Supporting sanitation infrastructure planning in low-income areas with URBANUS

Maria do Carmo O. Correia[1,2], Erick P. A. Barcelos[1], Augusto C. B. Pereira[1], Gerson Seluque Ferreira[2], Mirela T. Cazzolato[1]

[1]Institute of Mathematics and Computer Science – University of São Paulo (USP)
Av. Trabalhador São-carlense, 400 - Centro, 13566-590 – São Carlos, SP – Brazil

[2]SELCO Arquitetura e Engenharia Ltda – Ribeirão Preto, SP – Brazil

carmem@selco.com.br, brcls@usp.br

augustocbp@usp.br, gerson@selco.com.br, mirela@icmc.usp.br

Abstract. *Access to sanitation infrastructure remains a critical challenge in low-income urban areas, with direct consequences for public health and social equity. We propose URBANUS, a web-based decision-support platform that automates the preliminary design of sanitary sewer collection networks using publicly available geospatial data. The system models the urban street network as a graph, enriches vertices with topographic elevation, and applies a gravitational routing heuristic (RSPH) to generate a directed sewer network aligned with terrain constraints. An interactive editor allows specialists to inspect and adjust the graph before processing, keeping human expertise inside the computational workflow. A deep learning module for building detection from satellite imagery, based on fine-tuned models, enables accurate sewage contribution estimates per lot. A use case with a Brazilian municipality reduced the domain expert work from days to seconds. URBANUS aims to support engineers, municipalities, and policymakers in expanding coverage to underserved communities.*

1. Introduction

Access to adequate sanitation is a fundamental right. Yet, it remains out of reach for large portions of the population in low-income urban areas worldwide. In Brazil, the 2022 Census showed that 24.3% of the population still relies on precarious sanitation methods, with sanitation deficits concentrated in peripheral and informal settlements [Instituto Brasileiro de Geografia e Estatística 2024]. The consequences are well documented: inadequate sanitation is associated with diarrheal disease, soil-transmitted infections, and broader public health burdens that disproportionately affect the most vulnerable communities [Prüss-Ustün et al. 2019]. Expanding sanitation coverage in these contexts depends not only on political will and investment, but also on the technical capacity to plan collection networks efficiently and at a low cost.

Planning a sewage collection network is a technically complex task. It requires analyzing the road network, modeling flow based on terrain elevation, estimating sewage contribution per lot, and defining a viable layout for the collector network. Approaches based on graph theory have been explored in the literature for layout generation and optimization of sewer systems [Turan et al. 2019], demonstrating that a computational formulation of the problem can substantially reduce manual effort. In practice, however, this kind of computational support is typically delivered through specialized commercial software, which requires licensing costs and technical infrastructure that are often unavailable to small municipalities, social organizations, and engineering professionals operating in underserved communities.

This paper presents *URBANUS*, an open-source web system designed to support preliminary planning of sewage collection networks in low-income areas. Following the Front-End Loading (FEL) methodology, *URBANUS* targets the first planning stage, helping specialists screen public geospatial data, explore candidate layouts, and document preliminary alternatives before detailed engineering design, local surveys, and hydraulic validation [Motta et al. 2011]. The system combines graph-based modeling, public elevation data, interactive graph editing, and automated routing to bring computational support to early sanitation-planning decisions.

The main contributions are: (i) a graph-based pipeline for generating candidate sewage collector layouts from OpenStreetMap data and digital elevation models; (ii) an interactive web interface for expert inspection and refinement; and (iii) a real case study quantifying the reduction in data-entry and processing effort for preliminary layout generation, while preserving the need for later expert review and hydraulic validation.

The paper is organized as follows: Section 2 presents the background concepts. Section 3 discusses related work. Section 4 introduces the proposed method *URBANUS*. Section 5 presents the experimental validation. Finally, Section 6 gives the conclusions.

2. Background

URBANUS uses graph theory to represent both the street network and the sewage collector network. The urban mesh is modeled as a graph $G = (V, E)$, where V is the set of relevant points in urban space and E is the set of connections between them [Boeing 2017]. Vertices correspond to intersections, dead-ends, curves, intermediate points, collection points, or topographic points of interest. Edges correspond to street segments or collector segments, carrying attributes such as length, geometry, road type, and elevation.

