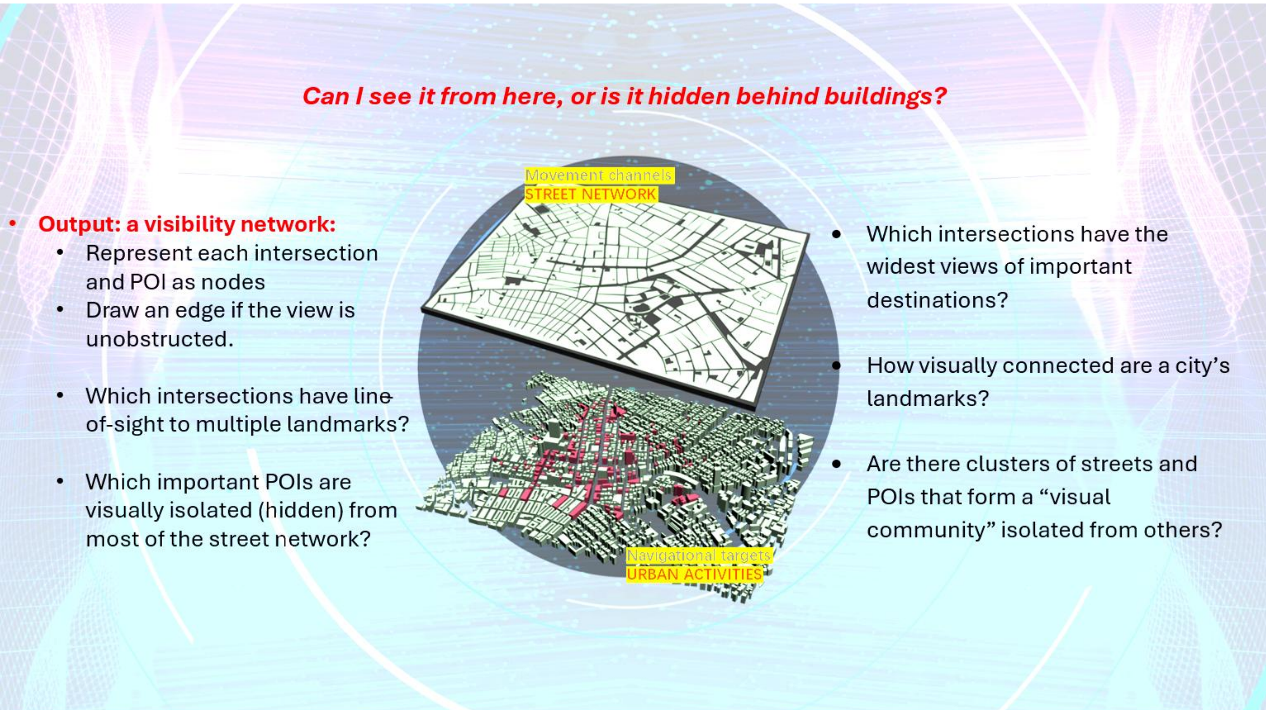
POI VizNet Tutorial – CUPUM 2025 Visual Intelligence Workshop

Abstract

This tutorial introduces **POI VizNet**, a QGIS-based toolkit for constructing visibility networks between urban points of interest (POIs) and street intersections.

Materials: Click

Introduction to POI VizNet and Its Purpose

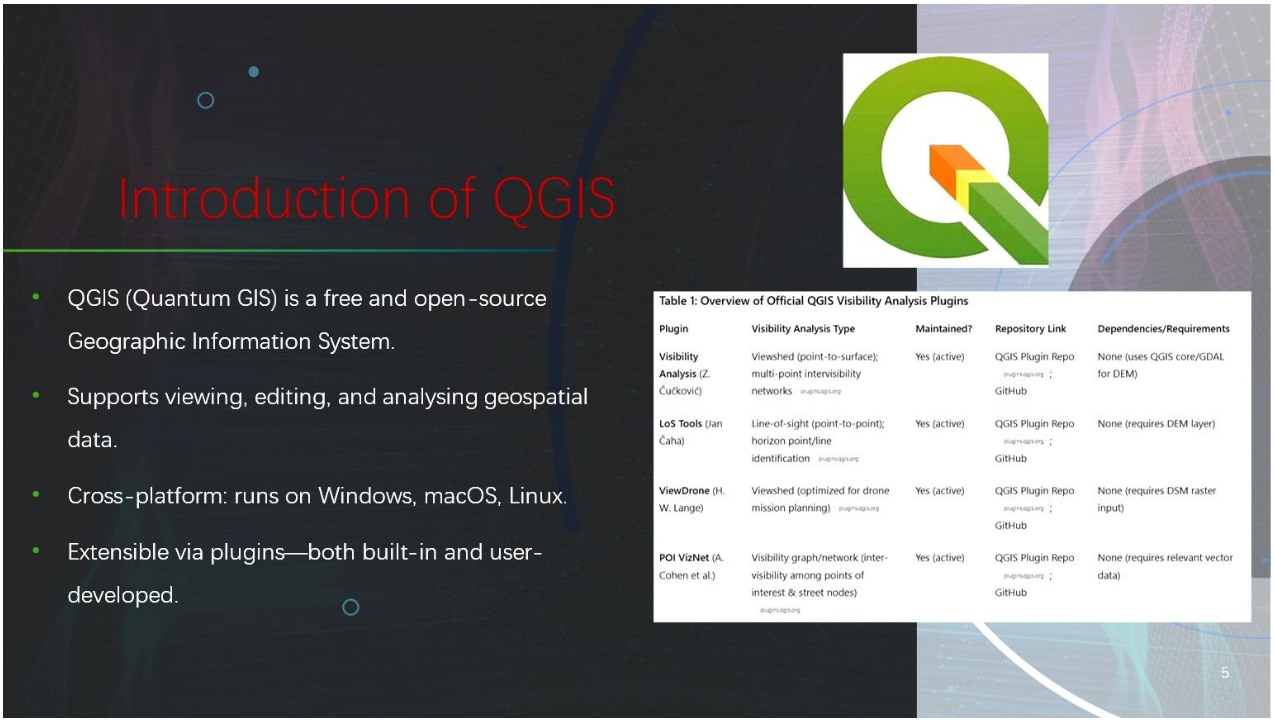


Imagine standing at various street intersections in a city and asking: "Can I see the museum from here, or is it hidden behind buildings?" POI VizNet would represent each intersection and the museum as nodes, and draw an edge if the view is unobstructed. Repeating this for all intersections and all important POIs yields a visibility network. By analyzing this network, one could find which intersections have line-of-sight to multiple landmarks, or conversely, which important POIs are visually isolated (hidden) from most of the street network.

