Vertical Mix Index: A Novel Metric for Vertical Land Use Mix in Procedural City Generation

Bertan Konuralp
Computer Science Department
TU Delft
Delft, Zuid-Holland, Netherlands
b.konuralp@student.tudelft.nl

Rafael Bidarra
Computer Science Department
Delft University of Technology
Delft, Zuid-Holland, Netherlands
R.Bidarra@tudelft.nl

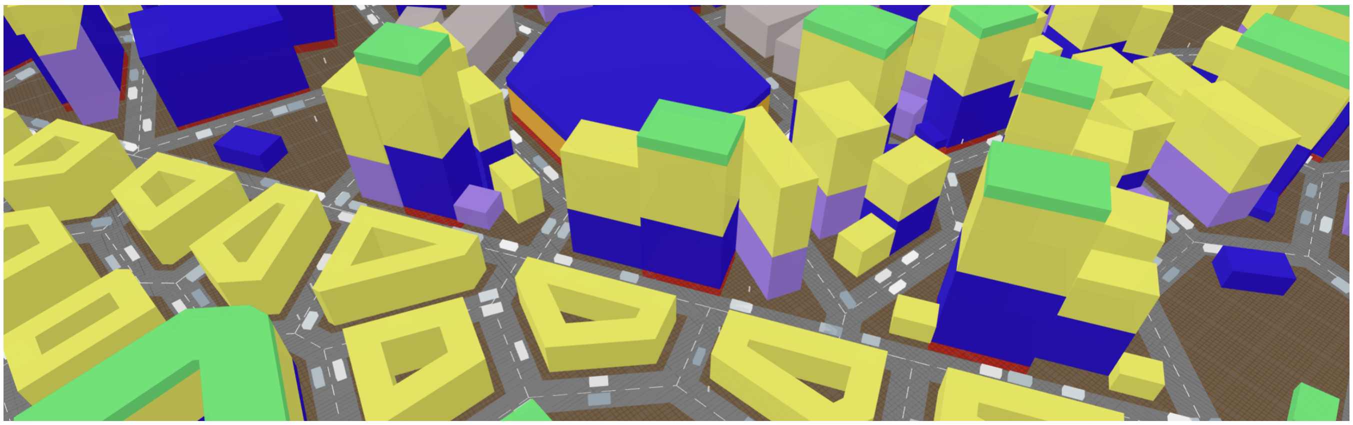


Figure 1: City center and residential neighborhood blocks with different Vertical Mix Indices

Abstract

Land Use Mix (LUM) is a central concept in urban development theory, advocating the integration of distinct primary functions such as residential, commercial, and public spaces within urban environments, contributing to increased street vitality, reduced car dependency, and support for alternative transport. However, LUM remains difficult to define and quantify. Existing frameworks decompose it into scale, dimension, and texture, where the vertical dimension refers to stacking functions within buildings (e.g., retail below residential). While horizontal LUM has been quantified, vertical LUM remains largely unmeasured. Procedural urban generation pipelines focus on urban form rather than functional composition; although density measures such as Floor Space Index (FSI) and Ground Space Index (GSI) are used, LUM is rarely incorporated as a controllable parameter. This paper introduces the Vertical Mix Index (VMI), a novel metric to quantify vertical LUM and integrate it into procedural urban generation. Results show that increasing VMI produces greater vertical functional mixing, though its effectiveness depends on alignment between density distribution, functional allocation, and generative constraints.

FDG '26, Copenhagen, Denmark

ACM ISBN 979-8-4007-2495-4/26/08
https://doi.org/10.1145/3815598.3815704

CCS Concepts

• **Information systems** → *Multimedia content creation.*

Keywords

procedural content generation (PCG), city generation, computational urban design, urban morphology, urban planning, urban density, land use mix, vertical land use mix, vertical mix index (VMI)

ACM Reference Format:
Bertan Konuralp and Rafael Bidarra. 2026. Vertical Mix Index: A Novel Metric for Vertical Land Use Mix in Procedural City Generation. In *Foundations of Digital Games (FDG '26), August 10–13, 2026, Copenhagen, Denmark.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3815598.3815704

1 Introduction

Procedural city generation has become an important tool for generating large-scale urban environments in games, simulation, and digital urban modeling. Modern systems are highly capable of generating extensive road networks, dense skylines, and visually convincing architectural variation [29]. User control within these systems is commonly expressed through macro-level density adjustments implemented either via population density maps, reference images or through parametric urban morphology indices such as the **Floor Space Index (FSI)** and **Ground Space Index (GSI)** [21, 24, 33]. Furthermore, land use allocation generators may be used to create distinct horizontal zones across the terrain, such as a city center or a residential district [8, 15, 18]. While density and zoning can influence the size, shape, height, and visual appearance of structures to imply their perceived function, they do not explicitly control the internal functional composition and vertical organization of buildings. As a result, explicit control over vertical land use mix remains largely absent from current procedural city generation workflows.

In real cities, urban functions rarely exist in isolation. Commercial ground floors are commonly combined with residential or office spaces above, parking is frequently integrated underground below buildings, and rooftops increasingly host recreational or public functions. These forms of vertical **land use mix (LUM)** emerge from land scarcity, density pressure, economic incentives, and accessibility constraints, especially in dense urban environments. One of the most influential advocates for mixed-use urbanism was Jane Jacobs in *The Death and Life of Great American Cities* [10], where she argued that active and walkable cities depend on the integration of multiple urban functions rather than strict single-use zoning. As cities become denser and increasingly vertical, the organization of functions across height becomes an important aspect of how urban environments are experienced.

This limitation in procedural city generation can become relevant in interactive and explorable virtual worlds. The vertical organization of functions may influence how urban environments are traversed, perceived, and experienced. Dense mixed-use environments can create stronger variation in activity, more layered navigation patterns, and more believable transitions between distinct functional spaces. Controllable vertical mixing can therefore provide procedural generation systems with a new dimension of urban variation beyond purely geometric form and strict zoning. Given this gap in procedural city generation methods, as well as the lack of a formal measure for vertical LUM, we present our main research question:

How can vertical LUM be quantified and integrated as a controllable design parameter within computational urban generation systems?

To investigate, we split it into the following three sub-questions:

RQ1: Which characteristics of real-world vertical LUM are relevant for procedural urban generation?

RQ2: How can vertical LUM be defined as a quantitative index?

RQ3: To what extent can this index be integrated into a procedural city generation system as a controllable design parameter?

The main contributions of this paper can be summarized as:

(1) The **Vertical Mix Index (VMI)**, a quantitative measure for vertical land use mix suitable for procedural city generation.
(2) A hierarchical framework for distributing density and vertical mixing targets across districts, neighborhoods, and blocks.
(3) A block generation pipeline for approximating target VMI, FSI, and GSI values.

