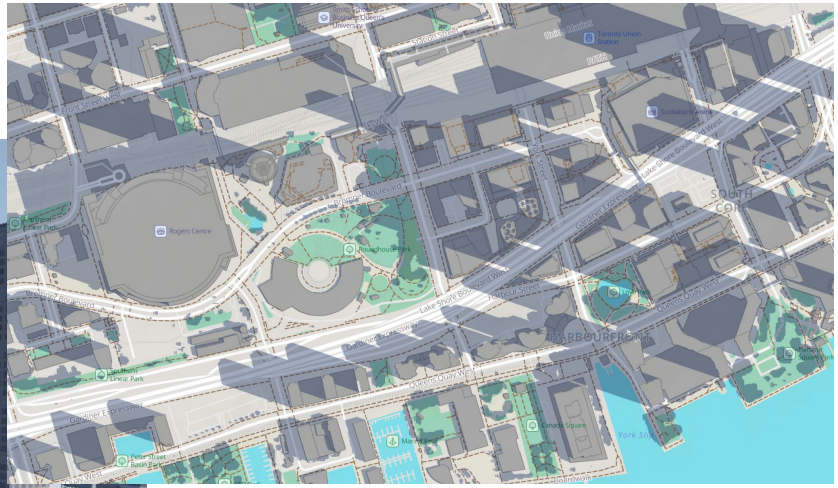


Figure 17: Comparison of shadows at approx 12pm between our 3d renderer and ShadowMap, a reference application [15]

Building and trees should cast shadows!

3a) Weather

Figure 18: Mock up of sunset during rainy weather

- What if it rains? Is it still a good sunset spot?
- Should I go somewhere else instead?

3b) Moving Vehicles

In some cities, public transit has become an iconic symbol!

This adds “liveliness” and “vibes” to a city.

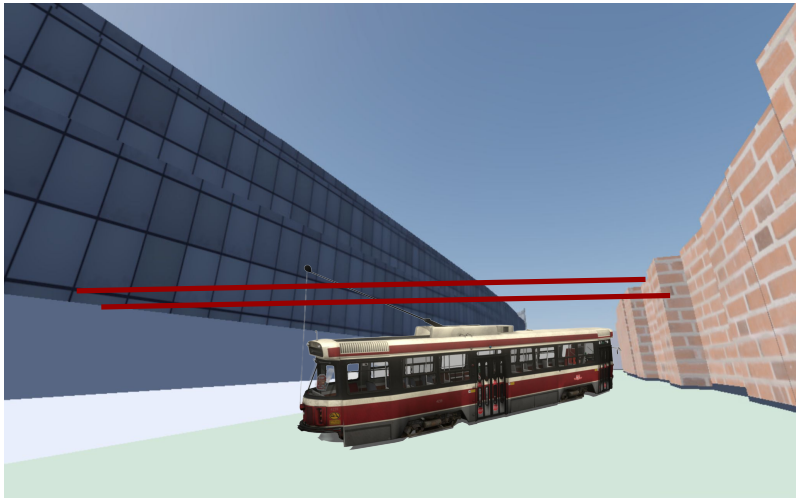


Figure 19: Mockup of 3d renderer with vehicles

Figure 20: Street cars in San Francisco is a historical symbol for the city [16]

