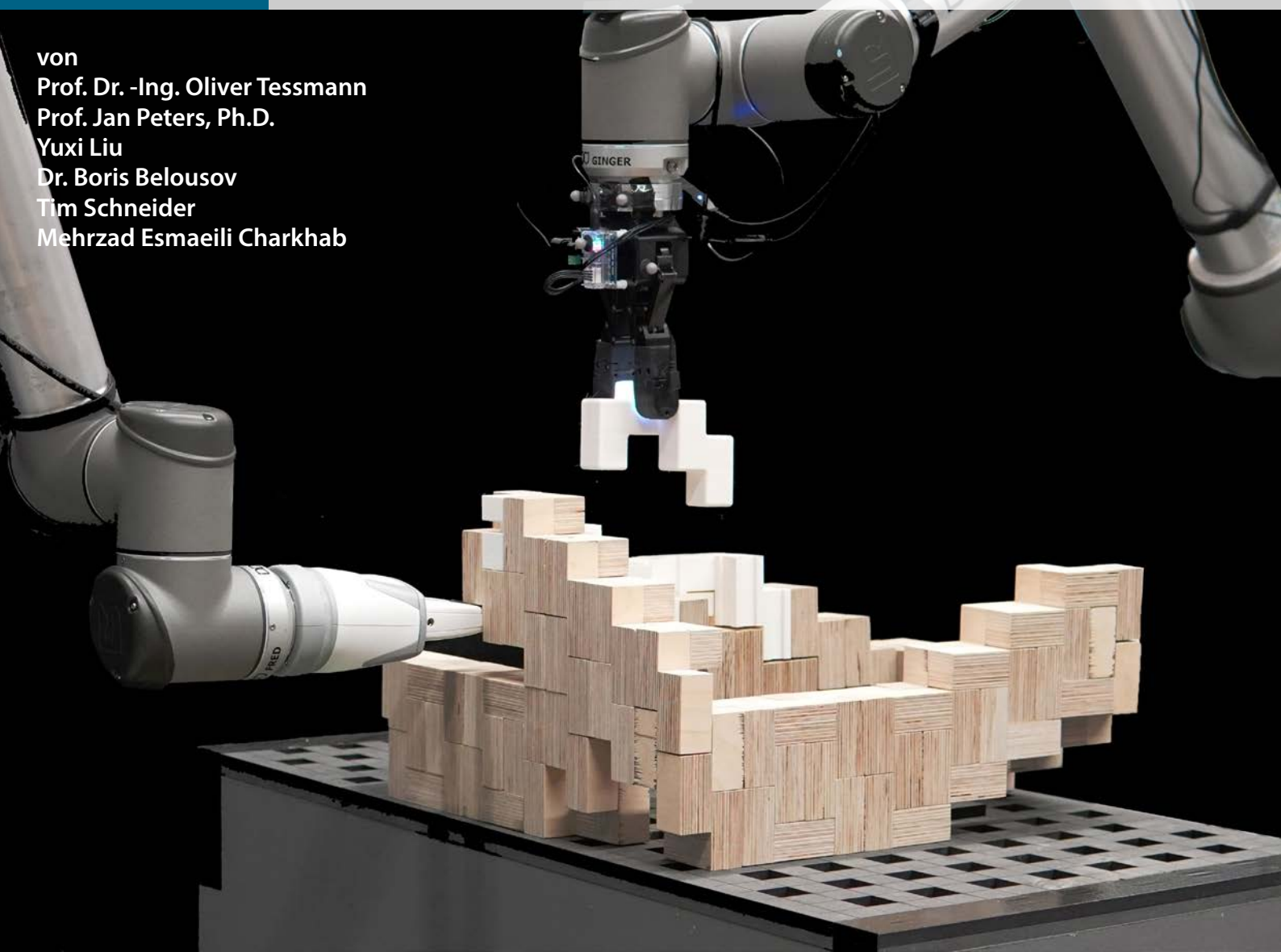


Tactile Robotic Assembly

Autonomes Fügen modularer Konstruktionen durch Roboter, Tastensensorik und Künstliche Intelligenz

