

N. Generative Modellierung – Verknüpfung von Wissen und Form

→ Architektur, Generative Modellierung,
Geometrische Datenverarbeitung,
Produktdesign, Prozedurale Modellierung,
Semantische Datenverarbeitung

Die Idee der generativen Modellierung besteht darin, die Generierung von hochkomplexen Objekten auf der Grundlage einer Reihe von formalen Konstruktionsregeln zu unterstützen. Anhand dieser Konstruktionsregeln wird eine Form nicht nur durch das geometrische Endergebnis aller angewandten Operationen, sondern durch eine Abfolge von Bearbeitungsschritten beschrieben: Formgebung wird zum Regeldesign. Dank seines generischen Charakters kann dieser Ansatz auf jede Domäne und jede beliebige Formdarstellung angewendet werden, die entsprechende erzeugende Funktionen bietet. Generative Modellierungstechniken bieten viele interessante Anwendungsvorteile, insbesondere etwa im Kontext der Architektur: Als Folge von Konstruktionsschritten beschreibt ein generatives Modell nicht nur ein Gebäude, sondern umfasst in gewisser Art auch das Wissen, wie es zu bauen ist. Diese Metainformationen werden unter anderem in Bibliotheksdiensten benötigt, um 3D-Daten sinnvoll zu indizieren und zu archivieren. Dieses Wissen kann genutzt werden, um z. B. Laserscans automatisch nach Architekturstilen und -perioden zu klassifizieren. Eine maschinenlesbare Darstellung, wie etwas gebaut wird, kann verwendet werden, um fehlende Teile mit plausiblen Schätzungen zu ergänzen. Weitere vielversprechende Anwendungsgebiete sind die Klassifizierung und Restaurierung von Kulturgütern oder die Unterstützung semantischer Techniken im Maschinenbau. Dieses Kapitel beschreibt das Paradigma der generativen Modellierung. Dabei steht die Kombination aus semantischen Informationen mit der Kodierung von Konstruktionsschritten im Mittelpunkt. Die Anwendungsmöglichkeiten werden an den Beispielen der Modellklassifizierung, Identifikation und der Formrekonstruktion illustriert.

N.1 Motivation

Die Beschreibung von 3D-Formen ist heute eine wichtige Voraussetzung für viele Anwendungen, z. B. im Bereich industrielle Fertigung (Formbeschreibung von Bauteilen), Stadtplanung und Architektur (Gebäude und Pläne der technischen Gebäudeausrüstung) oder Simulation (Formbeschreibung von Oberflächen). In der Computergrafik und der computergestützten Geometrie wurden bis heute viele Formrepräsentationen vorgestellt, um Formeigenschaften zu kodieren und zu verarbeiten, beispielsweise für die Modellierung, die Visualisierung und die Suche. Viele Formrepräsentationen sind **explizit**, d. h. sie kodieren direkt die Oberfläche oder das Volumen einer Form. Eine sehr einfache und weit verbreitete Variante direkter Kodierung ist z. B. die explizite Kodierung von Koordinaten und der Konnektivität der Eckpunkte eines Polygonnetzes. Im Gegensatz dazu beruhen generative Repräsentationen und Modellierungswerkzeuge auf der **impliziten** Kodierung von Formen. Diese beinhalten in der Regel algorithmische Vorschriften, durch welche eine Form anhand einer Menge von Basisformen und Operationen erzeugt wird. Generative Modellierung ist ein mächtiges Werkzeug für viele Anwendungsfälle wie etwa für die effiziente Generierung von Formvariationen oder musterbasiertes Suchen von Formen. Wir führen in diesem Kapitel in die Grundlagen des generativen Modellierens ein und geben eine Reihe von Anwendungsbeispielen, welche den Nutzen der Techniken veranschaulichen. Darüber hinaus beschreiben wir einige interessante Probleme für zukünftige Arbeiten.

N.2 Geometrische Sprachen

Ein **generatives Modell** ist eine algorithmische Beschreibung, die während der Ausführung erzeugende und modifizierende Funktionen aufruft. Das Endergebnis ist eine Instanz einer Familie von Modellen, die durch den Algorithmus und der Menge aller möglichen Aufrufparameter repräsentiert wird. **01** Die Umsetzungen dieses theoretischen Konzepts können wie folgt gruppiert werden: **Regelbasiert**: Ersetzungsregeln und deren sukzessive Anwendung auf eine initiale, einfache Datenstruktur erzeugen komplexe, geometrische Strukturen. **02 Skriptbasiert**: Der Einsatz eines Interpreters oder eines Übersetzers erlaubt die Verwendung beliebiger Programmiersprachen zur Erzeugung geometrischer Gebilde. **03 Datenflussbasiert**: Generative Modellersysteme mit grafischer Oberfläche arbeiten oft mit Datenflussgraphen, die Modellierprozesse auf formale Beschreibungen abbilden. **04**

Alle Ansätze haben ihren Ursprung im Übersetzerbau – jener Disziplin der Informatik, die sich mit dem Entwurf und der Programmierung von Compilern beschäftigt. Compiler sind Computerprogramme, die den Quelltext einer höheren Programmiersprache in den Zielcode in Maschinensprache übersetzen. Dort werden algorithmische Beschreibungen auf der Basis der zugrunde liegenden Grammatik klassifiziert. Laut **Chomsky-Hierarchie** ist eine Beschreibung, d. h. eine Sprache, rekursiv aufzählbar, kontextsensitiv, kontextfrei, oder regulär. Um eine algorithmische Beschreibung ausführen zu können, sind – unabhängig von der Ausführung durch einen Interpreter oder durch einen Übersetzer – folgende Schritte notwendig: Erstens: die lexikalische Analyse. Sie zerteilt den eingelesenen Quelltext in lexikalische, elementare Einheiten, die Tokens genannt werden. Zweitens: die syntaktische Analyse, auch Parsing genannt. Durch sie wird der Syntaxbaum erstellt. Drittens: die semantische Analyse. Sie vervollständigt den Syntaxbaum mit Attributen, unter anderem durch die Auflösung von Variablennamen.

Im Kontext der generativen Modellierung ist eine weitere Klassifizierung relevant. Sie orientiert sich am Berechnungsmodell und der Berechenbarkeit. Die Verwendung einer Turing-mächtigen **05** Programmiersprache in einem Modellersystem führt zu Problemen in der Nutzererfahrung. Aufgrund des Halteproblems kann die Terminierung eines algorithmischen Modells nicht vorab bestimmt werden, Endlosschleifen können also nicht immer erkannt werden und müssen explizit beendet werden. Diese Situationen werden vermieden, wenn die algorithmische Beschreibungssprache nicht Turing-mächtig ist, sondern lediglich primitiv-rekursive Funktionen erlaubt.

