VOLUMEN 42
Syntopia 2026

A parametric approach to creating 3D scenes

Un enfoque paramétrico para la creación de escenas 3D

Dzenis Yelyzaveta[1], Vovk Oleksandr[1]
[1] Department of Media Systems and Technologies, Kharkiv National University of Radio Electronics, Ukraine.
yelyzaveta.dzenis@nure.ua, oleksandr.vovk@nure.ua

Abstract

This paper examines a parametric approach to the creation of complex three-dimensional scenes as one of the key areas of development in modern computer graphics systems. It analyses traditional modelling methods and their limitations when scaling projects, as well as modern solutions based on procedural graphs, parametric dependencies and hybrid methods of geometry generation. Particular attention is paid to the transition from static modelling to dynamic scene description systems, in which geometry is generated based on a set of parameters and rules. It is demonstrated that parametric systems provide a significant increase in the efficiency of 3D content development, a reduction in development time and improved scene scalability.

Keywords: 3D graphics, parametric modelling, procedural systems, graph structures, scene generation, automation.

Introduction

Modern technologies for creating 3D content are undergoing a period of rapid transformation, driven by the shift from traditional manual modelling to automated and parametric methods of scene generation. The increasing complexity of digital projects in the gaming industry, architectural visualisation and virtual reality systems requires fundamentally new approaches to organising the modelling process.

Manual modelling, despite offering a high level of control over details, becomes inefficient when working with large scenes containing a large number of objects and repetitive elements. The main problems with this approach include high labour intensity, difficulty in scaling, and limited capabilities for rapid editing.

In response to these limitations, parametric and procedural modelling methods are being actively developed, based on describing a scene through a system of dependencies, rules and algorithms. In such systems, the geometry of objects is not specified directly, but is generated automatically based on input parameters and structural relationships.

Current research in the field of 3D generation shows that parametric approaches are becoming the foundation for building flexible, next-generation scene editing systems, including node-based environments and hybrid generation models.

The relevance of this research stems from the need to improve the efficiency of 3D scene creation processes and the industry's transition to automated content development pipelines.

Purpose:

The purpose of this study is to analyze the principles of parametric formation of 3D scenes, as well as to evaluate the effectiveness of its application in comparison with traditional approaches to modeling digital objects.

Objectives:

- Analyse modern methods and technologies for constructing 3D scenes in computer graphics systems.
- Study the principles of parametric modelling and identify its distinctive features compared to manual modelling.
- Analyse modern methods of procedural and graph-based geometry generation.
- Study the use of neural and hybrid models for scene generation and optimization.

- Identify the advantages of the parametric approach when modelling complex objects and scenes.
- Assess the practical effectiveness and prospects for the application of parametric systems in modern 3D modelling software.

Theoretical Foundations of Parametric Modelling

The parametric approach is based on representing a three-dimensional scene as a system of interrelated functions and dependencies. Unlike a static description of geometry, here an object is defined not by a set of vertices, but by a generation function (Alesmaa, n.d.).

At the heart of parametric modelling lies the idea of separating the description of an object from its final geometric form. The user specifies a set of characteristics, such as dimensions, proportions, the relative positioning of elements and constraints, and the system automatically generates the corresponding geometry. As a result, changing one or more parameters does not require the model to be completely rebuilt, but merely leads to its automatic recalculation whilst preserving the specified dependencies (Kleitsiotis *et al.,* 2025).

In general terms, a model can be represented as a mapping:

$$G = f(P) \tag{1}$$

where G - the final geometry of the scene, P – vector of parameters defining its structure.

Such a representation allows the scene to be interpreted as a dynamic system that changes as the input parameters vary. A key feature is the presence of dependencies between scene elements, which leads to the formation of a hierarchical data structure.

A major advantage of the parametric approach is the ability to reuse the same models across different projects. Instead of creating a large number of individual objects, the developer can use a single template, changing only the parameter values. This is particularly relevant when designing architectural objects, urban spaces and natural landscapes, where a high degree of variability is required whilst maintaining the overall structure of the scene. That this approach ensures a higher level of model reuse and simplifies the process of adapting them to different tasks. (Dzenis & Vovk, 2026a; Kleitsiotis *et al.,* 2025).

The transition from static modelling to parametric systems

The traditional model for creating 3D content is based on the sequential, manual construction of objects. Under this approach, each element of the scene is created as a separate object, the geometry of which is stored explicitly and edited directly by the user. This method provides a high level of control over the model's details; however, as the number of objects increases, the complexity of project maintenance and making changes rises significantly **(**Dzenis & Vovk, 2026a).