BBSR-
Online-Publikation
05/2024

von
Prof. Dr. -Ing. Oliver Tessmann
Prof. Jan Peters, Ph.D.
Yuxi Liu
Dr. Boris Belousov
Tim Schneider
Mehrzad Esmaeili Charkhab



Tactile Robotic Assembly

Autonomes Fügen modularer Konstruktionen durch Roboter, Tastensensorik
und Künstliche Intelligenz

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: 10.08.18.7-21.34

Projektlaufzeit: 08.2021 bis 01.2024

IMPRESSUM

Herausgeber

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

Fachbetreuerin/Fachbetreuer

Bundesinstitut für Bau-, Stadt- und Raumforschung
Referat WB3 „Forschung und Innovation im Bauwesen“
Dr. Michael Brüggemann
zb@bbr.bund.de

Autorinnen und Autoren

Technische Universität Darmstadt, Darmstadt
Fachbereich Architektur, Fachgebiet Digitales Gestalten
Fachbereich Informatik, Intelligent Autonomous Systems Group

Prof. Oliver Tessmann (Projektleitung)
Prof. Jan Peters
Yuxi Liu, M. Sc.
Mehrzad Esmaeili Charkhab, M. Sc.
Tim Schneider, M. Sc.
Dr. Boris Belousov

Redaktion

Technische Universität Darmstadt, Fachgebiet Digitales Gestalten
Yuxi Liu

Stand

Januar 2024

Gestaltung

Technische Universität Darmstadt, Fachgebiet Digitales Gestalten
Yuxi Liu

Bildnachweis

Alle Abbildungen: Technische Universität Darmstadt, Fachgebiet Digitales Gestalten und Intelligent Autonomous Systems Group

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

Tessmann, Oliver; Peters, Jan; Liu, Yuxi; Esmaeili Charkhab, Mehrzad; Schneider, Tim; Belousov, Boris, 2024: Tactile Robotic Assembly: Autonomes Fügen modularer Konstruktionen durch Roboter, Tastensensorik und Künstliche Intelligenz. BBSR-Online-Publikation 05/2024, Bonn.

Inhaltsverzeichnis

Kurzfassung	6
Abstract	7
1. Einführung	8
1.1 Themenfeld und Untersuchungsgegenstand	8
2. Problemstellung	9
2.1 Stand der Forschung	9
2.2 Forschungslücke/Entwicklungsbedarf	13
3. Zielstellung	14
3.1 Projektziele	14
3.2 Fragestellung	14
3.3 Methodischer Ansatz	15
3.4 Projektorganisation und Kooperationspartner	15
3.5 Arbeitspakete und Meilensteine	16
4. Projektverlauf	18
4.1 Das SL-Block System	18
4.1.1 Geometrisches Element: Der SL-Block	19
4.1.2 Kombinatorische Prinzipien von SL-Blocks und SL-Strands	20
4.1.3 Computergestützte Generierung von SL-Strands	24
4.1.4 Bottom-Up Entwurfsexperimente	30
4.2 Top-Down Entwurfsstrategien für das hierarchische Fügen von SL-Blöcken	32
4.2.1 Bauteilhierarchien im SL-Block System	32
4.2.2 Graphenbasierte Analyse der SL-Hierarchie	35
4.2.3 Hamiltonkreisproblem	38
4.2.4 3D Polyomino Kachelung	43
4.3 Materialisierung von SL-Blöcken	44
4.4 Visuelle taktile Sensorsysteme	46
4.4.1 DIGIT Sensor	47
4.4.2 Erkennung von Materialien durch Roboter mit visuell-taktilen Sensoren mittels KI	48
4.4.3 Berechnung der Kontaktkräfte und Drehmomente für die Greif- und Pfadplanung	48
4.5 Low-Level-Robotersteuerung für das Einsetzen von SL-Blöcken unter Verwendung von taktilem Feedback und Verstärkungslernen	50
4.5.1 Das Assembly Gym Framework	52
4.5.2 Robotische Konstruktionsaufgabe	55
4.5.3 Repräsentation taktiler Daten des DIGIT Sensors	56
4.5.4 Datenerfassung aus menschlichen Demonstrationen	58
4.5.5 Steuerung auf dem realen System (Implementierung)	60