■ 01

Ulrich Krispel, Christoph Schinko, Torsten Ullrich, *The Rules Behind – Tutorial on Generative Modeling*, in: *Proceedings of Symposium on Geometry Processing / Graduate School 12, 2014*, S. 2:1-2:49.

■ 02

Lars Krecklau, Darko Pavic, Leif Kobbelt, *Generalized Use of Non-Terminal Symbols for Procedural Modeling*, in: *Computer Graphics Forum 29, 2010*, S. 2291-2303; Pascal Müller, Peter Wonka, Simon Haegler, Andreas Ulmer, Luc Van Gool, *Procedural Modeling of Buildings*, in: *Proceedings of ACM Siggraph 25 (3) 2006*, S. 614-623.

■ 03

Christoph Schinko, Martin Strobl, Torsten Ullrich, Dieter W. Fellner, *Scripting Technology for Generative Modeling*, in: *International Journal on Advances in Software 4, 2011*, S. 308-326.

■ 04

Markus Lipp, Peter Wonka, Michael Wimmer, *Interactive Visual Editing of Grammars for Procedural Architecture*, in: *ACM Transactions on Graphics 27 (3) 2008*, S. 1-10 und Wolfgang Thaller, Ulrich Krispel, Sven Havemann, Dieter W. Fellner, *Implicit Nested Repetition in Dataflow for Procedural Modeling*, in: *Proceedings of the International Conference on Computational Logics, Algebras, Programming, Tools, and Benchmarking (Computation Tools) 3, 2012*, S. 45-50.

■ 05

Eine Turing-mächtige Programmiersprache ist nach der Berechenbarkeitstheorie von Alan Turing (britischer Mathematiker, 1912-1954) eine universelle Programmiersprache, in der jede Berechnung ausgedrückt werden kann, die sich in irgendeiner maschinenlesbaren Beschreibung ausdrücken lässt.

N.3 Geometrische Modelle

Als geometrische Modellierung bezeichnet man Methoden zur computer-gestützten Beschreibung der Formen geometrischer Objekte. Diese Methoden beinhalten unter anderem Computer-Aided Geometric Design (CAGD), algebraische Geometrie sowie algorithmische Geometrie. **06** CAGD beschäftigt sich mit der mathematischen Beschreibung von Kurven und Flächen hauptsächlich unter Verwendung parametrischer Gleichungen der Differentialgeometrie. Die Kombination von einfachen Formen zu komplexeren Modellen mittels boolescher Algebra wird Constructive Solid Geometry (CSG) genannt und ist ein prominenter Vertreter von Solid Modeling. Algebraische Geometrie ist die Erweiterung der klassischen analytischen Geometrie. Algorithmische Geometrie beschäftigt sich mit Design und Analyse von geometrischen Algorithmen mit Verbindungen zu numerischen Methoden, Berechenbarkeitstheorie und Komplexitätsanalyse. Die Wahl der Beschreibung ist oft eng mit der Aufgabe bzw. den technischen Gegebenheiten verbunden.

■ 06

Michael E. Mortenson, *Geometric Modeling*, in: John Wiley & Sons, 1985.

N.4 Geometrische und semantische Informationen

Ein prominentes Beispiel für den Einsatz generativer Techniken zur Modellierung sind die sogenannten **Formgrammatiken**. Sie wurden von Stiny und Gips **07** zur generativen Beschreibung von Bildern entwickelt. Dazu wurden Ansätze aus der formalen Sprachtheorie nach Noam Chomsky **08** verwendet. Dieser Ansatz wurde zur generativen Beschreibung von Architektur weiterentwickelt. **09** **10** Bei diesen Ansätzen beschreibt die Menge der Regeln die Konstruktionsvorschriften, welche die Erstellung eines Gebäudes beeinflussen. Moderne Ansätze beschreiben die Erstellung eines Gebäudemodells in mehreren Stufen: Wie in der Architektur üblich, zuerst in Form eines groben Massenmodells; schrittweise werden dann Details (durch Anwendung von Regeln) hinzugefügt.

■ 07

George Stiny, James Gips, *Shape Grammars and the Generative Specification of Painting and Sculpture*, in: *Best computer papers of 1971* 1, 1971, S.125-135.

■ 08

Noam Chomsky, *Three models for the description of language*, in: *IRE Transactions on Information Theory* 2, 1956, S. 113-124.

■ 09

Peter Wonka, Michael Wimmer, François Sillion, William Ribarsky, *Instant Architecture*, in: *International Conference on Computer Graphics and Interactive Techniques, ACM SIGGRAPH 2003* 22 (3) 2003, S. 669-677.

■ 10

Pascal Müller, Peter Wonka, Simon Haegler, Andreas Ulmer, Luc Van Gool, *Procedural Modeling of Buildings*, in: *Proceedings of ACM Siggraph* 25 (3) 2006, S. 614-623.

■ 11

Wolfgang Thaller, Ulrich Krispel, René Zmugg, Sven Havemann, Dieter W. Fellner, Shape grammars on convex polyhedra, in: *Computers & Graphics* 37 (6) 2013, S. 707–717.

■ 12

René Zmugg, Ulrich Krispel, Wolfgang Thaller, Sven Havemann, Martin Pszeida, Dieter W. Fellner, A New Approach for Interactive Procedural Modelling in Cultural Heritage, in: *Proceedings of the 40th Conference of Computer Applications and Quantitative Methods in Archaeology, 2012*, S. 190–204.

Der Aufbau entspricht also einer Grob-nach-fein-Struktur. Thaller et al. ¹¹ und Zmugg et al. ¹² nutzen volumenbasierte Verfeinerungsregeln mit konvexen Polyedern als Grundmodellierprimitiv. Ihr Regelsystem erlaubt einen gewissen Freiheitsgrad in der Form, auf die es angewendet wird, und passt sich dann automatisch an. Dadurch werden parametrische Modelle möglich, wie sie in ⁰¹ gezeigt sind.

□ 01

Das Modell links ist von der Rialto-Brücke in Venedig inspiriert, sein Hauptparameter ist sein Neigungswinkel. Das Modell rechts zeigt eine virtuelle Rekonstruktion des äußeren Teils vom Ostflügel-Pavillon des Louvre. Durch den Einsatz parametrisierter Elemente können Teile des Modells (durch Austausch von Regeln) nachträglich verändert werden.