2 Background & Related Work

This section reviews related work from urban development and procedural city generation, as this research bridges both domains through controllable vertical LUM. The historical development, definitions, and quantitative measures of LUM are first discussed, followed by existing forms of density and functional composition control within procedural city generation.

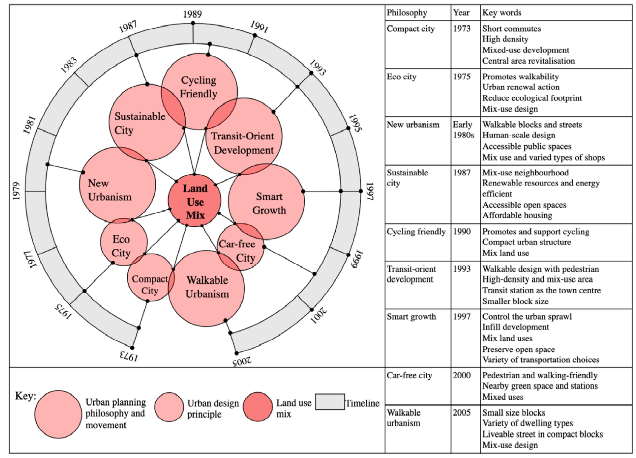


Figure 2: Examples of Land Use Mix related urban development theories since the 1970s [11]

2.1 Land Use Mix & Verticality

Land Use Mix (LUM), also referred to as mixed-use development, describes the spatial integration of multiple urban functions within a defined area. By bringing residential, commercial, office, and recreational functions into closer proximity, LUM has been associated with increased walkability, reduced automobile dependence, enhanced social interaction, and more active urban environments [10, 31].

Historically, urban functions were often organically intertwined through mixed residential and commercial structures, particularly in pre-industrial cities [34]. During the twentieth century, however, modernist planning and Euclidean zoning promoted the spatial separation of urban functions, reinforcing mono-functional urban patterns. Criticism of these approaches, most notably by Jacobs [10], later renewed interest in functional diversity and reintegration. Contemporary planning therefore increasingly approaches LUM not as the absence of zoning, but as the strategic integration of functions within regulated urban frameworks [34]. As a result, LUM has become a recurring principle within planning paradigms such as compact cities, transit-oriented development, and walkable urbanism (Figure 2).

Despite its widespread adoption, LUM remains difficult to define and measure consistently. LUM operates across multiple scales, textures, and spatial dimensions [9, 27], including the horizontal dimension of mixing across districts or neighborhoods, the vertical dimension of stacking functions within shared ground space [20], and the temporal dimension in which spaces accommodate different uses over time.

Vertical LUM most commonly appears through commercial ground floors with residential uses above, though more complex combinations may additionally integrate office, public, recreational, or parking functions [20]. As cities densify and land availability becomes constrained, vertical organization becomes increasingly important, reflected in concepts such as vertical urbanism [17] and volumetric urbanism [4], which conceptualize cities as layered three-dimensional systems rather than predominantly horizontal compositions.

2.2 Quantitative Indices in Urban Analysis

Quantitative measures are essential for comparing urban environments. Among the most widely used are density indices such as the Floor Space Index (FSI) and Ground Space Index (GSI), representing the ratios of total floor area and building footprint area relative to land area respectively.

Let A_{land} denote the total horizontal land area under consideration, A_{floor} the total accumulated floor area of all buildings, and $A_{\text{footprint}}$ the total building footprint area. The indices are then defined as:

$$\text{FSI} = \frac{A_{\text{floor}}}{A_{\text{land}}} \qquad \text{GSI} = \frac{A_{\text{footprint}}}{A_{\text{land}}} \tag{1}$$

Together, these indices describe built intensity and are widely used in comparative urban morphology analysis [25]. Various quantitative measures have been proposed to capture LUM, including diversity-based, interaction-based, and compatibility-based indices [34]. These may operate either as aggregate measures for the entire study area or as local measures across smaller spatial units such as grids or buffers [31]. Common approaches include entropy measures, clustering metrics, dissimilarity indices, and exposure-based methods.

Despite their usefulness, aggregate LUM measures suffer from several limitations because they compress complex spatial organization into a single value. Hess et al. [22] argue that such measures treat all land use combinations equally despite their differing urban implications, cannot distinguish between directly integrated functions and uses separated spatially, and assign identical scores to reversed proportional distributions that may produce very different urban environments. Manaugh et al. [19] further question whether perfectly even land use distributions should necessarily be considered optimal.

Although existing LUM measures capture horizontal land use relationships effectively, they were not designed to explicitly represent vertically overlapping functional organization within urban environments.

2.3 Procedural City Generation

Given the predominantly horizontal focus of existing LUM measures, the question arises how such concepts may be incorporated into procedural city generation. Procedural city generation encompasses a broad range of approaches for generating settlements, districts, or metropolitan environments across varying terrain conditions.

2.3.1 PCG Pipeline Philosophy. Procedural city generation commonly relies on hierarchical pipelines in which large-scale layout decisions are progressively refined into finer architectural detail [29]. Typical workflows generate terrain and land allocation first [8], followed by road networks [32], parcel subdivision, and building generation through rule-based grammars or parameterized transformations. CityEngine [24] remains a prominent example of this approach, while simulation-based systems such as agent-based models may integrate multiple stages to produce more organic settlement growth [16].

2.3.2 Applications of PCG. Within game development, procedural city generation serves multiple purposes. It may be deployed dynamically at runtime to generate unique city layouts during gameplay, or utilized offline as an assistive tool to support and accelerate manual world creation [29]. Procedural generation is also widely used within sandbox environments that support user-generated settlements. One notable example is the Generative Design in Minecraft (GDMC) competition [28], in which participants develop procedural algorithms capable of generating settlements on randomized Minecraft terrain.

In architectural and urban design contexts, PCG systems are often referred to as parametric urban design tools, where designers can regulate urban form through configurable parameters. These systems are commonly used to prototype buildings, blocks, or neighborhoods under varying density and spatial constraints.

These applications highlight the importance of controllable urban parameters, as different experiential, functional, or visual characteristics may be desired depending on the intended application domain.