4.6 SL-Blöcke, optimiert für die robotische Fügung	63
4.6.1 Entwurf und Fertigung von selbstkalibrierenden SL-Blöcken	64
4.6.2 Aufbau der Montageumgebung	68
4.6.3 Aufgaben- und Bewegungsplanung für den Roboter	69
4.6.4 Autonome robotische Montage	71
5. Ergebnisse	74
5.1 Demonstratoren	76
5.2 Bewertung/Diskussion/Zielerreichung	80
5.3 Ergebnistransfer	82
6. Fazit & Ausblick	84
7. Danksagung und Mitwirkende	86
8. Literaturverzeichnis	88
9. Abbildungsverzeichnis	93

Kurzfassung

Unsere Gesellschaft steht vor der globalen Herausforderung, die enormen notwendigen Bautätigkeiten der kommenden Jahrzehnte mit einer Industrie zu bewältigen, deren Produktivitätszuwächse gering sind und die unter einem Fachkräftemangel leidet. Gleichzeitig sind Innovationen notwendig, die eine drastische Reduzierung der CO₂-Emissionen und des Abfallaufkommens im Bauwesen ermöglichen. Ein wichtiger Baustein zur Meisterung dieser Herausforderung ist die enge Verknüpfung der gesamten Prozesskette vom Entwerfen, über das Materialisieren und Vorfertigung bis Bauteilmontage mit Hilfe von Künstlicher Intelligenz, Sensorik und Automatisierung. Moderne Sensorik erlaubt den Einsatz von Robotern auch ohne Abschirmung von Menschen. Automatisierung kann so schrittweise, kollaborativ und sozialverträglich erfolgen und in einer extrem kleinteiligen Industrie angemessen implementiert werden.

Dieses Projekt widmet sich dem letzten Arbeitsschritt der Bauteilmontage durch autonom agierende Roboter. Konkret werden prototypisch modulare Bausysteme aus seriell, industriell und wirtschaftlich hergestellten Elementen entwickelt. Die reversibel gefügten Elemente werden durch Roboter zu verzahnten Konstruktionen gefügt, demontiert und in anderer Form re-montiert. Unterschiedliche Fügeprinzipien werden untersucht, um Material- und Funktionsdifferenzierung (Tragen, Klima regulieren, Licht lenken etc.) von Elementen innerhalb einer Konstruktion zu ermöglichen. Die Elemente sind im Sinne des „Robot-Oriented_Design“ für die robotergestützte Montage optimiert. Die daraus erstellten Konstruktionen sind im Sinne der Kreislaufwirtschaft reversibel gefügt und durch Roboter demontierbar.

Das sequentielle Montieren zu Bauteilen eines Demonstrators erfolgt auf der Grundlage eines 3D-Modells, wobei das Maß der vorab programmierten Bewegungspfade der Roboter minimiert wurde. Vielmehr bestimmen Roboter ihre Bewegungsabläufe selber und „begreifen“ Gewicht, Dimension und Oberflächeneigenschaften von Bauteilen durch visuotaktile Sensorik. Maschinelles Lernen und Sensorik helfen dabei, Bauaufgaben, die derzeit aufgrund ihrer Komplexität nicht vorab zu programmieren sind oder eine permanente Anpassungsfähigkeit des Systems erfordern, autonom durchzuführen.

Das Projekt untersucht, wie architektonisch/konstruktive Aspekte für das verstärkende maschinelle Lernen (Reinforcement Learning, RL) als Belohnungen (Rewards) repräsentiert werden können. Stabile Konstruktionen werden beim verstärkenden Lernen belohnt (positive rewards). Herabfallende Elemente führten zu Bestrafungen (negative rewards), so soll der Roboter über KI und Reinforcement Learning ein Verständnis für Konstruktion und stabile Bauzustände entwickeln. Außerdem wurden algorithmische Entwurfswerkzeuge für elementierte Konstruktionen entwickelt, mit denen die neu gewonnenen gestalterisch/konstruktiven Möglichkeiten autonom bauender Roboter vollumfänglich ausgeschöpft werden können. Prototypen, Bauteile und Demonstrator-Konstruktionen zeigen exemplarisch die Anwendungsmöglichkeiten der im Projekt neu entwickelten Prozesse und Methoden.

