

Exposé zur Bachelorarbeit von Adrian Helberg bei Prof. Dr. Jenke

20. Oktober 2020

1 Problemstellung

Effizientes Objektdesign und -modellierung sind entscheidende Kernkompetenzen in verschiedenen Bereichen der digitalen Welt. Da die Erstellung geometrischer Objekte für Laien unintuitiv ist und ein großes Maß an Erfahrung und Expertise erfordert, ist dieses stetig wachsende Feld für Neueinsteiger nur sehr schwer zu erschließen. Die Forschung liefert hierzu einige Arbeiten zur **prozeduralen Modellierung**, um digitale Inhalte schneller und automatisiert zu erstellen. Gerade wenn es um die Darstellung natürlicher Umgebung geht, ist die Erstellung von ähnlichen Objekten, wie zum Beispiel verschiedene Bäume derselben Gattung eines Waldes, ein schwieriges Problem. Kleine Änderungen in prozeduralen Systemen können zu großen Veränderungen der Ergebnisse führen. Darum beschäftigt sich die **inverse prozedurale Modellierung** unter anderem mit dem Inferieren von Regeln und Mustern aus gegebenen Objekten, um diese nach bestimmten Regeln zu modellieren. Ein wichtiges Werkzeug hierbei ist die Verwendung einer **formalen Grammatik**, eine fundamentale Datenstruktur der Informatik, um Strukturen zu beschreiben. Eine spezielle Untergruppe sind die **L-Systeme**, die häufig bei der Beschreibung von Verzweigungsstrukturen und Selbstähnlichkeit zum Einsatz kommen.

Die Bachelorarbeit soll sich mit der Erstellung eines prozeduralen Systems zur Synthese von Ähnlichkeitsabbildern beschäftigen. Hierzu soll über eine Benutzerschnittstelle eine Struktur erzeugt werden, aus der ein parametrisiertes L-System inferiert werden kann, das dann zur Generierung von ähnlichen Strukturen genutzt werden kann.

1.1 Konzepte und Ideen

Im Folgenden wird auf die Methodik zum Ableiten eines L-Systems eingegangen und Konzepte und eigene Lösungsansätze präsentiert.

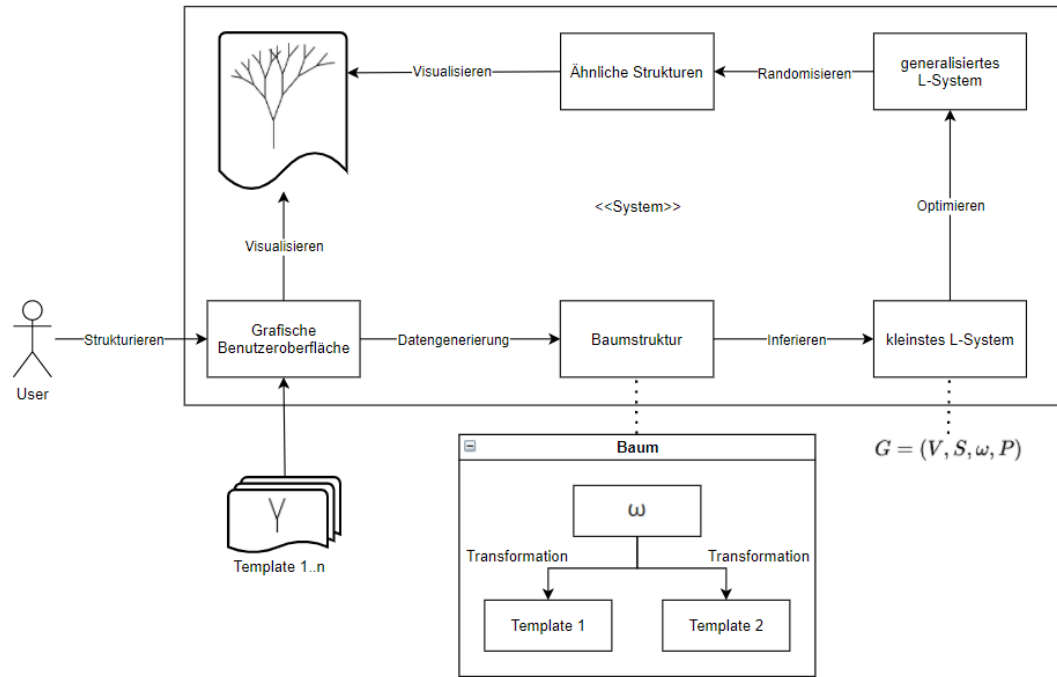


Abbildung 1: Architektur des Systems mit einigen Datenstrukturen

1.1.1 System

Ziel des Systems ist die Erstellung von Ähnlichkeitsabbildern einer Struktur, die über eine grafische Benutzeroberfläche erstellt wird.

