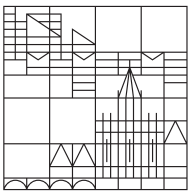
Description and Abstraction of Geometric Shapes, Natural Objects, and whole Landscapes

Dissertation zur Erlangung des
akademischen Grades eines
Doktors der Ingenieurwissenschaften (Dr.-Ing.)

vorgelegt von
Niese, Till

an der

Universität Konstanz



Mathematisch-Naturwissenschaftliche Sektion
Informatik und Informationswissenschaft

Konstanz, 2022

Konstanzer Online-Publikations-System (KOPS)
URL: http://nbn-resolving.de/urn:nbn:de:bsz:352-2-18q7vqetass5y6

Advisor

Prof. Dr. Oliver Deussen, University of Konstanz, Germany

Reviewers

Prof. Dr. Oliver Deussen, University of Konstanz, Germany

Prof. Dr. Bastian Goldlücke, University of Konstanz, Germany

Date of Submission

November 8th, 2021

Date of Defense

June 28th, 2022

Abstract

We evaluate and categorize what we see based on our accumulated experience, but more importantly, on how our visual system processes images. Shapes and patterns are essential factors here. The distribution and shape of plants, living creatures, and other natural objects are rarely purely random but are usually influenced by various factors. That is also the case for objects created by humans. The vast number of categorizations we rapidly make based on our environment assigns, therefore, a crucial role to distributions and shapes. Virtual scenes are ubiquitous in the film and game industries, urban planning, environmental simulation, and other areas of computer graphics. For each area, accurate distributions and shapes are vital because they are either essential to obtain usable information in simulation or needed to achieve the desired impact on the viewer of such a scene.

This thesis presents insights and methods for simplifying shapes while preserving their underlying information value and for purposefully generating plant distributions procedurally.
The first part of this thesis deals with the simplification of geometry. Despite increasingly powerful hardware, simplified geometries are necessary as the demand for more detailed scenes increases. However, these simplifications can significantly affect the appearance of objects. We present a method for simplifying 3D shapes based on Gestalt rules, which aims to preserve the characteristics of an object. For this purpose, 2D Gestalt principles are extended so that they can be applied to 3D space. Thus, critical visual structures can be preserved compared to previous methods. Two user studies validate the effectiveness of this method. A further study will investigate the impact of such simplifications on the categorization of these objects.
The second part of the thesis deals with the procedural generation of plant distributions. Especially in the field of urban planning and computer graphics, precise and plausible distribution of plants is crucial. They either serve a purpose or are needed to generate plausible distributions synthetically or reconstruct a given environment's characteristics based on actual data. We define procedural planting strategies and combine them with botanical rules such as distances between plants, planting density, or plant species diversity. The procedural model is easily controllable and can also reconstruct actual distributions while respecting botanical rules. For this purpose, we present a method that first generates semantic segmentation from satellite images, which can then be used to determine the parameter values for a planting strategy.

Finally, we provide insights into possible further research directions. For this, we investigate, among other things, how we can incorporate the findings from the user study into the simplification process, which possibilities exist in the area of procedural generation to extend it to other application areas, and which other factors can be included. We would also like to explore how the planting strategies can be extended to small vegetation.

Zusammenfassung

Wie wir Gesehenes bewerten und kategorisieren, basiert auf unseren gesammelten Erfahrungen, aber vor allem auch darauf, wie unser visuelles System Bilder verarbeitet. Formen und Muster sind hierbei ein wichtiger Faktor. Die Verteilung und das Aussehen von Pflanzen, Lebewesen und anderen natürlichen Objekten sind selten rein zufällig, sondern meist auf das Zusammenspiel verschiedene Faktoren zurückzuführen. Dies gilt auch für durch Menschen geschaffene Objekte. Die Vielzahl der Kategorisierungen, die wir schnell aufgrund unserer Umgebung treffen, weist den Verteilungen und Formen eine entscheidende Rolle zu. Virtuelle Szenen sind in der Film- und Spieleindustrie, der Städteplanung, der Simulation von Umwelteinflüssen und anderen Bereichen der Computergrafik allgegenwärtig. Für jeden dieser Bereiche sind korrekte Verteilungen und Formen wichtig, da sie entweder essenziell sind, um bei Simulation verwertbare Informationen zu erhalten, oder benötigt werden, um den gewünschten Eindruck beim Betrachter zu erreichen.

In dieser Arbeit werden Erkenntnisse und Methoden zur Vereinfachung von Formen unter Beibehaltung ihres Informationsgehaltes und zur zielgerichteten prozeduralen Erzeugung von Pflanzenverteilungen vorgestellt.

Der erste Teil dieser Arbeit beschäftigt sich mit der Vereinfachung von Geometrie. Trotz zunehmend leistungsfähigere Hardware ist es notwendig Geometrien zu vereinfachen, da gleichzeitig die Anforderung an immer detailreicheren Szenen steigt. Diese Vereinfachungen können jedoch das Aussehen von Objekten stark beeinflussen. Wir stellen eine Methode zur Vereinfachung von 3D-Formen auf der Grundlage von Gestaltregeln vor, welche das Ziel hat, die Charakteristik eines Objektes zu erhalten. Hierfür werden die 2D-Gestaltprinzipien erweitert, damit diese auf den 3D-Raum übertragen angewendet werden können. So können im Vergleich zu bisherigen Methoden wichtige visuelle Strukturen erhalten werden. Zwei Nutzerstudien validieren die Effektivität dieser Methode. In einer weiterführenden Studie werden die Auswirkungen solcher Vereinfachungen auf die Kategorisierung dieser Objekte untersucht.

Der zweite Teil der Arbeit beschäftigt sich mit der prozeduralen Generierung von Pflanzenverteilungen. Gerade im Bereich der Stadtplanung und der Computergrafik ist eine präzise und plausible Verteilung von entscheidender Bedeutung, da diese entweder einen Zweck erfüllen oder benötigt werden um die Charakteristik einer bestimmten Umgebung synthetisch generieren oder basierend auf realen Daten rekonstruieren zu können. Hierfür definieren wir prozedurale Pflanzstrategien und koppeln diese mit botanischen Regeln. Diese Modell ist einfach kontrollierbar und kann somit auch zur Rekonstruktion von realen Verteilungen, unter Berücksichtigung botanische Regeln, verwendet werden. Hierfür stellen wir eine Methode vor, die aus Satellitenbildern eine semantische Segmentierung erzeugt, welche für die Ermittlung der Parameterwerte einer Pflanzstrategie verwendet werden kann.

Abschließend geben wir Einblicke in mögliche weitere Forschungsrichtungen. Hierfür untersuchen wir unter anderem, wie die Erkenntnisse aus der Benutzerstudie in den Vereinfachungsprozess einfließen können, welche Möglichkeiten es gibt, um diese auf zusätzliche Einsatzgebiete zu erweitern und wie wir die Pflanzstrategien auf Kleinstvegetation erweitert werden können.

Acknowledgements

This thesis was made possible by a number of people to whom I would like to express my gratitude at this point. First and foremost, I want to thank my advisor Prof. Dr. Oliver Deussen, who gave me the opportunity to work in his group over many years. He was always available to provide me with guidance and championed all my endeavors. He gave me the freedom to work on my areas of interest, and his ideas and feedback were invaluable in advancing my research. I would like to thank my second advisor, Prof. Dr. Bastian Goldlücke for reviewing this thesis and his valuable comments.

Special thanks go to Julian Kratt, it was exciting and fun to collaborate with him in our joint project that became an essential part of my thesis; to Jochen Görtler and Marc Spicker for their many inspiring, profound, and motivating discussions, and Patrick Paetzold, who provided valuable input to my thesis and supported me with his technical knowledge for other projects at the group. I would also like to thank Ferdinand Eisenkeil, Marvin Gülzow, Rebecca Kehlbeck, Thomas Lindemeier, Mariam Mahmoud, Jens Metzner, and Thilo Spinner. Without them, the time in the group would not have been nearly as exciting and enjoyable.

Special thanks to Boris Neubert, who encouraged and supported my interest in computer graphics, to Sören Pirk, who continued to mentor and challenge me throughout my PhD; and Prof. Dr. Bedrich Benes for his invaluable contribution in numerous projects, they all made a significant contribution to my work. Without their collaborations and many intense discussions, this work would not have been possible.
A heartfelt thank you to Ingrid Baiker and Claudia Widmann for their open ear for all problems and their support in all administrative matters.

I am deeply grateful for the support of my parents, who have accompanied me throughout my life, giving me moral and emotional support, and I would also like to thank Anna Kern. They are the most important people in my life, and their love, encouragement, support, and patience helped me get through challenging times.

Contents

CHAPTER 1

Introduction

The creation of images and shapes is omnipresent in human history, starting with cave paintings, through sculptures and paintings, to synthetically created images using computers in modern times. They usual common element is that the creator wants to convey information or impressions to the viewer.

Synthetically generated scenes are becoming increasingly important. In the film industry, a whole set is often entirely constructed or at least enhanced with the help of computer programs. The imagery in games is becoming more and more realistic. Furthermore, computer-aided methods are increasingly used to create realistic representations needed for simulations or in planning, such as architecture or urban planning. An ever-increasing number of detailed shapes and objects must be designed and controlled by the artists to meet the growing demand for accuracy and realism, which is a time-consuming and elaborate task. This demand for more detailed representations makes it more difficult for artists to create them. Their increasing geometric complexity raises the resources required and thus the requirements of the hardware for processing and storing them. Therefore an ongoing topic in computer science is to research new methods to support artists in that process.

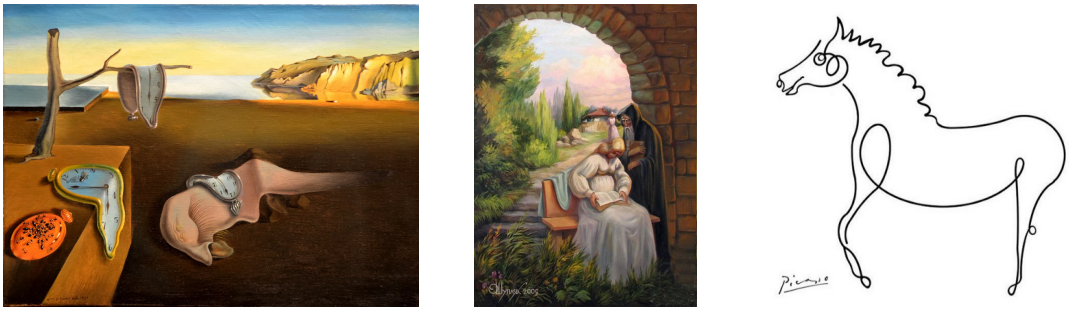
To be able to pass on information in an optimal and targeted way or to achieve the desired impression and at the same time being able to achieve that within the hardware limitations, we must understand how we process what we see and which parts of objects or distributions are essential to pass on certain information. Artists can use their experience and knowledge to perform such tasks. However, for computer-aided methods, formalizations must be found and transferred into algorithms. A substantial part of the formalizations originates from how we assess and classify what we see.

How we perceive our environment, classify these perceptions, and reach decisions based on them is determined by how the sensory perceptions are processed and what reference information we have gathered over time. The distribution and shape of living creatures and plants, or objects created by them, are rarely random. They are influenced by evolution, fulfill a purpose, or are guided by their habits and behavior.

This thesis focuses on the significance of shapes and distributions, particularly how simplification can affect their expression, how the essence of their expression can be preserved when simplified, and how this expression can be produced in procedural generation. This thesis is structured according to these parts.

1.1 Abstraction and Simplification of Shapes

Simplifying or manipulating shapes to draw focus to a specific area, create the desired impression, or reduce to specific information is a technique artists have always used in a wide variety of fields. How artists purposefully use such simplification or manipulation of shapes can be seen in figure 1.1.



(a) Salvador Dalí (b) Oleg Shuplyak (c) Picasso

Figure 1.1: Artists used the way we perceive (a) to purposefully give objects different properties through deformation, (b) to create an optical illusion by combining objects in such a way that they form another shape (c), or to make abstractions, as in line drawings.

This problem has inspired researchers to develop computer-aided computational models and representations that support the artist in this task or automate it completely. For this, it is necessary to understand what information shapes can contain and how our perception works.

Objects can contain a variety of visual information from which conclusions can be drawn about their properties. However, humans usually need only a few key features for an object's initial recognition and classification. Which features are essential for recognition is significantly influenced by how our visual system functions [PTK87; FG+03].

The shape of objects and their perceived characteristics are rarely purely random. Manufactured shapes are guided by functionality, such as stability, aesthetics, or fulfilment of other tasks, such as protection from environmental influences. The shape of living things and plants evolved over time to optimally adapt to their environment. These adaptations can be influenced by available resources, other organisms, or climatic conditions, among other factors. The appearance of natural objects, such as landscapes, is determined by

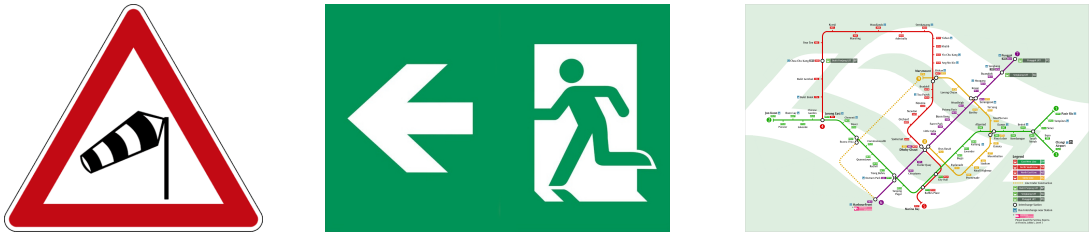
physical factors such as erosion. Everything we see thus contains visual clues to the semantic information that can be assigned.

The visual perception of living organisms has steadily adapted over time to extract the information needed for survival in the best possible way. How living organisms perceive their environment and the appearance of life forms is an evolutionary, mutually dependent process.

Visual perception is a multi-step process. Our visual system does not simultaneously process everything we see. Some of the visual information reaches our brain faster than others, is processed in different locations, and is used in different stages of our decision-making process [GL13; FG+03; LH87]. Our perceptual system has evolved so that the visual system does not process all the information it sees at once but instead performs a process of abstraction. Only certain information is therefore used for initial classification. For unconscious, quick decisions, these first impressions are of crucial importance. This visual information is combined with our learned information or past experiences to categorize and draw conclusions from what we perceive. However, which details of an object are crucial depends not only on the visual system alone but also on the context in which it is perceived and what information is essential to viewers.

How we perceive and process things is thus influenced by our environment and the knowledge acquired over time. Therefore, this and the way our visual perception works are vital information to consider for a simplification or abstraction process.

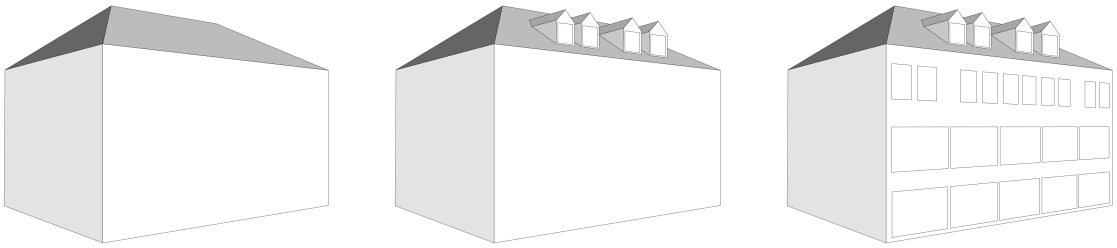
Since an object can contain more information than we need, our perception filters out unimportant details. A targeted simplification of an object down to its essential features can relieve our visual system and thus achieve easy and more quick recognition and processing of critical information. Among the best-known examples are symbols on information boards, traffic signs, or other informational signs. (see Fig. 1.2).



(a) (b) (c)

Figure 1.2: Simplifications are used to convey crucial information to the viewer quickly and clearly. Examples are (a) traffic and (b) information signs, where symbols are reduced to the most important and easily recognizable. Reducing details is not the only way to achieve simplifications, (c) changing the spatial arrangement, like using an octilinear layout, as in subway maps, can also reduce the complexity.

However, since every detail of an object also contains information to a certain extent, omitting it can increase the scope for interpretation. While omission can be applied to art as a desired stylistic technique, it can be problematic for other domains where information must be conveyed precisely, as it can bias the statements. (see Fig. 1.3)



(a) (b) (c)

Figure 1.3: (a) Shows an abstraction of a house having all windows removed. Due to missing features, it might not be categorized as a building. (b) With roof dormers added, we could likely categorize it as a residential building. (c) Adding windows modifies the classification further. Because of the large windows, the house is likely to be classified as a mixed-use house: commercial on the first floor, perhaps an office on the second, and a residential on the third.

Another use case for simplification or abstraction is to reduce the complexity of 3D geometry to be processed technically. This may be necessary, for example, for miniatures, as in the field of toys or model-making, or in 3D printing, where the manufacturing process limits the accuracy. However, simplifications are also crucial for real-time or offline rendering. The increasing demands for realism and accuracy of scenes and objects, and the resulting complexity, conflict with the memory and computational power constraints imposed by the hardware needed to render them. Reducing geometry can save computing power and thus enable real-time applications or provide faster results in offline rendering. Level of detail techniques are already used to solve that task and display objects with different complexities depending on their distance or size. However, generating the individual levels of detail for a 3D model while maintaining its expression is not trivial even with manual methods and finding automatic methods for this is a challenging task [Mit+13].

In this thesis, we present a new approach to shape abstraction and simplification that addresses specific issues in these problem domains:

Geometric Shape Analysis. A distinction must be made between two types of input models for the abstraction and simplification process.

First, there are models with a simple geometric representation consisting of polygon meshes. Their meshes are often the result of 3D scanning or photogrammetry. They can vary significantly in complexity and quality, resulting in meshes with poor connectivity or extreme cases consisting only of an accumulation of disconnected or incorrectly connected polygons. These models usually have little or no available information about

their semantics or structural composition. Thus, these models contain little information to guide a process of abstraction and simplification. Therefore, it is a great challenge to find algorithms for such models that can analyze them robustly and thus extract important structures and information about their composition.

Second, there are models which are manually generated by artists or by procedural models. Their meshes are generally of much better quality and usually contain information about their parts in the form of an object hierarchy. Nevertheless, analyzing their structure further may be necessary to find patterns for incorporation into the refinement process.

Perception. Since the abstraction and simplification of shapes are highly related to human perception, insights from cognitive science must also be accounted for in shape analysis in addition to purely geometric analysis. Besides examining the visual quality of the method, investigating how such abstractions affect the categorization of objects is also necessary. After a simplification, the classification of an object by the user may not only have changed from a specific to a more generic object category but also switched to a different non-matching specific object category. It, therefore, is vital to examine these effects and investigate how this can be incorporated, constraining certain simplification of the method.

Production of New Geometry. Once structural information is available, appropriate methods are required to simplify or abstract the model according to the required use case and statement. For efficient rendering, different levels of detail are required. These successively reduce the geometric complexity of the model so that continuous blending is possible. Furthermore, the initial visual impression must be preserved. Creating such methods is challenging, especially for models described by many geometric parts with semantic relationships. Therefore, simplification methods have to be designed and developed, which work with the specific geometric descriptions of the models and include the structural information from the analysis step. When abstracting, but also when simplifying objects, geometry is not only removed but also partially summarized, necessitating the re-creation of geometries. In the field of 3D printing, the possible level of detail is limited by the constraints of the printing process. While support structures can solve problems such as overhangs, fragile structures that exceed the printer's resolution can cause problems during printing. In these cases, the geometry must be simplified. For details such as braces of bridges with many small, separated elements, this can be solved by replacing these elements with a relief.

User-assisted Interaction. Involving the user so that they can control the abstraction according to their intentions is another crucial aspect because there is no guarantee that automated methods will always produce the exact result and aesthetic quality desired by the user. By involving the user, they can precisely influence the process and produce a result satisfying the user's perceptual needs. Therefore, the used algorithms and methods must have specific properties. An effective interactive abstraction tool must be easy and

intuitive to use, and coherent geometric models that retain the semantic structure of the original model have to be produced.

1.2 Distributions and Procedural Modeling

The most accurately possible representation of reality plays a significant role in computer graphics. Therefore, researchers are always looking for new methods to algorithmically represent various natural phenomena to be able to generate realistic images. Besides methods to represent physically correct processes such as cloth, fluids, or wind simulations, objects' distribution and interactions [Pir+12b] play an essential role.

Like appearance of natural object, plants, and life forms, their distribution is not purely random but follows rules and is due to a complex interaction of various factors. An important subfield of computer graphics is, therefore, the procedural generation of plausible structures and distributions.

Vegetation, as an essential component in our daily life, is present in almost all virtual scenes. Climatic conditions influence the occurrence, appearance, and frequency of a particular plant species. Additionally, environmental factors such as wind, soil conditions, or the available nutrients are critical influencing factors. Apart from natural factors, humans play an additional important role. Especially in urban or agriculturally used areas, the occurrence of plants is significantly controlled by humans. The distribution of plants and their species can therefore be used to derive far-reaching information. It allows us to conclude about the usage, wealth based on how well it is maintained, and the region of the area. (see Figure 1.4).

A complete simulation of these complex relationships is a time-consuming process. This work focuses on how these complex relationships can be approximated by a method that provides the possibility for the user to control and adjust the result and how this method can be extended using machine-learning techniques to automate the process of reconstruction.

In this thesis, we present a new method of formalizing plant distributions in the context of urban environments, which addresses specific issues in the following problem domains:

Modeling. A variety of different methods are available for creating convincing distributions of trees and plants. These methods fall into the following categories, among others: procedural, data-driven, manually placed, or biologically motivated. While biologically motivated simulation can provide the most accurate results, it requires a lot of computation time and detailed information about all environmental factors. In most applications


(a) Brazil, Rio de Janeiro


(b) Ireland, Celbridge


(c) Italiy, Roccasecca


(d) Japan, Tsuruga


(e) Sweden, Östersund


(f) Switzerland, Luzern

Figure 1.4: These Google Street View Images show different climatic and regional areas. On the basis of the combination of architecture and the identified plant species, we can narrow down its location.

in computer graphics, where vegetation is required, continuous simulations of the formation and change of vegetation are unnecessary, and a realistic snapshot is sufficient. Due to the complexity underlying the distributions of plants and trees, these underlying biological principles must be understood for a procedural generation of realistic plant distributions. Especially for the generation of plant positions in urban environments, only simple rule-based models exist so far. New methods incorporating complex biological factors and providing plausible results despite incomplete data are needed.

User-Assisted-Interaction. A procedural model is well-suited for filling large areas with plausible plant distributions. Involving the user to control the procedural model according to his preferences is, nevertheless, another essential domain. An automatic method does not always generate the results desired by the user, especially when considering locations with a distinctive appearance; thus, it is vital that the user also can influence the result of the model in a controlled manner. Therefore, the algorithms and methods used must have specific properties. The adjustment of parameters must be intuitive and provide comprehensible results. The user should be able to define the areas in which the adjustment of the parameters is to be applied with a simple and intuitively usable tool while maintaining plausible distributions and correct interactions with the environment.

Machine Learning. Machine learning has gained immense importance in all areas of science in recent years. Also, in computer graphics, machine learning is used to solve various problems such as enhancementing images, reducing noise, or supporting procedural modeling. For the distribution of plants, end-to-end learning could be used to generate

the corresponding tree positions based on environmental information such as building type, roads, or region. Since automatic methods do not always produce an artist's desired results, finding methods that allow further adjustments is crucial. For this purpose, suitable models have to be found that efficiently describe the distribution of plants, allow an intuitive way to control the outcome with meaningful parameters, and can be used in combination with machine learning.

1.3 Summary of Contributions

This thesis is based on several papers I primarily published in cooperation with other people. Here I list my contributions to these joint projects:

- An interactive method that allows users to specify how abstractions of complex 3D models should be performed through just a few strokes. The basic idea was to apply the known Gestalt principles to generalize the user input and derive a model abstraction from it, considering and preserving the shape, semantics, and perceptual patterns. For this purpose, the 2D Gestalt principles are extended to 3D space. The efficiency and effectiveness were tested and proven with two user studies. The author of this thesis was involved in developing the main contribution, which included implementing the application, preparing the user study, implementing the analysis of the 3D geometry, adapting the Gestalt principles to the 3D space, regenerating new geometry required by the abstraction, and writing the text for the paper. The results were published in

  J. Kratt, T. Niese, R. Hu, H. Huang, S. Pirk, A. Sharf, D. Cohen-Or, and O. Deussen. „Sketching in gestalt space: interactive shape abstraction through perceptual reasoning“. In: *Computer Graphics Forum*. Vol. 37. 6. Wiley Online Library. 2018, pp. 188–204

- A user study that explores how abstractions such as the one in the above paper affect the perception and categorization of buildings. The idea here was to determine which features on a building are essential to categorize it correctly and how these insights can be incorporated into this simplification process in future work. The author of this thesis was involved in developing the main contribution, which included extending the simplification method presented in the previous paper to specifically generate different simplifications of facades, preparing the user study, and writing the paper. The final results were published in

P. Tutzauer, S. Becker, T. Niese, O. Deussen, and D. Fritsch. „Understanding human perception of building categories in virtual 3D cities: a user study“. In: *XXIII ISPRS Congress, Commission II*. 2016, pp. 683–687
P. Tutzauer, S. Becker, D. Fritsch, T. Niese, and O. Deussen. „A study of the human comprehension of building categories based on different 3D building representations“. In: *PFG Photogrammetrie, Fernerkundung, Geoinformation* (2016), pp. 319–333

- A method that allows urban landscapes to be populated with complex 3D vegetation in both automated and user-guided ways. The idea is to introduce procedural placement models (PPMs) for vegetation in urban layouts that are environmentally sensitive to city geometry and identify plausible plant positions based on structural and functional zones in an urban layout. For this purpose, commonly used planting strategies are identified and formalized as PPMs. In addition, a method is presented that uses machine learning to derive the parameter values from satellite imagery required to produce plant distributions resembling those on these images. The efficiency and effectiveness were tested and proven with two user studies. The author of this thesis took the lead in the project and was directly involved in developing the main contribution, which included implementing the application, identifying the planting strategies, formalizing the PPMs, preparing the user studies, and writing the text for the paper. The final results were published in

  T. Niese, S. Pirk, M. Albrecht, B. Benes, and O. Deussen. „Procedural Urban Forestry“. In: *ACM Trans. Graph.* 41.2 (Mar. 2022). doi: 10.1145/3502220. url: https://doi.org/10.1145/3502220

1.4 Further Publications

Other publications in which the author of this thesis has participated are listed below. These works are not part of the thesis.

- A method that allows the calculation of the stages of development of a static tree model and approximates the natural growth of a tree. One of the main problems is to generate branches that have been pruned or died over time based on structural similarities. In addition to plausible growth animations and the generation of earlier tree stages, the method also allows the selective application of this method to individual parts of the model, which enables the user to utilize it as a modeling tool. The author of this thesis was involved in developing the main contribution and took the lead in developing the method, which included implementing the application, implementing the tree structure analysis enabled by a rotation base representation of the graph, developing the growth algorithm, and creating the user interface and

the interactive tool. Furthermore, the author contributed to the writing of the paper.

S. Pirk, T. Niese, O. Deussen, and B. Neubert. „Capturing and animating the morphogenesis of polygonal tree models“. In: *ACM Trans. on Graphics* 31.6 (2012), 169:1–169:10

- A novel method incorporates the effects of turbulent wind fields on tree growth into the growth model shown in our previous work. Since the appearance of trees is strongly influenced by external factors such as wind, the idea was to integrate wind into the growth model and also to be able to use it as a modeling tool. For this purpose, we extended the development model to include reactions to external stress by wind fields in the growth process. For this purpose, the model was extended to include effects such as branch breakage, bud abrasion, and force-induced deformation and stabilization. The author of this thesis was involved in developing the main contribution, which included creating the interface between the wind simulation and the growth model, incorporating the deformation and stabilization caused by force applied by the wind and the effects of branch breaking and bud abrasion on the growth, and contributed to writing the paper.

  S. Pirk, T. Niese, T. Hädrich, B. Benes, and O. Deussen. „Windy Trees: Computing Stress Response for Developmental Tree Models“. In: *ACM Trans. on Graphics* 33.6 (2014), 204:1–204:11

- A framework to distribute point samples with controlled spectral properties using a regular lattice of tiles with a single sample per tile. The core of the method is a word-based identification scheme to identify individual tiles in the lattice. The recursive scheme permits tiles to be subdivided into smaller tiles. For blue noise with varying densities, a bit-reversal principle to traverse sub-tiles recursively is employed. It is well-suited for different sampling scenarios in rendering, including area-light sampling (uniform and adaptive), and importance sampling. Other applications include stippling and distributing objects. The author of this thesis was involved in developing the main contribution and verifying its accuracy and correctness, including developing a reference renderer that allows comparing the sampling method presented in this paper with other state-of-the-art sampling methods, testing the sample with plant distributions, and generating the corresponding results for the paper.

  A. G. M. Ahmed, T. Niese, H. Huang, and O. Deussen. „An adaptive point sampler on a regular lattice“. In: *ACM Transactions on Graphics : TOG* 36.4 (2017). Article Number: 138. DOI: 10.1145/3072959.3073588

1.5 Thesis Structure

This thesis is structured as follows:

In Chapter 2, related works and state-of-the-art techniques are reviewed. Particularly, related methods for analyzing 3D shapes concerning segmentation, symmetries, and repetitive structures are discussed. Moreover, research in the human perception of geometric building structures and quantification of human perception of geometries is discussed. Lastly, we provide insight into the related work focusing on problematic aspects of plant and urban modeling, ecosystems, and learning-based methods. The content of this chapter, however, is given without claiming to provide a complete overview.

Chapter 3 presents a new method for a user-guided shape abstraction that takes the model's perceptual patterns and semantic structures into account by considering and formalizing 2D Gestalt principles in 3D space. The user can control the process using only a few simple sketches for the intended abstraction. Chapter 4 examines how the method proposed in Chapter 3 for simplifying objects affects their categorization. Here, we focus on how changes in the arrangement and size of facade objects affect the classification of buildings into use categories, such as residential, commercial, or industrial. For this purpose, the method proposed in Chapter 3 is used to generate different simplifications for a building.

Chapter 5 presents a method that allows filling virtual city models with plausible tree distributions or reconstructing tree positions for existing city models using satellite images. For this purpose, we first identified the common planting strategies and formalized them into a procedural placement model.

Chapter 6 concludes the thesis with a summary of the presented contributions and an overview of possible topics for future work.

CHAPTER 2

Related Work

What information shapes and distributions contain and how they affect our categorization process play an essential role in many research areas of computer graphics. Therefore, this has been studied in the areas of procedural generation of objects and distributions, abstraction and simplification, geometric modeling and shape analysis, and human perception. The areas of procedural generation and information-preserving abstraction and simplification have parallels, particularly in shape properties and their interaction with human perception. The foundations for shape properties lie in the areas of segmentation, symmetries, and regular structures. In the field of shape simplification, local and global algorithms operating on polygonal meshes have been introduced. For procedural generation, grammars, L-system, and shape grammars are important research areas. In the following section, we provide an overview of the essential works in these research areas.

2.1 Shape Analysis

For a purposeful simplification of shapes, a deeper understanding of them is required. Shape analysis describes the process of extracting relevant information from a shape. Considering that shapes described by polygonal surface mesh representations are commonly used in computer graphics, this work focuses on them. Many of the objects to be simplified often consist of only one or a few connected polygonal meshes. Segmentation of these meshes is thus a fundamental technique in shape analysis and serves as a starting point for many abstraction and simplification methods. Segmentation allows essential details of an object to be identified and relationships such as hierarchies, regularities, and symmetries to be determined. Especially, these structural relationships are vital because they often substantially impact human perception.

2.1.1 Mesh Segmentation

Mesh segmentation is the first important step in obtaining information about objects consisting of a single contiguous polygonal surface. For this purpose, this surface is divided into disjoint sets of polygons. A variety of segmentation methods exist for this purpose. For example, a subdivision can be performed based on geometric information such as edges or other surface properties. In the following, we overview different strategies. We use the structuring given in Shamir [Sha08].

Region-Growing. This technique uses a region-growing strategy. Therefore a polygon is chosen as the seed for a region. To this region, the surrounding polygons are added iteratively. Various geometric properties are used as criteria for joining. If none of the remaining surrounding polygons satisfy these criteria, a new region is started. Geometric criteria used in various methods include linear planarity [KT96], curvature [LDB05], or convexity [Cha+95; She07]. Besides single source region growing methods, methods [Eck+95; AFS06b; Lév+02] have been presented using multiple source seeds.

Hierarchical Clustering. Hierarchical clustering is similar to region-growing but solves some of its problems. For region growing, the result and the number of regions depend heavily on the choice of the initial seed. In contrast to region-growing, hierarchical clustering considers all clusters when choosing the best merging strategy. As with the region growing techniques, the difference between the various algorithms is primarily the geometric criteria used for merging. Attene et al. [AFS06c] use primitive fitting, and Garland et al. [GWH01] use a measure of planarity to evaluate the best merging strategy.

Iterative Clustering. For the previous two methods, the number of resulting clusters is unknown, and therefore this method is called non-parametric. Iterative clustering performs a parametric search for which the user selects a specific set of polygons as a prior. The Lloyd or Lloyd–Max algorithm is used as the basis for this approach. Each element is assigned to a cluster, and k representatives are re-evaluated in each iteration until the representatives stop changing. Due to that, the central concern with this method is convergence. Shlafman et al. [STK02] present a method to produce a segmentation of two objects that supports morphing between them. This method uses a distance measure between two surfaces, defined as a weighted combination of the difference in dihedral angle between the faces, to determine the cluster assignment. In *"Variational Shape Approximations"*, Cohen-Steiner et al. [CAD04] present a variant of the k-means algorithm. They use a set of proxies to represent the input mesh and apply an iterative clustering based on two distance metrics. This method was extended by Wu and Kobbelt [Wu 05] to allow the usage of planes, spheres, cylinders, and rolling ball blend patches besides ellipses.

Spectral Analysis. Spectral clustering is an algorithm that uses eigenvectors (and eigenvalues) of an appropriately defined matrix for partitioning. This method has its origin in spectral graph theory [CG97; Fie73]. The basic idea is to transform the data into a weighted graph where each node represents a pattern, and each weighted edge considers the similarity between two patterns. Thus, this method can be viewed as a graph cut problem.

In this section, a brief overview of the different segmentation methods has been provided. The survey published by Shamir [Sha08] offers a more detailed and comprehensive overview of these and other methods.

2.1.2 Symmetries and Regular Structures

Besides identifying individual parts within an object, the knowledge about their relationship is another critical aspect for an information-preserving simplification. We conclude that the relationship between objects is based on symmetries and regular patterns, so it is essential to recognize and preserve them in the simplification process. For this, similarities in geometries must be found, whereby possible transformations such as translations, rotations, and scalings must also be considered. There is a wide range of methods that address the detection of these symmetries. In this section, however, we restrict ourselves to global and partial symmetries and regular structures derived from them. A detailed overview of further methods can be found in Liu et al. [Liu+10] for images and in the surveys published by Mitra et al. [Mit+12b] and van Kaick et al. [Van+11] for 3D geometry.

Global Symmetry Detection. For global symmetry detection, transformations are needed that map the entire object onto itself. This method works on the entity of an object and therefore does not require any segmentation, which simplifies the problem space significantly. Among the first works addressing this topic is Atallah et al. [Ata84], which presented an $O(n \log n)$-optimal algorithm that detects all reflective symmetries for plane figure. Alt et al [Alt+88] presented an algorithm that uses rigid transformations on point sets to express exact or even approximate congruences and symmetries. By computing reliable global shape descriptors, the computations of global symmetries can be further simplified. Methods based on this approach include the method of Sun et al. [SS97] which utilizes the correlation of the extended Gaussian image, moment coefficient based methods [TMS08], or methods based on spherical harmonic coefficients [KFR04].

Partial Symmetry Detection. In addition to shapes exhibiting these *global symmetries*, there are other cases where self-similarities occur only in parts of a shape. *Global symmetries* can therefore be considered as a subset of the *partial symmetries*. One of the first works addressing the subject of detecting partial symmetries is the research by Zabrodsky et al. [ZPA95; ZW97], which introduces a measure of approximate symmetry. The measure is defined as *"minimum effort required to transform a given object into a*

symmetric one". Gal and CohenOr [GC06] presented a method that locates regions in surfaces described by triangular meshes, which are not similar at the numerical and topological levels but still represent approximately similar regions. For this purpose, local surface descriptors that efficiently represent the geometry of local regions of the surface are introduced. Podolak at el. [Pod+06] introduce a method that defines the *planar reflective symmetry transform* (PRST), which introduces a continuous measure of the reflectional symmetry of a 3D shape. It measures the symmetry concerning all planes of an object using its bounding volume. Another method to identify partial and approximate symmetries in 3D models was presented by Mitra et al. [MGP06]. This method solves this problem by pairing sample points on the surface mesh that share the same local shape signature (see Figure 2.1). The symmetries in the input model are indicated by clusters of these matches in a proposed transformation space.

Figure 2.1: Detection of symmetry using the method of (Mitra et al. [MGP06], Figure 1). Left shows the original model; middle, the detected partial and approximate symmetries, and the right the color-code deviations from perfect symmetry

A method matching locally coherent constellations of feature lines on the surfaces of a 3D object to calculats rigid symmetries and detect structural redundancy in geometric data sets was presented by Bokeloh et al. [Bok+09]. This method, however, is non-canonical because the used simultaneous region growing, which stops when symmetric areas collide, depends on arbitrary choices such as the seed points used for the region growing. In a later work, Bokeloh et al. [BWS10] present a similar algorithm that does not have this problem but leads to overlapping areas. Lipman et al. [Lip+10] present a method that is robust to noise and can find symmetries for non-rigid deformations without having any knowledge of the symmetry group beforehand.

Regular Structures. While symmetries deal with self-similarities in an object or parts of it, the field of regular structures builds on these findings and addresses whether regularities can be identified in the alignment of several self-similar parts of an object or several symmetrical objects. The recognition of patterns is an omnipresent topic in research since this allows further determination of relationships within an object or between objects. Besides symmetries, these regular structures are also used by our visual perception to deduce relationships.

By utilizing peaks in the autocorrelation function of images, Liu et al. [LCT04] determine the periodicity of texture patterns. Müller et al.[Mül+07b] present an image-based

method for detecting patterns in building facades. They use a hierarchical top-down subdivision that utilizes *mutual information* to detect similar floors and tiles in the image. Structure recognition also plays an essential role in the field of information compression. Shikhare et al. [SBM01] developed a method for geometry compression of large CAD models. It is a multistage process that first detects repetitions at the connected component level, and in the next step at the subcomponent level across components. It is also able to detect these repetitions at the overall component level.

Pauly et al. [Pau+08] presented a method for identifying repetitive structures in 3D geometries based on an algorithmic approach. In the first step, the input shape is decomposed into small local surface patches, and the similarity transformations between these patches are determined. The finding of the repetitive structures is then made possible by a pairwise similarity transformation and is done by a global optimization procedure in the transformation space. Figure 2.2 shows a set of objects and the regular structures detected by their method.

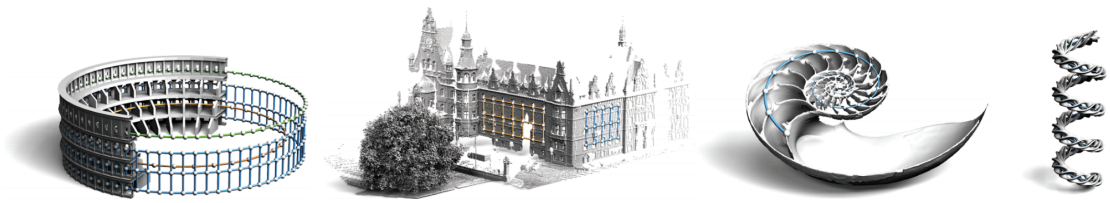


Figure 2.2: Results of regular structure detection based on the method presented by Pauly et al. [Pau+08](Figure 1). The regularities found are visualized by the grid structure.

2.2 Geometry-based Abstraction

A large number of works perform abstraction based on geometric properties. Attene et al. [AFS06a] approximate a 3D model with a set of simple primitives. This approximation is done by hierarchical face clustering of the input mesh followed by an automatic fitting of optimal shapes to the clusters. Mehra et al. [Meh+09] abstract three-dimensional shapes with a set of characteristic 3D curves and contours. Their method processes the input model in two stages: first, a closed envelope surface is generated as an approximation; then a hierarchical curve network is extracted. The network is used to reconstruct an abstract version of the input model, where fine details on the surface are smoothed out. In contrast to our approach, their method works on a fixed global scale and does not allow for local adjustments. Each part of the input model is abstracted equally and thus, visually important structures of the model might be lost.

Calderon and Boubekeur [CB17] recently propose a method to automatically generate bounding shape approximations of arbitrary complex meshes based on an asymmetric

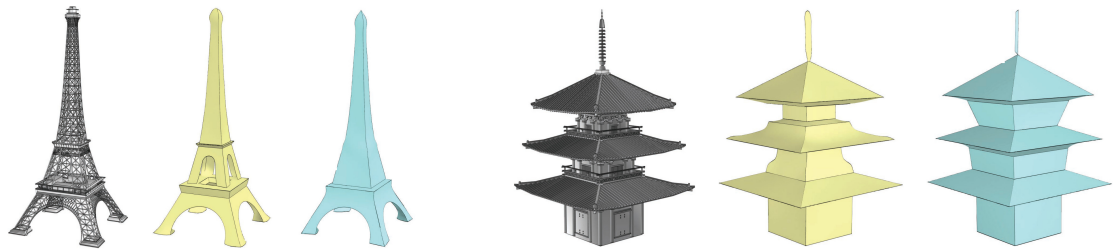


Figure 2.3: Abstractions of man-made shapes (Mehra et al. [[Meh+09]], Figure 13). The abstract representation is computed by representing the input model as a curve network. Each input model is given on the left, followed by two abstractions with different resolutions (yellow and blue).

morphological closing. The method produces shape proxies that are close to the input model, even for the coarsest level of approximation. The user can locally influence the proxy scale, resolution, and topology in an intuitive way by applying a brush tool. In contrast to our approach, the method does not account for perceptually important structures.

McCrae et al. [MSM11] present a learning algorithm to generate abstractions of shapes based on planar sections. This method focuses on a set of slices that describe the input model in an abstract form. Yumer et al. [YK12] assume that there is no single abstraction for one object and present a co-abstraction method that generates identity-preserving, mutually consistent abstractions for shape collections. The models in the resulting shape collection are abstracted to the maximum extent while maintaining their distinguishing characteristics.

Kada [Kad06] introduces a method for automatic generalization of 3D building models by remodeling the input shape based on half spaces. Forenberg [For07] uses scale-space theory: faces of the input model are moved against each other until 3D features of a certain scale are removed. Grabler et al. [Gra+08] extend this to automatically simplify the visual appearance of building models in tourist maps. The authors rectify the input model using a grid structure, decompose it into different parts, and implement a facet-shifting strategy to build the simplification. Chen et al. [Che+13] train a neural network with cartographers' expertise and knowledge about constructing ground plane simplifications to model a map generalization process. Mitra et al. [Mit+13] provide a comprehensive overview of the field of structure-aware geometry processing.