■ 13

Patrik Schumacher, Parametricism: A New Global Style for Architecture and Urban Design, in: *Architectural Design – Digital Cities* 79 (4) 2009, S. 14–23.

■ 14

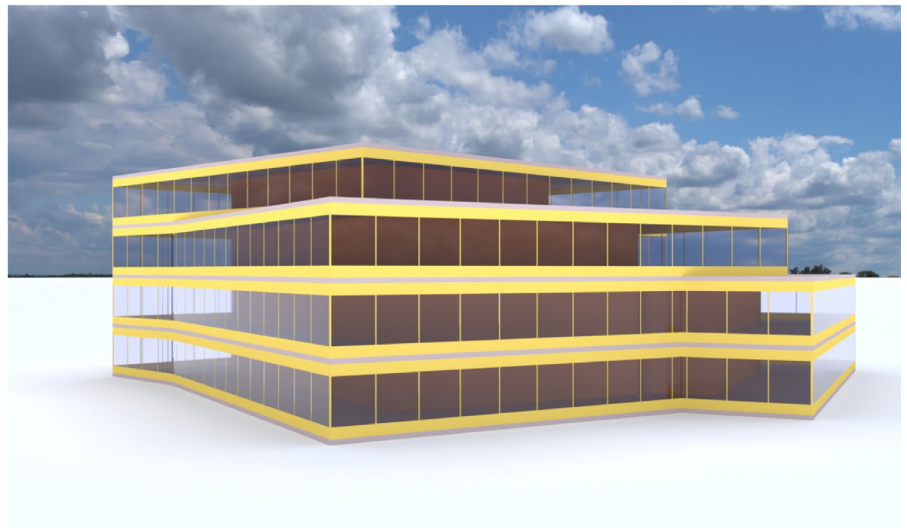
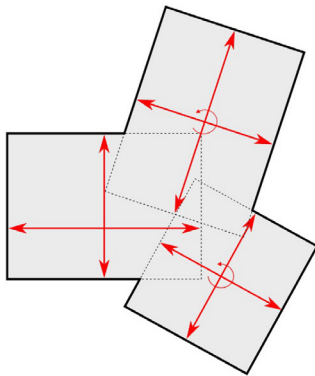
Wolfgang Thaller, Ulrich Krispel, Sven Havemann, Ivan Redi, Andrea Redi, Dieter W. Fellner, Developing Parametric Building Models – the GANDIS Use Case, in: *Proceedings of the 4th ISPRS International Workshop 3D-ARCH, 2011*, S. 163–170.

Basierend auf diesen hochentwickelten parametrischen Entwurfssystemen und Skripting-Techniken hat die Verwendung von Animations-, Simulations- und Formfindungstechniken eine neue Architekturrichtung mit radikal neuen Zielen und Werten inspiriert: den **Parametrismus**. ¹³

Im nächsten Beispiel werden semantische Informationen genutzt, welche durch die generative Beschreibung verfügbar sind: Thaller et al. ¹⁴ entwickelten eine Klasse von generativen Gebäudemodellen, welche neben einer Voransicht auch eine Abschätzung der Energieeffizienz ermöglichen. Der Klient kann so schon frühzeitig ein Gefühl dafür entwickeln, wie sich Designentscheidungen ökonomisch auswirken. Die generative Beschreibung besteht in diesem Beispiel aus mehreren Grundformen ⁰². Die Auswertung der Beschreibung zeichnet entsprechende geometrische Einzelteile gemäß ihrer Materialien aus, gruppiert die konkreten Flächenanteile nach Himmelsrichtungen und liefert damit eine Schätzung der Energieeffizienz.

□ 02

Ein generatives Gebäudemodell ermöglicht es, die Grundform eines Gebäudes mit einigen wenigen Parametern zu beschreiben (links). Die Auswertung der generativen Beschreibung liefert ein 3D-Modell (rechts) inklusive einer Abschätzung der Energieeffizienz.



■ 15

René Berndt, Christoph Schinko, Ulrich Krispel, Volker Settgest, Sven Havemann, Eva Eggeling, Dieter W. Fellner, Ring's Anatomy – Parametric Design of Wedding Rings, in: Proceedings International Conference on Creative Content Technologies 4, 2012, S. 72–78.

Ein weiteres Einsatzgebiet generativer Beschreibungen ist das **Produkt-design**. Berndt et al. ¹⁵ zeigen am Beispiel von Eheringen, dass sich mit einer generativen Beschreibung eine Vielzahl an möglichen Variationen realisieren lässt. Die Form eines Ringes wird dabei aus einem Profildpolygon, der Anzahl an Profilen, dem Radius, einer Profiltransformationsfunktion und weiteren Merkmalen wie verwendetem Material und ggfs. Besatz mit Edelsteinen erstellt. Diese generative Beschreibung deckt den Designraum von Eheringen einer Ringmanufaktur ab. ⁰³ zeigt einige Beispiele von Ringen, die erzeugt werden können. Zusätzlich zur visuellen Repräsentation lassen sich auch zusätzliche Information in die generative Beschreibung einbinden, die es erlauben, z. B. den Materialbedarf und den Preis des Ringes zu berechnen.

□ 03

Mit der generativen Beschreibung kann eine Vielzahl von Eheringen unterschiedlichster Ausführung erzeugt werden.



Die konzeptionelle Nähe einer generativen Beschreibung zu Algorithmen und deren Implementation in Programmiersprachen ermöglicht die Verknüpfung **semantischer Informationen** mit erzeugenden Funktionen. Strukturierte Kommentare kennzeichnen Klassen, Schnittstellen und Methoden sowie zugehörige Attribute. Diese Beschreibungen umfassen in der Regel die Autorenschaft sowie den Typennamen, den Methodennamen, die Art und die Anzahl der Parameter, die Wirkung der Methoden und das Laufzeitverhalten des kommentierten Quelltexts. Diese Metadaten, die den Algorithmus und den zugrunde liegenden Erzeugungsprozess beschreiben, können auf die erzeugte Geometrie übertragen werden und so die finale Form semantisch anreichern. Auf den ersten Blick scheinen diese Metadaten minimalistisch und **flach** – vergleichbar mit dem **Dublin-Core Modell**. Der große Unterschied entsteht jedoch in der Auswertung der strukturierten Kommentare und Annotationen zur Laufzeit, d. h. während der Interpretation des Algorithmus. Die hierarchischen Strukturen des Kontrollflussgraphen und des Programmablaufs können genutzt werden, um ontologische Ordnungen und Taxonomien auf die 3D-Struktur, ihre Teilstrukturen, Variablen und Funktionen abzubilden. Neben der semantischen Anreicherung ist insbesondere die Überwindung und Verknüpfung unterschiedlicher Inhaltsformen nennenswert: Da Ontologien und Taxonomien textuell definiert sind, macht ihre Abbildung auf 3D-Teilstrukturen es möglich, in diesen textuell zu suchen. In der Softwareentwicklung werden strukturierte Kommentare und Annotationen vorwiegend intern genutzt und nur selten dem Benutzer offengelegt; im Kontext der generativen 3D-Modellierung ist es jedoch sinnvoll, diese Informationen explizit bereitzustellen.