Unlike localized visibility analyses such as isovists or viewsheds (which compute the visible area from one viewpoint), a visibility network is global, connecting many points and capturing the overall structure of visual access in the city. By this toll we can answer the questions including:

- *Which intersections have the widest views of important destinations?*
- *How visually connected are a city's landmarks?*
- *Are there clusters of streets and POIs that form a "visual community" isolated from others?*

Installation and Setup Instructions



Plugin	Visibility Analysis Type	Maintained?	Repository Link	Dependencies/Requirements
Visibility Analysis (Z. Čučković)	Viewshed (point-to-surface); multi-point intervisibility networks plugins.qgis.org	Yes (active)	QGIS Plugin Repo plugins.qgis.org ; GitHub	None (uses QGIS core/GDAL for DEM)
LoS Tools (Jan Čaha)	Line-of-sight (point-to-point); horizon point/line identification plugins.qgis.org	Yes (active)	QGIS Plugin Repo plugins.qgis.org ; GitHub	None (requires DEM layer)
ViewDrone (H. W. Lange)	Viewshed (optimized for drone mission planning) plugins.qgis.org	Yes (active)	QGIS Plugin Repo plugins.qgis.org ; GitHub	None (requires DSM raster input)
POI VizNet (A. Cohen et al.)	Visibility graph/network (inter-visibility among points of interest & street nodes) plugins.qgis.org	Yes (active)	QGIS Plugin Repo plugins.qgis.org ; GitHub	None (requires relevant vector data)

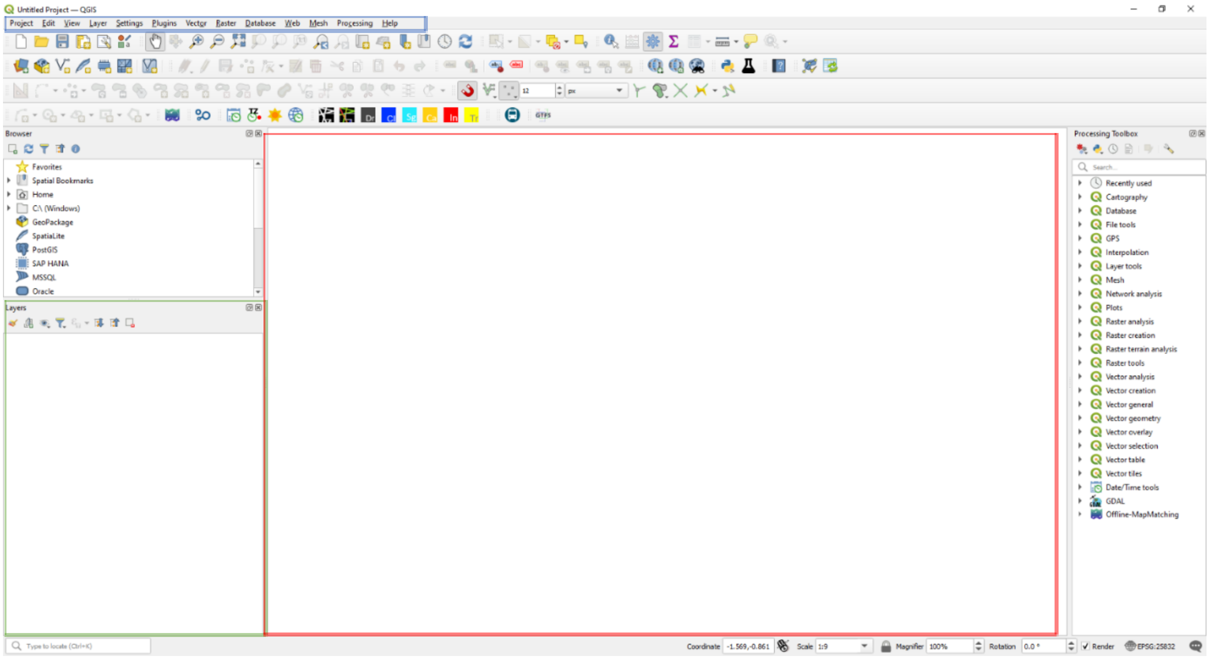
POI VizNet is implemented as a **plugin for QGIS** (Quantum GIS), a popular open-source desktop GIS platform. The tool is written in Python and easily installable through QGIS's plugin interface. Follow these steps to set up POI VizNet:

1. **Install QGIS (if not already):**

   Go to https://www.qgis.org/en/site/forusers/download.html and install version 3.34 as appropriate for your computer.

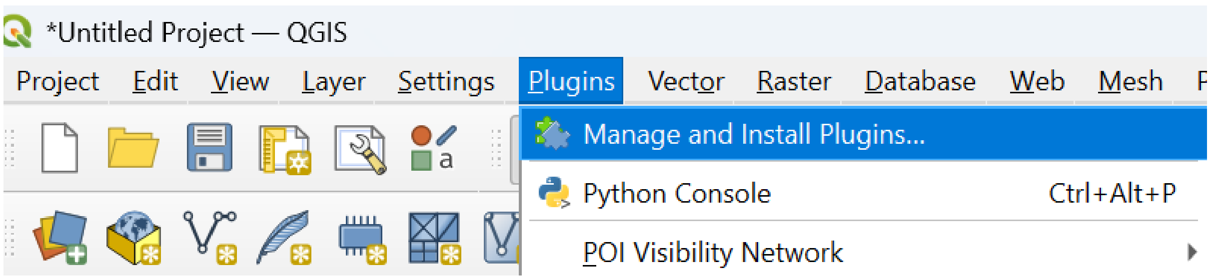
2. **Open new project**

   Open QGIS and start a new project. Your graphical user interface should look approximately like shown below.



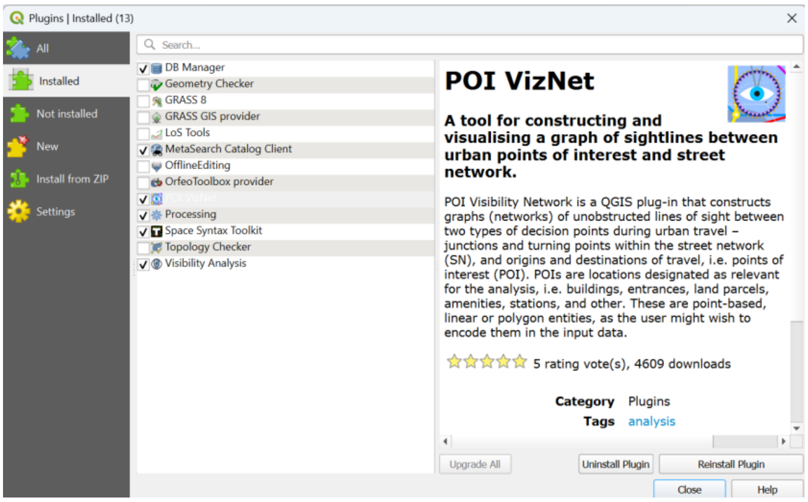
3. **Open the QGIS Plugin Manager:**

   - Launch QGIS and navigate to the menu **Plugins → Manage and Install Plugins…**. In the Plugins dialog, search for "POI VizNet" (the plugin might be listed as "POI Visibility Network").