Gal et al. [Gal+09] introduce iWires, an analyze-and-edit approach, to manipulate an existing 3D model while maintaining its characteristics. Wires are extracted from the input model and enhanced with information about geometric features and relations to other wires. The user can edit these wires while the system maintains wire features and relations. In contrast, our method employs freehand sketches that offer more flexibility compared to predefined sets of handles. Recent efforts concentrate on exploiting symmetries for shape manipulation tasks either through hierarchical grouping [Wan+11] or by

guiding the deformation and fitting of templates [Kur+14]. Fu et al. [Fu+16] perform structure-aware editing of man-made objects through capturing group specific priors. Their system allows the production of shape variations and structure-aware editing in real-time.

Even more recently, Nishida et al. [Nis+16b] leverage the effectiveness of user-defined sketches to guide the automated assembly of snippets of procedural grammars as building blocks to turn sketches into realistic 3D models. Unlike the previous approaches, our method aims at leveraging Gestalt rules as fundamental means for perceptual reasoning for shape abstraction and simplification.

2.3 Perception

Geometric-based approaches for simplifying and abstracting shapes work purely on geometric properties and do not consider human perception. However, not taking the perception into account can result in important information being unintentionally changed or removed. The researchers, therefore, investigated how human perception works and explored methods that account for it in the simplification process. The research on the human perception of 2D geometric objects stems from various branches of science, e.g., geoinformatics and photogrammetry, geography, cartography, or computer graphics. In the following, we give an overview of related work targeting the field of human perception. We show works that incorporate the results of perceptual studies in the simplification process and further works giving insights into human perception, focusing on geometry and building structures.

2.3.1 Gestalt-Based Abstraction

While the above-mentioned works perform simplification and abstraction mostly based on geometrical aspects, Nan et al. [Nan+11] use high-level Gestalt laws (see also Wertheimer [Wer23; Wer38]) to automatically simplify line drawings of architectural buildings (Fig. 2.4). They describe their scenes by a proximity graph connecting elements to their neighbors. Each Gestalt law yields weights for edges, forming a multi-label graph cut problem.

While their system inspired us, our problem domain is in 3D and therefore lacks a straightforward definition of Gestalt rules. Emerging from 2D to 3D exposes a vast number of new challenges, e.g., in resolving occlusion, conflicts in group dominance, and ambiguities in matching user sketches to groups, which have not been addressed before. Furthermore, we aim at an interactive Gestalt-based abstraction rather than a fully automatic process. With a few simple strokes, the user can freely and easily guide geometric abstraction

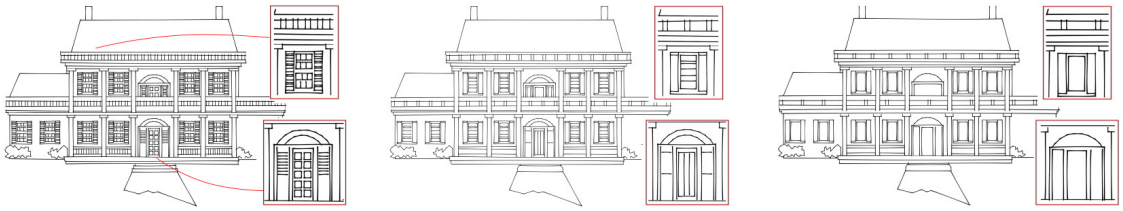


Figure 2.4: A progressive abstraction of a complex facade using on conjoining gestalts. (Nan et al. [Nan+11])

operations that respect both structural and perceptual groupings within the input 3D shape.

Zhang et al. [Zha+13], Wang et al. [Wan+15] and Li et al. [Li+04] exploit Gestalt principles for the grouping, generalization, and abstraction of 2D building footprints for urban abstraction. Such laws are used for assisting human sketching [Lin+13], for the selection of elements [Xu+12], or for simplifying sketches [LWH15] (law of closure). Some authors also highlight the importance of using Gestalt laws for geometric abstraction [Løv+13; Dan+14b; Kra+14].

Most of the existing methods for shape simplification only consider the generalization of line drawings or object compositions in 2D space. In contrast to these techniques, our approach focuses on the abstraction of 3D objects by employing an interactive feedback-loop based on Gestalt principles.

2.3.2 Human Perception of Geometric Building Structures

We are convinced that our transfer of Gestalt rules into 3D space is already a significant contribution in creating a targeted and information-preserving simplification method of 3D objects. However, we also wanted to explore what lies beyond these Gestalt rules and examine the extent to which other properties of objects and our knowledge influence which details are essential for classifying an object. Therefore, we have studied further works that use Gestalt rules, their results, and the limitations of their method.

Michaelsen et al. [Mic+12] refer to Gestalt-based groupings for the detection of 2D window structures in terrestrial thermal imagery. Within the wide field of visualization approaches, and Adabala et al. [Ada09] present a perception-based technique for generating abstract 2D renderings of building façades.

Approaches to the human perception of geometric building representations, which are not restricted to 2D structures or 2D visualizations but, instead, are directly located in 3D space are often developed in the context of cartography. In this context, most approaches aim to reduce the visual complexity of urban 3D representations to decrease the user's cognitive effort.

The prominent representatives of this approach are Glander & Döllner [GD09] or Pasewaldt et al. [Pas+14], who use cognitive principles for generating abstract interactive visualizations of virtual 3D city models. Both approaches focus on emphasizing landmarks while buildings that are supposed to be unimportant from a tourist's point of view are grouped and replaced by cell blocks. Instead of using Gestalt rules, this grouping is based on the infrastructure network. Other approaches realize the abstraction of virtual 3D cities by directly analyzing and modifying the geometric properties of the building models. For example, Sun et al. [Sun+11] propose a structure-preserving abstraction method that generates abstracted 3D building models by avoiding concave shapes.

All the works mentioned have one thing in common: They integrate perceptual principles in their methods for the recognition, generalization or abstraction of geometric building structures to reveal or emphasize building-related information. These perception-based methods, however, are all more qualitative than quantitative operations. That means, quantitative statements about the degree to which the respective information can be perceived by a human, or tasks like, for example, searching for the best abstraction to achieve a certain degree of perceptibility are not supported.

2.3.3 Quantifying Human Perception of Geometries

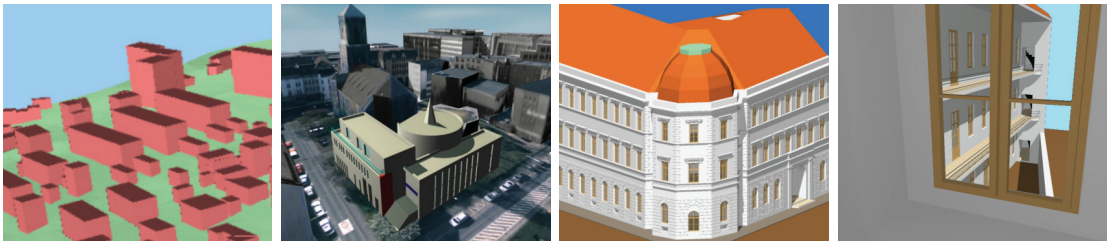
Existing attempts to quantify the human perception of geometric objects are closely linked to Gestalt principles and, therefore, limited to simple 2D structures. Desolneux et al. [DMM04] and Cao et al. [Cao+07] propose a probability measure to quantify the meaningfulness of groupings in cluster analysis for 2D shape recognition. Kubovy & van den Berg [KV08] provide a probabilistic model of Gestalt based groupings by proximity and similarity on regular 2D patterns. Michaelsen & Yashina [MY14] put the Gestalt principles in an algebraic setting to facilitate 2D object recognition in images.

To the best of our knowledge, the evaluation of complex 3D building geometries concerning their perceivable semantic information content, i.e., the quantification of perceptual insight, has not been addressed yet. We will take a first step in this direction based on our user study on the human perception of building categories and our transfer of the Gestalt rules into the 3D space.

2.4 Geometric Representation of Building Structures

To generating simplifications of building structures, it is also essential to know how they are represented in different domains. This information provides insight into why different simplifications have been chosen and why specific weaknesses of such a representation seem acceptable for these use cases.

The variety of geometric representations of urban scenes is wide: Most virtual 3D cities are a collection of 3D buildings given as boundary representations (BReps). Following CityGML, the OGC standard for 3D city models (Kolbeet et al. [KGP05], Gröger, G. & Plümer [GP12]), the geometric level of detail (LoD) of 3D building representations can range from LoD1 and LoD2 (LoD1: box models using flat roofs, LoD2: detailed roof structures, planar façades), which are available for the majority of the buildings of a 3D virtual city – over LoD3 (3D façade structures), which are usually only available for single landmarks and small test scenes – up to LoD4 (indoor models), which are not within the scope of our project. (See Fig. 2.5)



(a) LoD1 (b) LoD2 (c) LoD3 (d) LoD4

Figure 2.5: Different LoD representations as shown by Gröger, G. & Plümer [GP12]: a) box models using flat roofs, b) detailed roof structures and planar façades c) 3D façade structures, and d) indoor models

Due to increasing computing power, nowadays, urban scenes can also be represented based on dense unstructured 3D point clouds or triangle meshes. These models either are the direct output of laser scanning or are pushed by the development of Structure-from-Motion and dense multi-image matching techniques (Hirschmüller [Hir07], Agarwall et al. [Aga+11], Engel et al. [ESC14]), the result of photogrammetric derivation from images (Fritschet al. [Fri+11], Haala [Haa13], Mayer et al. [May+12]). Google Earth, for example, solely uses triangle meshes for its representations. These models thus avoid the derivation of geometrically and possibly also semantically interpreted BReps with a defined LoD. In the case of Google Earth, the primary information is given by the displayed streets and their names and objects of interest; the mesh serves here as additional visual support. Compared to simple satellite imagery, where the elevations are not well observable, such a mesh allows one to understand the area better.

However, BReps are required for all applications that go beyond pure visualizations. The advantage of such a representation is that the LoD level can be chosen based on the viewers' requirements. Generating representations for these LoD levels, however, is a complex task. In many areas where building models with different LoD representations are used for city models, the focus is on top-down views or views from higher elevations. The transitions between these LoD representations can also be very distinctive, resulting in so-called popping artifacts during the transitions, making them not well suited for applications where dynamic transitions between different LoD representations are required. Our research is a first step necessary so that LoD representations can be defined that are

more suitable for this purpose.

2.5 Procedural Modelling

Only recently, researchers started exploring approaches to model virtual environments with realistic traits of real urban landscapes [Sme+14]. In this domain, as in the case of simplification, semantic information is also of significant importance. While the field of abstraction investigates how such information can be preserved, modeling examines how those can be generated. Here, we focus on the problematic aspects of plant and urban modeling, ecosystems, and learning-based methods.

2.5.1 Urban Modeling

Urban structures are often modeled proceduraly [Wat+08]. In their seminal paper, Parish and Müller [PM01] used L-systems to model complex cities, and Wonka et al. [Won+03] applied split grammars to procedurally define buildings that were later extended by using subdivision [Mül+06] and by more advanced operations [SM15]. Purely procedural models of infinite cities were introduced by Merrell and Manocha [MM11; MM08]; the procedural modeling of street layouts has been described by using vector fields [Che+08]. Similarly, procedural approaches have been successfully applied to modeling façades [Mül+07a]. Urban modeling has been combined with urban simulation to generate viable cities [Van+09; Van+10] and city growth [Web+09].

However, most of the related work focuses solely on urban structures and considers vegetation only as an decorative add-on.

2.5.2 Inverse Procedural Modeling

Our approach is related to inverse procedural models, in that it learns plant placement from real cities and attempts to transfer it to synthetic ones by fitting parameters of a procedural model. An inverse procedural model for façades has been introduced by AlHalawani et al. [AlH+13]. Wu et al. [Wu+14] state that variations from a procedurally encoded single layout can be generated by the work of Bao et al. [BSW13]. The layered nature of façades has been used for inverse procedural modeling in [Li+11b; Ilč+15]. Exploiting structural symmetries was done in [Dan+14a]. Interactive alterations of shape grammars were utilized in [Dan+15]. Buildings can be encoded as L-systems by using the inverse procedural approach from [VAB10], modeled by using a procedural connection of structures [BWS10], or through binary integer programs [Kel+17]. Finding of the procedural models' parameters from existing data was investigated by Talton et al. [Tal+11]. They used expressions of L-system strings of modules to fit a generated structure to

an input. Ritchie et al. [Rit+15] attempt to control procedural programs and models using stochastic Monte Carlo methods. Structural patterns can be encoded by using the approach of Yeh et al. [Yeh+13] or encoded as L-systems by the work of Šťava et al. [Sta+10] (See Fig. 2.6).

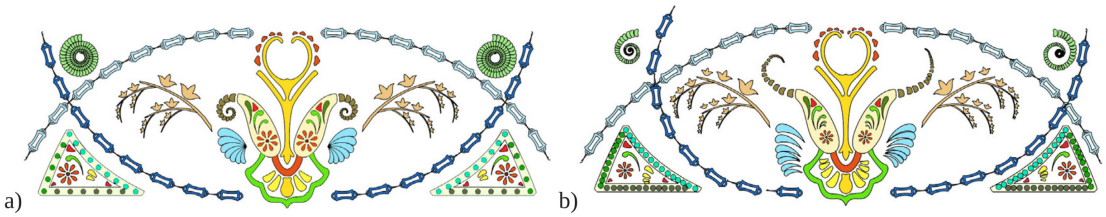


Figure 2.6: Šťava et al. [Sta+10] presents a method that takes as input a) a 2D vector image that is composed of atomic elements and automatically codes it as an L-system, b) which can be edited by the user through manipulation of the parameters of the L-system.

Recently, trained deep neural networks have been combined with inverse procedural modeling to allow for the interactive design of buildings by using sketches [Nis+16a], to find urban models from real world images [ZWF18], and for large-scale reconstruction [Kel+17]. Inverse procedural modeling has also been used to generate entire urban layouts in [Van+12; MV13]. Inverse procedural models primarily deal with regular structures (facades, buildings, cities), and only a few focus on stochastic problems (trees, distributions).

Our approach is inspired by previous works in that it attempts to define an inverse procedural model (urban forest) and finds its parameters. This, in effect, is used to augment an input urban model with vegetation.

2.5.3 Plant Modeling

Research has long focused on defining plausible branching structures based on fractals [AK84; Opp86] or L-Systems [Lin68; Pru86]. Other methods focus on rule-based modeling [LD99], inverse procedural modeling of trees [Sta+10; Sta+14], and finding L-system for branching structures [Guo+20]. Moreover, sketch-based modeling techniques allow artists to produce plant models interactively and in more nuanced ways [OOI07; Wit+09; IOI06]. Alternative approaches attempt to reconstruct plant models automatically either from images [Tan+07; Tan+08], videos [Li+11a], or scanned 3D point clouds [Xie+16; Liv+11]. Only just recently, several approaches also focus on the dynamic and realistic behavior of plant models, including growth [Pir+12a; Lon+12], the interaction with wind or fire [Pir+14; Pir+17], or as established through realistic materials [WZB17; ZB13].

Modeling the plants' response to its environment is of utmost importance to obtain realistic branching structures when positioned in groups or alongside obstacles [MP96]. Approaches exist to model this phenomenon by considering the self-organization of plants

[RLP07; Pal+09], through explicitly modeling the plasticity of branches [Pir+12b] or through the dynamic adaptation to support structures, as can be observed for climbing plants [BM02; Häd+17]. The growth, decay, and pruning of buds and branches play an essential role in plant development [Ref+88]; a phenomenon often parameterized in procedural models to develop convincing branching structures [Sta+14] (See Fig.2.7).

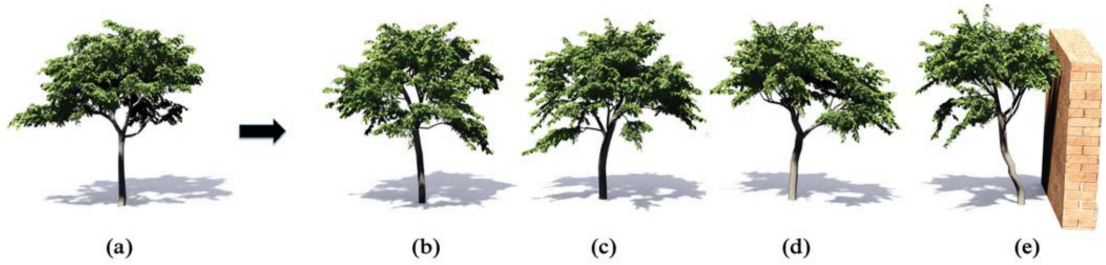


Figure 2.7: Šťava et al. [Sta+14] presented a method that takes a polygonal tree model as input (a), which is then processed by their inverse procedural modeling system to estimate the input parameters of the developmental model so that stochastically similar trees can be produced (b–d), their developmental model is also capable of producing environmentally sensitive trees models (e).

In our approach, once the location of the tree has been established, we grow the trees in the given location and adapt their shape by laws of competition for resources.

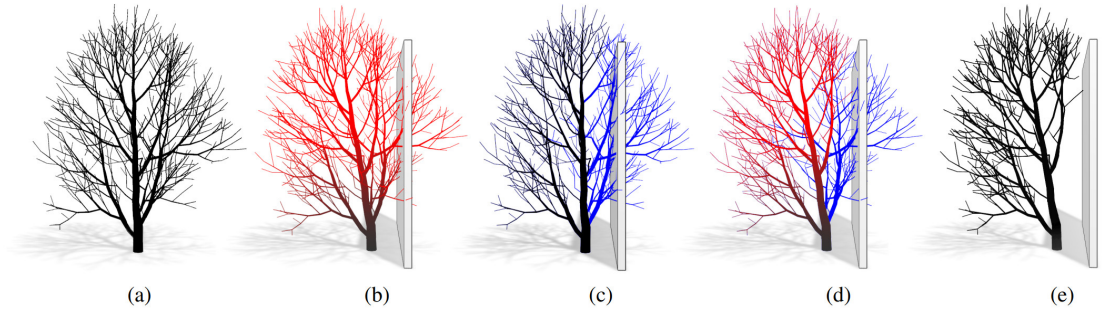


Figure 2.8: A tree model grown in open conditions (a) is transformed by the effect of the shadow cast by a wall. The color represents the difference between the input tree (a) and the transformed versions (e). Red expresses the amount of bending (b), and pruned branches colored blue (c), both transforms (d). (Pirk et al. [Pir+12b])

2.5.4 Ecosystems

Various works focus on ecosystem simulation. The seminal paper of Deussen et al. [Deu+98] introduced a competition for resources on the plant level, and this approach has been recently extended towards the competition of individual trees in layered ecosystems [Mak+19]. Various techniques attempt to simulate ecosystems considering different phenomena, such as erosion [Cor+17] or even by locally learning plant distributions and using them as interactive brushes [Gai+17; Emi+15]. Closely related to our approach is the work of

Benes et al. [Ben+11] that models urban ecosystems by combining wild ecosystem growth from [Deu+98] with controlled plant management. However, contrary to our work, the initial plant placement is purely ad hoc, and their approach does not allow for procedural plant placement that could be connected with real cities.

Our approach defines the procedural models and learns their distributions to populate an empty urban layout. Moreover, their approach is a simulation that seeds new trees and eliminates others by competition for resources over time. Our approach populates the entire city at once.

2.5.5 Learning-based Approaches

Some works have started to explore the capabilities of learning-based methods for scene generation and object placement. While neural networks have shown paramount performance on image classification, synthesis [WXH17; Kha+20], or inverse texture modeling [HDR19; Gue+20] tasks, properly placing objects into meaningful configurations is still a challenging problem. For arranging scenes, methods need to coherently generate plausible and continuous poses (translation and orientation) of objects and to one another. However, most neural network architectures only allow operating on fix-sized inputs and outputs, which makes placing arbitrary numbers of objects challenging. To this end, Ritchie and Wang [RWL18], as well as Wang et al. [Wan+19], propose methods for scene generation based on convolutional neural networks, while Zeng et al. [ZWF18] learn to reconstruct buildings by learning the parameters of a procedural model.

For outdoor scenes, Guerin et al. [Gué+17] and Kelly et al. [Kel+18] use generative adversarial networks to author textures for terrain and building details.

While these methods are only tangentially related to our work, they demonstrate the capabilities of neural networks for scene generation. Similar to these methods, we combine the advantages of image-based learning techniques with procedural modeling. Particularly, we aim to learn the parameters of procedural models with neural networks that allow us to place plants realistically.

CHAPTER 3

Interactive Shape Abstraction through Perceptual Reasoning

In this chapter we present an interactive method that allows users to easily abstract complex 3D models with only a few strokes. The key idea is to employ well-known Gestalt principles to help generalizing user inputs into a full model abstraction while accounting for form, perceptual patterns, and semantics of the model. Using these principles, we alleviate the user's need to explicitly define shape abstractions. We utilize structural characteristics such as repetitions, regularity and similarity to transform user strokes into full 3D abstractions. As the user sketches over shape elements, we identify Gestalt groups and later abstract them to maintain their structural meaning. Unlike previous approaches, we operate directly on the geometric elements, in a sense applying Gestalt principles in 3D. Respecting the Gestalt rules is essential to retain visual patterns used by our perceptional system to categorize objects. We demonstrate the effectiveness of our approach with a series of experiments, including a variety of complex models and two extensive user studies to evaluate our framework.

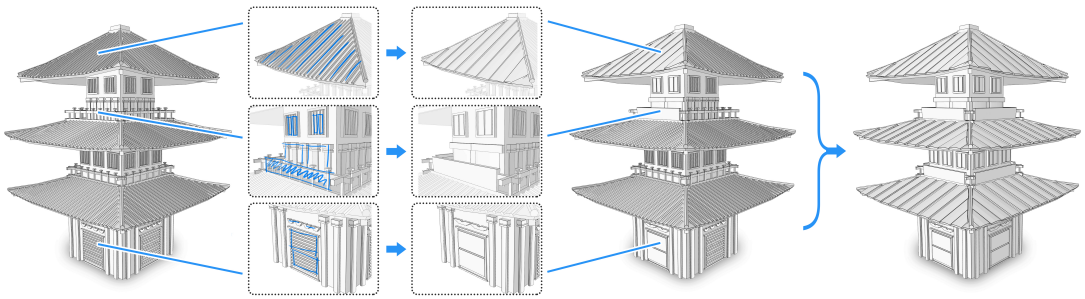


Figure 3.1: User-assisted abstraction of a Japanese house. The user sketches his intention on parts of the object. The system automatically finds Gestalt groups based on the loose scribbles and abstracts these groups accordingly. By automatically propagating abstractions to similar geometric parts, the whole model is abstracted (right).

3.1 Introduction

Reducing and simplifying 3D shapes while keeping their structural essence has been a challenge for artists, architects and cartographers for a long time. This inspired many researchers to develop computational models and representations as powerful means for guiding the observers attention to specific features and for expressing information effectively (e.g., [Meh+09; YK12; Wil11]). Many technical applications such as 3D printing or Level-of-Detail rendering benefit from geometric simplification because they require data often in a specific resolution or complexity. Abstracting a model while maintaining its semantic structure (structure-aware shape processing) is arguably one of the fundamental problems in shape modeling research [Mit+13; Bia+15].

In this chapter, we introduce an interactive method that allows users to easily sketch abstractions of complex models. Shape abstractions are meant to be observed by humans and judging the aesthetic qualities of an abstracted model are virtues that belong to humans or more precisely, to artists. Thus, we have to involve the human in shape abstraction and cannot leave it to a fully automatic process. An abstraction tool must be intuitive and easy to use, while guiding the user into producing coherent geometric models that maintain the perceived structure of the original. Our key idea for realizing this is to employ well-known Gestalt rules, which allow to maintain the form and overall patterns of such simplified shapes as perceived by humans. The challenge here is to combine the users' intent and Gestalt rules together in a computational framework. In a nutshell, a user expresses his intent by sketching over the 3D model. Our system interprets these sketches using their underlying geometric context, thus narrowing down the space of possible abstractions considerably. Then it generalizes the detected Gestalt groups by applying a concise series of 3D abstraction operations (Figure 3.1). Features such as visibility of group elements and whole groups will be used to resolve conflicts between different applicable Gestalt rules and for selecting proper abstraction operations.

Nan et al. [Nan+11] used Gestalt rules for the automatic 2D abstraction of façades. In contrast, we apply such rules on 3D elements while accounting for structural characteristics of the input model as well as for its visual perception. As mentioned above, we do not aim for an automatic abstraction, but to assist the user in his interactive abstraction operations. This defines a novel operational domain we denote as "Gestalt space". It is the abstract space that employs user-defined sketches to simplify 3D shapes while maintaining the constraints defined by Gestalt rules. Metaphorically, the user "sketches in Gestalt space", triggering a series of Gestalt-based operations on 3D objects. Results of our method are abstracted geometries that can be used for a number of applications. Abstraction can be performed in a way that the output can be printed in the given resolution of a 3D printer. The resulting models can also be created in a requested geometric complexity with respect to Gestalt-based perception, while the overall model characteristics are preserved. We demonstrate our technique on models of buildings and technical

artifacts. Our main contributions are:

- The formalization of a new operational domain, denoted as "Gestalt space", that assists users to abstract and simplify complex 3D models while maintaining their structural essence.
- A novel user interface that combines perceptual rules defined by Gestalt principles with 3D sketches that capture the users' intent.
- A framework that computes shape abstractions in real-time and thereby provides immediate feedback to efficiently operate even on complex shapes or on entire scenes.
- Abstraction results of a variety of 3D models and an effectiveness evaluation through two extensive user studies.

3.2 Overview

Our interactive system interprets user sketches and seeks for a sequence of Gestalt-based abstractions that best matches the users' intent. The output is a series of abstractions, each of which is an abstraction of the original model that considers Gestalt principles while preserving characteristic features of the 3D input.

Figure 3.2 presents a high-level overview of our method. We assume that the input 3D model is already segmented into low-level elements [Li+13] (see also Figure 3.14). Our method first analyzes the input model in terms of Gestalt principles, which are regarded as rules for visually grouping low-level elements into larger aggregated structures. Some of these rules can be quantified, namely similarity, proximity, continuity, closure, and regularity. Each rule forms independent (Gestalt) groupings.

The challenge is to resolve possible conflicts among the groupings while identifying the groups that match the users' intent – different Gestalt principles can be applied on the same shape. To resolve such conflicts, we formulate the grouping as an optimization problem (cf. [Nan+11]) and introduce an objective energy function, which encapsulates the characteristics of the Gestalt groups and the users' intent. By minimizing the energy, conflicts are relaxed and unique Gestalt groups are identified.

A fundamental problem of applying Gestalt-rules in 3D is *visibility* (Figure 3.3). The arrangement of 3D objects might form a Gestalt group but some of the objects are occluded by other surrounding objects. To overcome this limitation, we also consider the visibility of objects when resolving group conflicts in 3D by introducing two novel visibility-related measures, denoted as *Group Dominance* and *Element Visibility*.

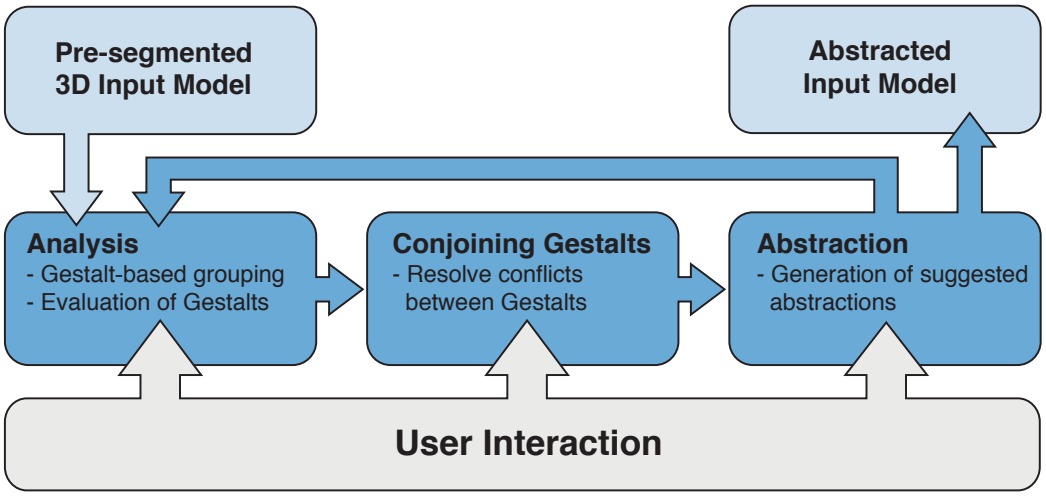


Figure 3.2: System overview: a 3D model is analyzed and potential Gestalt groups are precomputed. Based on user sketches, groups are selected and the model is abstracted accordingly.

Both visibility terms are integrated in the objective function for the optimization to favor visually more dominant groups in the abstraction process. Moreover, we use the visibility terms to determine the degree of abstraction applied to the groups. As we are interested in preserving visually prominent features, visible groups are abstracted more conservatively compared to partially covered or hidden groups.

Group dominance values reflect whether objects are seen as Gestalt groups from multiple views. We perform occlusion analysis and compute the average dominance for each group, which describes its relative importance. Element visibility expresses the extent by which an element is occluded by other elements in its vicinity, computed through ambient occlusion [ZIK98].

Next, we abstract the resulting groups by using one of three possible operations: (i) we create embracing objects in 3D (convex hulls, alpha shapes); (ii) we perform a visual summarization, where a large number of similar objects in a group are represented by a subset of these objects that are potentially scaled; (iii) or we substitute the group by a planar object that shows some engravings of group objects as a form of bas relief (see also Figure 3.8). The visibility terms will alter what form of abstraction is suggested for a group: occluded groups are simplified more significantly (e.g. by embracing objects) compared to fully visible ones.

For the abstraction it is not required to apply these operators explicitly. The user directly sketches over the 2D projection of the model to indicate his intent about the abstraction. The idea is that the resulting model simplification follows the user-defined sketches as closely as possible. For this, the system maps the sketches into the Gestalt space by assigning them to the Gestalt groups. The interpretation may not always have a single solution. To resolve these ambiguities, the system presents the user with a gallery of

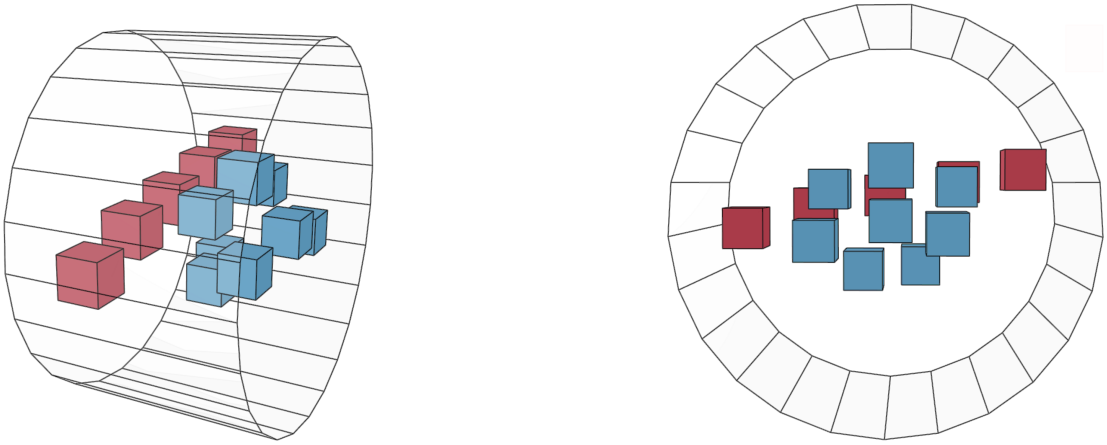


Figure 3.3: Visibility affects Gestalt formation: given two Gestalt groups in 3D (left), one group may be occluded by the other under certain viewpoints and thus will not be visible as a group anymore (right). The surrounding cylinder is only rendered to provide a better spatial orientation.

plausible solutions (Figure 3.9). Once a group is selected by the user, it can automatically be transferred to similar configurations of elements within the model. Here the user sketch only provides minimal guidance for the model abstraction.

3.3 Grouping Principles

Gestalt principles describe how humans tend to perceive arrangements of elements and thereby provide a fundamental means for perceptual reasoning about shape abstraction and modification. Unlike previous methods, our approach aims at providing perceptually plausible abstractions of 3D shapes. Sketching in Gestalt space allows to abstract shapes while maintaining their key visual features; we simplify what is perceptually not important. While psychologists differentiate a large number of such principles, our geometric simplification operations focus on similarity, proximity, regularity, continuity and closure (cf. Nan et al. [Nan+11]). Note, that such principles only describe groupings seen by the viewer; they do not provide us with concrete operations how to simplify them.

Figure 3.4 demonstrates how humans perceive arrangements of shapes by using Gestalt principles of similarity, proximity, regularity, closure and continuity. In particular, shapes of the same form are often perceived as distinct groups (Figure 3.4, a), whereas proximity and regularity also form clusters of shapes (Figures 3.4, b-c). The principle of closure describes the tendency of humans to complete a simple shape that is only shown in parts, while the continuity principle states that we continue the directions of shapes in the most simple way. Since both principles *add* content, we do not apply them for geometric abstraction. We do, however, apply a special variant of the law of closure by replacing a group of elements with a base shape that serves as its simplified representation.

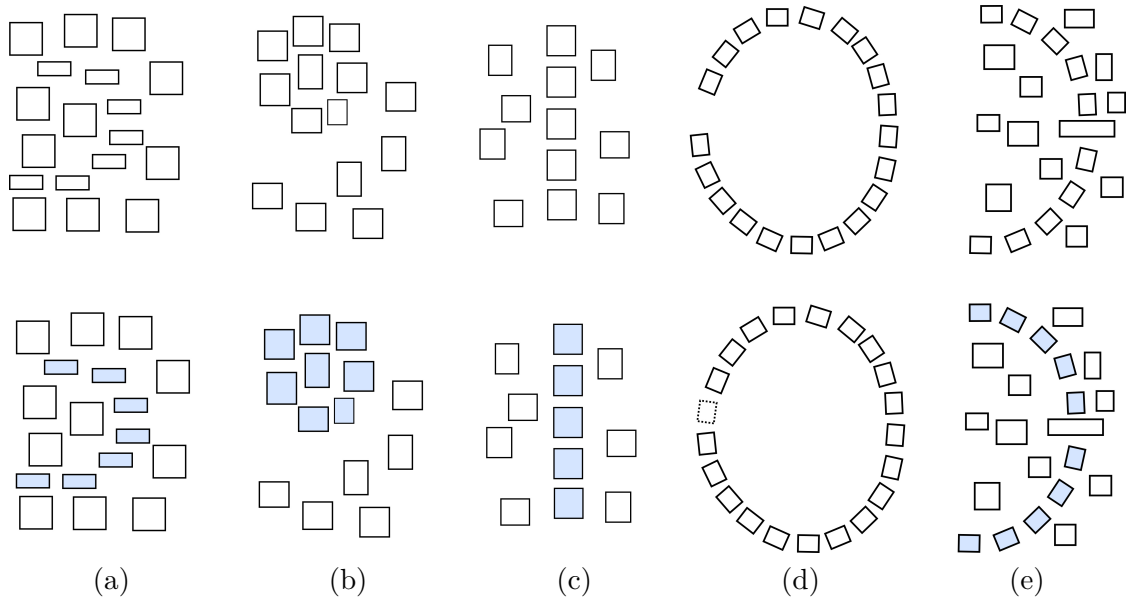


Figure 3.4: Gestalt principles: a) similarity; b) proximity; c) regularity; d) closure; e) continuity.

In many cases more than one Gestalt principle applies to the same set of elements, e.g. one set of elements might form a proximity group and another, overlapping set, might form a repetition group. In this case the geometric configuration determines the predominant principle. To find all potential Gestalt groups, we extend the 2D Gestalt rules to 3D. More specifically, we build a proximity graph G that connects the 3D elements of our scene to their direct neighbors similar to [Nan+11] and then try to find Gestalt groups. For each element p_i, we find its k-closest neighbors, p_j, and connect them with an edge e_{ij}. The edge is associated with a weight $d(p_i, p_j)$ that is related to the Hausdorff-distance $d_H(p_i, p_j)$ between the elements:

$$\begin{aligned} d(p_i, p_j) &= \max\{d_H(p_i, p_j), d_H(p_j, p_i)\}, \\ d_H(p_i, p_j) &= \max_{v_i \in p_i}\{\min_{v_j \in p_j}\{\|v_i - v_j\|_2\}\}, \end{aligned} \tag{3.1}$$

where v_i and v_j are vertices of elements p_i and p_j, respectively. Note that the edges of this graph connect the actual 3D elements of our pre-segmented input model. The closest vertices on those two elements (the ones that define the Hausdorff-distance from Equation 3.1) are used for building the edges. While *Proximity Groups* are detected by finding connected elements in G with distances (edge weights) below a given threshold t_p, we detect *Similarity Groups* by employing a similarity measure that compares the shapes of 3D objects [Bus+05] and try to find groups with a similarity between all their elements that is larger than a given value t_s. We identify *Regularity Groups* in 3D by finding paths in G that have a regular pattern. A path q that represents a regularity Gestalt in 3D is defined by a sequence of edges $(e_0, e_1, \ldots, e_n)$ in G, where the edge lengths vary only to

a small extent:

$$\bigcup\{q_i\} \quad | \quad \frac{1}{n}\sum_{j=0}^{n} |\|e_j\| - \bar{e}| < t_l, \tag{3.2}$$

where q_i are the elements of the regularity group and $\bar{e}$ is the average edge length of the path. Furthermore, the angles α_i and α_{i+1} between two successive edges along the path should only have a small variation as well:

$$\min\left(\frac{\alpha_i}{\alpha_{i+1}}, \frac{\alpha_{i+1}}{\alpha_i}\right) < t_a. \tag{3.3}$$

More specifically, every node of the proximity graph corresponds to a 3D element of the input model, thus, we assign the center defined by all vertices of an element to the corresponding node. Therefore, an edge has an orientation in 3D space and we can compute angles α_i and α_{i+1} between successive edges along the path. The thresholds t_l and t_a are used to control variations of lengths and angles that can occur between two successive edges. In addition of being arranged regularly, all elements within the group have to be similar. We find such paths by picking a node of the graph and check for all incident edges if it is possible to start a path in this direction. We keep track of all paths by labeling the corresponding edges.

To account for rotations within a regularity group we store the main axis of shape elements in the corresponding nodes. To evaluate such axis we apply Principle Component Analysis (PCA) [Jol86] on the vertices of each element. The PCA determines the directions with the highest variances of vertices, which are used as main axis of an element. Given these axes, we can compute the rotation needed to map one element onto the other. For regular structures, the rotations required to map consecutive elements onto each other is static. While this method seems to be sufficient to detect all regular structures in our scenes, more evolved techniques such as Pauly et. al. [Pau+08] may be used to identify more complex patterns.

3.4 Gestalt and Abstraction in 3D

A fundamental difference between abstractions in 2D and 3D is that the viewpoint in 3D plays an eminent role in perceiving shapes and groups. For example, objects can have large distances in 3D, while appearing as a proximity group from a certain viewpoint (Figure 3.5). Similar ambiguous and view-dependent effects can be demonstrated for all Gestalt principles. Some of these effects only occur for specific views, which are called as *accidental views*, as opposed to *generic views* that show objects and groups from a standard viewpoint [Bie87].

Since our goal is to create *object* abstractions, we are not interested in accidental views. Instead, we want to find Gestalt groups in generic views. Therefore, we have to define conditions under which a group of elements appears as Gestalt in such views. In the

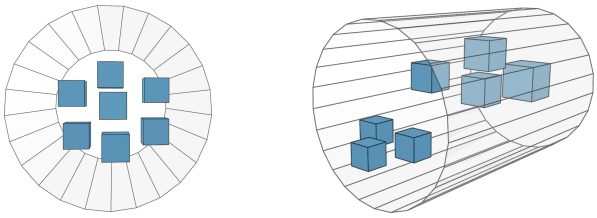


Figure 3.5: Distant objects in 3D are sometimes seen as proximity groups through perspective projection. The surrounding cylinders are only rendered to provide a better spatial orientation.

following we introduce two terms to evaluate the visibility of each Gestalt group: *Group Dominance* and *Element Visibility*. The terms are used for abstraction and reflect the visual importance of groups under generic views. We integrate the visibility properties into the optimization to resolve conflicts as well as to determine the amount of abstraction.

3.4.1 Group Dominance

3D Gestalt groups may be visually more dominant than others based on their location with regard to other groups. A group is considered to be dominant if it is perceived as a Gestalt group from multiple views. In order to evaluate the dominance for a view point we consider the projected area of the entire group and the sum of the projected areas of all individual elements of the group. Our idea is that if all elements of the group are clearly perceivable from the given view point, this also applies to the Gestalt. Further, we consider occlusion caused by other elements which lowers the perception of the Gestalt. We sample the sphere around a group to get a set of view directions $\mathcal{D}$, compute the associated visibility values, and obtain an average dominance value per group. This allows us to quantify what we considered as being *visible under generic views*. Figure 3.6 visualizes the group dominance of two conflicting Gestalt groups.

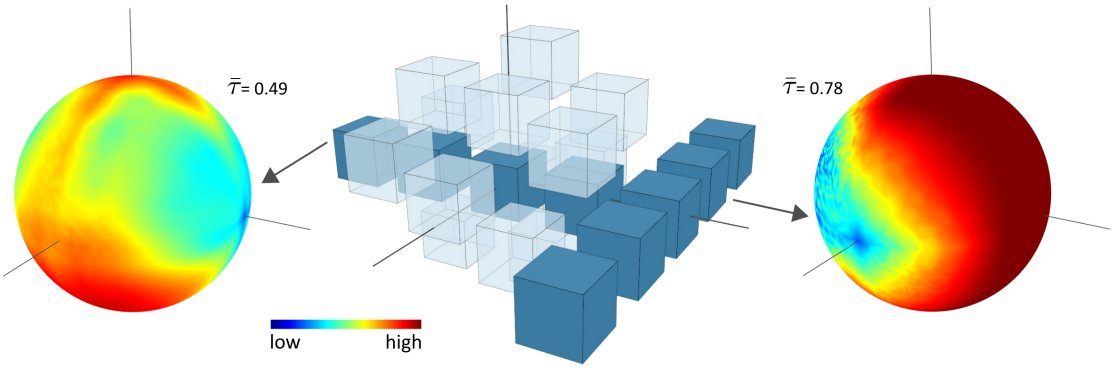


Figure 3.6: Visualization of group dominance: we sample the sphere around a group and compute from which directions the group is visible. This defines group dominance. Here, the visibility of two conflicting regularity groups (colored in dark blue) is blocked by surrounding elements (light blue). The corresponding spheres are shown on each side.