Formally, a scene created using the traditional method can be viewed as a collection of individual objects:

$$S = \{o_1, o_2, ..., o_n\}, \tag{2}$$

where o_i is created and edited independently of the other elements in the scene.

The main drawback of this approach is the weak interconnection between objects. Changing the properties of one element does not, as a rule, affect the other objects; therefore, large-scale adjustments require manual editing of a large number of components. In large-scale projects, this leads to increased time expenditure and raises the likelihood of errors arising from data inconsistencies (Dzenis & Vovk, 2026b).

In a parametric system, the scene is described differently – as an interconnected structure of dependencies:

$$S = F(P, R) \tag{3}$$

where P – parameters that determine the properties of objects, R – set of rules and constraints that define the logic governing their formation.

Contrary to static modelling, the parametric approach involves storing not only the final geometry but also information about how it was constructed. This allows the user to modify the model's initial parameters, after which the system automatically recalculates the associated elements of the scene. This mechanism provides a higher level of flexibility and allows the project to be quickly adapted to new requirements without the need to completely recreate objects.

Parameter-based systems offer particularly significant advantages when working with complex scenes containing a large number of similar or interrelated elements. In architectural visualisation, industrial design and the development of virtual environments, changing a single parameter can automatically update dozens or hundreds of objects, whilst maintaining the integrity of the scene structure (Alesmaa, n.d.; Ullrich *et al.,* 2010). This makes the parametric approach an effective tool for creating scalable and easily modifiable 3D projects.

Thus, changing a single parameter can lead to the recalculation of the entire scene, which fundamentally distinguishes this approach from static modelling.

Procedural graphs as an implementation of the parametric approach

Procedural graphs are a practical implementation of parametric systems, in which a scene is represented as a directed acyclic graph (Kim *et al.,* 2025). The graph's nodes perform operations on geometry or parameters, whilst the edges define the relationships between them.

This representation ensures the system's modularity and allows complex scenes to be developed as a combination of simple operations. In modern computer graphics systems, this approach is implemented as node-based interfaces, where the user interacts not with the geometry directly, but with the logic of its generation (Kim *et al.,* 2025; Dzenis & Vovk, 2026b).

One of the key advantages of procedural graphs is the ability to reuse individual parts of the graph. Once created, nodes and their combinations can be applied in various projects without the need to rebuild the generation logic. This helps standardise development processes and reduces the likelihood of errors when making changes to the scene structure.

An additional advantage is the clear visual representation of the relationships between model elements. Unlike the traditional approach, where the relationships between objects are often hidden within individual editing operations, the graph structure allows the process of geometry generation to be visually tracked at every stage of the calculation. This simplifies scene analysis and makes it easier to maintain the scene throughout the project lifecycle (Kim *et al.,* 2025).

It should be noted that the effectiveness of procedural graphs is particularly evident when working with large scenes containing many objects of the same type. In such cases, changes to the parameters of a base node are automatically propagated to all associated elements, ensuring consistency in the scene structure and significantly reducing the time required to modify it. This mechanism makes graph-based systems highly sought-after in architectural visualization, game world development and the creation of digital twins of real-world objects.

A comparative analysis of approaches

To assess the effectiveness of the parametric approach, it is necessary to compare it with other common methods of creating 3D content. In practice, developers use both traditional manual modelling and various forms of procedural generation (Table 1). Each approach has its own advantages and limitations, which become apparent when working with scenes of varying scale and complexity.

Table 1. Comparative characteristics of methods for creating 3D scenes.

Criterion	Manual modelling	Procedural systems	Parametric graphs
Scalability	Low	High	Very High
Scene update time	High	Medium	Low
Reusability	Limited	Medium	High
Control over details	High	Medium	Medium
Automation	None	Partial	Full

Source: Own elaboration.

Manual modelling provides maximum control over the geometry of objects and allows for the creation of unique scene elements with a high level of detail. However, as the number of objects increases, so does the amount of routine work, and making changes requires a significant amount of time. Procedural systems partially address this issue by automating content generation based on predefined algorithms (Ullrich *et al.,* 2010; Alesmaa, n.d.). Nevertheless, such systems are often limited by a set of predefined rules and may require additional configuration to achieve the desired result.

Parametric graphs represent a further development of the procedural approach. They allow a scene to be described through a network of interconnected parameters and operations, so that changing one element automatically affects related objects. This significantly simplifies project maintenance and makes the development process more flexible (Kim *et al.,* 2025; Kleitsiotis *et al.,* 2025).

The data presented shows that manual modelling remains an effective tool for creating individual objects and small scenes; however, its effectiveness diminishes as the complexity of the project increases. Procedural systems enable faster content production and a higher level of automation, but do not always provide sufficient flexibility in controlling the outcome. The best-balanced solution is parametric graphs, which combine a high level of automation with the ability to make rapid changes and reuse created components.