Abstract

Our society faces the global challenge of serving an enormous demand for housing now and in the coming decades with an industry whose productivity is traditionally low and which suffers from a shortage of skilled workers. At the same time, innovations are needed to dramatically reduce CO2 emissions and waste in construction. An important aspect of mastering this challenge is a continuous digital process chain from design, materialization, and prefabrication to finally component assembly, supported by artificial intelligence, robotics, and automation. Sensor technology allows the use of robots in collaboration with humans. Automation can thus be gradually implemented in the extremely fragmented construction industry.

This project is dedicated to the prototypical development of a modular building system and its assembly by autonomous robots. First, the interdisciplinary team of the Intelligent Autonomous System Group (IAS) and the Digital Design Unit (DDU) developed elements that can be dry-joined into interlocking assemblies. Elements differ in material and function, enabling load transfer, moisture regulation, light transmission, etc. The material system is optimized to be assembled, disassembled, and reassembled by robots. The sequential assembly into a demonstrator structure, including all assembly steps, is based on an abstract description (3D model).

Robots then determine their Task and Motion Planning (TAMP) and “understand” the weight, dimensions, and surface properties of components using visuo-tactile sensors. Machine learning and sensor technology help to autonomously perform construction tasks that currently cannot be programmed in advance due to the need for permanent adaptability.

The project investigates how architectural/construction aspects can be represented as rewards for Reinforcement Learning (RL). Stable constructions are rewarded in RL (positive rewards). Falling elements led to punishments (negative rewards), so the robot develops an understanding of construction and stable construction states via AI and RL. Algorithmic design tools for element-based construction are developed to fully exploit the novel design/construction capabilities of autonomous building robots.

1. Einführung

1.1 Themenfeld und Untersuchungsgegenstand

Das Projekt erforscht konstruktive, nachhaltige und produktivitätssteigernde Potentiale von Digitalisierung und künstlicher Intelligenz in der Architektur. Im Projekt wird maschinelles Lernen beim Bauen eingesetzt.

Konkret führen autonom agierende Roboter eigenständig Füge- und Montageprozesse aus. Das vollständige sequentielle Zusammenfügen zu einer Struktur (Bauteile und Demonstrator) aus verfügbaren Teilen inklusive aller Montageschritte erfolgt auf der Grundlage eines 3D-Modells. Roboter bestimmen dabei ihre Bewegungsabläufe selber und "begreifen" Gewicht, Dimension und Oberflächeneigenschaften von Bauteilen durch visuo-taktile Sensorik.

Maschinelles Lernen und Sensorik hilft dabei, Bauaufgaben, die derzeit aufgrund ihrer Komplexität nicht vorab zu programmieren sind oder eine permanente Anpassungsfähigkeit des Systems erfordern, autonom durchzuführen. Das Projektteam entwickelt algorithmische Entwurfswerkzeuge für elementierte Konstruktionen, mit denen die neu gewonnenen gestalterisch/konstruktiven Möglichkeiten autonom bauender Roboter vollumfänglich ausgeschöpft werden können.

Die im Projekt gewonnenen Erkenntnisse sind anhand von Prototypen und Demonstrator-Konstruktionen exemplarisch angewendet und evaluiert worden.

2. Problemstellung

2.1 Stand der Forschung

Das Projekt verknüpft Forschung zu trocken gefügten Bausystemen, digitalen Entwurfswerkzeugen für modulare und multi-materiale Konstruktionen mit robotisch autonomen Fügeprozessen auf der Basis von verstärkendem Lernen (Reinforcement Learning (RL)) und visuotaktile Sensorik. Es zielt auf eine durchweg digitale Planung und Fertigung für das Bauwesen im Sinne der Industrie 4.0 (Braun et al., 2019). Im Folgenden wird der Stand der Forschung hinsichtlich trocken gefügter Konstruktionen, digitalem Material, robotischem Fügen und RL für Assemblierung mittels taktile Sensorik erläutert.

