

BBSR-
Online-Publikation
41/2022

Digitale Methodik für eine integrierte maßstabsübergreifende Generierung, Analyse und Exploration von Planungs- varianten für Wohnungsbauprojekte

von

Jun.-Prof. Dr. Reinhard König
Vertr.-Prof. Dr.-Ing. Sven Schneider
Iuliia Osintseva
Andreas Berst

Digitale Methodik für eine integrierte maßstabsübergreifende Generierung, Analyse und Exploration von Planungsvarianten für Wohnungsbauprojekte

Gefördert durch:



Bundesministerium
für Wohnen, Stadtentwicklung
und Bauwesen

aufgrund eines Beschlusses
des Deutschen Bundestages

ZUKUNFT BAU
FORSCHUNGSFÖRDERUNG

Dieses Projekt wurde gefördert vom Bundesinstitut für Bau-, Stadt- und Raumforschung (BBSR) im Auftrag des Bundesministeriums für Wohnen, Stadtentwicklung und Bauwesen (BMWSB) aus Mitteln des Innovationsprogramms Zukunft Bau.

Aktenzeichen: 10.08.18.7-18.15

Projektlaufzeit: 11.2018 bis 12.2020

IMPRESSUM

Herausgeber

Bundesinstitut für Bau-, Stadt- und Raumforschung (BBSR)
im Bundesamt für Bauwesen und Raumordnung (BBR)
Deichmanns Aue 31–37
53179 Bonn

Fachbetreuer

Bundesinstitut für Bau-, Stadt- und Raumforschung
Referat WB 3 „Forschung im Bauwesen“
Dr.-Ing. Arnd Rose
arnd.rose@bbr.bund.de

Autorinnen und Autoren

Bauhaus-Universität Weimar, Lehrstuhl für Computational Architecture
Jun.-Prof. Dr. Reinhard König (Projektleitung)

Bauhaus-Universität Weimar, Lehrstuhl für Informatik in der Architektur
Vertr.-Prof. Dr.-Ing. Sven Schneider (Projektleitung)

Iuliia Osintseva, M. Sc. (Projektbearbeiterin)
Andreas Berst, M. Sc. (Projektbearbeiter)

Stand

März 2021

Bildnachweis

Titelbild: Bauhaus-Universität Weimar

Vervielfältigung

Alle Rechte vorbehalten

Der Herausgeber übernimmt keine Gewähr für die Richtigkeit, die Genauigkeit und Vollständigkeit der Angaben sowie für die Beachtung privater Rechte Dritter. Die geäußerten Ansichten und Meinungen müssen nicht mit denen des Herausgebers übereinstimmen.

Zitierweise

König, Reinhard; Schneider, Sven; Osintseva, Iuliia; Berst, Andreas, 2022: Digitale Methodik für eine integrierte maßstabsübergreifende Generierung, Analyse und Exploration von Planungsvarianten für Wohnungsbauprojekte. BBSR-Online-Publikation 41/2022, Bonn.

Inhaltsverzeichnis

1. Kurzfassung	5
2. Einleitung und Grundlagen	6
2.1 Planungsaufgaben im Wohnungsbau	6
2.2 Entwurfsraumexploration	6
2.3 Generative Methode und Struktur	6
2.4 Nutzerinteraktion im generativen Planungsprozess	8
3. Arbeitspakete	9
3.1 AP1 Generative Komponenten	9
3.1.1. Gebäudekubatur	9
3.1.2. Erschließung	28
3.1.3. Grundrisse	38
3.2 AP2 Analysewerkzeuge	40
3.3 AP3 Benutzerschnittstelle	45
3.4 AP4 Praxisbezug	48
4. Konklusion und Ausblick	52
Mitwirkende	53
Autorinnen und Autoren	53
Studentische Hilfskräfte	53
Projektpartner	53
Fachliche Betreuung	53
Literaturverzeichnis	54
Anhang	56
Anhang 1: Struktur und Funktionen der prototypischen Entwicklungsumgebung in python	56
Anhang 2: Building Divider: Methode zur Platzierung rechteckiger Wohnungen in konkaven Gebäudekubaturen gemäß dem Wohnungsschlüssel	57
Anhang 3:	60
Anhang 4:	60

1. Kurzfassung

Trotz des Anstiegs der Neubauten im Wohnungsbau gibt es zurzeit insbesondere in den Metropolregionen Wohnraummangel [1]. Nach aktuellen Schätzungen des Instituts der Deutschen Wirtschaft wurden alleine in den Jahren 2019 und 2020 341.700 neue Wohnungen benötigt. Dieser Bedarf konnte trotz stetig steigender Bautätigkeit nicht gedeckt werden [2]. Die Digitalisierung in der Baubranche hilft dabei, die Planung neuer Wohnungsbauten nachhaltiger, effizienter und wirtschaftlicher zu gestalten. Die Erstellung einer durchgängigen digitalen Prozesskette (Bauindustrie 4.0) von der Grundrisserstellung bis zur Bauteilfertigung macht die komplexen Planungs- und Bauprozesse flexibler, transparenter und effizienter. Dadurch sinken die Fehlerquoten sowie die Kosten für die Bauproduktion.

Aktuelle Entwicklungen in der Bauindustrie implementieren das Industrie 4.0-Paradigma vor allem bei der Bauausführung, wodurch sich neue Gebäude flexibel und kostengünstig an den jeweiligen Kontext oder formale Vorgaben anpassen lassen. Diese Entwicklung hat zur Folge, dass die Anzahl möglicher Entwurfsvarianten um ein Vielfaches größer wird, was wiederum neue Methoden erfordert, diesen großen Raum an Möglichkeiten effizient systematisch zu untersuchen. Anders ausgedrückt, kann man heute zwar effizient maßgeschneidert bauen; die Entwurfsmethoden bleiben bei dieser Entwicklung jedoch noch weit hinter den tatsächlichen Möglichkeiten zurück. Der Hauptgrund für diese Ausgangssituation ist, dass automatisierte Modellierung von Planungsvarianten bereits in frühen Planungsphasen zahlreiche komplexe Problemstellungen beinhalten und durch die große Anzahl an Freiheitsgraden einen sehr großen Lösungsraum bereithält.

Um unterschiedliche Varianten für diesen immensen Lösungsraum effizient und zielgerichtet zu erzeugen, können durch numerische Parameter gesteuerte Heuristiken eingesetzt werden [3], [4]. Ein anderer Weg den enormen Lösungsraum nach guten Varianten zu filtern, ist der Einsatz von Machine Learning Modellen [5]. Diese werden mit Hilfe von bestehenden Datensätzen von zum Beispiel Bebauungsvarianten trainiert, um sie schließlich auf neue Probleme anwenden zu können. Allerdings beschränken sich viele dieser Prototypen und Werkzeuge lediglich auf einzelne Teilprobleme des Planungsprozesses oder sind nur für spezifische Anwendungsfälle geeignet.

Basierend auf dieser Ausgangslage wurde im Rahmen des vorliegenden Projekts eine praxistaugliche Methodik für die systematische Generierung und Exploration von Planungsvarianten für komplexe Wohnbauprojekte entwickelt. Es wurde eine Gliederung in drei Maßstabsebenen, Gebäudevolumen, Erschließung/Wohneinheiten und Wohnungsgrundrisse vorgenommen. Diese Gliederung des Entwurfsprozesses in drei Ebenen bildet den Ausgangspunkt für die Aufstellung folgender Forschungsfragen:

- Welche Algorithmen sind für die jeweiligen Teilaspekte der Wohnungsbauplanung geeignet?
- Wie können Schnittstellen zwischen verschiedenen Generierungsschritten gestaltet werden, um einen ganzheitlichen generativen Prozess abzubilden?
- Wie kann der Nutzer die qualitativ-raumbezogenen Kriterien in die Generierung von Varianten einbeziehen?
- Inwieweit kann die entwickelte Methodik dazu beitragen, in der Planungspraxis schneller zu besseren Lösungen zu gelangen?

Unter Berücksichtigung dieser Forschungsfragen wurden in diesem Projekt verschiedene Generierungs- und Analysemodule entwickelt und kombiniert, um einen Planungsprozess über die verschiedenen Maßstabsebenen abzubilden. Die dabei entwickelten Prototypen wurden in enger Zusammenarbeit mit der Firma DIPLAN und der Firma Decoding Spaces erarbeitet, um einen starken Praxisbezug zu gewährleisten.

2. Einleitung und Grundlagen

2.1 Planungsaufgaben im Wohnungsbau

Bei der Planung eines Wohngebäudes müssen zahlreiche Aspekte berücksichtigt werden. Viele weitreichende Entscheidungen werden dabei bereits in der frühen Entwurfsphase (Leistungsphasen LP0 bis LP2) [6] getroffen. Das Ergebnis dieser Phase beeinflusst die wirtschaftliche, ökologische wie soziale Performance eines Gebäudes auf sehr lange Zeit. Daher ist es entscheidend, in der frühen Entwurfsphase eine hinsichtlich dieser Aspekte gute Lösung zu entwickeln. Um das zu erreichen, müssen systematisch mehrere Planungsvarianten erstellt, analysiert und verglichen werden (Entwurfsraumexploration). Da sich viele Planungskriterien gegenseitig beeinflussen beziehungsweise einander widersprechen, bilden möglichst gute Lösungen stets ein Gleichgewicht zwischen verschiedenen Planungskriterien (Pareto-optimal). Obwohl bereits zahlreiche generative Ansätze existieren, werden in der Praxis immer noch überwiegend manuelle Methoden (z. B. Skizzieren, Modellbau) zur Erzeugung von Varianten verwendet [7]. Folglich wird nur eine relativ kleine Anzahl an Varianten überprüft und manche gute Lösung nicht betrachtet. Daraus folgt, dass, auch wenn das endgültige Design den Anforderungen entspricht, wahrscheinlich bessere Pareto-optimale Lösungen bei der Suche übersehen wurden. Für eine möglichst umfängliche Generierung Pareto-optimaler Lösungen können parametrische Werkzeuge wertvolle Unterstützung bieten.

2.2 Entwurfsraumexploration

Der größte Vorteil von computerbasierten Generierungsmethoden ist die Erweiterung des Entwurfsraums (Design Space). In der Praxis kann jedoch der Lösungsraum aufgrund der Anzahl der möglichen Rahmenbedingungen, Entwurfsparameter und -kriterien extrem groß werden, wodurch der Berechnungsaufwand zu groß wird, um den kompletten Lösungsraum zu durchsuchen. Daher war es eine wichtige Anforderung an das Projekt, ein Generierungssystem zu entwickeln, das eine große Anzahl an sinnvollen Varianten möglichst zeiteffizient erstellen kann. Um diesem Ziel gerecht zu werden, wurde der Fokus der generativen Methoden auf die Replikation der gängigen Wohnungsbautypologien und Planungsschritte gelegt. Durch diese der Praxis entlehnte Eingrenzung werden gänzlich unbrauchbare Lösungen von vornherein ausgeschlossen, wodurch der Lösungsraum überschaubarer und die generative Methodik in der Praxis anwendbar bleibt.

2.3 Generative Methode und Struktur

Für die geometrische Erstellung von Planungsvarianten werden parametrische Modelle verwendet. Diese Modelle sind an verschiedene formgebende Parameter geknüpft, durch welche die Generierung der Modelle kontrolliert wird. Die Schwierigkeit in der Anwendung parametrischer Modelle zur Erzeugung von Gebäudeentwürfen liegt darin, die zahlreichen maßstabsübergreifenden Abhängigkeiten zu berücksichtigen. So, müssen bei der Planung eines Gebäudes Teilaspekte auf unterschiedlichen Maßstäben (z. B. Gebäudevolumen, Erschließung, Wohnungsaufteilung) aufeinander abgestimmt werden. Aspekte auf unterschiedlichen Maßstabsebenen beeinflussen sich dabei gegenseitig und sind daher nicht getrennt voneinander zu bearbeiten (so hat z. B. die Definition des Gebäudevolumens einen großen Einfluss auf die Möglichkeiten zur Definition des Erschließungssystems und vice versa).

Die im Projekt verfolgte generative Methode ist als Top-Down Ansatzes konzipiert. Das in Abbildung 1 illustrierte Konzept gliedert die Planung in drei Maßstabsebenen: (i) Kubaturen, (ii) Erschließungen und Aufteilung in Wohnungen, (iii) Wohnungsgrundrisse. Diese Aufteilung erlaubt es zum einen die komplexen Problemstellungen der einzelnen Ebenen gezielter anzugehen, zum anderen sorgt diese Gliederung dafür, dass übergeordnete Bauregeln besser umgesetzt werden können.

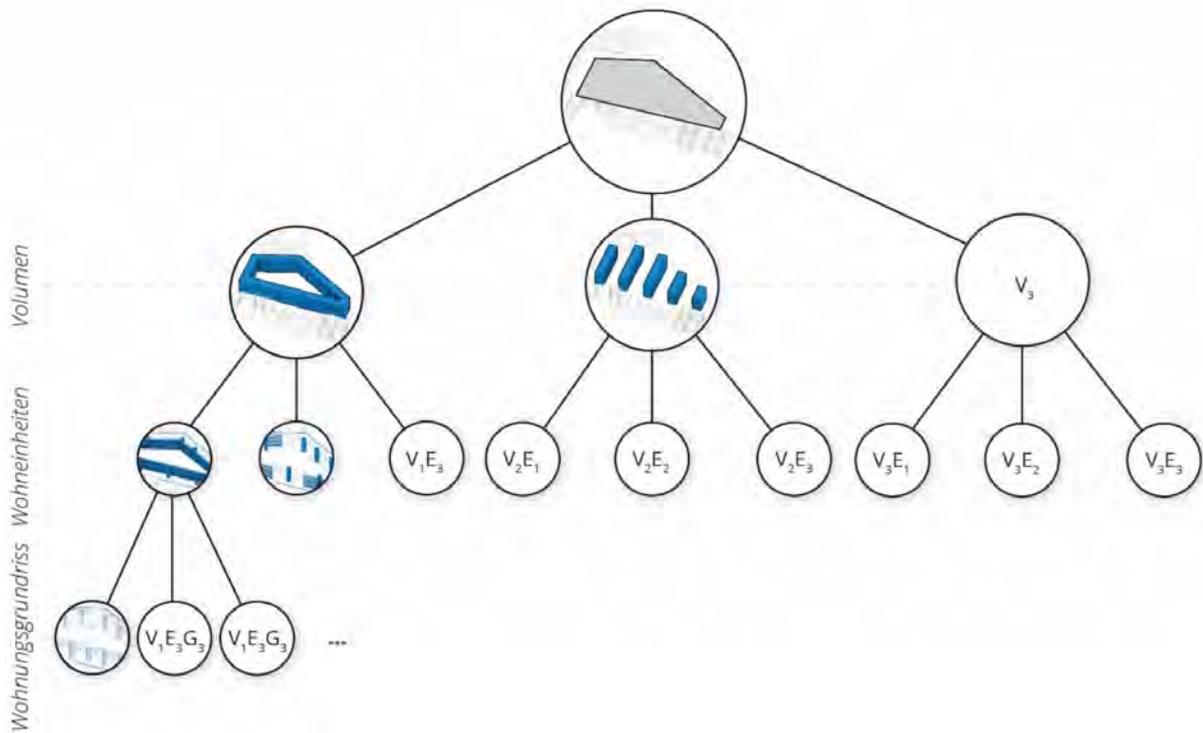


Abbildung 1
Darstellung des Top-Down Ansatzes, bei dem zuerst Gebäudekubaturen auf einem Grundstück (Ebene 1) erstellt werden, die weiter in Wohnungen aufgeteilt (Ebene 2) und bis in die Grundrisse (Ebene 3) ausgearbeitet werden.

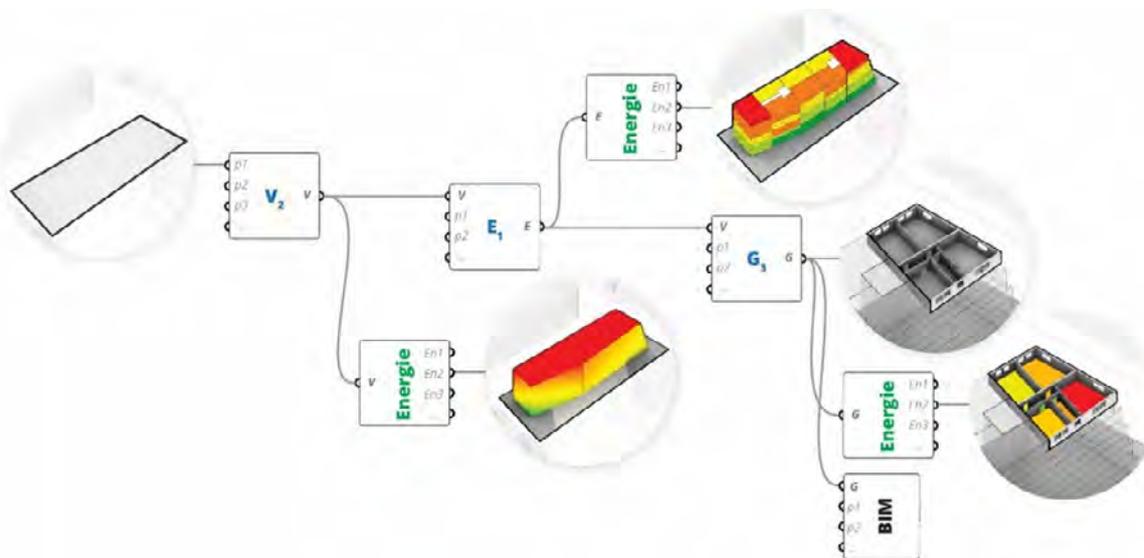


Abbildung 2
Schematische Darstellung der modularen generativen Methode.

Dieser Top-Down Ansatz ermöglicht es weiterhin, auf die einzelnen Maßstabsebenen angepasste Analysen durchzuführen, die dann in weiteren Schritten genauer spezifiziert bzw. als Input für weitere Generierungsschritte verwendet werden können (Abbildung 2). Um eine maßstabsübergreifende Generierung zu ermöglichen, wurden die Teillösungen in Form von Modulen umgesetzt. Es wurden Schnittstellen zwischen den verschiedenen generativen Algorithmen und Analysemethoden der einzelnen Maßstabsebenen geschaffen.

2.4 Nutzerinteraktion im generativen Planungsprozess

Durch die Erzeugung einer Vielzahl an Varianten, ergibt sich ein weiteres Problem: Wie lassen sich in der großen Menge an Varianten die Geeigneten identifizieren. Für diesen Zweck werden üblicherweise Performancekriterien (z. B. Kosten, Energieverbrauch, Solareintrag, Tageslicht, Sichtbeziehungen, Erreichbarkeit) einer jeden Variante berechnet. Anhand dieser Kriterien können die Varianten gefiltert bzw. miteinander verglichen werden. Jedoch existieren in der Regel auch kontext- bzw. situationsabhängige Anforderungen, die nicht anhand der existierenden Analysemethoden abgebildet werden können, sondern einen qualitativen raumbezogenen Charakter haben (z. B. Lage von Objekten im Raum). Beispielsweise: „ausgewählte Zimmer sollen nur nach bestimmten Himmelsrichtungen orientiert sein“ oder „die Eingänge in einem Wohnblock sollen gegenseitig einsehbar sein, um soziale Interaktion zu fördern“ oder „aus gestalterischen Gründen, soll das Volumen des Gebäudes an einer bestimmten Grundstücksseite mehrfach unterbrochen werden“. Grundsätzlich lassen sich solche Kriterien im Rahmen generativer Prozesse abbilden, jedoch fehlen bislang Methoden entsprechende Suchanfragen informationstechnisch auf eine für Planer einfache Art und Weise zu formulieren [8]. Um solche qualitativen raumbezogenen Faktoren besser in den Planungsprozess integrieren zu können, ist es vor allem wichtig, dem Entwerfer zu erlauben, Einfluss auf den Generierungsprozess zu nehmen. Deshalb wurde der Generierungsprozess so gestaltet, dass er nicht nur anhand numerischer Anforderungen kontrolliert wird, sondern es für den Entwerfer möglich ist, seine Erfahrungswerte in den Prozess einzubringen. Daher wurden in diesem Projekt wesentliche generative Module so konzipiert, dass Entwerfer über einfache Interaktionstechniken zum Beispiel mittels der Manipulation geometrischer Elemente die Formgenerierung über numerische Parameter hinaus beeinflussen können.

3. Arbeitspakete

Zur Erreichung der einzelnen Ziele wurde das Projekt in mehrere Arbeitspakete (AP) gegliedert. In **AP 1** wurden Generierungsalgorithmen für die drei Maßstabsebenen entwickelt und miteinander verknüpft. In **AP 2** wurden verschiedene Analysemodule integriert, um die erzeugten Planungsvarianten nach gewünschten Kriterien bewerten zu können. In **AP 3** wurden Methoden zur Steuerung des generativen Prozesses implementiert, um den Anwendern eine intuitive Steuerung der Generierungsmodule zu ermöglichen. Im Rahmen von **AP 4** wurden Workshops und Feedbackrunden mit Partnern aus der Praxis durchgeführt, um mögliche Problemstellungen besser zu verstehen und die Praxistauglichkeit des entwickelten Prototypen zu bewerten und zu verbessern. Die einzelnen Arbeitspakete dienten dabei zur inhaltlichen Strukturierung der einzelnen Aufgaben. Im Projektverlauf stellte es sich als sinnvoll heraus, die zeitliche Abfolge flexibel zu gestalten. So wurden mehrere Feedbackrunden bereits während der Entwicklung der generativen Logik durchgeführt.

3.1 AP1 Generative Komponenten

Im Rahmen dieses Arbeitspakets wurden verschiedene generative Komponenten entwickelt. Durch das Konzept der drei Maßstabsebenen wurde gewährleistet, dass die Komplexität jedes Abschnitts handhabbar blieb und im Projektverlauf kontinuierlich Zwischenziele mit eigenständigen Ergebnissen umgesetzt werden konnten. Dabei wurden die für die Implementierung der Komponenten wichtigen Kriterien in Workshops und Feedbackrunden zusammen mit dem Praxispartner DIPLAN [9] erarbeitet (siehe AP 4).

3.1.1. Gebäudekubatur

Auf der ersten Maßstabsebene der Gebäudekubatur bestand die Problemstellung darin, flexible Platzierungsalgorithmen für Kubaturen für verschiedene Grundstücksurrisse zu erzeugen. Die dafür entwickelte Generierungsmethode basiert auf der Replikation gängiger Typologien für Wohngebäude. Die meisten Bauten im Bereich des Wohnbaus lassen sich in Typologien kategorisieren [10], [11], [12]. In diesem Projekt haben wir uns auf drei Haupttypen konzentriert: Block, Zeile und Solitär (Abbildung 3). Weitere Gebäudeformen können durch Kombinationen dieser drei Typen erzeugt werden.



Abbildung 3
Drei grundlegende Wohngebäudetypologien: Block (Blockrandbebauung), Zeile und Solitär.