2.4 Controllability in Procedural City Generation

The balance between automation and user control remains a central challenge in procedural generation. Fully automated systems maximize scalability and diversity through stochastic variations, but offer limited direct influence over the output. Conversely, introducing user control constrains the generative space while improving design precision and intentionality. This creates an inherent tension between procedural autonomy and explicit authorial control. Modern procedural systems therefore increasingly combine automated generation with user-guided constraints and parameterized control mechanisms. [30]

2.4.1 Types of User Control. User control in procedural city generation may regulate infrastructure networks, land allocation, building morphology, and the distribution of functions across urban regions. Density acts as a unifying concept across many controllability dimensions, influencing infrastructure intensity, building morphology, and land use mixing. Consequently, indices such as FSI and GSI are commonly used to analyze existing cities, with density used to guide procedural generation. Additionally, land use allocation systems provide forms of 2D LUM through the horizontal distribution of zones across the terrain, which can be used to infer urban density. In turn, density can be used to visually imply functional composition through building morphology and façade textures. However, density does not explicitly regulate how functions are combined vertically within buildings Kim et al. [12] argue that procedural city generators should evolve beyond purely visual plausibility to incorporate functional, demographic, and infrastructural dimensions, emphasizing the importance of representing urban functionality in procedural environments.

2.4.2 Procedural Control Mechanisms. Muller et al. [24] introduced CityEngine, a procedural city generation system driven by geographical and sociostatistical maps controlling population density, zoning distribution, street patterns, building heights, elevation, and land-water boundaries. Similarly, Mizdal et al. [3] allow users to adjust village size and density within a terrain-aware settlement generator.

Other approaches emphasize direct or assisted interaction during generation. Benes et al. [2] allow users to modify urban nuclei and paint roads interactively, while Nishida et al. [23] generate road networks from user-provided OpenStreetMap examples. Image-guided approaches further infer urban morphology from visual reference data [13, 33].

Several approaches incorporate user control over horizontal LUM at the district or neighborhood scale. Lyu et al. [18] introduced a hierarchical procedural urban layout framework allowing users to control population distributions, road network patterns, and the proportions of residential, industrial, commercial, office, and green space land uses. Groenewegen et al. [8] similarly generate city layouts from high-level parameters such as city size, geographic location, and terrain conditions, distributing functional districts through attraction and repulsion relationships derived from urban land use theories. Lechner et al. [15] further emphasize artist steerability through interactive land use painting and editable development constraints.

Within architecture and computational urban design (CAAD), parametric approaches further emphasize controllable urban generation through adjustable urban morphology indices and rule-based spatial relationships. Mei et al. [21] developed a density-driven Grasshopper framework using planning indices such as floor area ratio (same as FSI) and coverage constraints to guide urban form generation, while Elzeni et al. [7] call for the use of more indices within urban generation and propose a classification framework with a variety of indices for urban morphology indicators. Similarly, Akay et al. [1] introduce a plot-based algorithmic framework that incrementally controls urban block formation through associative relationships between parcels, building configurations, and environmental performance constraints.

Collectively, these approaches demonstrate that contemporary procedural city generation systems can already provide substantial control over infrastructure generation, density, morphology, and horizontal land use organization. However, these controls primarily regulate geometric form and planar functional allocation. Explicit control over how multiple urban functions coexist and overlap vertically within individual building volumes remains largely absent from existing procedural generation workflows.

3 Methodology

This methodology addresses the integration of controllable vertical LUM within procedural city generation through three components. First, VMI is defined as a quantitative measure of vertical functional diversity. Second, density and vertical mixing targets are distributed hierarchically across the urban environment. Finally, block-level generation procedures are used to realize target GSI, VMI, and FSI values while respecting functional compatibility constraints.

3.1 Quantifying Vertical Land Use Mix

Before quantifying vertical LUM, it is important to note that the degree of observable vertical mixing depends on the granularity of functional categorization. For example, a broad "commercial" category may alternatively be subdivided into retail, dining, entertainment, or cafés. Finer classifications naturally reveal higher degrees of functional mixing and therefore remain an important contextual parameter when interpreting vertical LUM.

To quantify this phenomenon, we introduce the **Vertical Mix Index (VMI)**, a measure of vertical functional diversity within an urban area. VMI captures the extent to which distinct functional categories coexist vertically, normalized over the total land area. Unbuilt space is treated as a functional category itself, namely open space, representing pedestrian movement, recreation, and communal public use. This interpretation additionally allows future extensions involving elevated walkways, overhangs, tunnels through buildings, or bridges between buildings.

Let $A_{\text{land}} \subset \mathbb{R}^2$ denote the horizontal land area under consideration. For any horizontal position $p \in A_{\text{land}}$, let $c(p)$ denote the number of distinct functional categories present within the vertical column passing through p. By definition, $c(p) \geq 1$ for all locations.

The Vertical Mix Index is defined as:

$$\text{VMI} = \frac{1}{|A_{\text{land}}|} \iint_{A_{\text{land}}} c(p)\, dA, \tag{2}$$

where $|A_{\text{land}}|$ denotes the total horizontal land area.

Conceptually, VMI represents the expected number of distinct functional categories encountered within a vertical column sampled uniformly across the urban environment. VMI can be used both as a city-wide aggregate measure and as a block-level measure capturing localized forms of vertical LUM. From the perspective of existing LUM classifications, VMI is most closely related to diversity and interweaving-based measures, as it quantifies the co-location of distinct functional categories. Unlike entropy or proportion-based diversity indices, VMI does not measure proportional balance between land uses and neither does it encode the vertical ordering of functions. It should therefore be interpreted as a measure of vertical LUM rather than a complete representation of vertical land-use composition. Consequently, when used to guide procedural generation, VMI must be complemented by additional constraints governing functional composition.

3.2 Hierarchical Generation Framework

To evaluate the integration of VMI within procedural city generation, a simplified hierarchical generation framework is proposed in which density and functional composition can be explicitly controlled. Rather than reproducing detailed architectural realism, the framework focuses on the spatial organization and vertical overlap of urban functions. The generator operates hierarchically across four spatial scales:

$$\text{city} \rightarrow \text{district} \rightarrow \text{neighborhood} \rightarrow \text{block}$$

Density and vertical mixing targets are distributed top-down through the hierarchy, allowing heterogeneous local conditions while collectively satisfying aggregate city-level targets. Each zone is associated with a moderator responsible for managing and propagating its assigned density model to its child zones.

Following propagation, blocks estimate their required ground area from their assigned density parameters. These estimates are aggregated upward through the hierarchy until the city obtains an estimated required size. The terrain is then partitioned using a power-diagram approach to assign district, neighborhood, and block boundaries based on the required ground floors calculated. Once generated, blocks can be realized independently according to their assigned density models.

3.2.1 Density Model. The generation process is guided through a configurable density model consisting of quantitative urban parameters distributed hierarchically throughout the city. These include FSI, GSI, VMI, network complexity, and proportional distributions of functional floor space across categories such as residential, commercial, office, public, transit, and parking space.