Modularität

Digitale Entwurfswerkzeuge (Parametric Design, BIM) ermöglichen das Beschreiben von dreidimensionalen Objekten, die durch veränderliche Eingabeparameter in ihrer Form bestimmt und variiert werden. In einem modularen System kann dabei jedes Einzelbauteil zum Unikat mit nur einer einzigen Einbauposition werden, vergleichbar mit einem Puzzleteil, das nur an einem Ort zum großen Ganzen passt. Dieses Projekt verfolgt einen grundsätzlich anderen Ansatz, nämlich das Erzeugen von Großformen auf Basis von Familien serieller Bauteile, die durch geschickte Kombinatorik diese Großform bilden. Bauteile können zu unterschiedlichen Großformen gefügt, demontiert und anschließend zu anderen Formen gefügt werden. LEGO Bausteine sind ein sehr einfaches Beispiel für solche seriellen Module, die sehr vielfältig kombiniert werden können, ohne auf nur eine einzelne vordefinierte Großform reduziert zu sein. Neil Gershenfeld vom MIT Center for Bits and Atoms schlägt vor, den Begriff des Digitalen von den computernumerisch kontrollierten (CNC) Prozessen der Fertigung auf die Materialien selbst auszudehnen (Gershenfeld, 2012). Ein "digitales Material" ist nicht kontinuierlich und homogen wie Stahl, sondern besteht aus diskreten repetitiven Elementen, die durch reversible Verbindungen ein großes Ganzes bilden. Alle Elemente haben selbstkalibrierende geometrische Eigenschaften, die die Fügung präzise steuern.

Auch hier illustriert der Klemmbaustein (LEGO) mit seinen eindeutig definierten Klemmnoppen das Prinzip in Abgrenzung zum Mauerwerksstein, der eine ausgleichende Mörtelschicht benötigt, die eine nicht-reversible Verbindung erzeugt. Anders als beim kontinuierlichen 3D-Druck, können die Elemente aus unterschiedlichen Materialien bestehen, solange sie eine gemeinsame Fügelogik besitzen. Digitale Materialien bieten so volle Reversibilität und digitale Programmierbarkeit physischer Prozesse. Material und Funktionsdifferenzierung kann innerhalb eines Systems erfolgen. Digitale Materialien werden nicht 3D gedruckt, sondern 3D assembliert. Beim Konstruieren mit digitalem Material im architektonischen Maßstab (Gershenfeld et al., 2015) wird die Elementierung als gestaltende Kraft betrachtet und nicht als nachträgliche Rationalisierung der Großform. Rechenleistung des Computers und die richtigen digitalen Entwurfs-, Fertigungs- und Assemblierwerkzeuge ermöglichen es, mit großen Mengen von Elementen zu entwerfen und die Korngröße der Elementierung deutlich zu reduzieren. Element ist nicht mehr identisch mit Bauteil (Stütze, Wand, Platte), sondern ist potentiell nur noch ein Baustein von dutzenden, hunderten oder tausenden, die gemeinsam trocken gefügte Bauteile bilden.

In der Entwurforschung arbeiten die Architekten Jose Sanchez und Gilles Retsin mit kombinatorischen Entwurfsprozessen und selbst entwickelten digitalen Werkzeugen (Sanchez, 2016, Retsin, 2016). Die Architekten testen neue gestalterische Möglichkeiten, die sich aus Elementen und ihren Fügeregeln ergeben.

Trocken gefügte, verzahnte Elemente

Trocken und reversibel gefügte, verzahnte Systeme tauchen zu allen Zeiten in der Architekturgeschichte auf. Auf einem Zeitstrahl haben wir Systeme, Konzepte und realisierte Projekte angeordnet, um einen Überblick über bereits verwendete Systeme zu bekommen, aber auch um Konzepte, die sich nicht durchgesetzt haben, genauer zu analysieren. Leonardo da Vincis Zeichnungen zu Hebelstabwerken und Joseph Abeilles Flat Vault, aber auch Beispiele aus dem Holzbau in Asien gehören dazu. Die vielversprechendsten Systeme haben wir 3D nachmodelliert in einer Matrix miteinander verglichen (vgl. Abbildung 1). Eine grundlegende Anforderung an solche modularen Systeme ist, dass die Bauteile gemeinsam eine lastabtragende, raumbildende, überspannende oder sogar auskragende Konstruktion bilden und gleichzeitig für eine Nachnutzung reversibel von Robotern gefügt werden können. Projektrelevante Forschung findet sich im Bereich, Architektur, Bauingenieurwesen, Material- und Computerwissenschaften. Außerdem zeigen Prototypen und erste experimentelle Anwendungen Potentiale für den Transfer in die Praxis. Diese Arbeiten bilden die Grundlage, Ausgangsbasis und Kontext für dieses Projekt.