This representation enables classical graph properties to be used in interpreting the urban mesh [Turan et al. 2019]. The degree of a vertex indicates how many connections reach a point, helping to identify crossings and bifurcations. Paths represent possible sequences for flow conduction. Connectivity indicates whether all regions of the mesh remain reachable from a discharge or collection point, which is a fundamental requirement for ensuring full network coverage, avoiding isolated segments, and identifying points where direction changes or maintenance requirements imply physical structures such as manholes.

The input street network is initially treated as an undirected graph, since streets provide geometric connections between adjacent points without defining a direction of sewage flow. After topographic analysis, the resulting collector network is represented as a directed graph, where each arc indicates a preferred flow direction [Haydar et al. 2026]. Gravity is a fundamental physical constraint in sanitary sewer systems: flow must be conducted from higher elevations to lower elevations without pumping wherever possible [Turan et al. 2019]. Topography therefore directly influences path selection, edge orientation, and the interpretation of high points, low points, and grade breaks.

Since automatically extracted street data from sources may contain redundant intermediate nodes, the model also requires graph simplification [Boeing 2017]. The challenge is to reduce topological redundancy without discarding technically relevant elements, such as corners, sharp curves, collection points, and nodes constrained by maximum spacing requirements between manholes.

3. Related Work

Commercial systems for sewer and drainage networks provide resources for modeling, simulation, documentation, and operation. Bentley OpenFlows Sewer, for instance, is a professional suite for modeling and managing sanitary sewer systems, with geospatial data integration and hydraulic simulation tools [Bentley Systems, Incorporated 2024]. Such systems are robust, but they are usually directed toward complete professional environments and more detailed stages of the project lifecycle.

Urbano Hydra provides productivity resources for network projects, with synchronization between plans, profiles, and data tables [Peterschinegg GesmbH 2025]. This integration between spatial representation and tabular data is relevant to *URBANUS* because it reinforces the importance of keeping the network drawing, its attributes, and processing results consistent. However, *URBANUS* prioritizes the generation and preliminary analysis of routes from public data and topography instead of starting from an already consolidated technical drawing.

In the Brazilian context, GeoSan is a reference for the management and technical registry of water and sewer networks [Software Público Brasileiro 2014]. Its relationship with *URBANUS* is complementary: while registry systems focus on the operation and record of existing networks, *URBANUS* acts earlier, in the preliminary conception and feasibility analysis of new expansions. The platform also differs by emphasizing interactive graph editing before processing, keeping the specialist inside the computational workflow.

Academic works that apply genetic algorithms, optimization models, and search techniques to sewer network layout are also relevant [Haghighi and Bakhshipour 2012, Weng and Liaw 2005, Rodrigues et al. 2020]. These studies show that a computational formulation of the problem can reduce manual effort and support the search for alternatives. *URBANUS* is positioned in this context as an integration-oriented solution: besides the routing algorithm, it incorporates geospatial data, elevation, persistence, interactive editing, and result visualization in a web application.

4. Proposed Method: *URBANUS*

The proposed method *URBANUS* organizes the generation of a preliminary sewer network into three moments: data preparation, graph processing, and serialization of the resulting network. The process starts from an urban area selected by the user, from which the system obtains public street data, enriches vertices with elevation, builds an editable graph, and allows manual adjustments before processing.

4.1. Data preparation

During data preparation, urban streets are obtained from OpenStreetMap through queries to the Overpass API and represented as GeoJSON [OpenStreetMap contributors 2026].

Topography is obtained through the OpenTopography API, which returns digital elevation models in GeoTIFF format [OpenTopography 2026]. Because public sources differ in completeness and vertical accuracy, *URBANUS* treats them as inputs for preliminary screening rather than survey-grade data for final design. The system samples elevation values at street vertices, handles missing values or border artifacts, and keeps elevation attributes associated with graph nodes. This step transforms raw geographic data into a structure that can be manipulated by the editor and by the backend. Figure 1 illustrates the first interaction of this flow, in which the user defines a bounding box over the map and the system checks the selected area before fetching geospatial data.