Opportunities

1 Urban planner and Architects
2 Simulators for Driving, Flying, Disaster Planning

Horizontal Expansion

1) Digital Twin for Urban Planners

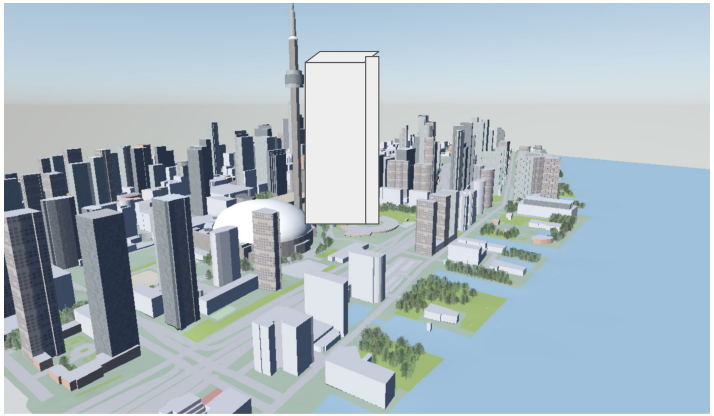


Figure 21: Mockup of 3d viewer supporting modifying structures and seeing the impact live

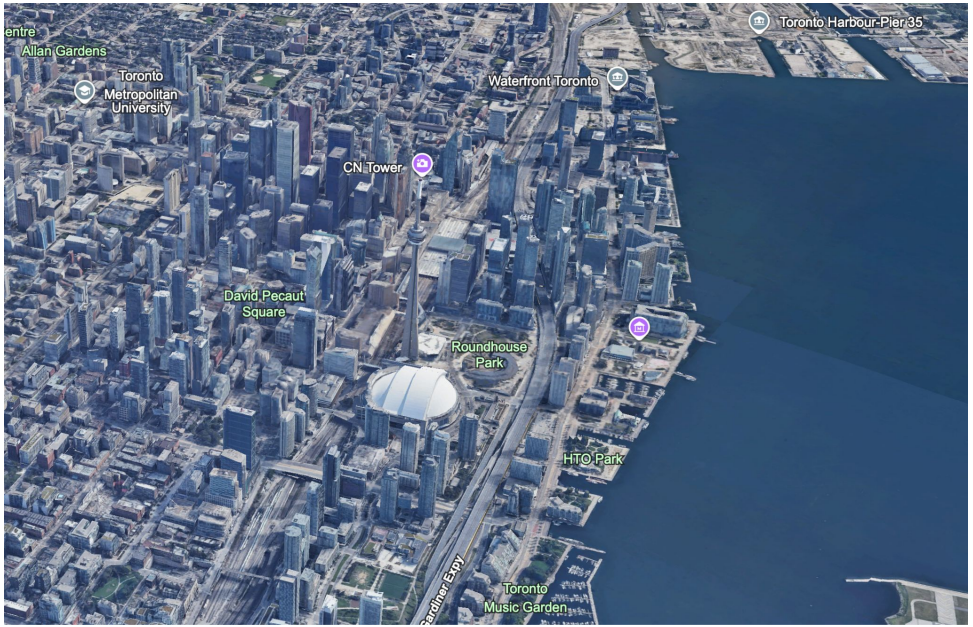


Figure 22: Screenshot of Toronto from Google Earth.

1) Market Validation

Digital twin is one of the fastest growing markets with a CAGR of 41%! [18]

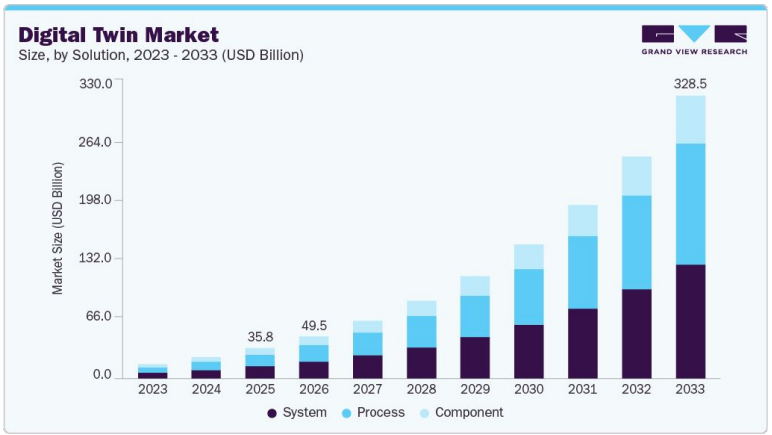


Figure 23: Digital twins market [18]