More specifically, we apply Poisson disk sampling directly on the sphere to sample the set of view directions $\mathcal{D}$. In all our experiments we chose a minimal geodesic distance of 0.1 between sample points. For a given view $\vec{v} \in \mathcal{D}$, we first estimate how many elements of the group are visible without considering other objects in the scene. We compute the ratio between the projected area of the entire group K and the sum of all projected areas of individual elements $\{\kappa_i\}$:

$$\rho_{\vec{v}} = \frac{area(K)}{\sum_{i=0}^{n} area(\kappa_i)}, \tag{3.4}$$

where $area()$ is the projected area of the elements and the group without considering any other objects in the scene. A value of $\rho_{\vec{v}} = 1$ indicates no occlusion within the group. If the value is close to zero, elements of the group are highly occluded and therefore the corresponding Gestalt is hardly perceivable. To evaluate the influence of other objects, we determine how much of the group is visible while considering all other elements in the scene. We compute the ratio between the projected area of the group K with and without considering these elements as:

$$\sigma_{\vec{v}} = \frac{area_T(K)}{area(K)}, \tag{3.5}$$

where T is the set of all objects in the scene without elements of K and $area_T(K)$ is the projected area of group K when considering the occlusion caused by T. Similarly, $\sigma_{\vec{v}} = 1$ if there is no occlusion caused by other elements in the scene; a value close to zero indicates occlusion is getting serious. The dominance value for a given view direction $\vec{v}$ is given by: $\tau_{\vec{v}} = \rho_{\vec{v}} * \sigma_{\vec{v}}$. Small values of τ indicate that the group is less visible from the given direction (caused by self-occlusion or through occlusion from other objects). Finally we compute the average dominance value for group K:

$$\bar{\tau}_K = \frac{1}{|\mathcal{D}|} \sum_{\vec{v} \in \mathcal{D}} \tau_{\vec{v}}. \tag{3.6}$$

3.4.2 Element Visibility

In addition to the group dominance we also evaluate the visibility for each of its elements. Even if a group has a large group dominance, some of its elements can be significantly occluded by surrounding objects. This disturbs the visual grouping and prohibits using Gestalt principles for certain views. To capture element interferences, we compute a per element visibility [ZIK98] that influences the data costs for elements in the optimization step:

$$A_{\kappa_i} = \frac{1}{2\pi} \int_{\Omega} V(\vec{\omega}) d\vec{\omega}. \tag{3.7}$$

$V(\vec{\omega})$ is the visibility function that is either 1, if the object is visible along direction $\vec{\omega}$, or 0 otherwise. The result of A_{κ_i} is in the range $[0, 1]$ and describes how much of the

sphere Ω centered at the position of κ_i is covered. We only consider occlusion caused by other objects in the scene, not the occlusion within the group itself.

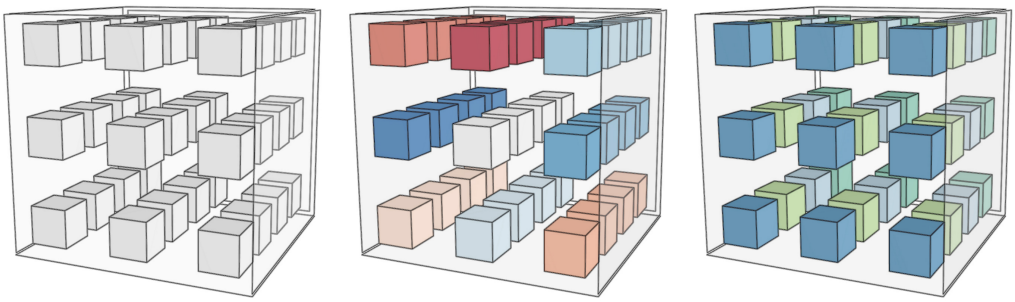
3.4.3 Energy Function

The average dominance value and the element visibility are used to infer the visual importance of 3D Gestalt groups. These measures are important as they specify which parts of a scene can be abstracted while maintaining the main shape characteristics. If a Gestalt group has large dominance and if its elements are not occluded by surrounding elements, abstracting these elements will affect the perception of the overall scene. Thus, for visual abstraction we aim for simplifying visible groups first, since these groups convey most information of the model. To account for these effects in our interactive system, we introduce element visibility A_p and average visual dominance $\bar{\tau}_l$ into the energy function defined in [Nan+11]:

$$\begin{aligned} E(f) = & \sum_{p \in P} (1 - A(p, f_p)) \cdot D(p, f_p) \sum_{p,q \in N} V_{p,q} + \\ & \sum_{l \in L} (1 - \bar{\tau}_l) \cdot h_l \cdot \delta_l(f), \end{aligned} \tag{3.8}$$

where $A(p, f_p)$ is the visibility value and $D(p, f_p)$ the data cost for an element p if the label f_p is assigned to it. $V_{p,q}$ is the smoothness cost for two neighboring elements p and q. The term $h_l \cdot \delta_l(f)$ represents the label cost with L being the entire set of labels. Please note that both visibility terms have to be inverted since the optimization seeks for minimizing the energy function. By weighting the individual costs of elements by their visibility, the data cost term reflects how well elements fit to the assigned Gestalt group. Occluded elements are penalized, whereas visible ones are favored in the optimization. Similarly, the label cost term favors configurations with only a few and cheap labels. By incorporating the dominance value, the label cost of highly occluded groups will be higher, whereas the cost of visible groups will be lower. Thereby, the optimization favors groups that are visually more important. Detailed definitions of the individual terms of Equation 3.8 can be found in the accompanying supplemental material.

Figure 3.7 shows how visibility modifies the optimization result. Around a 3D grid of cubes we placed planes rendered transparently for demonstration purposes. All sides of the grid are covered except the front side. The planes enclose the interior cubes and block their visibility from most view directions. Without considering the visibility, the optimization will combine the groups in the interior due to their proximity (b). If we consider the visibility of objects, the visible exterior is instead selected as a group and will be abstracted first (c). This will subsequently force all other (parallel) groups to be abstracted consistently and thus will propagate through the whole grid.



(a) (b) (c)

Figure 3.7: Effects of dominance and visibility: (a) Simple setup of a 3D grid of cubes covered by surrounding planes (transparent). All sides of the grid are covered except the front side. Results of the graph cut based optimization without considering the visibility (b) and with the modified energy function (c). Elements that belong to the same Gestalt group have the same color.

3.5 User-assisted Abstraction

Using the aforementioned Gestalt-based 3D principles, our goal is to employ user sketches and to derive an appropriate abstraction sequence. The Gestalt groupings during this process define a space, we call it the *Gestalt Space*, as the basis for the abstraction process. In the following we introduce our interactive tools used for abstraction, different operations that we apply to achieve the corresponding abstractions, and how user intent is incorporated into the optimization.

3.5.1 Operations for 3D Abstraction

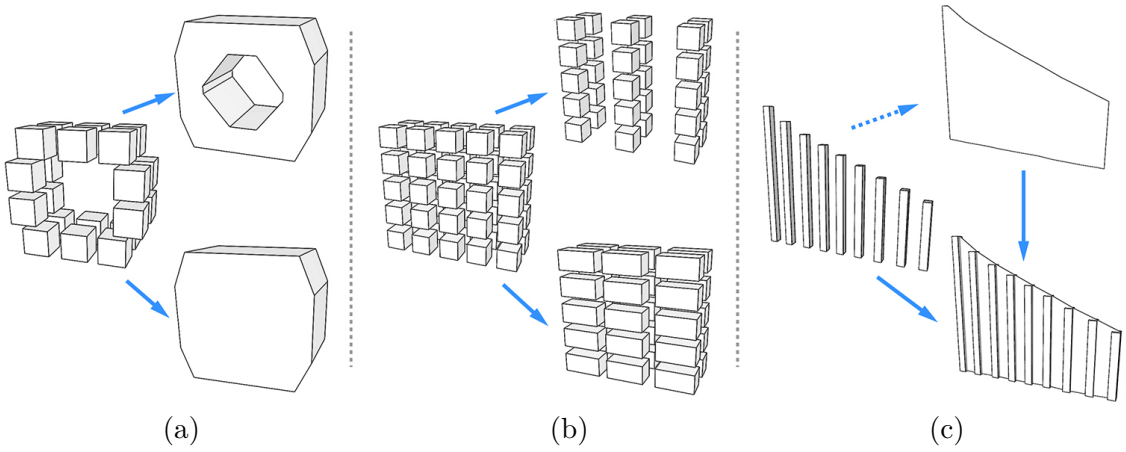
Abstraction is achieved by applying a sequence of Gestalt-based group simplifications. We implemented the following abstraction operations that are either selected automatically by the system based on the geometric configuration of a Gestalt group and its visibility or they are interactively selected based on the user sketch.

Embracing Objects. If objects are close together, i.e., their distance is small in comparison to the overall group extent, we abstract them by creating an embracing object (Figure 3.8, a). This operation can be performed by utilizing an (axis-parallel) bounding box, a convex hull or alpha shape [Akk+95]. It is also possible to use more complex simplification methods such as presented by Mehra et al. [Meh+09].

Visual Summarization. Larger groups of repeating elements (e.g., $n > 20$ by default) are visually summarized by a smaller number of elements that can additionally be scaled to match a given resolution criteria, e.g. for 3D printing (Figure 3.8, b).

Base Shape Substitution. Sometimes groups of repeating small elements cannot be scaled enough to match a given resolution criterion, because it could lead to excessive distortions

of the geometry. In this case we employ a specialized version of the Gestalt principle of closure. If a group is mostly defined along a plane, we determine the hull of the group and replace it with a plane. The original elements are then merged with this plane, similar to a bas-relief (Figure 3.8, c). Thereby, we keep important details of the original model while creating a larger object with less nuanced details (Figure 3.14).



(a) (b) (c)

Figure 3.8: Abstraction operations: (a) embracing object: a set of cubes can either be abstracted by an alpha shape or its convex hull; (b) visual summarization: a set of cubes can be abstracted by reduced number of elements with additional scaling; (c) base shape substitution: a set of long objects which is substituted by a base plane can be abstracted by engraving parts of the original surface into the base plane, which allows to keep the impression of the original geometry.

3.5.2 Interactive Tools

Gestalt Grouping Tools: Initially, the system performs the described Gestalt group optimization (Equation 3.8) on the input model. By hovering the mouse over the model, the user can see the groups and is able to split and join them by a cutting and a lasso tool. This enables the user to use his semantic knowledge of the object to direct the abstraction and to resolve ambiguities during the optimization. Furthermore, this helps to achieve user preferred styles, such as a preference for vertical elements.

We implemented a sketch-based interface that shows the input model and provides different interaction possibilities. The user can adjust the view to find the most appropriate view point and can sketch an abstraction. Figure 3.9 shows a screenshot of the user interface. Based on the sketch and the current viewpoint, the system determines which groups of the 3D model are affected and generates the corresponding abstraction. Often more than one possible solution exists. In this case, the proximity graph is copied and the simplification is applied to each configuration. All results are presented to the user, who can select the most favorable outcome to progress with the abstraction.

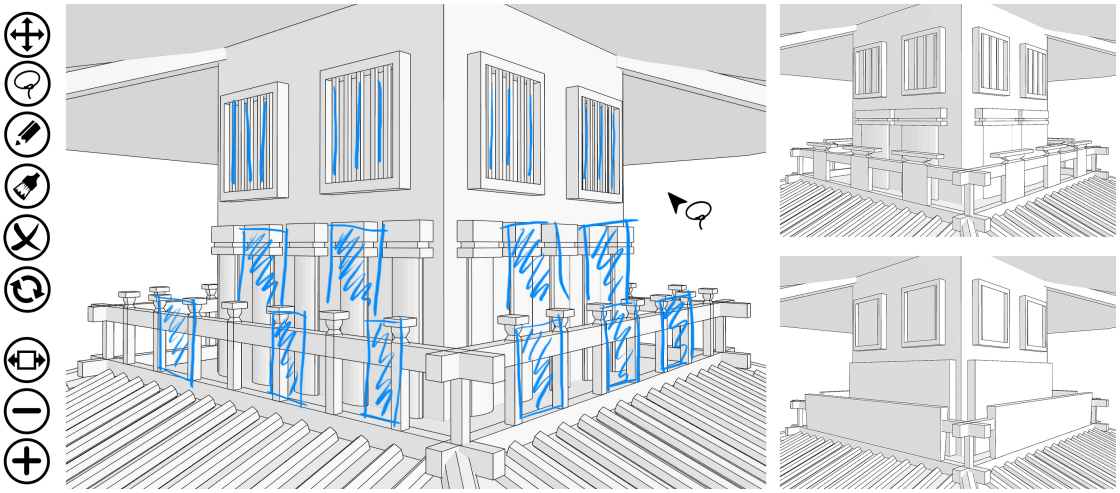
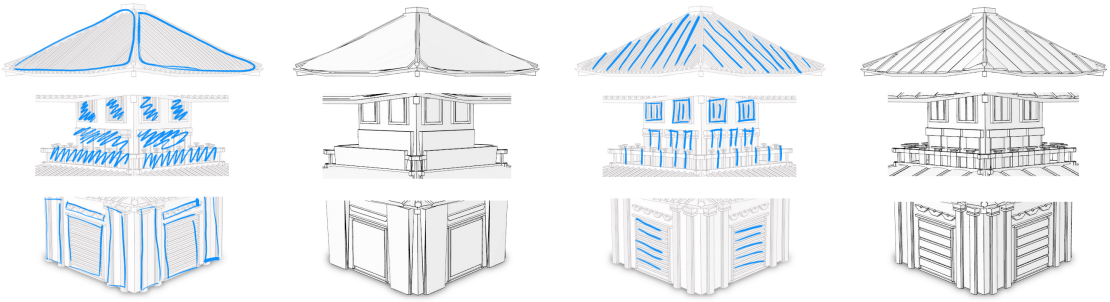


Figure 3.9: Our system interface. The user sketches an abstraction over the projected view of the input model. Different interaction modes such as selection or sketching can be used, abstraction results can be refined (scaling / changing the number of visual representatives, etc.), see buttons on the left. Two possible abstractions are shown on the right for the user to select.



(a) (b) (c) (d)

Figure 3.10: Abstraction of the Japanese house with distinct sets of sketches. Using closed sketches or zig-zag lines (a) result in abstractions using embracing objects (b). Single strokes (c) instruct the system to use visual summarization (d).

Sketching Tools. Since the abstraction of shapes is a highly subjective task, the user should be able to directly influence the abstraction process. We provide this functionality by allowing the user to express his intent on the model with some simple strokes. In most cases the user sketch consist of many individual strokes. We consider time-stamps and the proximity of such strokes to build stroke sets. Strokes appearing directly after each other are considered to belong together and are summarized to describe a stroke set. These sets are used to infer the type of abstraction applied to an underlying Gestalt group. Our system is able to interpret different types of sketches. By drawing some space-filling strokes, such as zig-zag or enclosing lines, embracing objects are used for abstraction (see Figure 3.10 a, b). Visual summarization is applied if the user draws individual lines over some regular structure (see Figure 3.10, c, d). The number of remaining exemplars of the regular structure then corresponds to the number of stroke sets. To indicate the desired scaling and spacing of the remaining objects, the user can sketch the shape of

elements, instead of drawing single lines. By doing this, the system scales the remaining elements accordingly. In case of combining space-filling sketches and individual stroke sets to describe the elements of an regular structure, the base shape substitution operator is applied for abstraction.

Another tool allows the user to transfer the abstraction to other, but similar parts of the model. This is done similar to Xing et al. [XCW14] and helps to abstract models more efficiently. We find such structures by employing a graph-isomorphism algorithm [Cor+01], which is extended by also employing 3D orientations. The algorithm finds all isomorphisms in the proximity graph, regardless if their 3D shape is similar or completely different. Therefore, we have to additionally check geometric correspondence within each isomorphism by measuring element-wise geometric distances. After the abstraction of a Gestalt group is done, the user activates the group transfer and the system detects all similar groups, which can then be abstracted similarly and automatically in an efficient way.

Automatic Abstraction. Additionally, the user is able to define an entire area to be abstracted automatically. This might involve a number of consecutive abstraction steps and is initiated by a special lasso tool. Please note, that the conflict between Gestalt groups are automatically resolved based on our visibility based optimization, thus, visual important groups are selected for abstraction. Based on the specified 2D area and the geometric configuration of the groups, a sequence of abstraction operations (embracing objects, visual summarization and base shape substitution) are executed and applied to these groups. Here, the abstraction operators are applied based on the type of Gestalt. Proximity and similarity groups will be abstracted by embracing elements, repetition groups will be processed by visual summarization, if elements of a repetition group form a very thin overall structure, we use a bas relief. Here, we also use the visibility to control the number of remaining elements and their scaling. The higher the occlusion of the regularity group the more elements are removed. The scaling of remaining elements is adapted in the way that the size of the abstracted group matches the size of the original group.

3.5.3 Incorporating User Intent

Coverage Analysis. To identify which groups are intended for abstraction we consider the coverage of the user sketch and the projected silhouette of the group. For this we employ the well-known *Precision* and *Recall* analysis. For each group we compute a bounding volume depending on the type of Gestalt. For proximity and similarity groups we use the convex hull of the group elements; for regularity groups an alpha shape is computed. These bounding volumes are then projected onto the 2D canvas and compared against the alpha shape of the 2D sketch. Figure 3.11 shows an example for a proximity group and a user sketch that partially covers this group.

Precision and Recall provide a measure to identify which group is intended for abstraction. A high Precision value indicates that the sketch covers most of the projected area of the group envelope, whereas a high Recall value indicates that the entire sketch falls nearly into the projected area. In order to determine which group is intended for abstraction we compute the F_1 score which combine Precision and Recall into a single value. This score is defined as $F_1 = 2 \cdot (P \cdot R)/(P + R)$. For abstraction we then consider that group with the highest score. Please note that the visibility of the selected group has to be taken into consideration, since the group might be occluded by other groups.

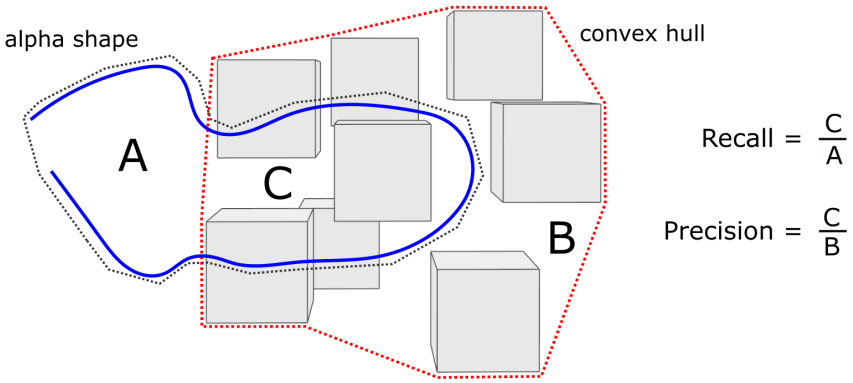


Figure 3.11: Precision and Recall computation. The convex hull of a proximity group (red, B) is projected onto the canvas. Based on the area of the alpha shape of the sketch (blue, A) and the overlap between the areas (C) we compute Precision and Recall.

Besides detecting the intended group for abstraction, we also use coverage analysis to control the visual summarization operator. We compute the number of stroke sets with a projected area that falls completely into the Gestalt group (Recall values close to one). This number is used to infer the number of representative exemplars of the abstracted regularity group. In the next step, we compare the projection of these exemplars with the dimensions of the stroke sets to determine a proper scaling factor for these elements that matches the sketch best, i.e., scales the exemplars to the size of strokes.

Adapting Group Importance. Only groups with Recall values larger than zero are further processed, since for all other groups the projection of the bounding volume is not covered by the user's sketch. However, due to group conflicts, such groups may still 'lose' during the graph-cut optimization and disappear during abstraction. To prevent this from happening, their *importance* value is increased. This is done by temporarily adjusting the data cost term for all elements of the group, in which the user is interested, so that the group will "win" during optimization:

$$D(p, f_p) := w_c \cdot \min_{f_c \in L_c} (D(p, f_c)) \,. \tag{3.9}$$

Here, f_p is the label assigned to element p, which is part of the potential group we are

interested in. L_c is the set of labels (potential groupings) that are in conflict with f_p. With this adaption we ensure that the data costs of all elements of group f_p have at least a value of the same costs for all other conflicting potential groups. Since the optimization seeks for minimizing the energy function, groups indicated by the user will "win" this way. The parameter w_c is in the range $[0, 1]$ and controls the importance. We set $w_c = 0.2$ as default for all our examples.

3.6 Evaluation

We conducted two user studies to evaluate the automatic 3D Gestalt-based grouping and our proposed sketch-based interface. The participants were both undergraduate and graduate students from different universities. The students are normal computer users without backgrounds in computer science. Additionally, we also provide statistics about the performance of our system.

3.6.1 3D Gestalt Grouping

In the first user study, we evaluated the efficiency and accuracy of our method to group 3D elements with respect to Gestalt principles. For this, we asked 15 students to manually define Gestalt groups based on how they perceive groups of elements. In total we showed five input models and their segmentation to the subjects. In Figure 3.14 the segmentation of two models is shown. By clicking on individual segments the subjects were able to manually build groups.

We recorded the time and history of applied operations (adding or removing elements to a group) to compare the efficiency. Table 3.1 summarizes the average timings needed to build one group and the average number of groups that were perceived. Depending on the model complexity, the process of building groups manually takes up to several minutes.

Model (Figure)	#segments	$\bar{t}[s]$	#groups
Japanese House (3.1)	754	22.98	64.80
Building (3.16)	1396	7.91	100.67
City (3.19)	148	17.43	16.13
Bridge (3.16)	517	11.55	44.87
Eiffel Tower (3.17)	424	13.63	30.40

Table 3.1: Overview of the average time $\bar{t}$ needed by the users to define one group and the average number of groups found per model.

To determine the accuracy, we compute the average F_1 measure over all manually generated groups for a given model with respect to the automatically detected ones. More specifically, for each user-defined group A and automatically generated group B, we can

compute the Precision (P) and Recall (R) value pair in a similar way as in Figure 3.11, where C is the set of segments that are shared by both groups. To account for the extent of group elements in the computation, we use the volume of segments $vol()$:

$$P = \frac{\sum_{\kappa_i \in C} vol(\kappa_i)}{\sum_{\kappa_i \in B} vol(\kappa_i)} \quad \text{and} \quad R = \frac{\sum_{\kappa_i \in C} vol(\kappa_i)}{\sum_{\kappa_i \in A} vol(\kappa_i)}. \tag{3.10}$$

Then the F_1 score is defined as $F_1 = 2 \cdot (P \cdot R)/(P + R)$. Finally, we find the best matching automatically generated group with highest F_1 score for each user-defined group and use the average over F_1 values to compute the accuracy.

Figure 3.12 shows the distribution of the average values per model. The user-perceived groups match the automatically generated groups well with high accuracy throughout the models. For the building model (Figure 3.20), we observe a relatively high spread within the average F_1 values. This can be explained by how the students build the Gestalt groups. Most students did not build individual groups for each interior window, but instead used one large group. Our system groups elements within each window, which results in low Precision values, thus lowering the F_1 score.

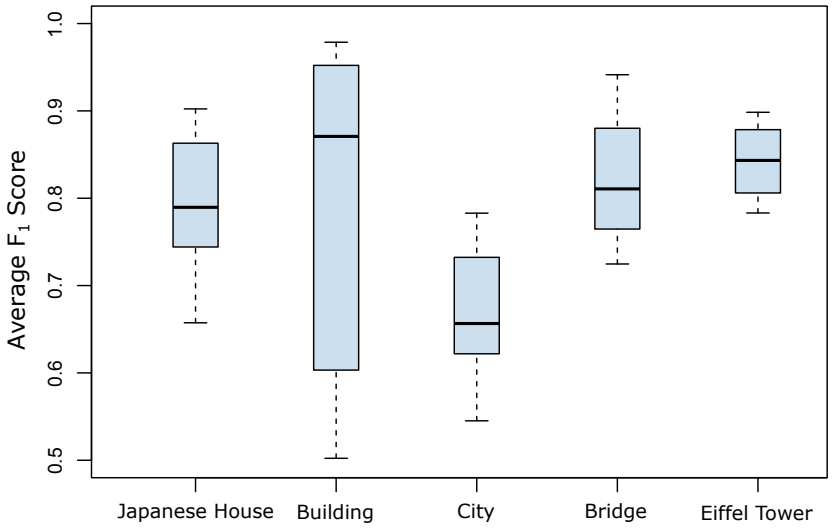
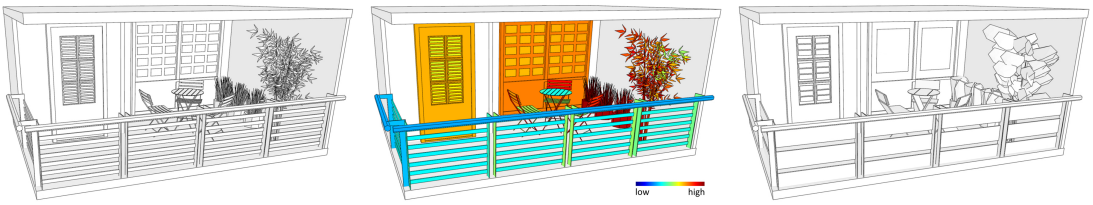


Figure 3.12: Boxplots of the average F_1 values per model.

In conclusion, our system is able to identify groups of elements automatically that are also perceived by most of the users. Since the grouping of elements is a highly subjective task, we measured subtle differences between manually and automatically generated groups (spread in average F_1 scores). Further, our proposed automatic grouping is performed much faster (Table 3.2). By exploiting the Gestalt grouping tools (see Section 3.5.2), the user can adjust the automatically detected groups according to his requirements.



(a) Input model (b) Group dominance visualization (c) Final abstraction

Figure 3.13: Abstraction of a 3D balcony model. The strength of the group simplification is based on the visibility, hence the higher the occlusion is the more significant the abstraction is.

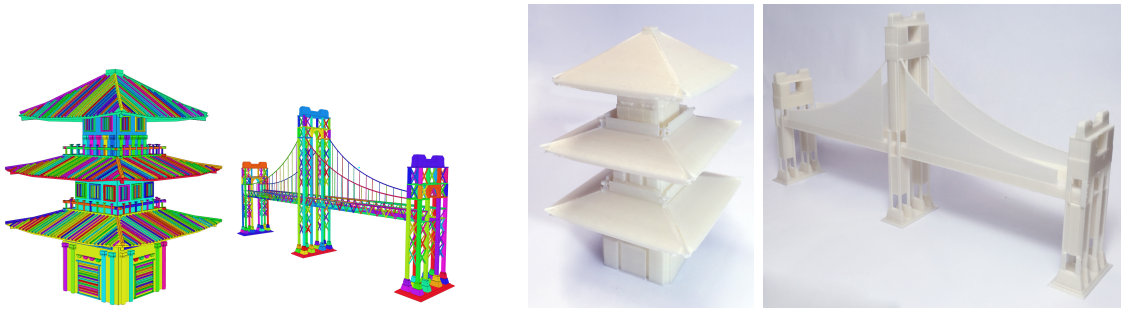


Figure 3.14: Segmentation of our input models for further processing and two abstracted models printed in 3D.

3.6.2 Sketching Interface

We conducted a second user study with 31 students with no experience in modifying or editing 3D shapes to evaluate the effectives of our system. For this task we considered the Japanese house in Figure 3.10 and the bridge model in Figure 3.16. For both models, we presented possible abstractions of different parts. Additionally, we showed multiple sets of sketches. The subjects were asked to rate how well a sketch represents a possible abstraction. The score is given in the range 0 to 10, where 0 means that a sketch does not fit well to the abstraction and a score of 10 indicates a sketch represents the abstraction very well.

It turns out that the interpretation of user sketches by our system fits very well the expectations of the resulting abstraction. With an average score of 8, most subjects expect a bounding volume for abstraction if the sketch has a closed shape, e.g., Figures 3.10 (a-b). If only single strokes were used, the expectations of the produced abstraction are met with an average score of 6.5, e.g., Figures 3.10 (c-d). Only using zig-zag lines without an enclosing shape was misleading (average score of 4.5). Our system indicates such zig-zag lines in the way that the abstraction is generated using embracing objects.

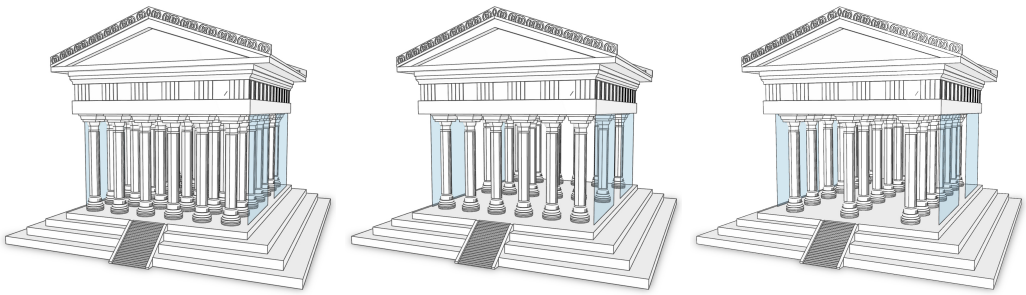
We also invited two people with modeling experience to use our system. The users reported that the interaction concept was considered to be very intuitive and easy to control. For objects with many elements, the selection of groups was identified to be

challenging as the system sometimes does not support to focus on selection areas in the projected view. However, once groups were selected properly, the solutions provided by the system in most cases matched the users expectations. The automatic selection of similar groups within complex objects, such as the façade of Figure 3.16, was considered to be very helpful and efficient.

3.6.3 Visibility

The mentioned visibility terms (group dominance and element visibility) indicate the importance of elements and groups. We use these terms to resolve conflicts if two or more groups act on the same element and to adapt the form of abstraction used to simplify groups. Please note that for visualization clarity, we omit the user sketches in the following examples.

Figure 3.13 shows a model of a balcony and demonstrates the usefulness of the integration of visibility computation into the abstraction process. Highly occluded parts (Figure 3.13 b, dark red) are simplified significantly. The strength of abstraction for the plants on the right side of the balcony is reduced successively due to receding occlusion. Also the abstraction of structures on the windows compared to those on the front door is adapted to visibility. Besides the adaption of the amount of abstraction, we also integrate the visibility terms into the objective function to resolve conflicts between groups in a meaningful way.



(a) (b) (c)

Figure 3.15: Effects of visibility. A model (a) is abstracted using the objective function without (b) and with (c) visibility consideration.

Our goal is to abstract parts of the model that are perceived as visually important. This importance is defined by Gestalt groups and our visibility analysis. Since visible groups communicate most of the information of the model, the optimization will favor those groups within the graph cut. Figure 3.15 shows a temple model and exemplifies how visibility integration affects the abstraction. For demonstrating purposes, some additional walls (rendered transparently) have been added on the left and right side of the columns. The columns of the temple are arranged on a grid, where each column is part of more than one group. The resulting ambiguities are resolved by the graph cut.

Figure 3.15 (b) shows an abstraction result where visibility was not considered. Here, the optimization favors groups that are occluded, which causes an abstraction similar to the original model given in Figure 3.15 (a). In contrast, Figure 3.15 (c) shows the result of an abstraction with our modified objective function. The resulting abstraction reduces the complexity of visible groups. The entire abstraction sequence of the temple model can be seen in the result Section.

3.6.4 Performance

We measured the performance of our system to demonstrate its usefulness for real-time editing. Table 3.2 shows computation times in milliseconds for the proposed method. We have implemented our system in C++ on a desktop computer with an Intel i7 processor at 3.2Ghz and 57GB RAM. Depending on the model complexity and the number of segments and groups, the overall computation time per interaction is between 41ms and 216ms. The table shows the number of groups the model consists of, the initial time to build the graph and groups, and the average time per interaction that is required to update the groups, to find isomorphisms and the total time per interaction including the simplification.

Model (Figure)	#grps	t_{ig}	t_{ug}	t_{iso}	t_{op}	#ops
Jap. House (3.1)	286	538	0.93	23.34	80.72	7
Building (3.16)	1824	2791	1.19	36.24	109.62	4
City (3.19)	113	52	1.53	2.83	4.65	11
Bridge (3.16)	206	695	0.31	45.77	215.83	10
Eiffel Tower (3.16)	320	146	0.03	7.42	41.37	8

Table 3.2: Performance statistics showing the number of groups #grps, the computation time t_{ig} for the initial grouping, the time t_{ug} needed to update the grouping after changes, the time t_{iso} needed to find similar structures, and the computation time t_{op} spent to apply one abstraction operation and the total number #ops of operations. All timings are average values and given in milliseconds.

3.7 Results

To show the usefulness of our system we interactively abstract a number of different models. We show manually simplified models and also demonstrate automatic abstraction sequences.

3.7.1 User-assisted Abstraction

In Figures 3.16 and 3.17 we show a number of user-assisted abstraction operations. In the first row of Figures 3.16, a user directs the abstraction of a building façade: first he selects two window frames and scribbles over the vertically repeating elements to substitute

them by five bounding boxes (a). By marking the right part of the sketch with two blue outlines, the system is enforced to completely replace the content by bounding boxes. The abstracted version is further simplified (b) by labeling the windows as a whole and subsequently by replacing vertical and horizontal repetitions of windows with bounding boxes (c). In the second row a complex bridge model with many structural repetitions is processed (e). In Figure 3.16 (f), the user first replaces the truss network with a small number of larger elements (visual summarization) and then replaces the fine strings that attach the road to the bridge with a solid plane using base shape substitution (g).

The resulting model can be printed with a 3D printer (see also Figure 3.14). Note that 3D printers have physical limitations in their resolution. This requires Level-of-Detail techniques to explicitly address these constraints and the visual affordance of printed models. We do not claim to provide a more efficient means in terms of the material consumption, but instead focus on maintaining important visual clues that define an object while simplifying it.

Figure 3.17 demonstrates that Gestalt-based selection and abstraction goes far beyond conventional processing possibilities. The user draws some steps over a circular staircase to indicate a group and at the same time his abstraction intention. The system finds all similar steps on the stairs and replaces them by a few appropriately scaled steps. In particular, we scale the stairs in such a way that the projected area of the element matches the area of the strokes. Moreover, the bounding volume of the entire model limits the scaling, thus, there is no change in the radius. Similarly, the system abstracts the Eiffel Tower them with just few strokes.

Figure 3.18 demonstrates the effect of dominance and visibility with a small temple (a). As the side view is blocked by the walls, the frontal view is visually more important and thus abstracted first, when the user only scribbles on the frontal part (b). Here, the user also scribbles on the basement of the temple to evoke summarization into two base plates. Note this is not a trivial operation as we have to first compute the Gestalt group and then find a representation of the input base plates by two other plates that are scaled accordingly. The user draws rectangles over the columns to indicate their desired size (d). Furthermore, he also marks columns in rows behind the first visible row. Thereby, he forces the system to abstract all rows in the same way. In Figure 3.19, we demonstrate that the system can also be applied for the abstraction of city models. Here, the user groups and visually summarizes buildings and whole building blocks.

3.7.2 Automatic Abstraction and Level-of-Detail

Besides user-guided abstractions our system also provides an automatic shape abstraction. Here, the user has to define an area of the model which is intended for abstraction. The system then performs a number of consecutive abstraction operations until a user-specific degree of abstraction is established. Thereby, the user forces the system to apply a

number of Gestalt abstractions without directing the graph cut and the minimal solution is selected without any user-defined adaptation of the weights.

In Figure 3.21 we show an automatic abstraction sequence applied on the Japanese house and a building model. For both examples the user selects the entire model for the abstraction. In every step we reduce the number of elements by a fixed percentage. Compared to a user-guided abstraction of the same building model (see Figure 3.20), the automatic process ends up with a different abstraction. Nonetheless, it is still a valid result. Figure 3.23 shows a comparison to the automatic method presented by Mehra et al. [Meh+09]. Without considering Gestalt principles important visual cues would be lost in the abstraction process.

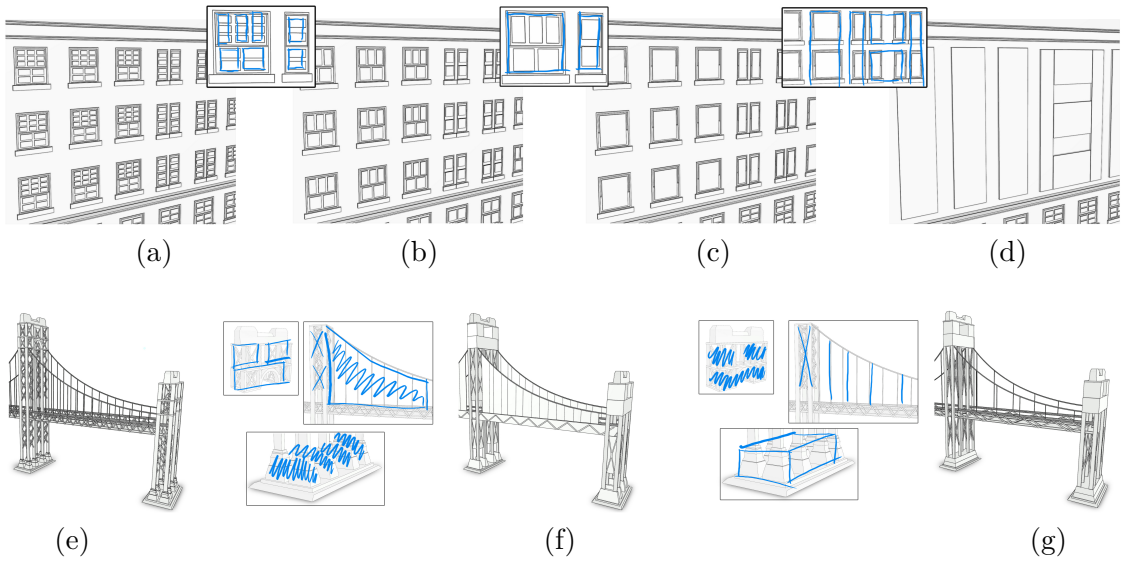
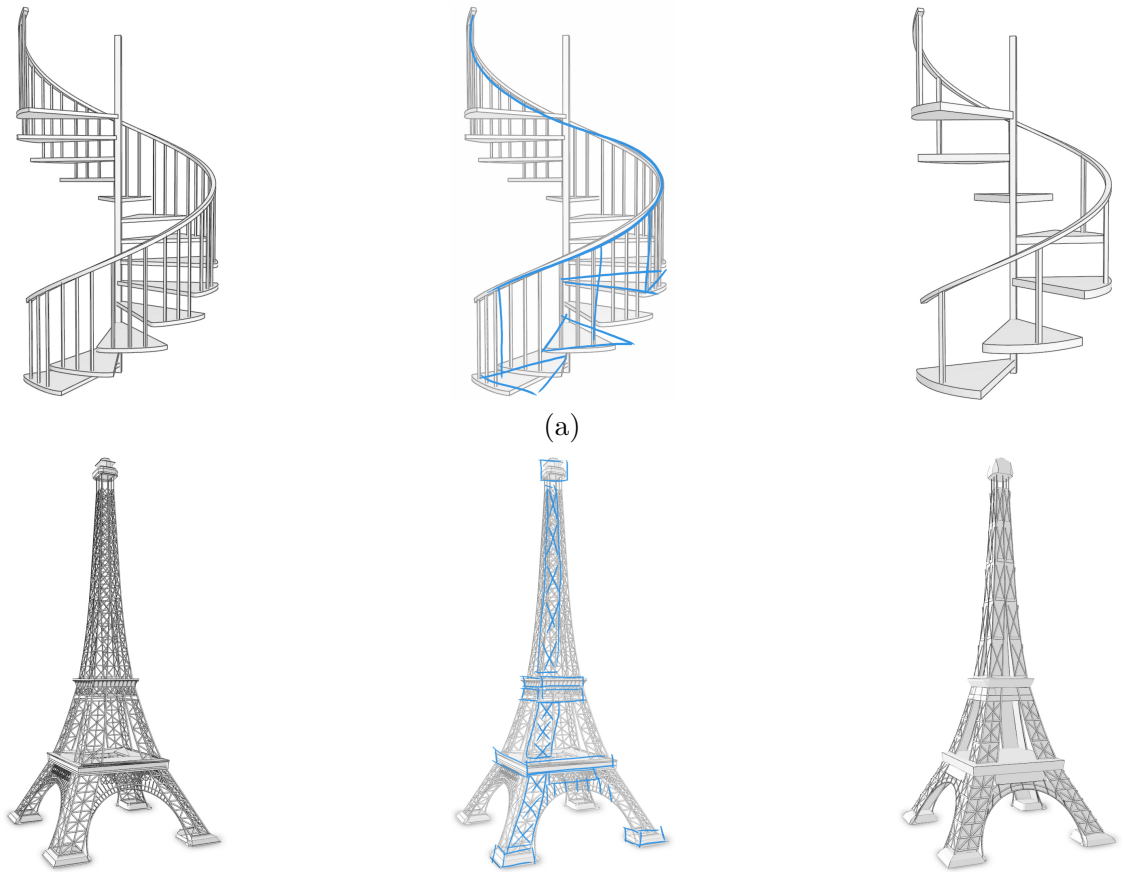


Figure 3.16: User-assisted abstraction (descriptions given in the text).

Furthermore, we asked a professional artist to abstract some of the presented models without seeing the results of our system. The task for him was to abstract and group together the elements of the given shape in a semantically meaningful way. Some steps of the automatic abstraction sequence are very similar compared to how an artist would simplify a model. Figures 3.20 (d-f) show the results, which closely match our abstraction results.

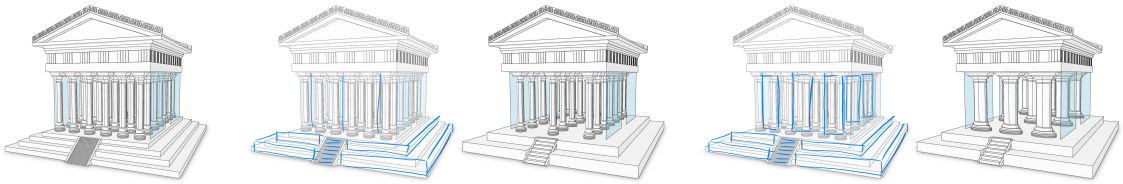
3.7.3 View-dependent Abstraction

Our method can also be applied to abstract models or even entire scenes in a view-dependent way while considering Gestalt principles. To achieve this we compute our visibility terms (group dominance and element visibility) from a given view on the model. This is different compared to previous results, where we evaluated these terms from the viewpoint of an element or a Gestalt group. This allows us to quantify the importance of



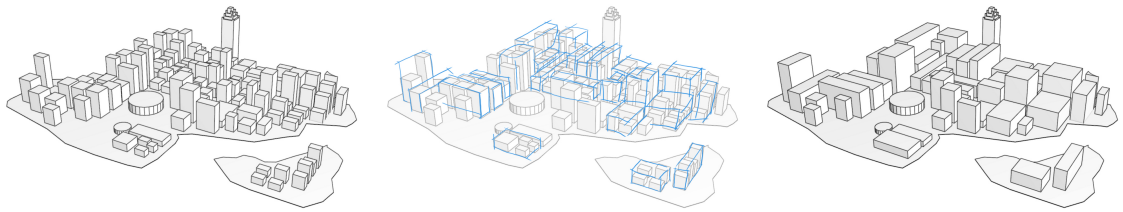
(a) (b)

Figure 3.17: User-assisted abstraction (descriptions given in the text).