The density distribution process is controlled through a global configuration file defining zone-specific tendencies for each parameter. Each index is associated with configurable bias and flexibility parameters. Bias values define directional tendencies relative to the parent zone, while flexibility values determine how strongly child indices may vary around those tendencies. Together, these parameters determine the number and characteristics of generated child zones and guide block-level realization.

3.3 Block-Level Generation Pipeline

Once the density models have propagated throughout the city hierarchy, each block is realized according to its assigned target density parameters. Block generation approximates these targets rather than matching them exactly. The process proceeds in three stages: building footprints are generated to approximate GSI, functionality counts and compatible sequences are assigned to building footprints to approximate VMI, and number of floors are assigned for each function to approximate FSI and functional floor-space demands.

3.3.1 Functional Volume Representation. During generation, functions are first organized as vertical sequences per building before being transformed into volumetric representations for storage and visualization. Generated spatial data is stored in a PostGIS database, where zones are represented as 2D polygons and functional spaces as 3D polyhedral volumes. Each functional volume corresponds to a specific urban function, whose height is determined by the assigned number of floors and a fixed floor height. To simplify vertical mixing, each floor hosts only a single function, although neighboring building footprints may touch or partially connect to form larger continuous building masses, creating floors with mixed functionalities. Roads are assumed to exist along block boundaries without being modeled explicitly as geometry. This abstraction, together with the omission of interior layouts of the functional volumes, allows the framework to focus on functional composition and vertical organization of the urban environment.

3.3.2 GSI Realization. Building footprints are generated using different realization strategies depending on the target density. For lower GSI values, building footprints are generated through an iterative growth process. The process begins by placing minimum-sized building footprint seeds using a Poisson-like stochastic distribution aligned to nearby block borders. These initial footprints then expand incrementally in directions with the greatest available space while ensuring block containment and preventing overlap with neighboring buildings. Growth continues until the target GSI is reached or no further feasible expansion remains possible.

For higher GSI values, an inset-based footprint generation method is used instead. In this mode, block boundaries are uniformly inset until the resulting geometry approximates the target GSI. Depending on the selected configuration, the inset geometry is either used directly as a compact central building mass or subtracted from the block to produce perimeter-block morphologies with interior courtyards.

3.3.3 VMI Realization and Functional Assignment. VMI realization happens through two stages. First, the number of functions that buildings need to host in order to approximate the target VMI is determined. This is done by iteratively incrementing building functionality counts using heuristic building selection while repeatedly evaluating the realized block-level VMI until the target is sufficiently approximated.

The second stage assigns the actual functions and vertical ordering within each building. Rather than assigning functions independently, the generator distributes compatible bottom-to-top function sequences across the block using a scarcity-aware allocation process. Candidate sequences are filtered according to compatibility and geometric feasibility constraints to ensure plausible vertical organization. The assignment process prioritizes functions with limited future placement opportunities to reduce infeasible configurations and produce more balanced functional distributions across the block. Detailed implementation details of this process are described in Section 4.

3.3.4 FSI Realization. Because the framework explicitly models multiple functional uses, aggregate FSI is internally decomposed into function-specific floor-space shares, making FSI realization equivalent to approximating the required floor space for each functional use individually.

Given building footprints and assigned numbers of functions per building, floor space is allocated through a greedy iterative process that reduces the remaining floor-space deficit for each function. At each step, the function with the largest unmet demand is selected, and additional floors are assigned to a feasible building already hosting that function while respecting height limits and constraints imposed by the assigned sequence. Building selection favors configurations that provide the most efficient increase in floor space, typically those with larger footprints, while avoiding over-concentration of a single function or excessive dominance of individual buildings to maintain balanced heights. This process continues until the desired functional distribution is approximated or no further valid assignments remain possible.

4 Block Generator VMI Implementation

This section describes the implementation details of the VMI-related components of block generation. Specifically, it addresses two problems: how target VMI values are approximated within generated blocks, and how compatible functions are selected and vertically ordered within building stacks.

4.1 Approximating VMI

The implementation calculates VMI separately over built and unbuilt areas before combining them into a single block-level value. Each building footprint is treated as a uniform spatial region with a distinct functional count, since each floor within a building is restricted to a single functional assignment.

Let $B = \{b_1, b_2, \ldots, b_n\}$ denote the set of generated buildings within a block. For each individual building b_i, we define:

- A_i as the horizontal footprint area of the building,
- c_i as the number of distinct functional categories assigned to the building.

The remaining unbuilt or open space within the block domain, A_{unbuilt}, is defined as:

$$A_{\text{unbuilt}} = A_{\text{land}} - \sum_{i=1}^{n} A_i. \tag{3}$$

Unbuilt open space is assumed to possess exactly one functional assignment and therefore contributes a baseline functional count of 1. The continuous global VMI target is subsequently calculated as an area-weighted sum across the entire block:

$$\text{VMI} = \frac{\sum_{i=1}^{n} A_i c_i + A_{\text{unbuilt}} \cdot 1}{A_{\text{land}}}. \tag{4}$$

During the generation process, functionality counts per building are iteratively incremented until the realized VMI approaches the VMI target. This process is guided by admissible functional combinations and building-specific feasibility rules. For example, the maximum number of functions that can be assigned to a building is bounded by the largest valid functional sequence available using the requested function set of the block, while geometric feasibility constraints may restrict the amount of vertical stacking achievable for a given building footprint.

4.2 Assigning Compatible Function Sequences

Because VMI alone does not dictate functional compatibility or vertical ordering, the generator uses predefined bottom-to-top function sequences filtered according to spatial and positional constraints, such as stack length limits, minimum footprint dimensions, and functions restricted to lower or upper floors.

Algorithm 1 summarizes the global sequence assignment process. Rather than assigning functions independently per building, the generator performs a scarcity-aware allocation that iteratively distributes function sequences while tracking remaining functional capacity targets. Each assigned sequence contributes the building footprint capacity toward the demand targets of every function contained within that sequence. This allows the system to prioritize functions with limited future placement opportunities and avoid assignments that would later prevent feasible configurations elsewhere in the block.