- Verzahntes Digitales Material:** Jonathan Hiller und Hod Lipson untersuchen auf Basis des Digitalen Materials räumliche Gitterpackungen, die über verzahnende Bestandteile gefügt werden. Es werden Konstruktionskriterien und selbst-kalibrierende 3D-Voxelgeometrien vorgestellt. Untersucht wird eine schnelle, wirtschaftliche, parallele Montage von mikroskaligen Voxeln mit dem Ziel einer digitalen Fertigung von perfekt wiederholbaren, präzisen Multi-Material-Objekten. Die Forschung ist für dieses Projekt relevant, weil skalierbare geometrische Eigenschaften und Anforderungen von digitalen Materialien und ihren Fügeprozessen untersucht werden (Hiller und Lipson, 2009).

Abbildung 1

Ausschnitt aus einer Rechertabelle zu trocken gefügten Konstruktionsprinzipien und ihren Eigenschaften. Spalte 1: Wachsmann Kubus, bestehend aus 12 identischen verzahnten Kuben. Spalte 2: Offene Fingerzinkung. Spalte 3: Dreidimensionale Überblattung. Spalte 4: Dreidimensionale Hakenblattung. Spalte 5: Klemmbausteine. Spalte 6: Topologisch verzahnte Tetraeder. Spalte 7: Räumliche Packung von vier verschiedenen verzerrten Tetraedern (Danzer projectiles). Spalte 8: Räumliche Packung von Polyedern. Spalte 9: Aggregierte, discrete Elemente. Spalte 10 SL-Blöcke.

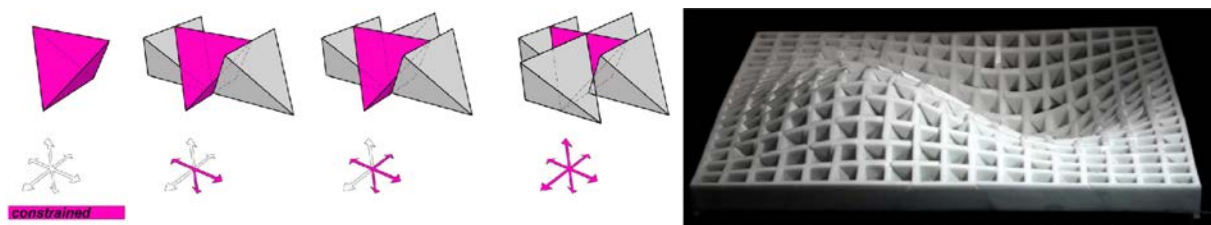
SYSTEM	Wachsmann Cube	Simple Box Joinery	Kengo Kuma Joinery	Complex Jap. Joinery	LEGO	Tetrahedron	Danzer Packing	Convex Polyhedra	Gilles Retsin Modules	SL Block
Application	Constr. Knot	Furniture	Architecture	Wood Work	Kid's Toy	Platonic Solid	Space Packing	Research	Architecture	Kid's Toy
Attributes	wooden knot connecting joint light weight	handcraft connecting joint timber constr.	ARCH joinery linear system timber interlocking	complex joinery interlocking logic handcraft	playful assembly global design digital fabrication	mathematical geom. study regular polyhedra	studies geom. packing geometry-filling	irregular polyhedra geom. study space-filling	experimental digital fabrication architectural scale	playful assembly global design digital fabrication
Type / Logic	Voxelar Grid compact	Cubic Grid angular	Connection Grid linear	Subtraction crossing	Quadratic Grid stacking	Equilateral symmetrical	Packing solid	Convex polyhedral	Quadratic Grid directioning	Voxelar Grid octocubic
System	JOINERY SYSTEM					OPEN SPACE SYSTEM				
Properties	Parametrization ●●●●● Simplicity ●●●●● Flexibility ●●●●●	●●●●●	●●●●●	●●●●●	●●●●●	●●●●●	●●●●●	●●●●●	●●●●●	●●●●●
Modules										
Feature	COMPACT	SIMPLICITY	THREE-DIMENSIONAL	RESILIENCE	FLEXIBILITY	GEOMETRICAL INTERLOCKING	PACKING	SPACE-FILLING	ARCHITECTURAL DESIGN	SELF-INTERLOCKING