Unser Framework ermöglicht es, diese drei Typologien in ihrer Grundform zu replizieren und weitere Variationen aus diesen Grundformen zu erzeugen. Dadurch können wir vielseitige und baubare Entwurfsvarianten erstellen. Zur Generierung werden verschiedene Inputparameter benötigt, welche die Kontextfaktoren sowie auch Generierungsparameter als Anforderungen an den Wohnbau übermitteln. Am Ende der Generierung werden die Bebauungsvarianten in Form eines 3d Modells pro Bebauungsvariante gespeichert und für die weiteren Analysen & Simulationen bereitgestellt. (Abbildung 4)

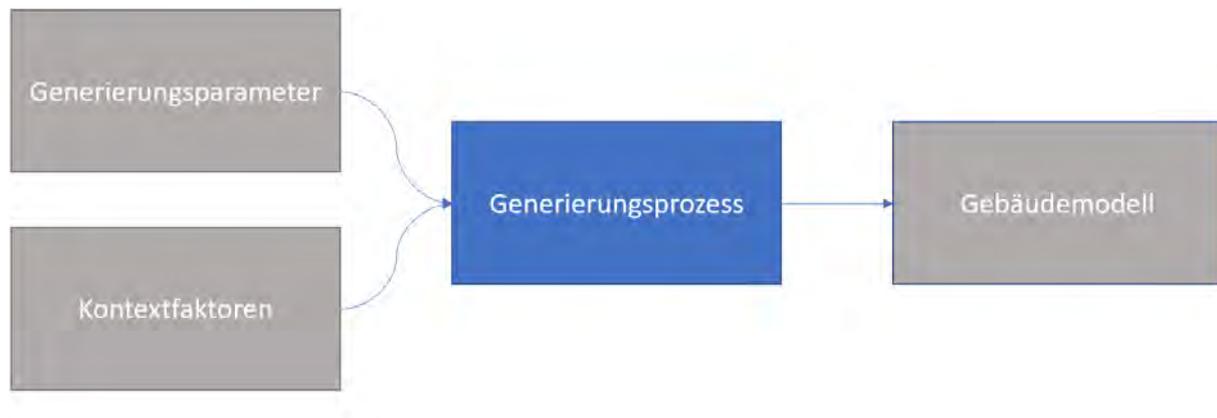


Abbildung 4
Darstellung des Generierungsprozesses.

3.1.1.1 Inputs

Die für die Generierung erforderlichen Inputs können in zwei Gruppen eingeteilt werden: Generierungsparameter und Kontextfaktoren, dabei bestehen die Kontextfaktoren aus geometrischen Daten (Kontextgeometrie) und nicht-geometrischen Daten (Bebauungsregeln). Die Kontextgeometrie wird durch das Grundstück und, falls verfügbar, die zusätzlichen Informationen wie Straßen, öffentlichen Räume und die Nachbarbebauung repräsentiert. Das Grundstück muss vom Nutzer selbst angelegt werden, da bislang keine bundesweiten digitalen Katasterdaten zu Flurstücken zur Verfügung stehen. Um diesen Prozess zu vereinfachen wurde ein webbasiertes Interface von unserem Praxispartner DIPLAN GmbH erstellt. Nach Eingabe der Adresse werden durch dieses Werkzeug zunächst alle relevanten Kontextfaktoren aus Openstreetmap [13] in einer Karte angezeigt (Abbildung 5). Daraufhin kann der Nutzer den Grundstücksumriss seines Wohnbauprojektes mit einer Polyline in die Karte einzeichnen. Ein vektorbasierter Umriss wird im nächsten Schritt als eine Rhino3d-kompatible Datei erzeugt und heruntergeladen. Falls die Grundstücksumgebung bei Openstreetmap vorhanden ist, wird die Kontextgeometrie in einem bestimmten Radius vom Grundstückszentrum automatisch nachgeladen, um später mehr Informationen für kontextsensitive Analysen bereitzustellen. Wenn allerdings in Openstreetmap keine Daten zu den Gebäudehöhen hinterlegt sind, werden alle Nachbarbauten mit einer Höhe von 3 m versehen. Die richtige Höhe muss dann manuell in Rhino3d hinzugefügt werden (Abbildung 6).

Es ist durchaus möglich, dass die Daten von Openstreetmap die Nachbargeometrien nicht korrekt bzw. gar nicht abbilden. In diesem Fall muss die Integration der geometrischen und semantischen Daten manuell durch den Nutzer erfolgen. Beispielweise können – falls vorhanden – die City0GML Daten zur Modellierung der umliegenden Gebäuden genutzt werden. Ebenso können Festlegungen aus Bebauungsplänen hinterlegt werden.

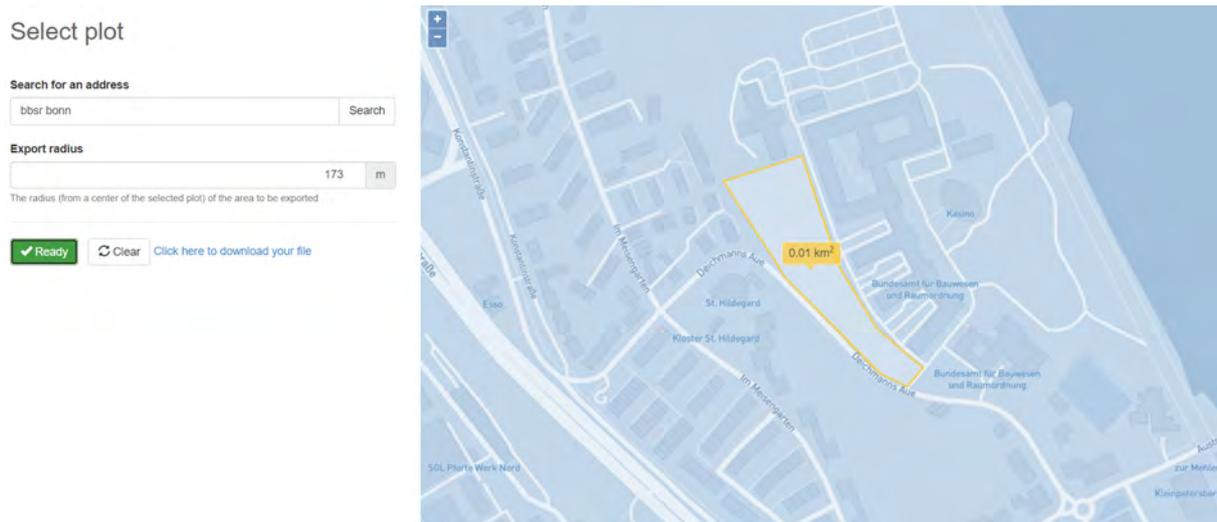


Abbildung 5
Interface zum Zeichnen des Grundstücksumrisses und zum Herunterladen von Kontextgeometrien [25].

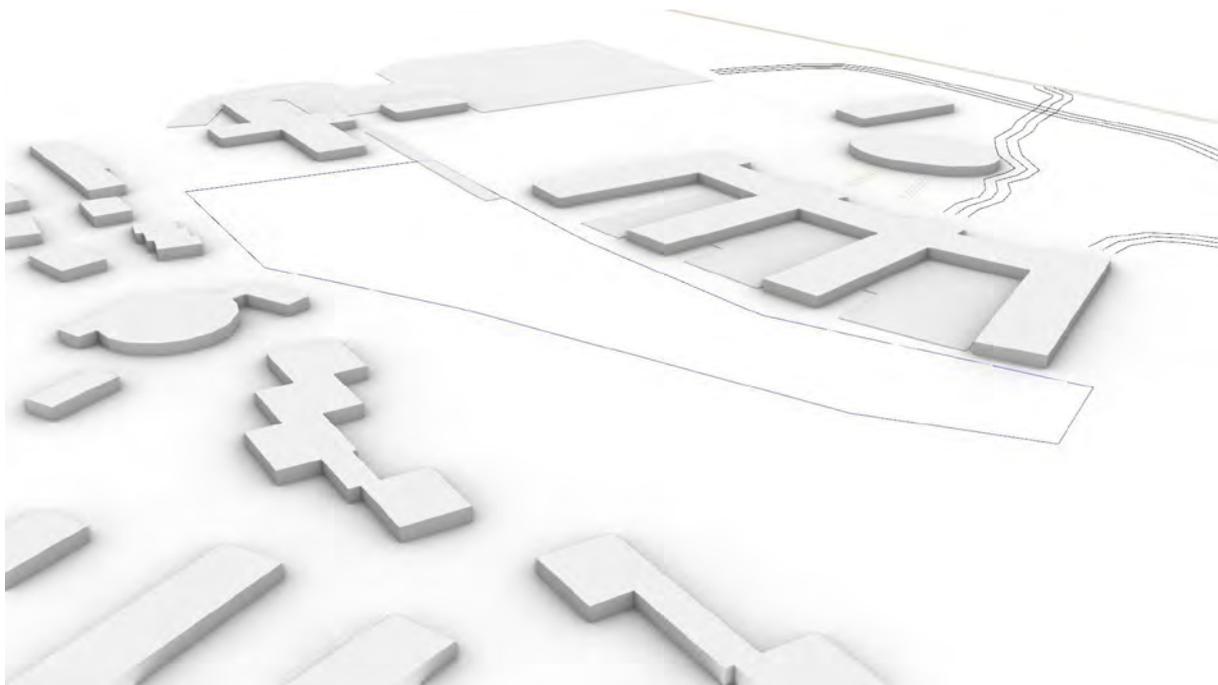


Abbildung 6
Umgebungsdatei geöffnet im Rhinoceros 3d. Aufgrund mangelnder Daten zur Gebäudehöhe werden alle Nachbarbauten nur auf 3 m Höhe extrudiert. Weitere Anpassungen müssen vom Nutzer durchgeführt werden.

Die Kontextgeometrie wird im Weiteren für die Bewertung der Generierungsergebnisse benötigt. So ist es beispielsweise üblich für das Genehmigungsverfahren eines Wohnprojekts dessen Einfluss auf die angrenzende Bebauung zu prüfen (z. B. durch eine Verschattungs- oder Tageslichtanalyse). Ziel einer solchen Analyse ist es nachzuweisen, dass sich die Menge des Tageslichteinfalls für die Nachbarn aufgrund der neuen Planung nicht übermäßig mindert. Weiterhin wird die Kontextgeometrie zur Überprüfung der „Einbindung in die Umgebung“ benötigt [14].

Zu den Kontextfaktoren zählen auch die ortsabhängigen Bebauungsvorschriften. Von diesen haben wir uns im Projekt vorrangig mit der Berücksichtigung der lokalen Abstandsflächen-Koeffizienten beschäftigt. Anhand der Landesbauordnungen wurden Tabellen mit Koeffizienten für die Abstandsflächen für verschiedene Baugebiete erstellt. Als Abstandsfläche definiert das deutsche Bauordnungsrecht die Fläche vor den Bauwerken, welche von der Bebauung frei gehalten werden soll [15]. Grundsätzlich müssen die Abstandsflächen auf dem Grundstück selbst liegen. Die Tiefe der Abstandsflächen bemisst sich als Multiplikation der Wandhöhe mit dem Abstandsflächen-Koeffizient $x=0.2...1.0$.

Nachdem der Nutzer ein entsprechendes Baugebiet ausgewählt hat, wird der Abstandsflächen-Koeffizient automatisch berechnet und für die weitere Generierung verwendet um sicherzustellen, dass die Entwürfe gemäß den örtlichen Bauvorschriften genehmigungsfähig bleiben. Obwohl die Abstandsflächen in der Regel auf dem Grundstück selbst liegen müssen, ist es möglich diese bis zur Mitte öffentlicher Flächen zu verlegen. Dabei muss jedoch gesichert sein, dass diese nicht überbaut werden. Eine weitere Ausnahme bildet eine Baulinie auf dem Bebauungsplan, womit die Gebäudeposition im Voraus bestimmt wird. Solche Baulinien können benutzt werden, um bei Lückenschließung eine ununterbrochene Fassadenfront zu erstellen. Bei der Generierung werden dementsprechend die Kontextdaten dazu verwendet, mögliche Freiheitsgrade für die Platzierung der Abstandsflächen abzuleiten (Abbildung 7)

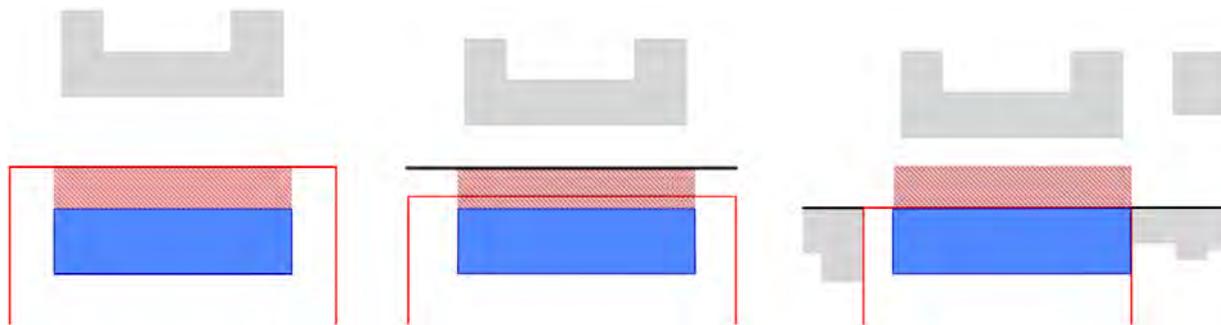


Abbildung 7

links – grundsätzliche Anwendung der Abstandsflächen, mitte – Verlegung der Abstandsflächen bis zur Mitte des öffentlichen Raums, rechts – nicht Einhaltung der Abstandsflächen aufgrund von vorhandener Baulinie

Die Generierungsparameter bestimmen zudem wichtige Gestaltungsparameter und erlauben es, den Lösungsraum weiter einzuzugrenzen. So kann der Nutzer einen bestimmten Wertebereich für die Gebäudetiefe eingeben, um unökonomische (sehr schmale) oder schlecht belichtete Gebäudeformen (mit hoher Gebäudetiefe) zu vermeiden. Auf dieselbe Weise kann die maximale und minimale Geschoszahl im Vorfeld eingegrenzt werden. Auch die Höhe eines Geschosses kann vor der Generierung festgelegt werden. Weitere Inputs, welche die Interaktion im Sinne qualitativ raumbezogener Faktoren ermöglichen, werden in den folgenden Abschnitten im Detail erklärt.

Nach der Inputs Angabe wird die Generierung der Typologien durchgeführt, dabei kann nur eine Typologie pro Grundstück ausgewählt werden. Bei der Blocktypologie wird erst Grundform Blockrandbebauung – erstellt, welche weiter mit verschiedenen Manipulationen geändert wird um diverse Bebauungsvarianten zu produzieren. Bei der Zeilentypologie kann der Nutzer verschiedene Anordnungsmuster entlang der Grundstücksränder ausprobieren. Die Solitär Typologie wurde nach ersten Tests in Absprache mit dem Praxispartner DIPLAN nicht weiterentwickelt, da diese Typologie die kleinste Dichte erzeugt und nur gering bei der Entwicklung urbaner Quartiere eingesetzt werden kann.

3.1.1.2 Datenstruktur und Aktionen

Durch eine standardisierte geometrische Repräsentation mit der Bezeichnung *BuildingInstance* kann der generative Prozess in einzelne Aktionen gegliedert werden. Aktionen bezeichnen in diesem Projekt die Manipulationen einer Grundform und wurden als modulare Grasshopper-Komponenten entwickelt. Jede der Aktionen führt eine spezifische geometrische Veränderung der Gebäudeform aus und kann entweder direkt auf eine geometrische Grundform angewandt oder in einer Sequenz an Aktionen eingereiht werden. Alle Aktionen können in beliebiger Reihenfolge in einer Generierungssequenz kombiniert werden. Dadurch wird die Flexibilität des Generierungssystems gewährleistet und dem Nutzer erlaubt, den Verlauf der Generierung nach seinen Wünschen zu beeinflussen und anzupassen. Zudem kann die Wirkung einer Aktion über eine Reihe veränderbarer Inputparameter manuell gesteuert werden. Jede Aktion benötigt neben diesen Parametern die *BuildingInstance* als Input und gibt die manipulierte Geometrie in diesem standardisiertem Format aus.

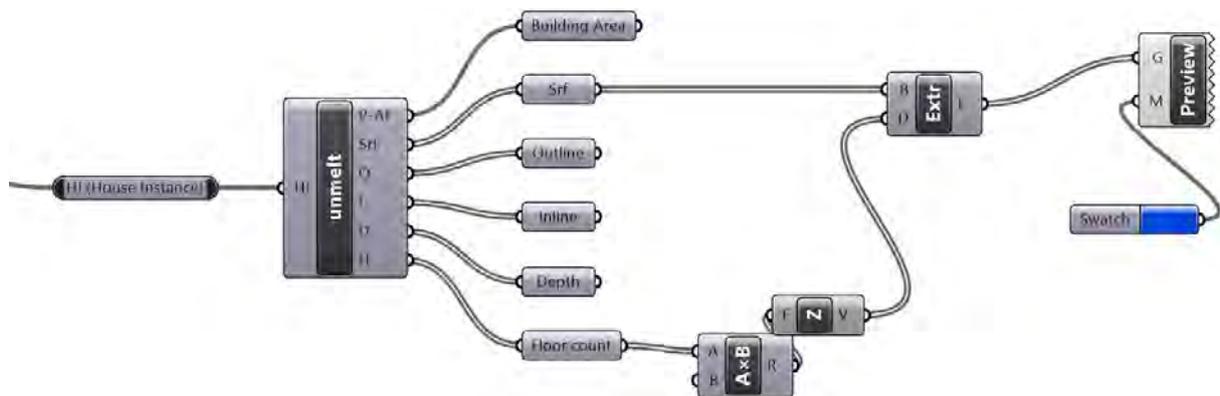


Abbildung 8
Die Darstellung der Geometrie – Parameter „BuildingInstance“ und dessen Bestandteile.

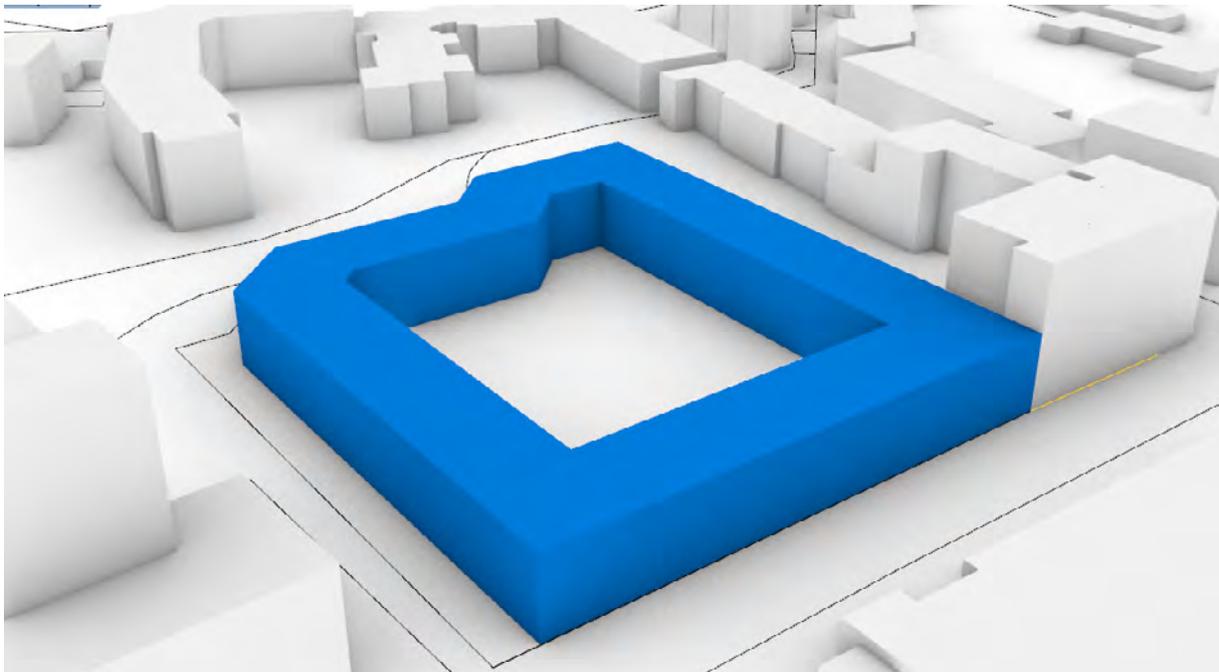


Abbildung 9
Darstellung einer Bebauungsvariante, welche mit 2 Parametern vom „BuildingInstance“ erzeugt werden kann.

Die *BuildingInstance* selbst besteht aus sechs Parametern: Bebaubare Fläche (geschlossene Polyline), Grundflächenform (Surface), Außenfassadenkontur „Outline“ (Polyline), Innenfassadenkontur „Inline“ (Polyline), Gebäudetiefe (Zahl) und Geschosszahl (Ganzzahl) (Abbildung 8). Aus der Grundfläche und Geschosszahl wird im nächsten Schritt eine geometrische Darstellung der Kubaturen erzeugt (Abbildung 9). Die bebaubare Freifläche ist für die weiteren Manipulationen wichtig, um die Abstandsflächen weiterhin frei von der Bebauung zu halten. Fassadenkonturen und Gebäudetiefen sind Hilfsinformationen, die ebenfalls in der *BuildingInstance* zwischengespeichert werden, um den Berechnungsaufwand in folgenden Schritten zu minimieren.

3.1.1.3 Blocktypologie

Mit dem Generierungsmodul „Blocktypologie“ wird ein geschlossener Gebäudeblock (Grundform) – generiert. In einem ersten Schritt wird der Grundstücksumriss um die gewünschte Gebäudetiefe eingerückt (Offset). Abhängig von der gewünschten Gebäudehöhe wird die notwendige Abstandsfläche berechnet und die bebaubare Fläche für die weitere Generierung erstellt. Es ist möglich, in einem Generierungsprozess gleichzeitig mehrere Gebäudetiefen und Gebäudehöhen auszuprobieren, indem der Nutzer anstatt einer Zahl einen Wertebereich festlegt. Nachdem die Grundform erstellt wurde, kann der Nutzer verschiedene Aktionen anwenden und die Grundform verändern (Abbildung 10).

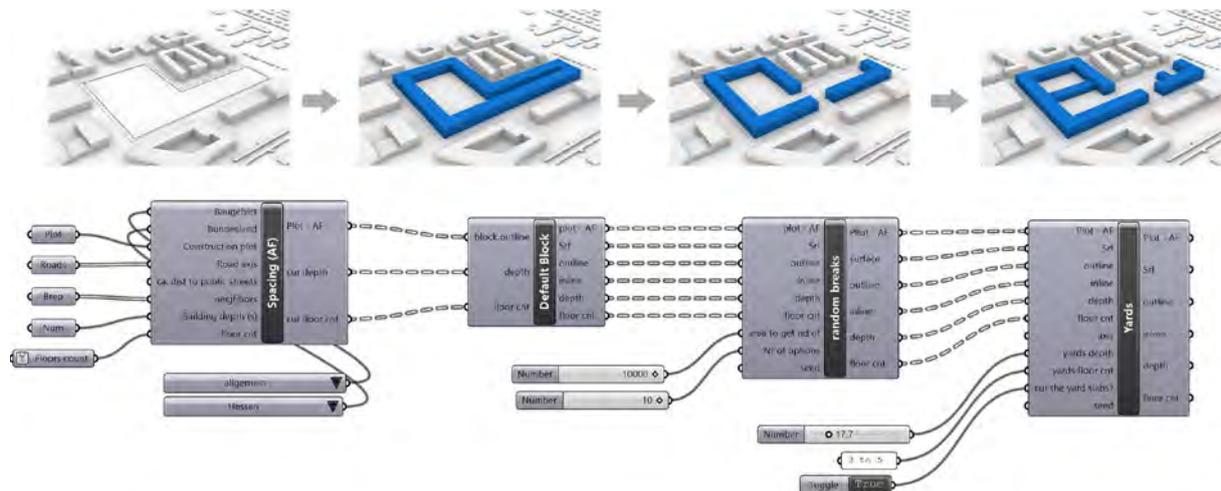


Abbildung 10
Vereinfachte Darstellung der Generierung: 1. Abstandsflächenermittlung – 2. Basic Block – 3. Aktion zur Formmanipulation A – 4. Form-Manipulation B.



Abbildung 11
Die Aktion „Random Breaks“ erzeugt Durchgänge im Blockrand an verschiedenen Stellen, wobei die Reduktion der Fläche gleich bleibt.

Jede Manipulation der ursprünglichen Blockform führt dazu, dass die Gesamtbebauungsfläche sich entweder vergrößert oder verkleinert. Dieser Flächengewinn oder -verlust kann im Grundriss erfolgen, indem die Grundfläche erweitert oder ein Teil davon rausgeschnitten wird. Außerdem kann die Flächen- und Formänderung in der Höhe erfolgen, indem die Geschoszahl von Gebäudefragmenten geändert wird. Die meisten Aktionen führen eine spezifische Manipulation an mehreren Stellen eines Blocks durch und erzeugen damit mehrere Varianten pro Grundform, wobei die Flächenveränderung konstant gehalten werden kann (Abbildung 11).