Algorithm 1 Scarcity-Aware Function Sequence Assignment

Require: Buildings B
Require: Functional capacity targets D
Require: Valid bottom-to-top function sequences S
Ensure: Function sequence assigned to each building

$A \leftarrow \{f \mid D_f > 0\}$
$U_f \leftarrow 0$ for all $f \in A$
$R \leftarrow B$
for all $b \in B$ **do**
  $C_b \leftarrow \text{GenerateCandidateSequences}(b, S, A)$
  $C_b \leftarrow \text{FilterByConstraints}(C_b, b)$
end for
while $R \neq \emptyset$ **do**
  $\Delta_f \leftarrow \max(0, D_f - U_f)$ for all $f \in A$
  $A' \leftarrow \{f \mid \Delta_f > 0\}$
  if $A' = \emptyset$ **then**
    break
  end if
  for all $b \in R$ **do**
    $C'_b \leftarrow \{s \in C_b \mid \forall f \in s, f \in A'\}$
  end for
  $C' \leftarrow \{C'_b \mid b \in R\}$
  $H \leftarrow \text{ComputeHostability}(C', R, A')$
  $(b^*, s^*) \leftarrow \text{SelectBestPair}(C', H, \Delta)$
  if (b^*, s^*) is invalid **then**
    break
  end if
  $\text{AssignSequence}(b^*, s^*)$
  for all $f \in s^*$ **do**
    $U_f \leftarrow U_f + \text{FootprintCapacity}(b^*)$
  end for
  $R \leftarrow R \setminus \{b^*\}$
end while

Residential, commercial, office, and transit functions are treated as flexible uses that may vertically mix with compatible functions. In contrast, parking is restricted to the bottom of compatible stacks, public space to the top of compatible stacks, and public buildings are not compatible to be mixed with, and are only allowed as standalone structures.

The ComputeHostability function estimates how many remaining buildings can still accommodate each active function. $H_{\text{any}}(f)$ counts the number of remaining buildings that can host function f in at least one candidate sequence, while $H_{\text{crit}}(f)$ identifies cases where f can only be accommodated in highly restricted ways, such as through singleton candidates. These values allow the algorithm to prioritize functions with limited future placement opportunities.

The SelectBestPair function ranks feasible building–sequence pairs using a hierarchical prioritization scheme based on function scarcity, future placement flexibility, unmet demand, and secondary preference ordering. Sequences containing functions with limited future placement opportunities are prioritized first, while additional penalties discourage assignments that would prevent critical functions from being placed later in the process. Preference is also given to assignments that improve coverage of remaining functional deficits and encourage vertical mixing when other criteria are equal. By prioritizing constrained functions before more flexible ones, the generator is able to produce more balanced functional distributions while reducing infeasible future assignments. Following sequence assignment, each selected function receives a minimum floor allocation used to initialize the subsequent FSI realization step.

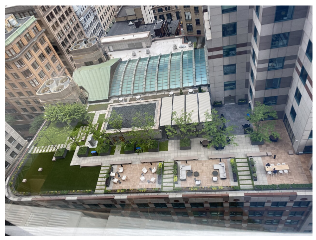
5 Results & Discussion

This section presents and discusses the results of the proposed VMI framework. Real-world examples of vertical LUM are first shown to contextualize the targeted forms of mixing, followed by a

(a) Commercial and residential [14]. Glasgow, Scotland.

(b) The Cube, Birmingham. Contains a mixture of commercial, residential, and office space [6].

(c) Underground integration [26]. Münster, Germany.



(d) Rooftop functions [5]. 75 State Street, Boston.

Figure 3: Examples of vertical LUM across different configurations.

generated multi-neighborhood city demonstrating the distribution of indices, functional FSI proportions and the generated functional compositions. Results for varying VMI targets are then examined to illustrate their effect on the generated environment and the resulting quality of vertical mixing. Finally, achieved VMI values are compared against assigned targets to evaluate the controllability, feasibility, and structural limitations of the proposed approach.

5.1 Vertical LUM in the Real-World

Figure 3 illustrates some common forms of vertical LUM in urban environments. The most typical configuration consists of retail or commercial functions at ground level with residential uses above, reflecting both accessibility and desirability: retail benefits from street exposure, while residential units are positioned above to provide separation from street-level activity. Underground spaces are commonly used for car parking, particularly where land is scarce. More complex combinations, such as office and residential mixing, also occur but are less common due to differing functional requirements and compatibility constraints. A more recent trend is the use of rooftops as functional spaces, including green areas, recreational spaces, or public amenities such as terraces and sports facilities.

Vertical LUM is not arbitrary, but guided by functional compatibility, accessibility requirements, and economic considerations. Certain combinations occur frequently and can therefore be considered relatively natural pairings, while others require stronger design or economic incentives. These recurring patterns motivate the compatibility constraints and sequencing rules employed within the procedural generation framework.

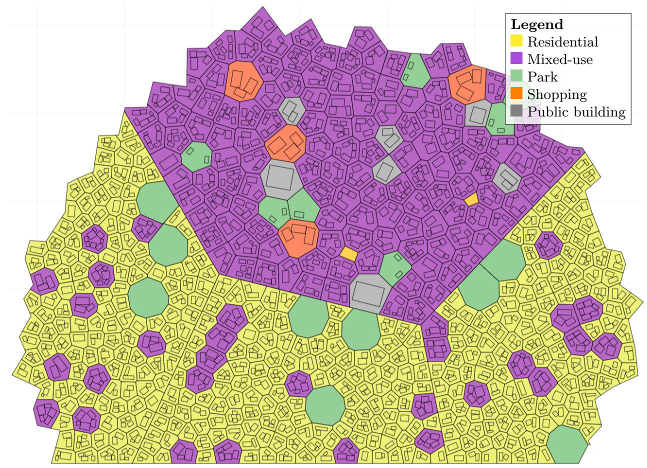


Figure 4: Layout consisting of three neighborhoods: two residential and one city center. Config: 2FSI, 0.5GSI, 1.5VMI

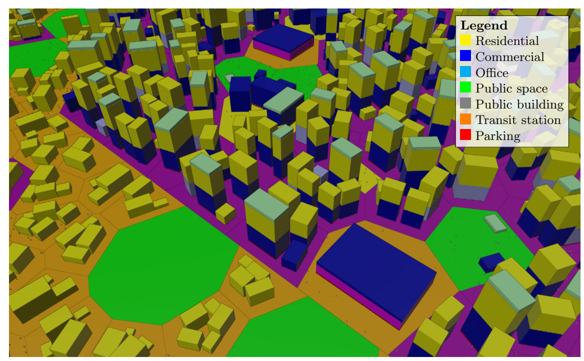


Figure 5: Corresponding 3D visualization of generated functional volumes. Config: 2FSI, 0.5GSI, 1.5VMI.