N.5 Identifikation, Klassifikation und Rückgewinnung von Informationen

Techniken zur generativen Modellierung finden auch im Bereich von **Information-Retrieval-Systemen** vielfältige Anwendung. Retrieval-Systeme im Allgemeinen werden dazu eingesetzt, um in größeren Sammlungen Objekte anhand ihrer Ähnlichkeiten zu suchen oder zu klassifizieren. [16](#) [17](#) [18](#)

Der klassische Ansatz für dieses Problem besteht in der Extraktion sogenannter **Feature-Vektoren** aus jedem Objekt einer zu durchsuchenden Sammlung. Feature-Vektoren stellen aus mathematischer und algorithmischer Sicht eine stark vereinfachte Repräsentation der Ausgangsobjekte dar, auch wenn sie im Allgemeinen hochdimensional sind. Meist werden sie aus einer expliziten Repräsentation eines 3D-Modells extrahiert. Im Gegensatz zur impliziten Repräsentation kodieren Feature-Vektoren in der Regel nicht genügend Informationen, um die Form des ursprünglichen Objekts vollständig aus ihnen rekonstruieren zu können. Feature-Vektoren können direkt zu Zwecken der Ähnlichkeitsberechnung genutzt werden, etwa durch die Berechnung ihrer euklidischen Distanz oder anderer Metriken. [19](#)

Retrieval-Techniken können aber auch für die Klassifizierung von 3D-Objekten benutzt werden. Bei gegebener Klassifizierung einer Sammlung von 3D-Modellen (dem Objektraum) kann die Klasse eines neu hinzugefügten Objekts z. B. durch eine Ähnlichkeitssuche nach den bereits klassifizierten Modellen gefunden werden. Die Klassifizierung und die Distanz der ähnlichsten Objekte dienen anschließend als Basis für die Klassifikation des neuen Objekts. Generative Modellierung kann in diesem Zusammenhang auf vielfältige Weise genutzt werden. Die folgenden Abschnitte beschreiben nur einige ausgewählte Anwendungsszenarien, es gibt darüber hinaus noch andere. 3D-Feature-Vektoren werden meist aus expliziten Repräsentationen von 3D-Objekten extrahiert, sie lassen sich jedoch auch sehr einfach aus generativen Beschreibungen von 3D-Objekten extrahieren. Soll etwa die Ähnlichkeit zwischen verschiedenen Instanzen ermittelt werden, die aus derselben generativen Beschreibung mit unterschiedlichen Parametern erzeugt konkatiniert werden.

■ 16

Johan W. H. Tangelder, Remco C. Veltkamp, A survey of content based 3D shape retrieval methods, in: Multimedia Tools and Applications 39 (3) 2008, S. 441–471.

■ 17

Michalis Savelonas, Ioannis Pratikakis, Konstantinos Sfikas, Partial 3D Object Retrieval combining Local Shape Descriptors with Global Fisher Vectors, in: Eurographics Workshop on 3D Object Retrieval 8, 2015, S. 23–30.

■ 18

Christoph Schinko, Thomas Vosgien, Thorsten Prante, Tobias Schreck, Torsten Ullrich, Search and Retrieval in CAD Databases – a user-centric State-of-the-Art Overview, in: Proceedings of the International Joint Conference on Computer Vision and Computer Graphics Theory and Applications (VISIGRAPP), 12, 2017, S. 306–313.

■ 19

Robert Gregor, Andreas Lamprecht, Ivan Sipiran, Tobias Schreck, Benjamin Bustos, Empirical Evaluation of Dissimilarity Measures for 3D Object Retrieval with Application to Multi-feature Retrieval, in: Proceedings 13th International Workshop on Content-Based Multimedia Indexing (CBMI 15), 2015.

■ 20

Harald Grabner, Torsten Ullrich, Dieter W. Fellner, Content-based Retrieval of 3D Models using Generative Modeling Techniques, in: Proceedings of EUROGRAPHICS Workshop on Graphics and Cultural Heritage (Short Papers / Posters) 12, 2014, S. 9–12.

■ 21

Philip Shilane, Patrick Min, Michael Kazhdan, Thomas Funkhouser, The Princeton Shape Benchmark, in: Shape Modeling International 8, 2004, S. 1–12.

■ 22

Torsten Ullrich, Dieter W. Fellner, Generative Object Definition and Semantic Recognition, in: Proceedings of the Eurographics Workshop on 3D Object Retrieval 4, 2011, S. 1–8.

■ 23

Christoph Schinko, Martin Strobl, Torsten Ullrich, Dieter W. Fellner, Scripting Technology for Generative Modeling, in: International Journal on Advances in Software 4, 2011, S. 308–326.

Generative Techniken können im Kontext des Information Retrieval verwendet werden, um den Objektraum, also die Menge aller Objekte einer Objektklasse oder Domäne, aufzuspinnen. Der Objektraum als solcher wird durch einzelne **Vertreterobjekte** repräsentiert. Diese Vertreterobjekte, die aus einem konkreten Satz von Parametern erstellt wurden, dienen in der Arbeit von Grabner et al. [20](#) als Trainingsobjekte für inhaltsbasierte Retrievalmethoden: Als generative Objektklassen dienen u. a. Autos (Limousinen). Diese Objektklasse wird verwendet, um Trainingsdaten für inhaltsbasierte Retrievalmethoden zu erstellen. Obwohl keines der generierten Modelle im Princeton Shape Benchmark [21](#) enthalten ist, kann dieser mit der generativen Beschreibung klassifiziert werden; d. h. die generative Beschreibung einer Limousine ist ausreichend, um mittels Ähnlichkeitssuche alle Limousinen zu identifizieren.