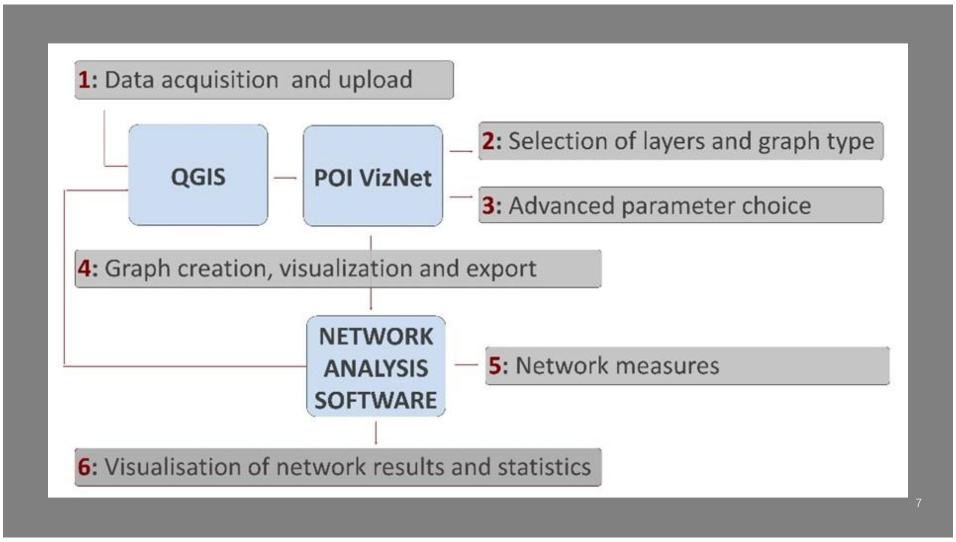


   - Find POI VizNet in Repository: The plugin is published in the official QGIS Plugins Repository, so it should appear in the search results.

- Click on "POI VizNet" in the list. You should see the plugin's name and description.
- Install the Plugin: Click the Install button. QGIS will download and enable the POI VizNet plugin. After installation, you may need to restart QGIS (if prompted).
- Verify Installation: After QGIS restarts, go to Plugins → Installed and ensure POI VizNet is listed and checked as enabled.

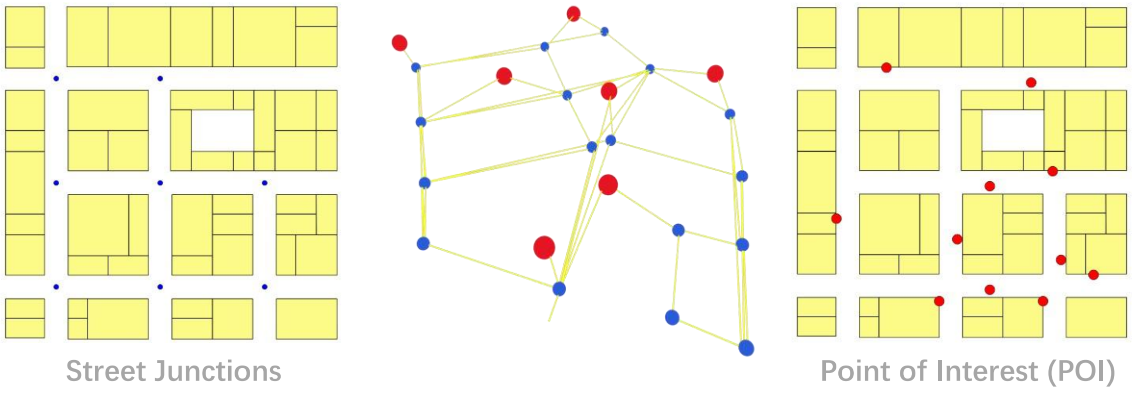


- POI VizNet is **accessible and maintained via QGIS's plugin manager**. If you have an internet connection, QGIS will fetch the plugin automatically. If offline or behind firewalls, you can manually download the plugin from the QGIS plugin repository website or the GitHub repository and install it as a zip package.



Data Requirements and Formats

Two types of decision points within the urban layout



Before using POI VizNet, you must prepare the necessary spatial datasets. Acceptable formats include Shapefile.

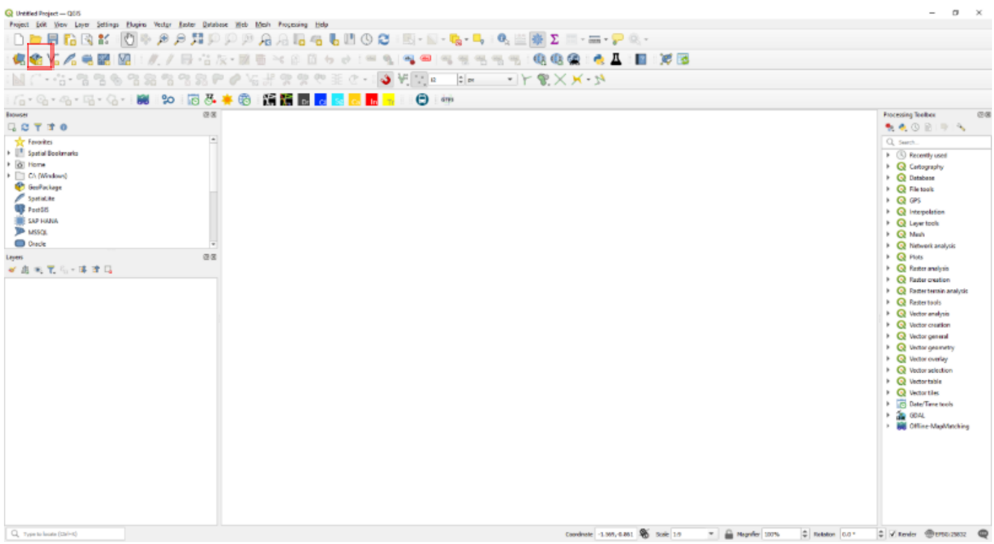
In POI VizNet, two main types of “decision points” are considered as graph nodes:

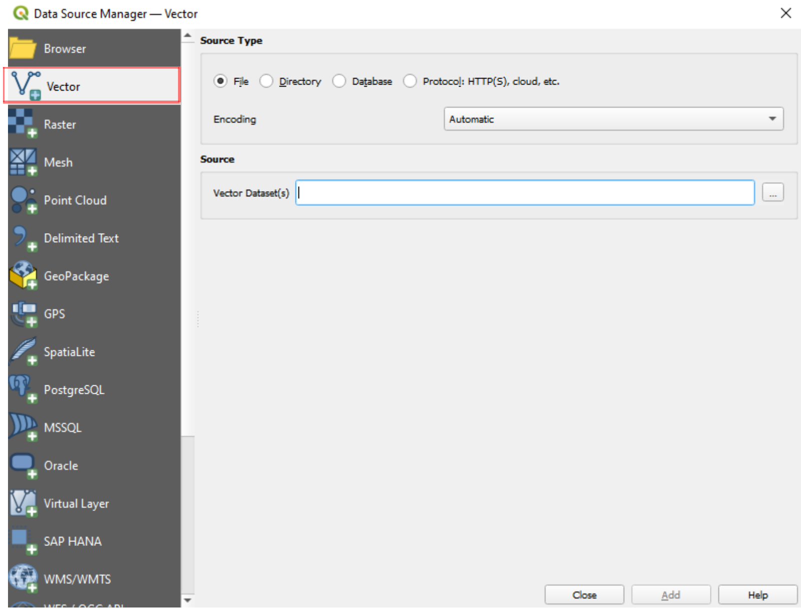
(1) **Street network nodes** – junctions or turning points along the street network,

(2) **POI locations** – origins or destinations of travel (these could be specific buildings, parcels, entrances, or any important locations).