(a) (b) (c) (d) (e)

Figure 3.18: User-assisted abstraction when visibility is included (descriptions given in the text).

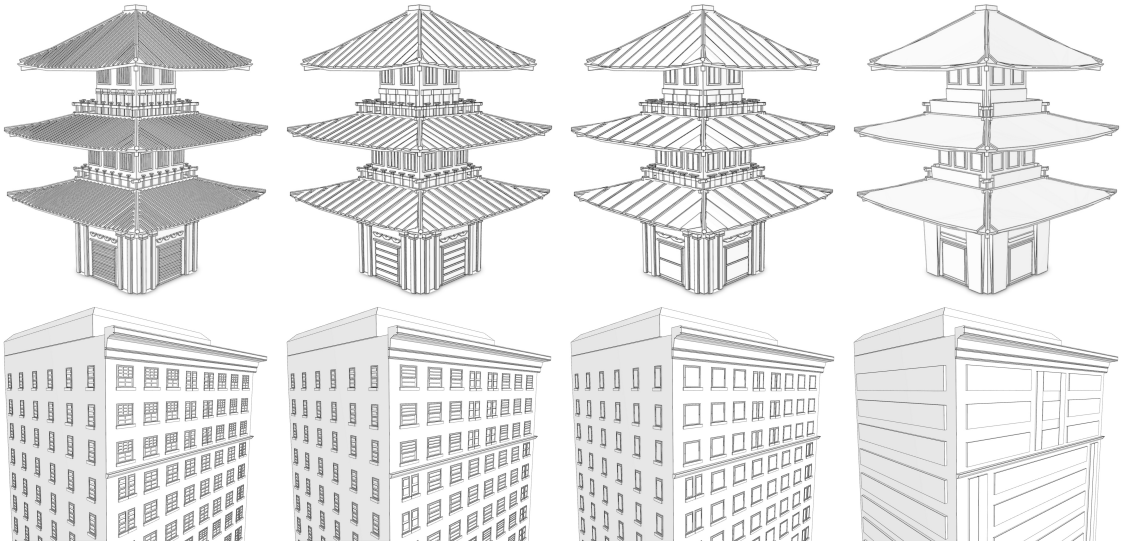


(a) (b) (c)

Figure 3.19: Abstraction of a city model. With a few strokes the user is able to combine building models and replace them by embracing objects or visual summarization. This way even complex city models can be processed very efficiently.

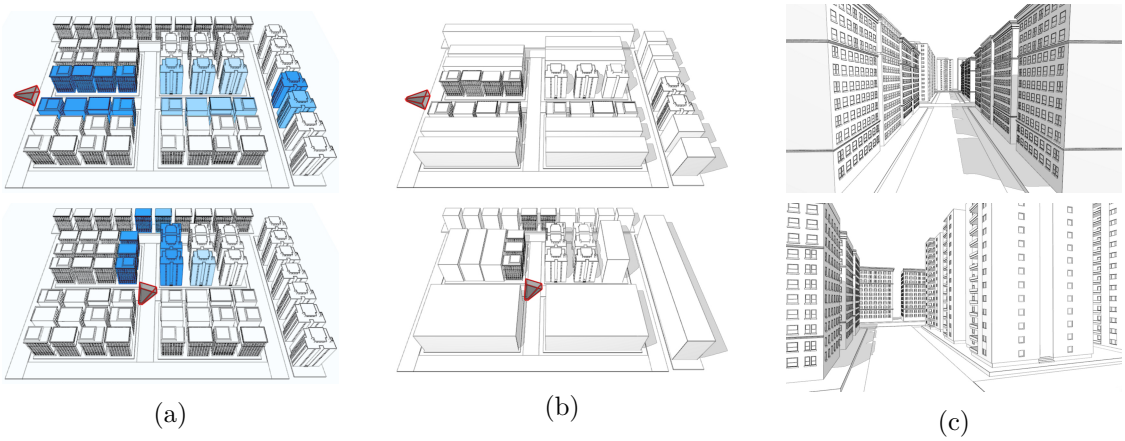
(a) (b) (c) (d) (e) (f)

Figure 3.20: A building model (a) is automatically abstracted using user-defined directions (indicated by blue arrows). First, the user indicates his preference for vertical abstraction in the first abstraction step (result in b), then in the next step for horizontal abstraction (result in c). Building models (d-f) are the abstraction results produced manually by a professional modeling artist, which are similar to our automatically generated abstractions.



(a) (b) (c) (d)

Figure 3.21: Automatic Level-of-Detail sequences. Here, a number of abstraction operations are automatically applied to the input model to match an intended degree of abstraction, i.e., a given number (range) of elements.



(a) (b) (c)

Figure 3.22: View-dependent abstraction of a city model. Each row shows the abstraction for a specific viewpoint on the scene. The camera position and orientation of each viewpoint is indicated by the red camera frustum. Based on each view we compute our visibility terms, which are then used to guide our Gestalt-based optimization and to determine the amount of abstraction. A colored-coded visualization of element visibility is shown in (b). Buildings that are visible are colored in blue. The final view-dependent abstractions shown from above and from the perspective of each camera are illustrated in (b) and (c).

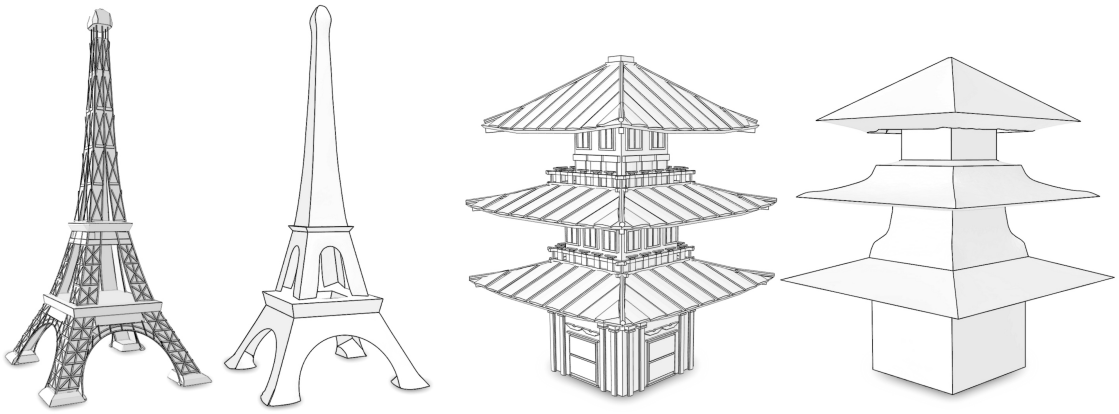


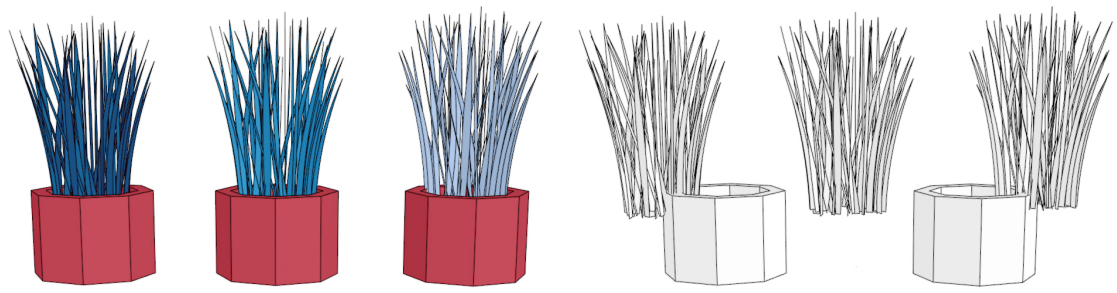
Figure 3.23: Models automatically abstracted with the method presented by Mehra et al. [Meh+09] (right) in comparison to our abstractions (left). While Mehra et al. create very rough approximations, in our case visual important details remain.

elements and groups for a specific view on the model, rather than computing a general importance for *generic views*. Figure 3.22 shows an example of a city scene, where we applied our view-dependent abstraction. The scene consists of regularly arranged buildings, forming different Gestalt groups. Please note that most of these groups conflict with each other. Each row of Figure 3.22 shows an automatic abstraction of the scene from a different viewpoint. Based on each view we evaluate the visibility terms, which are then used within our optimization to resolve conflicts between Gestalt groups and to determine the amount of abstraction. Figures 3.22 (a) and (b) illustrate the color-

coded element visibility and the resulting abstractions shown from above. We use color coding to indicate highly visible buildings in blue and occluded ones in white. The abstractions shown from the perspective of the cameras are given in (c). By comparing the results of the two viewpoints it can be seen that conflicts between groups are resolved differently. The optimization favors regularity groups that are mostly aligned with the viewing direction due to higher visibility. This allows us to keep relevant structures in the scene that are perceived from a given view. We also use the visibility to adjust the amount of abstraction applied to groups. While visible buildings remain unchanged, highly occluded ones are abstracted significantly. Even if some parts of the city are completely occluded, we use embracing objects for abstraction, which allows us to keep plausible shadows in the scene. The accompanying video shows a tracking shot for the city model.

3.7.4 Limitations

Our system has some limitations. For the visual summarization abstraction we first reduce the number of exemplars of a group and scale the remaining elements. The scaling is performed in the direction of the main axis of the element determined by the Principle Component Analysis (PCA). However, if these directions are not aligned with the symmetry axis of the element, the shape might be distorted after scaling. Another limitation regarding the scaling emerges if we want to abstract groups that lie on curved surfaces. Here the maximum scaling factor of the elements is limited by the curvature. If the scaling is chosen to large elements might not be attached to the surface anymore.



(a) (b)

Figure 3.24: Failure case: given three plants (a) with the detected regularity group (red) and proximity groups (blue), simplification (b) of the regularity group does not account for scene composition.

Although our system is able to simplify nested shape elements that are provided by the segmentation, it is not possible to detect those dependencies automatically with the current implementation. This is shown in Figure 3.24. We have three plants and the detected Gestalt groups indicated by different color: one regularity group (red) and three proximity groups (blue). Even though multiple plants would form a regularity group, our system is not able to detect it. Moreover, the connection between two regularity groups

that are close to each other cannot be automatically resolved by our system. This might result in a wrong abstraction (Figure 3.24, b). To overcome these problems, we have to rely on the segmentation.

3.8 Summary

In this chapter we applied Gestalt principles for the abstraction of complex 3D models. Fully automatic or guided by a number of user sketches, Gestalt principles are applied to elements of the input and visual groups are simplified by a number of operations such as embracing with bounding objects, visual summarization, or base shape substitution. We introduced two novel visibility terms to account for the perceptual importance of 3D Gestalt groups. This allows us to resolve conflicts between Gestalt groups in a meaningful way, where visual important groups are favored for abstraction. Moreover, visibility is used to control the amount of abstraction. We abstracted building models, technical artifacts, and a city model. In most cases our system supports the creation of semantically meaningful abstract representations with only a few user interactions that can be compared to what professional artists will do to abstract shapes. We also showed that our method can be applied to abstract larger scenes in a view-dependent way while still accounting for Gestalt principles

In this work we only implemented the most important Gestalt principles, in the future other, more subtle ones, will follow. We also want to further explore the conceptional space between 2D Gestalt principles and 3D modeling, as we only scratched the surface of possibilities for using Gestalt principles as a means for shape abstraction. Adapting 3D model representations to user perception is a challenging problem for future works in geometric abstraction.

CHAPTER 4

Comprehension of Building Categories based on different Representations

The previous chapter presented a method to combine the gestalt principles with a sketch-based interface to simplify complex 3D objects. In this chapter, we will focus on how this simplification affects perception. In particular, how the classification of building categories is affected by those simplifications.

Virtual 3D cities are becoming increasingly important as a means of visually communicating diverse urban-related information. Since humans are the direct recipients of this information transfer, it is vital that the 3D city representations account for humans' spatial cognition. Thus, our long-term goal is to provide a model for the effective perception-aware visual communication of urban- or building-related semantic information via geometric 3D building representations, which induce a maximum degree of perceptual insight in the user's mind. We also want to incorporate these findings into our simplification method to generate different LoD representations automatically without changing the categorization.

The first step towards this goal is to understand better a human's cognitive experience of virtual 3D cities. In this context, the thesis presents a user study on the human ability to perceive building categories, e.g., residential home, office building, building with shops, etc., from geometric 3D building representations. The study reveals various dependencies between geometric properties of the 3D representations and the perceptibility of the building categories. Knowledge about which geometries are relevant, helpful, or obstructive for perceiving a specific building category is derived. The importance and usability of such knowledge are demonstrated based on a perception-guided 3D building abstraction process.

4.1 Introduction

Virtual 3D cities are used in a growing number of applications: They are the basis for decision-makers in areas such as urban planning, policy-making for environmental aspects, or planning for evacuation and emergency response. Moreover, 3D city models have also entered people's everyday life in the meantime via 3D navigation and tourist information systems or computer games and augmented reality applications.

Besides providing *geometric* information on the represented buildings, virtual 3D cities can also serve as a medium to visually communicate urban- or building-related *semantic* information. In this case, the 3D representations should enable the users to fast and intuitively comprehend the respective semantics without wasting mental workload on non-relevant information. The degree of insight that people obtain via the visual communication of semantics strongly depends on what kind of geometric 3D building representations are used. Geometric 3D representations that fit people's visual habits and urban legibility can help achieve a quick and accurate understanding of urban spatial information.

Due to the multitude of different sensors, algorithms, and modeling concepts used for acquiring and processing geodata in urban areas, virtual 3D cities can be based on various data types and ways of modeling, e.g., unstructured 3D point clouds, meshed surfaces, textured or non-textured volumetric 3D models with different levels of detail and abstraction. However, the question *'Which of these geometric 3D representations is, given a context, best suited to enable a maximum understanding of the information that is intended to be transmitted?'* is still an open problem.

Depending on the application and the requirements going along with it, the provision of virtual 3D cities may involve considerable investments with respect to costs, time, and expertise for data acquisition and processing.

Thus, it is highly unsatisfactory that it is not known beforehand whether the desired degree of understanding can be reached by means of the generated virtual 3D building representations or whether a smaller solution would have been sufficient. Questions like these are of special relevance for systems developers that work with 3D virtual cities, e.g. 3D navigation systems, virtual reality applications, computer games etc. The overall goal of this project is to provide a tool that developers of such systems can use to determine which kind of geometric 3D representation will enable the user to gain the required degree of insight: The tool will allow to quantify, predict and enhance the degree of perceptual insight induced by specific 3D building representations in a specific context.

However, the basis for all that – profound knowledge on the human's ability to understand semantics from 3D building structures – is still missing.

This chapter provides an important first step towards the project's overall goal by identifying perceptual aspects that are relevant for the understanding of semantic information inherent in geometric 3D building structures. Generally, it depends on the application as to which specific building-related semantic information needs to be understood by the user.

Semantic issues of interest may be: building category, architectural style, historical relevance, state of preservation, etc. Out of these, we will exemplarily address the semantic issue 'building category', which covers basic semantic information: Being able to quickly understand the category of buildings when moving through virtual 3D cities means support for various applications, e.g. navigation, house hunting, real estate management, spatial marketing, as it will help users to orient themselves and enable intuitive and efficient exploration.

Within this chapter, we will present a user study that we developed and conducted in order to reveal the required knowledge about how a human understands building categories from geometric 3D building representations.

In more detail, we will focus on two questions:

1. Which representation type is for which building category the most suitable?
2. Which geometric building properties and structures are relevant for the perceptibility of a particular building category?

Moreover, we will demonstrate how the derived knowledge about perceptually relevant geometric structures can be applied to improve the interpretability of 3D building abstractions.

4.2 Development and Conduction of the User Study

The overall goal of this user study is to obtain knowledge about the user's comprehension of building categories in virtual 3D cities. In more detail, the study is designed to investigate different aspects of how different types of building representations affect the user's decision of classifying a building into a certain category.

Analyses are expected to provide answers to questions such as *'Which representation type is for which building category the best?'* or *'Which geometric building properties and structures are relevant for the perceptibility of a particular building category?'*. Knowledge like that can be of great benefit when – given a specific application – the task is to provide the best suitable building representations which can be interpreted most intuitively, and, thus, enable the user to achieve a quick and correct understanding of building-related semantic information.

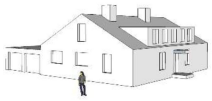
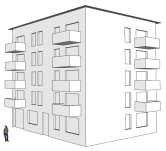
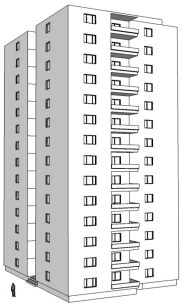
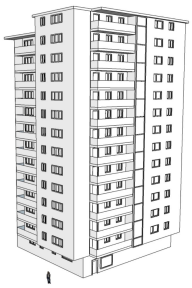
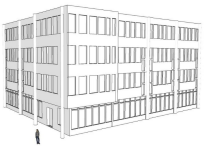
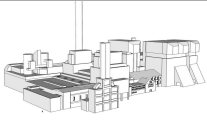
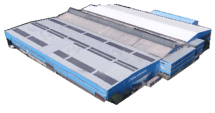
	LoD3 models		Textured Meshes (Google Earth)	Images (Street View)
One-Family Building (OFB)				
Multiy-Family Building (MFB)				
Residential Tower (RT)				
Building With Shops (BWS)				
Office Building (OFF)				
Industrial Facilities (IF)				

Figure 4.1: Examples for building categories and representation types used in the study (Google Earth/Street View, ©2015 Google).

Within a 3D navigation tool for example it is not crucial to provide the highest level of detail, since users should be able to identify essential structures with a glimpse. While Virtual Reality applications, as its name implies, aim at preferably detailed representations.

The data basis and the setup of the user study are described in section 4.2.1; the applied evaluation metrics are presented in section 4.2.2. The results of the study as well as a first application scenario showing how the derived knowledge can be used for perception-aware abstraction processes will be part of section 4.3.

4.2.1 Data Basis and Setup

The category of a building is reflected in both *geometric* building properties, e.g. building size, roof shape, size, number and arrangement of windows etc., and textural information.

In order to separate the influences of both aspects as well as possible, the following representation types are used within the study: (a) untextured LoD3 models for analyzing solely influences of *geometric* building and façade properties, (b) textured meshes/LoD2 models (from Google Earth) as well as images from Google Street View for analyzing influences of textural information.

Each of the three representation types are shown to the user in a way that at least two façades per building are visible. To avoid the assignment of the user being influenced by the building's environment, only the building itself appears; the environment of the building is not represented.Research on the influences of the environment will be part of our future work.

Within the study, users have to classify buildings into six characteristic building categories extracted from the ALKIS feature catalogue (ADV 2015):

- One-Family Building (OFB)
- Multi-Family Building (MFB)
- Residential Tower (RT)
- Building With Shops (optionally with partial residential usage) (BWS)
- Office Building (OFF)
- Industrial Facility (IF)

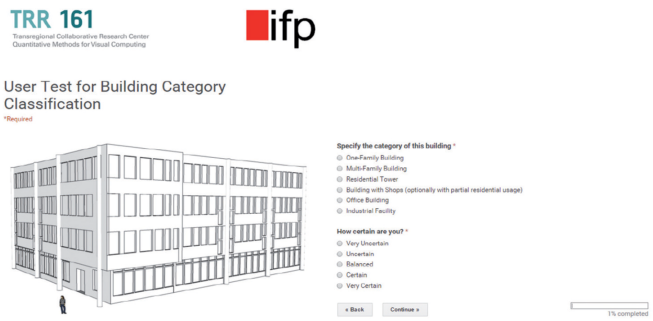


Figure 4.2: Exemplary page of the study with a building model to be classified.

The buildings which are to be classified are randomly taken from German cities (mostly Stuttgart), i.e., between 15 and 20 candidates of each building category are selected. For all these candidates LoD3 models have been modeled manually. For 60% of the buildings, additionally, textured meshes/LoD2 models from Google Earth and/or images from Google Street View are provided. Fig. 4.1 gives examples of the building categories and representation types presented to the user.

The user study is conducted as an online survey for the test person's convenience as well as faster evaluation reasons. At the beginning of the survey, some general information about the user is obtained, namely:

- Gender
- Age
- Graduation
- Subject of study
- Nationality
- Previous experiences in 3D virtual reality worlds (computer games, Google Earth, CAD modeling etc.)

Subsequently, the actual building category classification follows. All in all, 165 different building representations have to be classified by each participant. The representations are shown to the test person in random order. After the classification of each representation, users have to rate their level of certainty (reaching from 'Very Uncertain to 'Very Certain' in 5 selection options). Based on the self-assessment for each classification, a relation between user correctness and certainty can be examined. This metric can give further information about whether the user is aware of being wrong in the current classification.

4.2.2 Evaluation Metrics

The actual reference category for each model is obtained by extracting the type of use from the digital city base map and 3D data from the City Surveying Office of Stuttgart. To compare differences between the user's classification and the actual ground truth, all surveys are evaluated, and typical classification quantities such as confusion matrix, commission/omission errors and user's/producer's accuracy are computed Moreover, in order to obtain deeper knowledge on the user's perception, for each building category, the ground truth buildings are compared to the classified buildings. Aiming at quantifiable results, this comparison is based on computing geometric building properties inherent in LoD3 models.

The following properties are evaluated: building footprint, number of floors, floor height, total building height, number of windows per façade, mean window surface area, window-to-wall-surface ratio, number of entrances, mean entrance surface area, number of balconies, mean balcony surface area, the different appearance of ground-floor compared to remaining floors, relative frequency of different roof types. The window-to-wall-surface ratio is given as the ratio of mean window surface area and the mean façade area (wall surface minus windows, doors etc.).

Considering the property *'different appearance of the ground floor (GF) as compared to the remaining floors'*, 4 different aspects are analyzed: different arrangement, size and shape of windows in GF, as well as different ground plan in GF than in other floors. Each of these four aspects can take either the value 1 (different) or 0 (equal).

Thus, the 4 mean values, which are computed for all representatives of a building category, express the degree of geometric difference between ground-floor and remaining floors. Considering the property *'different roof types'*, we discriminate between five different roof shapes: flat, saddle, hipped, mono-pitch and complex. Correspondingly, a roof complexity value ranging from 1 (simple) to 5 (complex) for each building category is computed as the weighted mean, with the weights being the occurring amount of each roof type within the class.

Based on these metrics, the discrepancy between ground truth and the user's perception is investigated. As the first step within this evaluation, the ground truth data is analyzed. For each building presented to the user in the test, the above-stated features are determined. Since every building has been labeled into one of the 6 building categories presented in section 4.2.1, it is possible to calculate mean values of the features for each building category. These values can be considered representative for the respective category.

In a second step, the six building categories are set up again, however, 'as-perceived' this time. This means that for each category the entirety of all buildings classified into the respective class by all users is registered. Then again the mean values for each feature

are computed, representing the 'as-perceived' or 'as-expected' features for each category. With this procedure, a comparison between the actual properties of a building class and the ones that were expected by the users is possible (see section 4.4.1).

4.3 Results and Application

This section is structured as follows: Overall results of the users' classification will be presented in section 4.3.1. Based on these results, concrete knowledge on the users' perception of building categories is derived in section 4.4. Finally, a first application of the obtained knowledge, namely perception-based abstraction, is presented in section 4.4.

4.3.1 Classification Results of User Study

In total, 96 test persons have participated in the user study. On average, the duration of the study was approximately 50 minutes. The participants' mean age is 24.8 years. The majority of the participants are students from Germany and abroad. In the following, we will first evaluate the classification results based on the entirety of all users (section 4.3.2). Afterward, the results will be evaluated with respect to different groups of users (section 4.3.3).

	OFB GT	MFB GT	RT GT	BWS GT	OFF GT	IF GT	Sum Class
OFB	1483	59	1	69	1	2	1615
MFB	475	2462	137	460	62	8	3604
RT	6	377	2042	95	103	1	2624
BWS	23	222	87	1626	493	57	2508
OFF	18	237	513	265	1983	64	3080
IF	11	3	4	77	142	2172	2409
Sum GT	2016	3360	2784	2592	2784	2304	15840

Table 4.1: Confusion matrix for building classification (see section 4.2.1 for abbreviations of building categories).

	Producer Accuracy (%)	User Accuracy (%)	Commission Error (%)	Omission Error (%)
OFB	73.6	91.8	8.2	26.4
MFB	73.3	68.3	31.7	26.7
RT	73.3	77.8	22.2	26.7
BWS	62.7	64.8	35.2	37.3
OFF	71.2	64.4	35.6	28.8
IF	94.3	90.2	9.8	5.7

Table 4.2: Classification metrics obtained from confusion matrix.

4.3.2 Evaluation based on the entirety of all users

Tab.4.1 depicts the confusion matrix for the building classification. Column headers 'GT' indicate ground truth.

The producer accuracy is given as the ratio of correctly classified buildings with regard to all ground truth buildings in this class. However, user accuracy is more interesting for this work – it is the fraction of correctly classified buildings with respect to all buildings classified to the current class. *Commission errors* correspond to buildings that were classified to a particular class, yet are actually belonging to another. *Omission errors are buildings* that actually belong to the ground truth class but were classified to a different category. The results can be seen in Tab. 4.2.

Obviously, One-Family Buildings and Industrial Facilities could be identified best with both over 90 percent user accuracy. Users have most difficulties with the classes Office Buildings, Building with Shops and Multi-Family Buildings which are indicated by user accuracies between 64.4% and 68.3%. Reasons for that will be further explained in section 4.4.

Besides the classification result, for each building, the users should also rate their certainty for the particular decision. For 22 buildings, the correct classification result was below 50%, with mean correctness of 32.4% for these buildings. However, the mean certainty value for the same buildings is 3.78, which translates to a certainty level of closely to *'Certain'*. This reflects the issue of the users who often not even know their current misinterpretation of the data. Even more: The user might feel certain in his wrong classification. Therefore, it is necessary to use derived knowledge about the difference between perception/expectation and reality to optimize the building representation for the user's needs.

4.3.3 Evaluation based on different groups of users

In the following, we will analyze whether different groups of users come to different classification results. The participants of the study have been quite homogeneous with respect to *age* (90% between 18 and 30 years), *graduation* (over 90% higher education entrance qualification, Bachelor or Master), and *subject of study* (over 95% engineering studies). However, clearly separable user groups of meaningful size can be identified with respect to *gender* (71% male, 29% female), the users' *origin* (38.5% German, 61.5% foreign) as well as the users' *previous experience* with 3D virtual reality worlds (75% experience, 25% no experience). Thus, the user study is additionally evaluated with respect to the latter three properties.

For this purpose, the same accuracy measures as in section 4.3.2 have been determined, this time, however, for the different user groups separately. Significance tests in form of

Student's t-tests are carried out to search for significant differences in the classification results between those user groups.

Evaluation based on gender of users: 71% of the participants were male, 29% female. In order to examine whether there are genderspecific differences in the way of how humans perceive building categories, the results of both groups have been evaluated separately and compared to each other. The analysis shows no significant differences between male and female users.

Evaluation based on origin of users: To investigate influences of the user's origin on the classification results, an evaluation based on the user groups *'German'* and *'foreign'* has been performed. 38.5% of the users in the survey are from Germany, complementary 61.5% of the users have another nationality, distributed all over the world. Since all building models presented in the survey are located in Germany, and architectural construction for equal building types might vary throughout the world, this distinction seems eligible. However, tests on features in each building category did not reveal any significant difference between foreign and German users.

Evaluation based on users' previous experience: Further, the factor of self-assessment with regards to previous experiences in 3D virtual reality worlds is examined. 75% of the test persons stated that they have previous experience in this subject, whereas 25% stated they don't. However, the results for this subject are somewhat ambiguous, since experience in the topic of 3D virtual reality worlds could be interpreted quite widespread. Tests unveiled no significant difference between users with previous experience and novices.

As no significant differences in the classification results of the aforementioned user groups can be identified, all subsequent evaluations and interpretations in section 4.4 will be based on the entirety of all participants.

4.4 Derivation of Knowledge on Building Perception

Based on the findings described in section 4.3.1, we will now go a step further and try to derive coherences between the perceptibility of the building categories and several properties of the 3D representations. To find answers to questions such as 'Which representation type is for which building category the best?' or 'Which geometric building properties and structures are relevant for the perceptibility of a particular building category?', we proceed as follows: In section 4.4.1, we extract geometric dependencies, i.e., dependencies between the perceptibility of a building's category and the building's geometric properties. In section 4.4.2, the perceptibility with respect to different representation types is analyzed.

4.4.1 Perceptually relevant building structures

		Footprint (m 2)	# Floors	Floor Height (m)	Total Height (m)	# Windows Per Façade	Ø Window Surface Area (m 2)	Window/Wall Surface Ratio (%)	# Entrances	Ø Entrance Surface Area (m 2)	# Balconies	Ø Balcony Surface Area (m 2)	Different Window Arrangement in GF	Different Window Sizes in GF	Different Window Shapes in GF	Different Ground Plan in GF	Roof Complexity
Ground Truth	**OFB**	115.30	2.1	3.30	9.58	8.3	1.33	16.1	1.7	4.13	0.2	1.04	0.67	0.27	0.33	0.13	2.7
	MFB	238.41	4.0	2.94	14.68	27.8	1.93	30.0	1.4	2.99	2.2	2.06	0.18	0.18	0.18	0.06	2.7
	RT	573.01	15.5	2.73	43.67	110.8	1.94	29.1	1.2	3.53	9.5	4.24	0.36	0.36	0.36	0.21	1.0
	BWS	697.51	3.7	3.84	17.61	34.9	3.58	52.2	1.6	24.90	1.1	10.08	0.91	0.91	0.82	0.55	2.3
	OFF	868.26	5.9	4.09	24.94	96.8	4.94	127.9	1.4	7.93	0.1	3.42	0.53	0.41	0.41	0.53	1.0
	IF	10812.6	2.5	24.63	59.60	23.0	10.17	10.1	5.3	13.33	0.0	0.00	0.53	0.47	0.47	0.40	1.4
As Classified	**OFB**	148.69	3.1	8.65	18.25	22.2	3.47	25.9	1.6	4.99	0.9	1.80	0.53	0.30	0.30	0.20	2.3
	MFB	252.92	5.8	5.43	22.98	48.7	2.31	32.5	2.2	7.35	2.6	3.82	0.49	0.38	0.40	0.29	1.9
	RT	328.95	6.9	3.55	24.19	61.0	3.84	56.9	1.4	8.39	3.0	4.36	0.46	0.39	0.38	0.28	1.8
	BWS	432.80	5.7	6.78	26.69	56.5	4.52	53.6	2.3	9.86	2.0	3.30	0.54	0.47	0.46	0.36	1.8
	OFF	480.00	6.3	6.61	28.11	59.9	3.84	52.5	2.3	8.97	2.1	3.34	0.49	0.44	0.41	0.32	1.7
	IF	912.6	5.0	11.10	36.78	59.3	6.19	61.7	2.8	13.48	1.5	3.88	0.53	0.47	0.49	0.40	1.4

Table 4.3: Geometric properties of the building categories as given in the ground truth (upper part of the table), and as classified by the users in the study (lower part of the table).

The goal is to derive geometric building properties and structures which are relevant or essential for the perceptibility of a specific building category. Following this goal, we first analyze the geometric properties of the building categories' representatives of our ground truth (see paragraph (a)). Afterward, the same analysis is done for building categories as perceived by the users (see paragraph (b)).

(a) Metrics of building categories reference The geometric building features introduced in section 4.2.2 are evaluated for each ground truth category (Tab. 4.3 (top part)). Based on that, it is tested whether the different building categories significantly differ in their geometric features. For that purpose, multiple significance tests are performed for each building feature's class mean. In the following, some significant characteristics for each building category within the ground truth are listed:

- One-Family Buildings have a significantly smaller footprint than all other categories besides Buildings With Shops. The total building height, the number of floors and the number of windows are smaller than in all other classes.

- For Multi-Family Buildings the total number of floors is significantly higher than for One-Family Buildings and Industrial Facilities, yet lower than for Residential Towers and Office Buildings. Accordingly, the total number of windows is higher than for One-Family Buildings but lower than for Residential Towers and Office Buildings. Multi-Family Buildings only differ in few features from Buildings With

Shops, hence the more important they are. The mean window surface is significantly smaller than for Buildings With Shops. Related thereto, a different arrangement, size, and shape of windows on ground-floor and a different ground-floor itself as compared to the remaining floors is significantly more important for Buildings With Shops than for Multi-Family Buildings.

- The most important feature of Residential Towers is the total number of floors, which is significantly higher than for all other building categories. Apart from Multi-Family Buildings, to which no significant difference is detected, the total amount of balconies is higher than in all other categories
- To distinguish Buildings With Shops from the rest, the most important features are different arrangement, size and shape of windows on ground floor as well as different ground floor itself in comparison to the remaining floors. These properties are significantly higher than in all other categories.
- Two features are salient for Office Buildings: The total amount of windows per façade, and the number of floors is significantly higher than for all other categories (except Residential Towers). Moreover, the mean entrance surface area is significantly higher than for One-Family Buildings, Multi-Family Buildings and Residential Towers. To distinguish Office Buildings from Buildings With Shops, a higher number of windows per façade as well as a higher amount of floors is characteristic. Accordingly, the ground floor and first floor resemble each other more in contrary to Buildings With Shops.
- For Industrial Facilities the footprint is the predominant feature because it is significantly higher than in all other categories. The window-to-wall-surface ratio is lower than for Multi-Family Buildings, Residential Towers, Buildings With Shops and Office Buildings.

(b) Metrics of building categories as perceived/ expected by users As done before for the ground truth data, mean features are computed (Tab. 4.3 (bottom part)), this time based on the total amount of buildings all users classified into the respective class. To compare the ground truth data with the results from all users, a significance test for the differences in all corresponding features is computed – this way discrepancies in the user's perception or expectation and ground truth can be revealed. The most important findings in this evaluation are:

- For One-Family Buildings significant tests revealed, that there is no difference in perception and ground truth.
- For Multi-Family Buildings a different arrangement of windows on the ground floor as well as a different ground floor itself in comparison to the remaining floors of the buildings is expected. Additionally, in the users' perception Multi-Family Buildings have a higher number of floors.

- To classify a building as Residential Tower, for users, the number of floors can be less and the total height lower in comparison to ground truth. However, a single floor height is expected to be higher than for the ground truth.
- Buildings With Shops are considered to have a higher number of floors than in reality.
- For Office Buildings users are expecting a higher number of balconies.
- Industrial Facilities are expected to have more windows per façade and a bigger number of floors, too.

4.4.2 Findings based on building representation type

By separating the evaluation into geometric and textural representation types, their impact onto the classification results can be measured. For 60% of the models at least two different representation types for the same building are available. The mean correctness for untextured LoD3 models is at 69.2%. Whereas a slightly higher correctness could be achieved for the textured meshes/LoD2 models from Google Earth with 75.4%. However, the most accurate classification result with 79.3% is based on the images from Google Street View. To determine whether the results actually differ from each other, again significance tests for the differences between geometric and textured representations results have been performed. The difference between untextured LoD3 models and textured meshes/LoD2 models from Google Earth is not significant but there is a significant difference between the geometric representation and images from Street View. One reason for the superior correctness obtained for Street View representations could be the viewpoint of the models. As exemplarily shown in the last column of Fig. 4.1, all images are captured looking slightly upwards and thus resembling the human perspective. The viewing angle dependency on classification results is beyond the scope of this chapter and will be addressed in further research. For building categories that are easily separable from the rest like One-Family Buildings and Industrial Facilities, a geometric representation is sufficient in the majority of cases. Particularly for buildings that are belonging to somewhat more ambiguous categories like Buildings With Shops, Office Buildings and Multi-Family Buildings additional textural information improves the classification results.

4.5 Application: Perception-Based Abstraction

The knowledge derived in section 4.4.1 describes geometric 3D building properties and structures which are characteristic for a specific building category. A lot of applications where users have to move about in virtual 3D cities can benefit from this perceptual knowledge.

Particularly when the virtual 3D city consists of abstracted, geometrically simplified buildings, e.g. when applications are visualized on small screens, it is even more important that the abstracted building representations still contain those geometric properties and structures which are essential for perceiving the correct category of the buildings. In the following, we will show how such perceptual knowledge can be embedded in a 3D abstraction process. Effects on the perceptibility of the buildings' categories will be demonstrated based on representative examples.

In a preprocessing step, information about perceptual relevance is attached to the respective 3D structures to provide semantically enriched building representations as input for the abstraction process. Based on Nan et al. [Nan+11], we create different abstractions of buildings based on human perception. Nan et al.[Nan+11] applied different Gestalt rules to drawings of façades which helped them to group drawing elements and to represent them by other elements. We extended this idea to the three-dimensional blocks formed by the façade elements and use it for abstracting given buildings.

During this process, we use the Gestalt laws of proximity, regularity and similarity to group blocks together and represented the results by larger blocks. The preservation of geometric properties and 3D structures, which are essential for perceiving the correct building category, is ensured by translating them into geometric constraints as restrictions for the abstraction process.

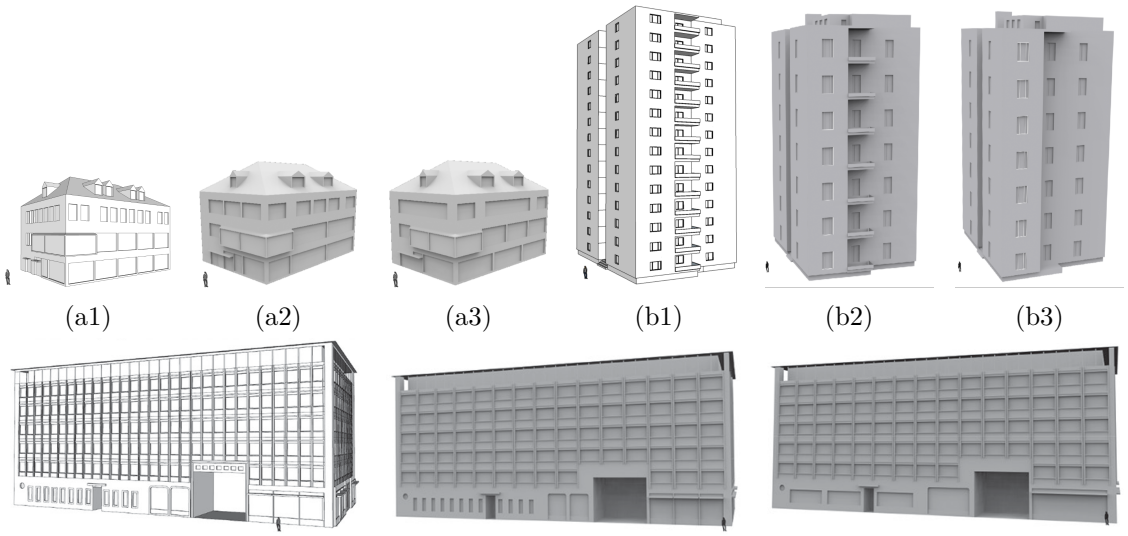


Figure 4.3: Application of conclusions drawn from the survey. For $x \in a, b, c$: (x1) original building model, (x2) abstraction based on features important for the user to classify into the respective correct category, (x3) 'free' abstraction without restrictions.

Fig. 4.3 (a1), (b1) and (c1) depict the original building models, respectively followed by two different results of the abstraction process. For the first abstraction, parameters have been chosen based on features that are important for the user to correctly classify

a building. The second abstraction is completely free, meaning that no restrictions were made during the abstraction.

Fig. 4.3 (a) depicts a model belonging to the class of *Building With Shops*. The first abstraction incorporates the properties learned to be important for *Building With Shops* as mentioned in section 4.4.1, paragraph (a). The ratio of the window size between the ground floor, the first floor and the remaining floors is preserved as well as the arrangement of the windows. The second model is a free abstraction.

As a result of the abstraction process, both models (a2) and (a3) have merged dormers. However, the window shapes and distribution have changed. For example, (a2) retains smaller windows in the upper floor, while (a3) has a merged window front. This merged window front destroys the building's original property of having significantly bigger windows in the lower floors than in the remaining floors, which was detected to be an important feature of *Buildings With Shops*, though.

In Fig. 4.3 (b) a *Residential Tower* is depicted. For both abstractions, windows have been merged over two floors, as a consequence the total building height appears to be smaller and the number of floors decreases with increasing single floor height at the same time. This exactly corresponds to the findings made for the users' expectation of the category *Residential Tower* (Tab. 4.3). The important feature *'balcony'* is maintained in the first abstraction, the second abstraction however drops it. This way, model (b2) retains the appearance of a residential building, whereas model (b3) is more neutral and, thus, could also be interpreted as an *Office Building*.

Fig. 4.3 (c) shows the example of an *Office Building*. The abstracted model (c2) keeps the characteristic structure of the ground floor but merges windows in the upper floors, thus still closely resembling the original. Model (c3) though merges windows and entrances in the ground floor. As a consequence, the model might rather be perceived as *Building With Shops* than *Office Building*.

4.6 Summary

With the aim of deriving knowledge on the human's ability to understand semantics from 3D building structures, we presented a user study on the user's comprehension of building categories based on different 3D building representations. Within the study, the users were asked to classify consecutively presented single building representations into categories. During the whole classification process, the users additionally had to rate their level of certainty. The representations shown to the users were untextured LoD3 models, textured meshes/LoD2 models from Google Earth, and images extracted from Google Street View.

Analyses of the user study reveal clear coherences and dependencies between the correctness of classifications and the model representation type. In general, it is conducive to have textural information for buildings.

The classification accuracy of LoD3 models mainly depends on whether the building models show properties that have been detected as perceptually relevant for the respective building category. As these properties are not inherent in all representatives of the categories mentioned, users sometimes experience difficulties to distinguish between differnt categories. Moreover, the majority of the users is not even aware of their misinterpretations which makes perception-adapted building representations an even more important issue. Therefore, it is crucial to guide the representation based on features that are significantly characteristic for the respective building category.

As a first application, we demonstrated how such knowledge about the human's perception of building-related semantic information can be used for the perceptually adapted abstraction of 3D building models.

In our future work, we plan to further extend our findings about the human ability to understand building categories based on user studies which will be implemented in the our interactive abstraction framework presente in chapter 3. The perceptual knowledge gained from those analyses can be embedded into this framework with which it will be possible to not only maintain perceptually relevant geometric properties and structures as it was the case in the perception-guided abstraction but it will additionally allow to modify, emphasize, add or remove perceptually relevant structures in a targeted manner in order to automatically generate models that can be classified more easily into the respective correct building category.

CHAPTER 5

Procedural Urban Forestry

The placement of vegetation plays a central role in the realism of virtual scenes. In this chapter, we introduce procedural placement models (PPMs) for vegetation in urban layouts. These PPMs are environmentally sensitive to city geometry and allow identifying plausible plant positions based on structural and functional zones in an urban layout. PPMs can either be directly used by defining their parameters or learned from satellite images and land register data. This allows us to populate urban landscapes with complex 3D vegetation and consequently to enhance existing approaches for generating urban landscapes. Our framework's effectiveness is shown through examples of large-scale city scenes and close-ups of individually grown tree models. We validate the results generated with our framework with a perceptual user study and its usability based on urban scene design sessions with expert users.