Als **inverse generative Modellierung** wird eine Technik bezeichnet, die ein generatives Modell so parametert, dass es einem konkreten Objekt (vielmehr der digitalen Repräsentation davon) möglichst entspricht. In der Arbeit von Ullrich und Fellner [22](#) wird ein Algorithmus beschrieben, der folgende Fragen beantwortet:

- Kann ein konkretes Objekt (die Punktwolke eines 3D-Scans) durch ein generatives Modell beschrieben werden?
- Ist dies der Fall, was sind die Parameter des generativen Modells, um eine gute Beschreibung des konkreten Objekts zu erhalten?

Die Implementierung verwendet hierarchische Optimierungsroutinen, die auf unscharfer Geometrie basieren. Dafür wird die generative Beschreibung als Funktion mit Parametern betrachtet. Die Registrierung bzw. die Parameterschätzung basiert auf der Minimierung einer Fehlerfunktion. Der Optimierungsalgorithmus bestimmt die Parameter so, dass der geometrische Abstand möglichst vieler Punkte der Punktwolke zum generativen Modell minimal ist. Im Zuge der Optimierung ist es erforderlich, die Ableitung der generativen Beschreibung zu bilden. Schinko et al. [23](#) haben für diesen Zweck die Programmierumgebung Euclides entwickelt, welche Algorithmen mathematisch differenzieren kann.

N.6 Neue Anwendungen der Generativen Modellierung aus aktuellen Forschungsprojekten

Methoden der generativen Modellierung können die semantische Lücke zwischen Domänenwissen und Geometrie verringern. Wir besprechen im Folgenden exemplarische Forschungsergebnisse aus unserer Arbeit. Dort wird generative Modellierung zur semantischen Anreicherung und Klassifikation von Gebäudedaten und Kulturgut sowie zur Restauration von unvollständig erhaltenen archäologischen Artefakten eingesetzt.

N.7 Projekt DURAARK: Anreicherung von Gebäudedaten mit generativer Modellierung

■ 24

Chuck Eastman, Paul Teicholz, Rafael Sacks, Kathleen Liston, *BIM Handbook – A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors*, in: John Wiley & Sons, Inc., 2008.

Im Kontext des sogenannten **Building Information Modeling (BIM)** wird der gesamte Lebenszyklus eines Gebäudes digital abgebildet ²⁴ – von der Planung und Konstruktion über dessen Nutzung und Instandhaltung bis zum Abriss und der Dokumentation für die Nachwelt. Diese Daten sind für viele Gruppen von großem Interesse: für Architekten und Städteplaner, das Facility Management, aber auch für die breite Bevölkerung. Für Gebäude, die in jüngster Zeit gebaut wurden, sind viele dieser Daten bereits digital vorhanden. Für viele ältere Gebäude sind Gebäudedaten jedoch entweder gar nicht vorhanden oder nicht in einem kohärenten Modell zusammengeführt. Im Rahmen des EU-Forschungsprojektes **DURAARK** (Durable Architectural Knowledge) wurden Methoden und Werkzeuge für die automatische inhaltsbasierte Anreicherung von digitalen Gebäudedaten und deren Langzeitarchivierung geschaffen. BIM-Daten bestehen im Wesentlichen aus zwei Ausprägungen von Information: einerseits geometrischer Information, wie z. B. Grundrisspläne, 3D-Modelle, Form und Verlauf von Leitungsrohren und dergleichen, und andererseits semantischer Information, wie Materialeigenschaften, Ereigniszeitpunkte oder Verknüpfung mit für die Planung und den Bau verantwortlichen Personen und vieles mehr.

■ 25

Ran Yu, Ujwal Gadiraju, Besnik Fetahu, Stefan Dietze, *Adaptive Focused Crawling of Linked Data*, in: *16th International Conference on Web Information Systems*, Springer International Publishing, 2015, S. 554–569.

Die Datenanreicherung im DURAARK-Projekt fokussiert auf semantische Anreicherung durch Nutzbarmachung von Werkzeugen der Semantic-Web-Technologie: In der Arbeit von Yu et al. ²⁵ wurde ein sog. **focused crawler** entwickelt, welcher nach einer vorgegebenen Stichwortmenge öffentlich verfügbare Daten sucht und indiziert. Damit kann beispielsweise ein Gebäude aus dem öffentlichen Raum automatisch mit relevanten, öffentlich verfügbaren Daten verknüpft werden – wie zum Beispiel dem Verlauf des Wetters an der Gebäudeposition oder der Biografie des Erbauers. Der zweite Teil, die geometrische Anreicherung, besteht in der Erfassung von geometrischen Baubestandsdaten durch Vermessung sowie der Ableitung zusätzlicher, schwer zu vermessender Daten: Das Verfahren von Ochmann et al. ²⁶ erlaubt es, Grundrisspläne und Raumverbindungsgraphen aus Abstandsmessdaten (Punktwolken) abzuleiten sowie Elemente wie Türen und Fenster einzelnen Wänden dieses digitalen Modells zuzuordnen. Aus dieser Repräsentation wird in Kombination mit Bildinformationen (Fotos) in dem Verfahren von Krispel et al. ²⁷ die Position von Stromleitungen ermittelt: Aus Bilddaten werden sichtbare Endpunkte von Stromleitungen, wie z. B. Schalter oder Steckdosen, über einen Klassifikationsalgorithmus identifiziert, welcher mit einem repräsentativen Set von Schaltern und Steckdosen trainiert wurde. Anhand der Positionen der Wände, der bekannten Raumverbindung und den Endpunkten wird unter Einbeziehung der geltenden technischen Normen die Lage der Leitungen ermittelt – die Normen werden über einen Regelsatz beschrieben, also ein generatives System. Mit diesen Regeln wird für die endgültige Hypothese der Suchraum für die Leitungsverläufe auf die den technischen Normen entsprechenden Fälle beschränkt.

■ 26

Sebastian Ochmann, Richard Vock, Raoul Wessel, Reinhard Klein, *Automatic Reconstruction of Parametric Building Models from Indoor Point Clouds*, in: *Computers & Graphics* 54, 2016, S. 94–103.

■ 27

Ulrich Krispel, Henrik Evers, Martin Tamke, Torsten Ullrich, *An Automatic Hypothesis of Electrical Lines from Range Scans and Photographs*, in: *Proceedings 16th International Conference on Computing in Civil and Building Engineering* 7, 2016, S. 815–822.