Neben der *BuildingInstance* verfügt jede Aktion über eine Reihe numerischer und geometrischer Inputs, wodurch die Generierung vom Nutzer interaktiv gesteuert werden kann (Abbildung 12). Numerische Inputs beinhalten hauptsächlich Flächeninhalte, die während der Manipulation hinzugefügt oder abgezogen werden. Da eine Aktion mit derselben Auswirkung auf die Geschossflächenzahl in verschiedenen Blöcken ausgeführt werden kann (Abbildung 13), führt jedes Aktionsmodul die Formmanipulation automatisiert an mehreren Positionen des Blocks durch und generiert verschiedene Varianten gleichzeitig. Dabei wurde darauf geachtet, dass nur valide und dem Vorgabewert entsprechenden Varianten als Ergebnis zurückgegeben werden. Wenn der Nutzer eine genaue Vorstellung zur Lokalisierung einer Aktion hat und das Ausprobieren zufälliger Optionen nicht wünscht, kann er mittels geometrischer Inputs Einfluss auf die Position einer Aktion nehmen.

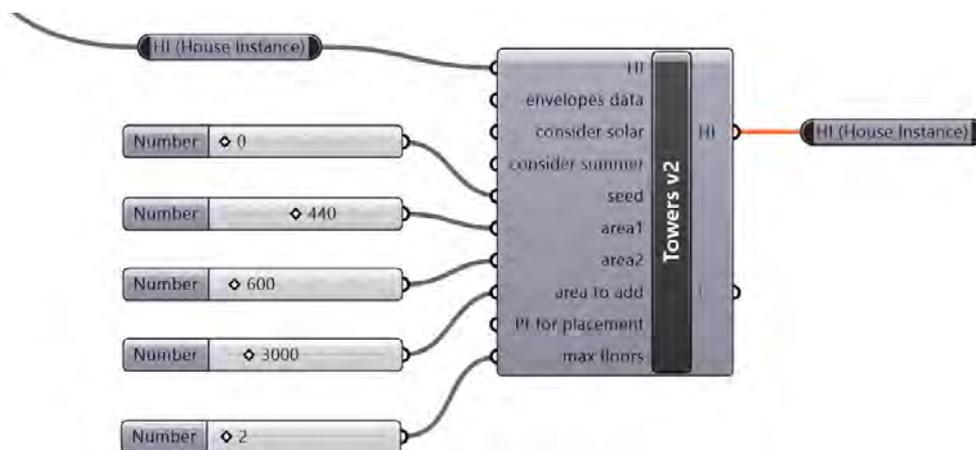


Abbildung 12
Beispiel für die Steuerung eines Aktion-Moduls („Towers“).



Abbildung 13
In diesem Beispiel wird die Aktion „Towers“ dargestellt. Der Nutzer kann entweder die genaue Platzierung des Turms mit dem Punkt-Attraktor bestimmen oder aber der Aktion erlauben, verschiedene Optionen entlang des ganzen Gebäudekontour zu generieren.

In Tabelle 1 sind die möglichen Input-Formate zur Steuerung der Aktionen aufgelistet:

Tabelle 1
Überblick über die verschiedenen Interaktionsmöglichkeiten von Aktionen.

Typ des Nutzer-Inputs	Graphische Erläuterung
Attraktor-Punkt zur genauen Platzierung der Aktion Beispiel: Platzierung von Rücksprüngen („Setbacks“).	
Attraktor-Linie zur genauen Platzierung der Aktion. Beispiel: Platzierung der Innenhofbebauung.	
Numerische Werte zur direkten Veränderung der Geometrie. Beispiel: Tiefe der Dachterrassenbebauung.	

3.1.1.4 Übersicht der Aktionen zur Block-Erzeugung.

3.1.1.4.1 Open Edge

Diese Aktion öffnet den geschlossenen Block, indem sie einen Rand oder mehrere Ränder des Umrisses löscht. Damit können sinnvolle Bebauungsvarianten für konkave und schmale Grundstücke erzeugt werden, bei denen ein geschlossener Block zu Formüberschneidungen führt (Abbildung 14).

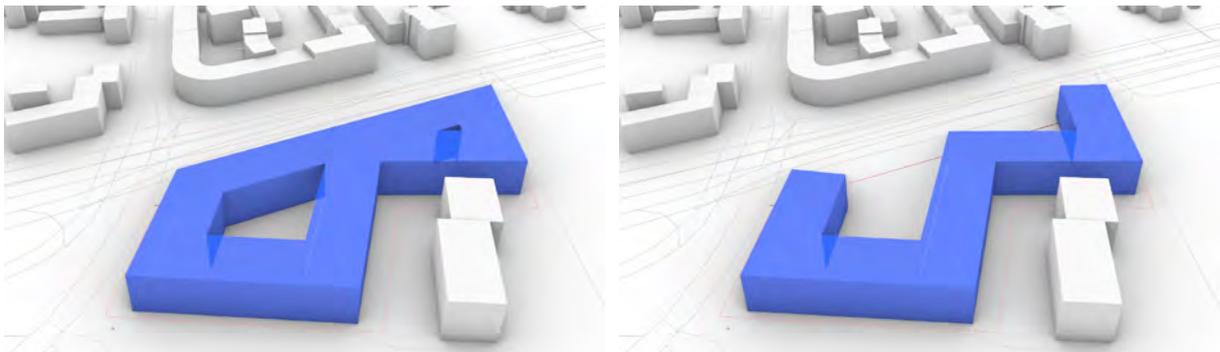


Abbildung 14
Auswirkung der Aktion „Open Edge“ auf eine geschlossenen Block.

Inputparameter	Parametertyp	Auswirkung
Edge index:	Zahl	Bestimmt, von welchem Grundstück-Segment das Löschen der Segmente startet. Wenn hierzu keine Angaben kommen, versucht die Aktion einmal von jedem Rand anzufangen.
Nr of edges to delete:	Zahl	Anzahl der Segmente, die gelöscht werden sollen.
Simplify short edges	Boolscher Wert	Bestimmt, ob kurze Segmente mit Längeren vereint werden sollen. Dadurch können unnötige Wiederholungsvorgänge vermieden werden, denn kurze Segmente haben möglicherweise nur einen geringen Einfluss auf die resultierende Form.

3.1.1.4.2 Setbacks

Diese Aktion erstellt Rücksprünge (Abbildung 15) an den Ecken des Blocks oder entlang der Blockränder um zum Beispiel einen öffentlichen Platz zu erstellen, die Monotonie langer Fassaden zu unterbrechen, die GFZ zu vergrößern oder den Innenhof des Blocks zur Bebauung zu nutzen.



Abbildung 15
Auswirkung der Aktionen „Setbacks Corners“ (mitte) und „Setbacks Edges“ (rechts).

Inputparameter	Parametertyp	Auswirkung
Dimensions range	Zahl	Minimale und maximale Dimension für die Erstellung eines Rücksprungs. Die genaue Zahl aus der angegebenen Bandbreite wird mit dem Zufallsgenerator gewählt.
Point index	Zahl	Bestimmt die Platzierung des Rücksprungs. Wenn dieser Wert nicht definiert ist, versucht die Aktion Rücksprünge an jedem Rand/jeder Ecke zu erstellen, ohne dass es zu Konflikten (Überlappung von Gebäudevolumen) kommt.

3.1.1.4.3 Random Breaks

Diese Aktion erstellt Durchgänge im Block, um zum Beispiel die GRZ zu reduzieren, um Verbindung von Straßennachsen und Blickachsen zur Kontextbebauung zu ermöglichen, um lange monotone Fassaden zu unterbrechen und um komplexe Formen zu vereinfachen (Abbildung 16)

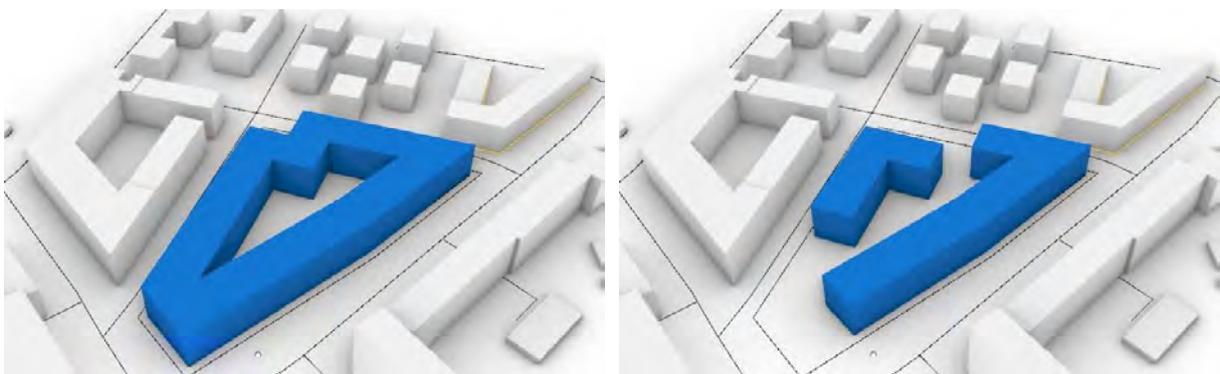


Abbildung 16
Auswirkung der Aktion „Random Breaks“.

Inputparameter	Parametertyp	Auswirkung
Area	Zahl	Geschossflächenzahl zum Löschen, wobei dieselbe Zahl mit einem größeren oder mit mehreren schmalen Durchgängen erreicht werden kann.
Nr. of options	Zahl	Die Anzahl an Varianten, die getestet werden soll. Wenn z. B. 6 Varianten pro Block entstehen sollen, wird diese Aktion versuchen, zweimal einen breiten Durchgang sowie auch zweimal zwei Durchgänge und anschließend zweimal drei schmale Durchgänge zu platzieren und dabei immer dieselbe Flächenzahl zu beseitigen. An welchen Stellen genau die Durchgänge platziert werden, wird mit einem Zufallsgenerator entschieden.
Seed	Zahl	Kontrolliert den Zufallsgenerator.
User Input	Linie	Erlaubt es dem Nutzer, die genaue Platzierung der Durchgänge zu bestimmen.

3.1.1.4.4 Cornerbreaks

Diese Aktion versucht, alle Ecken des Blocks aufzulösen (Abbildung 17). Das ist bei komplexen Grundstücksformen nützlich und hilft, die Ecken zu minimieren, die später auf Grundrissebene schwierig zu lösen sind (z. B. Belichtung, Zugang).

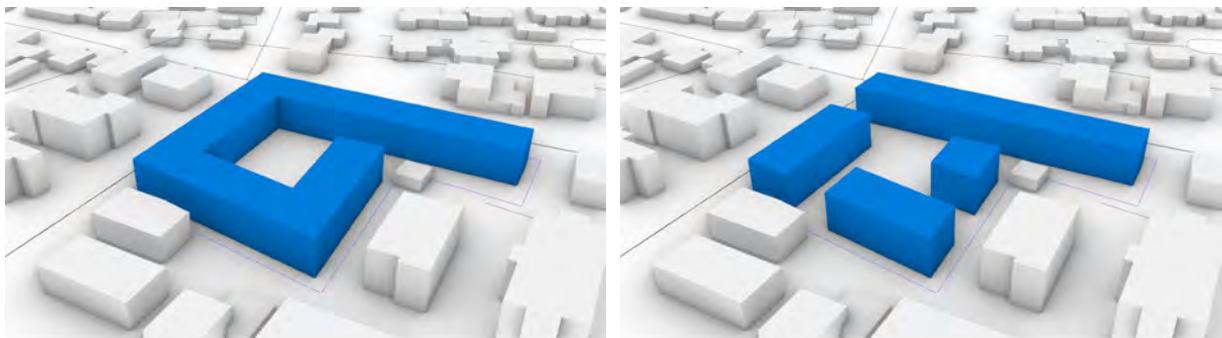


Abbildung 17
Auswirkung der Aktion „Cornerbreaks“

Inputparameter	Parametertyp	Auswirkung
Width	Zahl	Bestimmt, wie breit die Eckdurchgänge sein sollen.

3.1.1.4.5 Reduce Height

Diese Aktion schneidet die vorhandene Form zuerst an den Ecken und unterteilt damit die Gebäudegrundfläche in kleinere Fragmente. Fragmente, welche weniger Grundfläche als

200 m² aufweisen, werden gelöscht (Abbildung 18). Danach wird eine bestimmte Geschossflächenzahl gelöscht indem die Geschosszahl bei manchen der Fragmenten reduziert wird. Um die Geschossflächenzahl möglichst exakt zu erzeugen, werden die Optionen systematisch durchsucht, indem jedes der Fragmente bei einem Durchlauf bestehen bleibt und nur zwischen den restlichen Fragmenten entschieden wird, welche verkleinert werden müssen. Daraus ergibt sich, dass die Anzahl der getesteten Optionen = X^2 entspricht, während X die Anzahl der Fragmente bezeichnet.

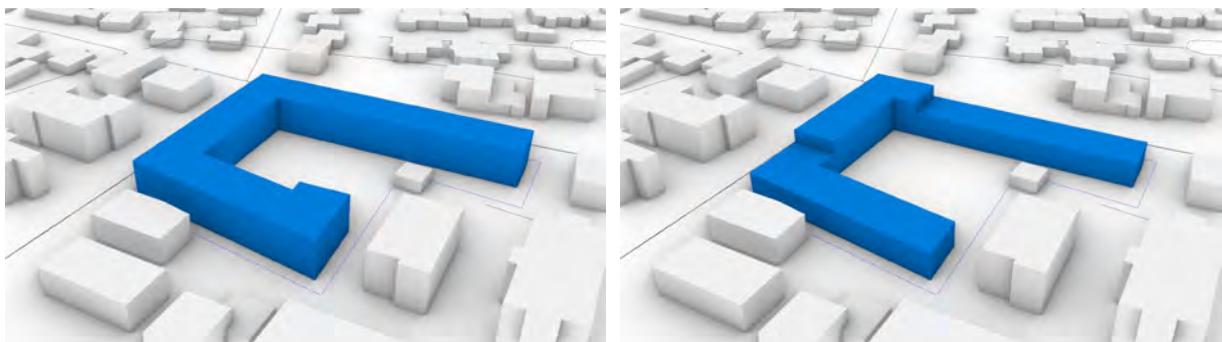


Abbildung 18
Auswirkung der Aktion „Reduce Height“.

Inputparameter	Parametertyp	Auswirkung
Area to delete	Zahl	Die Geschossflächenzahl, die subtrahiert werden soll.
Cut at corners	Boolscher Wert	Es legt fest, ob die Unterteilung in Fragmente direkt an der Ecke oder mit einem Abstand vom Eckpunkt stattfindet, und zwar mit welchem Abstand.

3.1.1.4.6 Yard Buildings

Diese Aktion füllt den Innenraum vom Block mit den zusätzlichen Volumina, um die GFZ zu vergrößern (Abbildung 19). Es gibt verschiedene Muster dafür, die Gebäude im Innenhof zu platzieren, hauptsächlich aber versucht die Aktion die Gebäuderiegel in der Mitte vom Innenhof bzw. bei mehreren Gebäuderiegeln diese jeweils von der Mitte des Blockrandes ausgehend zu platzieren, ohne dass sie sich überschneiden. Bei konkaven Grundrissen ist es aber durchaus sinnvoll, die Generierung anhand der vom Nutzer gezeichneten Achsen zu ermöglichen.

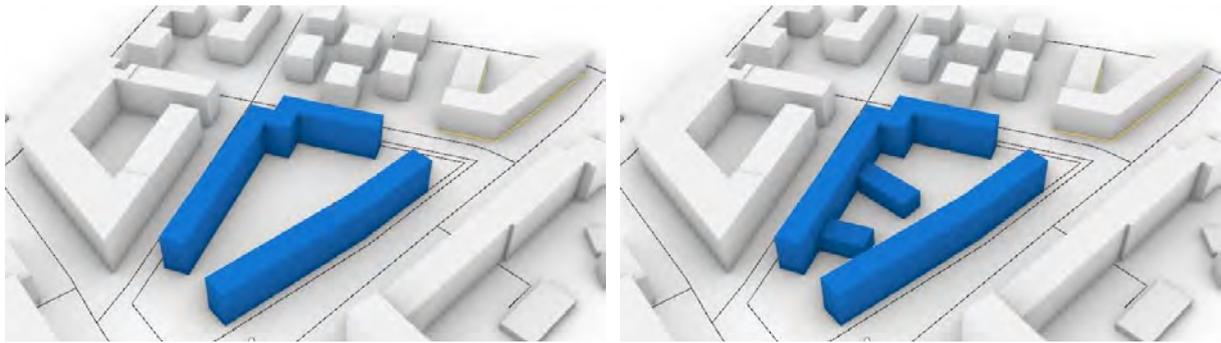


Abbildung 19
Auswirkung der Aktion „Yard Buildings“.

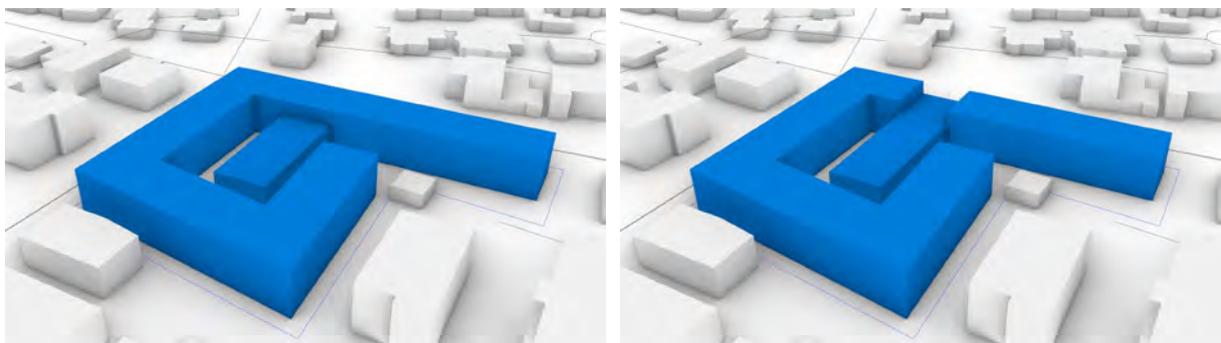


Abbildung 20
Auswirkung des Inputs „cut the yard slabs“: links – wenn es ausgewählt wurde, rechts – wenn die Innenhofbauten nicht abgeschnitten werden sollen.

Inputparameter	Parametertyp	Auswirkung
Axis	Linie(n)	Hierdurch kann der Nutzer selbst Achsen für die Gebäudegenerierung zeichnen.
Depth	Zahl	Breite der Gebäude im Innenhof
Floors	Zahl	Geschosszahl der Gebäude im Innenhof

3.1.1.4.7 Rooftops

Diese Aktion erzeugt Dachaufbauten auf dem Blockrand (Abbildung 21). Die Platzierung erfolgt automatisch, basierend auf der Straßennähe. Wenn Straßen entlang des Blockrands angrenzen, werden die Dachaufbauten nach außen gerichtet. Wenn der entsprechende Straßenrand nur eine Grenze zwischen zwei Grundstücken ist, werden die Dachaufbauten Richtung Innenhof generiert um die Abstandsflächen zwischen Nachbarn nicht zu verletzen. Die Dachaufbauten werden erzeugt, indem die Innen- oder Außenkontur auf eine bestimmte Tiefe skaliert wird.

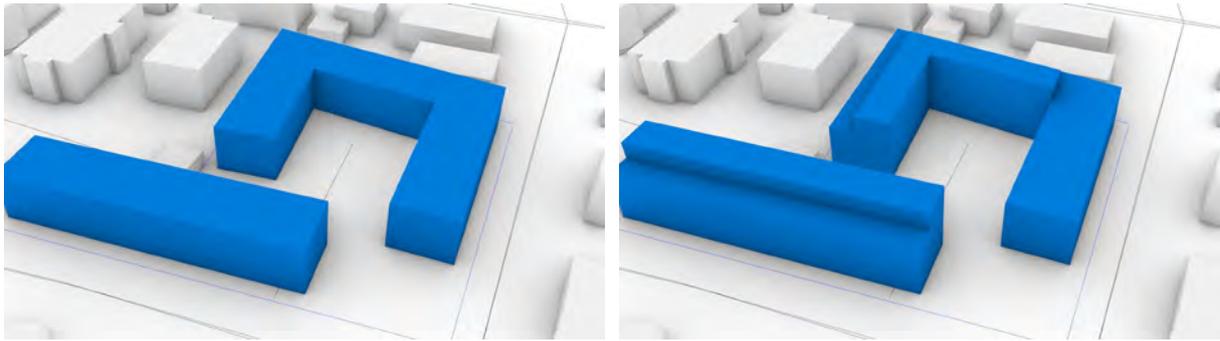


Abbildung 21
Auswirkung der Aktion „Rooftops“.

Inputparameter	Parametertyp	Auswirkung
Towards yard	Boolescher Wert	Bestimmt die Ausrichtung der Terrassen (nach innen/außen).
% length to use	Zahl zwischen 0.00 und 1.00	Diese Zahl bestimmt, wie viel % der Fassadenlänge für die Generierung der Terrassen benutzt wird.
cut corners	Boolescher Wert	Dieser Input erlaubt es, die Eckenbildung bei den Terrassen zu vermeiden.

3.1.1.4.8 Towers

Dieser Schritt platziert Türme entlang des Gebäudeblocks, indem systematisch verschiedene Positionen entlang der kompletten Blocklänge getestet werden, um eine bestimmte GRZ zu erreichen (Abbildung 22). Wenn die gewünschte zusätzliche Geschossflächenzahl mit einem Turm nicht erreicht wird, versucht die Aktion einen weiteren Turm zu platzieren. Da die Geschossflächenaufstockung die Abstandsflächen verletzen könnte, wird zuerst ein Abstandsflächen-Envelope generiert. Dabei wird ein Punktraster auf dem gesamten Grundstück erstellt. Unter Berücksichtigung der umgebenden Bebauung und der ortsüblichen Koeffizienten wird die maximale mögliche Höhe für jeden Punkt berechnet, wodurch eine Art dreidimensionale Bebauungsgrenze generiert wird. Anhand desselben Rasters werden auch Solar Envelopes [16] für die Winter- und Sommerzeit mit dem Grasshopper LadyBug Plugin [17] berechnet, um zusätzlich zu den Baugesetzen, den Faktor Sonnenlicht besser zu berücksichtigen (Abbildung 23).

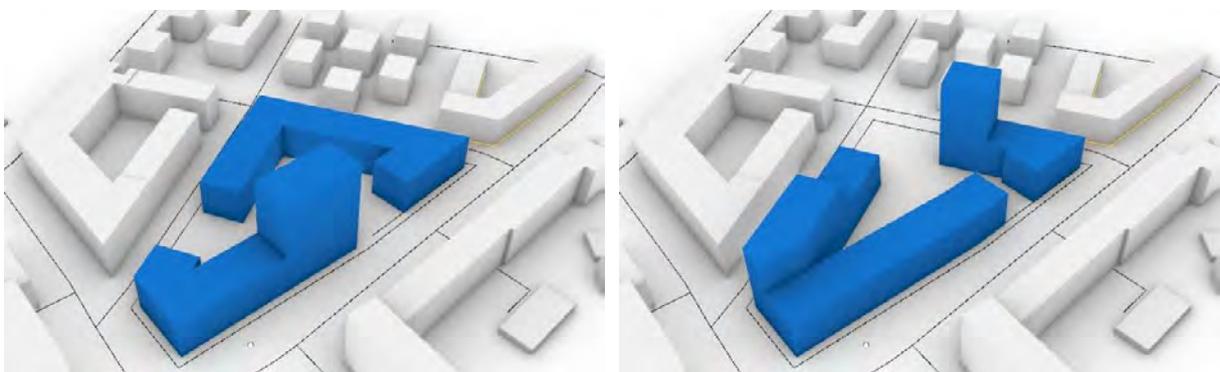


Abbildung 22
Beispiele zur Generierung von Türmen, die beide dem Abstandsflächen-Envelope entsprechen.