We should import the data that we are going to use for the analysis (Buildings.shp, SideWalk.shp, POI.shp).

- Open Data Source Manager – Vector – [navigate to location of .shp files] – Add “Buildings.shp, POI.shp, SideWalk.shp”

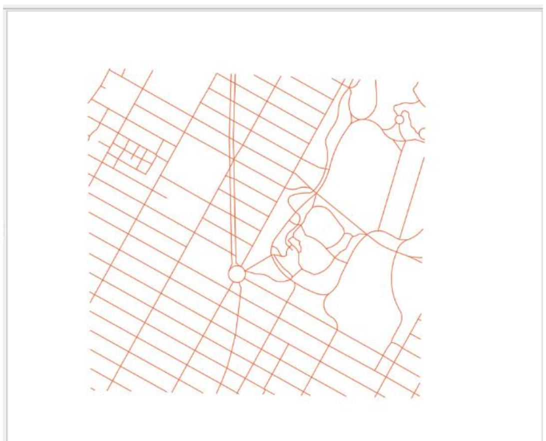




The plugin operates on three key layers which must be loaded into QGIS **before** running the analysis

- **Street Network Layer (Polyline):** This should be a line dataset representing the street centrelines of the area of interest. Each street segment is typically a line feature. The plugin will use this to derive street intersection points (graph nodes).
  - Ensure the street network is topologically correct.

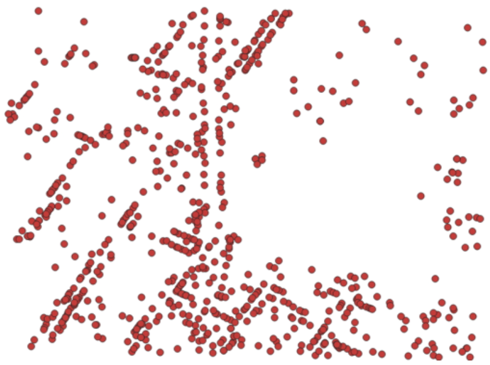
Tip: You can obtain street network data from OpenStreetMap (e.g., via QGIS's QuickOSM plugin or Geofabrik downloads) or city open data portals.



- **Building Footprints Layer (Polygon or Polyline):** A polygon layer of building outlines (footprints) is required to serve as **visibility constraints** (obstacles). If you only have building outlines as polylines (edges), POI VizNet will convert them to filled polygons internally.

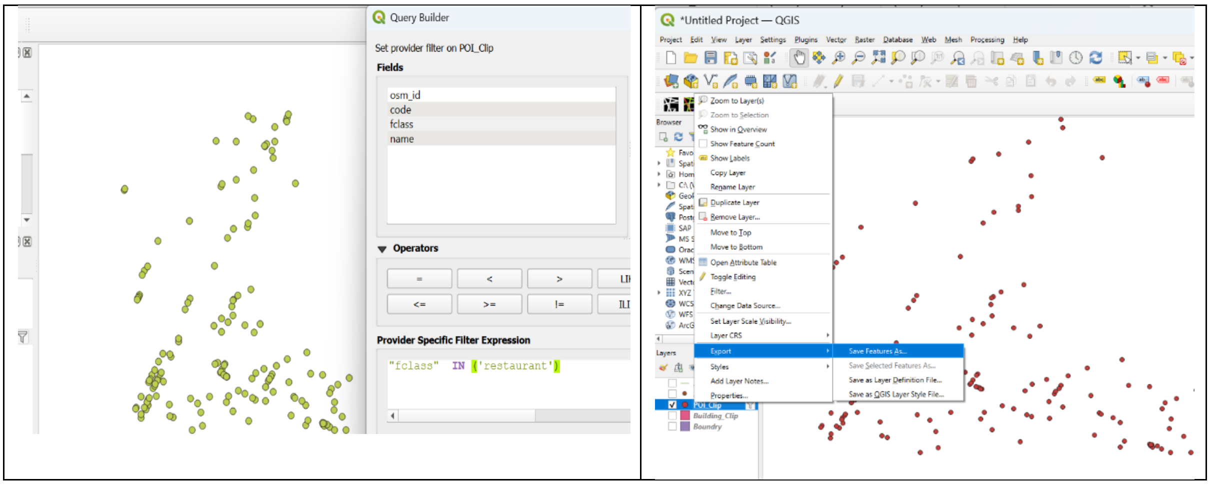
- note that smaller obstacles like trees or minor structures are not accounted for – the analysis assumes open space except where building polygons exist.

- **POI Layer (Point/Polyline/Polygon):** A layer of "points of interest" representing key locations or destinations (e.g., coordinates of attractions, transit stops, etc.).



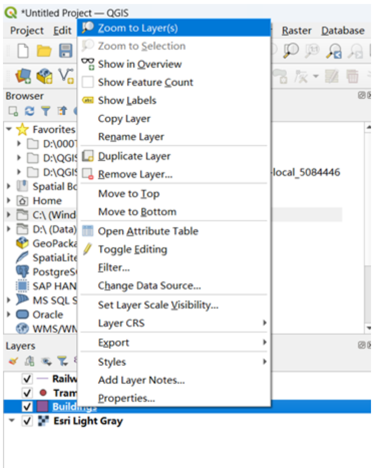
Tip: POI VizNet will convert any non-point POI features into points (by taking centroids) because the visibility graph nodes must be point locations. If a POI (like a shop or entrance) is located *inside* a building polygon (which is common if your POI data is a point placed at a building's interior), the plugin will project it to just outside the building's edge to simulate a visible location on the façade. This ensures the POI is treated as "just in front of" its building for line-of-sight purposes.

Too many POIs (hundreds or thousands) can make the graph dense and complex. The plugin doesn't inherently distinguish types of POIs, but you can run it on different POI layers or filter your layer before running.