N.8 Projekt CityFit: Klassifikation von Gebädefassadenelementen mit generativer Modellierung

Eine wichtige Aufgabe der Digitalisierung ist die Erfassung von digitalen Gebäudemodellen aus realen Gebäuden. Methoden der geometrischen Rekonstruktion von Gebäuden aus Messdaten unterliegen dabei typischerweise Ungenauigkeiten bei der Bemessung und / oder Rekonstruktion. Daher kann es von großem Nutzen sein, entsprechendes Vorwissen in die Rekonstruktion miteinfließen zu lassen beziehungsweise zu versuchen, die Design-idee des Architekten mit zu rekonstruieren. Zum Beispiel kann die Konfidenz ²⁸ eines schlecht erkannten Fensters erhöht werden, wenn im Stockwerk darunter und darüber Fenster mit hoher Konfidenz erkannt wurden und in der Fassade Fenster üblicherweise immer spaltenweise angeordnet sind.

Die bereits vorgestellten Formgrammatiken (engl. *shape grammars*) wurden im österreichischen Forschungsprojekt CityFit verwendet, um typische Strukturen der klassischen Architektur, wie z. B. Symmetrie und Wiederholung, in eine Rekonstruktion am Beispiel von neoklassischen Fassaden einfließen zu lassen. Im Verfahren von Riemenschneider et al. ²⁹ wird eine orthogonale Projektion der Fotoaufnahme einer Fassade mithilfe einer zweidimensionalen Formgrammatik analysiert. Diese Grammatik erlaubt Operationen der Unterteilung (Split) sowie in der Architektur übliche Unterteilungs- und Symmetrieregeln. Für jeden Pixel einer orthografischen Ansicht der Fassade wird zuerst eine Klassifizierung in Strukturelemente durchgeführt mit den Klassen Wand, Fenster, Tür etc. Anhand dieser Klassifizierung wird dann ein Parse-Baum der Shape-Grammatik abgeleitet, ähnlich dem Parsen von formalen Grammatiken im Compilerbau. Dieser Parse-Baum enthält die konkrete Abfolge der Grammatikoperationen, deren Auswertung Bereiche der Fassade in die entsprechenden Elemente klassifiziert. Einerseits wird durch Einbeziehen der Strukturregeln das Klassifizierungsergebnis verbessert, andererseits kann der Parse-Baum direkt in eine andere generative Beschreibung transformiert werden, welche z. B. ein 3D-Modell der Fassade generiert.

■ 28

Unter Konfidenz versteht man die statistisch-stochastische Zusicherung, dass ein berechnetes (fehlerbehaftetes) Ergebnis mit einer bestimmten Wahrscheinlichkeit zutreffend ist.

■ 29

Hayko Riemenschneider, Ulrich Krispel, Wolfgang Thaller, Michael Donoser, Sven Havemann, Dieter W. Fellner, Horst Bischof, *Irregular lattices for complex shape grammar facade parsing*, in: *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, S. 1640–1647.

N.9 Projekt ProFits: Klassifikation von Kulturgut mit generativer Modellierung

■ 30

Christoph Schinko, Martin Strobl, Torsten Ullrich, Dieter W. Fellner, *Modeling Procedural Knowledge – a generative modeler for cultural heritage*, in: *Proceedings of EUROMED 2010 – Lecture Notes on Computer Science 6436*, 2010, S. 153–165.

■ 31

Torsten Ullrich, Christoph Schinko, Thomas Schiffer, Dieter W. Fellner, *Procedural Descriptions for Analyzing Digitized Artifacts*, in: *Applied Geomatics 5 (3)* 2013, S. 185–192.

■ 32

Torsten Ullrich, Dieter W. Fellner, *Generative Object Definition and Semantic Recognition*, in: *Proceedings of the Eurographics Workshop on 3D Object Retrieval 4*, 2011, S. 1–8.

■ 33

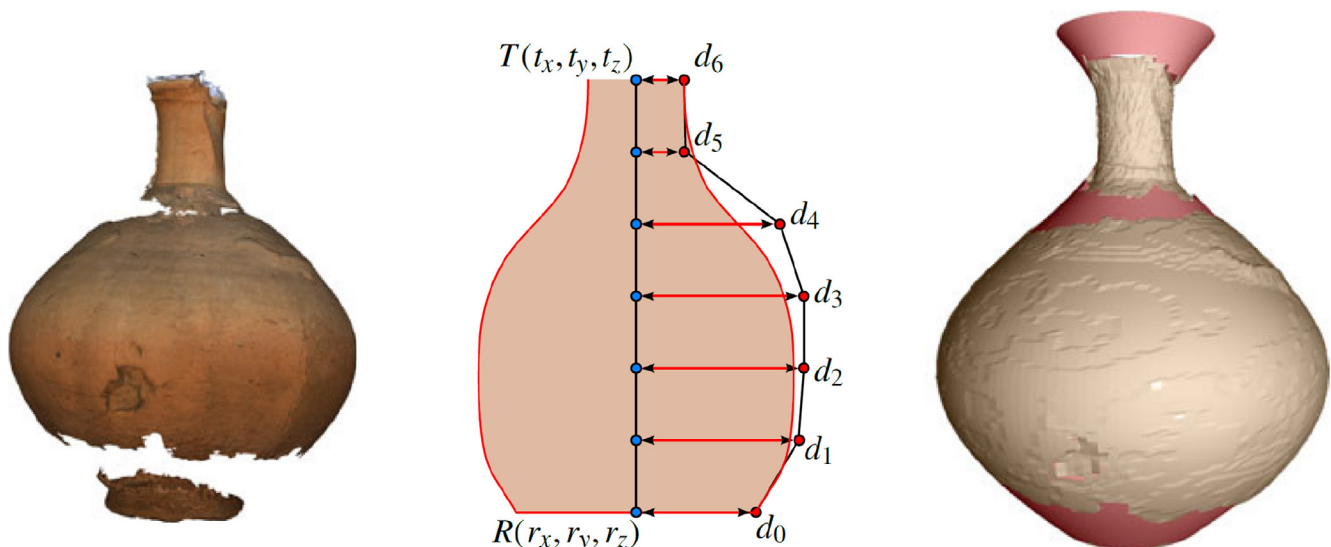
Torsten Ullrich, Volker Settgest, Dieter W. Fellner, *Abstand: Distance Visualization for Geometric Analysis*, in: *Proceedings of the Conference on Virtual Systems and MultiMedia Dedicated to Digital Heritage (VSMM) 14*, 2008, S. 334–340.