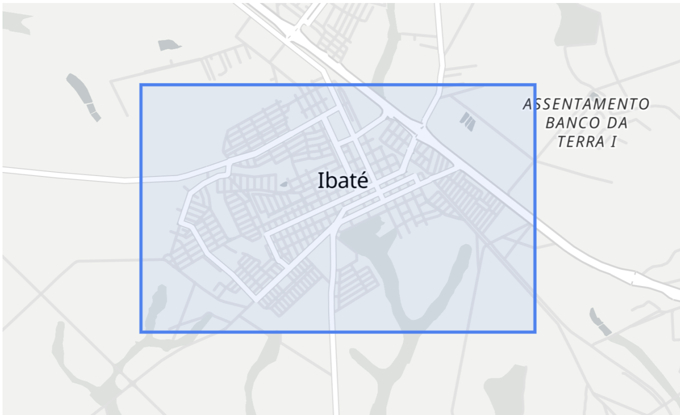

Figure 1. Area-selection step used to define the region processed by *URBANUS*.

4.2. Edited graph as processing input

Before processing, the user can inspect and edit the graph. This decision is methodologically important because the current system does not depend on an automatic reconstruction from the original GeoJSON after the user has made changes. The edited graph becomes the ground truth for the pipeline. In this way, the backend processes exactly the structure that the user visualized and adjusted, reducing the possibility of divergence between the interface and the computed network. Figure 2 shows two complementary editor views of the same project: one emphasizing road categories and another emphasizing elevation values sampled along the graph.

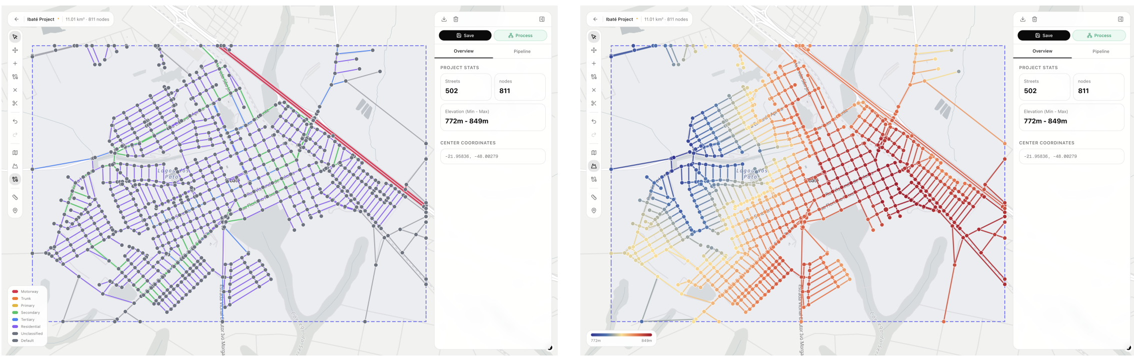

Figure 2. Editable graph views: road-category visualization on the left and elevation visualization on the right.

4.3. Graph processing pipeline

Processing starts by reconstructing the graph using the NetworkX Python library. Then, the system applies safeguards over elevation data, identifies mandatory nodes, marks direction changes, removes redundant nodes, and applies spacing rules between manholes. It then detects local maxima and minima, identifies grade breaks, and selects collection points. These steps prepare the graph for routing by preserving relevant elements of the mesh and reducing points that do not contribute to network interpretation.

Gravitational routing uses the Repeated Shortest Path Heuristic (RSPH) to connect mandatory nodes and collection points to an outlet or to local collectors. The heuristic performs successive least-cost path searches, updates the network under construction, and favors the reuse of trunks. After routing, the pipeline repairs missing edge coverage, removes cycles when necessary, optimizes the position and quantity of nodes, and assigns inspection accessories to the remaining physical nodes. The output is a serialized preliminary network with nodes, edges, lengths, slopes, elevations, flow directions, and relevant point markers. Figure 3 details the routed network after processing, including flow-direction arrows and collection-point markers used to inspect the generated solution.