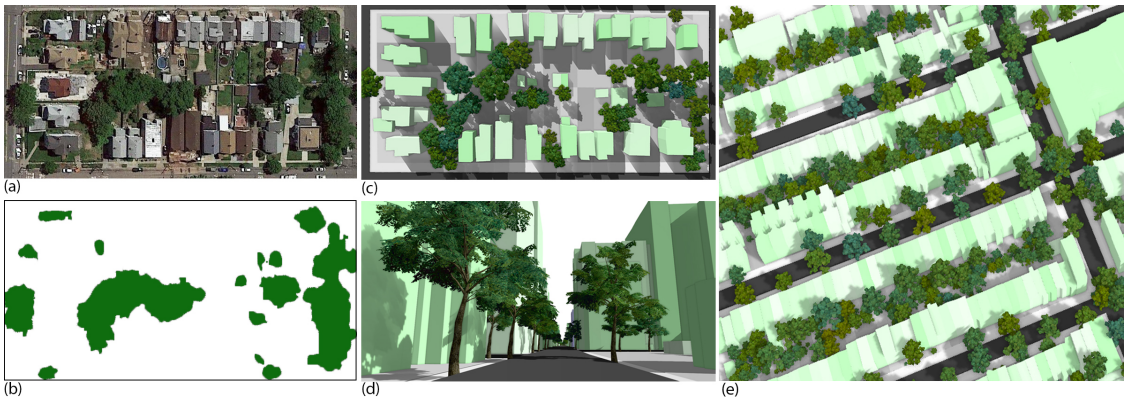


Figure 5.1: Steps of our learning-based plant population method: we use satellite images (a) and predict coverage maps for vegetation (b). We use these maps to identify plant regions for reconstruction (c) and to learn the parameters for our procedural models when populating new virtual cities with complex plants (d), which significantly increases the realism of urban landscapes (e).

5.1 Introduction

The visual simulation of urban models and the generation of their 3D geometries are fundamental open problems in computer graphics that have been addressed by many approaches. Existing methods range from modeling façades, buildings, city block subdivisions, to entire cities with viable street and road systems. Synthetically generated city models already exhibit a high degree of realism. However, cities are immersed in vegetation, but only very little attention was dedicated to the interplay of urban models and vegetation in computer graphics. Many approaches have considered ecosystem simulations. The prevailing algorithms use plant competition for resources as the main driving factor of their evolution either on the level of entire plants [Deu+98] or on the level of branches [Mak+19]. Unfortunately, these approaches fail in urban areas because urban trees have only limited space available to compete for resources. They are heavily affected by surrounding urban areas and human intervention.

The term urban forest refers to vegetation in urban areas [MHW15]. Vegetation has many practical functions: it limits and controls air movement, solar radiation, heat, humidity, and precipitation. It can also block snow and diminish noise. Moreover, an essential function of vegetation is to increase city aesthetics. Urban forests are not planted at once but managed over time. Dead trees are removed, and new trees are planted. Living trees are pruned for visibility or utility services. In contrast to real cities, we face a different situation in computer graphics. An existing algorithm generates a city model without vegetation, and we need to find suitable locations for individual trees. Simulating urban forest evolution, *i.e.,* by using the algorithm by Benes et al. [Ben+11], is time-consuming and challenging to control.

We introduce a procedural method for the advanced placement of vegetation to increase urban models' overall realism. We are inspired by urban rules that control which trees and bushes can be planted and how tall they are allowed to grow. These rules vary for individual areas; they are relaxed in industrial zones. People also have more flexibility in their properties, but they are enforced in public zones of a city and around important landmarks. Therefore, we introduce procedural placement models – strategies for generating plant positions – along with parameters to enable an automatic placement of vegetation, faithful to the characteristic features of plant distributions within the different municipality zones of a city. We show that placement models and parameters together provide an efficient means of controlling urban landscapes' interactive modeling.

Moreover, we can populate city models with static tree geometry and dynamic models of plants that can grow and change their shape in response to environmental changes or human intervention. This allows us to apply simulation models that describe how a city or its areas would change if more or less effort could be spent on maintaining them. Having such dynamic urban ecosystems allows users to visually predict and control gardening

effects in a city and make such models more realistic since they inhibit decay and different order levels.

While procedural placements can be used directly to populate urban layouts, we also show that placement models can be used to learn plant distributions of real cities. We use satellite images and land register data to train deep neural networks to learn trees and other plants' distributions in our procedural placement models' parameter space. While placement models act as a strong prior to regularize finding plausible placements, learning parameter values also enable users to efficiently author scenes through intuitive parameters. The example in Fig. 5.1 shows a satellite image (a) and the predicted coverage map (b). We use coverage maps to identify areas where to place vegetation (c) and learn the procedural models' parameters. Once the parameters are obtained, we can automatically populate city models with complex models of plants (d) to increase their realism (e).

Our main contributions are: (1) we advance the state-of-the-art in modeling vegetation in urban landscapes by introducing a procedural modeling framework that is based on the idea to factorize the complexity of plant placement into manageable components; (2) we introduce a set of procedural placement models along with their parameterization to capture a large variety of placement patterns; (3) we use a novel pipeline for learning plant distributions in cities from satellite data; we convert satellite images into coverage maps and then learn the placement parameters of our procedural models.

5.2 Overview

Generating plausible vegetation models for virtual urban landscapes faces two significant challenges: first, plant placement varies across different functional and demographic zones (Fig. 5.2a)– an industrial zone may only have a small number of non-managed plants, while residential areas not only have regularly placed trees alongside roads but also in gardens and parks. The planting rules depend on culture, habits, city rules, etc. They are difficult to quantify. Second, plant models need to simulate growth and interaction with their environment to generate vegetation with high visual fidelity. Moreover, urban trees are often pruned or may lack resources (water or light), which hinder their growth and affect their structure.

To address these challenges, we propose a two-stage procedural modeling pipeline. First (Fig. 5.2), we introduce PPMs (b) to generate plausible plant positions based on placement strategies and known planting rules for vegetation. A PPM can be defined for each functional or demographic zone of a city (*e.g.,* residential, commercial, or industrial) and operates on single lots of land (realty). Each PPM has a different set of rules parameterized by structural and positional parameters to capture the various kinds of planting patterns found in real cities. Second, once the plant positions are generated, we use a

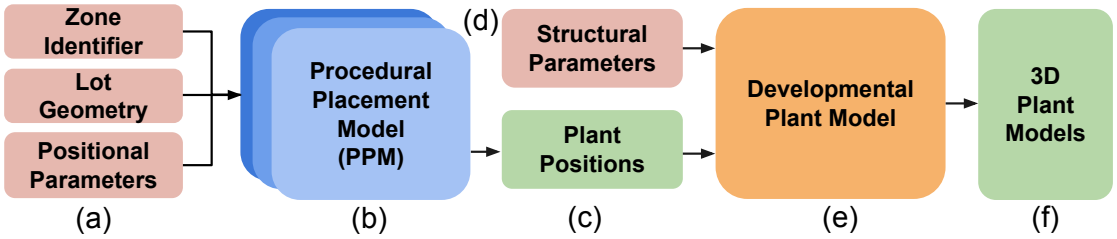


Figure 5.2: To place vegetation in urban environments we propose procedural placement models (b) that implement placement strategies for vegetation based on the geometry of individual lots, positional parameters, and a zone identifier (a). After plant positions (c) have been generated we use a developmental model (e) along with structural parameters (d) to jointly grow plants, which results in realistic 3D plant models.

state-of-the-art developmental model (Fig. 5.2, e) for growing plants. Given the plant's location and environment, the growth process generates unique and realistic branching structures.

Finally, we have developed a novel learning-based pipeline for populating models of real cities with vegetation. First, we convert satellite images of urban landscapes to *vegetation coverage maps* by using a style-transfer network (Fig. 5.14, b). The coverage maps represent areas that are covered with above-ground vegetation. Second, we learn a mapping from the coverage maps to the parameters of our PPMs (Fig. 5.14, d). Given our pipeline and the parameter values obtained from real satellite images, we can generate vegetation similar to what can be observed in the satellite images.

5.3 Planting Rules

Road networks define landscapes as administrative or functional zones [Wad+07], and they can be further classified into rural, exurban, suburban, and urban areas [MHW15]. All the involved plants form the urban forest, an umbrella term referring to trees, shrubs, and bushes found in urban and suburban areas.

A common way of introducing vegetation into an urban forest is by replacing a dead tree. Only newly created developments have large areas directly populated by vegetation. When a new neighborhood is built, a city will plant regularly spaced trees and bushes parallel to roads and sidewalks by applying municipal tree ordinances [Gre95] (see also [MHW15, pg 254]). The neighborhood is subdivided into blocks and blocks into individual lots left to the owners to plant the vegetation as needed. Typically, the city only defines specific planting rules such as the distance between individual trees should depend on the tree height, or the distance is derived from the soil the tree requires to survive [End18]. Trees should not obstruct views at intersections. They should have a certain distance from the curb and sidewalks [BR93]. Vegetation must not block

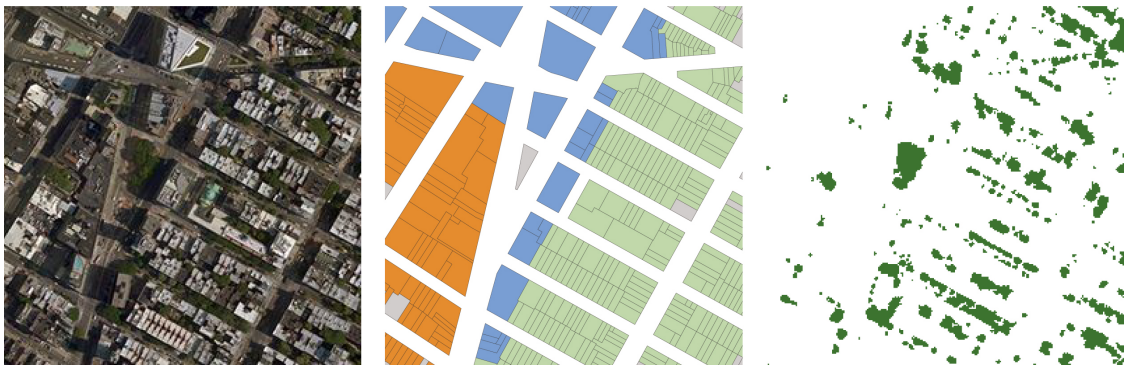


Figure 5.3: Urban layout: satellite images (left), zone data for individual lots (middle), and coverage maps (right) are available in public datasets. We use zone data and lot geometry as inputs to our procedural models and learn to predict their parameter values from the coverage maps.

house entrances for emergency purposes. These functional restrictions are also combined with aesthetic constraints: vegetation should not be planted in the proximity of windows [MHW15]. Most of these rules are incorporated into a so-called building activity area (or building envelope) that is an extension of the building's 2D projection by about 600cm perpendicularly from each building wall and 150cm from each driveway.

At a higher level, we aim to generate vegetation for the various types of zones procedurally. Therefore, we assume that each urban layout, either real or synthetically generated, can be divided into such zones. Specifically, we use a zonal layout commonly used in urban planning [Wad+07; Wad02] and urban simulations [Van+09; Van+10; Web+09] and divide an urban layout into five zones: 1) *residential* includes houses and buildings where people live, 2) *commercial* consists of businesses such as department stores, malls, and small stores, 3) *industrial* zones include factories and other production services, 4) *street* zones, which describe areas next to roads. We add a category (5) *other* that includes parks, non-managed areas, areas close to railroads, unassigned regions, etc.

As shown in Fig. 5.3, we further assume that a city layout is organized as individual lots, where each lot represents a property that may be occupied by a building. Given a lot and its zone type, we then define a PPM that places vegetation individually into each lot.

Based on the observations from municipal tree ordinances [Gre95; MHW15] and previous work on urban forests [Ben+11], we define six tree planting rules and show them in different real-world images in Sec. 5.4.1. *Random* placement within a lot follows a Poisson-disc distribution preventing trees from being in close proximity. Trees are often planted along lot *boundaries* as a noise barrier, but can also be *clustered* forming areas with grass and shade. Along streets trees often serve as a natural barrier and are planted in an *equidistant* manner along the medial axis of a lot. We can also observe *a single* tree within a lot or a *regular* placement.

In addition, trees are rarely planted at once and their distributions are mostly an emergent phenomenon of growth over long periods of time. Our objective is to populate an empty urban model at once. Thus we define the planting rules as geometrical distributions that allow us to encode plant populations as as procedural models, as shown in the next section.

5.4 Procedural Urban Vegetation

Vegetation for an urban landscape is generated in two steps: first, we apply a PPM to seed plants individually for each zone according to their functional types. After the virtual plants have been planted, we use a developmental model that dynamically grows them in their locations while interacting with the surrounding environment. This allows plant adaptation to their environment, such as bending and shedding of branches due to the competition for resources, resulting in vegetation with high visual fidelity.

5.4.1 Procedural Placement Models - PPMs

We seek to model plant morphology and the variance of plant placement across different municipality zones to distribute vegetation in an urban landscape realistically. Defining and parameterizing rules for obtaining plausible plant positions, while adhering to urban features such as buildings and streets, is intractable. Therefore, we factorize the problem into specifying placement models for the different zones as denoted in Sect. 5.3 (industrial, commercial, residential, street, and other) and for each lot.

The factorization allows us to define a manageable parameterization along with placement strategies for the different zones. Each placement model defines a concise strategy to place vegetation into a single lot. For example, we have models to place vegetation randomly, along the edges of a lot, equidistantly, etc. Moreover, we define the PPMs in a context-sensitive way. This means to maintain a global appearance, a PPM can query adjacent lots to adjust its parameters (*e.g.,* the distance between trees alongside a road in one lot should be the same in the neighboring lot). A PPM is a tuple

$$\mathcal{M} = \langle \mathcal{S}_g, \mathcal{P}_p, \mathcal{P}_s \rangle \,, \tag{5.1}$$

where $\mathcal{S}_g$ is a function implementing a *placement strategy* (rules) with $g \in \{R, B, C, E, S, I\}$ (see Sect. 5.4.2 and Tab. 5.2), $\mathcal{P}_p$ is a set of *positional* parameters to define the placement of plants, and $\mathcal{P}_s$ is a set of *structural* parameters for the morphological appearance of vegetation within the lot.

Lots and buildings are defined as 2D polygons possibly concave and with holes (see Fig. 5.4): $P_L = \{V_L, E_L\}$, $P_H = \{V_H, E_H\}$, where V_L and V_H denote the vertices of a lot

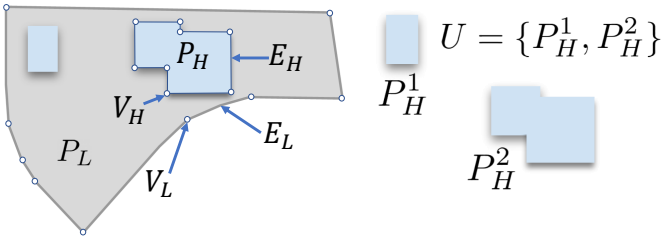


Figure 5.4: Lots and buildings are represented 2D polygons possibly concave. V_L and V_H denote the vertices of a lot (L) and buildings (H). A lot can include multiple buildings (or other structures), resulting in holes.

(L) and buildings (H) and E_L and E_H the edges of the polygon for lot and building, respectively. A lot can include multiple buildings (or other structures): $U = \{P_H^i\}$. The polygon $P = P_L - \cup P_b^i, \forall P_b^i \in U$, defines the area of a lot that can be covered by vegetation; the PPM only places vegetation within the geometric shape of the polygon P. A set of plant positions for a single lot is then generated as

$$\mathcal{X} = \mathcal{S}_g(\mathcal{V}_p, \mathcal{V}_s, P, Z, \mathcal{K}), \tag{5.2}$$

where $\mathcal{V}_p$ and $\mathcal{V}_s$ denote the parameter values for positional $\mathcal{P}_p$ and structural $\mathcal{P}_s$ parameters, P is the polygon of a single lot, Z is a zone identifier, and $\mathcal{K}$ is the context of a lot. We use Z to select parameter values for each lot. For example, a residential and a commercial lot may use the same strategy (*e.g.,* boundary) but differ in their parameter values (*e.g.,* different species are used). This is illustrated in Fig. 5.5. Generating vegetation with the same value for Z produces a uniform appearance (the same settings are used for every lot), while varying Z with the functional zones generates a diverse yet coherent appearance. Put differently, Z allows us to control the placement of vegetation on a global scale. Finally, we use $\mathcal{K}$ to modify the input parameters according to the neighbors of a lot to allow for consistent global appearance as detailed in Sect. 5.4.5.

To summarize: a PPM defines a placement strategy and structural and positional parameters for populating single lots. Varying these parameters' values generates different plant positions within the constraints of the strategy at a local scale while changing the parameters jointly – *e.g.,* based on zoning types – allows us to vary vegetation at a more global scale.

5.4.2 Placement Strategies

A placement strategy $g \in \{R, B, C, E, S, I\}$ (Random, Boundary, Cluster, Equidistant, Single, and Individual tree) defines rules for placing the plants and how the parameters are used.

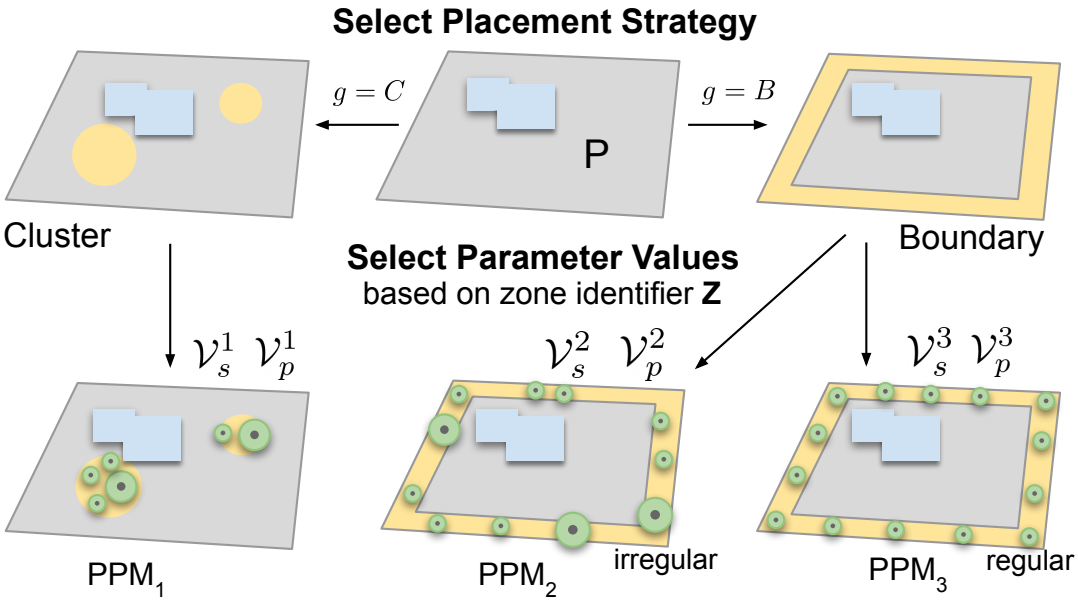


Figure 5.5: Given a lot, we use a placement strategy to define the placement of vegetation. The zone identifier Z is used to select parameter values for structural $\mathcal{V}_s$ and positional parameters $\mathcal{V}_p$. Together, strategies and parameters allow us to generate vegetation with globally similar appearance depending on the municipality zones within a city.

To implement the different placement strategies, we compute active areas within each lot that define where the vegetation can be placed. For the strategies *random* and *single* the entire lot polygon P_L is used, while for the strategies *boundary* and *cluster* we define active areas within the polygon; *i.e.,* we define a boundary along the edge of the polygon towards its center for *boundary* and a circular area around a randomly selected point within the polygon for *cluster*. For *equidistant*, we compute the medial axis of the polygon and then generate equidistant plant positions along the axis. The strategy *single* defines a single plant's random placement within the entire lot. Finally, for *regular* we compute a lot-aligned lattice and place plants at the center of each cell. Fig. 5.6 shows four of our six placement strategies and their parameter variations.

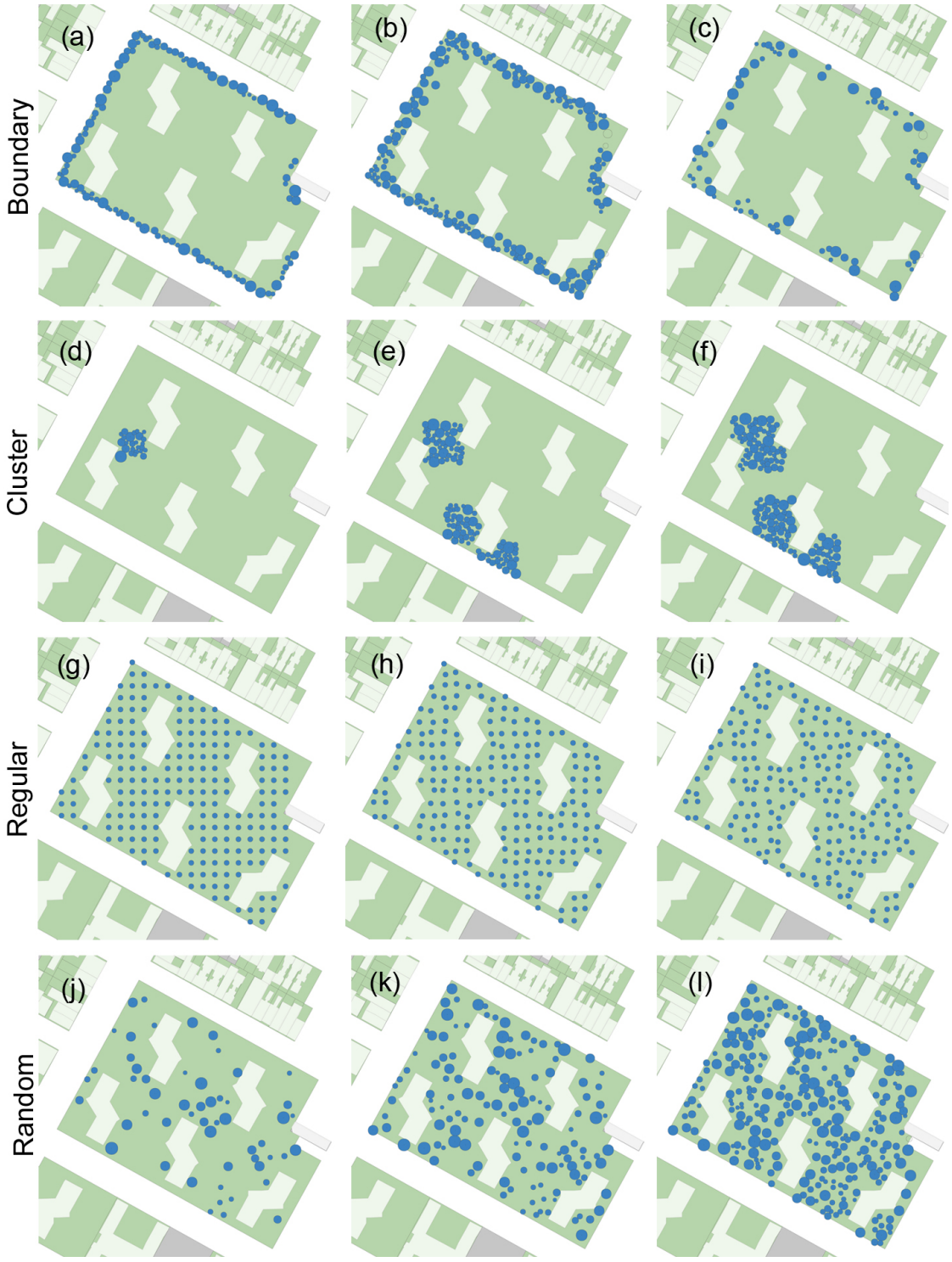


Figure 5.6: Variations of positional parameters on a single lot with different placement strategies. (a)-(c): strategy *boundary* with narrow (a) and wide (b) boundary size, and less density (c). (d)-(f): strategy *cluster* with a single cluster (d) and multiple clusters (e) of different sizes (f). (g)-(i): strategy *regular* with no (g), medium (h), and high (i) jitter. (j)-(l): strategy *random* with low (j), medium (k), and high (l) density.

5.4.3 Positional Parameters

Random, Boundary, and Cluster: The placement strategies are parameterized by the positional parameters shown in Tab. 5.1. We use the Variable Radii Poisson-Disk Sampling [Mit+12a] to generate plant positions within active areas of a lot (see Fig. 5.7). More specifically, we are interested in generating a set of points $\mathcal{X}$ with spatially varying point density. A new position sample y is assigned a radius $r(y) : \Omega \rightarrow \mathcal{N}(\mu, \sigma)$, where $\mathcal{N}$ denotes a normal distribution with mean μ and variance σ. The new position sample y is accepted and added to the set if $|y - x| \geq r(x) + r(y) \forall x \in \mathcal{X}$.

For the *boundary* placement strategy we define the boundary size as parameter β that defines an area along the normal of the edge of a polygon towards its center. To implement the *cluster* strategy, we randomly sample points in a lot and define the cluster area as a circle with a radius κ. A lot can have a variable number of clusters with the maximum number defined by π. For both strategies, *boundary* and *cluster*, we first compute the active regions (boundary, cluster circles) before generating sample positions.

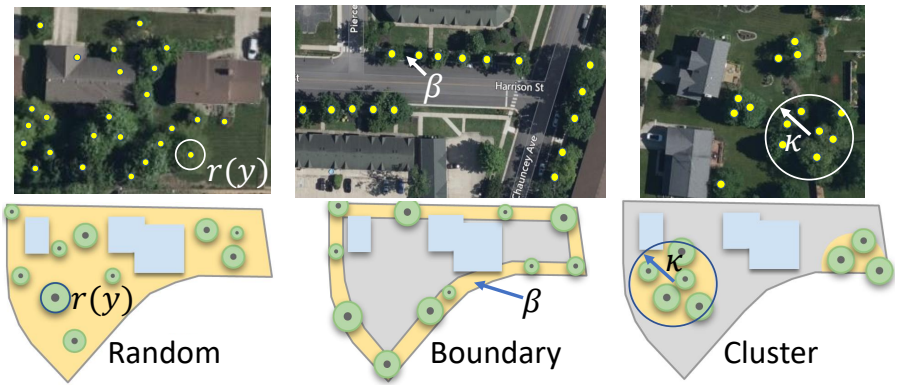


Figure 5.7: *Random*, *Boundary*, and *Cluster* placement strategies use Variable Radii Poisson-Disk Sampling to position trees.

Regular and Equidistant: allow for semi-regular vegetation placement. For the *regular* strategy, we compute a regular lattice based on the bounding box of a lot and define the size of cells with ω and their orientation with η. We optionally jitter the positions using ψ within each cell. To implement the *equidistant* strategy, we first compute the medial axis of the lot polygon P_L [Cho+97] and then equidistantly place plants along the axis based on the distance parameter δ. We model the *density of vegetation* for all placement strategies by defining the parameter τ, which deactivates position samples in $\mathcal{X}$. A value of $\tau = 1$ activates all samples, while a value of $\tau \leq 1$ randomly deactivates them until all samples are deactivated ($\tau = 0$). Finally, we define the radius ξ for the context $\mathcal{K}$ of a lot. The context is defined as the adjacent lots, and we use it to model context-sensitivity (see Sec. 5.4.5).

Single: Finally, we sample one random position in the lot for the strategy *single* (see Fig. 5.8).

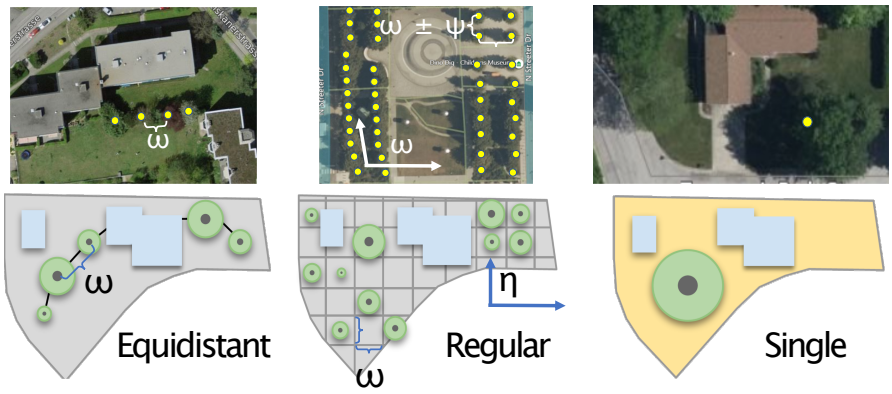


Figure 5.8: *Equidistant* uses a distance value ω, *Regular* a distance value ω with an optionally jitter ψ, and an orientation η, and *Single* one random sample for positioning

Building Envelope: Trees should not be too close to buildings and should not obstruct doors and windows. We adopted the concept of building envelopes [MHW15] that defines the clearance distances from the buildings. Moreover, we extend the envelope in front of doors and windows to avoid their blockage (see Fig. 5.9).

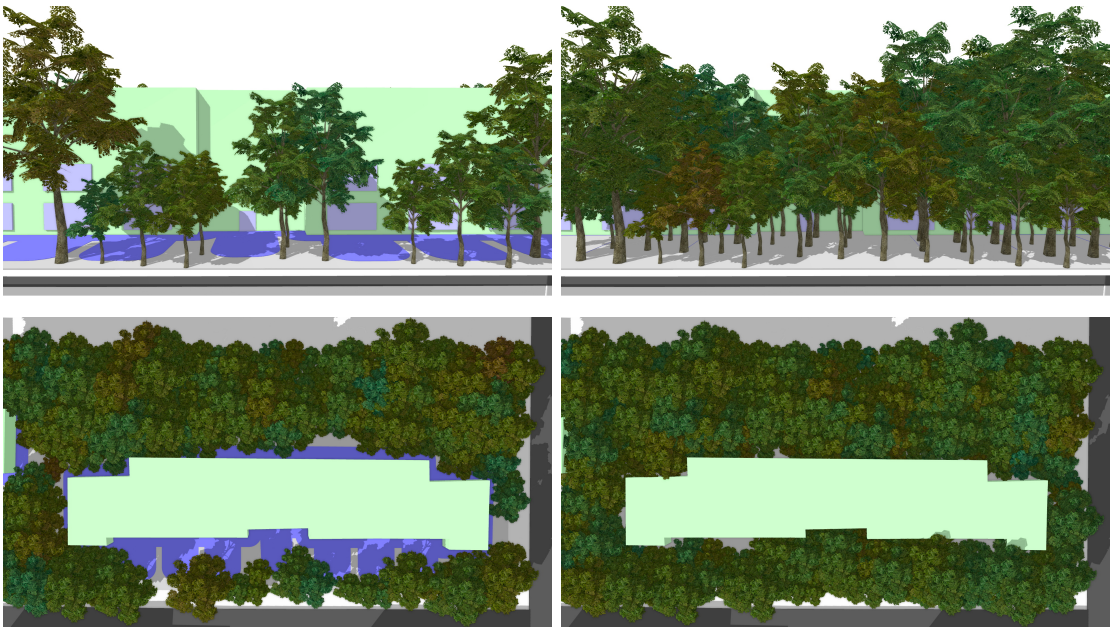


Figure 5.9: Left: the building envelope (blue) defines a zone where plants cannot be planted to avoid proximity to walls and blockage of door and windows. Right: plant placement without considering the building envelope.

Tab. 5.1 summarizes the positional parameters along with their ranges, and Tab. 5.2 shows the placement strategies and their corresponding positional parameters. Examples

of changing the values of positional parameters are shown in Fig. 5.6.

Table 5.1: Positional and Structural Parameters for PPMs

	Parameters	Meaning	Range/Dimensions
Positional	μ	Plant envelope mean	[1m - 10m]
	σ	Plant envelope variance	[0.1 - 2]
	τ	Vegetation density	[0-1]
	β	Boundary size	[0m - 5m]
	κ	Cluster radius	[1m - 20m]
	π	Max number clusters	[0 - 5]
	ω	Regularity grid size	[5m - 50m]
	ψ	Regularity jitter	[0 - 1]
	η	Regularity orientation	[0 - 180°]
	δ	Equidistant spacing	[0m - 10m]
	ξ	Radius of context	[0m - 300m]
Structural	α	Max plant age	[0 - 100 years]
	ρ	Tree vs shrub ratio	[0 - 1]
	θ	Species diversity	[0 - 1]
	γ	Pruning factor	[0 - 1]
	λ	Num. species	[1 - 10]

Table 5.2: Placement Strategies and used Positional Parameters.

Strategy	Symbol	μ	σ	τ	β	κ	π	ω	ψ	η	δ	ξ
Random	R	✓	✓	✓								✓
Boundary	B	✓	✓	✓	✓							✓
Cluster	C	✓	✓	✓		✓	✓					✓
Equidistant	E	✓	✓	✓							✓	✓
Single	S	✓	✓	✓								✓
Regular	I	✓	✓	✓				✓	✓	✓		✓

5.4.4 Structural Parameters

We define *structural parameters* to model the morphology of individual trees as well as the plant population within a lot. Based on the computed plant positions we define a plant seed as the tuple

$$\mathcal{T} = \langle\, p, \alpha, \phi, \gamma \rangle\,, \tag{5.3}$$

where $p \in \mathcal{X}$ is the plant position, α its maximum age, ϕ denotes a species identifier, and γ is a pruning factor. To generate branching structures we grow a plant with a developmental model (see Sec. 5.4.6) and jointly simulate its growth with all other plants in a lot.

We define several species ($n = 10$) for the whole urban landscape by selecting parameter values for our developmental model [Pal+09]. We then use the species identifier ϕ to associate one of the species to a seed. We further control this selection by using the parameter ρ, which defines the tree vs. shrub ratio in a lot. A value of $\rho = 1$ assigns all seeds tall-growing species, while a value of $\rho = 0$ only associates short growing ones.

To vary the number of used species in a lot, we use the parameter θ. We randomly select one of the species as the dominant species in a lot and use θ as a ratio to control the

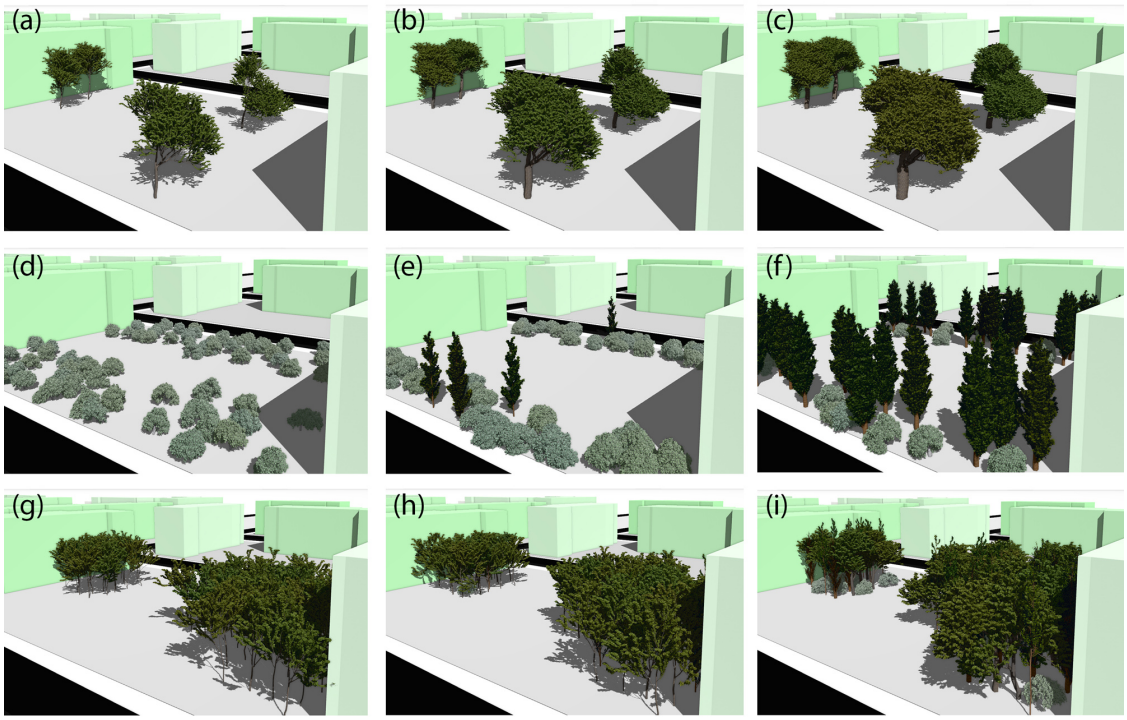


Figure 5.10: Variations of structural parameters. Top row: variations of age parameter from young (left) to old (right). Middle row: changes of tree to shrub ratio from only shrubs, to mostly trees. Bottom row: variations of species diversity from a single species (left) to multiple species (right).

number of seeds associated with the dominant species and all other available species. A value of $\theta = 0.5$ sets half of the available seeds to the dominant species and the other half with randomly selected ones.

Finally, we may prune a plant by a bounding volume for the tree crown of a fully developed model. This allows us to generate a more organized appearance of vegetation, *e.g.,* along avenues or highways. Branches that reach out of the volume are cut off. We scale this volume by γ; a value of $\gamma = 1$ will leave a plant unpruned, while a value of $\gamma \leq 1$ scales the bounding volume and therefore results in a pruned plant. After pruning, we again simulate the plant growth to develop smaller branches and leaves. Fig. 5.13 shows an example of the pruning of trees; other variations of structural parameters are shown in Fig. 5.10.

5.4.5 Context-Sensitive Rules

So far, lots have been treated as individual units without any mutual relationship. However, each lot has its context that are its surrounding roads and neighboring lots. The neighbors often share similar planting rules provided by the applying municipal tree ordinances [Gre95; MHW15]. To account for the context of lots we want to adjust planting rules.

While context-sensitive plant seeding has not been addressed before, there is a body of related work on the environmental sensitivity of individual plants that is closely related to a plant's ability to adapt to varying conditions *e.g.,* it may bend its branches against gravity (gravitropism) or grow towards the brightest spot (phototropism). A plant optimizes different functions by using this plasticity. Context-sensitivity can be proceduralized as context-dependency, for example, by using environmental query modules in Open L-systems Fetching the context values is a two-pass method: first, the context is queried, then the values are interpreted by the procedural system.

Inspired by this previous work, we generalize the context dependency to PPM.

Let us recall that each PPM from Eqn. (5.1) has associated a placement strategy $\mathcal{S}_g$ and two sets of parameters $\mathcal{P}_p$ and $\mathcal{P}_s$. Each lot has a set of parameter values from Eqn. (5.2) $\mathcal{V}_p$ and $\mathcal{V}_s$. Moreover, it considers the context (*i.e.,* the neighborhood) $\mathcal{K}$ of the lot that is being populated with plant positions Eqn. (5.2). Futher, let us denote a particular lot L and its parameter values as $\mathcal{V}^L$. In the following text, we will omit the lower index s and p because the parameters are calculated in the same way. The context is the set of lots within radius ξ centered on the lot L and weighted by a 2D Gaussian. The values of the corresponding parameters (see Tab. 5.1) of the neighbors and the lot L are weighted according to the distance resulting in a context-updated parameter set $\tilde{\mathcal{V}^L}$ as:

$$\tilde{\mathcal{V}^L} = \sum_{\forall \mathcal{V}^K \in \mathcal{K}} w\left(d(L, L^K)\right) \mathcal{V}^K, \tag{5.4}$$

where $w\left(d(L, L^K)\right)$ is the Gaussian-weighted distance between the center of the lot L and L^K within the investigated context, and $\mathcal{V}^K$ are the values of the parameters of the lot L^K. The updated parameter values $\tilde{\mathcal{V}^L}$ are then used for the PPM.

Note that this process can be considered as a diffusion of the parameters within radius ξ. Also, to avoid a race condition when one lot serves as a context of another one and vice versa, we calculate the context-updated parameters $\tilde{\mathcal{V}^L}$ into a different map. In this way, the calculation does not depend on the order of the lot selection and can also be done in parallel. Note that if we would apply Eqn. (5.4) multiple times, the parameters' values would be smoothed out into an average over the entire layout. Fig. 5.11 shows the effect of using context-sensitivity on a regular placement of trees. The first row shows two lots with regular tree placement with an abrupt change to a random placement in neighboring lots that is smoothed out into a semi-random transition when the context is used (bottom row).

5.4.6 Developmental Plant Model

After generating plant positions, we jointly grow the plants in the computed locations of a single lot. Our developmental model is based on the work of Palubicki et al. [Pal+09];

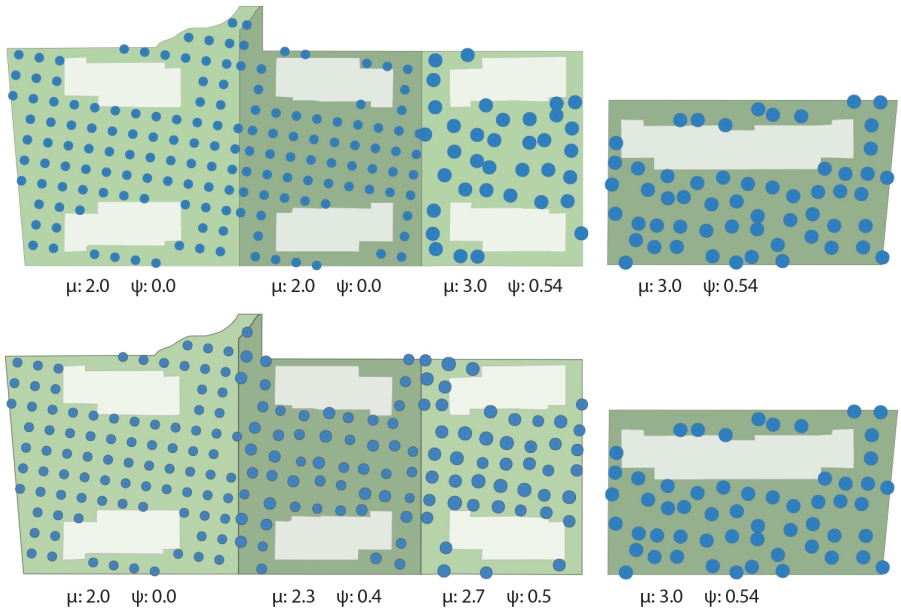


Figure 5.11: Context-sensitivity: we calibrate the parameter values of a lot with those of adjacent lots (context). Here we show two lot configurations with regular placement strategy and variations over the parameters μ and σ. For the lots shown in the top row context-sensitivity is turned off and plant placement changes abruptly from one lot to another, while for the bottom row we show context-sensitivity across lots and the resulting calibration of parameters (context radius: $\xi = 180m$).

a tree is a modular system (leaves, buds, stems, and internodes). An internode is a plant stem between two or more leaves, and a tree is composed of a succession of internodes.