Ein generatives Modell beschreibt ein **prozedurales Wissen** ³⁰ – die Grundlage einer fundierten Schätzung der fehlenden Daten. Das folgende Beispiel illustriert eine mögliche Anwendung: ³¹ ⁰⁴ zeigt ein Artefakt der Sammlung des Museums Eggenberg in Graz sowie ein generatives Modell (in der Mitte), welches den Aufbau der Vasen beschreibt. Die generative Beschreibung hat 13 Freiheitsgrade, die in ihrer Gesamtheit das geometrische Spektrum der Familie dieser Artefakte abdecken.

In einem ersten Schritt wird die generative Beschreibung mit dem 3D-Scan registriert. Dieses sogenannte **Fitting** bestimmt die Parameter der generativen Beschreibung so, dass das algorithmische Ergebnis der Beschreibung mit genau jenen Parametern zu einem 3D-Modell führt, welches den 3D-Scan am besten beschreibt. ³² In einer anschließenden Varianzanalyse werden die Unterschiede zwischen dem 3D-Scan und der generativen Instanz berechnet. ³³ Die Abweichung beider Modelle zueinander beschreibt die Güte des Fitting-Ergebnisses. Ist das Ergebnis gut genug, kann der 3D-Scan vervollständigt werden. Dazu werden im dritten Schritt, der sogenannten Varianzdokumentation, beide Modelle zusammengefügt. Der 3D-Scan, die **verrauschte Realität**, wird dann – ohne Fehler oder Datenverlust – als Offset der idealen Beschreibung gespeichert.

□ 04

Generative Methoden ermöglichen automatische Verfahren zur semantischen Datenanreicherung. Eingangsdaten links, das generative Modell Mitte und die Vervollständigung rechts.



Dieser Ansatz bietet mehrere Vorteile. Die Registrierung und die Varianzanalyse erkennen automatisch die Zugehörigkeit eines 3D-Scans zu einer Artefaktfamilie. Ist der Fehler klein genug, kann – wie im Beispiel – ein 3D-Scan eindeutig als Vase identifiziert werden. Zusätzlich werden bei der Bestimmung der Parameter direkte Kenngrößen (Höhe, Radius etc.) bestimmt, von denen weitere Kenngrößen (Oberfläche, Volumen etc.) abgeleitet werden können. Diese automatische semantische Anreicherung macht aus einer anonymen 3D-Punktwolke einen Datensatz, der einer Objektfamilie zugeordnet werden kann und dessen Kenngrößen bekannt sind; damit kann der Datensatz verschlagwortet und somit in einer Bibliothek auffindbar gemacht werden [34] [35]. Ferner ermöglicht die Varianzdokumentation die Darstellung einer plausiblen Vervollständigung. [04]

■ 34

Volker Settgast, Torsten Ullrich, Dieter W. Fellner, *Information Technology for Cultural Heritage*, in: *IEEE Potentials* 26 (4) 2007, S. 38–43.

■ 35

Torsten Ullrich, Volker Settgast, René Berndt, *Semantic Enrichment for 3D Documents: Techniques and Open Problems*, in: *Proceedings of the International Conference on Electronic Publishing* 14, 2010, S. 374–384.

N.10 Projekt PRESIOUS: Restaurierung von archäologischen Ausgrabungsstücken mit generativer Modellierung

Im Rahmen des EU-Forschungsprojekts PRESIOUS (Predictive digitization, restoration and degradation assessment of cultural heritage objects) wurde generative Modellierung genutzt, um fragmentierte bzw. beschädigte digitalisierte 3D-Objekte aus der Archäologie zu restaurieren. [36] Oft liegen z. B. archäologische Ausgrabungsstücke in stark beschädigtem Zustand vor. Oft sind einzelne Artefakte in mehrere Teile zerbrochen und wurden während der virtuellen Restaurierung zunächst automatisiert zusammengesetzt. Einzelne Fragmente können zu stark beschädigt, unauffindbar, erodiert oder z. B. aus pragmatisch-technischen Gründen nicht digitalisierbar sein. Um auch hier zu einem plausiblen Restaurierungsvorschlag zu kommen, wurden Retrieval-Techniken untersucht, die es ermöglichen, unvollständigen virtuell zusammengesetzten Fragmenten global ähnliche, jedoch geometrisch einfache Vorlagen (Templates) zuzuordnen [37]. Diese Zuordnung erlaubt das automatisierte Auffinden von Stellen, an denen fehlende Teile vermutet werden und ermöglicht so die Visualisierung oder den 3D-Ausdruck von plausiblen Restaurationslösungen. Zum Einsatz kamen dabei also zwei Techniken: Für das Retrieval wurden Feature-Vektoren auf Basis einer expliziten Repräsentation des digitalisierten Artefakts und eines Template verwendet, zur Erzeugung der Templates diente eine generative Modellierung. Technisch realisiert wurde die generative Modellierung mit Skripten, die randomisierte Parameterwerte in eine parametrisierte CSG-Beschreibung auf Basis von

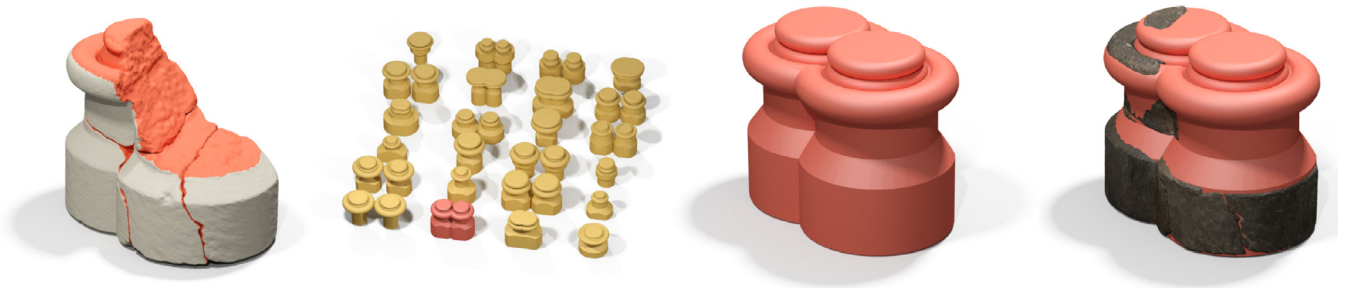
■ 36

Georgios Papaioannou, Tobias Schreck, Anthousis Andreadis, Pavlos Mavridis, Ivan Sipiran, Konstantinos Vardis, *From Reassembly to Object Completion – A Complete Systems Pipeline*, in: *Journal on Computing and Cultural Heritage* 10 (2) 2016, S. 8ff.