Figure 24: News report on digital twins GIS [19]

2) Flight Simulator

Photo-realistic renders + flight camera controls + physics

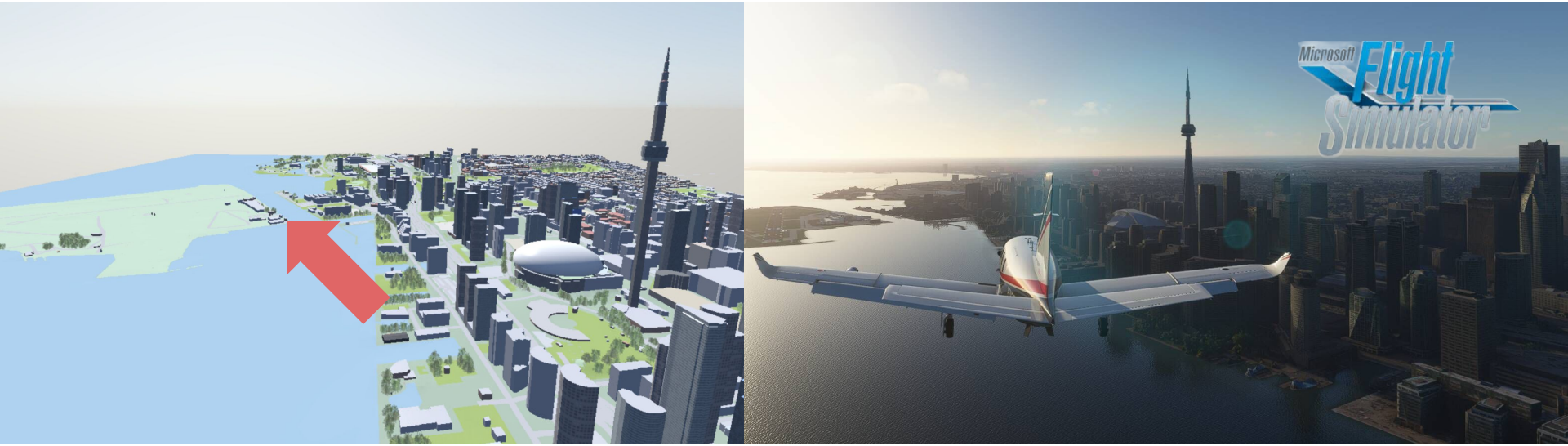


Figure 25: Comparison between 3d renderer fly over view and microsoft flight simulation

2) Market Segment: Urban Air Mobility

- Simulations market for autonomous vehicles research
- Enthusiasts, gamers (MS Flight), or even real pilots

eVTOL operations and ultimately realise vertiport functionality. The vertiport has been designed following the design specifications provided by the EASA. The contributions of this paper are as follows:

- A high-fidelity DT of a vertiport was created using Unreal Engine V4.27, AirSim V1.8, and Cesium for Unreal 2.0, incorporating real-time data through Python 3.17 interfaces. The environment accurately represents key infrastructure components for simulation and analysis.
- Adverse weather scenarios were simulated to study the wind effects on eVTOL take-off and landing. Wind deflection data enabled assessment of safety margins and operational limits

Figure 26: Research on using digital twin based on game engine for simulation in UAV research [22]