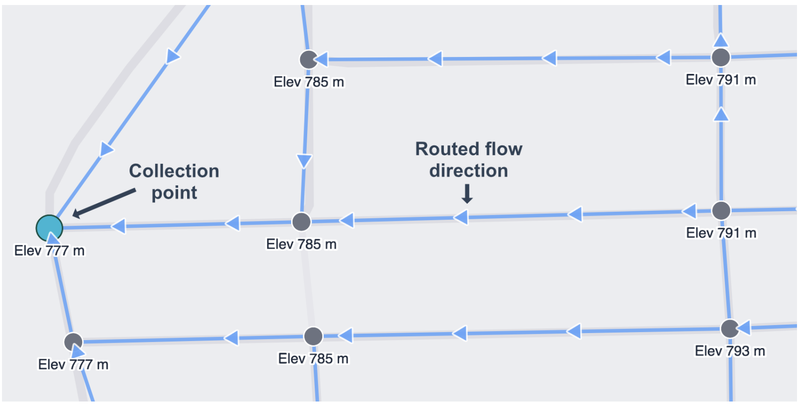

Figure 3. Detailed view of the processed network, highlighting RSPH routing, flow direction, sampled node elevations, and the selected collection point (in blue).

4.4. Architecture and implementation

The system is organized as a polyglot monorepo. The JavaScript/TypeScript part uses a pnpm workspace, while the Python part uses a uv workspace. This organization keeps related applications and packages in the same repository, separating responsibilities without duplicating domain concepts. The source code is available at `https://github.com/rckbrcls/urbanus`. The frontend was developed with Next.js, React, TypeScript, Tailwind CSS, MapLibre, React Map GL, Zustand, and React Query. The interface includes map area selection, street import, interactive graph editing, visualization controls, editing state, result panels, and alternation between the original mesh and the processed network. The editor supports node selection, movement, creation, connection, splitting, and removal, as well as undo and redo actions. The backend was developed in Python with FastAPI, Pydantic, SQLAlchemy, GeoAlchemy2, Shapely, Rasterio, NumPy, httpx, and NetworkX. FastAPI exposes routes for project creation, listing, reading, and deletion, as well as routes for node extraction, elevation enrichment, and network processing. NetworkX is used as the graph structure for algorithmic steps, while PostGIS provides geospatial persistence.

Projects are persisted in PostgreSQL with the PostGIS extension. The database stores metadata, selected area, map center, statistics, street GeoJSON, and the processed network. Node and edge tables store spatial geometries and attributes such as elevation, node type, length, slope, and auxiliary properties. In this way, the processed result can be saved, reloaded, and inspected later. Shared TypeScript packages were also created for geographic types, constants, and utilities, together with a Python package containing Pydantic types and geographic calculations. This division follows the principle of sharing contracts, not implementations: each ecosystem keeps its own implementation, while the central concepts remain aligned through types, constants, and parity tests.

4.5. Automatic building detection

Sewage contribution estimates for each segment of the collector network depend, among other factors, on the type and density of the buildings served. In traditional planning, this information is gathered manually or approximated by aggregate indicators, such as lot frontage length [Rodrigues et al. 2020]. To improve the accuracy of this estimate, URBANUS incorporates an automatic building detection module based on satellite imagery.

The module adopts deep learning approaches for semantic segmentation, automatically identifying the area occupied by buildings directly from medium- and high-resolution satellite images. Techniques based on Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) have demonstrated strong performance on this task, extracting feature hierarchies that capture both local edge and texture details and long-range contextual relationships between image regions [Alidoost and Arefi 2018, Dosovitskiy et al. 2021]. The literature also points to hybrid architectures, such as CTCFNet and MMFFNet, that combine CNNs and Transformers to handle high interclass similarity and variations in urban density [Zhang et al. 2025, Zhou et al. 2024], representing promising directions for future work on this module.

In this work, two pre-trained models were experimentally evaluated with fine-tuning on the Massachusetts Buildings Dataset [Mnih 2013]: SAM (*Segment Anything Model*), a general-purpose segmentation model based on Transformers, and DeepLab V3, a CNN-based model with dilated convolutions.

From an integration standpoint, the module is designed to expose an API that receives the georeferenced area selected by the user and returns segmentation masks and the estimated built area per lot. These data feed the sewage contribution calculation in the URBANUS backend, allowing future versions of the collector network routing to be sensitive to the actual building density rather than relying on frontage-length estimates.