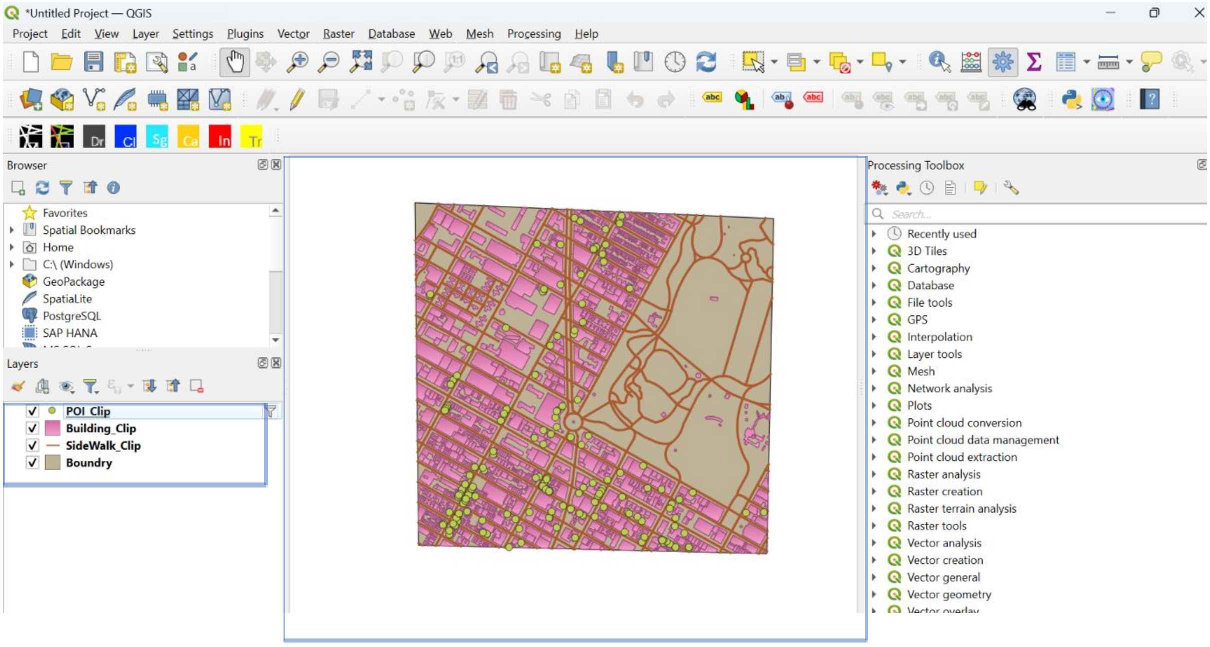


Coordinate Reference System (CRS): It is crucial that **all layers use the same coordinate system**, especially a projected (Cartesian) coordinate system for distance calculations. POI VizNet will automatically **reproject all input layers to WGS 84 / Pseudo-Mercator (EPSG:3857)** at the start of the run.

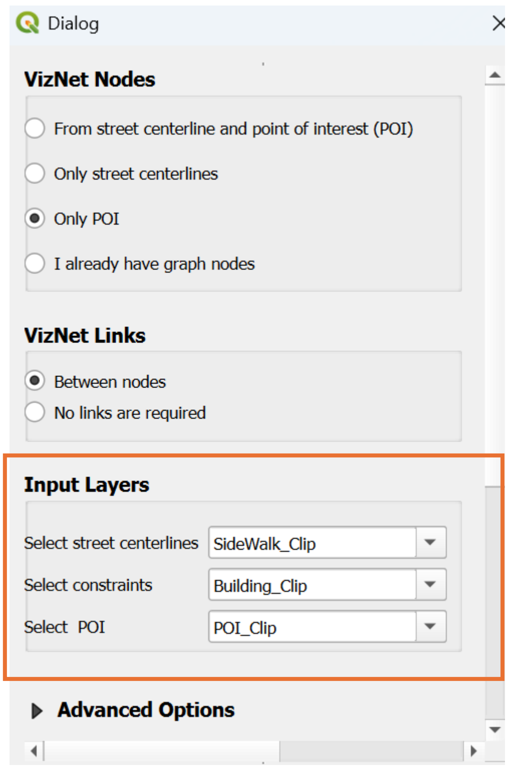
- Right click on one of those layers and select "Zoom to layer".



You can now see the buildings, the road lines and the POIs in the study area both in the map and the panel (position and colour of items might vary).



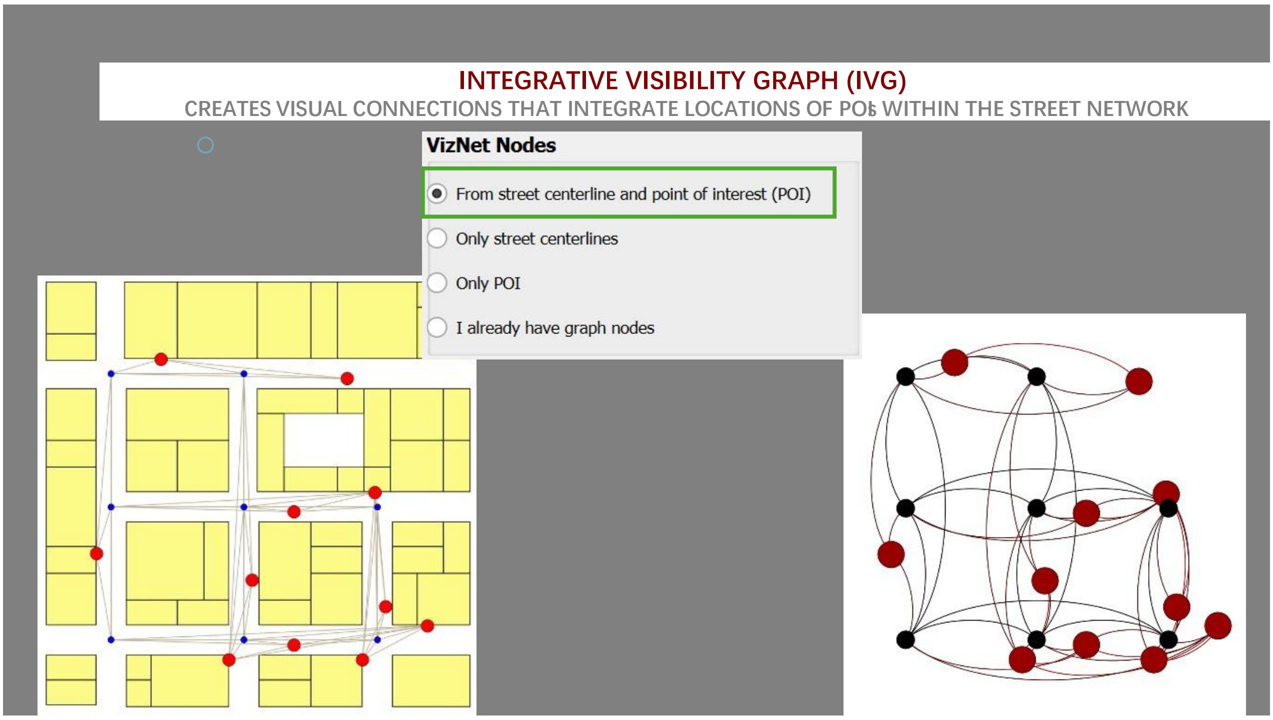
In the POI VizNet dialog, specify the input layers for **Streets**, **Buildings**, and (if needed) **POIs**. There will be fields or dropdown menus to select each required layer from the ones already loaded in QGIS (e.g., “Select street network layer”, “Select building layer”, “Select POI layer”).