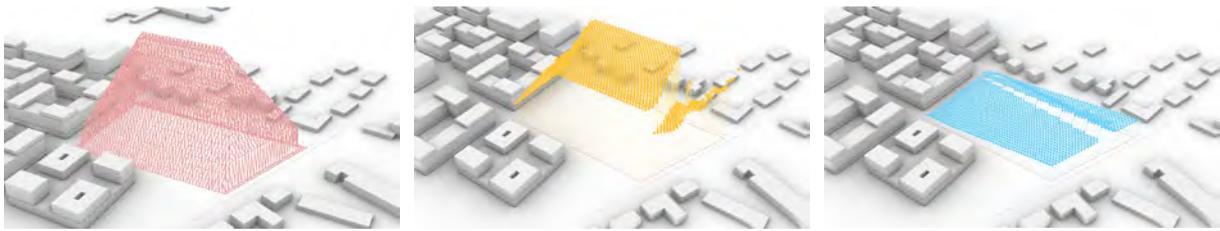


Abbildung 23
Envelopes: links – Abstandsflächen Envelope, mitte – Solar Envelope für die Sommerzeit, rechts – Solar Envelope für die Winterzeit.

Der Nutzer kann selbst entscheiden, welche dieser Envelopes kombiniert werden sollen, der Abstandsflächen-envelope bleibt aber immer als Grundlage bestehen.

Inputparameter	Parametertyp	Auswirkung
Area min & Area max	Zahl	Steuert die Grundfläche für die Turm-Generierung.
Seed	Zahl	Kontrolliert den Zufallsgenerator.
Area to add	Zahl	Hinzuzufügende Geschossflächenzahl
Placement index	Zahl	Bestimmt an welcher Gebäudekante ein Turm platziert werden soll.
Max floors	Zahl	Begrenzt die Geschosszahl für den Turm, denn möglicherweise ist die im Bebauungsplan vorgesehene maximale Gebäudehöhe/Geschosszahl kleiner als die Höhe, die sich aus den Abstandsflächen ergibt.

3.1.1.5 Zeilentypologie

Die algorithmische Logik in dieser Typologie arbeitet nicht mit der Manipulation einer Grundform, sondern mit der Platzierung verschiedener Anordnungsmuster. Damit sind Platzierungen mehrerer Zeilen auf einem Grundstück möglich. Es werden die vier Hauptbegrenzungslinien des Grundstücks identifiziert. (Abbildung 24). Damit beschränkt sich das Generierungsmuster auf vier Möglichkeiten – ausgehend von jedem Grundstücksrand kann ein anderes Muster implementiert werden.

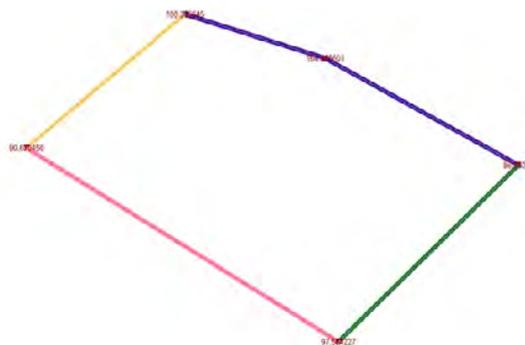


Abbildung 24
Übersetzung der beliebigen Anzahl der Grundstücksränder zu den vier Hauptbegrenzungslinien.

Von jedem Rand können nun Zeilenmuster generiert werden. Über die Reihenfolge der Ränder bestimmt der Nutzer. Es ist möglich, jedem der vier farblich hervorgehobenen Ränder ein eigenes Zeilenmuster zuzuweisen. Wenn jedoch keine freie Stellen am Grundstück bleiben, bricht die Generierung automatisch ab um Überschneidungen zu vermeiden (Abbildung 25).

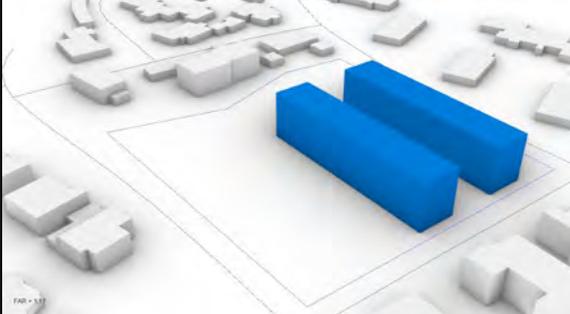
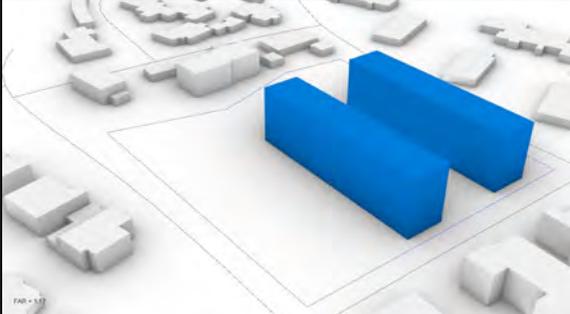
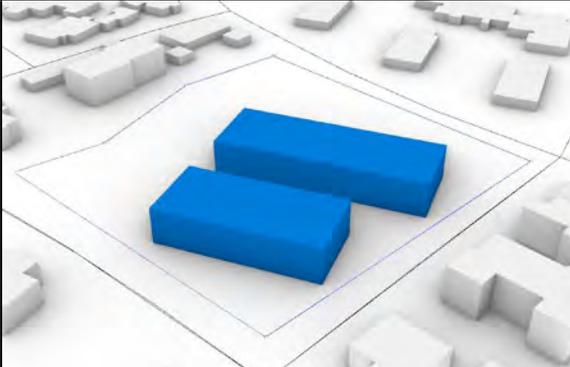


Abbildung 25
Platzierung der Zeilen entlang der Grundstücksgrenzungslinien in einer vom Nutzer bestimmten Abfolge.

Die Parameter für die Zeilen können bei jedem der Ränder unterschiedlich eingestellt werden. Die wichtigsten Parameter sind in Tabelle 2 aufgelistet:

Tabelle 2
Interaktionsparameter für die Zeilengenerierung.

Parameterbezeichnung	Demonstration der Auswirkung
Reihenfolge der Ränder für die Generierung	
Anzahl der Zeilen	
Länge der Zeilen	

Parameterbezeichnung	Demonstration der Auswirkung
Punkt-Attraktor zur Zeilenplatzierung	
Geschosszahl (dabei werden die Abstände zwischen Gebäuden automatisch vergrößert gemäß der Abstandsflächenverordnung)	
Rotation (90° Winkel lässt die Zeile entlang des Randes angeordnet werden)	
Gebäudetiefe	

Weiterhin können für die erzeugten Zeilen zusätzliche Aktionen angewandt werden, sowie Aufteilung in kleinere Abschnitte (Reihenhäuser) oder Erzeugung der Dachaufbauten (Abbildung 26).

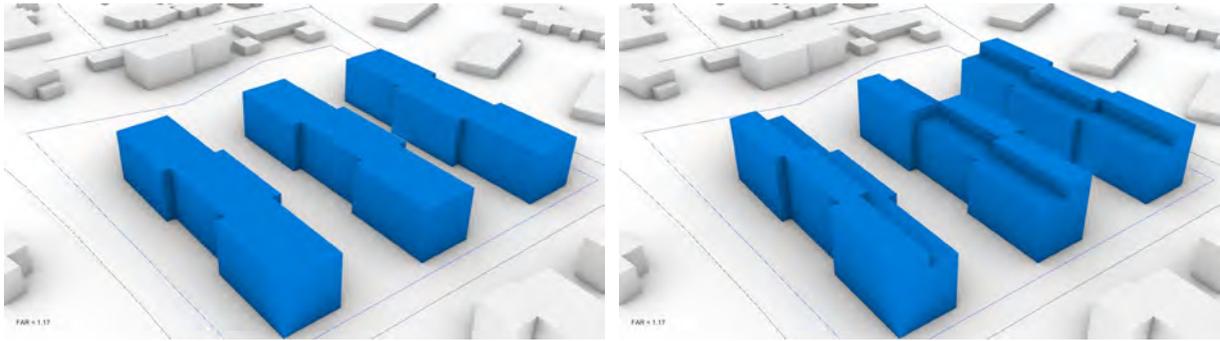


Abbildung 26

links: Unterteilung der Zeilen in kleinere Abschnitte nach gewünschter Fläche mit der Möglichkeit, diese weiter zu manipulieren. Rechts: Erzeugung der Dachterrassen bei den Zeilen.

Damit die Anordnung ähnlicher länglicher Bauriegel an einem größeren Grundstück nicht zu monotonen strukturlosen Mustern führt, wurde vor der Generierung noch ein zusätzlicher Schritt bei der Grundstücksvorbereitung eingeführt. Wenn das Grundstück eine bestimmte Fläche überschreitet wird es in mehrere kleinere Bauflächen unterteilt. Die maximale Grundstücksgröße, welche nicht weiter unterteilt werden soll, kann der Nutzer selbst bestimmen. Die Unterteilung des Grundstücks verbessert die Ergebnisse des generativen Moduls bei konkaven Grundstücken. Die kompakter unterteilten Formen können mit dieser Methode besser mit verschiedenen Zeilen gefüllt werden und liefern Ergebnisse mit besserer Flächennutzung (Abbildung 27).

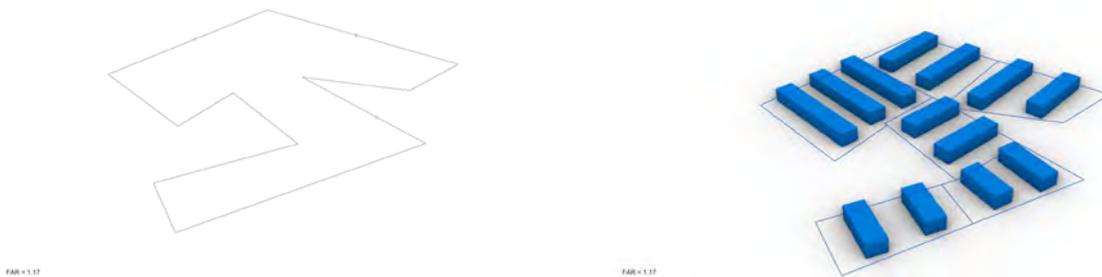


Abbildung 27

Beispiel einer Zeilengenerierung bei einem konkaven Grundstück mit einem komplexen Umriss.

Bei der Grundstücksunterteilung wird zum einen versucht, die konkaven Umriss in konvexe Polygone zu unterteilen. Dabei wird zuerst überprüft, welche Ecken große Abweichungen vom 90° Winkel haben (Abbildung 28-1). In jeder dieser Ecken wird jeweils jede der eckbildenden Linien einmal verlängert (Abbildung 28-2). Weiterhin wird überprüft, ob eines der Grundstückseckpunkte innerhalb eines definierten Abstands – zum Beispiel drei Meter – zur eckbildenden Linie liegt (Abbildung 28-3). Liegt ein Punkt innerhalb dieses Abstands, wird er mit dem ursprünglichen Eckpunkt verbunden, wodurch sich eine Schnittlinie ergibt (Abbildung 28-4). Ansonsten wird die ursprüngliche eckbildende Linie als Schnittlinie benutzt, um das Grundstück zu unterteilen. Nachdem die Abfolge für alle stumpfen Winkel durchgeführt wurde, werden die Varianten verglichen und es wird diejenige ausgewählt, welche das Grundstück in möglichst gleiche Flächen teilt (Abbildung 29).

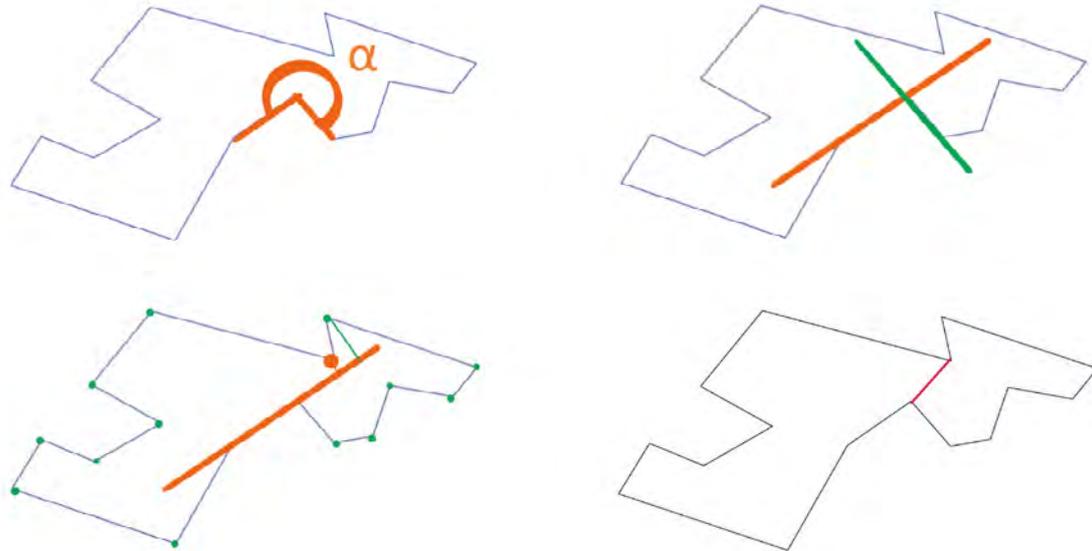


Abbildung 28
Die Schritte zur Unterteilung eines Konkaven Grundstück.

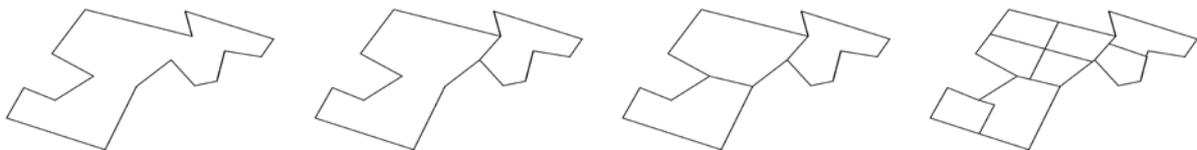


Abbildung 29
Schritte der Unterteilung vom Grundstück in kleinere weniger konkave Fragmente.

Bei sehr komplexen Grundstücken, die aus mehreren Bauflächen bestehen, würde die im vorherigen Schritt angeführte Anzahl der Parameter, die zur Kontrolle der Generierung an jedem einzelnen Baufeld notwendig ist, zu groß und unübersichtlich werden. Daher wurde die Generierung für komplexe Grundstücke vereinfacht, indem auf jedem unterteilten Baufeld die Zeilen nur entlang einer Richtung platziert werden können. Da es immer mehrere Möglichkeiten für die Ausrichtung gibt, kann der Nutzer zwischen drei automatisch erzeugten Varianten entscheiden (Abbildung 29). Mit den verbleibenden Input-Parametern kann der Nutzer über die Gebäudetiefe, die Geschosshöhe, sowie auch über die weitere Unterteilung in Reihenhäuser entscheiden.

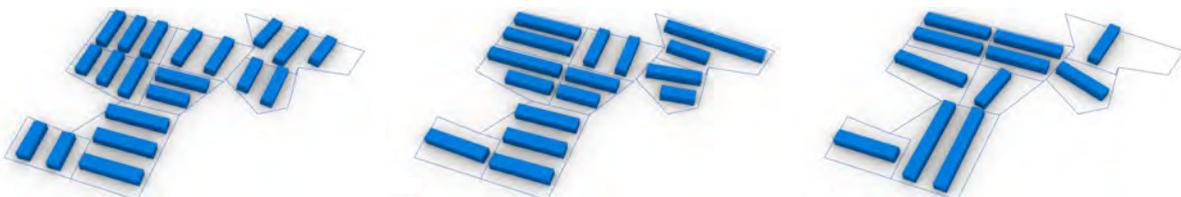


Abbildung 30
Drei Modi der automatisierten Zeilen-Generierung für ein komplexes Grundstück.

Abseits der typologiebasierten Lösung in Rhino3d/Grashopper wurde ein experimentelles Verfahren in der Programmiersprache Python umgesetzt. Das Ziel dieses Verfahrens war es für die Zukunft eine bessere Schnittstelle zu den im Python-Ecosystem verfügbaren Machine Learning Bibliotheken zu schaffen. Weiterhin eröffnet die Umsetzung eines rein programmatischen Verfahrens außerhalb der Rhino3d Umgebung, mehr Möglichkeiten zur Parallelisierung und Beschleunigung von Algorithmen. Dafür wurde ein Prototyp entwickelt, mit dem einfache Formrepräsentationen (vgl. Anhang 1) schnell mittels Parametern erzeugt werden können und mit einer schrittweisen Platzierungslogik eine Lösung erreicht wird. Um einzelne Lösungen zusätzlich visuell überprüfen zu können wurde die Renderengine der pygame Bibliothek angebunden. Die Abbildung 31 zeigt die besten Varianten aus 100 generierten Lösungen, mit den zugehörigen Inputvariablen, die mit einer zufallsbasierten Platzierungsmethode erzeugt wurden. Zufallsbasiert heißt in diesem Fall, dass Parameter zufällig aus einem bestimmten Zahlenbereich gewählt wurden. Damit die Lösungen bewertet werden können, wurde prototypisch eine einfache Analyse implementiert, welche aus einer generierten Lösung die Geschossflächenanzahl (GFZ) ableitet.

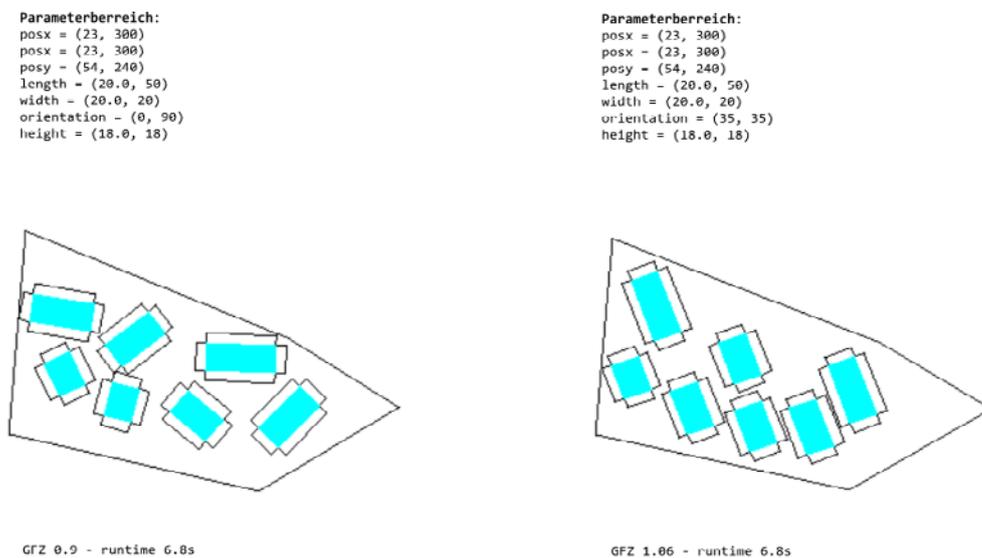


Abbildung 31 Platzierungslösungen, die mit einem Lösungsalgorithmus für zwei im Ausrichtungswinkel unterschiedliche Parameter-Sets erzeugt wurden.

Trotz des eher konzeptionellen Outputs sehen wir großes Potential darin dieses Verfahren in Zukunft weiter zu verbessern, um verschiedene generative Ansätze miteinander zu vergleichen. Weiterhin eignet sich dieses Verfahren dazu, den Einsatz von Machine Learning Algorithmen zur Lösung geometrischer Teilprobleme weiter zu erforschen.

3.1.2. Erschließung

In diesem Schritt werden die erzeugten Kubaturen in Wohneinheiten unterteilt und mit vertikalen Erschließungen versehen. Dabei ist gemäß der Erkenntnisse aus Diskussion mit dem Praxispartner DIPLAN (siehe Kapitel 3.4) der Wohnungsmix die wichtigste Vorgabe.

Im Wohnungsbau existieren grundsätzlich drei Erschließungstypologien: Spänner, Flur und Laubengang, dabei wird für die Generierungsmethode die Letztgenannte als Sonderfall der Flur-Typologie betrachtet (Abbildung 32).



Abbildung 32
Zwei Typologien für die Erschließung und Unterteilung in Wohneinheiten: Spänner-Erschließung (links) und Flur-Erschließung (rechts).

3.1.2.1 Flurtypologie

Für die Entwicklung der Flurtypologie wurde das vom Praxispartner DIPLAN entwickelte Flexhouse (Abbildung 33) als Ausgangspunkt verwendet, wobei die entwickelte Typologie bei den beliebigen Gebäudetypen anwendbar ist. Besonders bei der Flexhouse-Struktur ist, dass die vertikalen Erschließungselemente (Treppe und Aufzug) in einem relativ breiten, innenliegenden Flurbereich platziert werden, wodurch die gesamte Fassadenfläche für die Wohnungen zur Verfügung stehen. Damit dieselbe Flurtypologie jedoch auch für andere Gebäudetypen benutzbar bleibt, kann der Nutzer die Platzierung der Erschließungskerne entlang der Fassade auswählen (vergleiche Abbildung 45).

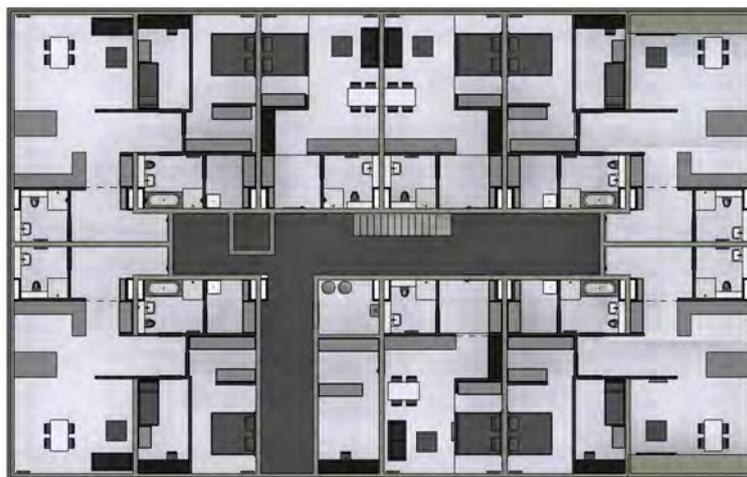


Abbildung 33
Beispielhaus mit Flexhouse-System.

Bei dem Flexhouse-System sollen alle Wohnungen 9 Meter tief sein, während die Breite der Wohnungen durch die Kombination von fünf Modulen variiert werden kann. Daraus ergeben sich fünf verschiedene Wohnungstypen (Abbildung 34). Das System ist darauf ausgelegt, die Wände und Decken größtenteils als vorfabrizierte Elemente im Betonfertigteile-Werk von EMC [17] herzustellen und vor Ort zu montieren. Die Wand- und Deckenelemente können innerhalb der maximalen Fertigungsgrößen in beliebigen Abmessungen produziert werden. Die EMC Fertigteile erzwingen keine Modulbauweise und bieten aber gleichzeitig die Vorteile der Vorfertigung (höhere Produktions- und Montagegeschwindigkeit, sowie niedrigere Kosten).



Abbildung 34
Die fünf Wohneinheiten, aus welchen das von DIPLAN GmbH entwickelte Flex-System besteht.

Außer der Basistypologie mit zwei Reihen der Wohnungen, die entlang des Flures erschlossen werden, ist es auch möglich eine längere Wohnungsreihe zum Flur zu platzieren (Abbildung 35 links) um somit die Gebäude-tiefe von 21,2 Meter auf 17,77 Meter zu reduzieren, da tieferen Gebäudevolumen nicht auf allen Grundstücken platziert werden können. Es ist außerdem möglich nur eine Reihe der Wohnungen zu erschließen, woraus sich die Laubengangtypologie ergibt (Abbildung 35 rechts).