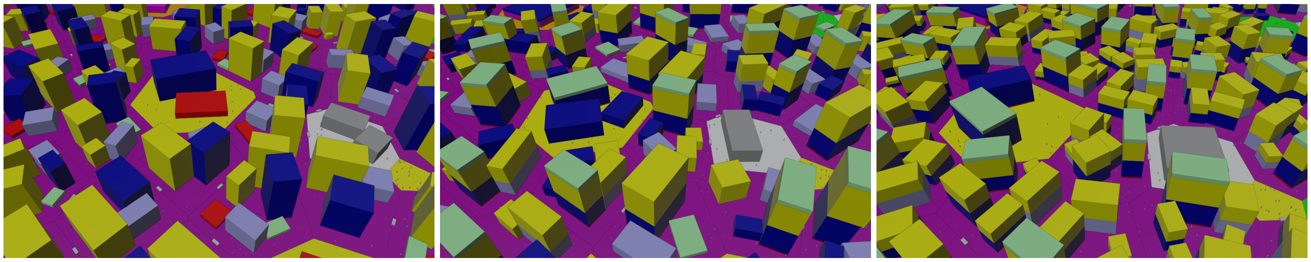
5.2 Generated Functional Urban Structure

Figure 4 shows an example of a generated city layout. Building footprints are shown in this top-down representation, while Figure 5 presents the corresponding 3D realization of the generated functional volumes. The block boundaries indicate the roads, which are left out of the visualizations for visual focus.

The example highlights the contrast between mono-functional residential blocks in residential neighborhoods and more mixed-use blocks in the city-center neighborhood. Residential neighborhoods remain comparatively mono-functional, whereas city-center regions accommodate broader vertical mixing. The following subsections further investigates how varying VMI targets affect the resulting functional composition of these urban environments.

5.3 Controllability and Effects of VMI

Figure 6 shows the effect of varying the VMI target while keeping all other parameters constant, focusing on city-center blocks.



(a) Target VMI = 1 **(b) Target VMI = 1.5** **(c) Target VMI = 3**

Figure 6: City realization keeping the same block layout for three different VMI values. 2.5FSI, 0.5GSI

At low VMI values (Figure 6a), the absence of vertical mixing forces functions to occupy separate buildings. Because each function must independently satisfy its floor-space requirements, this produces noticeable imbalances in building heights, demonstrating how accommodating multiple functions within a block without vertical LUM can result in fragmented and uneven urban morphology.

At intermediate VMI levels (Figure 6b), mixed-use configurations begin to emerge. The generator prioritizes common combinations discussed in Subsection 5.1, most notably commercial–residential pairings and underground parking integration. Some buildings additionally incorporate rooftop gardens or recreational spaces. Office functions often remain mono-functional, as sufficient mixing is already achieved through more common combinations.

At higher VMI values (Figure 6c), the generator attempts to maximize vertical overlap to approximate the assigned target. Mono-functional blocks, such as public building blocks, remain unchanged because no additional functions are requested. In contrast, mixed-use blocks begin to exhibit more complex vertical stacks, including less common combinations such as office–residential mixtures. Rooftop public space also becomes more common. As multiple functions are accommodated within the same footprint, building heights become more balanced while the resulting environments exhibit greater vertical and functional variation. In procedural virtual urban environments, this can help create more layered spaces, denser local activity patterns, and more varied traversal opportunities.

Comparing these scenarios shows that blocks preserve their intended identities while altering their internal vertical organization. This reflects the hierarchical distribution model implemented, where each block has a specific purpose and therefore specific functional demands. Blocks with restricted functional compositions remain capped regardless of increased VMI target, indicating that achievable vertical mixing depends strongly on the broader functional organization of the city hierarchy. This also illustrates one of the limitations discussed in Section 2.2: aggregate city-wide measures cannot distinguish local diversity, as blocks with no vertical mixing may still exist within a city exhibiting a high overall VMI. Overall, these observations suggest that VMI provides a meaningful mechanism for controlling vertical organization.

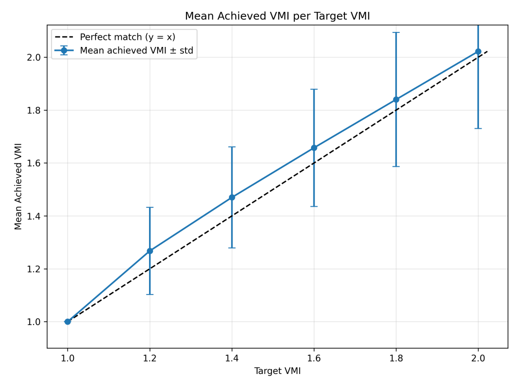


Figure 7: Mean achieved VMI for each target VMI value across generated blocks with one SD bars. 2FSI, 0.5GSI

Figure 8: VMI Closeness Heatmap on 2D layout of city center. 2.5FSI, 0.5GSI and 2VMI

5.4 Accuracy of Realized VMI

Figure 7 shows that the mean realized VMI increases approximately linearly with the target VMI, indicating that VMI functions as a controllable parameter at the city scale. A slight positive bias is observed, with achieved values exceeding their targets. This results from the per-floor function assignment used by the generator, where adjustments occur in increments that may overshoot the

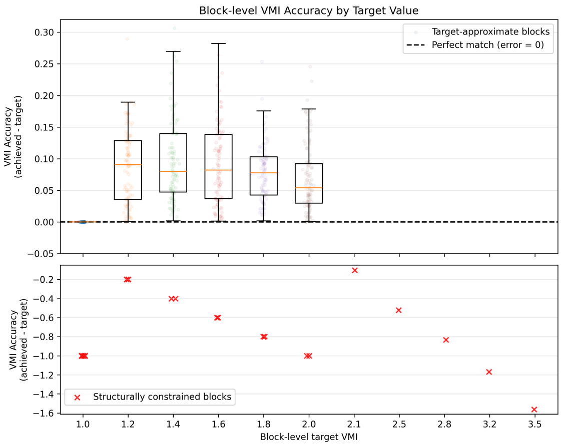


Figure 9: Block-level VMI accuracy grouped by target VMI. The top panel shows blocks that closely approximate their assigned VMI targets, with relatively small deviations. The bottom panel shows structurally constrained blocks that remain substantially below their assigned targets due to limited functional diversity or density requirements. 2FSI, 0.5GSI.

target. Finer control may be achieved through smaller building footprints or multi-functional floors.

The standard deviation bars in Figure 7 primarily reflect variation of target VMI's between individual blocks rather than failure to satisfy the aggregate city-wide target. This again highlights how city-wide averages may conceal substantial local variation.

While the aggregate results indicate strong controllability, block-level results reveal structural constraints. Figure 8 visualizes the closeness between achieved and target VMI values per block, while Figure 9 plots block-level deviations across multiple generated cities. Many blocks approximate their targets closely, typically with only a small positive overshoot as discussed earlier.

However, a subset of blocks consistently remain below their assigned VMI targets. These generally fall into two categories. The first consists of blocks with restricted functional compositions, such as public building or specialized commercial blocks, where limited function combinations constrain achievable vertical mixing. The second consists of blocks with restricted density conditions, such as plazas or parks, where insufficient floor space relative to the building footprint limits the number of achievable vertical layers.