Visibility Graph Types (Analysis Modules)

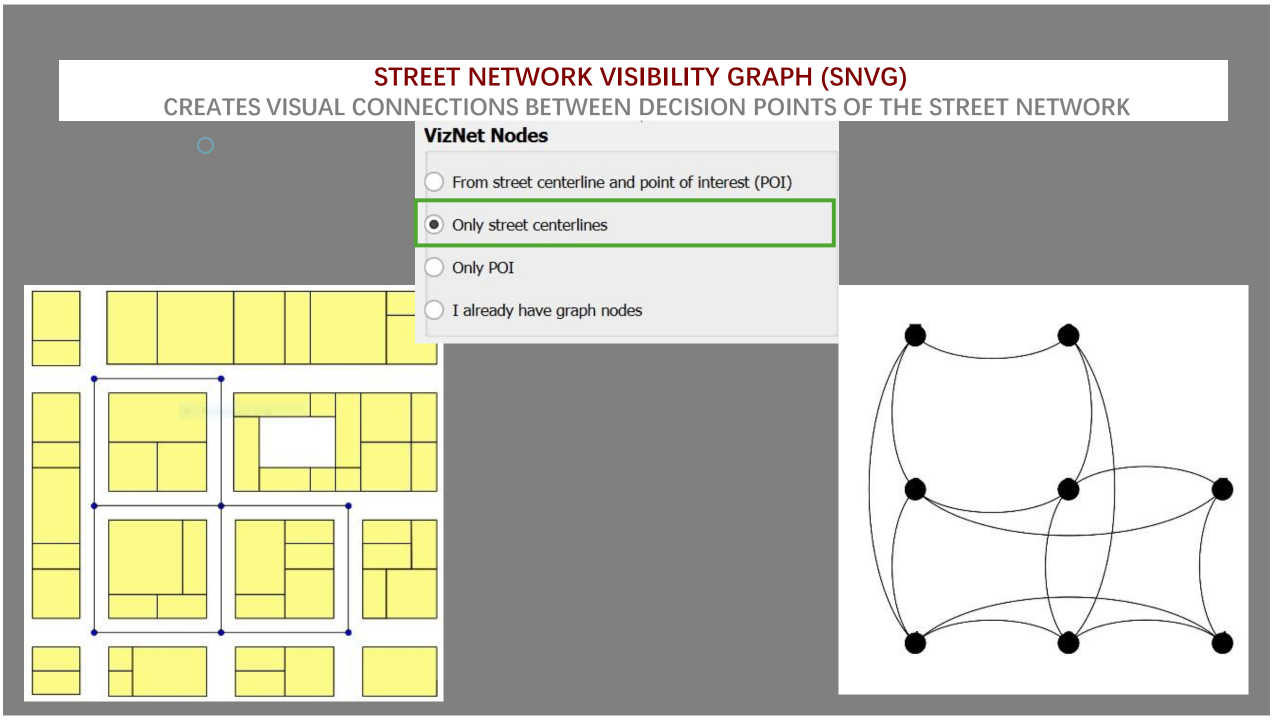
There are **three main analysis modules**, each corresponding to a different graph interpretation. You should choose which types of nodes will be included based on your analysis. This determines which pairs of points the tool will attempt to connect with sight lines.

1. **Integrative Visibility Graph (IVG):** This mode integrates POIs with the street network.



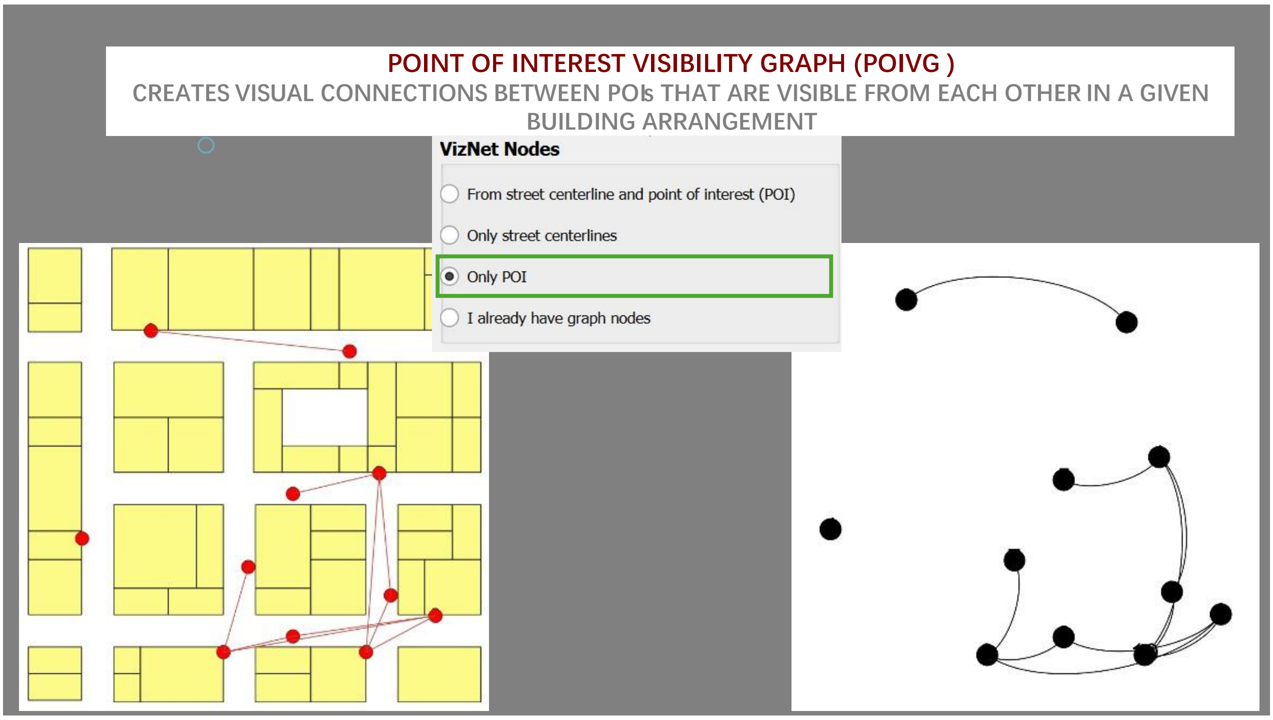
   - The idea is to examine the **visual connectivity of each POI within the street network** – effectively, how well can one see a given POI from various decision points in the vicinity.
   - In an IVG, nodes include both intersections and POIs, and edges represent lines-of-sight between an intersection and a POI (if mutual visibility exists in open space)

2. **Street Network Visibility Graph (SNVG):** This graph connects **street intersections to other intersections** if they are intervisible (you could stand at one intersection and see the other down the street, around buildings).



- It's essentially a visibility graph of the street network itself, ignoring POIs.
- In SNVG, only street junctions/turning points are nodes; an edge between two junctions means you can look from one and see the other without buildings obstructing the view.