Abbildung 35
Links: Variation der Erschließung mit Flexhouse-System, während eine Wohnungsreihe gedreht wird. Rechts: Variation der Erschließung mit Flexhouse-System, wobei nur eine Wohnungsreihe erschlossen wird und somit den Laubengang-Typ bildet.

Um die Unterteilung in die Wohneinheiten durchzuführen, ist die wichtigste Vorgabe, der vom Nutzer vorgegebene Wohnungsmix. Die Umsetzung dieses Mixes ist nicht trivial, da bei größeren Wohnungsbauprojekten ein Grundstück in der Regel mehrere Gebäude beinhaltet, die verschiedene Geschosszahlen haben. Die Wohnungen sind in der Regel aus konstruktiven Gründen in allen Geschossen gleich verteilt. Daraus folgt, dass sich der Wohnungsmix, der sich aus der geometrisch möglichen Unterteilung aller Regelgeschossflächen in Wohneinheiten ergibt, deutlich unterscheiden kann vom gesamten Wohnungsmix, der aus der einfachen Multiplizierung der Wohnungszahl mit entsprechender Geschosszahl pro Gebäude resultiert. Die Wohnungen so zu verteilen, dass sich der gewünschte Wohnungsmix ergibt, ist dementsprechend ein multivariates Optimierungsproblem, bei dem man zwischen der Minimierung der Restflächen und der Einhaltung des Wohnungsmix abwägen muss. Dazu kommt, dass geometrische Restriktionen der gewünschten Wohnungen das Problem zusätzlich verkomplizieren können. Durch die Verwendung des DIPLAN Flexhouse-Systems wird die Problemkomplexität reduziert, da die Wohnungen dieses Systems alle dieselbe Tiefe und rechteckige Grundformen haben (vergleiche Abbildung 34). Mit dieser Eingrenzung können wir die Wohnungsbreite pro Wohnungstyp als fixen Parameter nutzen, was die Berechnung deutlich vereinfacht. Dadurch können wir den Umriss des Gebäudes als eine Linie betrachten und das Platzierungsproblem entlang der außenliegenden Fassadenlänge lösen.

Um dieses Verfahren zu implementieren werden bei einem Gebäude zuerst die Fassadenlinien (Abbildung 36, links) sowie innenliegende Ecken erkannt, welche dunkle unbeleuchtete Raumbereiche bilden, wo keine Wohnungen platziert werden können. In diesen Ecken werden Treppenhäuser platziert. Im nächsten Schritt werden die Gebäude unter Berücksichtigung der BauGB in Brandabschnitte unterteilt (Abbildung 36, rechts).



Abbildung 36
links: Das Erkennen der Eck-Fragmente, rechts: Aufteilung in die Brandabschnitte.

Weiterhin werden je nach Erschließungstypologie, die für die Erschließung notwendigen Flure gebildet. Bei der Laubengangtypologie wird dazu eine der Fassadenlinien, abhängig von der Nutzerangabe, nach innen versetzt (Abbildung 37). Bei der Mittelflurerschließung werden beide Fassadenlinien nach innen versetzt und bilden somit einen innenliegenden Flur und zwei Wohnungsreihen (Abbildung 38).



Abbildung 37
Flurbildung für die Laubengangtypologie



Abbildung 38
Bildung eines innenliegenden Flures.

Nachdem die Konturen für die Wohnungsreihen innerhalb des Gebäudes definiert sind, beginnt die Unterteilung in einzelne Wohnungen. Das dafür entwickelte Modul „Building Divider“ – benötigt vom Nutzer zwei Inputs:

- Gewünschter Wohnungsmix in Prozenten (Proportionen)
- Wohnungsgrößen entsprechend den im Wohnungsmix angegebenen Wohnungstypen

Da man den gewünschten Wohnungsmix fast nie exakt in der Gebäudekubatur erreichen kann, werden die drei nächstmöglichen Proportionen ausgesucht. Zum Beispiel kann eine Proportion von 60 %–30 %–10 % für einen Gebäuderiegel von 100 Meter Länge niemals durch die verfügbaren Wohnungstypen mit den Breiten von 10, 20 und 40 m erreicht werden. Der nächstmögliche Wohnungsmix wäre für diesen Block dementsprechend 50–10–40 %. Daher erfolgt zuerst eine Berechnung, welche die Anzahl der Wohnungstypen bestimmt, die im gesamten Gebäudeabschnitt untergebracht werden können. Jede Wohnungsmix-Variante erzeugt eine eigenständige geometrische Lösung für die Wohnungsverteilung (Abbildung 39). Durch dieses Vorgehen kann gewährleistet werden, dass bei der Wohnungsaufteilung keine oder nur geringe Restflächen übrig bleiben.



Abbildung 39
Anpassung des ursprünglichen Wohnungsmixes.

Im nächsten Schritt werden die Wohnungstypen nach der berechneten Verteilungsproportion den jeweiligen Gebäudeabschnitten zugewiesen. Dabei müssen die Geschosshöhen der einzelnen Gebäude berücksichtigt werden: wenn eine Wohnung eines bestimmten Typs im Erdgeschoss eines Abschnittes platziert wird, wird sie aus konstruktiven Gründen in den oberen Geschossen darüber platziert. Daraus folgt, dass der Wohnungsmix von den Gebäudehöhen stark beeinflusst wird und diese daher bei der Berechnung der Proportionen berücksichtigt werden muss.

Der Algorithmus für die Unterteilung eines Gebäudes in Wohnungen funktioniert nach der Brut-Force-Methode: am Anfang wird eine Liste von erforderlichen Wohnungen aller Typologien erstellt. Bei jeder Iteration wird versucht, alle möglichen Wohnungen eines Typs an der nächsten freien Stelle zu platzieren. Dabei können manchmal nicht alle Gebäudeabschnitte „befüllt“ werden. Zum Beispiel kann ein Gebäuderiegel mit einer Länge von 100 Metern nicht ohne Restflächen mit 11 Meter breiten Wohnungen bestückt werden.

Nachdem alle Wohnungstypen verteilt wurden, können die Restflächen für jeden Gebäudeabschnitt berechnet werden. Da die Berechnung für alle möglichen Wohnungskombinationen relativ lange dauern kann, wird eine zulässige maximale Restfläche festgesetzt. Sobald die gesamte Restfläche überschreitet wird, wird der Prozess abgebrochen (Abbildung 40). Letztendlich werden alle berechneten Varianten miteinander verglichen und die Variante mit der geringsten Restfläche wird ausgewählt und für die Platzierung der Wohnungstypen verwendet.

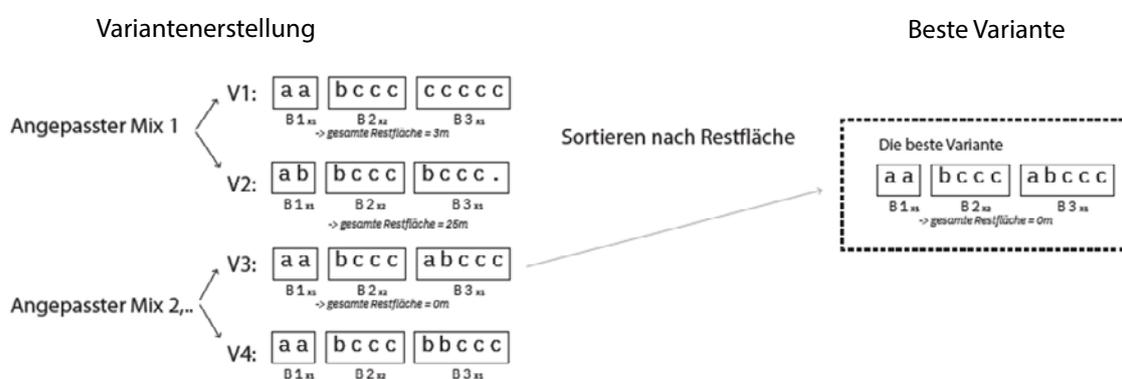


Abbildung 40 Anwendung möglicher Abfolgen der Wohnungskombinationen, um den gewünschten Wohnungsmix zu erreichen.

Die gewünschten Wohnungen werden im Anschluss in den zugehörigen Gebäude- bzw. Brandabschnitten platziert. Dabei werden auch die Restflächen sichtbar (Abbildung 41), die jedoch in einem späteren Schritt den Nachbarwohnungen zugeordnet werden.

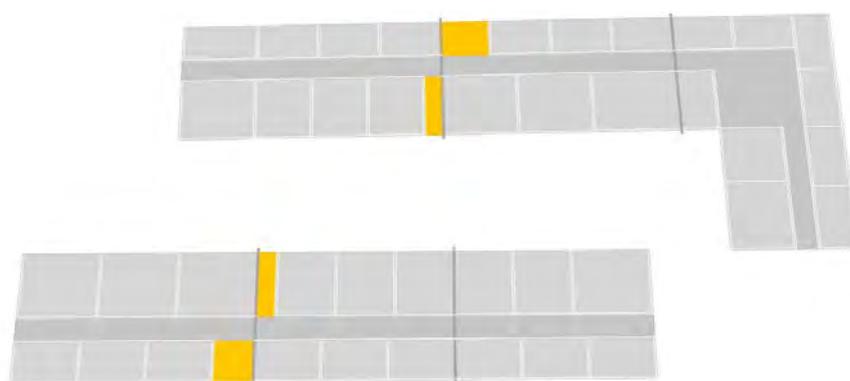


Abbildung 41 Platzierung der festgelegten Wohnungskombination in der Gebäudeform. Die gelben Schraffuren zeigen Restflächen.

Im nächsten Schritt werden je nach Nutzer-Angabe (Abbildung 42) die Eingänge im Erdgeschoss platziert. Der Nutzer kann dabei die Eingangsbreite sowie die Orientierung zum Innenhof oder zur Straße auswählen.

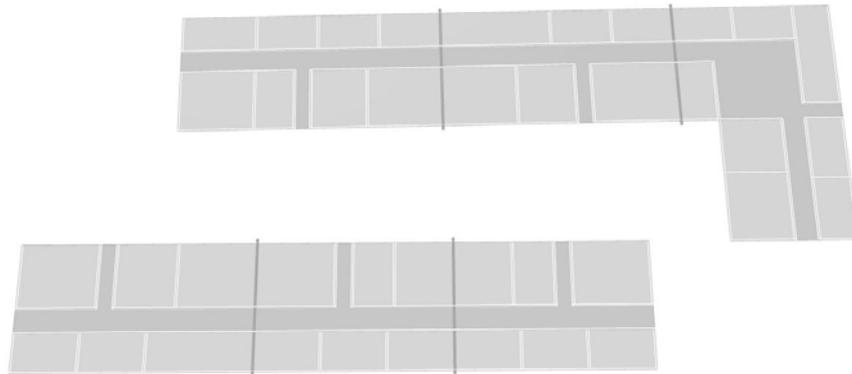


Abbildung 42
Erstellung der Grundgeschoss-Eingänge.

Danach werden die Eingangstüren der einzelnen Wohnungen in den Plan eingefügt. Weiterhin werden die Treppen und die Aufzüge so gewählt, dass ausreichend Abstand zu den Wohnungseingängen bleibt (Abbildung 43).

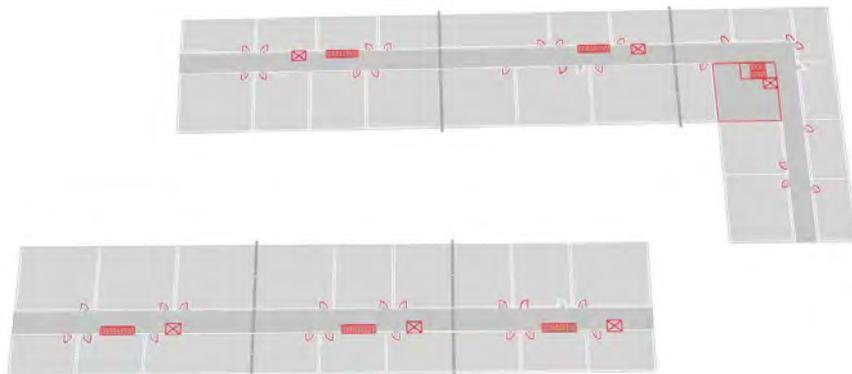


Abbildung 43
Platzierung der Türen, Treppen und Aufzüge.

Im letzten Schritt werden nicht notwendige Flurbereiche in jedem Gebäudeabschnitt entfernt und die entstandenen Flächen auf die angrenzenden Wohnungen verteilt (Abbildung 44).

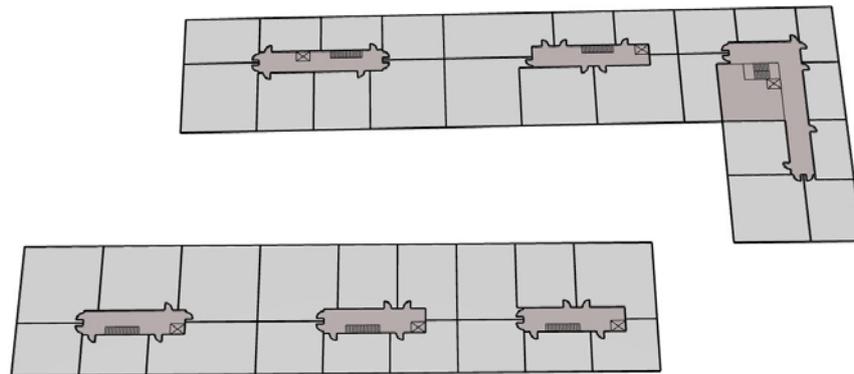


Abbildung 44
Reduzierung der Flurbereiche und Verteilung der gewonnenen Fläche auf Wohneinheiten.

Ferner wurde die Erschließungstypologie so erweitert, dass der Nutzer über einen Parameter die innenliegenden Flure an die Außenfassade anschließen kann (Abbildung 45). Dabei wird der zusätzlich benötigte Platz für die Flurbereiche bereits bei der Platzierung der Wohnungen berücksichtigt.

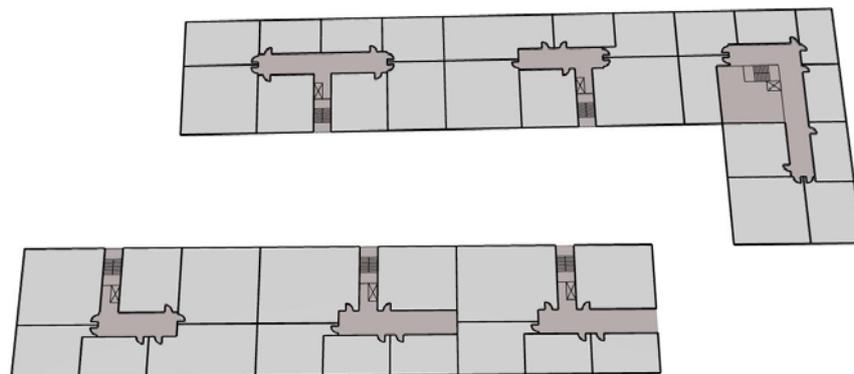


Abbildung 45
Treppenhäuser entlang der Fassade als eine mögliche Variation der Flur-Erschließung.

3.1.2.2 Spänner-Typologie

Bei dieser Typologie werden die Wohnungen in jedem Abschnitt möglichst ohne Flur an die Treppenhäuser angeschlossen. Bei dieser Erschließungstypologie können Wohnungen rechteckig oder L-förmig sein. Dadurch steigt der Komplexitätsgrad bei der Minimierung der Restflächen unter Einhaltung des vorgegebenen Wohnungsmixes. Selbst unter Ausschluss der Formbestimmungs- und Platzierungsaufgaben ist der Lösungsraum dabei durch $x^{w \cdot g}$ definiert. Wobei x die Anzahl der möglichen Wohnungen eines Typs innerhalb der Grundfläche beschreibt, g die Anzahl der einzelnen Gebäude und w die Anzahl der Wohnungstypen. Bei einem Standardfall von $g=5$ Gebäuden, $w=5$ verschiedenen Wohnungstypen und einer möglichen Anzahl von platzierbaren Elementen jedes Wohnungstyps von $x=50$ gibt es damit 50^{25} mögliche Lösungen. Je nach Fallbeispiel können diese Werte

für x, g und w noch höher angesetzt werden. Daher lassen sich nicht alle Lösungsmöglichkeiten in tragbarer Berechnungszeit testen. Der Lösungsraum muss daher mittels Heuristiken nach guten Lösungen durchsucht werden. Im Rahmen des Projekts wurde ein Modul entwickelt, das diese Berechnung mit Hilfe des Greedy Search Algorithmus [18] löst. Durch schrittweise Platzierung und die stetige Wahl der Platzierungsmöglichkeit, die das Gesamtergebnis verbessert, findet diese Methode in überschaubarer Zeit meist brauchbare lokale Optima. Weiterhin hat die Gewichtung der beiden Zielkriterien (Minimierung der Restfläche und Minimierung der Abweichung vom Vorgabemix) starken Einfluss auf das Ergebnis. Die Qualität der Ergebnisse variiert dabei stark. Für den produktiven Einsatz sollten hier auch alternative Heuristiken in Betracht gezogen werden.

Um diese Typologie beispielhaft umzusetzen, wurde der allgemeine Wohnungsmix als Input mit dem Wohnungsmix pro Gebäudeabschnitt übernommen. Der Nutzer kann also die Anforderungen an den Treppenkern (innenliegend/außenliegend, zum Innenhof oder zur Straße, Treppenproportionen) sowie die Anzahl und die Größe der zu platzierenden Wohnungen pro Treppenkern festlegen. Weiterhin wird der gesamte Erdgeschossriss in möglichst gleiche Fragmente unterteilt (Abbildung 46), die im Nachhinein mit den gewünschten Wohnungen gefüllt werden.



Abbildung 46
Unterteilung einer komplexen Form in möglichst gleiche Fragmente.

Um die konkaven Gebäudevolumina in Abschnitte zu unterteilen, wird ein mehrstufiges Verfahren benutzt. Im ersten Schritt werden die Gebäudeecken behandelt (Abbildung 47). Weiterhin werden alle geraden Gebäudeteile in Fragmente von einem Meter Breite unterteilt. Die größeren Eckfragmente werden in eine positionsbezogene geordnete Liste eingefügt. Diese Liste wird genutzt, um zu prüfen, wie die entsprechenden Brandschutzabschnittsgrenzen gesetzt werden müssen, um möglichst gleichmäßige Gebäudeabschnitte zu erzeugen. Dabei hat die Wahl des Ausgangspunktes eine entscheidende Auswirkung auf das Ergebnis der Aufteilung in die einzelnen Abschnitte. Deshalb wird der Ausgangspunkt in mehreren Durchläufen verschoben, um die gleichmäßigste Unterteilungsvariante zu finden. Dafür wird die Minimierung des akkumulierten Flächenunterschieds zwischen den einzelnen Brandschutzabschnitten als Zielkriterium genutzt.

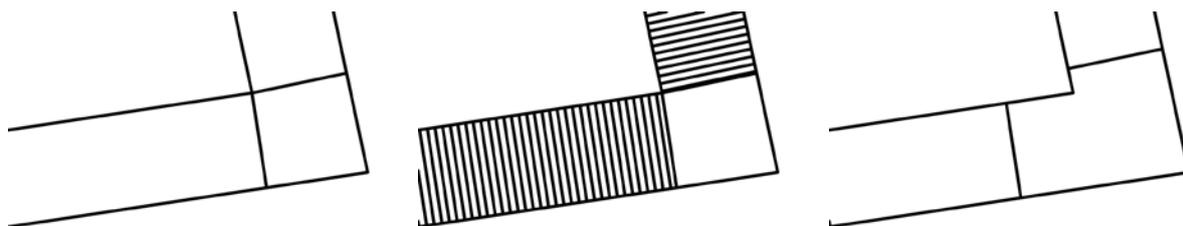


Abbildung 47
Demonstration der Methode zur Aufteilung einer beliebigen Form in die Fragmente möglichst gleicher Fläche.

Abschließend werden die gewünschten Wohnungstypen in den Abschnitten platziert. Dabei werden die Treppenhäuser nach der Nutzerangabe orientiert. Bei einer Nord-Süd Orientierung wird immer versucht, die Treppenhäuser an der nördlichen Fassade zu platzieren. Dieser Schritt ist der planerischen Praxis entlehnt und gewährleistet, dass die kleineren Wohnungen nach Süden orientiert sind und dadurch gut belichtet werden. So ist es möglich, in einem Gebäudeabschnitt zwei bis vier Wohnungen zu platzieren, unabhängig davon, ob die Treppe innen (unbelichtet) oder außen (belichtet) liegt (Abbildungen 48–50).

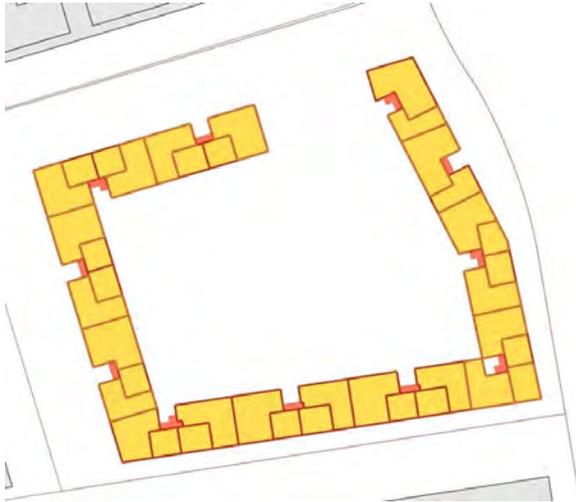


Abbildung 48
Erschließungen mit außenliegendem Vierspänner.



Abbildung 49
links: Erschließungen mit außenliegendem Dreispänner, rechts: Erschließungen mit außenliegendem Zweispänner



Abbildung 50
links: Erschließungen mit innenliegendem Zweispänner, rechts: Erschließungen mit innenliegendem Dreispänner

3.1.3. Grundrisse

Für diesen Schritt wurde eine Lösung des Praxispartners DIPLAN in Kooperation mit der ETH Zürich entwickelt. Die Aufgabe der Grundrissgenerierung ist ein multivariates Optimierungsproblem, bei dem die einzelnen Kriterien gewichtet werden müssen. Die Optimierung zielt darauf ab, dass die Wohnungsform restlos mit Zimmern gefüllt wird und deren Größen und Beziehungen den Zielkriterien eines gegebenen Raumprogramms entsprechen. Zusätzlich müssen auch Anforderungen an die natürliche Belichtung erfüllt wer-



Abbildung 51
Schematische Darstellung der Suche an passende Grundrisse mittels Machine Learning Algorithmen.

den. Um diese Methode zur Grundrissgenerierung zu testen, wurde eine Grundrissbibliothek aus 271 Wohnungs-Layouts erstellt, welche neben Grundrisslayout auch Daten zu Flächen, Zimmeranzahl, Außenwände und Erschließung beinhaltet. Mit der erstellten Grundrissbibliothek als Datengrundlage wurden mittels einer auf *Machine Learning* (ML) basierten Methode [19] Grundrisse für neue Wohnungsformen generiert. Als ML Algorithmus wurden vor allem Self Organizing Maps (SOM) verwendet. Dabei werden die einzelnen Grundrisse und ihre Kennzahlen nach Ähnlichkeit der Formeigenschaften (z. B. Umriss) und numerischen Eigenschaften (z. B. Grundfläche) geordnet. Für die Erzeugung neuer Grundrisse werden innerhalb der geordneten Datenstruktur einer SOM die ähnlichsten bereits bekannten Lösungskandidaten ermittelt und das Ergebnis als Näherung zwischen diesen Kandidaten erzeugt (Abbildung 51). Das entwickelte Verfahren ist damit bis zu einem gewissen Grad robust genug, um mit beliebigen Wohnungssumrissen umzugehen. Allerdings ist es für eine praktische Anwendung notwendig den Algorithmus mit einer größeren Grundrissbibliothek zu trainieren, um bessere Lösungen zu erhalten.