These results suggest that deviations from target VMI often reflect structurally infeasible assignments rather than poor approximation performance. In particular, achievable vertical mixing is constrained by both functional diversity and the density measures FSI and GSI.

5.5 Theoretical Bounds on VMI

The previous observations reveal recurring upper bounds on achievable VMI across different block types. These limits can be formalized by deriving a theoretical upper bound under given density (FSI and GSI) and functional conditions.

Let the built footprint cover a fraction GSI of the land area, while the remaining $1-\text{GSI}$ contributes one function as open space. Under the idealized assumption that each floor within the built footprint hosts a distinct function, the average number of distinct functions over built columns is FSI/GSI. Therefore:

$$\text{VMI}_{\max} = (1 - \text{GSI}) + \text{GSI}\frac{\text{FSI}}{\text{GSI}} = 1 + \text{FSI} - \text{GSI}. \quad (5)$$

In practice, however, the number of available functions is finite. Let $D_{\max}$ denote the maximum number of distinct functions that may occur within a vertical column. Combining both density and functional constraints yields:

$$\text{VMI}_{\max} = 1 + \text{GSI}\left(\min\left(D_{\max}, \frac{\text{FSI}}{\text{GSI}}\right) - 1\right). \quad (6)$$

In practice, these bounds may be further constrained by the employed compatibility rules and additional system-level restrictions such as maximum building heights.

6 Conclusion

This paper introduced the **Vertical Mix Index (VMI)** as a quantitative measure and controllable parameter for vertical **land use mix (LUM)** within procedural city generation. Unlike traditional LUM measures, which focus on proportional balance or horizontal spatial relationships, VMI directly captures the average number of distinct vertically overlapping functions within an urban environment.

To investigate the applicability of VMI within computational generation, a hierarchical procedural framework was proposed in which VMI is propagated alongside density parameters (FSI and GSI) from the city scale down to districts, neighborhoods, and blocks. Within this system, blocks are generated by approximating density and vertical mixing targets. Once building footprints are created, functional volumes are assigned using a scarcity-aware assignment algorithm with compatibility and sequencing rules to approximate the assigned VMI targets.

The results demonstrate that VMI can function as a meaningful control parameter within procedural urban generation. Increasing target values produced progressively more vertically mixed environments, from largely mono-functional urban layouts to more complex configurations involving commercial–residential integration, underground parking, rooftop functions, and office–residential overlap. The experiments revealed that achievable VMI is inherently constrained by density targets, the diversity of requested functions, and compatibility restrictions between functional categories. Consequently, not all blocks are capable of satisfying their assigned targets.

These findings suggest that vertical LUM can be formalized not only as an analytical measure, but also as a controllable design parameter within generative urban systems. Incorporating vertical LUM into urban analysis may support more detailed urban comparison and analysis by capturing forms of vertical functional organization overlooked by horizontal LUM measures. Furthermore, within procedural generation, controllable vertical overlap may contribute to virtual cities with greater vertical traversal, localized diversity, and layered spatial experiences. More broadly, the framework is not inherently limited to conventional urban land uses, as the represented functions may correspond to more abstract environmental, spatial, or gameplay-related categories requiring vertically layered organization.

Future work may extend the framework through richer representations of urban infrastructure and building geometry. This includes support for stand-alone underground structures, transportation infrastructure, and elevated pedestrian systems such as skyways, which may further increase achievable vertical overlap. Building representations could also be expanded to support multiple functions within a single floor, volumetric overhangs, and tapered upper floors that introduce additional semi-open or public space. Furthermore, the hierarchical propagation of target indices may be improved through more advanced communication between parent and child zones, enabling preemptive compensation mechanisms for areas that are structurally unable to satisfy their assigned VMI targets. This may improve the distribution of vertical mixing and reduce cases where local regions fail to achieve assigned VMI targets.

References

[1] Mert Akay and Olgu Caliskan. 2025. A Parametric Approach to Plot-Based Urban Design: A Climate-Responsive Algorithmic Control for the Generation of Urban Block: A Parametric Approach to Plot-Based Urban designM. Akay and O. Çalışkan. *URBAN DESIGN International* 0, 0 (Oct. 2025), 1–22. doi:10.1057/s41289-025-00291-6 Advance online publication.

[2] Jan Beneš, Alexander Wilkie, and Jaroslav Křivánek. 2014. Procedural Modelling of Urban Road Networks. *Comput. Graph. Forum* 33, 6 (Sept. 2014), 132–142. doi:10.1111/cgf.12283

[3] Tiago Boelter Mizdal and Cesar Tadeu Pozzer. 2018. Procedural Content Generation of Villages and Road System on Arbitrary Terrains. In *2018 17th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*. IEEE, Piscataway, NJ, USA, 205–2056. doi:10.1109/SBGAMES.2018.00032

[4] Gerhard Bruyns, Christopher Higgins, and Darren Nel. 2020. Urban Volumetrics: From Vertical to Volumetric Urbanization and Its Extensions to Empirical Morphological Analysis. *Urban Studies* 58, 5 (July 2020), 922–940. doi:10.1177/0042098020936970

[5] CaribDigita. 2023. Rooftop Terrace at 75 State Street. Wikimedia Commons. https://commons.wikimedia.org/wiki/File:Rooftop_terrace_at_75_State_Street.jpg Licensed under CC BY-SA 4.0.

[6] Keith Edkins. 2016. The Cube. Geograph. https://www.geograph.org.uk/photo/5020414 Licensed under CC BY-SA 2.0.

[7] Mostafa Elzeni, Ashraf Elmokadem, and Nancy Badawy. 2021. Classification of Urban Morphology Indicators towards Urban Generation. *Port-Said Engineering Research Journal* 0 (Sept. 2021), 0–0. doi:10.21608/pserj.2021.91760.1135

[8] Saskia A. Groenewegen, Ruben M. Smelik, Klaas Jan De Kraker, and Rafael Bidarra. 2009. Procedural City Layout Generation Based on Urban Land Use Models. In *Eurographics 2009 - Short Papers*. The Eurographics Association, Groene Loper 3, 5612AE Eindhoven, 4 pages. doi:10.2312/EGS.20091045

[9] Eric Hoppenbrouwer and Erik Louw. 2005. Mixed-Use Development: Theory and Practice in Amsterdam's Eastern Docklands. *European Planning Studies* 13, 7 (Oct. 2005), 967–983. doi:10.1080/09654310500242048

[10] Jane Jacobs. 1961. *The Death and Life of Great American Cities*. Random House, New York.