The primary plant development is controlled by the expansion of buds that are either apical (terminal) or lateral (axial). Branches expand at their tips by expanding their apical buds or on sides by growing lateral buds. Buds use signaling by the growth hormone Auxin to prevent overgrowth and to control apical dominance [Keb17]. Secondary plant development (cambial growth) is the thickening of a tree trunk and branches [Kra+15] simulated by expanding their radii using da Vinci's rule (see [MT14] for a discussion).

Trees compete for space by seeking light (phototropism) and avoiding collisions and overcrowding. Many different algorithms have been implemented to capture plant competition for resources (see [RLP07; MP96] and [Pir+16] for an overview). We use the space occupation approach of [RLP07; Pal+09], which controls the growth by randomly scattered particles that attract growing branches. We also simulate phototropism by computing buds' illumination and bending the growth direction towards the brightest spot visible from a bud. Apical control and branching parameters are simulated by using the growth model from [Sta+10] with the set of parameters.

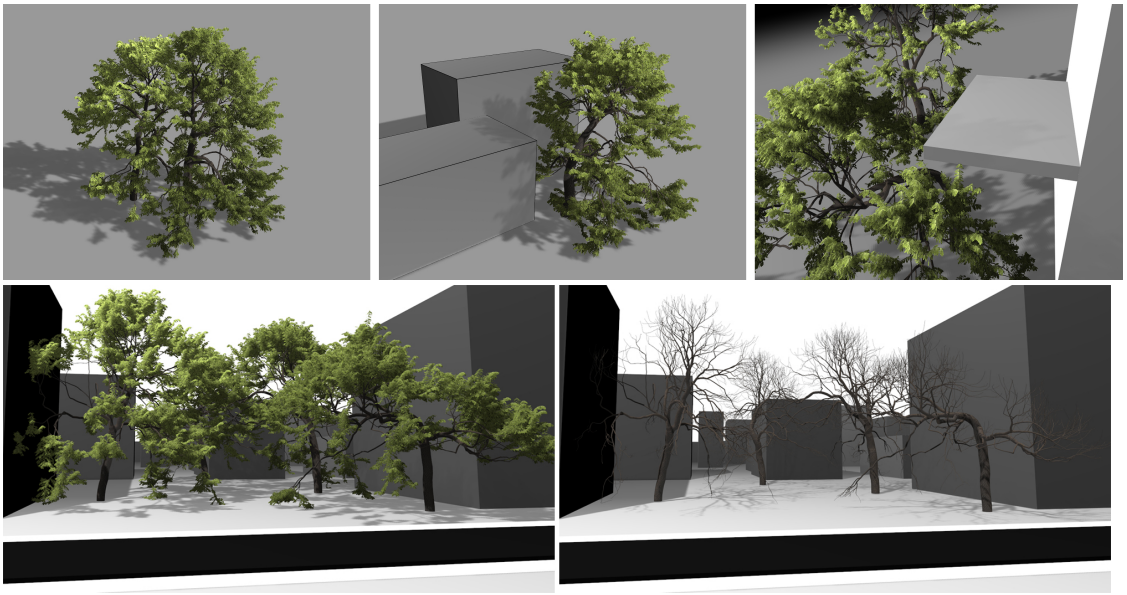


Figure 5.12: Top row: trees grown in different environmental conditions. From left to right: two trees close to each other, close to a set of buildings, and underneath a balcony. Bottom row: the growth response of a group of trees in an urban environment generates complex and unique branching structures.

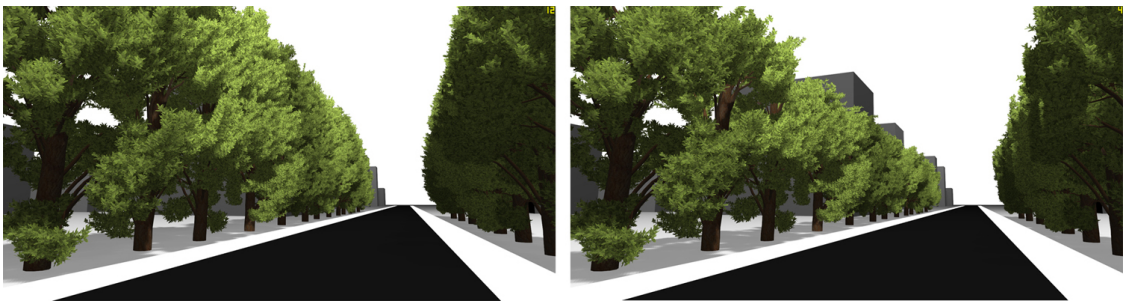


Figure 5.13: Pruning of branches allows for the adjustment and organization of tree form. Here trees along a street are severely pruned to form a hedge ($\gamma = 0.7$).

5.5 Learning Vegetation Placement

Learning plant positions directly from image data is a challenging problem that cannot be easily addressed by existing neural network architectures or other methods. To obtain plant positions in an end-to-end manner, a network would have to either output a variable number of plant positions or operate on a fixed size domain, such as an image. The latter requires to obtain plant positions as a post-processing step, which is error-prone. Furthermore, generating ground truth data pairs of satellite images and plant positions (*e.g.,* GPS coordinates) for training a neural network is challenging (see Sect. 5.7.3 for a discussion). Moreover, an end-to-end deep learning-based system would sacrifice an in-depth understanding of the underlying mechanisms. It would not allow for low-level control that is needed in interactive editing.

Therefore, to recover the placement and appearance of natural urban landscapes, we aim to learn plant distributions in our parameter space of positional parameters. This has the advantage that our above-defined PPMs act as a prior, which helps to regularize the training of our network and, in turn, to generate plausible plant positions. Furthermore, learning the procedural model parameters maps images to comprehensible and intuitive parameters, providing an efficient way to further edit plant placements.

5.5.1 Learning Plant Placements

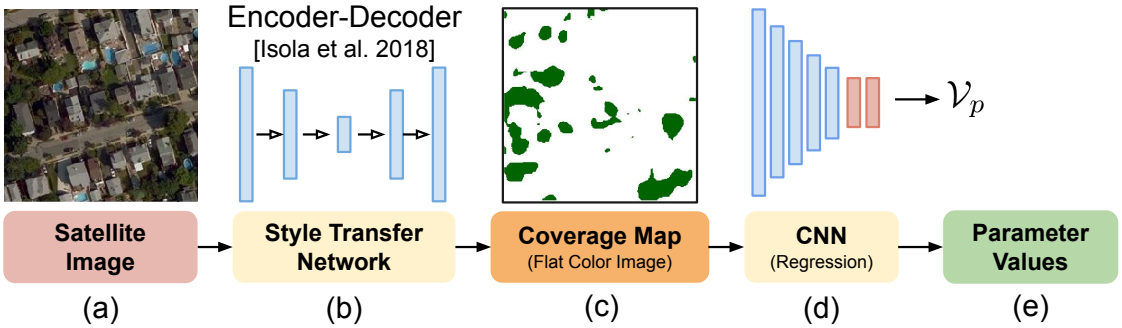


Figure 5.14: Neural network pipeline: we use a style-transfer network (b) trained on data pairs from NYCOpenData [NYC19] to convert satellite images (a) to coverage maps (c). To learn parameter values for our PPMs (for which no ground truth data for satellite images exist) we generate pairs of coverage maps and parameter values with the PPMs of our framework. We then train a CNN (d) to obtain parameter values (e) for the estimated coverage maps of the real satellite images.

We use a two-stage neural network pipeline to learn the parameters of our PPMs: first, we translate satellite images to semantic maps that describe vegetation coverage (Fig. 5.14, a-c). Second, we learn the positional parameters from coverage maps with a lightweight convolutional neural network (Fig. 5.14, d, e). This pipeline has the advantage that we do not need to rely on pairs of satellite images and positional parameters for training, but instead on pairs of coverage maps and positional parameters, which can be generated synthetically with our PPMs.

To translate satellite images to coverage maps, we used a style-transfer deep neural network [Iso+16]. A coverage map is a flat-colored image where every pixel color is based on whether the corresponding pixel in a satellite image represents vegetation. Coverage maps have less complex visual traits and are similar to real and synthetic data. Therefore, the network can learn this transfer. We used pairs of satellite images and coverage maps publicly available for some cities [NYC19] to train the style-transfer network to learn coverage maps from satellite images. This allows us to obtain coverage maps of cities for which coverage data does not exist. Fig. 5.15 shows examples of training data and generated coverage maps.

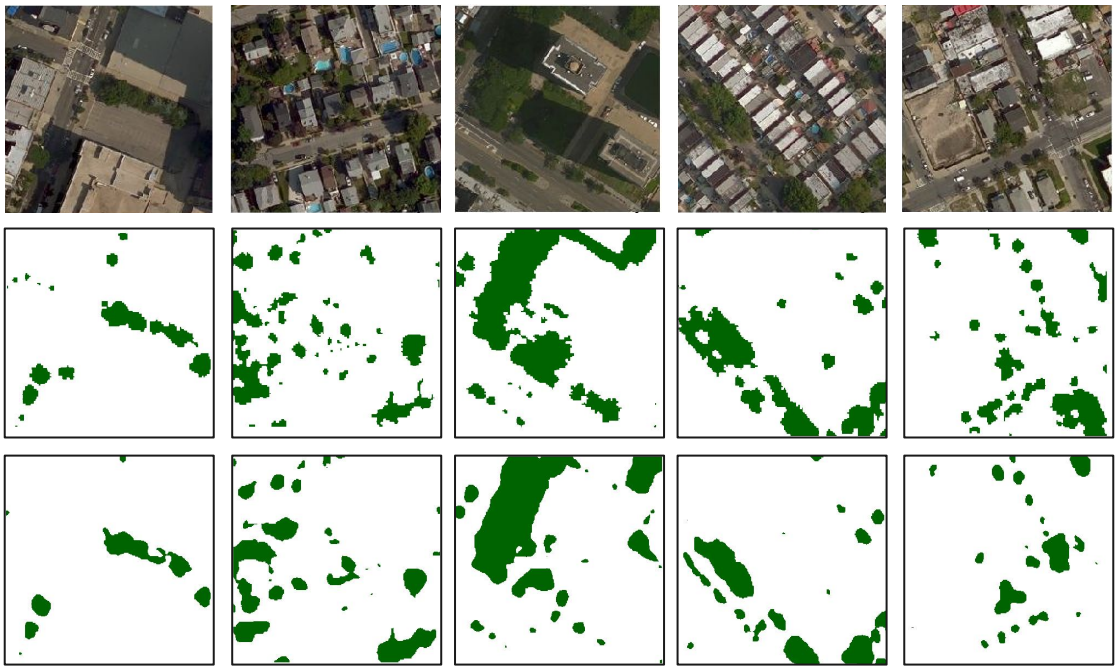


Figure 5.15: Learning of coverage maps: we use satellite images (top) and ground truth coverage maps (middle) from NYC Open Data to train a neural network for style-transfer. After training the network is able to predict coverage maps (bottom) from satellite images.

We then train a neural network to obtain positional parameter values ($\mu, \sigma, \tau, \beta, \kappa, \pi$, see Tab. 5.1) from the coverage maps. Training is done on synthetically generated pairs of coverage maps and positional parameters obtained from our PPMs. Specifically, we define the generated coverage maps as $q \in \mathcal{Q}$ for which we know the corresponding positional parameters $\mathcal{P}_p \in \mathcal{U}$. The network can thus be defined as

$$f(q) : \mathcal{Q} \rightarrow \mathcal{U}.$$

To summarize: stage one of our pipeline learns coverage maps from satellite images, which – in stage two – enable us to obtain the positional parameters of our PPMs. Together this allows us to generate vegetation positions for individual lots with similar characteristics as observed in the satellite imagery (*e.g.,* plant distance, density, etc.). Once the parameters are generated, we stencil the coverage map with each lot's geometry and identify areas to place vegetation for a reconstruction. We convert the regions into polygons and then use our *random* placement strategy to generate plant positions within the covered areas of a lot (Fig. 5.16). Please note that the random strategy is regularized by the positional parameters values (Tab. 5.1) learned by the CNN neural network (see Fig. 5.14, d, e). As a coverage map defines the areas where vegetation should be placed within a lot, it is sufficient to only rely on a random placement here.

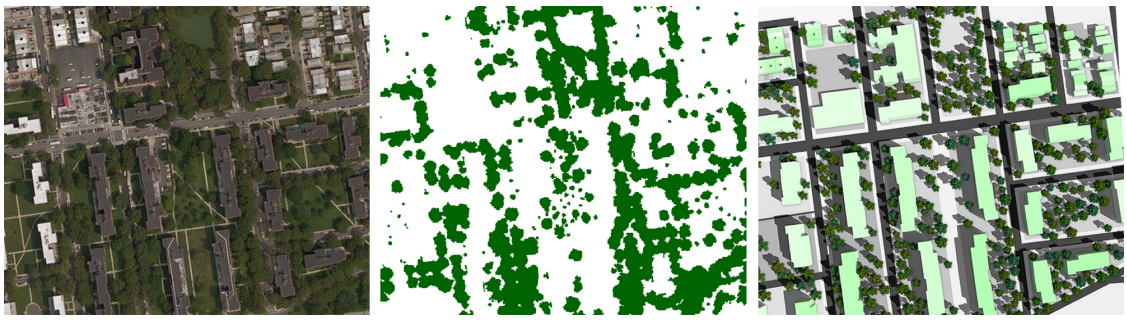


Figure 5.16: Vegetation placement based on real data: we use vegetation cover-age maps (middle) to identify active regions for individual lots and populate them with our PPMs. This allows us to generate plant distributions (right) similar to what can be observed in satellite images (left).

5.5.2 Data and Training

For training the Pix2Pix style-transfer network, we rely on the publicly available implementation of the original model implemented in Python. We train the network on 20K pairs of satellite images and coverage maps. This data is generated in three steps: first, we obtain satellite images from Google maps with a resolution of 256×256 pixels per image. The images correspond to the lowest level of the tile graph. Second, we use the vector data of streets, buildings, and lots from NYCOpenData [NYC19] and render them into image tiles of resolution 256x256. Third, we generate coverage maps by converting the vegetation coverage data provided by NYCOpenData (total rasterized resolution of 316.003 x 312.400 pixels) by reprojecting the data – provided in the geospatial data format: EPSG:2263 - NAD83 / New York Long Island – for each tile to match the Mercator projection used by Google Maps. We use the default hyperparameter settings for Pix2Pix [Iso+16]; the network converged after training for 200 epochs. We then use the network to convert satellite images of urban landscapes to coverage maps. The geometry of single lots is also obtained from the NYC Open Data. Our urban modeling framework operates on longitudinal and latitudinal coordinates, which allows us to register satellite images, lot data, and coverage maps, enabling us to render satellite images and publicly available maps (*e.g.,* Open Street Maps) in the same framework. Our regression CNN consists of five convolutional layers (32 units) followed by two dense layers (64 units) with relu activations for all except the last layer. We use our PPMs to synthetically generate 21K pairs of (coverage map, positional parameter value)-pairs to train the network. To regress the positional parameters, we use mean squared error as loss function and can achieve 95% accuracy for predicting the validation data's parameters. We use an 80% – 20% split for training and testing data. All results shown in the paper are generated from validation data.

5.6 Implementation and Results

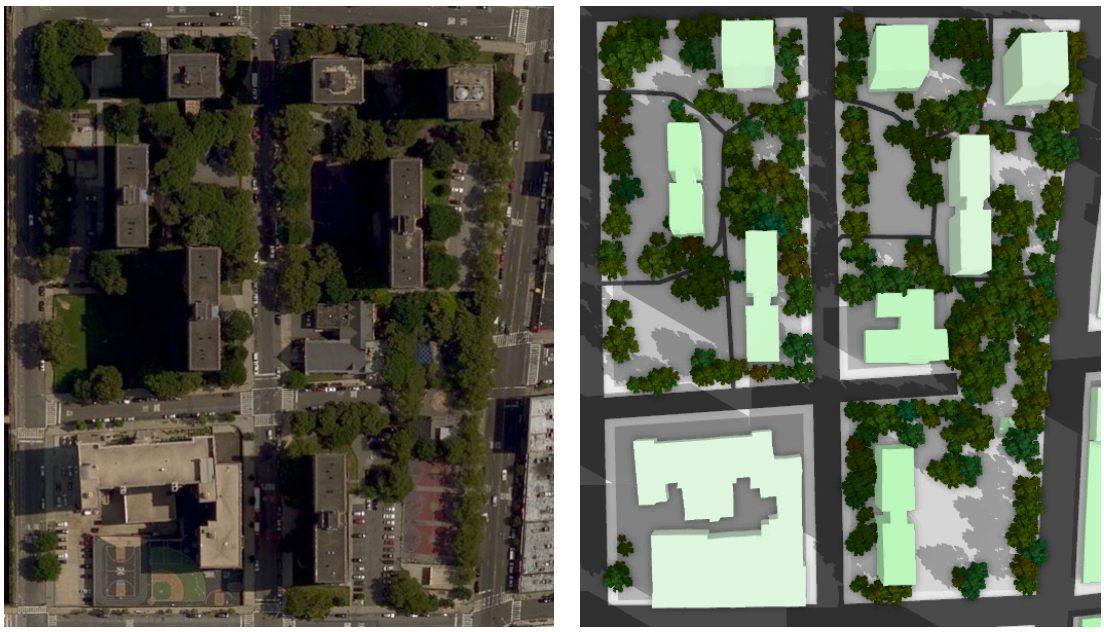


Figure 5.17: Given a reference satellite image (left), we can create a similar distribution of plants by manually choosing the strategies and the corresponding parameters. Here we used the strategies random, boundary, and cluster for the different lots (right).

Our interactive framework for modeling and rendering urban landscapes was implemented in C++ and OpenGL. All results have been generated on an Intel(R) Core i7-7700K, 8x4.2GHz with 32GB RAM, and an NVIDIA GeForce RTX 2080 GPU with 12 GB RAM.

The most demanding online task is the generation of tree geometry. We simplify this by representing trees by their skeletons that are generated on the CPU. We further offload the mesh generation of the branch surfaces into a geometry shader on the GPU. Similarly, leaves are generated as textured quads that are also generated on the fly. Buildings and other structures are rendered as extruded outlines. While we cannot render large plant populations in real-time, our framework allows us to explore placement strategies and parameter settings. To render large scenes (*e.g.,* Fig. 5.27), we use a level-of-detail scheme that successively replaces tree geometry with billboards and point primitives according to the distance from the camera. Tab. 5.3 shows parameter values for most figures shown in the thesis.

5.6.1 Interactive Authoring

We demonstrated that PPMs can automatically place vegetation into urban landscapes based on the lot data. The geometry of individual lots can either be obtained from

publicly available datasets or as a part of the modeling process, for synthetically generated layouts.

However, PPMs operate on polygons, and they were designed with interactive authoring in mind. The user can use a brush tool to draw an area on a map. We then convert the sketch to a polygon and assign a PPM. Depending on its placement strategy, the PPM will then generate plant positions according to the geometry of the polygon and its associated placement strategy (Fig. 5.19). Furthermore, a user can directly draw the vegetation coverage for individual lots or polygons. Like learning the coverage maps from satellite images, sketching a coverage map replaces the placement strategy for a lot. The PPM then places plants based on the positional and structural parameters, which provides a convenient way for more nuanced vegetation placement.

This process also allows us to generate even more diverse zones if necessary. For example, it is possible to define individual zones for back and front yards, the vegetation along streets, or even parks. Our approach's key idea is to factorize the complexity of defining a complex procedural model into more manageable placement strategies. A PPM only works on a single polygon and generates plant positions for this geometry. This way, it is easy to extend our approach by new placement strategies.

5.6.2 Results

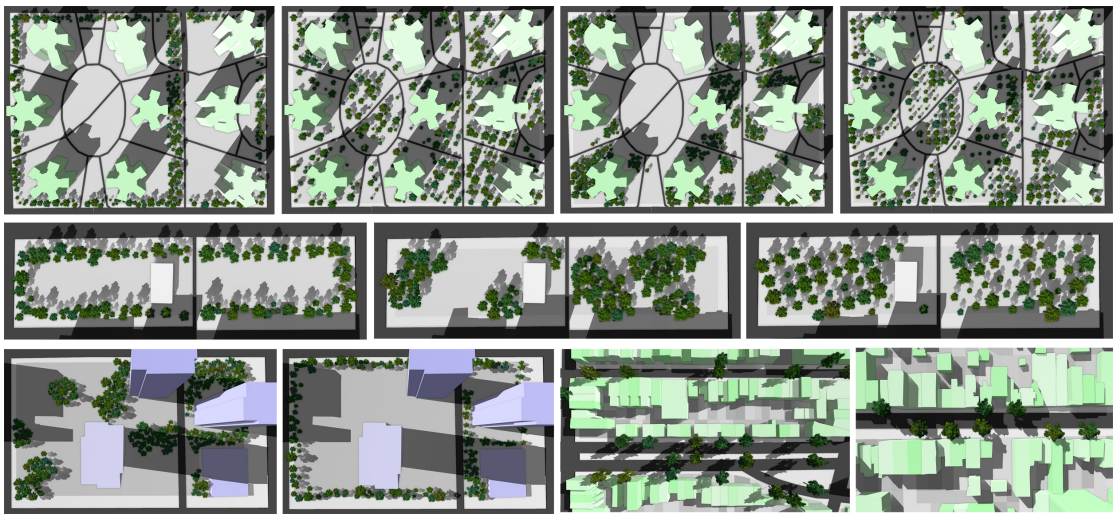


Figure 5.18: Top-down renderings of plant distributions for three municipality zones generated with different placement strategies. Top row: the placement strategies boundary, random, cluster, and regular for a residential lot of buildings. Middle row: the placement strategies boundary, cluster, and regular for the lot of a public park. Bottom row: the placement strategies cluster, boundary for a commercial lot (left) and the placement of trees with medial axis along streets with equidistant spacing set to: $\delta = 13m$ (right).

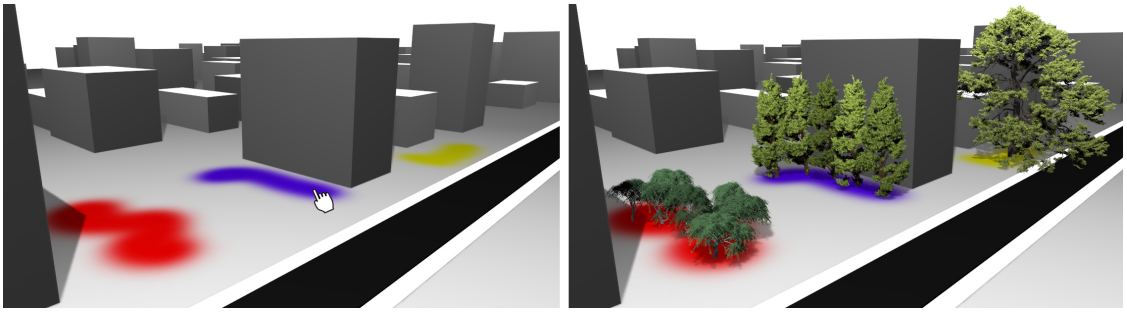


Figure 5.19: A user can interactively sketch placement zones with a brush tool (left). Each placement zone is converted to a polygon and assigned a placement strategy to grow plants (right). Here we show the strategies medial axis (blue), single (yellow), and random (red).

Figure 5.20: The placement of vegetation changes with the size of the active areas within a lot. While the used cluster strategy initially generates plants in the entire lot, transitioning to less available space due to a larger building (white) generates more organized plant positions at the boundary of the lot.

Figs. 5.1, 5.21, and 5.27 show perspective and top-down renderings of urban landscapes along with the vegetation generated by our framework. For these results, we used coverage maps to reproduce vegetation placement similar to the real scenes. Fig. 5.18 shows results where we only used our procedural model, without additional coverage maps. For both cases, the produced plant populations show characteristic visual traits found in real vegetation distributions at the city-scale. Based on our placement strategies, we can generate complex urban vegetation patterns in combination with the positional and structural parameters.

Moreover, we show vegetation placements for the different municipality zones (residential, park, commercial) in Fig. 5.18. Positional parameters allow us to generate planting patterns as commonly found in these areas. At the same time, we can also produce structural variations by selecting the number of species, their height, and their age (Fig. 5.10). Additionally, we can control the pruning of plants to generate more organized plant shapes (Fig. 5.13). In an urban setting, buildings often shade larger areas. Trees growing in these regions strive to grow out of the shadow toward the light. This interaction of a tree with other trees and close-by buildings generates complex and unique branching structures. Fig. 5.12 shows the modeling result of trees grown in varying environmental conditions.

Figs. 5.19 and 5.20 show the capabilities of our framework for the interactive authoring of urban landscapes. In Fig. 5.19, a user drew regions for vegetation onto the ground of an urban layout; each brush tool was assigned a different placement strategy and set of parameter values. Our method then converted the sketched areas to polygons and applied different PPMs. Fig. 5.20 shows how the placement of vegetation changes when the size of a building on a lot increases. While with a small building, there is more space for random plant configurations, the placement transitions to more organized plant positions when the building's size increases.

Figs. 5.26 and 5.27 show vegetation placement results for large scenes generated with our framework. To generate the results in Fig. 5.26 we manually defined the entire park as a single lot (middle) or generated multiple smaller lots (right). For the result shown in Fig. 5.27 we rely on the lot data provided by NYCOpen Data [NYC19] for the vegetation placement. When lot data is provided, our framework enables the efficient generation of realistic urban scenes.

Fig. 5.21 shows a comparison of using different placement strategies. Given the satellite images of different urban scenes (a), our method is able to closely approximate the real scenes by using coverage maps and PPMs (b). Additionally, we compare our results to different variation as ablation studies. In (c) we place plants randomly without any parameter regularization (fully random), but we use the coverage maps to define areas where vegetation can be placed. This setup does not account for the coherent parameterization of plants, which results in less realistic populations of plants. Tree species and their age are not selected consistently and plants are placed too close to buildings and to other plants. In (d) we show the result of our *random* placement strategy that generates random plant positions with regularized structural parameters (e). The result of placing plants *fully random*, without coverage map and without any regularization across the positional and structural parameters is shown in (e). To validate our results we also manually labeled plant positions and used their longitude and latitude coordinates to render them at their real positions in our framework (f). This allows us to evaluate the visual quality of synthetically generated plant positions compared to real plant distributions.

Finally, Fig. 5.17 shows the result of generating a scene by carefully fine-tuning the parameters of our PPMs (random, boundary, and cluster strategies) against a real urban environment.

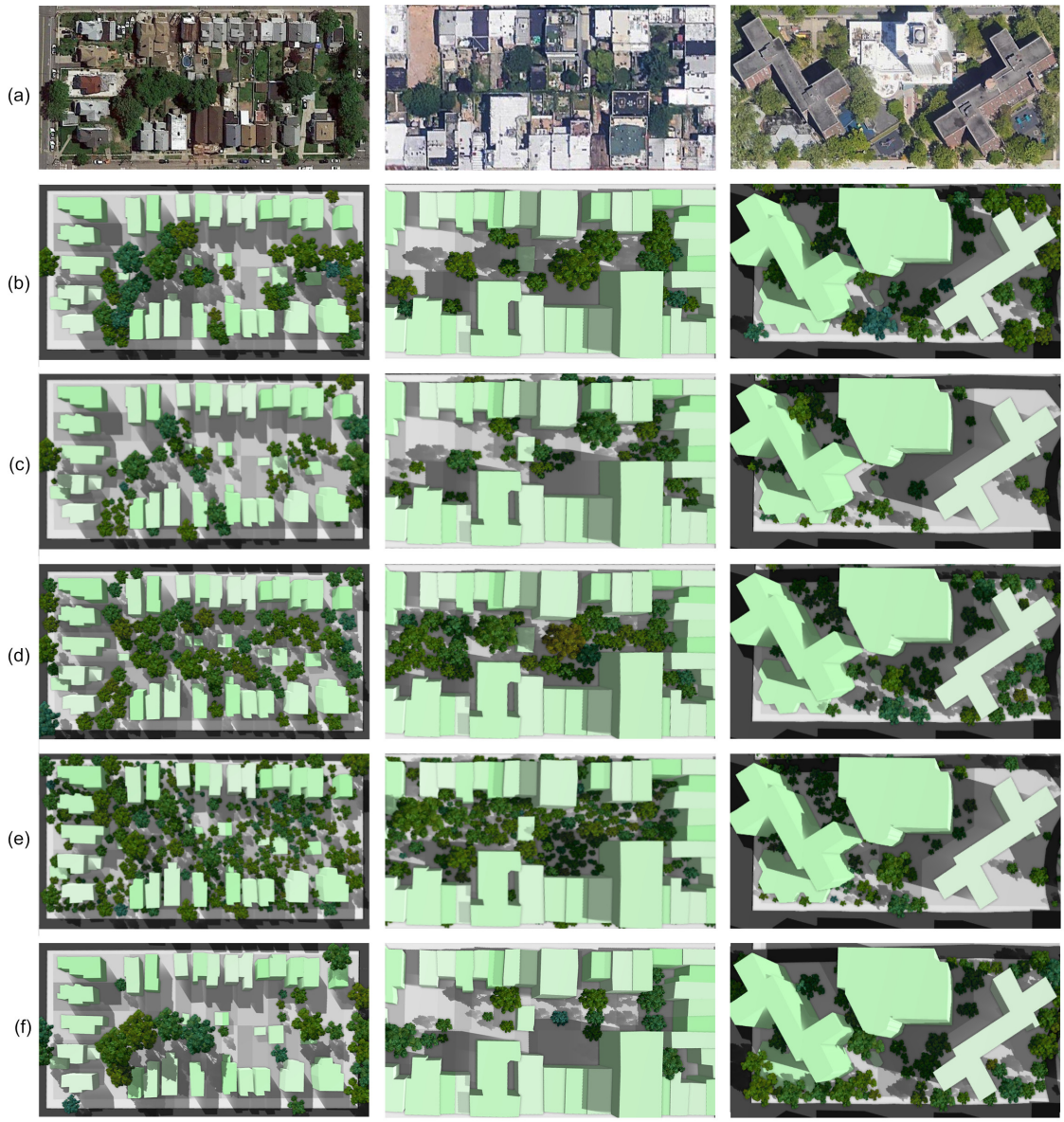


Figure 5.21: We seek to generate plant populations as what can be observed in satellite images of urban scenes (a). By using coverage maps and PPMs (*random* strategy) our method is able to generate highly similar plant populations (b). Additionally, we compare our results to different variations as ablation studies. In (c) we place plants randomly without any parameter regularization (fully random), but we use the coverage maps to define areas where vegetation can be placed. The result of using no coverage map along with our *random* placement strategy is shown in (d). In (e), we show the result of placing trees without coverage map and without any regularization of parameters (fully random). To validate our results we also manually labeled plant positions and used their longitude and latitude coordinates to render them at their real positions in our framework (f). This allows us to evaluate the visual quality of synthetically generated plant positions compared to real plant distributions.

5.7 Evaluation, Discussion, and Limitations

To validate point distributions generated with our placement models, we performed a user study to evaluate the perceived realism of plant distributions generated with our PPMs and real data. Additionally, we asked expert users to compare the usefulness of our modeling approach for authoring plant distributions compared to the manual placement of individual plants. Finally, we measured the distance of generated and ground truth point sets of plant positions.

5.7.1 Perceptual User Study

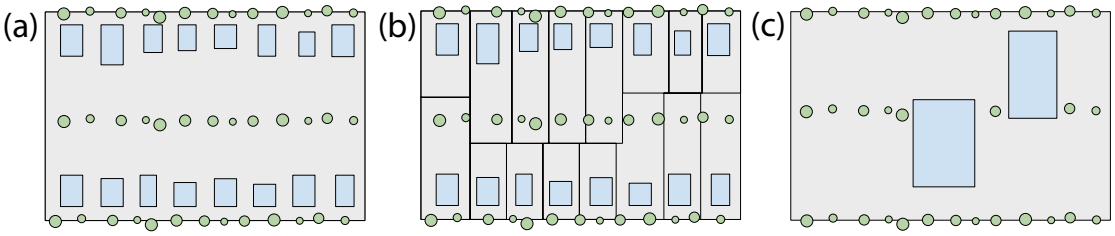
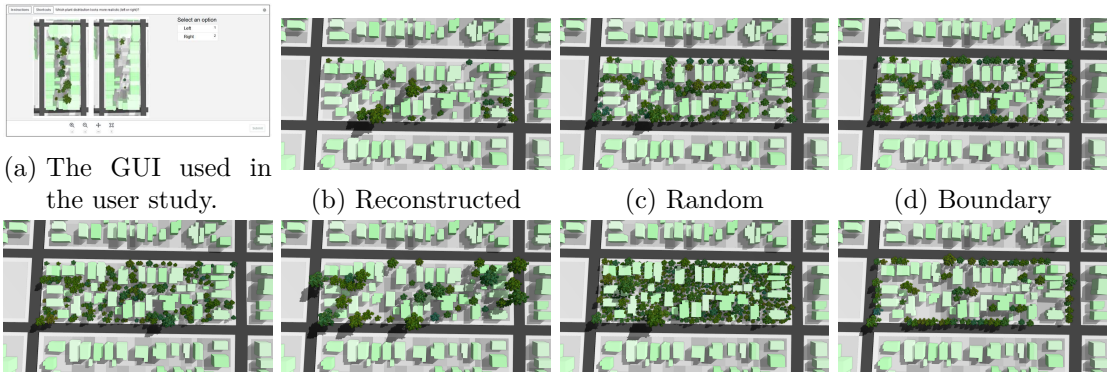


Figure 5.22: The method presented in Benes et al. [Ben+11] places the trees according to procedural rules by using a single strategy for a whole block. Managed ecosystem simulation then makes the trees grow, seed, and die by competition leading to semi-random distributions. Our method works on individual lots, places all plants at once and does not require simulation to populate the urban model. a) With the method of Benes et al. [Ben+11] trees planted at the front and back of the block and along its main axis. b) An overlay of the same block with the actual lots of individual properties. This indicates that the method does not consider lot boundaries. c) When the method of Benes et al. [Ben+11] is used for a single lot plants are placed in an unrealistic manner as buildings and other lot features (e.g. lot shape) are not considered.



(a) The GUI used in the user study. (b) Reconstructed (c) Random (d) Boundary

(e) Cluster (f) Regular (g) Fully random (h) [Ben+11]

Figure 5.23: Examples of images shown to the participants of the user study. Random pairs of images were selected, and the participants were asked which plant populations looks more realistic.

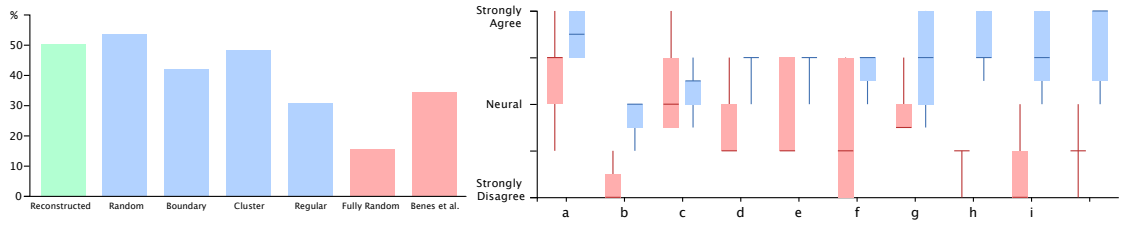


Figure 5.24: Left: the results of a user study. Subjects were asked to select the more realistic vegetation placement based on various strategies compared to real plant distributions. The green bar shows the selection of scenes generated by reconstructing vegetation based on coverage maps, placement based on the PPM strategies (blue), and the baselines fully random placement and [Ben+11] (red). Right: experts' rating of manual plant placement (red bars) compared to using the PPM strategies (blue bars). Questions a-j are listed in Tab. 5.25b.

We generated two sets of images for the user study, one with trees placed by our PPMs and another based on real data. It is difficult to generate images resembling satellite photographs, because of varying lighting, scattering, etc. To avoid this bias and to maintain a similar appearance, we rendered real and synthetic plant distributions by using our framework (see Fig. 5.21f). We identified 30 lots with varying plant placements and produced plant positions using all placement strategies for these lots and rendered them as top-down images using our framework. We generated the real data by manually identifying plants in satellite images of these lots and marked their positions. We then loaded these positions into our framework and rendered them in the same rendering style for both categories. Furthermore, we chose top-down views for evaluating placement strategies, as this allows us to assess the respective distributions of plants.

We then performed a two-alternative force check (2AFC) on the images, for which we generated pairs of images showing real and synthetic plant distributions. The synthetic data shows distributions generated with our placement strategies (random, boundary, cluster, regular), the reconstruction based on the coverage maps (reconstructed), using the planting strategy introduced by [Ben+11], and an entirely random placement (fully random). The *random* placement strategy places plants based on Poisson Disk sampling, which generates random plant positions. However, some parameters for this strategy (*e.g.,* age, plant species, etc.) are not selected randomly. This allows us to create plant distributions with a similar visual appearance. On the other hand, placing plants *fully random* means that all parameter values are sampled fully randomly, which results in incoherent and thus unrealistic plant distributions.

For *fully random* parameters of the PPM are not regularized at all but instead each parameter value is chosen randomly in its defined range, which may result in a very unrealistic appearance (*e.g.,* trees may stand unrealistically close to each other). The methods for plant placement of [Ben+11] are based on blocks and not individual lots. Consequently, these placement strategies assume a specific layout of buildings that cannot

be used for individual lots (see Fig. 5.22) - the majority of areas selected for the user study have a layout the reflects that limitation. In total, we have selected 30 city blocks in New York City of up to 26 lots and populated them with our strategies (see Fig. 5.23). We randomly shuffled their arrangement (left-right) and their order. The image pairs (see Fig. 5.23 a) were shown to 107 users from Mechanical Turk (MT), and we made sure that only MT masters (reliable users) were answering the study. We asked the users, "Which plant distribution looks more realistic (left or right)?" The user had to choose one image. Each PPM category and real data received multiple rankings from every user.

The results of this evaluation are shown in Fig. 5.24 (left plot). The green bar shows the selection of scenes that were generated by reconstructing vegetation based on coverage maps. The blue bars show the selection results of placements with our pipeline, and the red bars shows the result for the baseline [Ben+11] and fully random placement. When comparing the results, the placement with coverage maps and our *random* PPM strategy was selected to be realistic in 50% of the cases. This indicates that our method generates plant distributions that cannot be distinguished from real distributions. Out of our PPM strategies, *random* was selected as the most realistic. In 53% of the cases, it was perceived as more realistic compared to the real placement. The strategy *boundary* was preferred in 44% of the cases, *cluster* placement in 48%, and *regular* was perceived as more realistic in 30% of the cases. The fully random placement – as the lower bound baseline, without any PPM strategy involved – was perceived as realistic only for 16% , the baseline of [Ben+11] was chosen only in 35% of the cases of the shown image pairs.

5.7.2 Usability for Content Creation

(a)

	Question
a)	I think that I would like to use this system frequently.
b)	I found the system unnecessarily complex.
c)	I thought the system was easy to use.
d)	I think the outcome of the result was easy to control.
e)	I was satisfied with the outcome of the result.†
f)	I think the results look convincing/realistic.†
g)	I think it was able to meet the constraints.†
h)	I was satisfied with the outcome of the result.‡
i)	I think the results look convincing/realistic.‡
j)	I think it was able to meet the constraints.‡

(b)

Figure 5.25: a) Five experts were asked to populate this lot based on a predefined set of requirements (See text for more details), and b) to rate their experience between 1 (strongly disagree) to 5 (strongly agree) with respect to the questions above. †: before showing real images; ‡: after showing real images.

To validate our method's effectiveness for content creation, we asked expert users (five 3D artists) to compare our placement strategies to random placement of plants. These

strategies serve as presets to generate a realistic plant placement. In contrast to manually defining the details of every single plant (*e.g.,* the position, species, age, size, etc.), they automatically generate plant positions while respecting structural and positional constraints. We created a simple GUI with two brush tools: one for the manual placement of plants and one for the placement with PPM strategies. When using the manual brush tool, users have to specify the brush radius and the tree parameters. The user then interactively (by clicking with the mouse) generates multiple trees in the radius of the brush with random unconstrained positions. To precisely place a single tree, a small radius with a single mouse click can be used. The second brush tool uses the PPM strategies. Here a user defines the PPM parameters and sketches an area in a lot to automatically place plants with the selected and configured strategy.

We asked five experts to populate a given lot (Fig. 5.25a) based on a predefined set of requirements: (1) ensure that the trees are not too close to the buildings; (2) populate the border of the lot with trees in the red marked area with consistent space and width; (3) populate the orange areas with randomly placed trees – with high density in the right and low density in the left area; (4) populate the green area regularly; and (5) create 2-3 clusters of trees within the blue area. The participants first received a brief introduction to the system and were then tasked to familiarize themselves with the UI without knowing the problem definition. After this introductory phase, the subjects were shown the problem definition and asked to solve the task once with manual placement using the manual brush tool and another time with the PPM brush tool. Whether to start with the manual or the PPM brush tool was changed for each expert. We then asked the experts to rank their experience based on several questions (Fig. 5.25b). We used five-level Likert scale (*Strongly Agree, Agree, Undecided, Disagree, Strongly Disagree.* The results of this assessment are shown in Fig. 5.24 (right plot). For each question (a-j), we show the distribution of answers as box plots for the manual placement (left, red) and the PPM placement (right, blue).

These results show that our method provides an effective means to populate urban scenes with vegetation efficiently, since PPMs were rated as favorable for vegetation placement compared to manual placement. In addition we performed a qualitative study, in which the users commented that while the manual placement provides more control to precisely place plants, it takes much longer to populate larger areas. The users also stated that generating realistic distributions is more difficult with manual placement.

Finally, we asked the experts to automatically place trees by selecting a strategy and a lot, without brushing placement regions. For this setup, the experts unanimously stated that this technique provides less control but allows for fast and realistic vegetation placement in large areas. They further mentioned that the PPM brush tool and the automatic PPM placement produced excellent overall results and were preferred over manual placement to quickly achieve realistic-looking results. They also noted that a combination of manual

and PPM-based placement would be desired when vegetation needs to be placed toward specific objectives.

5.7.3 Discussion and Limitations

Our framework allows us to place and simulate vegetation in urban landscapes. To this end, our focus was on generating convincing distributions of plants for synthetic and real city models. Because defining rules for all possible variations of plants in urban landscapes is intractable, we factorized the problem of placing plants into several placement strategies. Each strategy provides a concise set of rules and parameters to describe the positional and structural properties of vegetation within individual lots. Together, placement strategies and parameters allow us to generate realistic distributions of plants within an urban layout's functional zones. Besides, we use a state-of-the-art developmental model for plants to simulate their environmental response.

We generate distributions of vegetation that resemble what can be observed in satellite imagery; our focus was not on precisely reconstructing every plant of a real environment. While this is arguably important, it would require further analysis (*e.g.,* through deep learning) of satellite images and additional data sources, such as coverage maps. To this end, we think that procedurally generated vegetation can help to generate training data for more advanced analysis pipelines. Compared to manually placing vegetation, our method provides more control and capabilities for the efficient authoring of vegetation placement for city models.