3. **Point-of-Interest Visibility Graph (POIVG):** This mode connects **POI to POI** – creating edges between POIs that have direct lines-of-sight between them.



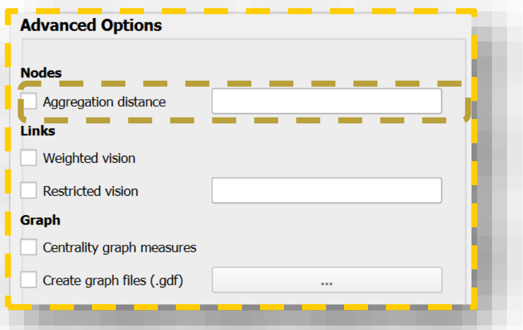
- Only POIs serve as nodes in the graph.
- POIVG is useful for analysing, for example: a
    - network of landmarks: which landmarks are mutually visible?
    - Do important buildings have line-of-sight to each other (forming clusters), or are they visually siloed by the urban fabric?

Advanced Parameters

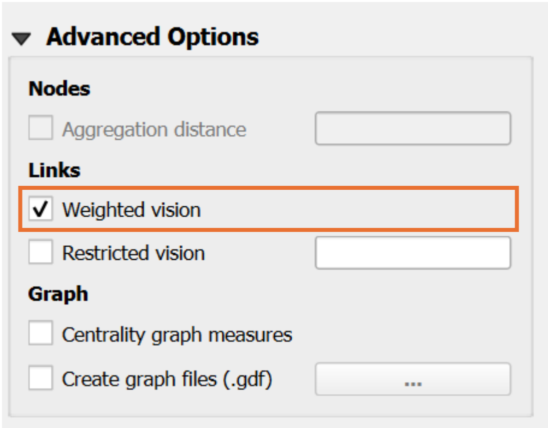
Before running, you have the option to set three important parameters: **Aggregation Distance**, **Viewing Distance Limit** and **Perceptual Perspective**. These correspond to how far and how perspective-weighted the visibility analysis should be:

- Aggregation Distance: POI VizNet will attempt to identify intersections automatically, including merging duplicates (via a quadtree search and clustering).

    - It merges intersection points within a default tolerance into a single node to handle cases where road data isn't perfectly snapped. You can adjust this threshold if needed. By default, a 20 m radius is used, meaning any intersections within 20 m are conflated to one point.

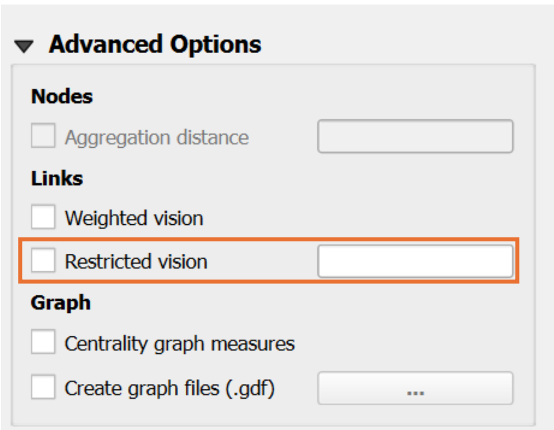


- *Perceptual Perspective (Weighted vision):* Human perception gives more weight to nearer visual targets than far ones. POI VizNet offers a “perspective effect” option to **weight edges by distance**. If enabled, this does *not* change which edges exist, but assigns a weight (or score) to each sightline inversely proportional to distance.



    - Closer connections get higher weight (indicating stronger visual impact), and farther ones lower weight.

- *Viewing Distance Limit:* By default, POI VizNet assumes unlimit ed line-of-sight distance – if two points can see each other at any distance, it will connect them. However, in reality, human vision (and interest) has a range.
    - You can impose a maximum distance (e.g., 400 meters) beyond which the tool will not create a visibility edge. For example, a **“500 m” viewing distance cap** means a POI will only be connected to intersections within 500 m that have direct line-of-sight. Any intersections farther than 500 m, even if theoretically visible, would be ignored.
    - This can significantly reduce clutter in large graphs and focus on the *local visual environment*.



Another option is creating graph file:

- The **Graph data file** saved to disk. POI VizNet exports the graph in a standard network format. it uses **GDF** (Graph Data Format). GDF is a text format (CSV-like) that lists nodes and edges in separate sections. The plugin likely saved a .gdf file either in a default location or one you specified in the dialog.

Run the analysis and generate outputs

Now it's time to run the analysis. Click the **"OK"** button in the plugin dialog. POI VizNet will construct a graph data structure: nodes (points) and edges (connections).

Once completed, POI VizNet automatically adds the results to your QGIS project for visualization. After running, check your QGIS Layers panel. POI VizNet will have added:

- A **"Sight Lines" layer** which contains all the visible connections drawn. Each line connects two points (an intersection-to-POI, intersection-to-intersection, or POI-to-POI depending on graph type). The attribute table for this layer usually includes an ID for each line, its length, and possibly the weight (if weighting was on) This layer is useful for quick visual inspection on the map.
- A **"Nodes" layer** (point layer) which contains the nodes used in the graph. This includes street intersection nodes and/or POI nodes depending on the mode. The nodes might have attributes like an ID, a type/category (to distinguish intersection vs POI), and coordinates.

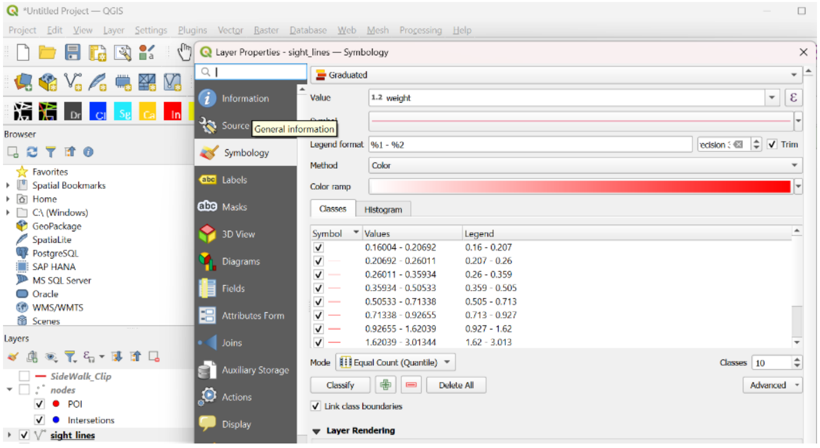
At this point, **the visibility network is constructed and ready**.