[11] Jiacheng Jiao, John Rollo, and Baibai Fu. 2021. The Hidden Characteristics of Land-Use Mix Indices: An Overview and Validity Analysis Based on the Land Use in Melbourne, Australia. *Sustainability* 13, 4 (Jan. 2021), 1898. doi:10.3390/su13041898

[12] Joon-Seok Kim, Hamdi Kavak, and Andrew Crooks. 2018. Procedural City Generation beyond Game Development. *SIGSPATIAL Special* 10, 2 (Nov. 2018), 34–41. doi:10.1145/3292390.3292397

[13] Suzi Kim, Dodam Kim, and Sunghee Choi. 2020. CityCraft: 3D virtual city creation from a single image. *Vis. Comput.* 36, 5 (May 2020), 911–924. doi:10.1007/s00371-019-01701-x

[14] Michal Klajban. 2018. 415-447 Victoria Road, Glasgow, Scotland. Wikimedia Commons. https://commons.wikimedia.org/wiki/File:415-447_Victoria_Road,_Glasgow,_Scotland_02.jpg Licensed under CC BY-SA 4.0.

[15] Thomas Lechner, Pin Ren, Ben Watson, Craig Brozefski, and Uri Wilenski. 2006. Procedural modeling of urban land use. In *ACM SIGGRAPH 2006 Research Posters* (Boston, Massachusetts) *(SIGGRAPH '06)*. Association for Computing Machinery, New York, NY, USA, 135–es. doi:10.1145/1179622.1179778

[16] Thomas Lechner, Ben Watson, Uri Wilensky, and Martin Felsen. 2025. Procedural city modeling. arXiv:2507.18899 [cs.GR] https://arxiv.org/abs/2507.18899

[17] Zhongjie Lin and José L. S. Gámez (Eds.). 2018. *Vertical Urbanism: Designing Compact Cities in China*. Routledge, London. doi:10.4324/9781351206839

[18] Xiaoming Lyu, Qi Han, and de Vries. 2017. Procedural Modeling of Urban Layout: Population, Land Use, and Road Network. *Transportation Research Procedia* 25 (Dec. 2017), 3337–3346. doi:10.1016/j.trpro.2017.05.194

[19] Kevin Manaugh and Tyler Kreider. 2013. What Is Mixed Use? Presenting an Interaction Method for Measuring Land Use Mix. *Journal of Transport and Land Use* 6, 1 (April 2013), 63–72. doi:10.5198/jtlu.v6i1.291

[20] Daniel R. Mandelker. 2023. ZONING FOR MIXED-USE DEVELOPMENT. *Real Property, Trust & Estate Law Journal* 58, 1 (March 2023), 1.

[21] Zihan Mei, Yue Pan, Jack Cheng, and Jose Luis Garcia Del Castillo Lopez. 2021. Cross-Scale and Density-Driven City Generator - Parametric Assistance to Designers in Prototyping Stage. In *eCAADe 2021: Towards a New, Configurable Architecture*. eCAADe (Education and Research in Computer Aided Architectural Design in Europe), Novi Sad, Serbia, 563–570. doi:10.52842/conf.ecaade.2021.1.563

[22] Paul Mitchell Hess, Anne Vernez Moudon, and Miles G. Logsdon. 2001. Measuring Land Use Patterns for Transportation Research. *Transportation Research Record* 1780, 1 (Jan. 2001), 17–24. doi:10.3141/1780-03

[23] G. Nishida, I. Garcia-Dorado, and D. G. Aliaga. 2016. Example-Driven Procedural Urban Roads. *Comput. Graph. Forum* 35, 6 (Sept. 2016), 5–17. doi:10.1111/cgf.12728

[24] Yoav I. H. Parish and Pascal Müller. 2001. Procedural modeling of cities. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. Association for Computing Machinery, New York, NY, USA, 301–308. doi:10.1145/383259.383292

[25] Meta Berghauser Pont and Per Haupt. 2023. *Spacematrix: Space, Density and Urban Form - Revised Edition*. TU Delft OPEN Books, Delft, the Netherlands. doi:10.59490/mg.38

[26] Dietmar Rabich. 2017. Münster, LVM – 2017 – 6279-85. Wikimedia Commons. https://commons.wikimedia.org/wiki/File:M%C3%BCnster,_LVM_--_2017_--_6279-85.jpg Licensed under CC BY-SA 4.0.

[27] Alan Rowley. 1996. Mixed-Use Development: Ambiguous Concept, Simplistic Analysis and Wishful Thinking? *Planning Practice & Research* 11, 1 (Feb. 1996), 85–98. doi:10.1080/02697459650036477

[28] Christoph Salge, Michael Cerny Green, Rodgrigo Canaan, and Julian Togelius. 2018. Generative design in minecraft (GDMC): settlement generation competition. In *Proceedings of the 13th International Conference on the Foundations of Digital Games* (Malmö, Sweden) *(FDG '18)*. Association for Computing Machinery, New York, NY, USA, Article 49, 10 pages. doi:10.1145/3235765.3235814

[29] Ruben M. Smelik, Tim Tutenel, Rafael Bidarra, and Bedrich Benes. 2014. A Survey on Procedural Modelling for Virtual Worlds. *Computer Graphics Forum* 33, 6 (2014), 31–50. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12276 doi:10.1111/cgf.12276

[30] Ruben M. Smelik, Tim Tutenel, Klaas Jan de Kraker, and Rafael Bidarra. 2010. Integrating procedural generation and manual editing of virtual worlds. In *Proceedings of the Workshop on Procedural Content Generation in Games*. ACM, New York, NY, USA, 2:1–2:8. doi:10.1145/1814256.1814258

[31] Yan Song, Louis Merlin, and Daniel Rodriguez. 2013. Comparing measures of urban land use mix. *Computers, Environment and Urban Systems* 42 (2013), 1–13. doi:10.1016/j.compenvurbsys.2013.08.001

[32] Edward Teng and Rafael Bidarra. 2017. A Semantic Approach to Patch-Based Procedural Generation of Urban Road Networks. In *Proceedings of the 12th International Conference on the Foundations of Digital Games (FDG)*. ACM, New York, NY, USA, 71:1–71:10. doi:10.1145/3102071.3110569

[33] Xiaowei Zhang, Aly Shehata, Bedrich Benes, and Daniel Aliaga. 2020. Automatic Deep Inference of Procedural Cities from Global-scale Spatial Data. *ACM Transactions on Spatial Algorithms and Systems (TSAS)* 7, 2 (Oct. 2020), 6:1–6:28. doi:10.1145/3423422

[34] Yuefei Zhuo, Xin Jing, Xiaoying Wang, Guan li, Xu Zhongguo, Yang Chen, and Wang Xueqi. 2022. The Rise and Fall of Land Use Mix: Review and Prospects. *Land* 11 (12 2022), 2198. doi:10.3390/land11122198