1.1.2 Überblick

Abbildung 1 zeigt, wie das o.g. System aufgebaut ist:

- I. Strukturieren: Der Benutzer nutzt die grafische Benutzerschnittstelle, um aus einzelnen, atomaren Strukturen (**Templates**) eine zusammenhängende Struktur zu erstellen. Neben der Position der Templates können auch Parameter wie Rotation angepasst werden.
- II. Visualisieren: Das Ergebnis der Strukturierung ist jederzeit sichtbar. Einfache Liniensegmente können mit einem **Turtle-Algorithmus** gezeichnet werden.
- III. Datengenerierung: Das Ergebnis der Strukturierung wird in einer Baumstruktur organisiert, in der jeder Knoten einem bestimmten Template entspricht und die zugehörigen Kanten die geometrischen Transformationen relativ zum Elternknoten beschreibt.
- IV. Inferieren: Aus der Baumstruktur kann eine formale Grammatik in Form eines kleinen L-Systems abgeleitet werden, indem identische topologische Strukturen (Unterbäume) gefunden werden können.
- V. Optimieren: Hier werden ähnliche Produktionsregeln des L-Systems mithilfe einer **Kostenfunktion**¹ zusammengefasst (*engl. Merge*), um diese mit nicht-deterministischen Regeln und Rekursion zu generalisieren.
- VI. Randomisieren: Jedes Symbol der Grammatik nimmt eine Liste an Parametern entgegen, die in diesem Schritt nach bestimmten Kriterien pseudo-randomisiert werden, um Variationen zu erstellen und anschließend zu visualisieren.

¹Die Kostenfunktion setzt sich sowohl aus der Länge der Grammatik, als auch der Distanz von der alten zur neuen, generalisierten Grammatik nach dem Merge zusammen

1.2 Einführung

Die Einführung in das Themas könnte wie folgt aussehen:

Mit der Digitalisierung der Welt erlangt die Erstellung von digitalen Inhalten, wie 3D Modelle für Computerspiele, Webdesigns oder Visualisierung von Architektur, zunehmend an Bedeutung. Darum werden Verfahren gesucht, um Objekte dieser Felder formal zu beschreiben und somit kodifizierbar (*engl. codify*) zu machen. Hierbei bilden sich folgende Klassen heraus:

- **Modellierung:** Um einen physikalischen Körper in ein digitales Objekt zu überführen, wird mithilfe von Abstraktion (oder Modellierung) ein mathematisches Modell erstellt, das diesen Körper formal beschreibt. 3D Grafiksoftware, wie Blender [7], wird genutzt um geometrische Körper zu modellieren, texturieren und zu animieren.
- **Prozedurale Modellierung:** „*It encompasses a wide variety of generative techniques that can (semi-)automatically produce a specific type of content based on a set of input parameters.*“ [6]
„Prozedurale Modellierung beschreibt generative Techniken, die (semi-)automatisch spezifische, digitale Inhalte anhand von deskriptiven Parametern erzeugen“ (*Übersetzt durch den Autor*)
- **Inverse prozedurale Modellierung (IPM):** *Aliaga et al.* [1] spricht bei der inversen prozeduralen Modellierung von dem Finden einer prozeduralen Repräsentation von Strukturen bestehender Modelle.

Die Modellierung mithilfe von Grafiksoftware ist eine vergleichbar händische, langwierige Erstellung von Objekten. Hierbei hat der Designer (= Modellierer) die volle Kontrolle über die Strukturen des Objektes.

Bei der prozeduralen Modellierung werden spezifische Strukturen eines zu erstellenden physikalischen Objektes generalisiert und meist über eine Grammatik und globale Parameter abgebildet. Während bei der klassischen Modellierung die menschliche Intuition und bei der prozeduralen Modellierung eine parametrisierte Grammatik vorausgesetzt wird, arbeitet IPM mit bestehenden Modellen und extrahiert („lernt“) die Strukturen des Objektes, die automatisch in eine formale Grammatik überführt werden können. Die Generierung von prozeduralen Modellen ist ein wichtiges, offenes Problem [1].