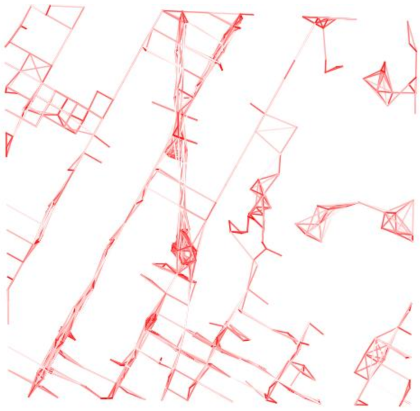
Interpreting Results: Visualization & Analysis

Visually Interpretation:

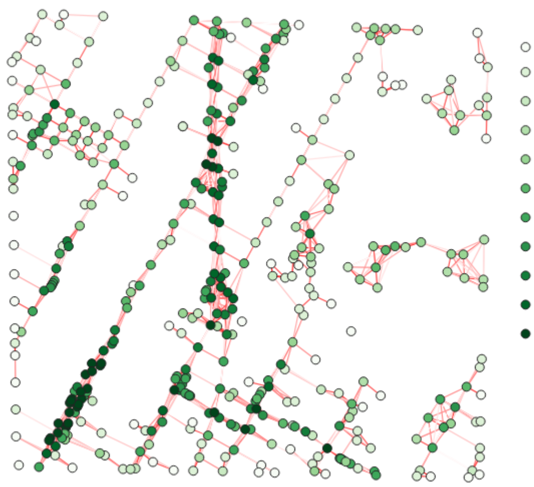
- Do you see a concentration of sight lines around certain streets or POIs?
- Are some POIs completely without any connecting lines (they are hidden)?

Tip: Exporting and using a network visualization tool can help clarify the structure (Gephi or NetworkX)





Degree Centrality: Count of edges per node. For a POI, the degree equals how many intersections can see it (visual exposure). For an intersection, degree is how many other intersections and/or POIs it sees (a measure of how visually connected that vantage point is).

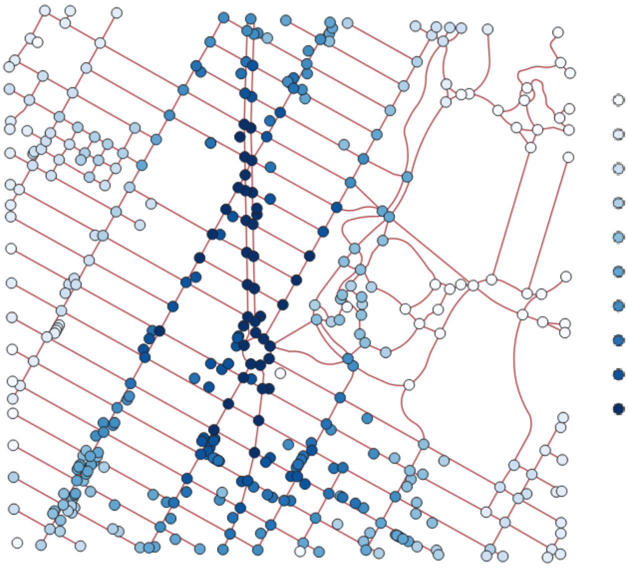


- the number of locations that can be seen from each location within given morphological conditions

Closeness Centrality: In the visibility network, closeness could indicate how visually “central” a node is in terms of average steps (hops) it can see others.

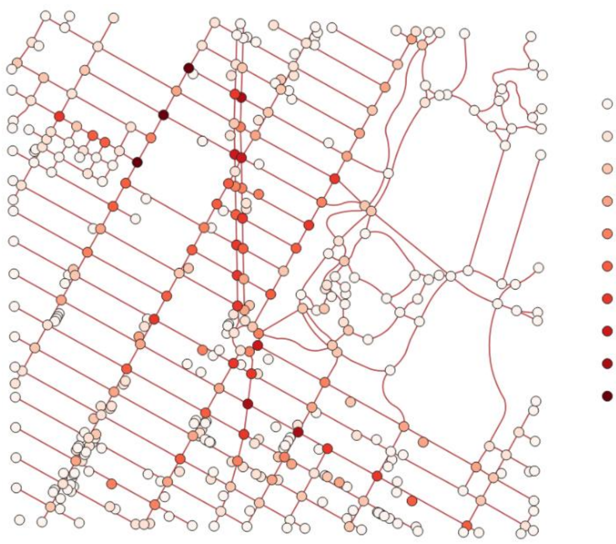
It’s calculated as the **inverse of the average shortest path length** from that node to every other node:

$$\text{Closeness}(v) = \frac{1}{\sum_{u \neq v} d(v,u)}$$



Betweenness Centrality: This might show intersections that lie on many shortest visual paths between others – perhaps crucial gateways for sight. For example, a narrow street that is the only line-of-sight between two areas will have high betweenness in the graph (like a visual ‘bridge’).

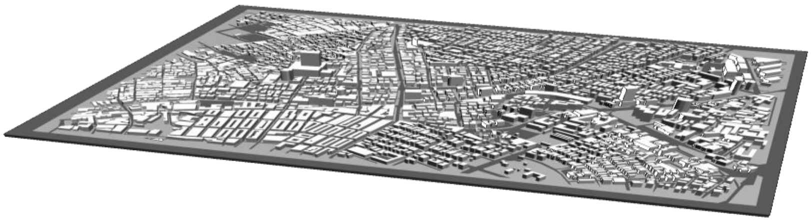
- It may be **crucial for navigation**, signage, or placement of visual anchors (e.g., signs, artworks, lighting).



Case Study:

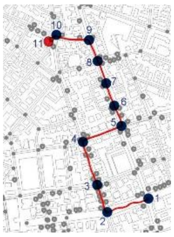
visibility of urban activities and pedestrian routes:

An Experiment in a Virtual Environment



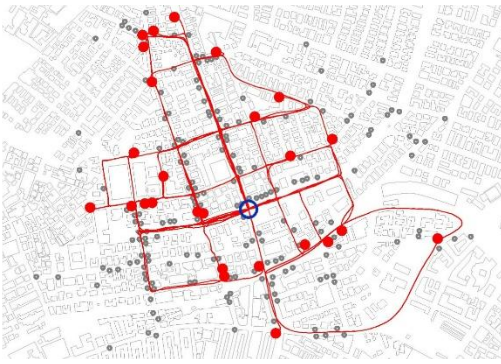
Objective: to examine whether and how urban attractors, distributed within the street network, affect humans' route choices.

- Visibility scores:
    - Visible Intersections (SN degree centrality)
    - Visible Cafés (Café Index)
    - Visible Cafés and Intersections (IVG degree centrality)
- The route's total visibility is defined as an average of the visibility scores.



Statistical comparison of the VR routes with simulated random trajectories (z-test)

	SN visibility	Café visibility	Integrative visibility
Simulated sample	10,000 agents		
Simulated Mean	**11.80**	**20.85**	**32.66**
Standard deviation	**1.64**	**6.79**	**7.50**
VR sample size	40 participants		
Human Mean	**11.20**	**22.54**	**33.74**
Z-score	**2.33**	**-1.57**	**-0.91**
P-value	**0.0099**	**0.058**	**0.181**



- **Human route decisions, in contrast to random choices, are affected by the visibility of the environment.**
- **In the predefined wayfinding task, when looking for the specific activity, visibility of street network has less influence on the route choice and even neglected in favor of functional visibility.**