5. Experimental validation

So far, the functional basis of *URBANUS* has been implemented in its web version. The most relevant partial result is the consolidation of a complete flow, still under validation, that goes from the creation of a geographic project to the generation of a preliminary sewer network processed by graph algorithms. The current version already allows users to create and list projects, open the editor, visualize streets and nodes, edit the graph, run the pipeline, save the processed network, and reload the persisted state. Figure 4 shows a local example of this transformation, comparing the enriched editable graph before processing with the directed network produced after the pipeline.

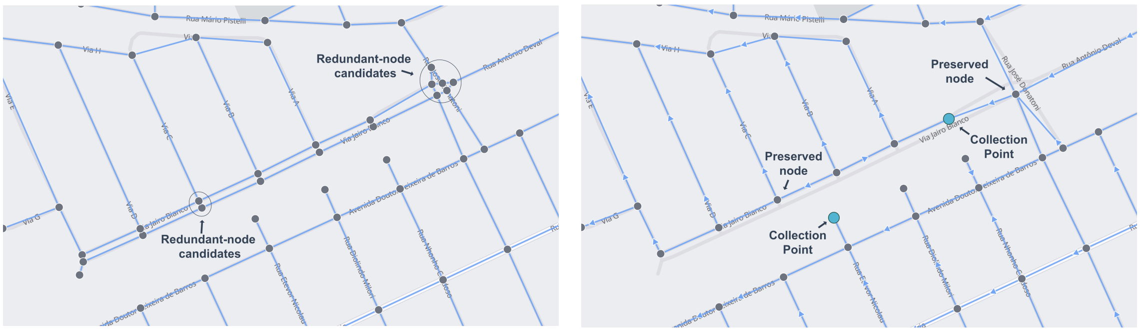

Figure 4. Local graph example before processing on the left and after sewer-network processing on the right.

The current validation is functional. It verifies whether internal steps produce coherent structures: street extraction, elevation enrichment, graph construction, editing, persistence, routing, coverage preservation, and visualization of the processed network. This includes automated tests for geographic calculations, area validation, node rules, parity between implementations, classification, sanitization, extrema detection, edge cost, RSPH, coverage repair, cycle prevention, elevation handling, and node reduction.

5.1. Building detection with DL models

Table 1 compares the segmentation results obtained before and after fine-tuning each model: SAM and DeepLab V3. Both models were fine-tuned on the Massachusetts Buildings Dataset using 151 images of 1500×1500 pixels, tiled into 512×512 patches with a stride of 256 pixels. The dataset was split into 137 training, 10 validation, and 4 test images. Fine-tuning produced substantial gains in both models. Figure 5 shows examples of segmentation outputs. SAM improved its mean IoU from 0.160 to 0.586, while DeepLab V3 increased from 0.177 to 0.600, reaching a mean Dice of 0.747. These values indicate segmentation quality sufficient to feed sewage contribution estimates into the URBANUS pipeline. We notice that SAM, originally trained as a general-purpose segmentation model, achieved performance comparable to DeepLab V3 after fine-tuning, suggesting strong adaptability to the satellite imagery domain.

Table 1. Building segmentation results before and after fine-tuning

Model	Fine-tuning	Mean IoU	Mean Dice
SAM	No	0.160	–
SAM	Yes	0.586	0.734
DeepLab V3	No	0.177	0.293
DeepLab V3	Yes	0.600	0.747

5.2. Case study: Verdelândia, MG

To evaluate the practical utility of *URBANUS*, a case study was conducted on the Barreiro do Rio Verde neighborhood in the municipality of Verdelândia, Minas Gerais, Brazil. The city was selected due to its low Human Development Index (HDI of 0.58) and its