Aktuelle Ansätze sind:

- Segmentierung von geometrischen Objekten in Ähnlichkeitsgruppen, um Muster (*engl. Patterns*) zu erkennen
- Kontrollierte Generierung durch Finden optimaler Parameter und Regeln

1.3 Grundlegende Arbeiten

Dieses Kapitel beschreibt die relevanten Themen der Bachelorarbeit und stellt einige wissenschaftliche Quellen vor

- ✎ L-Systeme [3] sind Ersetzungssysteme, die zur formalen Beschreibung von Zellteilung eingeführt wurden, und atomare Teile mithilfe von Produktionsregeln ersetzen. Die bekanntesten L-Systeme sind Zeichenkettenbasiert und wurden von *Noam Chomsky* eingeführt. Sie ersetzen parallel Buchstaben eines Wortes, die von einer Grammatik über eine Sprache akzeptiert werden. L-Systeme können parametrisiert oder nicht-parametrisiert und kontextfrei oder kontextsensitiv sein.
 - ⇒ Der Logo-Turtle-Algorithmus [5] beschreibt ein Vorgehen zur graphischen Beschreibung von L-Systemen, bei dem jeder Buchstabe in einem Wort einer bestimmten Zeichenoperation zugewiesen wird.
- ✎ Prozedurale Modellierung [6]: Siehe Kapitel 1.2 Einführung
 - ⇒ Inverse prozedurale Modellierung [1] gilt als Ansatz für „Reverse Engineering“ von Eingabedaten zum Lernen von Regeln und Merkmalen, die zur prozeduralen Modellierung verwendet werden können.

1.4 Verwandte Arbeiten

Relevanten Arbeiten zu den o.g. Themen werden hier aufgeführt

- ✎ Unter Inferieren von L-Systemen [4] versteht man eine Vielzahl an Algorithmen, die Daten untersuchen, um automatisiert L-Systeme zu generieren, die diese Daten beschreiben.

2 Organisatorisches

- Prototypische Entwicklung eines Systems zur Erstellung „ähnlicher“ 2D-Strukturen anhand eines Input-Bildes, das mit einer grafischer Oberfläche erstellt wird
- Zweiwöchige Meilensteine mit Besprechungen (Videokonferenz)
- Phasen:
 - Vorbereitung
 - Literaturstudium
 - Problemstudium
 - Praktische Arbeit
 - Schriftliche Arbeit
- Das Online-Repository ist unter [2] zu finden

3 Zeitplan

- ✓ bis 29.08.2020: Erarbeitung Basisquelle
- ✓ 14.09.2020: Kennenlerngespräch, erstes Themengespräch
- ✓ bis 28.09.2020: Version 1.0 des Exposés
- ✓ bis 13.10.2020: Einigung auf eine neue Fragestellung
- ✓ bis 18.10.2020: Version 2.0 des Exposés, Zweitgutachter auswählen
- bis 25.10.2020: Problemstudium, Anmeldung der Bachelorarbeit
- bis 27.10.2020: Ausarbeitung des Software-/Hardwarestacks
- bis 01.11.2020: Erstellung des Projektplans mit Arbeitspaketen
- ab 02.11.2020: Bearbeitung des praktischen Teils der Bachelorarbeit, paralleler Beginn des schriftlichen Teils

Abbildungsverzeichnis

1	Systemarchitektur	2
---	-----------------------------	---

Literatur

- [1] Daniel G. Aliaga, Ilke Demir, Bedrich Benes, and Michael Wand. Inverse procedural modeling of 3d models for virtual worlds. In *ACM SIGGRAPH 2016 Courses*, SIGGRAPH '16, New York, NY, USA, 2016. Association for Computing Machinery.
- [2] Github. <https://github.com/adrian-helberg/bachelor>. GitHub, Inc. (2020).
- [3] A Lindenmayer. Mathematical models for cellular interactions in development. i. filaments with one-sided inputs. *Journal of theoretical biology*, 18(3):280—299, March 1968.
- [4] Ian McQuillan, Jason Bernard, and Przemyslaw Prusinkiewicz. *Algorithms for Inferring Context-Sensitive L-Systems*, pages 117–130. 01 2018.
- [5] P Prusinkiewicz. Graphical applications of l-systems. In *Proceedings on Graphics Interface '86/Vision Interface '86*, page 247–253, CAN, 1986. Canadian Information Processing Society.
- [6] Ruben M. Smelik, Tim Tutenel, Rafael Bidarra, and Bedrich Benes. A survey on procedural modelling for virtual worlds. *Comput. Graph. Forum*, 33(6):31–50, September 2014.
- [7] Pablo Vazquez and Francesco Siddi. About blender. <https://www.blender.org/about/>, 2017. Blender Foundation (2002).