An analysis of the results shown in Table 1 indicates that the parametric approach has the greatest potential for developing large and dynamically changing scenes. Due to the ability to centrally manage parameters, it reduces development time, improves project scalability and enables more efficient use of the development team's resources (Kleitsiotis *et al.,* 2025).

The table shows that parametric graph systems provide the best balance between controllability and automation, particularly when working with large scenes.

Practical Realization in Modern Systems

The practical use of the parametric approach has become widespread in modern software packages for creating 3D graphics (Vovk & Dzenis, 2026; Vovk *et al.,* 2026). Advances in computing power and the emergence of specialised tools have enabled a shift from manually modelling individual objects to constructing entire scenes based on parameters, algorithms and dependency graphs (Figure 1). As a result, the development time for complex content has been significantly reduced and the flexibility to make changes at all stages of production has increased.

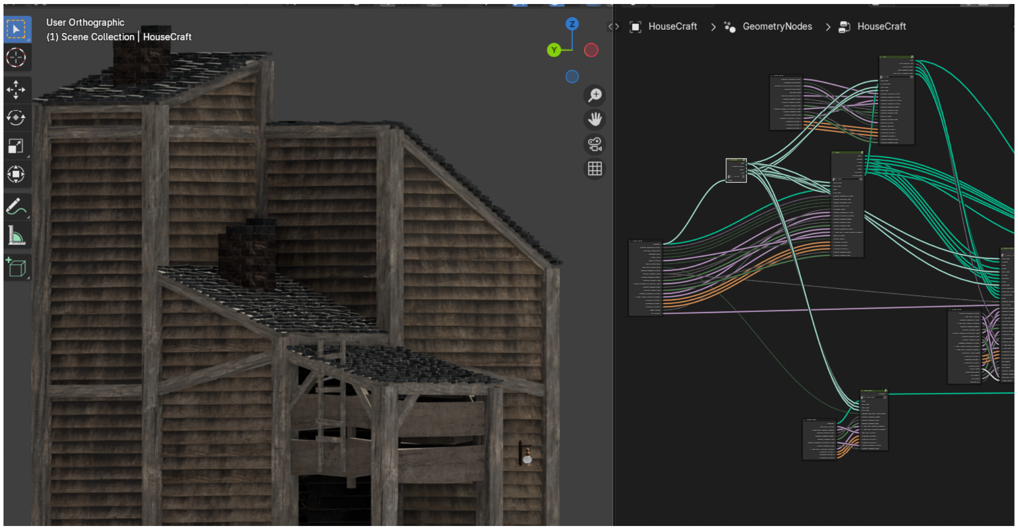

Figure 1. An example of a graph structure for parametric object generation (Blender).
Source: Own elaboration.

The most common form of parametric modelling is node-based systems, in which the process of creating an object or scene is represented as a computation graph. Each node in such a graph performs a specific operation: creating primitives, modifying geometry, generating random values, applying transformations, or combining multiple objects. The connections between nodes determine the order of computations and data transfer. This allows the user to change parameters at any stage of model construction without having to recreate it.

One of the best-known examples is the Geometry Nodes system in Blender. This tool allows you to create complex structures, such as vegetation, road networks, architectural elements and procedural landscapes, using a set of interconnected nodes. Changing a few parameters can trigger an automatic recalculation of the entire scene, whilst preserving the project's logical structure.

Another example is Houdini's procedural tools, which are widely used in the film industry and computer game development. In Houdini, virtually all elements of a scene are constructed using dependency graphs, making the system particularly effective for creating large cities, natural landscapes, visual effects and simulations of physical processes. Thanks to the parametric description of objects, developers can quickly create numerous variations of content without having to manually model each element (Ullrich *et al.,* 2010).

An important area of development for parametric systems is their integration with scripting languages. The use of Python, VEX and other languages allows the capabilities of standard tools to be extended and custom generation algorithms to be created. This approach ensures a high level of automation and allows the system to be adapted to specific project requirements. As a result, the user works not so much with the geometry directly, but with the logic of its formation and the rules for constructing the scene.

An additional advantage of modern parametric systems is their ability to integrate with artificial intelligence and procedural generation technologies. Algorithms can automatically generate object variants, distribute scene elements in space, or select generation parameters based on specified constraints (Chen *et al.,* 2025; Sun *et al.,* 2024.). This is particularly relevant for projects containing a large number of similar objects, such as scaffolding, urban blocks or industrial complexes.