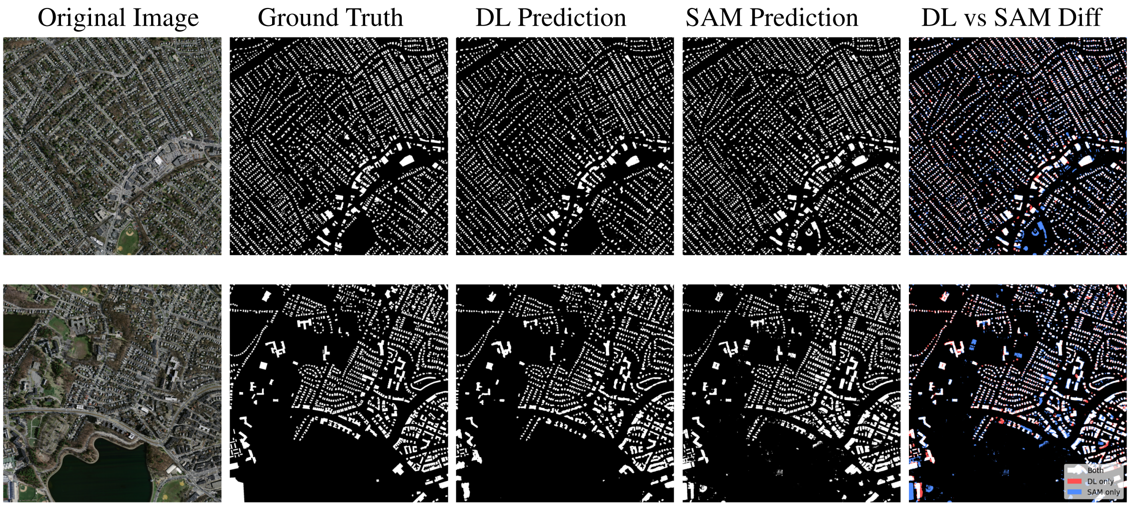

Figure 5. Qualitative segmentation results of the fine-tuned SAM versus DeepLab V3 models in the Massachusetts Buildings Dataset.

irregular terrain, both representative of the context for which *URBANUS* is designed. The case study covered 21 street alignments, drawn from a total of 242 in the municipality.

A domain expert performed the network planning manually using the same input data as *URBANUS*, ensuring a fair comparison. The manual process comprised four stages: (i) data preprocessing, including file cleanup and longitudinal profile generation for each alignment; (ii) manual placement of manholes (PVs) at intersections and topographic inflection points, with identification of high and low points to determine flow direction; (iii) hydraulic validation, consisting of filling spreadsheets with sewage contribution, flow rates, hydraulic radius, flow velocity, and tractive stress for each pipe segment; and (iv) correction and re-validation of segments that did not meet design constraints.

The total time estimated for the expert to complete the 21 alignments was approximately two weeks of working days, broken down as follows: file import and cleanup (approx. 2.5 h), high/low point identification (approx. 3.5 h), manhole and pipe placement (approx. 1.5 h), parameter transcription to spreadsheet (approx. 6 h), and verification and corrections (approx. 6 h). Extrapolating to the full 242 alignments of the municipality, the expert estimated approximately six months of work under normal working conditions.

For the same 242 alignments, the total data-entry and processing time in *URBANUS* was less than a minute. This represents a drastic reduction relative to the expert baseline, without the risk of transcription errors inherent to manual spreadsheet-based workflows. The building detection module was not validated in this use case: the entry of each building lot was provided manually by the specialist.

Beyond processing time, the case study evaluated the geometric accuracy of the node placement produced by *URBANUS*. Two algorithmic steps were assessed against the expert solution used as ground truth. The first step, which enforces a minimum spacing of 100 m between manholes as required by design standards, achieved 100% precision across all 21 tested alignments. The second step, which clusters nearby nodes and merges them into a single representative corner node, achieved a precision of 89.5% on the same alignments. These results indicate that *URBANUS* produces geometrically sound preliminary layouts suitable for further specialist refinement.

6. Conclusions

This paper presented *URBANUS*, an open-source web platform for preliminary planning of sanitary sewer collection networks in low-income urban areas. *URBANUS* combines graph-based modeling of street and collector network, automatic topographic enrichment from public elevation data, interactive graph editing, and a DL module for building detection from satellite imagery.

The system is under active development. On the algorithmic side, the manhole clustering and alignment refinement steps will be extended to handle denser and more irregular street configurations. The building detection module will be more tightly integrated into the main pipeline, enabling sewage contribution estimates to be computed automatically, avoiding manual lot-by-lot input. Future work includes per-lot instance counting and classification of individual buildings, which will enable sewage contribution estimates at finer granularity. Hybrid CNN-Transformer architectures and multimodal fusion strategies with digital surface models (DSM) will also be investigated to improve accuracy in high-density urban scenarios [Zhou et al. 2024].