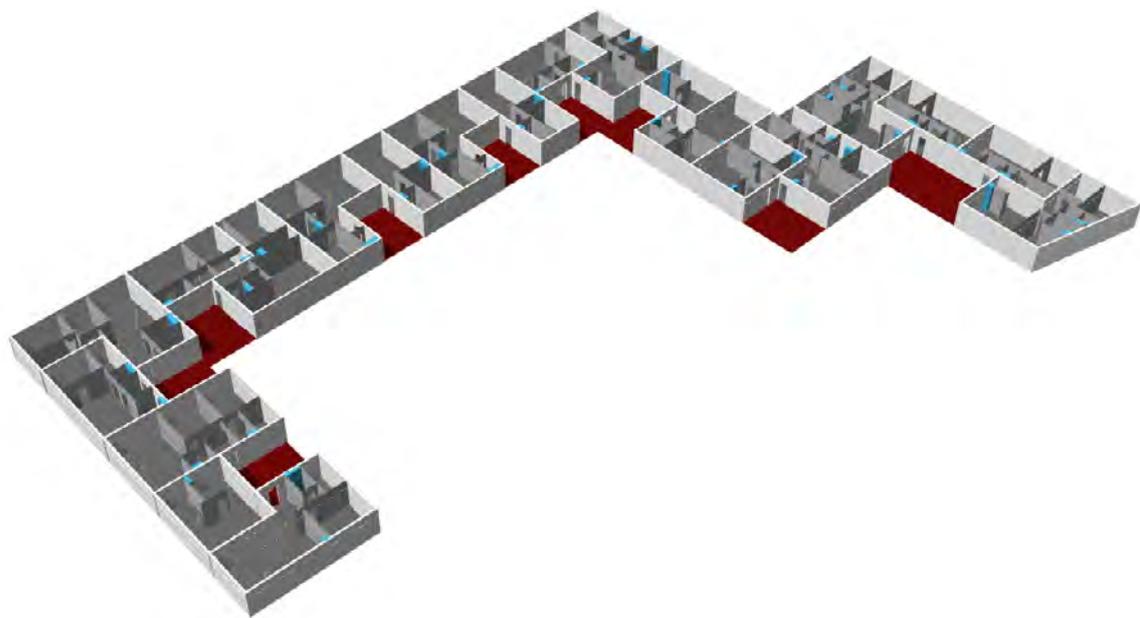


Abbildung 52
Prototypische Generierung der Grundrisse mittels Machine Learning Algorithmus in den parametrisch generierten Wohnungssumrissen.

Beim Testen des Prototyps wurde als noch zu lösendes Problem beispielsweise festgestellt, dass bei unbekanntenen Ausgangssituationen mehrere Innenwände nicht an die tragenden Wände angeschlossen werden (Abbildung 53).



Abbildung 53
Darstellung eines mit ML Algorithmen generierten Grundrisses, was aufgrund der nicht genau zu der Suchanfrage passenden Referenz aus der Bibliothek eine Lücke zwischen Wänden aufweist.

3.2 AP2 Analysewerkzeuge

In Zusammenarbeit mit DIPLAN wurden folgende wichtige Key Performance Indicators (KPI) erarbeitet, die für die Auswertung der Bebauungsvarianten umgesetzt wurden:

- GRZ
- GFZ
- Erdgeschossfläche
- Bruttogeschossfläche
- Gesamtvolumen
- Decken als Gesamtfläche
- Wände als Gesamtfläche
- Baukosten (anhand der Flächenpauschale)
- Einschätzung der Anzahl an Parkplätzen in einer unterirdischen Garage
- Maximale Anzahl der Wohnungen gemäß dem Stellplatzschlüssel
- Wohnungsmix

- Gesamtzahl der Wohnungen
- Nutzfläche

Neben der Geschossflächenzahl als eines der maßgeblichen Kriterien zur Bewertung der Varianten, werden weitere Kriterien berechnet, um Bebauungsvarianten mit gleicher oder ähnlicher Geschossflächenzahl zu differenzieren. In Tabelle 3 sind die KPIs für drei Bebauungsvarianten eines Testfalls vergleichend dargestellt:

Tabelle 3
KPIs und Visualisierung von drei verschiedenen Bebauungsvarianten.

Kriterium	Variante 1	Variante 2	Variante 3
			
			
GFZ	1.74	1.78	1.81
GRZ	0.41	0.41	0.43
Erdgeschossfläche	1.753	1.752	1.877
BGF	7.536 m ²	7.706 m ²	7.846 m ²
Deckenfläche	9.290	9.460	9.724
Wandfläche	4.959	5.121	4.778
Einschätzung Parkplätze	94	94	94
Baukosten- pauschale	15,1 Mio	15,4 Mio	15,7 Mio
Maximale Anzahl der Wohnungen gemäß Park- schlüssel	94	94	94
Tatsächliche Zahl der Wohnungen	88	84	96
Nutzfläche	74 %	72 %	76 %
Wohnungsmix dabei Input- Wohnungsmix = 45:30:25	45:31:22	32:24:28	44:32:20

Anhand dieser KPIs kann beispielsweise festgestellt werden, dass die dritte Variante etwas mehr Bruttogesamtfläche hat und entsprechend mehr Wohnungen erzeugt wurden. Allerdings überschreitet bei Variante 3 die mögliche Wohnungszahl das mögliche Maximum, welches durch die Anzahl der möglichen Parkplätze auf dem Grundstück begrenzt ist. Ein weiteres Beispiel ist, dass obwohl die zweite Variante im Vergleich zur ersten Variante mehr Fläche hat, die Nutzfläche in der ersten Variante höher ist. Mit einem solchen KPI-Vergleich wird eine solide Grundlage zur transparenten Entscheidungsfindung geschaffen.

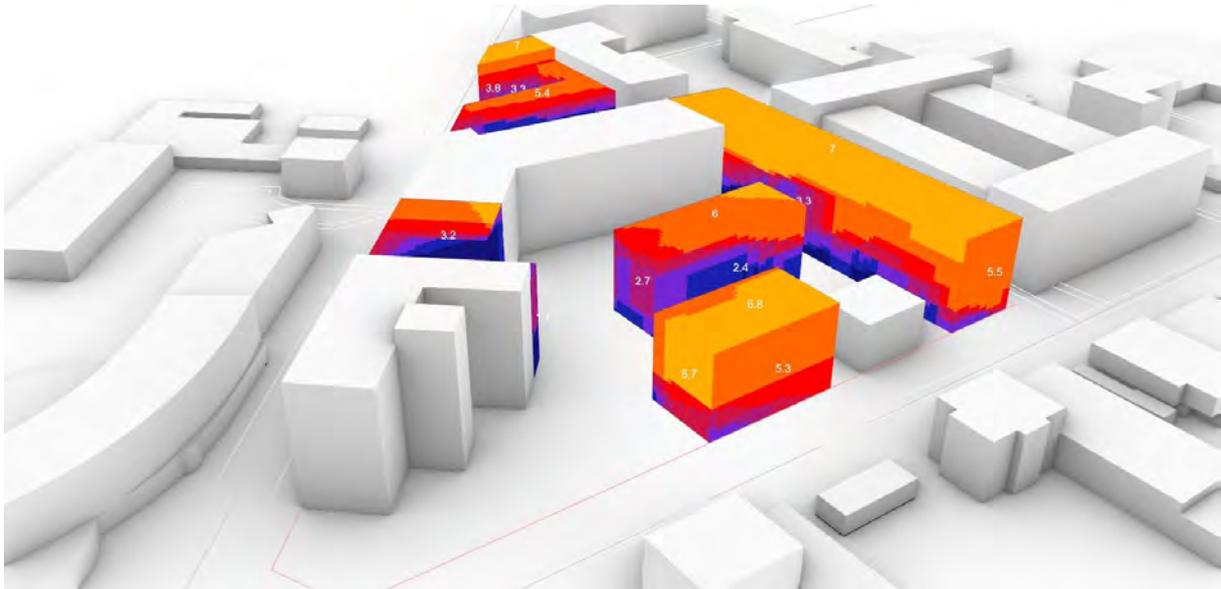


Abbildung 54
Tageslichtstudie für eine Bebauungsvariante (die Zahlen zeigen die Anzahl der Sonnenstunden pro Fassade während der Winterzeit).

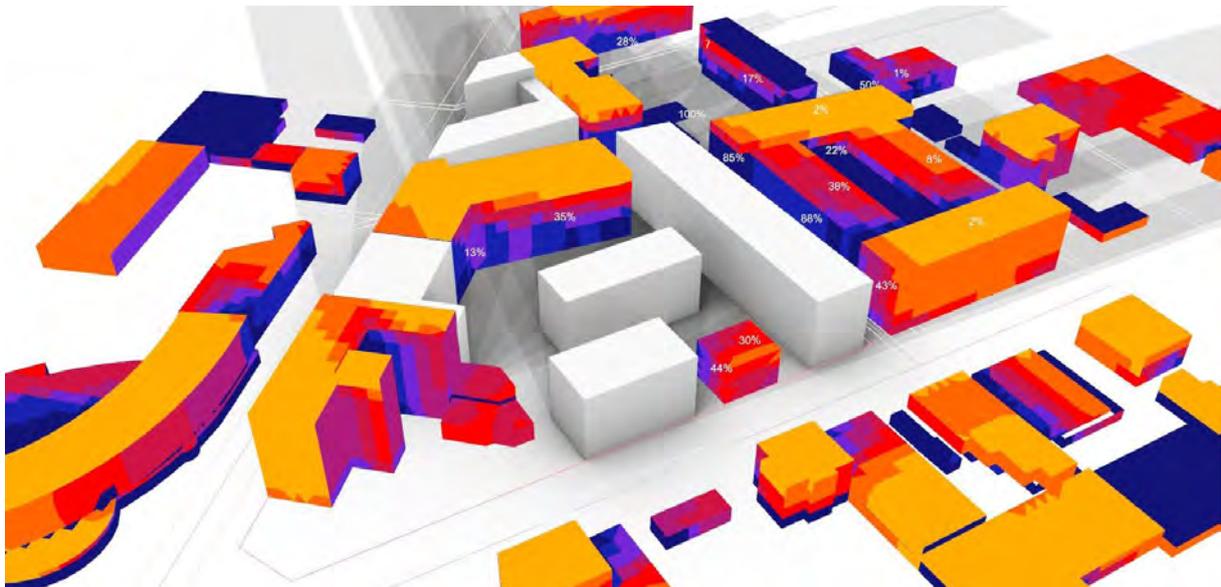


Abbildung 55
Verschattungsstudie für die Nachbarbebauung für die Winterzeit. Die Prozentzahlen zeigen die Verschlechterung der Lichtverhältnisse pro Fassadenfläche im Vergleich zur Situation ohne der neuen Bebauung.

Es wurden außerdem Vorlagen für die Tageslichtstudien mithilfe des Grasshopper-Plugins Ladybug [18] entwickelt. Damit kann zum einen die Qualität des Lichteinfalls bei neuen Bauvorhaben eingeschätzt werden (Abbildung 54), zum anderen können Verschattungsstudien durchgeführt werden, die den Einfluss vom Neubau auf die Nachbargebäude simulieren und somit Auskunft geben, wie stark sich die Lichtverhältnisse für die Nachbarn verschlechtern (Abbildung 55).

Um aus Hunderten möglichen Bebauungsvarianten schnell die besten Ergebnisse ausfindig machen zu können, wird das im Vorgänger-Projekt ILCO entwickelte System zur Entwurfsraumexploration verwendet, der *Design Space Explorer* (DSE) [20].

Die Varianten, die nach den Analysen mit verschiedenen Kennwerten verbunden werden, können beim DSE hochgeladen und die Performance jeder Variante direkt visuell mit anderen verglichen werden (Abbildung 56).

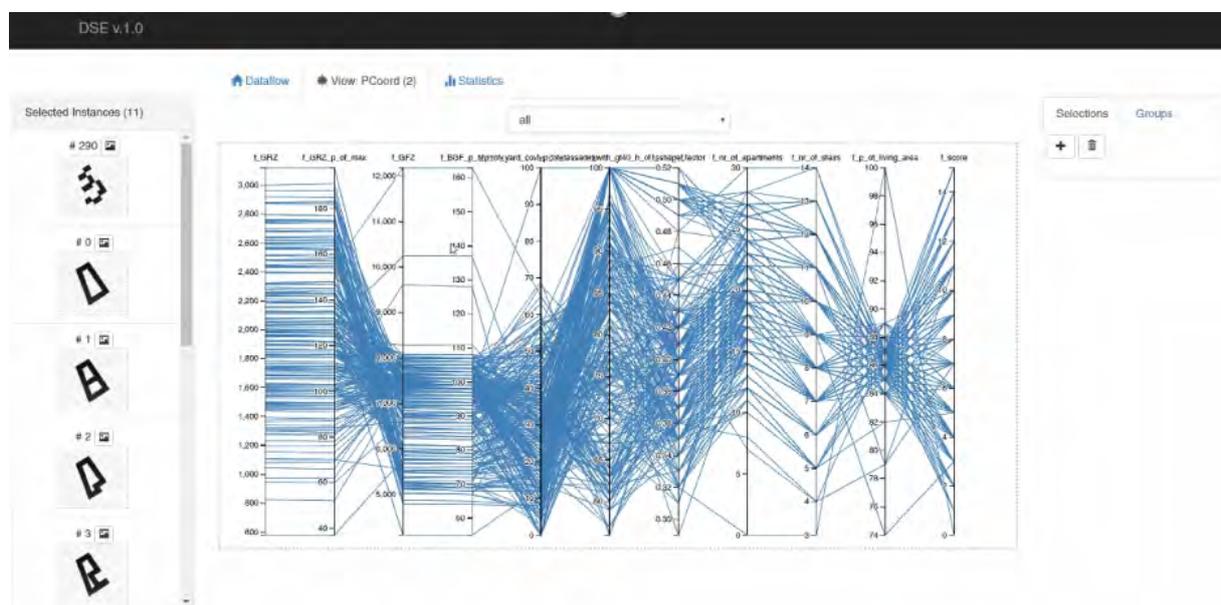


Abbildung 56
Anzeige aller erzeugten Varianten mit Performance-Vergleich mittels Parallelem Koordinatensystem.

Dabei können Varianten schnell interaktiv gefiltert werden, indem der Nutzer bei den gewünschten Kriterien die Bandbreite eingrenzt (Abbildung 57).

Weiterhin können mittels Self-Organizing-Maps (SOM) automatisch topologisch ähnliche Bebauungsvarianten gruppiert werden, was einen guten Überblick über Gruppen ähnlicher Entwurfsvarianten ermöglicht (Abbildung 58).

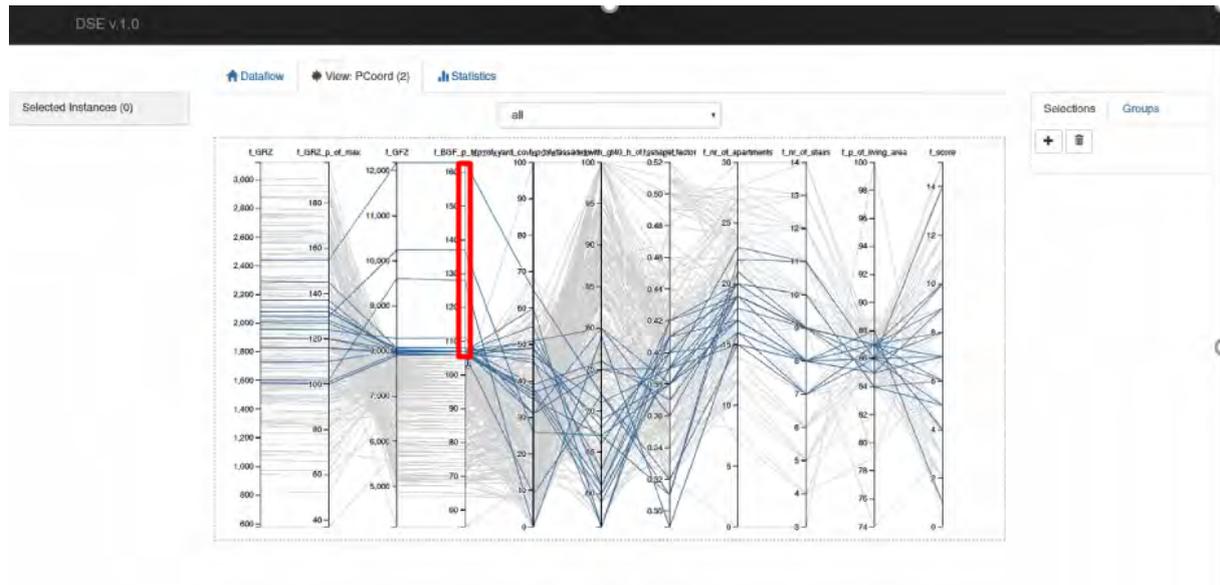


Abbildung 57
Filterung der Varianten anhand Eingrenzung der Performancekriterien.

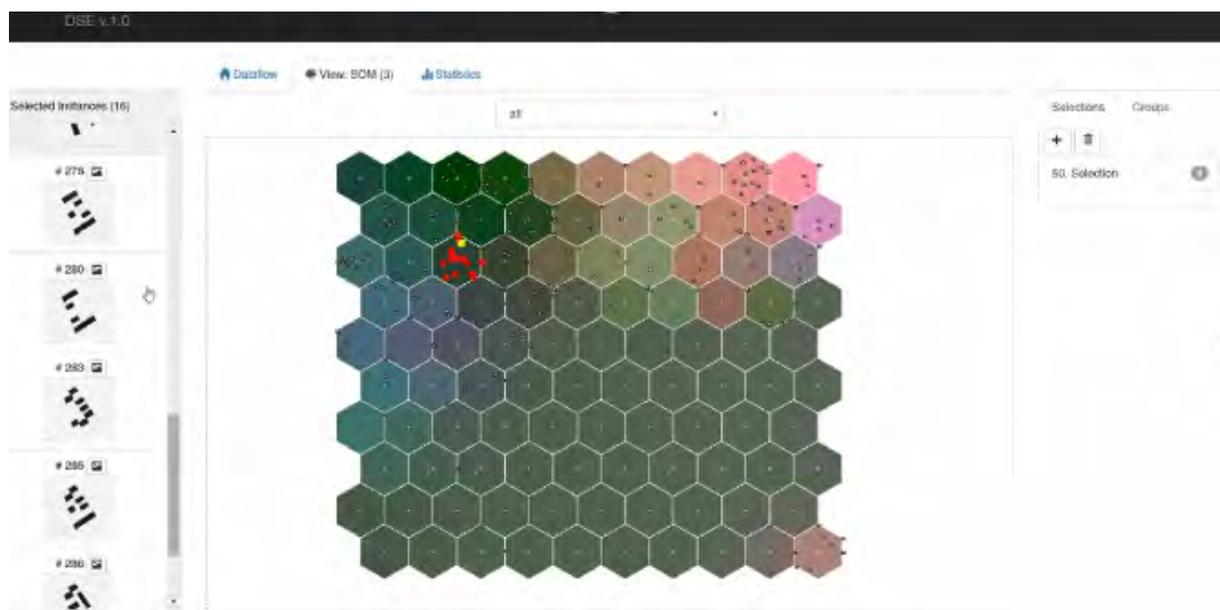


Abbildung 58
Gruppierung topologisch ähnlicher Varianten mittels SOM

Die oben genannten quantitativen Analysen bilden nur eine beschränkte Grundlage für die Evaluierung der Bebauungsvarianten, da sie die qualitativen Aspekte eines Projektes nicht abbilden können. Da aber die qualitativen Kriterien bei der Planung eine sehr wichtige Rolle spielen, wurden in diesem Projekt verschiedene Interaktionsmöglichkeiten bei der Maßstabebene der Generierung entwickelt (siehe AP3. Benutzerschnittstelle). Durch diese wird mittels geometrischer Festlegungen dem Planer ermöglicht, die qualitativen Aspekte einer Entwurfsidee in die Generierung einzupflegen und damit sicherzustellen, dass die wichtigen qualitativen Faktoren in den Bebauungsvarianten berücksichtigt werden. Hierbei ist darauf zu achten, dass die zu berücksichtigenden Faktoren von der städtebaulichen Ebene auf die detaillierteren Gebäude-Planungsebenen übertragen werden.

3.3 AP3 Benutzerschnittstelle

Mit der Umsetzung des Prototypen innerhalb der etablierten Softwareumgebung Rhino3d/Grashopper wurden die entwickelten Komponenten in der grafischen Oberfläche von Grashopper integriert. Damit sind diese für den geschulten Anwender einfach in eigene Prozesse innerhalb dieser Umgebung integrierbar. Die Steuerung der generativen Logik kann dabei vollständig über die Benutzeroberfläche von Grashopper abgewickelt werden und liefert direkt in Rhino3d eine Visualisierung der generierten Geometrie.

Als Interaktionsmöglichkeiten zur Kontrolle des generativen Prozesses, stehen dem Anwender verschiedene Wege offen. Auf der Maßstabsebene der Gebäudekubaturen kann der Anwender die Gebäudeform zuerst durch die Auswahl der typologischen Module und der Aktionssequenz (Abbildung 59) beeinflussen. Weiterhin können für jedes Modul verschiedene Inputparameter verändert werden. Zudem können einige Module durch geometrischen Input beeinflusst werden. So reagiert die Blocktypologie beispielsweise auf eingezeichnete Baulinien. Ferner kann die Position von Gebäudedurchbrüchen über das Setzen von Markern in der Geometrie bestimmt werden. In verschiedenen Feedbackrunden wurde besonders der geometrische Input als Möglichkeit für eine intuitive Lösungssuche hervorgehoben. Allerdings ist für die Bereitstellung geometriebasierter Interaktionsformen ein deutlich höherer Entwicklungsaufwand notwendig. Gerade die qualitativen raumbezogenen Anforderungen konnten mittels geometriebasierter Interaktion gut integriert werden.

Sequenz 1



Sequenz 2



Sequenz 3



Abbildung 59
Beispiele generierter Gebäudekubaturen, die aus drei verschiedenen Aktionssequenzen resultieren.

Auf der Maßstabsebene der Erschließungen wurden die Inputparameter auf boolesche Werte begrenzt (Abbildung 60). Ziel dieser Einschränkung war es, die Anzahl der Inputparameter für diese Maßstabsebene deutlich zu reduzieren und die Steuerung nur auf die wesentlichen Eingaben zu beschränken. In den Workshops wurden bei dieser Herangehensweise die Übersichtlichkeit und Einfachheit positiv bewertet. Allerdings wurde auch der Wunsch nach Möglichkeiten zur präziseren Kontrolle für die Generierung der Varianten geäußert.

Die Inputmöglichkeiten auf der Maßstabsebene der Wohnungsaufteilung wurden wegen des angewandten Machine Learning Verfahren auf den gewünschten Wohnungsmix begrenzt (Abbildung 61). Die generierten Varianten sind daneben abhängig von der zugrundeliegenden Grundrissbibliothek. Für die frühe Planungsphase wurde der Wohnungsmix als Steuerelement besonders wegen seiner Einfachheit und guten Integration an den planerischen Prozess ausgewählt.