Consequently, the use of graph structures, procedural algorithms and programmable dependencies enables the creation of scalable and easily modifiable scenes, making this approach one of the most promising areas of development in computer graphics technology.

Results of the study

The results of the research conducted suggest that a parametric approach to creating 3D scenes is an effective alternative to traditional modelling. The methods examined demonstrate significant advantages when working on large-scale projects that require the creation of a large number of interconnected objects and the ability to modify them quickly.

It has been established that parametric systems significantly reduce the labour required to create complex scenes by automating repetitive operations and reusing previously developed components. The use of graph structures allows for effective management of dependencies between objects, ensuring that the scene is updated consistently when the initial parameters are changed. Procedural generation helps to increase content variability and simplifies the creation of a large number of unique objects without the need for manual modelling. Furthermore, the integration of parametric methods with scripting tools expands automation capabilities and allows the generation process to be adapted to specific project requirements.

An analysis of modern software solutions shows that the parametric approach is particularly effective in architectural visualisation, game world development and the creation of virtual environments, where scalability, flexibility and the ability to make rapid changes are crucial. This not only reduces development time but also improves the consistency of the scene structure, which has a positive impact on the quality of the final result.

The proposed approach demonstrates high practical significance in the context of modern requirements for digital content production and can be regarded as one of the most promising areas of development for 3D scene creation technologies.

Conclusions

The parameter-based approach to creating 3D scenes represents the most promising direction in the development of modern computer graphics technology, focusing on the transition from labour-intensive manual modelling to intelligent and automated content generation systems. The main advantage of this approach is the ability to describe a scene using a system of parameters, relationships and rules, which ensures flexible scene management, dynamic adaptability and a high degree of scalability.

The study demonstrates that parametric systems reduce a company's resource costs (by optimising time and labour costs) for developing complex scenes, increase the level of component reuse, and ensure the consistency of the generated content. This approach is widely used in the gaming industry, architectural visualisation, virtual and augmented reality systems, as well as for modelling complex objects and large-scale scenes containing a large number of interconnected objects. The use of a parametric approach allows for the generation of a multitude of scenes based on an initial set of rules.

Prospects for further research in this field lie in the development of hybrid content generation methods that combine parametric modelling, procedural algorithms, graph-based data representation structures, and artificial intelligence technologies This will enable the creation of adaptive environments capable of automatically generating 3D content based on specified constraints and user preferences.

References

Alesmaa, M. (n. d.). Procedural 3D modeling: Concepts and techniques. *Alpha3D*. https://www.alpha3d.io/kb/3d-modelling/procedural-3d-modeling/. Accessed 25.05.2026.

Chen, Y., Shao, G., Shum, K. C., Hua, B. S., & Yeung, S. K. (2025). Advances in 3d neural stylization: A survey. International Journal of Computer Vision, 133(8), 5026-5061. https://doi.org/10.1007/s11263-025-02403-9.

Dzenis, Y., & Vovk, O. (2026a). Geometric principles of constructing 3d models. *Radio electronics and youth in the XXI century*. 2, 155-157.

Dzenis, Y., & Vovk, O. (2026b). The use of procedural modelling in the creation of 3D objects. *Memorias de SYNTOPIA*. 34-36.

Zhang, X., Gao, D., Li, N., & Dai, A. (2025). ProcGen3D: Learning Neural Procedural Graph Representations for Image-to-3D Reconstruction. *arXiv preprint,* 2511, 07142. https://doi.org/10.48550/arXiv.2511.07142.

Kleitsiotis, I., Tsirogiannis, G., & Likothanassis, S. (2025). Efficient prior specification in procedural 3D modelling. *Simulation Modelling Practice and Theory*, (144), 103165. https://doi.org/10.1016/j.simpat.2025.103165.

Sun, J. M., Wu, T. & Gao, L. (2024) Recent advances in implicit representation-based 3D shape generation. *Vis. Intell*., (2), 9. https://doi.org/10.1007/s44267-024-00042-1.

Ullrich, T., Schinko, C., & Fellner, D.W. (2010). Procedural modeling in theory and practice. *WSCG 2010, Poster Papers*. 15-18.

Vovk, O., & Dzenis, Y. (2026). Analysis of software tools for 3D character modeling. *Print, Multimedia & Web*. 2, 207-209.

Vovk, O., Dzenis, Y., & Borovynska, Y. (2026). Key stages in the development of a 3D character for computer games. *Print, Multimedia & Web*. 1, 329-332.

Zhu, C. Y., Liu, X. Y., Xu, K., & Yi, R. J. (2025). A survey on 3D editing based on NeRF and 3DGS. *Front. Comput. Sci*. 20, 2004701. https://doi.org/10.1007/s11704-025-41176-9.