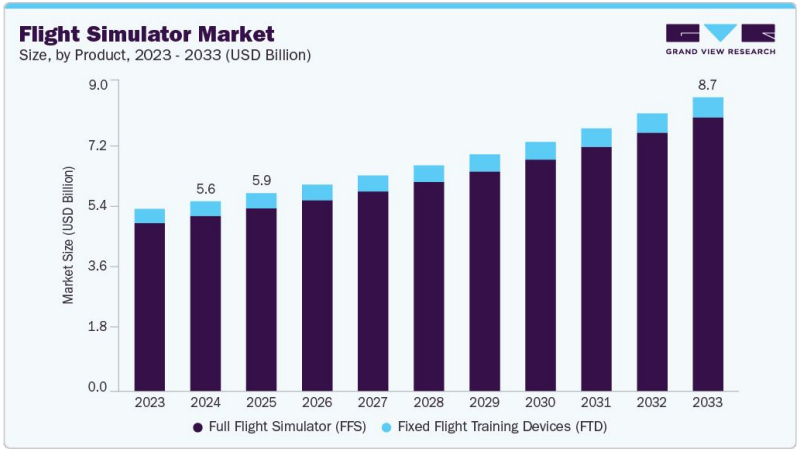
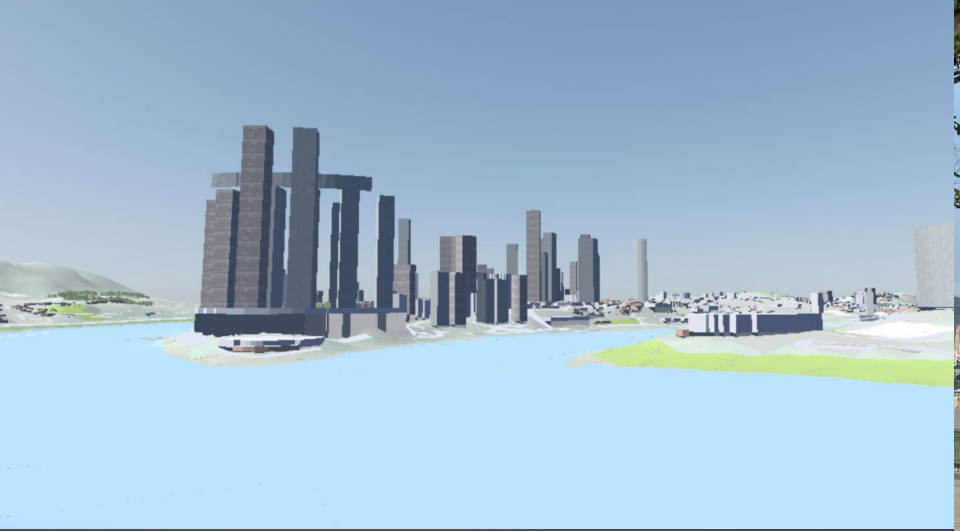


Figure 27: Market growth of flight simulator market [21].

~~**Goal**: Make Urban Environments Easier to Understand and Explore~~

Goal: Making Spatial Intelligence more Accesible

Thank you.
Questions?




WATER TAXI
WATER TAXI

References (1/4)

[1] A. Ali, J. Chiang, and R. Saluja. “cd-007 OP1 Presentation Slides.” As Seen on Quercus Submission.

[2] “ArcGIS Logo.” Roi4Cio. Accessed: Apr. 29, 2026. [Online]. Available: https://old.roi4cio.com/fileadmin/user_upload/ArcGIS.png

[3] NREL. “NREL Main Image.” GAstic Images. Accessed: Apr. 29, 2026. [Online]. Available: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSY-YrP97gJpsxxvXpAwja15IGjWW05rPqtDQ&s

[4] “OpenGL Logo Image.” Wikimedia Commons. Accessed: Apr. 29, 2026. [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/thumb/e/e9/Opengl-logo.svg/3840px-Opengl-logo.svg.png

[5] “GLM Logo.” chromium Git repositories - Git at Google. Accessed: Apr. 29, 2026. [Online]. Available: https://chromium.googlesource.com/external/github.com/g-truc/glm/

[6] J. Nielsen, Usability Engineering. Elsevier, 1994. Accessed: Feb. 25, 2026. [Online]. Available: https://dl.acm.org/doi/pdf/10.5555/2821575

References (2/4)