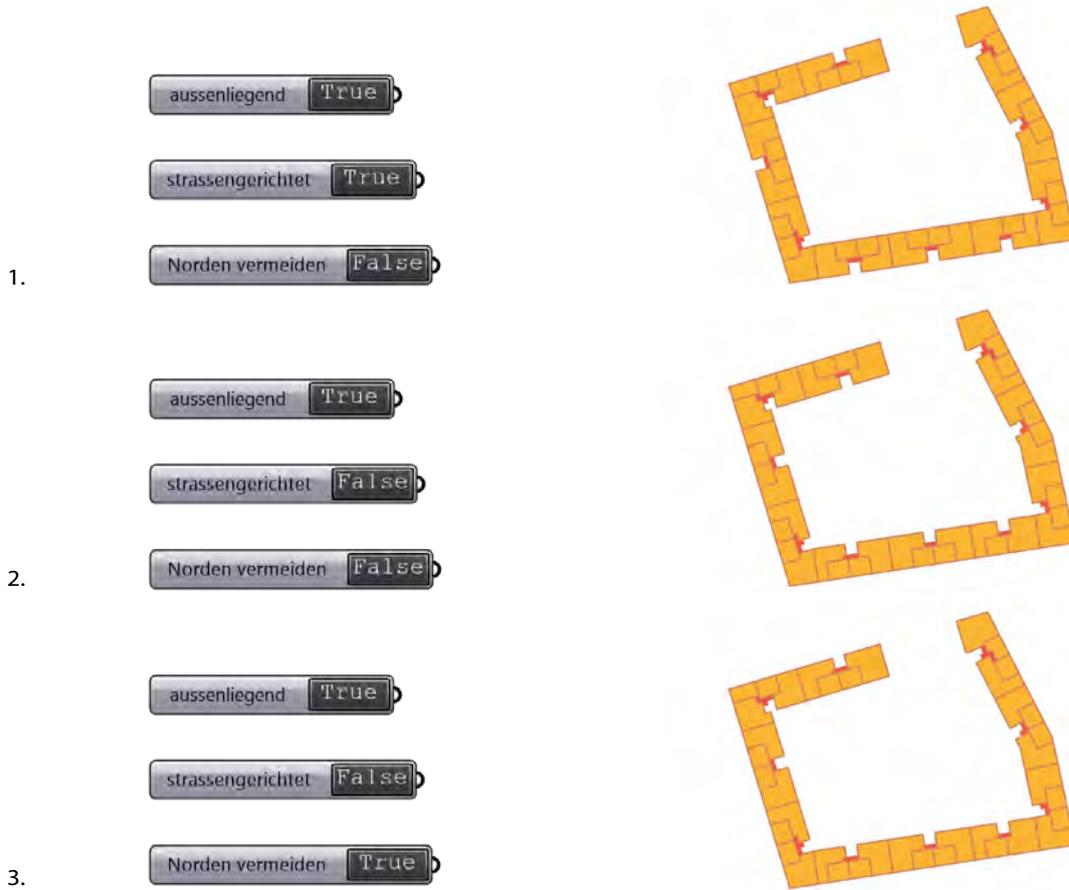


Abbildung 60
Gegenüberstellung der booleschen Inputparameter des Erschließungsmoduls (links) und der damit erzeugten Unterteilung (rechts).

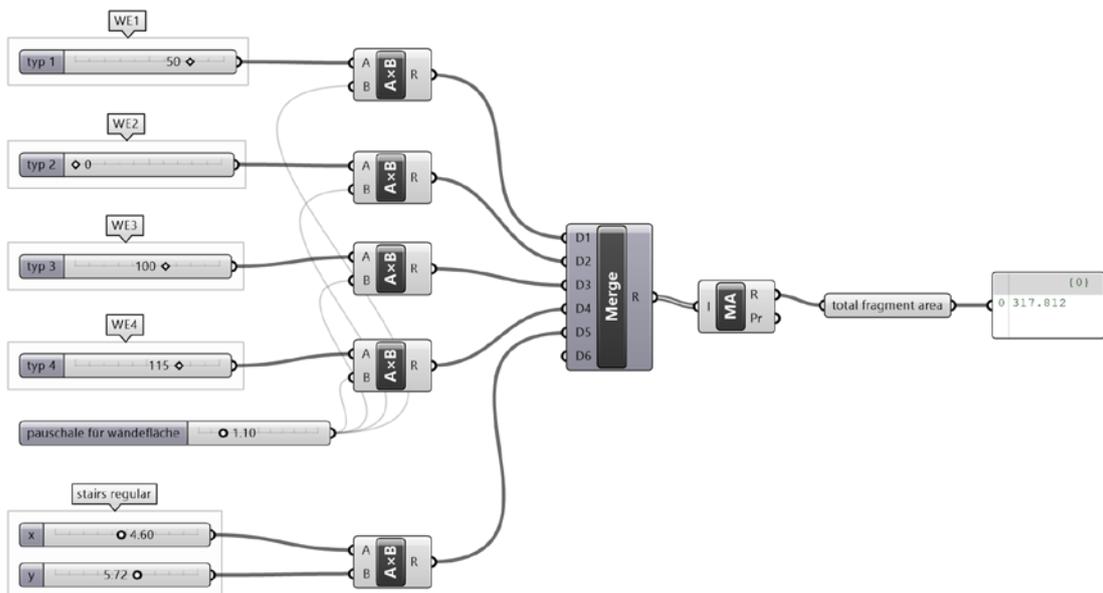


Abbildung 61
Eingabemodalität für den Wohnungsmix

Um die Performance der generierten Lösungen besser überwachen zu können, wurden ausgewählte Analyseergebnisse in die grafische Oberfläche von Rhino3d/Grasshopper eingebunden (Abbildung 62). Eine genauere Überprüfung der einzelnen Varianten kann in einem nächsten Schritt mit dem Design Space Exploration Framework vorgenommen werden. Dafür wurde eine Schnittstelle geschaffen, mit welcher, erzeugte Lösungen einschließlich ihrer Analysewerte in eine json Datei exportiert und an das Design Space Exploration Framework gesandt werden können.

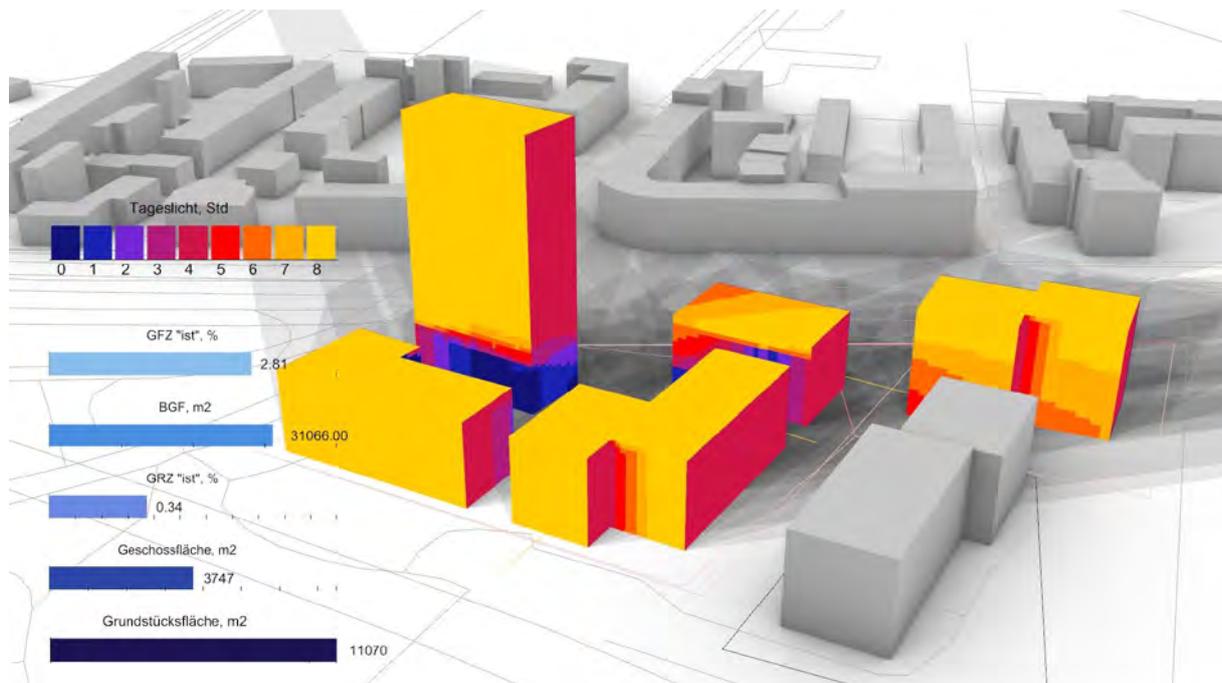


Abbildung 62
Darstellung von Analyseergebnissen innerhalb von Rhino3d/Grasshopper

Neben den oben dargestellten Input- und Outputmodalitäten wurde in diesem Projekt getestet, ob die entwickelten Algorithmen auch im Browser ohne eine lokale Installation der Rhinoceros/Grasshopper Software eingesetzt werden können. Dafür wurden die Komponenten in eine Shape Diver Anwendung [21] integriert. Für diesen Test wurden verschiedene Teile der Erzeugungslogik vereinfacht [22]. Das Modell kann dabei mittels numerischer sowie geometrischer Parameter im Browser gesteuert werden. Weiterhin war es möglich, die Auswahl des Baugebietes, die Auswahl der Aktionssequenz, sowie die teilweise Steuerung einzelner Aktionen abzubilden. Zum Beispiel konnten die Marker zur Festlegung der Gebäudedurchbrüche direkt im Browser bewegt werden (Abbildung 63). Die Laufzeitumgebung von Shape Diver gestattet allerdings nur die Ausführungen einer generativen Sequenz. Durch die direkte Ausführung und das Anzeigen der generierten Lösung kann der Nutzer in Echtzeit die Konsequenzen von Parameteränderung im Modell sehen und spielerisch erkunden.

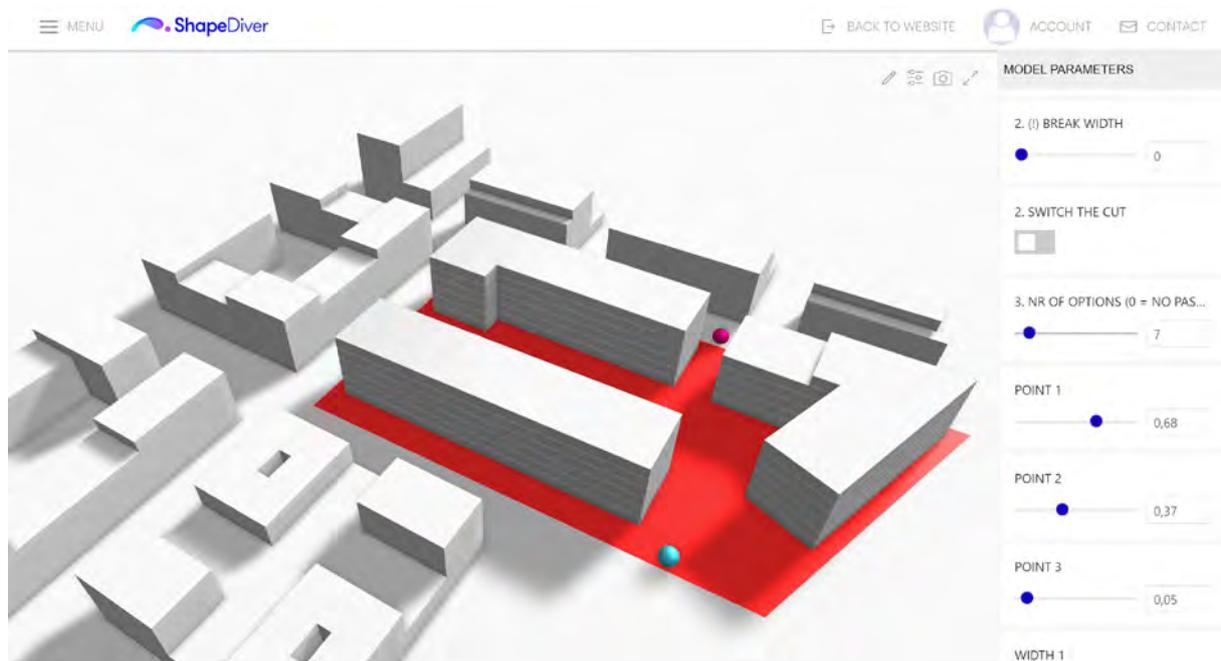


Abbildung 63
Steuerung und Visualisierung der Generierungsaktionen mittels geometrischer Parameter auf der Webplattform ShapeDiver.

3.4 AP4 Praxisbezug

Im Laufe des Projektes wurden Workshops und Feedbackrunden mit Architekten und Planern durchgeführt. Mehrere Workshops fanden beim Praxispartner DIPLAN statt, um gemeinsam die Herangehensweise beim Entwerfen in der frühen Planungsphase zu bestimmen. Dabei wurden verschiedene Generierungsansätze gemeinsam bewertet und die algorithmische Umsetzung besprochen. Darüber hinaus konnte der Austausch zwischen den Generierungsmethoden in Rhino/Grasshopper und der Machine Learning Anwendung der Firma DIPLAN definiert werden. Weiterhin wurden in Besprechungen die für die Praxis wichtigen Kennwerte festgelegt, die begleitend für die Generierung erstellt wurden, um eine Entscheidungsgrundlage für die Auswertung der Varianten zu erschaffen. Zudem hat DIPLAN einen Workshop mit dem Kölner Architekturbüro „ASTOC“, unter anderem beteiligt an Masterplan von Hafen City Hamburg, organisiert. Bei diesem Workshop wurde das „DigiWo“ Projekt inhaltlich vorgestellt und praktisch an einem aktuellen Projekt des Büros ASTOC bis zur Grundrissenerzeugung getestet (Abbildung 64).

Insgesamt ergab sich ein durchaus positives Feedback zur Praxistauglichkeit, die meisten Planer würden solche oder ähnliche Generierungs- und Analysetools in der frühen Planungsphase sehr nützlich finden, da diese Planungsphase sehr kurz ist und tatsächlich nur eine kleine Anzahl an Varianten anhand der Erfahrungswerten ausprobiert werden kann. Mit der automatisierten Generierungsmethode könnte diese Zahl deutlich vergrößert und somit auch Varianten mit potenziell besserer Performance gefunden werden.

Als Limitierungen der entwickelten Methode wurden folgende Punkte genannt:

- Fehlender Zugang zu Katasterdaten der Gemeinden.
- Fehlende Anbindung zur Topographie – zurzeit werden alle Grundstücke als flache Umriss behandelt.

- Mangel an Möglichkeiten, innerhalb der Generierung die Zwischenergebnisse manuell zu ändern (z. B. Gebäude rotieren, frei bewegen, extrudieren, abschneiden).
- Mangelnde Möglichkeit, eigene Grundrisse als Datenbasis für die Generierung auszuwählen, damit ausschließlich diese benutzt werden.
- Fehlendes Brandschutzkonzept bei der Erzeugung der Kubaturen.



Abbildung 64
Fotografische Dokumentation der Workshops (Fotograf: Jannis Bruns)

Um einen besseren Überblick zu der von ASTOC Architekten angemerkten Brandschutzproblematik zu bekommen, wurde ein Experteninterview mit der Planungsgruppe *Geburtig* zu diesem Thema durchgeführt. Um zu prüfen, wie die Unterteilung der Wohneinheiten gemäß den Brandschutzvorschriften funktionieren kann, wurden bei diesem Treffen unterschiedliche Brandschutzvorgaben durchgesprochen. Allerdings ließen sich die zahlreichen Auflagen und Sonderfälle nicht mehr im zeitlichen Projektrahmen unterbringen.

Das Projekt hat durch das Engagement der Firma DIPLAN einen engen Praxisbezug herstellen können. Der umgesetzte Prototyp und die im Projekt gesammelten Erfahrungen und Erkenntnisse wurde von der Firma DIPLAN für die Verbesserung und Weiterentwicklung ihrer eigenen Machine Learning basierten Lösung genutzt (Abbildung 64). Die Methoden zur Generierung von Kubaturen und der Ansatz zur Erschließungsplanung wurde unter anderem genutzt, um größere Datensätze zu erstellen, die nach Prüfung und Anpassung durch Architekten und Planer in die Datengrundlage ihrer eigenen Softwarelösung integriert werden können.

Der Praxispartner DeCodingSpaces [23], welcher eine Reihe an Methoden zur Generierung und Analyse von Entwürfen auf städtebaulicher Ebene – von Straßennetzgenerierung bis zur Gebäudeplatzierung auf einzelnen Parzellen – entwickelt hat, stellt seine Online-Plattform zur Verfügung, auf welcher die im Projekt entwickelten Komponenten öffentlich zugänglich gemacht werden. Auf der DeCodingSpaces Plattform werden außer den Komponenten, auch Tutorials zur Anwendung der entwickelten Methoden veröffentlicht.

Das entwickelte Framework wurde zudem im Rahmen eines Praxistests evaluiert, wofür das ehemalige Wettbewerbsgebiet „MK6 Theresienhöhe, München“ [24] verwendet wurde. Dafür wurde die Block-Typologie verwendet. Ziel der Evaluation war es, Entwürfe zu generieren die ähnlich, wie die von Architekten manuell entwickelten Varianten performen. Die generierten Entwürfe wurden mit den besten Lösungen des Wettbewerbs anhand der Bebauungsdichte (GFZ) verglichen. Dieses Vorgehen ermöglichte es ferner, die von uns entwickelte Generierungsmethode hinsichtlich Benutzerinteraktion zu testen, indem formale Anforderungen bei der Erstellung der Varianten nach Möglichkeit befolgt werden mussten. Damit wurde erfolgreich gezeigt, wie die

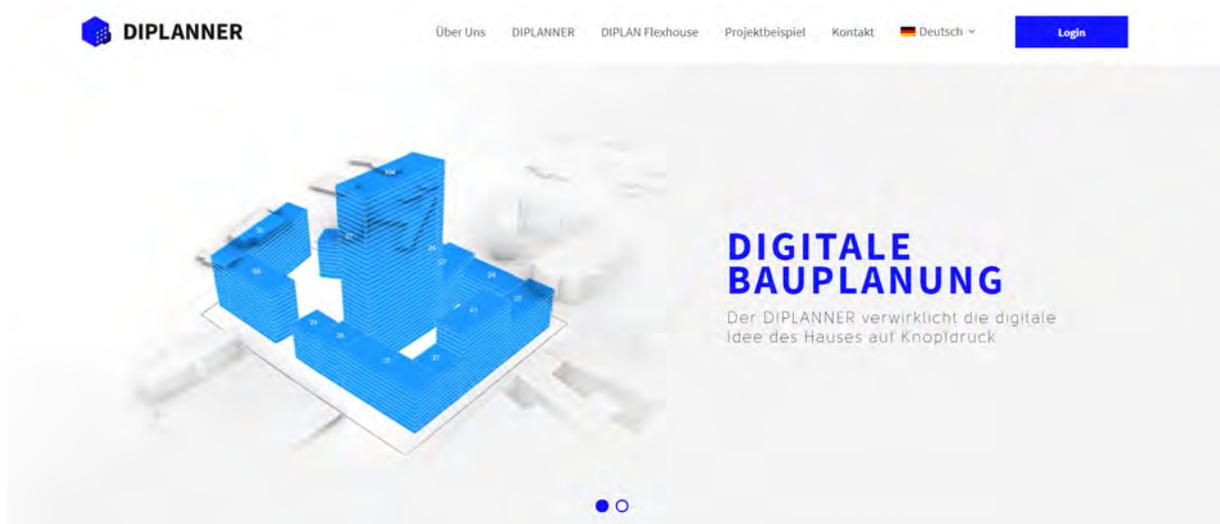


Abbildung 65
Umsetzung der entwickelten Generierungsmethoden in Form einer Online-Plattform zur Erstellung der Machbarkeitsstudien „DIPLANNER“ [16].

im Rahmen dieses Projekts entwickelten Methoden und Softwaremodule in einem architektonisch-städtebaulichen Arbeitsprozess während der frühen Entwurfsphase eingesetzt werden können.

Eine der wesentlichen Herausforderungen für Planer ist die in der Regel offene Anzahl von zu berücksichtigenden Zielen und Anforderungen eines Projekts, infolgedessen muss der Entwurf immer wieder überarbeitet und angepasst werden. Ausreichende Interaktionsmethoden mit den generativen Methoden sind daher für Änderungen im Zuge der weiteren Projektentwicklung dringend erforderlich. Die verschiedenen Analysen, mittels Entwurfsvarianten erlauben es, unmittelbar die Konsequenzen einer Entwurfsänderung zu bewerten, wodurch die Transparenz bei der Entscheidungsfindung gefördert wird.

Die erste von uns generierte Bebauungsvariante ist ein geschlossener Block mit mehreren schmalen Durchgängen entlang einer Fassade und einem höheren Abschnitt an der südlichen Ecke des Blocks. Es wurde die folgende Generierungssequenz verwendet (Abbildung 66):

Basic Block → Random Breaks → Towers

Hier war der Blockumriss sowie ähnliche Höhenverteilung leicht zu erreichen. Die Zuordnung der Durchgänge ist in einer Sequenz schwer umzusetzen. An dieser Stelle wäre es möglich, die gewünschten Durchgänge mit der Angabe der selbst-gezeichneten Achsen zu erstellen, dabei würde sich aber die Bandbreite der automatisch erzeugten Varianten verkleinern.

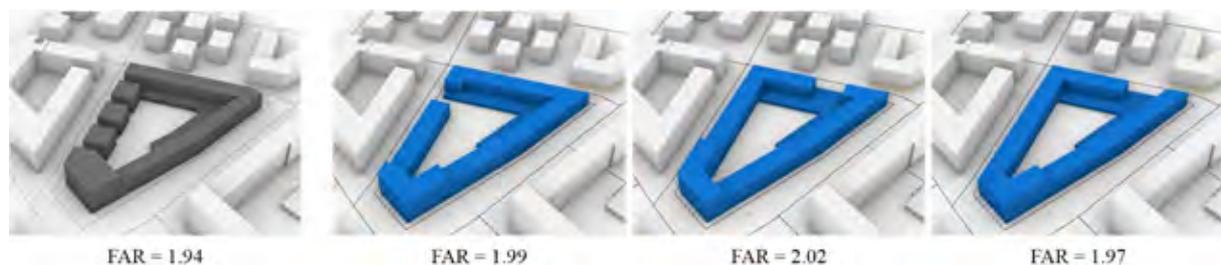


Abbildung 66
Beispielhafte Gegenüberstellung einer Lösung aus dem Wettbewerb (links) mit den automatisch generierten Varianten (rechts)

Die zweite Variante, der Block beinhaltet mehrere breite Durchgänge. Es wurde dieselbe Sequenz der Aktionen wie in der ersten Variante angewandt, dabei wurden die Parameter geändert, um andere Bebauungsvarianten zu erstellen (Abbildung 67). Bei dieser Variante wurde deutlich, dass bei unserer Generierungsmethode eine Möglichkeit fehlt, eine Anforderung für verschiedene Gebäudeabschnitte mit verschiedenen Gebäudetiefen zu definieren.



Abbildung 67
Beispielhafte Gegenüberstellung einer Lösung aus dem Wettbewerb (links) mit den automatisch generierten Varianten (rechts).

Die dritte Variante fügt der Randblockbebauung Hofgebäude hinzu und schneidet die südliche Ecke ab. Die Abfolge der benutzten Aktionen:

Basic Block → Random Breaks → Inner Yards

Hier konnten die Hofbauten (Abbildung 68) mit den selbst gezeichneten Achsen sehr schnell und präzise erstellt werden. Es fehlt allerdings die Möglichkeit, punktuell in die Generierung einzugreifen und die Zwischenergebnisse manuell zu verändern, zum Beispiel um eine Gebäudeecke abzurunden.



Abbildung 68
Beispielhafte Gegenüberstellung einer Lösung aus dem Wettbewerb (links) mit den automatisch generierten Varianten (rechts).

4. Konklusion und Ausblick

Im Rahmen des Projekts wurde ein Prototyp in Rhino3d/Grashopper für die maßstabsübergreifende Generierung von Entwurfsvarianten für den Wohnungsbau entwickelt. Dafür wurde ein Top-Down Ansatz über drei Maßstabsebenen (Gebäudekubaturerschließung und Grundriss) ausgearbeitet. Mit dieser Herangehensweise konnte der Generierungsprozess für Wohnungsbauten in Teilprobleme gegliedert werden. Die gesonderte Betrachtung der Teilprobleme ermöglicht es auf Details in den einzelnen Ebenen besser einzugehen. So konnten praktische Anforderungen, die in enger Zusammenarbeit mit den Architekten und Planern erarbeitet wurden, besser in die Generierungsmethode integriert und somit ein starker Praxisbezug hergestellt werden. Dank dieser praxisnahen Umsetzung wird der Prototyp bereits von der Firma DIPLAN genutzt und weiterentwickelt. Ein weiterer Praxispartner, die Firma *DeCodingSpaces* wird zudem verschiedene Komponenten aus diesem Projekt in seine Rhino3d/Grashopper-Toolbox übernehmen und als *DeCodingSpaces Lab Post* veröffentlichen.