■ 37

Robert Gregor, Ivan Sipiran, Georgios Papaioannou, Tobias Schreck, Anthousis Andreadis, Pavlos Mavridis, *Towards Automated 3D Reconstruction of Defective Cultural Heritage Objects*, in: *Proceedings of the EUROGRAPHICS Workshop on Graphics and Cultural Heritage* 12, 2014, S. 135–144.

OpenSCAD einfügen. Die generative Modellierung dient hier der Erzeugung einer großen Anzahl von Modellen für die Formvervollständigung. Die Verwendung von in Textform repräsentierten CSG-Bäumen reduziert den benötigten manuellen Aufwand zum Erstellen der parametrisierten, impliziten Modellbeschreibung. Sobald die zugeordneten Templates im Verlauf der weiteren Restauration am Artefakt ausgerichtet worden sind, kann auf Basis von lokalen Feature-Vektoren das Kopieren von kleineren geometrischen Oberflächenstrukturen auf die Basisgeometrie des Templates (teil-) automatisiert werden. 05 illustriert diesen Prozess.



□ 05

Ein unvollständiges Kapitell (links) dient als Anfrage zur Ähnlichkeitssuche in einer Sammlung von generativ erzeugten Templates (Mitte links). Ein dem Kapitell ähnliches und anschließend passend transformiertes Template (Mitte rechts) kann dazu verwendet werden, die Form des Kapitells zu ergänzen (rechts). Alle hier gezeigten Templates sind randomisiert erzeugte Instanzen derselben generativen Beschreibung.

N.11 Zusammenfassung

Wie wir in diesem Kapitel zu erläutern versucht haben, sind generative Formrepräsentationen mächtige Werkzeuge zur Unterstützung vieler fortgeschrittener Anwendungen. Generative Verfahren können u. a. zur Suche, zur Klassifikation oder zur Formsynthese verwendet werden. Dies sind wichtige Aufgaben, auch in Bezug auf die Verarbeitung von immer größeren Formdatenmengen, welche in naher Zukunft durch verbesserte Digitalisierungsprozesse erwartet werden können.

Wir sehen in diesem Bereich offene Fragen, deren Bearbeitung sich als akademisch interessant und praktisch nützlich erweisen könnte. Wir stellen fest, dass Anwendungen, z. B. im Bereich der Modellierung oder Formrestauration, typischerweise auf bestehende generative Beschreibungen

zurückgreifen. Hier ist das Vorhandensein von Sammlungen von Modellen wünschenswert, welche viele relevante Objektklassen abbilden können. Solche generativen Modellrepositorien könnten die Arbeit in Kunst- und Baugeschichte in vielfältiger Weise unterstützen, beispielsweise durch Ableiten eines 3D-Modells aus einem annotierten Grundrissplan zur Erzeugung einer Hypothese einer historischen Gebäudeform. Weiteres Potenzial ergibt sich aus der automatisierten Vervollständigung von zerstörten Gebäuden unter Berücksichtigung der Stilepoche oder der Möglichkeit einer Stilklassifikation durch Ähnlichkeitssuche in generativen Stilrepositorien.

Uns sind derzeit keine sehr großen generativen Modellierungsrepositorien bekannt. Dies mag mit dem Erstellungsaufwand zusammenhängen, der mit der Erzeugung von generativen Formrepräsentationen einhergeht. Aktuelle Arbeiten versuchen, aus bestehenden expliziten Beschreibungen generative Prozesse zu lernen, mit denen neue, plausible Formen synthetisiert werden können. ³⁸ Dies könnte ein Schritt in Richtung Erzeugung von Formrepositorien sein. Ein damit zusammenhängendes Problem ist die Frage, wie generative Beschreibungen aus verschiedenen Ansätzen (z. B. Verwendung unterschiedlicher Algorithmen, Formoperationen etc.) ineinander übersetzt werden können, um heterogene generative Formrepositorien gemeinsam nutzen und durchsuchen zu können. Erste Ansätze dazu existieren ³⁹, die Vielzahl generativer Ansätze übersteigt die Konvertierungsmöglichkeiten jedoch deutlich.

In der Erzeugung von expliziten Repräsentationen aus parametrischen Repräsentationen sind die Parameter der generativen Verfahren so geeignet zu wählen, dass daraus das gewünschte Ausgabeobjekt resultiert. Dies ist zumeist ein iterativer Prozess, der oft durch Versuch und Fehler geleitet ist, sofern beim Benutzer kein Hintergrundwissen über die generative Beschreibung vorhanden ist. Ein einfaches Sampling des Parameterraums produziert in der Praxis oft auch nicht plausible Ergebnisse. Die Entwicklung von geeigneten Such- oder Validierungsprozessen wäre nützlich, um die Formsynthese automatisch zu unterstützen. Ein großer Vorteil generativer Verfahren ist, dass diese eine Art von Semantik über die Formerzeugung kodieren können. Dies kann potenziell zu neuen Verfahren des Formverstehens von z. B. digitalisierten Formdaten führen, etwa durch Anwendung von Klassifikations- und Suchverfahren. Während generative Verfahren bereits in vielen Spezialanwendungen eingesetzt werden, wird in anderen Bereichen mit expliziten Repräsentationen gearbeitet. Die Einbeziehung generativer Techniken in das Informatikstudium bzw. die Schulcurricula könnte deren Anwendung fördern. Ein verbreitetes Beispiel im Bereich 2D-Formrepräsentation ist z. B. die Verwendung der sogenannten Turtle-Metapher in der Ausbildung der Programmiersprache LOGO bzw. der computergestützten Geometrieerzeugung. Es könnte pädagogisch interessant sein, an vergleichbaren generativen Ansätzen für die 3D-Geometrie zu arbeiten.

■ 38

Haibin Huang, Evangelos Kalogerakis, Benjamin Marlin, *Analysis and Synthesis of 3D Shape Families via Deep-learned Generative Models of Surfaces*, in: *Computer Graphics Forum* 34, 2015, S. 25–38.

■ 39

Martin Strobl, Christoph Schinko, Torsten Ullrich, Dieter W. Fellner, *Euclides – A JavaScript to PostScript Translator*, in: *Proceedings of the International Conference on Computational Logics, Algebras, Programming, Tools, and Benchmarking (Computation Tools)* 1, 2010, S. 14–21.