As an alternative to learning parameters with the neural network pipeline from Fig. 5.14, we experimented with learning plant positions with Pix2Pix [Iso+16] in an end-to-end manner. We used satellite images as input and images with plant positions and building geometry as an output for this setup. The goal was to obtain the plant positions from the images in a post-processing step. Training this network was not successful for two reasons: it is challenging to get ground truth data pairs of satellite images and plant positions. While some datasets contain trees' GPS positions, they only store these positions for trees along streets, which is not useful for learning plant positions of an entire city. Second, the results of the network produced were not satisfactory. We suspect that the ground truth images were too sparse (*i.e.,* too few tree positions and building geometry) to provide a meaningful training signal.

A limitation of our current implementation is that we cannot obtain structural parameters with our learning pipeline. Such parameters cannot be learned from coverage maps; learning them from top-down satellite images was unsuccessful. Another limitation of our current approach is that we focus on medium and large trees and do not place smaller plants, such as flowers, bushes, or grass. While fixed models of flowers could be placed with our placement strategies (for example, by using agent-based models [BCS03]), there exists no integrated developmental model that would allow us to develop trees and flowers

jointly. Therefore, we decided only to simulate the growth response of trees to their environment. Furthermore, we do not model plants that are shaped through advanced topiary. More research would be required to explore how pruning affects growth, *e.g.,* for hedges.

Table 5.3: Parameter values we used to generate the figures in this thesis.

Fig.	Strategy	μ	σ	τ	β	κ	π	ω	ψ	η	α	ρ	θ	λ
5.6 a	B	3.13	0.35	1.0	4.0									
5.6 b	B	3.13	0.35	0.8	10.0									
5.6 c	B	3.13	0.35	0.4	10.0									
5.6 d	C	3.13	0.35	1.0		10.0	1							
5.6 e	C	3.13	0.35	1.0		10.0	3							
5.6 f	C	3.13	0.35	1.0		15.0	3							
5.6 g	I	3.13	0.00	1.0				9.0	0.00	30°				
5.6 h	I	3.13	0.00	1.0				9.0	0.30	30°				
5.6 i	I	3.13	0.00	1.0				9.0	0.55	30°				
5.6 j	R	3.13	0.35	0.2										
5.6 k	R	3.13	0.35	0.4										
5.6 l	R	3.13	0.35	1.0										
5.10 a	R	3.4	0.15	0.2							14	0.0	0.0	4
5.10 b	R	3.4	0.15	0.2							20	0.0	0.0	4
5.10 c	R	3.4	0.15	0.2							30	0.0	0.0	4
5.10 d	B	2.2	0.0	0.9	10						20	1.0	0.0	4
5.10 e	B	2.2	0.0	0.9	10						20	0.7	0.0	
5.10 f	B	2.2	0.0	0.9	10						20	0.5	0.0	4
5.10 g	C	3.2	0.25	1.0		10.0	3				16	0.0	0.0	4
5.10 h	C	3.2	0.25	1.0		10.0	3				16	0.1	0.3	4
5.10 i	C	3.2	0.25	1.0		10.0	3				16	0.2	0.4	4
5.20	C	3.5	0.35	1.0		20.0	18				16	0.0	0.0	1

5.8 Summary

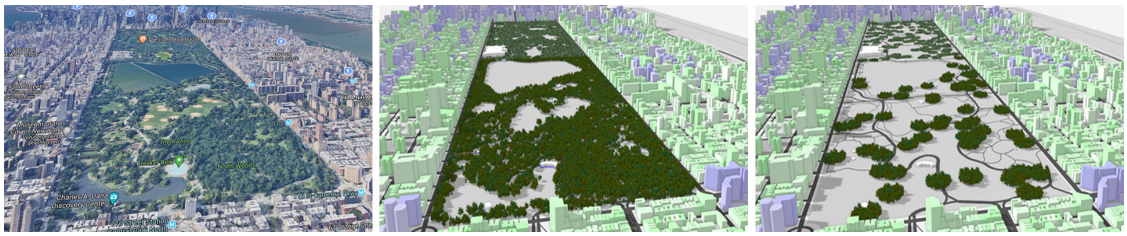


Figure 5.26: Left: Google maps view of New York (Central Park). Our framework generated two variations of plant placements (middle, right) for an initially empty city model. Middle: 54k plant positions were generated in about 60 seconds with a *random* strategy. Right: a different plant population generated with the strategy *cluster* (16k plants).

We have presented a novel framework for populating synthetic and real urban landscapes with vegetation. To this end, we introduced procedural placement models that allow us to generate plant positions realistically and grow individual plants into individual lots jointly. The key idea to our approach is that complex vegetation patterns among different zoning types of a city can be factorized into a set of simple placement rules. A PPM implements these rules and – together with their parameterization – allows to generate complex vegetation patterns with high visual fidelity. Moreover, the PPMs are context-sensitive and read the immediate neighborhood, enabling us to smooth out abrupt changes in placement.

To populate vegetation into real city models, we have used a state-of-the-art style-transfer network to translate satellite images to vegetation coverage maps. These coverage maps allow us to determine the distribution of vegetation within individual lots of a city, which allows us to reconstruct vegetation similar to what can be observed in real data. Instead of reconstructing vegetation at city scale precisely – which is intractable – our goal is to generate convincing and plausible details for reconstructing existing cities or populating entirely new virtual cities with vegetation.

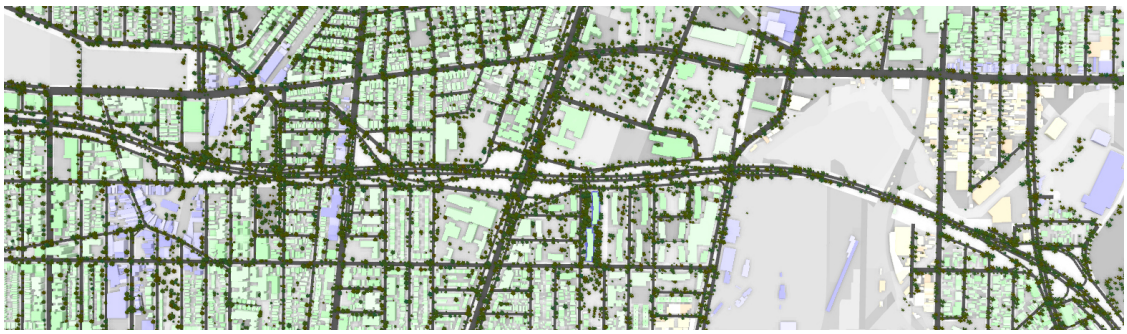


Figure 5.27: Our framework enables to efficiently place vegetation for large urban areas. To reconstruct vegetation for larger urban areas we predict coverage maps and populate the detected areas with our random strategy for each lot.

CHAPTER 6

Conclusion and Future Work

In this thesis, we have introduced new methods and paradigms to improve purposeful abstractions and simplifications of shapes and studied how the simplification of buildings affects the perception and the resulting classification. We also introduced a method for procedural placement models (PPMs) for populating city models with plausible plant distributions and, based on this, a method that allows satellite images to be used to derive the parameters for those models. This chapter summarizes our findings and contributions to these areas and outlines new research questions for future work.

6.1 Abstraction and Simplification of Shapes

6.1.1 Conclusion

As mentioned at the beginning of this thesis, the quality of abstraction is strongly connected to human perception. An abstraction must retain the structures in an object perceived as characteristic by human perception to ensure a correct categorization of these objects. Existing abstraction methods rely primarily on purely geometric properties and do not include essential insights from cognitive science.

Therefore, in chapter 3, we proposed a novel interactive tool built on Gestalt principles for abstracting 3D shapes. The key idea was to develop this method so that the insights from perceptual theory are incorporated as an essential component in the simplification process. For this purpose, we initially employed the well-known Gestalt principles to account for the model's shape, perceptual patterns, and semantics by using structural features such as repetition, regularity, and similarity.

One of the biggest challenges was to transfer the 2D Gestalt rules to 3D space. While in 2D space, symmetries and patterns are easy to distinguish, in 3D space, the spatial arrangement, the viewing angle, and visibility play an additional role. A shape group

formed due to the spatial arrangement of 3D objects can be partially obscured by surrounding objects, whereby this group is no longer perceived as such. At the same time, objects that spatially do not form a Gestalt group can be perceived as such due to the viewing angle. By analyzing visibility, groups that exist because of the spatial arrangement of objects but are obscured by other objects can be simplified to a greater extent without losing information. However, groups that exist not because of the spatial arrangement but because of limited viewpoints may need to be preserved.

In addition to incorporating perception, we have also created a method allowing the user to control the simplification process interactively. Previous techniques have not taken into account the use of sketches to control the abstraction. In contrast, our method allows the user to convey the desired abstractions even for complex 3D models with a few simple strokes to the system. The method calculates this abstraction in real-time, providing instant feedback for efficient work even with complex shapes.

In chapter 4, we further explored how such simplifications affect the classification of objects. We, in particular, explored how abstraction of facades affects the perception of buildings. To derive knowledge on the human's ability to understand semantics from 3D building structures, we presented a user study on the user's comprehension of building categories based on different 3D building representations. Within the study, the users were asked to classify consecutively presented single building representations into the categories *One-Family Building*, *Multi-Family Building*, *Residential Tower*, *Building With Shops*, *Office Buildings* and *Industrial Facility*. During the whole classification process, the users additionally had to rate their level of certainty. The representations shown to the users were untextured LoD3 models, textured meshes/LoD2 models from Google Earth, and images extracted from Google Street View.

Analyses of the user study reveal clear coherences and dependencies between the correctness of classifications and the model representation type. In general, having textural information for buildings is conducive.

The overall classification accuracies for textured meshes/LoD2 models from Google Earth and images from Google Street View are 75.4% and 79.3% and, thus, significantly higher than the classification accuracy of untextured LoD3 models, which lies at 69.2%. Particularly for buildings that belong to somewhat more ambiguous categories such as *Buildings With Shops*, *Office Buildings* and *Multi-Family Buildings*, additional textural information improves the classification results. However, for building categories that are easily separable from the rest like *One-Family Buildings* and *Industrial Facilities*, a geometric representation in the form of a LoD3 model is sufficient in the majority of cases.

The classification accuracy of LoD3 models mainly depends on whether the building models show properties that have been detected as perceptually relevant for the respective building category. Examples for such perceptually relevant geometries and structures are the occurrence of balconies for *Residential Towers*, the different appearance of ground

floor and remaining floors for *Buildings With Shops*, or the high *windows-to-wall-surface* ratio for *Office Buildings*. As these properties are not inherent in all representatives of the categories mentioned, users sometimes experience difficulties distinguishing between *Buildings With Shops*, *Multi-Family*, and *Office Buildings*. Moreover, most users are unaware of their misinterpretations, making perception-adapted building representations even more critical. Therefore, it is crucial to guide the representation based on the significant characteristic features for the respective building category. The knowledge gathered in the investigation of ground truth features and the significant features as perceived or expected by users can then be used to generate virtual 3D models that support and improve the correct perception of building categories.

As a first application, we demonstrated how such knowledge about the human perception of building-related semantic information could be used for the perceptually adapted abstraction of 3D building models using our method presented in chapter 3. The characteristic properties of building structures that turned out to be essential for the recognition of a specific building category are maintained during the abstraction process. In contrast, unimportant or even obstructive structures are simplified to a much greater extent or even totally neglected. By doing so, the recognition of the building category can be preserved even in abstracted building representations.

6.1.2 Future Work

The method presented in chapter 3 and the user study conducted in chapter 4 have shown that perception plays an essential role in the creation of abstract representations because it is the human perceptual system that perceives and judges the final abstraction. We considered perception in our work by conducting user studies to identify structures and patterns commonly used to abstract shapes. We have also introduced quantitative measures to identify parts of a model that follow Gestalt principles. These help us create abstractions that consider the shape, pattern, and basic semantics of the model.

Our future work will expand beyond the purely pattern-based aspects and explore perception-based abstractions in more detail, continuing the findings from chapter 4. While Gestalt rules lay an initial foundation for how the simplification must be applied, they lack additional semantics information. For semantics-preserving simplifications, the method must be extended to take the context into account. The perceptual knowledge gained from such analyses will not only allow maintaining perceptually relevant geometric properties and structures as was the case in the perception-guided abstraction. It will also allow modifying, emphasizing, adding, or removing perceptually relevant structures in a targeted manner to automatically generate models that can be classified more easily.

Semantic aware abstraction

Which details of an object are essential to classify it correctly depends on the information that has to be conveyed or is relevant to our decision-making process. A single tree or an entire forest can be represented with only a few details and still be perceived as such, but information such as species or age can be lost or distorted. Facades of houses can be represented by simple geometric shapes and rows of windows by single proxy objects without losing the information that it is a building, but it can prevent the correct assignment to a building category. It is therefore essential to identify which semantic information exists for an object and how they are represented. Objects can also differ in their structural features, which can further contain semantic information. While natural objects tend to have organic shapes, artificial objects such as buildings or machines often have more geometric patterns.

Possible future work could therefore address how this information can be identified through semantic categorization and explore how incorporating these findings can be used to preserve essential information needed for a correct classification while further simplifying structures that contain information not needed for a particular use case. For reliable semantic segmentation, it is necessary to determine which details and patterns within an object represent a particular semantic category and to develop a method for automatically assigning these details and patterns to the corresponding categories. However, there are many such categories, and we cannot investigate all of those at once; consequently, the first step would be to limit ourselves to a specific use case. Since we have already addressed the impact of simplification of buildings on their classification into building categories in chapter 4 and the importance of correct plant distributions in chapter 5, it is reasonable to explore these topics further.

A simplification of an object often is performed in the context of an entire scene, so it is reasonable to define semantic categories that apply not only to a single object type but are capable of describing an entire scene. Therefore, categories must be found which can be applied to multiple object types. Semantic categories such as zones (residential, commercial, industrial, or mixed usage), geographic location, property value, or maintenance would be conceivable when describing urban environments. These must be specified in more detail using suitable parameters, such as the level of maintenance or the geographical region to which an object is assigned. In the following step, details within an object essential to these categories have to be identified.

Based on this, a method has to be developed that recognizes the semantic categories contained in an object and determines the respective parameter value. Such a method enables further investigation of how simplification affects the parameter values and the categorization by a viewer. We could use this to extend our method allowing the user to apply purposeful simplification by specifying which semantic categories must be preserved

and to what extent. The values of the parameters could be used as a guideline for this process.

Context Aware Abstraction and Simplification

We have only considered the objects on an individual basis so far. However, objects are rarely perceived in an isolated manner but rather in the context of other objects or in a specific use case.

Use Cases. While in the field of children's toys, a rough categorization is usually sufficient and the proportions of the objects are less important, a correct representation is even more critical for realistic models such as in city planning or the field of photorealistic image generation. Therefore, it is essential to know in which use case the simplification will be applied, as this significantly impacts which details are relevant and must be retained.

As a future work we want to identify such use cases and determine which information is essential. The semantic categories can be used for this purpose and be assigned to these use cases with different levels of importance. The creation of different LoD representations can be viewed as a specialized use case that does not exist independently but instead extends another use case. While the use case itself defines what information is essential, the used LoD method further defines the importance of that information for the different LoD representations. In this context, it must be investigated whether the importance of information for LoD representations with fewer details is always lower compared to a representation with more details or whether certain semantic information has greater importance for LoD representations with few details than for LoD representations with many details. It is reasonable to assume that LoD representations with a low level of detail will lack various smaller structures, resulting in other, more prominent structures becoming significantly more important. For house facades, at lower LoD representation, information such as the contents of windows, labels on those windows, or on the facade that allows conclusions to be drawn about the category of the house may no longer be present or identifiable, making other features that would still allow proper categorization more critical.

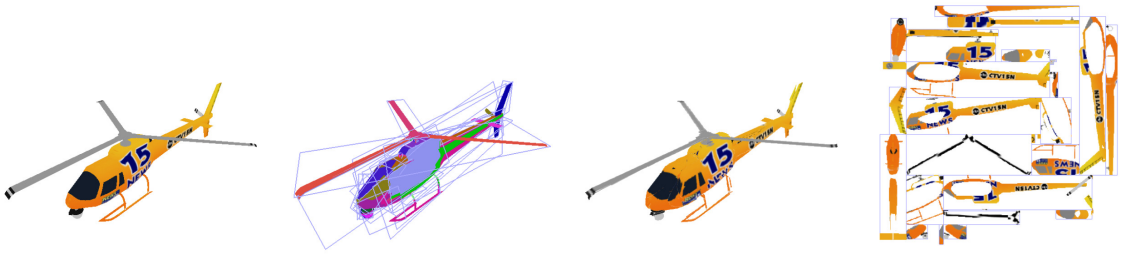
Spatial Context. Besides the use case, the spatial context in which the object is perceived also plays a significant role. Other objects in spatial proximity to the object in the viewer's focus provide additional information that can help to classify the object correctly. This additional context can significantly impact how an object is perceived or what details are deemed necessary for classification.

To include this spatial context in the simplification process, the object's environment must be further analyzed. Semantic categories can also help in this regard to gain a deeper insight into the information available in a scene and possible relationships between objects.

One conceivable way to determine these relationships would be to cluster the objects based on such semantic categories. We, therefore, want to explore an approach that would determine individual spatial clusters for each semantic category based on the values of the objects' category parameters. These clusters could identify which objects provide mutual categorization based on similar semantic information and spatial proximity. The clusters could also determine which information is critical to certain areas and must be retained. Additionally, outliers could be identified, indicating that a particular object stands out from its surroundings and can therefore be considered unique, and its expression must therefore be preserved. This additional information can enable an object to be simplified even more aggressively if the information provided from the context ensures that the classification can still be performed correctly or be used to prevent specific simplifications so that required classifications are retained.

Replacement of Geometry with Textures

Our method so far only addresses a geometry-based approach to simplification. Besides geometry-based simplification to reduce the complexity of 3D models, another standard method in computer graphics is to replace complex geometry with textures, e.g., books on a shelf or the complex foliage of a tree. The idea is that 2D textures can represent the 3D geometry of a whole object or parts of it depending on the LoD level. Since textures can be rendered quickly and independently of model complexity, this increases rendering speed. A standard method for this is the use of billboard clouds [Déc+03] (See Fig. 6.1) that replace complex geometry with a set of spatial oriented and textured planes to approximate the original 3D-geometry.



(a) (b) (c) (d)

Figure 6.1: Example of a billboard cloud from [Déc+03] (Figure 1): (a) Original model (5,138 polygons) (b) false-color rendering using one color per billboard to show the faces that were grouped on each (c) View of the 32 textured billboards (d) the billboards side by side

Another standard method for reducing complex geometry is to use displacement maps to represent complex surface descriptions, such as roughness or cracks. Both methods have in common that geometric details of an object are replaced with textures without changing the visually perceived spatial extent of the object. How much a detail contributes to an objects' spatial extent depends not only on the geometry itself but also on

the viewing angle. Thus, a texture may be correct from a single viewing angle but lead to visual inconsistencies, such as misalignment with adjacent geometry, when the same texture is used for a different viewing angle.

While those methods already produce stunning and plausible results, we believe those methods could be further improved by incorporating our research findings. Our method performs comprehensive analyses of the visibility and relationship of object parts. The information obtained from these analyses can be combined with texture-based approaches to more efficiently identify parts of an object that textures can replace. The information about their visibility could be used to determine which details are candidates to be replaced by textures and minimize the number of textures required to represent the details from different viewpoints correctly. The information about their visibility could also determine which details can be replaced by textures for which LoD representations. In addition, the information about grouping obtained by the Gestalt principles can be used to determine which details are suitable to be combined for a replacement by textures.

In the area of house facades, this could be used in combination with the previously proposed semantic analysis to enable an even more significant simplification of the geometry while at the same time preserving the correct classification.

6.2 Procedural Modelling

6.2.1 Conclusion

As mentioned at the beginning of this thesis, the quality of procedural plant distributions depends on providing results that match real-world distributions as closely as possible. Besides the purely visual aspect, where we judge the plausibility of the distributions based on our visual experience, their distributions also contain information from which we can draw further conclusions about their environment.

Essential visual aspects are phenomena such as *crown shyness* [GPO09] or other biological and physical rules that influence the distance of plants to each other or other objects. Besides visual aspects given by botanical and physical rules; there are also rules given by plant placement created by humans. These include avoiding conflict with windows, doors, or paths, fulfilling functions such as visual or sound barriers, or purely aesthetic purposes. Considering that these rules are ubiquitous in most cases regardless of region, they provide an important visual cue for assessing the plausibility of plant distribution. In addition to these rules, other factors influence the distribution. These include the degree of maintenance by humans, the region, or the type of land use that influences the occurrence and frequency of plant species. For example, a high occurrence of a particular species allows conclusions about the climatic region for the shown scenery. In previous

methods, distributions of plants were done either by simulations, manual placement, and randomly, or only by simple, rudimentary patterns.

So far, however, no method formalized these rules to be used in a procedural model. Therefore, in chapter 5, we have presented a method of formalizing these planting rules. Furthermore, we have presented a machine learning model, which allows for determining the parameter values for a planting strategy based on satellite images. For this purpose, we first trained a style transfer network, which translates satellite images into coverage maps, and then defined and trained a network to determine the strategy's parameter values for those coverage maps. Moreover, our proposed framework calculates the distribution for the lots in real-time, allowing immediate feedback for efficient parameter adjustment. We, therefore, created an interactive authoring tool that allows changing of strategies and their parameters in real-time and allows controlling of the area where the strategies are applied using a brush tool.

We also introduced a quantitative measure to evaluate the quality of distributions and compared it with actual distributions. We have demonstrated our system's effectiveness through two perceptual user studies and one study using our interactive authoring tool.

One of the biggest challenges was to identify the most common planting strategies and define PPMs for those that have parameters that are intuitive to control, reflecting all the essential aspects of plant distributions for each strategy, and that these PPMs are defined in such a way that their parameter values can be derived from actual data through machine learning. Appropriate parameter descriptions are also crucial because, besides efficiently generating plant distributions, the method should include information that can be used with plant growth models to reproduce plant-to-plant or plant-to-object interactions. Additionally, the method should be defined so that the strategies used can be extended without changing the way the method itself works.

6.2.2 Futurework

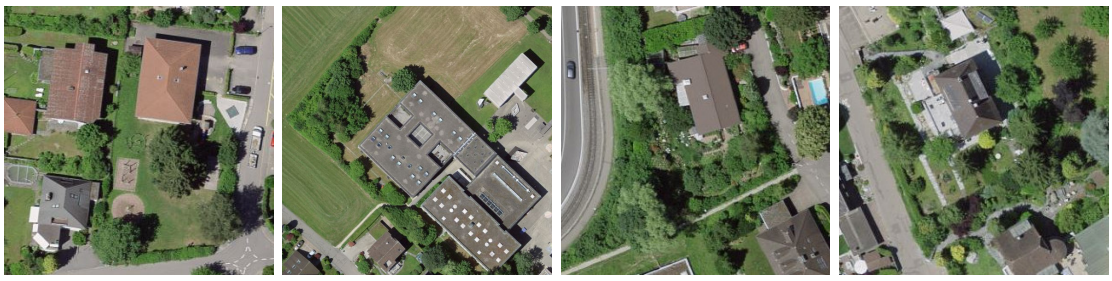
The original idea was to populate city models so that the resulting plant distribution corresponds to a particular region. Regions can be defined by climate zones, wealth, type of use (residential, commercial, office, and industrial), and other categories. For this purpose, we wanted to find a method that allows us to derive typical distributions from actual data and transfer them to an empty city model but at the same time offers the user the possibility to control this process as much as possible and to make adjustments afterward. Therefore, a method is needed to perform this adjustment in a targeted and predictable way. A procedural model appeared to be a suitable option. However, since there was no formalization of such a procedural model for plant distribution, we decided to first focus on creating that formalization in chapter 5. A well-thought-out formalization would then make it possible to extend this method with additional capabilities.

We see several avenues for future work, such as exploring enhanced placement strategies to capture more of the variation of vegetation placements observed in real cities, which will be discussed in the following sections.

Extending the Placement Strategies

In the presented method, the most common distributions of plants were formalized as strategies. As shown in Figure 6.2, these strategies cannot directly represent all existing distributions. While our reconstruction method can generate these distributions using a coverage map by restricting the area in which a strategy is applied, the strategies themselves cannot directly generate these distributions. However, this is not limitation of the method itself, as other strategies can extend it. Nevertheless, the question arises whether it is indispensable to add another strategy or to extend an existing strategy for each new distribution, or whether this must be achieved by extending the way the method itself is working. The significant advantage of these simple strategies is that the parameter set of the individual strategies is clear, and the result of these strategies is easier to control. Adding additional parameters to these strategies can affect their controllability and be problematic because the intended use case may no longer clearly defined.

The current formalization of the method only allows applying one planting strategy for a defined area. However, our interactive modeling tool technically allows applying multiple strategies to the same area while retaining the plants added in a previous step and still fulfilling the constraints such as the distance between plants, buildings, or the selected species variation. Such a combination of strategies for the same area already allows for reproducing a significantly larger variety of distribution with the existing set of strategies. To include such a combination into the formalization of our method, we want to investigate different possibilities.



(a) (b) (c) (d)

Figure 6.2: Lots with plant distributions that are currently not covered by our method but are combinations of existing strategies. The lots a) and b) are surrounded by plants (*border* strategy) and have additional plants within the lot area (*cluster* and *single* strategy). c) and d) are lots surrounded by plants (*border* strategy) and a large amount of plants distributed within the lot (*random* strategy) (Source: Bing Maps)

Subdivision of the Area. Without significant changes to the method and its formalization, the area in which the strategies have to be applied could be further subdivided. The advantage of this approach is that it does not change the existing method itself. It would only require an additional step that further subdivides the area of a lot and assigns the corresponding strategies to these new areas. However, a significant disadvantage is that for each area, only a single strategy can still be applied. Additionally, the subdivision algorithm must decide which strategy to apply where and requires further detailed information of the strategies to generate shapes and sizes for those areas meeting the requirements of the corresponding strategy.

Layered Strategies. Our brush base modeling tool shows that the method already allows us to continue applying strategies to the same area without removing the already placed trees while maintaining the spacing rules between the trees. Therefore, it is reasonable to add a layered approach to the formalization of the method. However, the question arises in which order the different strategies should be applied to the same area. For this purpose, different levels of importance could be assigned to the strategies, reflecting the order in which these strategies are typically applied by landscapers when populating a lot. A possible order could be: First, *equidistant* for street trees, then the *boundary* strategy for hedges or trees along the lot boundary, followed by the other strategies such as *cluster*, *random*, or *regular* distribution for the inner area of the lot. While simply combining a few basic strategies allows this method to be as flexible as possible, it will presumably make learning the parameters and the strategies used much more difficult. However, since this method is promising, we would like to further investigate this approach in future work.

Extending our Learning Approach

It seems a promising direction for future research to investigate further how neural networks can be generalized for more diverse urban data to learn parameters for scene generation. Currently, machine learning is only used for the reconstruction process to infer the distribution of plants from satellite imagery by determining the parameters' values that reproduce the distribution using a specific strategy.

However, machine learning offers even more possibilities. Using a city model for which the plant locations are known, in combination with the existing zones and additional information such as neighborhood, land value, and region, we could learn which planting strategies to use for this combination of input information and which parameter values to apply to these strategies. Such a network could then generate proper plant positions for empty city models based on that meta-information.

To achieve this, we would first extend the existing learning method to determine the most suitable planting strategy for a given lot. Furthermore, we would investigate how addi-

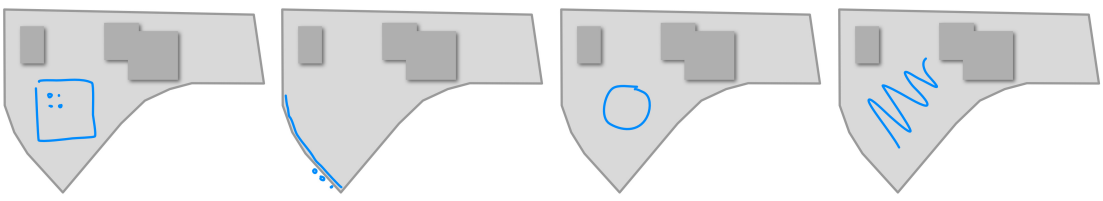
tional meta-information such as land value, geographic region, zoning, and neighborhood information can be included in the learning process.

Such a method would not only enable a city to be automatically filled with vegetation according to its meta-information but would also allow the artist to purposefully change the appearance of an area afterward, not through the parameters of the PPMs, but based on the desired appearance. In this case, the artist would no longer change the actual parameters of the planting strategies but the parameters of the meta-information, from which the learned model then derives the parameters of the PPMs. While end-to-end learning methods could also achieve this to some extent, we believe that PPMs would provide better control over the constraints that must be satisfied for a plausible distribution of plants.

Weighted and Sketch-based Interactive Modeling

Our interactive modeling tool currently only supports the possibility to paint the areas in which the strategies should be applied. However, it does not allow the user much control over which parts within this area should be planted more densely or where clusters should be prioritized. In addition to simply sketching the areas, one could therefore imagine a pressure-sensitive brush. The areas that should be planted more densely would be drawn with more pressure than those that should be filled with less density.

Another extension of the method would be to not manually select the planting strategy but to adopt the approaches from chapter 3 for a sketch-based interface so that the user can specify the planting strategies to be used through simple sketches. For that purpose, each strategy could be described by a set of easy-to-use strokes (see Figure 6.3). Combining these techniques, the artist could first indicate the desired density and then sketch the desired strategies into those areas. As in our other sketch-based approach, the user could then refine the placement predicted by the system using the already existing UI while getting real-time feedback on the updated placements.



(a) (b) (c) (d)

Figure 6.3: This figure shows how sketches could be used to define the planting strategies. a) *regular* could be represented by a box and an initial set of points defining the distances c) *equidistant* using a line with an initial set of points indicating the distances b) *cluster* by a circle indicating the desired radius, and d) *random* placement through a zigzag line.

The manual filling of a city with plausible distribution is a time-consuming process. Considering that the distribution of plants in lots of the same district is usually similar, we could imagine a method allowing the artists to transfer the distributions they have already defined for individual lots to a larger area. In chapter 3, we have already presented a method where the artist can highlight already performed simplifications and transfer them to other areas. The same would be conceivable for our method of distributing plants. We can imagine that this could be used both for a straightforward transfer of the strategies used, but it could also be combined with our existing machine learning method or with the extension we suggested for that in our future work.

Context-sensitivity

The placement of vegetation depends on many factors and is influenced by its surroundings. Urban planners and owners of private properties use vegetation to shape an area's appearance and fulfill specific requirements. Their concepts mainly influence the species used and their distribution. Besides that, vegetation distributes itself based on ecological factors. In section 5.4.5, we have already addressed this subject by including a building envelope, distances between trees, plant growth, and the parameter values of neighboring lots. As shown in Fig. 5.11, the parameters used for one lot are adjusted with those of neighboring lots. However, the description of the building envelopes and rules for adjusting the parameters based on the neighboring lots are still very rudimentary, as the method currently uses only a simple approach to contextual parameter fitting. Nevertheless, the results show that the method is already capable of taking such factors into account.

However, it would be interesting to explore further how the context influences vegetation and incorporate it in our method.

Maintenance. One of several factors affecting the distribution of plants depends on whether and to what extent they are maintained and shaped by humans. While vegetation left to itself results in a self-balancing system, vegetation maintained by humans is characterized by the intended function. Therefore, a first step would be to determine how much humans influence the vegetation in a given area. However, this cannot be described by a simple parameter because different types of maintenance have different characteristics.

Maintenance can be roughly divided into the following categories:

- **Not maintained.** Can be found in designated nature reserves, abandoned regions, or other regions completely left to themselves. There, the vegetation keeps itself balanced and grows fully based on natural patterns.

(a) (b) (c) (d)

Figure 6.4: Areas with a different degree of maintenance: a) Nature reserves are usually left entirely to themselves (*not maintained*) b) Most forests are under forestry use, which means that the occurrence of trees is usually planned and maintained. However, there is no excessive maintenance to provide a habitat for animals. (*semi-maintained*). c) In areas around power lines, trees are regularly pruned to avoid interference (*maintained*). d) For agriculturally used areas, the vegetation is controlled significantly. (*planned*) (Sources: a) www.pronatura.ch c) wikimedia.org d) Google Maps)

- **Semi-maintained.** Most of the natural regions that are located near populated areas are maintained to a certain degree. Such *regions* are forests, unutilized larger meadows, riparian areas, and similar areas. In these regions, the landscape and vegetation are maintained mainly to prevent overgrowth. However, an attempt is made not to interfere too much with the ecological balance.

- **Maintained.** To ensure the functionality of critical infrastructure, vegetation is heavily controlled in certain regions. Plants in these areas are trimmed regularly not to damage or interfere with these infrastructures. These include paths, roads, railways, air traffic, power lines, and more. In these regions, the maintenance is mostly about limiting growth and less about controlling species that predominate there. However, the possible usage of herbicide and the rigorous mowing and pruning of plants strongly impact the dominant species.

- **Planned.** This can be seen in gardening, agriculture, and private properties on which the owner heavily controls what kind of species will exist in the area. Planned planting, as in agriculture, can also be strongly influenced by regional regulations. Such regulations could preclude the excessive use of fertilizers or pesticides and, for example, force farmers to plant fields with alternating vegetation. These, in turn, influences the appearance of rural areas.

In addition to the categories described, there may be several other special cases, such as a completely newly developed area for which all vegetation is initially replanted and thus planned but where individual areas are later left to themselves as much as possible.

The categories shown are not fixed but can transition into each other. A property with well-maintained gardening could become abandoned and thus transition into an unmaintained region. Additionally, regions could be maintained only at a specific time of year

as is the case, for example, for summer residences. For these, the maintenance might be reduced or suspended at the remaining periods, which would then exhibit the characteristics of an unmaintained area during this time.

One way to address this might be to populate an area with the initial maintenance level and then gradually transition to the target level using a simulation. Doing this for many lots can be a time-consuming task, and we thus would like to find a method to represent this in our procedural model. However, we cannot just blend those maintenance levels with a simple function but must consider that the transition from one maintenance level to the other is not equally fast in both directions. While overgrowth depends on the rates of plant growth, the opposite direction depends mainly on human resources.

Functionality is besides maintenance another essential role that influences the distribution of vegetation. Our method currently covers minimum distances to buildings and other trees and has a rudimentary approach to keep areas around windows and doors free from vegetation. We want to investigate additional functions to extend the methods present in section 5.4.5 to a more complex model.

Therefore, we want to closely exaine what functionalities exist and how city planners and landowners utilize plants to fulfill these functionalities. In the following, we show a list of functionalities, we would like to include in future research.

- **Air Quality.** Improving the air quality in cities is an important topic in city planning. While this is inevitable by reducing the root cause of air pollution in the long run, it is nevertheless important to support this by beneficial planting in the cities. However, the effect of vegetation on air quality is a controversial topic with a seemingly contradictory conclusion. Vos et al. [Vos+13] refers to it as the *green paradox*. The conclusion hereby is that while dense rows of trees along roads are not well-suited to improve air quality due to the resulting lower ventilation, according to this study, there is no reason why more vegetation should not be planted in areas with high air pollution, but care must be taken that ventilation is not impaired.

- **Temperature Control.** Due to reduced ventilation, asphalt, traffic, air conditions, and other factors, the temperature in cities is often higher than in the surrounding rural areas. High temperatures reduce the quality of life in cities and harm the durability of roads. Asphalt can heat up dramatically in the summer, and heavy traffic can wear down the roads rapidly. Speak et al. [Spe+20] investigated in their paper how the tree traits of different species affect the urban ground shade cooling. Moreover, the optimal planting of trees can significantly impact the temperature exchange. Richards et al. [Ric+20] have investigated in their paper how different species of plants affect the temperature in cities. Some plants were found to be better suited to reduce the temperature, while others are more disadvantageous regarding that.

- **Noise Protection.** Noise pollution from traffic as well as commercial and industrial buildings is a persistent problem in large cities. While in the area of highways or industrial sites, there is the possibility of counteracting this by noise barriers; these have the disadvantage of not being aesthetically pleasing and can only be applied limitedly. Therefore, other ways are used to reduce noise pollution originating from commercial buildings or urban streets. Here, the use of plants is an effective option, as their uneven surface partially absorbs sound, which can significantly reduce noise pollution. However, not all plants are equally suitable for this task. Fan et al. [Fan+10] have investigated the relationship between tree characteristics - especially the properties of their leaves - and their ability to absorb sound.

- **Erosion Prevention.** Heavy deforestation and vegetation removal can reduce the stability of the soil. This reduced stability can lead to the undercutting of roads or landslides on slopes during heavy rain. It can also cause trees to lack support in stronger winds, making them easier to uproot and causing damage to nearby infrastructure. These factors are essential for urban planners in urban and vegetation planning, especially for cities built near hills and mountains.

- **Visual Protection.** Visual protection is another factor that influences the choice of planting. In the case of private properties, the main reason for this is privacy. The extent to which plants are used as privacy screens varies greatly from region to region. Plants are also used for aesthetic purposes to obscure nearby commercial or industrial properties, for example. However, commercial property owners, such as industrial companies, also use plants as a visual barrier for increasing public acceptance, among other reasons.

- **Aesthetics.** In addition to the purely functional use of plants, aesthetics play a significant role. Plants are used on all kinds of properties to improve their visual appearance. While this has partially already been covered by *Visual Protection*, there it has only been described that plants are used in general to obscure the view in visually less pleasant areas. However, this is a broader topic dealing with the selection of plant species and their arrangement on a lot, and it is probably the most difficult to describe because it is a very subjective topic, strongly influenced by the region and the time.

These are by no means all imaginable functions, but in the field of urban planting, they are probably among the most important ones we want to deal with first. In our future work, we would like to extend our method so that vegetation is planted according to those factors. *Visual Protection* and *Aesthetics* and *Erosion Prevention* can be handled mostly independent from each other. *Air Quality*, *Temperature Control* and *Noise Protection* however heavily depend on each other, and therefore a method needs to be found to combine those.

Since we have already addressed the influence of wind on the growth of plants in our previous work [Pir+14] and implemented a wind simulation model based on *Smooth Particle Hydrodynamics* for this purpose, it is reasonable to consider the areas of *Air Quality*, *Temperature Control* and *Noise Protection* as a possible future work topic. For this purpose, our method has to be extended either to incorporate pollution information based on real-world data or to predict those values based on the surrounding. For a ventilation simulation, an airflow model is required to predict wind and temperature exchange behavior.

On the first look, air quality and temperature control might be conflicting because the effects of heat and wear down of streets and the suggestion not to plant dense rows of trees along streets with heavy traffic might contradict. However, temperature differences could create a constant airflow throughout the cities that reduce the temperature. It would be interesting to investigate whether taller trees that cast more shade but do not retain too much air would be beneficial.

Even though the final goal in our method is to reproduce these effects without a simulation model, an accurate wind model is needed to evaluate our method for its effectiveness. Even though we have shown in our previous work that *Smooth Particle Hydrodynamics* is an efficient method to simulate wind flows, we should look at further methods to investigate which of these methods gives the most reliable results. Therefore we would look at the method of Kurppa et al. [Kur+18] which uses that large-eddy simulation LES model PALM to simulate airflow and ventilation in cities, and the method of Vardoulakis et al. [Var+03] who are using parametric models, wind tunnel simulations, and field measurements for this purpose. We would compare these methods for efficiency and accuracy to check which can be used as a wind model for our system or whether a combination of those is necessary.

Advanced Interpolation of Parameters and Strategies

Our method already enables changing of most of the parameters such as density, size, and distance between trees and buildings without changing the positioning of the trees significantly and thus ensures that such adjustment leads to predictable changes. In Figure 5.11 we have already shown that we can match the appearance of neighboring lots by harmonizing their parameters.

However, this method works only to a limited extent when different planting strategies are used between those lots. The same problem arises when different states within the same lot, such as well-maintained or less-maintained, have to be interpolated because the strategy for these two states may differ. An implementation of layered strategies, as described earlier, could be a first step to address this problem. The further problem is that plant positions are regenerated from scratch when the parameters are changed. Even though we have ensured that our sampling always produces the same plant position for a

given lot, it is still possible that when new plant positions are added by one strategy, the placement of plants through another strategy is blocked due to the distance rules that must be followed.

This possible inconsistency is especially problematic if the changes within a lot should be temporally coherent. Applications where temporal coherence is essential could be time-lapse animations for games or movies. We would investigate how to blend strategies and preserve existing plant position when changing parameters in future work.

Optimized city planning and quality measurments

Building cities in such a way that the quality of life increases is an ongoing research topic. Yang et al. [Yan+19] have used performance-based planning to assess the impact of urban building morphology on local climate surface temperatures under different wind conditions. Their research showed that urban architectural patterns were one of the most important drivers of climate change. However, incorporating these findings into city planning is a complicated task, as seen in our other proposals for future work. Once we have incorporated these into our methods, it is an excellent opportunity to explore how such a system can optimize the planning of a city.

In addition to learning planting patterns based on parameters such as land value, learning could also be used to optimize the planting of vegetation in cities to achieve specific goals. Such goals could be better air quality, optimization of temperature, and ventilation. Especially for the construction of green cities, where minimization of particulate matter and energy optimization is the goal, this method could significantly support the planning of such a city. Besides, this method could be used to improve the quality of life further. It could be used to distribute parks and other green spaces optimally so that residents do not have a long travel distance to get there, have sufficient size for the expected number of visitors, and place trees to target noise pollution or ideally prevent it altogether. It could also be used for existing cities to solve a specific problem, such as transforming those to green cities, or giving a quality measure for how well those already fulfill these requirements or how they can be optimized.

Including a proper simulation of wind, air pollution, temperature development, and noise expansion allows the study of how different planting strategies and different choices of parameters affect these factors. A suitable neural network could then learn which parameters and strategies to choose for optimal planting.

Besides choosing a suitable learning model, the biggest challenge here will be efficient training data generation. An accurate simulation could be time-consuming and thus represent a bottleneck in the generation of the training data. Therefore, how large the area under consideration must be to obtain reliable results must first be studied. Furthermore, a purely random selection of parameters and strategies for each lot and all

(a) Banbury (UK) (b) Luzern (CH) (c) Portland (USA) (d) SanMarion (I)

Figure 6.5: Google Street View images showing small vegetation from different regions.

streets in the area under investigation would lead to an immense number of combination possibilities. To reduce the number of possible combinations, we have to investigate to what extent the parameter space and the possible combinations of strategies can be restricted.

Small Vegetation

In our method, we so far have limited ourselves to the placement of trees and bushes and have not considered small vegetation. However, this vegetation plays an equally important role, as it allows further conclusions to be drawn about an area, especially for detailed views. In the case of private properties and parks, it allows one to draw conclusions about their wellness and maintenance and distinguish how an area is used.