Aufgrund der Komplexität einzelner Teilprobleme und den zahlreichen Anforderungen aus der Praxis an das generative System hat sich die Untersuchung auf die wichtigsten Aspekte für jede Maßstabsebene konzentriert. Um eine hohe Praxisrelevanz zu gewährleisten, wurden zur Einhaltung des zeitlichen Projektrahmens gezielte Abgrenzungen vorgenommen. So entwickelte die Firma DIPLAN das generative Verfahren für die Grundrissebene, die in den maßstabsübergreifenden Prozess integriert wurde. Das Forschungsteam an der Bauhaus-Universität Weimar konzentrierte sich auf die Entwicklung von generativen Lösungen für die Gebäudekubatur- und Erschließungsebene. Ebenfalls wurden die relevante Analysemodule entwickelt (z. B. Tageslichtanalyse). Mit dieser Fokussierung konnte der gesamte maßstabsübergreifende Prozess umgesetzt werden. Trotzdem sehen wir noch Potential in der weiteren Untersuchung der einzelnen Maßstabsebenen und der Verbesserung der Methoden. Im vorliegenden Projekt wurden für die Bearbeitung der Teilprobleme verschiedene Heuristiken aus der Entwurfspraxis entlehnt. Es zeigte sich im Projektverlauf, dass sich mit diesen auf Erfahrungswerten basierenden Herangehensweise in praktikablen Laufzeiten gute Ergebnisse erzielen lassen.

Eine weitere Aufgabenstellung des Projekts bestand darin, verschiedene benutzerfreundliche Möglichkeiten zur Prozesssteuerung zu implementieren. Diese wurden anhand der für das Interface der visuellen Programmierumgebung Grashopper entwickelten Komponenten umgesetzt. Dabei kann der Nutzer sowohl über numerische Parameter als auch über geometrische Inputs in Rhino3d direkten Einfluss auf den Generierungsprozess nehmen. Weiterhin kann der Nutzer die erzeugten Lösungen dank des modularen, sequentiellen Aufbaus des Verfahrens über selbst gewählte Kombinationen der einzelnen Komponenten beeinflussen. Ein geometrisches Gebäudemodell stellt die gemeinsame Datengrundlage dar, die nicht nur den Austausch zwischen den einzelnen Modulen innerhalb einer Maßstabsebene ermöglicht, sondern auch für den Datenaustausch über Maßstabsgrenzen hinweg funktioniert. Neben der parametrischen Steuerung, erwies sich die Interaktion über geometrische Elemente, für den Anwender als intuitive Möglichkeit den generativen Prozess zu kontrollieren. In weiteren Forschungsarbeiten sollten insbesondere die Ansätze zur Nutzerinteraktion erweitert und die in diesem Projekt gemachten Beobachtungen durch Nutzerstudien überprüft werden. Auf Basis des direkten Austausches mit der Planungspraxis zeichnet sich zudem die Notwendigkeit von Multiuserinteraktionsmöglichkeiten als interessantes Forschungsgebiet ab, welches beispielsweise die Abwicklung kooperativer Aus handlungsprozesse in den Blick nimmt.

Zusammenfassend wird zum Abschluss dieses Projekts ein generativer Prototyp präsentiert, welcher für maßstabsübergreifende Generierungsprozesse erfolgreich demonstriert wurde. Mit dem Prototyp können in kurzer Zeit zahlreiche praxistaugliche Planungsvarianten für verschiedene Grundstücksformen erzeugt werden können.

Mitwirkende

Autorinnen und Autoren

König, Reinhard, Jun.-Prof. Dr. (Bauhaus-Universität Weimar)
Schneider, Sven, Vertr.-Prof. Dr.-Ing. (Bauhaus-Universität Weimar)
Osintseva, Iuliia, M. Sc.
Berst, Andreas, M. Sc.

Studentische Hilfskräfte

Gavrilov, Egor, B. Sc.
Oravec, Martin, M. Sc.

Projektpartner

DIPLAN Gesellschaft für Digitales Planen & Bauen GmbH
DecodingSpaces GbR

Fachliche Betreuung

Referat WB 3 „Forschung im Bauwesen“
Dr.-Ing. Arnd Rose

Literaturverzeichnis

- [1] Streit, Matthias. "Der Wohnraummangel ist gravierend" [Webseite]: <https://www.handelsblatt.com/finanzen/immobilien/immobilien-der-wohnraummangel-ist-gravierend-/19358896.html?ticket=ST-1414098-bhdbORYUrTTqyaOAJclk-ap4>, letzter Aufruf am 23.08.2019
- [2] Henger, R., Voigtländer, M., "IW-Report 28/2019 Ist der Wohnungsbau auf den richtigen Weg?" (2019).
- [3] Rodrigues, Eugénio. "Automated floor plan design: generation, simulation, and optimization." (2014).
- [4] Das, Subhajit et al. "Space plan generator: Rapid generation & evaluation of floor plan design options to inform decision making." (2016).
- [5] Chaillou, Stanislas. "AI+ Architecture: Towards a New Approach." Harvard University (2019).
- [6] "Verordnung über die Honorare für Architekten- und Ingenieurleistungen (Honorarordnung für Architekten und Ingenieure – HOAI)." Bundesrat, Drucksache 395/09 (30. April 2009).
- [7] Wilson, L., Danforth, J., Davila, C. C., and Harvey, D. "How to Generate a Thousand Master Plans: A Framework for Computational Urban Design." SimAUD (2019), Seiten 113-119.
- [8] Schneider, Sven; König, Reinhard. "Interaktion bei der Computergestützten Generierung von Layouts." Arbeitspapiere Bauhaus-Universität Weimar (2011).
- [9] DIPLAN Gesellschaft für Digitales Bauen und Planen GmbH [Webseite]: <https://www.diplanbau.com/>, letzter Aufruf am 18.11.2020
- [10] a+t research group. "Why density? Debunking the myth of the cubic watermelon." a+t architecture publishers (2015).
- [11] Buerkin, T., Peterek, M., Petereck, M. "Stadtbausteine." Birkhäuser (2016).
- [12] Reicher, C. "Städtebauliches Entwerfen." Springer, Vieweg (2017), Seiten 54 - 81.
- [13] Openstreetmap [Website]: <https://www.openstreetmap.de/>, letzter Aufruf am 24.09.2019.
- [14] Interface zum Zeichnen des Grundstücks anhand Kartenvorlage sowie zum Runterladen von Kontextgeometrien [Webseite] <https://deltacodes.eu/projects/diplan/index.html>, letzter Aufruf am 18.09.2019.
- [15] Baugesetzbuch, §34 "Zulässigkeit von Vorhaben innerhalb der im Zusammenhang bebauten Ortsteile" [Webseite] https://www.gesetze-im-internet.de/bbaug/___34.html, letzter Aufruf am 18.11.2020.
- [16] Deutsche Bauordnungen [Webseite]: <http://www.bauordnungen.de/html/deutschland.html>, letzter Aufruf am 28.09.2019.
- [17] Knowles, R.L., Berry, R. D., Department of Energy & Solar Energy Research Institute, "Solar envelope concepts: moderate density building applications: final report." "Solar Energy Information Data Bank, Golden, Colo (1980).

-
- [18] Moudsari, R. S., Pak, M., Adrian Smith + Gordon Gill Architecture "Ladybug: a parametric environmental plugin for grasshopper to help designers create an environmentally-conscious design." Proceedings of International Building Performance Simulation Assosiation, Chambery, France (2013), Seiten 3128-3135.
- [19] Greedy algorithm. Encyclopedia of Mathematics. [Webseite]: http://encyclopediaofmath.org/index.php?title=Greedy_algorithm&oldid=34629, letzter Aufruf am 25.09.2020.
- [20] Zaghoul, Mohamed. "Machine-Learning aided Architectural Design-Synthesize Fast CFD by Machine-Learning." (2017).
- [21] Fuchkina, E., Schneider, S., Bertel, S., Osintseva, I. "Design Space Exploration Framework - A modular approach to flexibly explore large sets of design variants of parametric models within a single environment." Proceedings eCAADe Conference - Volume 2 (2018) Seiten 367-376.
- [22] Shape Diver Anwendung zur interaktiven Nutzung der parametischen Modellen im Browser [Webseite] <https://www.shapediver.com>, letzter Aufruf am 18.11.2020
- [23] Ein parametrisches Modell als Beispiel zur Nutzerinteraktion [Webseite] <https://app.shapediver.com/m/digiwo-block-2-0-3>, letzter Aufruf am 18.11.2020
- [24] Abdumawla, A., Bielik, M., Buš, P., Mei-Chih, C., Denmark, M., Fuchkina, E., Miao, Y., Knecht, K., König, R., and Schneider, S. "Decoding Spaces" 2019.
- [25] Ergebnisse eines Wohnungsbau Wettbewerbes in München "MK6 Theresienhöhe" [Webseite] <https://www.competitionline.com/de/ergebnisse/241367>, letzter Aufruf am 24.09.2019

Anhang

Anhang 1: Struktur und Funktionen der prototypischen Entwicklungsumgebung in python

In dieser Testumgebung konnten einzelne generative Lösungsansätze für die Gebäudeplatzierung im zwei-dimensionalen Raum programmiert und getestet werden. Dafür wurden einfache geometrische Repräsentationen wie beispielsweise Formen, Linien und Punkte sowie planungstechnische Elemente wie zum Beispiel Grundstücksgrenzen (Site) oder Gebäude (Segment) in entsprechenden Python Klassen umgesetzt (Abbildung 69). Eine generierte Anordnung von mehreren Segmenten innerhalb einer begrenzten Fläche wurde in eine Lösungsklasse (Solution) gekapselt. Diese Klasse diente als Ausgangsbasis, um einzelne generierte Lösungen zu analysieren und zu vergleichen. Die Repräsentation wurde dabei möglichst abstrakt gehalten um später für andere Maßstabsebenen wiederverwendet werden zu können.

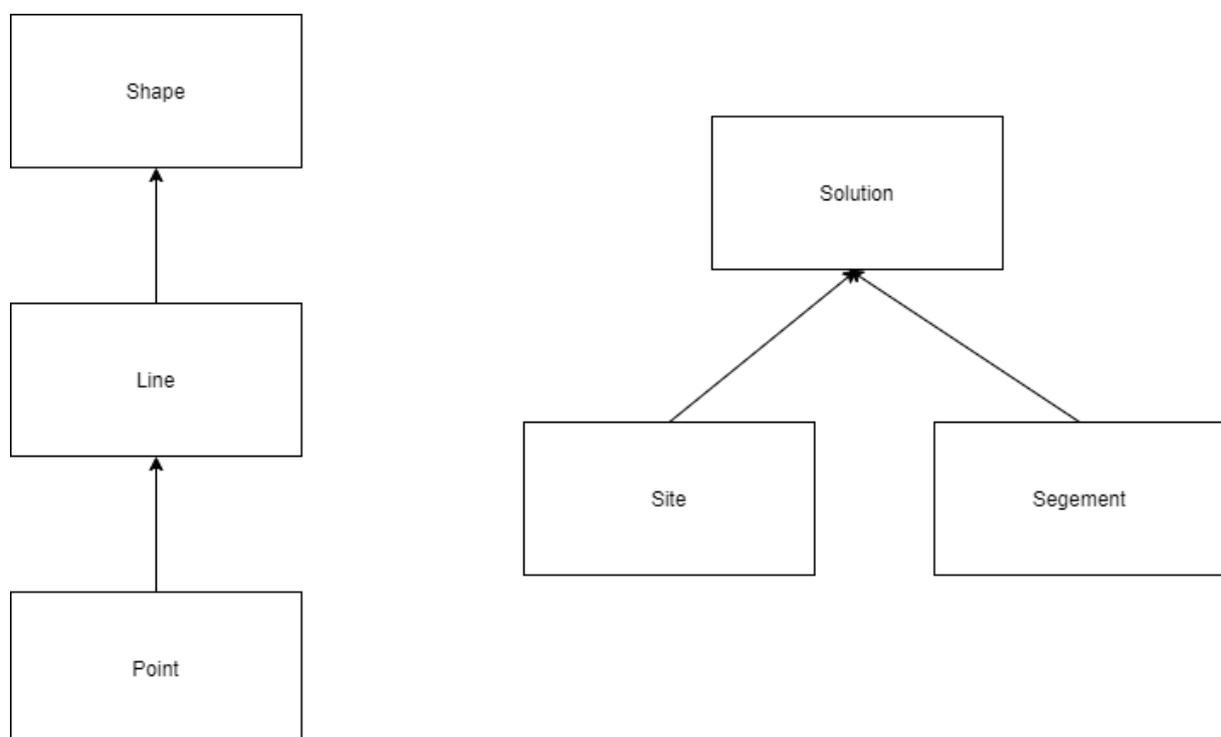


Abbildung 69
Zusammensetzung und Abhängigkeiten der umgesetzten Repräsentationsstrukturen. Jedes der abgebildeten Objekte wurde im Rahmen einer Klassenstruktur in Python umgesetzt.

Spezielle semantische Klassen, wie das Gebäudesegment wurden um zusätzliche Funktionen erweitert, um wichtige städteplanerische Aspekte, wie Abstandsflächen abzubilden. Gebäudesegmente selbst sind durch ihre Position (Punktkoordinaten), Länge (Meter), Breite (Meter), Orientierung (Winkelmaß) und Höhe (Meter) definiert. Mit diesen Inputparametern können schnell geometrische rechtwinklige Formen erzeugt und platziert werden. Dabei werden die entsprechende Abstandsflächen auf Basis eines definierbaren Faktors und der angegebenen Gebäudehöhe angelegt.

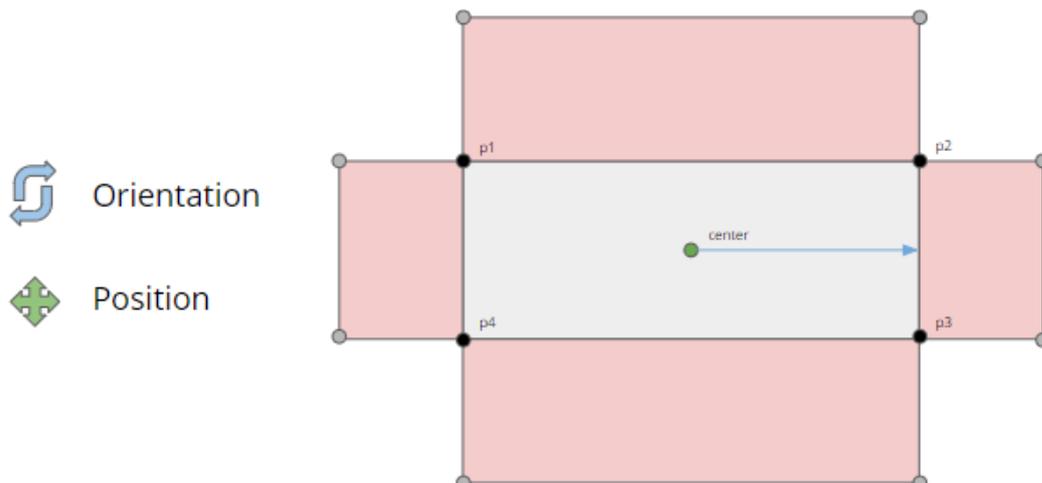


Abbildung 70
Visuelle Darstellung der Zusammensetzung eines Gebäudesegments einschließlich Abstandsflächen und der zugrunde liegenden Punkt- und Linienrepräsentation.

Eine weitere Funktion, die in dieser Umgebung angelegt und getestet wurde, war die Einhaltung von Grundstücksgrenzen und Abstandsflächen. Dementsprechend wurde eine Kollisionsabfrage implementiert, welche einzelne Gebäudeanordnungen auf mögliche Kollisionen überprüft. Um diesen Schritt möglichst effizient durchzuführen, wurden zuerst die Überschneidungen der einzelnen Radiale überprüft und im zweiten Schritt die entsprechenden Außenlinien auf Überschneidungen getestet.

Anhang 2: Building Divider: Methode zur Platzierung rechteckiger Wohnungen in konkaven Gebäudekubaturen gemäß dem Wohnungsschlüssel

Da man den gewünschten Wohnungsmix fast nie exakt in der Gebäudekubatur erreichen kann, werden die drei nächstmöglichen Proportionen ausgesucht. Zum Beispiel kann eine Proportion von 60 %–30 %–10 % für einen Block von 100 Meter Länge niemals durch die verfügbaren Wohnungstypen mit den Breiten von 10, 20 und 40 m erreicht werden. Der nächstmögliche Wohnungsmix wäre für diesen Block dementsprechend

INPUT →

Gewünschter Mix (%): 10 - 25 - 65

Wohnungsbreiten: a - b - c

Blöcke: B1 - B2 - B3

PROZESS →

Angepasster Mix 1 (%): 12.5 - 24 - 63.5

Wohnungssatz: a,a,b,b,c,c,c,c,c,c

Angepasster Mix 2 (%): 15 - 28 - 57

Wohnungssatz: a,a,a,b,b,c,c,c,c,c

Angepasster Mix 3 (%): 8 - 26 - 66

Wohnungssatz: a,b,b,b,c,c,c,c,c

Abbildung 71
Anpassung des ursprünglichen Wohnungsmixes.

50–10–40 %. Daher erfolgt zuerst eine Berechnung, welche die Anzahl der Wohnungstypen bestimmt, die im gesamten Block untergebracht werden können. Jede Wohnungsmix-Variante erzeugt eine eigenständige geometrische Lösung für die Wohnungsverteilung (Abbildung 71). Durch dieses Vorgehen kann gewährleistet werden, dass bei der Wohnungsaufteilung keine oder nur geringe Restflächen übrigbleiben.

Im nächsten Schritt werden die Wohnungstypen nach der berechneten Verteilungsproportion den jeweiligen Gebäudeabschnitten zugewiesen. Dabei müssen die Geschosszahlen der einzelnen Gebäude berücksichtigt werden: wenn eine Wohnung eines bestimmten Typs im Erdgeschoss eines Gebäudes platziert wird, wird sie aus konstruktiven Gründen in den oberen Geschossen darüber platziert. Daraus folgt, dass der Wohnungsmix von den Gebäudehöhen stark beeinflusst wird und diese daher bei der Berechnung der Proportionen berücksichtigt werden muss.

Der Algorithmus für die Unterteilung eines Gebäudes in Wohnungen funktioniert nach der Brut-Force-Methode: am Anfang wird eine Liste von erforderlichen Wohnungen aller Typologien erstellt. Bei jeder Iteration wird versucht, alle möglichen Wohnungen eines Typs an der nächsten freien Stelle zu platzieren. Dabei können manchmal nicht alle Gebäude „befüllt“ werden. Zum Beispiel kann ein Gebäude mit einer Länge von 100 Metern nicht ohne Restflächen mit 11 Meter breiten Wohnungen bestückt werden.

Nachdem alle Wohnungstypen verteilt wurden, können die Restflächen für jedes Gebäude berechnet werden. Da die Berechnung für alle möglichen Wohnungskombinationen relativ lange dauern kann, wird eine zulässige maximale Restfläche festgesetzt. Sobald die gesamte Restfläche überschreitet, wird der Prozess abgebrochen (Abbildung 72).

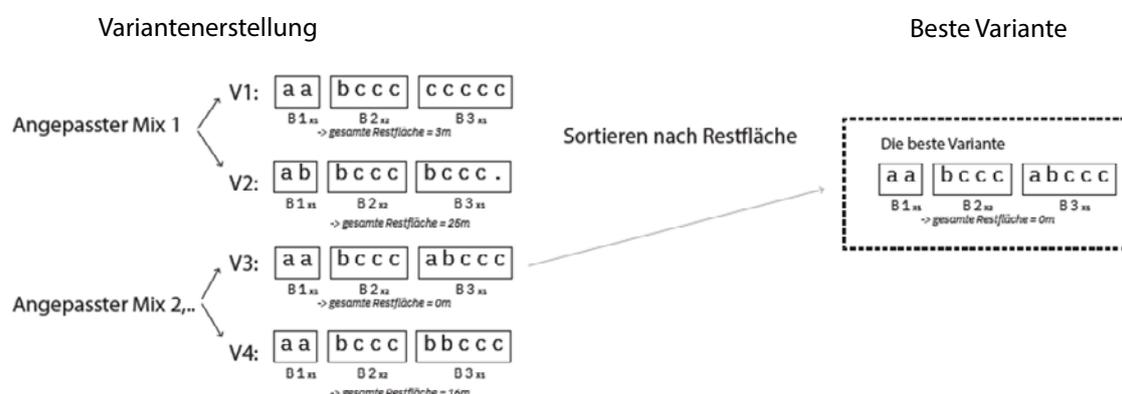


Abbildung 72

Anwendung möglicher Abfolgen der Wohnungskombinationen, um den gewünschten Wohnungsmix zu erreichen.

Der beschriebene Prozess für die Wohnungsverteilung wird auf unterschiedliche Weise vereinfacht. Zum Beispiel werden Aufteilungen [A-A-B-C] und [A-A-C-B] gleichwertig behandelt, denn die Reihenfolge der Wohnungen kann in diesem Schritt vernachlässigt werden. Infolgedessen werden die Varianten in geordneter Folge der Größe nach zusammengesetzt. Letztendlich werden alle berechneten Varianten miteinander verglichen und die Variante mit der geringsten Restfläche wird ausgewählt und für die Platzierung der Wohnungstypen verwendet (Abbildung 73, 74).

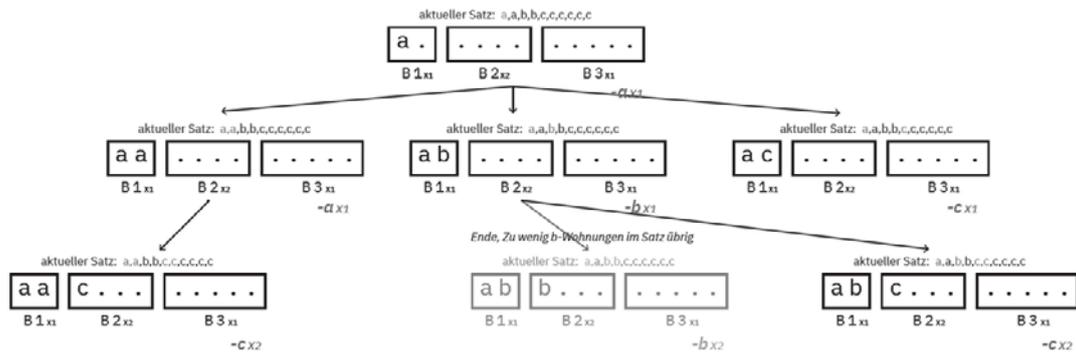


Abbildung 73
Brute-Force Ansatz zur Suche nach Wohnungskombinationen.

```

[0;0]
VARIANT 0 (total rest = 88.0010429776858m, total flat length = 758.8m):
BLOCK 0
-5.42-5.42-5.42-8.13-
BLOCK 1
-5.42-5.42-5.42-8.13-
BLOCK 2
-5.42-5.42-
BLOCK 3
-
BLOCK 4
-8.13-10.84-10.84-
BLOCK 5
-8.13-10.84-
VARIANT 1 (total rest = 88.0010429776858m, total flat length = 758.8m):
BLOCK 0
-5.42-5.42-5.42-8.13-
BLOCK 1
-5.42-5.42-5.42-8.13-
BLOCK 2
-5.42-5.42-
BLOCK 3
-
BLOCK 4
-8.13-10.84-10.84-
BLOCK 5
-8.13-10.84-
    
```

Abbildung 74
Der Output vom Unterteilungsmodul – die Reihenfolge der Wohnungstypen, die mit Wohnungsweiten gekennzeichnet werden.

Anhang 3:

Osintseva, I.; König, R.; Berst, A.; Bielik, M.; Schneider, S., 2020: Automated Parametric Building Volume Generation: A Case Study for Urban Blocks. SimAud Proceedings.

Anhang 4:

Osintseva, I.; Berst, A.; Bielik, M.; König, R.; Schneider, S., 2019: Typology-based generation of multi-family-residential buildings. Proceedings of International Conference on Optimization-Driven Architectural Design