These results should be read as preliminary evidence for the early planning stage. *URBANUS* supports candidate generation and specialist inspection, but it does not replace executive design, survey-grade topographic data, or independent hydraulic validation. On the validation side, the system will be evaluated on other municipalities with distinct topographic and urban morphology profiles. We aim to have a more comprehensive assessment of generalization across different urban contexts, especially where public map and elevation data may be incomplete or less precise. By lowering the technical and financial barriers to sanitation planning, *URBANUS* aims to support engineers, municipalities, and policymakers in expanding sewer coverage to the communities that need it most.

Acknowledgments

This work was supported by the São Paulo Research Foundation (FAPESP – grants #2026/11130-2, #2024/15430-5, #2024/13328-9), the Coordination for Higher Education Personnel Improvement (CAPES – Finance Code 001), the National Research Council (CNPq and PIBIC 2025-3154), and the University of São Paulo (PRPI 1032 - grant #207).

References

Alidoost, F. and Arefi, H. (2018). A CNN-based approach for automatic building detection and recognition of roof types using a single aerial image. *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 86(5–6):235–248.

Bentley Systems, Incorporated (2024). OpenFlows Sewer: Wastewater collection system modeling and management. Accessed: 2025-12-08.

Boeing, G. (2017). OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65:126–139.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N.

(2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Haghighi, A. and Bakhshipour, A. E. (2012). Optimization of sewer networks using an adaptive genetic algorithm. *Water Resources Management*, 26(12):3441–3456.

Haydar, B., Chahinian, N., and Pasquier, C. (2026). Reconstructing sewer network topology using graph theory. *Water*, 18(2):222.

Instituto Brasileiro de Geografia e Estatística (2024). Censo demográfico 2022: Características dos domicílios – resultados do universo. Technical report, IBGE, Rio de Janeiro. Accessed: 2026-05-25.

Mnih, V. (2013). *Machine Learning for Aerial Image Labeling*. PhD thesis, University of Toronto. Introduces the Massachusetts Buildings Dataset. Accessed: 2026-03-29.

Motta, O. M., Quelhas, O. L. G., and Farias Filho, J. R. (2011). Alinhando os objetivos tecnicos do projeto as estrategias de negocio: contribuicao da metodologia FEL no preplanejamento de grandes empreendimentos. *Revista Gestao Industrial*, 7(4):99–117.

OpenStreetMap contributors (2026). OpenStreetMap. Accessed: 2026-05-19.

OpenTopography (2026). OpenTopography: High-resolution topography data and tools. Accessed: 2026-05-19.

Peterschinegg GesmbH (2025). Urbano Hydra: Infrastructure pipe network design. Accessed: 2025-12-08.

Prüss-Ustün, A., Wolf, J., Bartram, J., Clasen, T., Cumming, O., Freeman, M. C., Gordon, B., Hunter, P. R., Medlicott, K., and Johnston, R. (2019). Burden of disease from inadequate water, sanitation and hygiene for selected adverse health outcomes: An updated analysis with a focus on low- and middle-income countries. *International Journal of Hygiene and Environmental Health*, 222(5):765–777.

Rodrigues, G. P. W., Farias, G. M., Costa, L. H. M., and Castro, M. A. H. d. (2020). Otimização do traçado de redes coletoras de esgoto sanitário via algoritmo genético. *Revista DAE*, 68(222):164–177.

Software Público Brasileiro (2014). GSAN/GeoSan: Sistema integrado de gestão de serviços de saneamento e GeoSan – módulo GIS. Accessed: 2026-05-05.

Turan, M. E., Bacak-Turan, G., Cetin, T., and Aslan, E. (2019). Feasible sanitary sewer network generation using graph theory. *Advances in Civil Engineering*, 2019:8527180.

Weng, H. T. and Liaw, S. L. (2005). Establishing an optimization model for sewer system layout with applied genetic algorithm. *Journal of Environmental Informatics*, 5(1):26–35.

Zhang, S., Li, M., Zhao, W., Wang, X., and Wu, Q. (2025). Building type classification using CNN-Transformer cross-encoder adaptive learning from very high resolution satellite images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 18:976–994.

Zhou, W., Persello, C., and Stein, A. (2024). Hierarchical building use classification from multiple modalities with a multi-label multimodal transformer network. *International Journal of Applied Earth Observation and Geoinformation*, 132:104038.