This type of vegetation can occur in various forms, including undergrowth in forests and parks, wild vegetation along roads or railroads, planned vegetation in gardens and parks, and mixed forms. (See fig. 6.5)

In future work, we would first look at planned vegetation in parks and gardens. This vegetation includes, for the most part, flowers, grasses, and shrubs. These are usually planted along paths or defined areas and often serve a purely decorative purpose. Additionally, smaller plants such as flowers can be even more affected by the climate than larger plants. Their decorative purpose also results in changes regarding spacing rules and locations near windows.

To explore these further, we would first look at the rules for the layout of parks and gardens. General descriptions of various garden styles can be found in books such as Adam's Gardens Through History [Ada91]. Gardens here can be categorized broadly into formal and non-formal gardens. Formal gardens in the European, Middle Eastern, and Near Eastern traditions have paths and ponds laid out in regular grids, radial arrangements, symmetrical shapes, or a combination of these. In contrast, paths in non-formal parks are randomly shaped, follow the contours of the terrain, and are not straight. The layout also influences the way small vegetation is distributed. While more accurate planting can be

found in formal gardens, in non-formal gardens also, at first sight, more random-looking distributions are plausible.

Therefore, in our future work, we would like to investigate the different scenarios for these small vegetations, how their distributions can be formalized, and how this can be incorporated into our method.

List of Figures

Bibliography

[Ada09] N. Adabala. „A technique for building representation in oblique view maps of modern urban areas“. In: *The Cartographic Journal* 46.2 (2009), pp. 104–114 (see p. 24).

[Ada91] W. H. Adams. *Gardens through history: Nature perfected*. Abbeville Press New York, 1991 (see p. 124).

[AFS06a] M. Attene, B. Falcidieno, and M. Spagnuolo. „Hierarchical Mesh Segmentation Based on Fitting Primitives“. In: *Vis. Comput.* 22.3 (Mar. 2006), pp. 181–193 (see p. 21).

[AFS06b] M. Attene, B. Falcidieno, and M. Spagnuolo. „Hierarchical mesh segmentation based on fitting primitives“. In: *The Visual Computer* 22.3 (2006), pp. 181–193 (see p. 18).

[AFS06c] M. Attene, B. Falcidieno, and M. Spagnuolo. „Hierarchical mesh segmentation based on fitting primitives“. In: *The Visual Computer* 22 (Mar. 2006), pp. 181–193. doi: 10.1007/s00371-006-0375-x (see p. 18).

[Aga+11] S. Agarwal et al. „Building rome in a day“. In: *Communications of the ACM* 54.10 (2011), pp. 105–112 (see p. 26).

[Ahm+17] A. G. M. Ahmed et al. „An adaptive point sampler on a regular lattice“. In: *ACM Transactions on Graphics : TOG* 36.4 (2017). Article Number: 138. doi: 10.1145/3072959.3073588 (see p. 14).

[AK84] M. Aono and T. Kunii. „Botanical Tree Image Generation“. In: *IEEE Comput. Graph. Appl.* 4(5) (1984), pp. 10–34 (see p. 28).

[Akk+95] N. Akkiraju et al. „Alpha Shapes: Definition and Software.“ In: *Proceedings of the 1st International Computational Geometry Software Workshop*. 1995, pp. 63–66 (see p. 41).

[AlH+13] S. AlHalawani et al. „Interactive Facades Analysis and Synthesis of Semi-Regular Facades“. In: *CGF* 32 (2013), pp. 215–224 (see p. 27).

[Alt+88] H. Alt et al. „Congruence, similarity, and symmetries of geometric objects“. In: *Discrete & Computational Geometry* 3.3 (1988), pp. 237–256 (see p. 19).

[Ata84] M. J. Atallah. „On symmetry detection“. In: (1984) (see p. 19).

[BCS03] B. Benes, J. A. Cordóba, and J. M. Soto. „Modeling Virtual Gardens by Autonomous Procedural Agents“. In: *Proceedings of TPCG*. IEEE Computer Society, 2003, p. 58 (see p. 103).

[Ben+11] B. Benes et al. „Urban Ecosystem Design“. In: *I3D*. 2011, pp. 167–174 (see pp. 30, 76, 79, 99–101).

[Bia+15] S. Biasotti et al. „Recent Trends, Applications, and Perspectives in 3D Shape Similarity Assessment“. In: *Computer Graphics Forum, Online Preprint* (2015) (see p. 32).

[Bie87] I. Biederman. „Recognition-by-components: a theory of human image understanding.“ In: *Psychological review* 94.2 (1987), p. 115 (see p. 37).

[BM02] B. Benes and E. U. Millán. „Virtual Climbing Plants Competing for Space“. In: *CA '02: Proceedings of the Computer Animation*. IEEE Computer Society, 2002, p. 33 (see p. 29).

[Bok+09] M. Bokeloh et al. „Symmetry detection using feature lines“. In: *Computer Graphics Forum*. Vol. 28. 2. Wiley Online Library. 2009, pp. 697–706 (see p. 20).

[BR93] D. Bloniarz and H. Ryan. „Designing alternatives to avoid street tree conflicts“. In: *Journal of Arboriculture* 19 (1993), pp. 152–152 (see p. 78).

[BSW13] F. Bao, M. Schwarz, and P. Wonka. „Procedural facade variations from a single layout“. In: *ACM Trans. on Graphics* 32.1 (2013), 8:1–8:13 (see p. 27).

[Bus+05] B. Bustos et al. „Feature-based Similarity Search in 3D Object Databases“. In: *ACM Comput. Surv.* 37.4 (Dec. 2005), pp. 345–387 (see p. 36).

[BWS10] M. Bokeloh, M. Wand, and H.-P. Seidel. „A connection between partial symmetry and inverse procedural modeling“. In: *ACM Trans. on Graphics*. Vol. 29. 4. ACM. 2010, p. 104 (see pp. 20, 27).

[CAD04] D. Cohen-Steiner, P. Alliez, and M. Desbrun. „Variational Shape Approximation“. In: *ACM Trans. Graph.* 23.3 (Aug. 2004), pp. 905–914. doi: 10.1145/1015706.1015817. url: https://doi.org/10.1145/1015706.1015817 (see p. 18).

[Cao+07] F. Cao et al. „A unified framework for detecting groups and application to shape recognition“. In: *Journal of Mathematical Imaging and Vision* 27.2 (2007), pp. 91–119 (see p. 25).

[CB17] S. Calderon and T. Boubekeur. „Bounding Proxies for Shape Approximation“. In: *ACM Transactions on Graphics (Proc. SIGGRAPH 2017)* 36.5 (July 2017) (see p. 21).

[CG97] F. R. Chung and F. C. Graham. *Spectral graph theory*. 92. American Mathematical Soc., 1997 (see p. 19).

[Cha+95] B. Chazelle et al. „Strategies for Polyhedral Surface Decomposition: An Experimental Study“. In: *Proceedings of the Eleventh Annual Symposium on Computational Geometry*. SCG '95. Vancouver, British Columbia, Canada: Association for Computing Machinery, 1995, pp. 297–305. doi: 10.1145/220279.220311. url: https://doi.org/10.1145/220279.220311 (see p. 18).

[Che+08] G. Chen et al. „Interactive Procedural Street Modeling“. In: *ACM SIGGRAPH 2008*. Los Angeles, California, 2008 (see p. 27).

[Che+13] B. Cheng et al. „Building simplification using backpropagation neural networks: a combination of cartographers' expertise and raster-based local perception“. In: *GIScience & Remote Sensing* 50.5 (2013), pp. 527–542 (see p. 22).

[Cho+97] H. I. Choi et al. „New Algorithm for Medial Axis Transform of Plane Domain“. In: *CVGIP: Graphical Model and Image Processing* 59 (1997), pp. 463–483 (see p. 84).

[Cor+01] L. P. Cordella et al. „An improved algorithm for matching large graphs“. In: *In: 3rd IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition, Cuen*. 2001, pp. 149–159 (see p. 44).

[Cor+17] G. Cordonnier et al. „Authoring landscapes by combining ecosystem and terrain erosion simulation“. In: *ACM Trans. on Graphics* 36.4 (2017), p. 134 (see p. 29).

[Dan+14a] M. Dang et al. „SAFE: Structure-aware Facade Editing“. In: *CGF* (2014) (see p. 27).

[Dan+14b] M. Dang et al. „SAFE: Structure-aware facade editing“. In: *Computer Graphics Forum*. Vol. 33. 2. Wiley Online Library. 2014, pp. 83–93 (see p. 24).

[Dan+15] M. Dang et al. „Interactive Design of Probability Density Functions for Shape Grammars“. In: *ACM Trans. on Graphics* 34.6 (2015), 206:1–206:13 (see p. 27).

[Déc+03] X. Décoret et al. „Billboard Clouds for Extreme Model Simplification“. In: *ACM SIGGRAPH 2003 Papers*. SIGGRAPH '03. San Diego, California: Association for Computing Machinery, 2003, pp. 689–696. doi: 10.1145/1201775.882326. url: https://doi.org/10.1145/1201775.882326 (see p. 112).

[Deu+98] O. Deussen et al. „Realistic Modeling and Rendering of Plant Ecosystems“. In: *Proc. of Sigg*. SIGGRAPH '98. ACM, 1998, pp. 275–286 (see pp. 29, 30, 76).

[DMM04] A. Desolneux, L. Moisan, and J.-M. Morel. „Gestalt theory and computer vision“. In: *Seeing, Thinking and Knowing*. Springer, 2004, pp. 71–101 (see p. 25).

[Eck+95] M. Eck et al. „Multiresolution Analysis of Arbitrary Meshes“. In: *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '95. New York, NY, USA: Association for Computing Machinery, 1995, pp. 173–182. doi: 10.1145/218380.218440. url: https://doi.org/10.1145/218380.218440 (see p. 18).

[Emi+15] A. Emilien et al. „WorldBrush: Interactive Example-Based Synthesis of Procedural Virtual Worlds“. In: *ACM Trans. on Graphics* 34.4 (2015) (see p. 29).

[End18] T. A. Endreny. „Strategically growing the urban forest will improve our world“. In: *Nature communications* 9.1 (2018), p. 1160 (see p. 78).

[ESC14] J. Engel, T. Schöps, and D. Cremers. „LSD-SLAM: Large-scale direct monocular SLAM“. In: *European conference on computer vision*. Springer. 2014, pp. 834–849 (see p. 26).

[Fan+10] Y. Fan et al. „The investigation of noise attenuation by plants and the corresponding noise-reducing spectrum“. In: *Journal of environmental health* 72.8 (2010), pp. 8–15 (see p. 121).

[FG+03] J. M. Findlay, I. D. Gilchrist, et al. *Active vision: The psychology of looking and seeing*. 37. Oxford University Press, 2003 (see pp. 6, 7).

[Fie73] M. Fiedler. „Algebraic connectivity of graphs“. In: *Czechoslovak mathematical journal* 23.2 (1973), pp. 298–305 (see p. 19).

[For07] A. Forberg. „Generalization of 3D building data based on a scale-space approach“. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 62.2 (2007), pp. 104–111 (see p. 22).

[Fri+11] D. Fritsch et al. „Multi-sensors and multiray reconstruction for digital preservation“. In: *Photogrammetric Week*. Vol. 11. 2011, pp. 305–323 (see p. 26).

[Fu+16] Q. Fu et al. „Structure-adaptive Shape Editing for Man-made Objects“. In: *Computer Graphics Forum* 35.2 (2016), pp. 27–36 (see p. 23).

[Gai+17] J. Gain et al. „EcoBrush: Interactive Control of Visually Consistent Large-Scale Ecosystems“. In: *CGF*. Vol. 36. 2. 2017, pp. 63–73 (see p. 29).

[Gal+09] R. Gal et al. „iWIRES: An Analyze-and-edit Approach to Shape Manipulation“. In: *ACM Trans. Graph.* 28.3 (July 2009), 33:1–33:10 (see p. 22).

[GC06] R. Gal and D. Cohen-Or. „Salient geometric features for partial shape matching and similarity“. In: *ACM Transactions on Graphics (TOG)* 25.1 (2006), pp. 130–150 (see p. 20).

[GD09] T. Glander and J. Döllner. „Abstract representations for interactive visualization of virtual 3D city models“. In: *Computers, Environment and Urban Systems* 33.5 (2009), pp. 375–387 (see p. 25).

[GL13] C. D. Gilbert and W. Li. „Top-down influences on visual processing“. In: *Nature Reviews Neuroscience* 14.5 (Apr. 2013), pp. 350–363. doi: 10.1038/nrn3476. url: https://doi.org/10.1038/nrn3476 (see p. 7).

[GP12] G. Gröger and L. Plümer. „CityGML–Interoperable semantic 3D city models“. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 71 (2012), pp. 12–33 (see p. 26).

[GPO09] J. W. Goudie, K. R. Polsson, and P. K. Ott. „An empirical model of crown shyness for lodgepole pine (Pinus contorta var. latifolia [Engl.] Critch.) in British Columbia“. In: *Forest ecology and management* 257.1 (2009), pp. 321–331 (see p. 113).

[Gra+08] F. Grabler et al. „Automatic Generation of Tourist Maps“. In: *ACM Trans. Graph.* 27.3 (Aug. 2008), 100:1–100:11 (see p. 22).

[Gre95] G. W. Grey. *The urban forest: Comprehensive management*. John Wiley & Sons, 1995 (see pp. 78, 79, 87).

[Gué+17] É. Guérin et al. „Interactive Example-based Terrain Authoring with Conditional Generative Adversarial Networks“. In: *ACM Trans. on Graphics* 36.6 (2017), 228:1–228:13 (see p. 30).

[Gue+20] P. Guehl et al. „Semi-Procedural Textures Using Point Process Texture Basis Functions“. In: *Comp. Graph. Forum* 39.4 (2020), pp. 159–171. doi: 10.1111/cgf.14061. url: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14061 (see p. 30).

[Guo+20] J. Guo et al. „Inverse Procedural Modeling of Branching Structures by Inferring L-Systems“. In: *ACM Trans. Graph.* (2020) (see p. 28).

[GWH01] M. Garland, A. Willmott, and P. S. Heckbert. „Hierarchical Face Clustering on Polygonal Surfaces“. In: *Proceedings of the 2001 Symposium on Interactive 3D Graphics*. I3D '01. New York, NY, USA: Association for Computing Machinery, 2001, pp. 49–58. doi: 10.1145/364338.364345. url: https://doi.org/10.1145/364338.364345 (see p. 18).

[Haa13] N. Haala. „The landscape of dense image matching algorithms“. In: (2013) (see p. 26).

[Häd+17] T. Hädrich et al. „Interactive Modeling and Authoring of Climbing Plants“. In: *Comput. Graph. Forum* 36.2 (2017), pp. 49–61 (see p. 29).

[HDR19] Y. Hu, J. Dorsey, and H. Rushmeier. „A novel framework for inverse procedural texture modeling“. In: *ACM Tran. on Grap. (TOG)* 38.6 (2019), pp. 1–14 (see p. 30).

[Hir07] H. Hirschmuller. „Stereo processing by semiglobal matching and mutual information“. In: *IEEE Transactions on pattern analysis and machine intelligence* 30.2 (2007), pp. 328–341 (see p. 26).

[Ilč+15] M. Ilčík et al. „Layer-Based Procedural Design of Façades“. In: *CGF* 34.2 (2015), pp. 205–216 (see p. 27).

[IOI06] T. Ijiri, S. Owada, and T. Igarashi. „Seamless Integration of Initial Sketching and Subsequent Detail Editing in Flower Modeling“. In: *CGF* 25.3 (2006), pp. 617–624 (see p. 28).

[Iso+16] P. Isola et al. „Image-to-Image Translation with Conditional Adversarial Networks“. In: *CVPR* (2016), pp. 5967–5976 (see pp. 91, 93, 103).

[Jol86] I. Jolliffe. *Principal Component Analysis*. Springer Verlag, 1986 (see p. 37).

[Kad06] M. Kada. „3D Building Generalization based on HalfSpace Modeling“. In: *In: Proceedings of the ISPRS Workshop on Multiple Representation and Interoperability of Spatial Data*. 2006 (see p. 22).

[Keb17] T. H. Kebrom. „A Growing Stem Inhibits Bud Outgrowth – The Overlooked Theory of Apical Dominance“. In: *Frontiers in Plant Science* 8 (2017), p. 1874 (see p. 89).

[Kel+17] T. Kelly et al. „BigSUR: large-scale structured urban reconstruction“. In: *ACM Trans. on Graph.* 36.6 (2017) (see pp. 27, 28).

[Kel+18] T. Kelly et al. „FrankenGAN: Guided Detail Synthesis for Building Mass Models Using Style-Synchonized GANs“. In: *ACM Trans. Graph.* 37.6 (2018), 1:1–1:14 (see p. 30).

[KFR04] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. „Symmetry descriptors and 3D shape matching“. In: *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*. 2004, pp. 115–123 (see p. 19).

[KGP05] T. H. Kolbe, G. Gröger, and L. Plümer. „CityGML: Interoperable access to 3D city models“. In: *Geo-information for disaster management*. Springer, 2005, pp. 883–899 (see p. 26).

[Kha+20] A. Khan et al. „A survey of the recent architectures of deep convolutional neural networks“. In: *Artificial Intelligence Review* 53.8 (Dec. 2020), pp. 5455–5516. doi: 10.1007/s10462-020-09825-6. url: https://doi.org/10.1007/s10462-020-09825-6 (see p. 30).

[Kra+14] J. Kratt et al. „Non-realistic 3D Object Stylization“. In: *Proceedings of the Workshop on Computational Aesthetics*. CAe '14. Vancouver, British Columbia, Canada: ACM, 2014, pp. 67–75 (see p. 24).

[Kra+15] J. Kratt et al. „Woodification: User-Controlled Cambial Growth Modeling“. In: *CGF* 34.2 (2015), pp. 361–372 (see p. 89).

[Kra+18] J. Kratt et al. „Sketching in gestalt space: interactive shape abstraction through perceptual reasoning“. In: *Computer Graphics Forum*. Vol. 37. 6. Wiley Online Library. 2018, pp. 188–204 (see p. 12).

[KT96] A. Kalvin and R. Taylor. „Superfaces: polygonal mesh simplification with bounded error“. In: *IEEE Computer Graphics and Applications* 16.3 (1996), pp. 64–77. doi: 10.1109/38.491187 (see p. 18).

[Kur+14] C. Kurz et al. „Symmetry-Aware Template Deformation and Fitting“. In: *Computer Graphics Forum* 33.6 (2014), pp. 205–219 (see p. 23).

[Kur+18] M. Kurppa et al. „Ventilation and Air Quality in City Blocks Using Large-Eddy Simulation-Urban Planning Perspective“. In: *Atmosphere* 9.2 (2018). doi: 10.3390/atmos9020065. url: https://www.mdpi.com/2073-4433/9/2/65 (see p. 122).

[KV08] M. Kubovy and M. Van Den Berg. „The whole is equal to the sum of its parts: A probabilistic model of grouping by proximity and similarity in regular patterns.“ In: *Psychological review* 115.1 (2008), p. 131 (see p. 25).

[LCT04] Y. Liu, R. T. Collins, and Y. Tsin. „A computational model for periodic pattern perception based on frieze and wallpaper groups“. In: *IEEE transactions on pattern analysis and machine intelligence* 26.3 (2004), pp. 354–371 (see p. 20).

[LD99] B. Lintermann and O. Deussen. „Interactive Modeling of Plants“. In: *IEEE Comput. Graph. Appl.* 19.1 (1999), pp. 56–65 (see p. 28).

[LDB05] G. Lavoué, F. Dupont, and A. Baskurt. „A new CAD mesh segmentation method, based on curvature tensor analysis“. In: *Computer-Aided Design* 37.10 (2005), pp. 975–987. doi: https://doi.org/10.1016/j.cad.2004.09.001. url: https://www.sciencedirect.com/science/article/pii/S0010448504002003 (see p. 18).

[Lév+02] B. Lévy et al. „Least squares conformal maps for automatic texture atlas generation“. In: *ACM transactions on graphics (TOG)* 21.3 (2002), pp. 362–371 (see p. 18).

[LH87] M. Livingstone and D. Hubel. „Psychophysical evidence for separate channels for the perception of form, color, movement, and depth“. In: *Journal of Neuroscience* 7.11 (1987), pp. 3416–3468. doi: 10.1523/JNEUROSCI.07-11-03416.1987. url: https://www.jneurosci.org/content/7/11/3416 (see p. 7).

[Li+04] Z. Li et al. „Automated building generalization based on urban morphology and Gestalt theory“. In: *International Journal of Geographical Information Science* 18.5 (2004), pp. 513–534 (see p. 24).

[Li+11a] C. Li et al. „Modeling and Generating Moving Trees from Video“. In: *ACM Trans. on Graphics* 30.6 (2011), 127:1–127:12 (see p. 28).

[Li+11b] Y. Li et al. „2D-3D fusion for layer decomposition of urban facades“. In: *ICCV*. 2011, pp. 882–889 (see p. 27).

[Li+13] Q. Li et al. „Geometric structure simplification of 3D building models“. In: *{ISPRS} Journal of Photogrammetry and Remote Sensing* 84 (2013), pp. 100–113 (see p. 33).

[Lin+13] D. Lindlbauer et al. „Perceptual grouping: selection assistance for digital sketching“. In: *International conference on Interactive tabletops and surfaces*. ACM. 2013, pp. 51–60 (see p. 24).

[Lin68] A. Lindenmayer. „Mathematical models for cellular interaction in development“. In: *Journal of Theoretical Biology* Parts I and II.18 (1968), pp. 280–315 (see p. 28).

[Lip+10] Y. Lipman et al. „Symmetry factored embedding and distance“. In: *ACM SIGGRAPH 2010 papers*. 2010, pp. 1–12 (see p. 20).

[Liu+10] Y. Liu et al. „Computational Symmetry in Computer Vision and Computer Graphics.“ In: *Foundations and Trends in Computer Graphics and Vision* (2010), pp. 1–195 (see p. 19).

[Liv+11] Y. Livny et al. „Texture-lobes for Tree Modelling“. In: *ACM Trans. on Graphics* 30.4 (2011), 53:1–53:10 (see p. 28).

[Lon+12] S. Longay et al. „TreeSketch: interactive procedural modeling of trees on a tablet“. In: *Proc. of the Intl. Symp. on SBIM*. 2012, pp. 107–120 (see p. 28).

[Løv+13] T. Løvset et al. „Rule-based method for automatic scaffold assembly from 3D building models“. In: *Computers & Graphics* 37.4 (2013), pp. 256–268 (see p. 24).

[LWH15] X. Liu, T.-T. Wong, and P.-A. Heng. „Closure-aware Sketch Simplification“. In: *ACM Trans. Graph.* 34.6 (2015), 168:1–168:10 (see p. 24).

[Mak+19] M. Makowski et al. „Synthetic Silviculture: Multi-scale Modeling of Plant Ecosystems“. In: *ACM Trans. on Graphics* 38.4 (2019), 131:1–131:14 (see pp. 29, 76).

[May+12] H. Mayer et al. „Dense 3D reconstruction from wide baseline image sets“. In: *Outdoor and Large-Scale Real-World Scene Analysis*. Springer, 2012, pp. 285–304 (see p. 26).

[Meh+09] R. Mehra et al. „Abstraction of Man-made Shapes“. In: *ACM Trans. Graph.* 28.5 (Dec. 2009), 137:1–137:10 (see pp. 21, 22, 32, 41, 52, 55, 130).

[MGP06] N. J. Mitra, L. J. Guibas, and M. Pauly. „Partial and Approximate Symmetry Detection for 3D Geometry“. In: *ACM Trans. Graph.* 25.3 (July 2006), pp. 560–568 (see p. 20).

[MHW15] R. W. Miller, R. J. Hauer, and L. P. Werner. *Urban forestry: planning and managing urban greenspaces*. Waveland press, 2015 (see pp. 76, 78, 79, 85, 87).

[Mic+12] E. Michaelsen et al. „Gestalt grouping on facade textures from ir image sequences: Comparing different production systemse“. In: *International Archives of Photogrammetry, Remote Sensing and Spatial Information Science* 39.B3 (2012), pp. 303–308 (see p. 24).

[Mit+12a] S. Mitchell et al. „Variable Radii Poisson-Disk Sampling“. In: *CCCG* (Jan. 2012) (see p. 84).

[Mit+12b] N. J. Mitra et al. „Symmetry in 3D Geometry: Extraction and Applications“. In: *EUROGRAPHICS State-of-the-art Report*. 2012 (see p. 19).

[Mit+13] N. Mitra et al. „Structure-aware Shape Processing“. In: *SIGGRAPH Asia 2013 Courses*. SA '13. Hong Kong, Hong Kong: ACM, 2013, 1:1–1:20 (see pp. 8, 22, 32).

[MM08] P. Merrell and D. Manocha. „Continuous Model Synthesis“. In: *ACM Trans. on Graphics* 27.5 (2008) (see p. 27).

[MM11] P. Merrell and D. Manocha. „Model Synthesis: A General Procedural Modeling Algorithm“. In: *TVCG* 17.6 (2011), pp. 715–728 (see p. 27).

[MP96] R. Měch and P. Prusinkiewicz. „Visual models of plants interacting with their environment“. In: *ACM SIGGRAPH 96*. New York, NY, USA: ACM, 1996, pp. 397–410 (see pp. 28, 89).

[MSM11] J. McCrae, K. Singh, and N. J. Mitra. „Slices: A Shape-proxy Based on Planar Sections“. In: *ACM Trans. Graph.* 30.6 (Dec. 2011), 168:1–168:12 (see p. 22).

[MT14] R. Minamino and M. Tateno. „Tree branching: Leonardo da Vinci's rule versus biomechanical models“. In: *PloS one* 9.4 (2014), e93535 (see p. 89).

[Mül+06] P. Müller et al. „Procedural modeling of buildings“. In: *ACM Trans. on Graphics* 25.3 (July 2006), pp. 614–623 (see p. 27).

[Mül+07a] P. Müller et al. „Image-based Procedural Modeling of Facades“. In: *ACM Trans. on Graphics* 26.3 (2007) (see p. 27).

[Mül+07b] P. Müller et al. „Image-based procedural modeling of facades“. In: *ACM Trans. Graph.* 26.3 (2007), p. 85 (see p. 20).

[MV13] A. Martinovic and L. Van Gool. „Bayesian grammar learning for inverse procedural modeling“. In: *CVPR*. 2013, pp. 201–208 (see p. 28).

[MY14] E. Michaelsen and V. V. Yashina. „Simple gestalt algebra“. In: *Pattern recognition and image analysis* 24.4 (2014), pp. 542–551 (see p. 25).

[Nan+11] L. Nan et al. „Conjoining Gestalt Rules for Abstraction of Architectural Drawings“. In: *ACM Trans. Graph.* 30.6 (Dec. 2011), 185:1–185:10 (see pp. 23, 24, 32, 33, 35, 36, 40, 72).

[Nie+22] T. Niese et al. „Procedural Urban Forestry“. In: *ACM Trans. Graph.* 41.2 (Mar. 2022). doi: 10.1145/3502220. url: https://doi.org/10.1145/3502220 (see p. 13).

[Nis+16a] G. Nishida et al. „Interactive sketching of urban procedural models“. In: *ACM Trans. on Graphics* 35.4 (2016), p. 130 (see p. 28).

[Nis+16b] G. Nishida et al. „Interactive Sketching of Urban Procedural Models“. In: *ACM Trans. Graph.* 35.4 (July 2016), 130:1–130:11 (see p. 23).

[NYC19] NYCOpenData. „The Next Decade of Open Data“. In: (2019). url: https://data.ny.gov/ (see pp. 91, 93, 97).

[OOI07] M. Okabe, S. Owada, and T. Igarashi. „Interactive Design of Botanical Trees Using Freehand Sketches and Example-based Editing“. In: *ACM SIGGRAPH Courses*. San Diego, California: ACM, 2007 (see p. 28).

[Opp86] P. E. Oppenheimer. „Real time design and animation of fractal plants and trees“. In: *Proc. of SIGGRAPH* 20.4 (1986), pp. 55–64 (see p. 28).

[Pal+09] W. Palubicki et al. „Self-organizing Tree Models for Image Synthesis“. In: *ACM Trans. on Graphics* 28.3 (2009), 58:1–58:10 (see pp. 29, 86, 88, 89).

[Pas+14] S. Pasewaldt et al. „Multi-perspective 3D panoramas“. In: *International Journal of Geographical Information Science* 28.10 (2014), pp. 2030–2051 (see p. 25).

[Pau+08] M. Pauly et al. „Discovering Structural Regularity in 3D Geometry“. In: *ACM Trans. Graph.* 27.3 (Aug. 2008), 43:1–43:11 (see pp. 21, 37).

[Pir+12a] S. Pirk et al. „Capturing and animating the morphogenesis of polygonal tree models“. In: *ACM Trans. on Graphics* 31.6 (2012), 169:1–169:10 (see pp. 14, 28).

[Pir+12b] S. Pirk et al. „Plastic trees: interactive self-adapting botanical tree models“. In: *ACM Trans. on Graphics* 31.4 (2012), 50:1–50:10 (see pp. 10, 29).

[Pir+14] S. Pirk et al. „Windy Trees: Computing Stress Response for Developmental Tree Models“. In: *ACM Trans. on Graphics* 33.6 (2014), 204:1–204:11 (see pp. 14, 28, 122).

[Pir+16] S. Pirk et al. „Modeling Plant Life in Computer Graphics“. In: *ACM SIGGRAPH 2016 Courses*. Anaheim, California, 2016 (see p. 89).

[Pir+17] S. Pirk et al. „Interactive Wood Combustion for Botanical Tree Models“. In: *ACM Trans. on Graphics* 36.6 (2017), 197:1–197:12 (see p. 28).

[PM01] Y. I. H. Parish and P. Müller. „Procedural Modeling of Cities“. In: ACM SIGGRAPH 2001. 2001, pp. 301–308 (see p. 27).

[Pod+06] J. Podolak et al. „A planar-reflective symmetry transform for 3D shapes“. In: *ACM SIGGRAPH 2006 Papers*. 2006, pp. 549–559 (see p. 20).

[Pru86] P. Prusinkiewicz. „Graphical Applications of L-systems“. In: *Proceedings on Graphics Interface '86/Vision Interface '86*. Vancouver, British Columbia, Canada: Canadian Information Processing Society, 1986, pp. 247–253 (see p. 28).

[PTK87] T. Poggio, V. Torre, and C. Koch. „Computational vision and regularization theory“. In: *Readings in computer vision* (1987), pp. 638–643 (see p. 6).

[Ref+88] P. de Reffye et al. „Plant Models Faithful to Botanical Structure and Development“. In: *SIGGRAPH Comput. Graph.* 22.4 (1988), pp. 151–158 (see p. 29).

[Ric+20] D. Richards et al. „Differential air temperature cooling performance of urban vegetation types in the tropics“. In: *Urban Forestry & Urban Greening* 50 (2020), p. 126651. doi: https://doi.org/10.1016/j.ufug.2020.126651. url: https://www.sciencedirect.com/science/article/pii/S1618866719304613 (see p. 120).

[Rit+15] D. Ritchie et al. „Controlling procedural modeling programs with stochastically-ordered sequential Monte Carlo“. In: *ACM Trans. on Graphics* 34.4 (2015), p. 105 (see p. 28).

[RLP07] A. Runions, B. Lane, and P. Prusinkiewicz. „Modeling Trees with a Space Colonization Algorithm“. In: *Conference on Natural Phenomena*. NPH–07. 2007, pp. 63–70 (see pp. 29, 89).

[RWL18] D. Ritchie, K. Wang, and Y.-A. Lin. „Fast and Flexible Indoor Scene Synthesis via Deep Convolutional Generative Models“. In: *CVPR*. 2018 (see p. 30).

[SBM01] D. Shikhare, S. Bhakar, and S. Mudur. „Compression of large 3D engineering models using automatic discovery of repeating geometric features“. In: *Signal Processing* 19.20 (2001), p. 15 (see p. 21).

[Sha08] A. Shamir. „A survey on Mesh Segmentation Techniques“. In: *Computer Graphics Forum* 27.6 (2008), pp. 1539–1556. doi: https://doi.org/10.1111/j.1467-8659.2007.01103.x. url: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2007.01103.x (see pp. 18, 19).

[She07] V. K. D. J. A. Sheffer. „Shuffler: Modeling with interchangeable parts“. In: *Visual Computer journal* (2007) (see p. 18).

[SM15] M. Schwarz and P. Müller. „Advanced Procedural Modeling of Architecture“. In: *ACM Trans. on Graphics* 34.4 (Proceedings of SIGGRAPH) (2015), 107:1–107:12 (see p. 27).

[Sme+14] R. M. Smelik et al. „A Survey on Procedural Modelling for Virtual Worlds“. In: *CGF* 33.6 (2014), pp. 31–50 (see p. 27).

[Spe+20] A. Speak et al. „The influence of tree traits on urban ground surface shade cooling“. In: *Landscape and Urban Planning* 197 (2020), p. 103748. DOI: https://doi.org/10.1016/j.landurbplan.2020.103748. URL: https://www.sciencedirect.com/science/article/pii/S0169204619309338 (see p. 120).

[SS97] C. Sun and J. Sherrah. „3D symmetry detection using the extended Gaussian image“. In: *IEEE transactions on pattern analysis and machine intelligence* 19.2 (1997), pp. 164–168 (see p. 19).

[Sta+10] O. Stava et al. „Inverse Procedural Modeling by Automatic Generation of L-systems“. In: *CGF* 29.2 (2010), pp. 665–674 (see pp. 28, 89).

[Sta+14] O. Stava et al. „Inverse Procedural Modelling of Trees“. In: *CGF* 33.6 (2014), pp. 118–131 (see pp. 28, 29).

[STK02] S. Shlafman, A. Tal, and S. Katz. „Metamorphosis of Polyhedral Surfaces using Decomposition“. In: *Computer Graphics Forum* 21.3 (2002), pp. 219–228. DOI: https://doi.org/10.1111/1467-8659.00581. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-8659.00581 (see p. 18).

[Sun+11] X. Sun et al. „Automated abstraction of building models for 3D navigation on mobile devices“. In: *2011 19th International Conference on Geoinformatics*. IEEE. 2011, pp. 1–6 (see p. 25).

[Tal+11] J. O. Talton et al. „Metropolis procedural modeling“. In: *ACM Trans. on Graphics* 30.2 (2011), p. 11 (see p. 27).

[Tan+07] P. Tan et al. „Image-based Tree Modeling“. In: *ACM Trans. on Graphics* 26.3 (2007) (see p. 28).

[Tan+08] P. Tan et al. „Single Image Tree Modeling“. In: *ACM Trans. on Graphics* 27.5 (2008), 108:1–108:7 (see p. 28).

[TMS08] G. Tzimiropoulos, N. Mitianoudis, and T. Stathaki. „A unifying approach to moment-based shape orientation and symmetry classification“. In: *IEEE Transactions on Image Processing* 18.1 (2008), pp. 125–139 (see p. 19).

[Tut+16a] P. Tutzauer et al. „A study of the human comprehension of building categories based on different 3D building representations“. In: *PFG Photogrammetrie, Fernerkundung, Geoinformation* (2016), pp. 319–333 (see p. 13).

[Tut+16b] P. Tutzauer et al. „Understanding human perception of building categories in virtual 3D cities: a user study“. In: *XXIII ISPRS Congress, Commission II*. 2016, pp. 683–687 (see p. 13).

[VAB10] C. A. Vanegas, D. G. Aliaga, and B. Benes. „Building reconstruction using manhattan-world grammars“. In: *CVPR* 0 (2010), pp. 358–365 (see p. 27).

[Van+09] C. A. Vanegas et al. „Interactive Design of Urban Spaces Using Geometrical and Behavioral Modeling“. In: *ACM SIGGRAPH Asia*. Yokohama, Japan, 2009 (see pp. 27, 79).

[Van+10] C. A. Vanegas et al. „Modelling the Appearance and Behaviour of Urban Spaces“. In: *CGF* 29.1 (2010), pp. 25–42 (see pp. 27, 79).

[Van+11] O. Van Kaick et al. „A survey on shape correspondence“. In: *Computer graphics forum*. Vol. 30. 6. Wiley Online Library. 2011, pp. 1681–1707 (see p. 19).

[Van+12] C. A. Vanegas et al. „Inverse Design of Urban Procedural Models“. In: *ACM Trans. on Graphics* 31.6 (Nov. 2012), 168:1–168:11 (see p. 28).

[Var+03] S. Vardoulakis et al. „Modelling air quality in street canyons: a review“. In: *Atmospheric Environment* 37.2 (2003), pp. 155–182. doi: https://doi.org/10.1016/S1352-2310(02)00857-9. url: https://www.sciencedirect.com/science/article/pii/S1352231002008579 (see p. 122).

[Vos+13] P. E. Vos et al. „Improving local air quality in cities: To tree or not to tree?“ In: *Environmental Pollution* 183 (2013). Selected Papers from Urban Environmental Pollution 2012, pp. 113–122. doi: https://doi.org/10.1016/j.envpol.2012.10.021. url: https://www.sciencedirect.com/science/article/pii/S0269749112004605 (see p. 120).

[Wad+07] P. Waddell et al. „Incorporating land use in metropolitan transportation planning“. In: *Transportation Research Part A: Policy and Practice* 41.5 (2007), pp. 382–410 (see pp. 78, 79).

[Wad02] P. Waddell. „UrbanSim: Modeling urban development for land use, transportation, and environmental planning“. In: *Journal of the American planning association* 68.3 (2002), pp. 297–314 (see p. 79).

[Wan+11] Y. Wang et al. „Symmetry Hierarchy of Man-Made Objects“. In: *Computer Graphics Forum* 30.2 (2011), pp. 287–296 (see p. 22).

[Wan+15] Y. Wang et al. „A Gestalt rules and graph-cut-based simplification framework for urban building models“. In: *International Journal of Applied Earth Observation and Geoinformation* 35 (2015), pp. 247–258 (see p. 24).

[Wan+19] K. Wang et al. „PlanIT: Planning and Instantiating Indoor Scenes with Relation Graph and Spatial Prior Networks“. In: *ACM Trans. on Graphics* 38.4 (2019) (see p. 30).

[Wat+08] B. Watson et al. „Procedural Urban Modeling in Practice“. In: *IEEE Computer Graphics and Applications* 28.3 (2008), pp. 18–26 (see p. 27).

[Web+09] B. Weber et al. „Interactive Geometric Simulation of 4D Cities“. In: *CGF* 28.2 (2009), pp. 481–492 (see pp. 27, 79).

[Wer23] M. Wertheimer. „Untersuchungen zur Lehre von der Gestalt. II“. In: *Psychological Research* 4.1 (1923), pp. 301–350 (see p. 23).

[Wer38] M. Wertheimer. „Laws of organization in perceptual forms“. In: *A Source Book of Gestalt Psychology*. Ed. by W. Ellis. Routledge and Kegan Paul, 1938, pp. 71–88 (see p. 23).

[Wil11] A. Willmott. „Rapid Simplification of Multi-attribute Meshes“. In: *Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics*. HPG '11. Vancouver, British Columbia, Canada: ACM, 2011, pp. 151–158 (see p. 32).

[Wit+09] J. Wither et al. „Structure from silhouettes: a new paradigm for fast sketch-based design of trees“. In: *CGF* 28.2 (2009), pp. 541–550 (see p. 28).

[Won+03] P. Wonka et al. „Instant Architecture“. In: *ACM Trans. on Graphics* 22.3 (2003), pp. 669–677 (see p. 27).

[Wu 05] J. Wu Leif Kobbelt. „Structure Recovery via Hybrid Variational Surface Approximation“. In: *Computer Graphics Forum* 24.3 (2005), pp. 277–284. doi: https://doi.org/10.1111/j.1467-8659.2005.00852.x. url: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2005.00852.x (see p. 18).

[Wu+14] F. Wu et al. „Inverse Procedural Modeling of Facade Layouts“. In: *ACM Trans. on Graphics* 33.4 (2014), 121:1–121:10 (see p. 27).

[WXH17] X. Wu, K. Xu, and P. Hall. „A survey of image synthesis and editing with generative adversarial networks“. In: *Tsinghua Science and Technology* 22.6 (2017), pp. 660–674 (see p. 30).

[WZB17] B. Wang, Y. Zhao, and J. Barbič. „Botanical Materials Based on Biomechanics“. In: *ACM Trans. on Graphics* 36.4 (2017), 135:1–135:13 (see p. 28).

[XCW14] J. Xing, H.-T. Chen, and L.-Y. Wei. „Autocomplete Painting Repetitions“. In: *ACM Trans. Graph.* 33.6 (Nov. 2014), 172:1–172:11 (see p. 44).

[Xie+16] K. Xie et al. „Tree Modeling with Real Tree-Parts Examples“. In: *TVCG* 22.12 (2016), pp. 2608–2618 (see p. 28).

[Xu+12] P. Xu et al. „Lazy Selection: A Scribble-based Tool for Smart Shape Elements Selection“. In: *ACM Trans. Graph.* 31.6 (Nov. 2012), 142:1–142:9 (see p. 24).

[Yan+19] J. Yang et al. „Local climate zone ventilation and urban land surface temperatures: Towards a performance-based and wind-sensitive planning proposal in megacities“. In: *Sustainable Cities and Society* 47 (2019), p. 101487. doi: https://doi.org/10.1016/j.scs.2019.101487. url: https://www.sciencedirect.com/science/article/pii/S2210670718324569 (see p. 123).

[Yeh+13] Y.-T. Yeh et al. „Synthesis of Tiled Patterns Using Factor Graphs“. In: *ACM Trans. on Graphics* 32.1 (2013), 3:1–3:13 (see p. 28).

[YK12] M. E. Yumer and L. B. Kara. „Co-abstraction of Shape Collections“. In: *ACM Trans. Graph.* 31.6 (Nov. 2012), 166:1–166:11 (see pp. 22, 32).

[ZB13] Y. Zhao and J. Barbič. „Interactive Authoring of Simulation-ready Plants“. In: *ACM Trans. on Graphics* 32.4 (2013), 84:1–84:12 (see p. 28).

[Zha+13] L. Zhang et al. „A spatial cognition-based urban building clustering approach and its applications“. In: *International Journal of Geographical Information Science* 27.4 (2013), pp. 721–740 (see p. 24).

[ZIK98] S. Zhukov, A. Iones, and G. Kronin. „An ambient light illumination model“. English. In: *Rendering Techniques '98*. Ed. by G. Drettakis and N. Max. Eurographics. Springer Vienna, 1998, pp. 45–55 (see pp. 34, 39).

[ZPA95] H. Zabrodsky, S. Peleg, and D. Avnir. „Symmetry as a continuous feature“. In: *IEEE Transactions on pattern analysis and machine intelligence* 17.12 (1995), pp. 1154–1166 (see p. 19).

[ZW97] H. Zabrodsky and D. Weinshall. „Using bilateral symmetry to improve 3D reconstruction from image sequences“. In: *Computer vision and image understanding* 67.1 (1997), pp. 48–57 (see p. 19).

[ZWF18] H. Zeng, J. Wu, and Y. Furukawa. „Neural procedural reconstruction for residential buildings“. In: *Proceedings of the ECCV*. 2018, pp. 737–753 (see pp. 28, 30).