[7] “GLSL Logo.” Wikimedia Commons. Accessed: Apr. 29, 2026. [Online]. Available: https://commons.wikimedia.org/wiki/File:GLSL_Logo_(Unofficial).svg

[8] “DownDetector.” OSM Logo. Accessed: Apr. 29, 2026. [Online]. Available: https://cdn2.downdetector.com/static/uploads/logo/openstreetmap.png

[9] “Shader Ball 1.” Github Repository. Accessed: Apr. 29, 2026. [Online]. Available: https://user-images.githubusercontent.com/10408010/48678255-b7612500-eb78-11e8-8fef-09cf0066ee5c.png

[10] “Shader Ball 2.” Github Repository. Accessed: Apr. 29, 2026. [Online]. Available: https://user-images.githubusercontent.com/10408010/48678247-add7bd00-eb78-11e8-9ac8-e418ee4225bf.png

[11] evan4129. “LOD/Billboard Summer Trees Pack by EFX (@evan4129).” Sketchfab. Accessed: Mar. 1, 2026. [Online]. Available: https://sketchfab.com/3d-models/lodbillboard-summer-trees-pack-d88d67f701c1438cbf5098ff27d4e1f2

[12] AmbientCG. “Decal Types, Textures” AmbientCG. Accessed Apr 20, 2026 online. Available: https://ambientcg.com/list?type=material%2Cdecal%2Catlas&sort=popular

References (3/4)

[13] “Introduction to Shading.” ScratchAPixel. Accessed: Apr. 29, 2026. [Online]. Available: https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-shading/shading-normals.html

[14] “Jesus the Redeemer” SketchFab. Accessed: Apr 29, 2026 [Online]. Available: https://skfb.ly/CRws

[15] “Toronto shadow” ShadowMap. Accessed: Apr 29, 2026 [Online]. Available: https://shademap.app/@43.64169,-79.38425,15.65709z,1777470956150t,0b,0p,0m

[16] “San Francisco Streetcar” Wikipedia. Accessed: Apr 29, 2026 [Online]. Available: https://en.wikipedia.org/wiki/File:Cable_Car_No._1_and_Alcatraz_Island.jpg

[17] “Level of detail • OSM Buildings.” OSM Buildings. Accessed: Apr. 29, 2026. [Online]. Available: https://osmbuildings.org/blog/2018-02-28_level_of_detail/

[18] Digital Twins Market. Gradview Research. Accessed Apr 29, 2026. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/the-global-flight-simulator-market

[19] “Digital Twins, GIS Help Cities React to Changes in Real Time” Government Technology. Accessed: APr 29, 2026. Available: https://www.govtech.com/opinion/digital-twins-gis-help-cities-react-to-changes-in-real-time

References (4/4)

[20] “CN Tower in Toronto gone missing while doing challange” MS Flight Forum. Accessed: Apr 29, 2026. [Online] Available: https://forums.flightsimulator.com/t/cn-tower-in-toronto-gone-missing-while-doing-challange/340757

[21] Flight Simulator Market. Gradview Research. Accessed Apr 29, 2026. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/the-global-flight-simulator-market

[22] UAV Operations and Vertiport Capacity Evaluation with a Mixed-Reality Digital Twin for Future Urban Air Mobility Viability. MDPI. Accessed Apr 29, 2026. [Online]. Available: https://www.mdpi.com/2504-446X/9/9/62

[23] D. S. Johnson and L. A. McGeoch, “The Traveling Salesman Problem: A Case Study in Local Optimization,” in *Local Search in Combinatorial Optimization*, E. H. L. Aarts and J. K. Lenstra, Eds. Princeton, NJ, USA: Princeton University Press, 1997, pp. 215–310.

[24] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[25] D. David L. Applegate, R. Robert E. Bixby, V. Václav Chvátal, and W. William J. Cook, “Finding cuts in the TSP (A preliminary report),” *Operations Research Letters*, vol. 20, no. 5, pp. 199–206, 1997.