Quelle: DDU, TU Darmstadt 2021

- Rekursiv verzahnte Elemente:** Wissenschaftler der Nanyang Technological University und der Tel Aviv University entwickeln eine mathematisch formalisierte Methode zum Entwurf rekursiv verzahnter 3D Puzzles. Eine bestimmte Sequenz des Zusammensetzens oder Zerlegens sorgt auch für Verzahnung und Stabilität während des Bauens (Song et al., 2012). Das gleiche Team entwickelt 2017 den Ansatz für verzahnte, rekonfigurierbare Möbel weiter. Ein Graph bildet den Möglichkeitsraum unterschiedlicher Konfigurationen ab (Song et al., 2017). Die hier vorgeschlagene Methode ist für das Projekt relevant, da sie Geometriegenerierung und ihre gleichzeitige Verzahnung miteinander verknüpft und über Graphen die kombinatorischen Möglichkeiten repräsentiert.
- Topologisch verzahnte Elemente/Bauteile:** Die topologische Verzahnung ist ein Konstruktionsprinzip, bei dem Elemente nur durch ihre Geometrie und die Topologie ihrer Fügung ein frei spannendes, nicht monolithisches, reversibles Tragsystem bilden (vgl. Abbildung 2). Eine Randeinspannung oder Spannkabel nehmen den Horizontalschub auf. Die Elementierung erlaubt das Fügen von Bauteilen aus unterschiedlichen Materialien. Die trockene Verbindung erlaubt die Wiederverwendung der Bauteile. Die Fragmentierung verhindert den Versagensfall durch sich fortpflanzende Rissbildung. Yuri Estrin erforscht topologisch verzahnte Strukturen als Multi-Material Systeme (Estrin et al., 2011). Mather et al. entwickeln Finite-Elemente (FE) Modelle zur Untersuchung der mechanischen Eigenschaften solcher Systeme (Mather, et al., 2011). Weizmann et al. untersuchen die Korrelation zwischen Blockgeometrie und strukturellem Verhalten in flachen Baugruppen aus konvexen ineinandergreifenden Blöcken (Weizmann et al., 2021). Giuseppe Fallacara erforscht historische verzahnte Steinkonstruktion und entwickelt daraus zeitgenössische, mit Robotern geschnittene Konstruktionen aus Stein (Fallacara, 2018).

Abbildung 2

Links: Tetraeder, die sich gegenseitig stützen. Je mehr Objekte miteinander verzahnt werden, umso weniger Freiheitsgrade hat der mittlere Tetraeder. Rechts: Ein 3d gedrucktes Modell einer topologisch verzahnten Struktur mit parametrisch ausdifferenzierten Elementen.



Quelle: DDU, TU Darmstadt 2019

Die Architekten AAU ANASTAS haben für den Museumsshop des Benediktinerklosters in Abu Ghosh (St. Mary of the Resurrection Abbey) eine topologisch verzahnte Deckenkonstruktion realisiert (www.auanastas.com, aufgerufen: 25.04.2021). Ihr Projekttitle "The flaut vault" verweist auf eine Erfindung von Joseph Abeille aus dem Jahr 1699. Abeille präsentiert aus Stein geschnittene, an zwei Kanten abgeflachte Tetraeder und ostomorphe Blöcke, die topologisch verzahnt werden. Ossoinack (1955) und Murphy (1967) patentieren Herstellung und Fügung topologisch verzahnter, trocken gefügter Mauerwerkssysteme. Youssef, Erler, Jäger von der TU Dresden erforschen im Rahmen eines ZukunftBau Projekts „REMOMAB“ rückbaubare Lösungen im Mauerwerksbau (Youssef et al., 2019). Die Beispiele sind für dieses Forschungsprojekt relevant, da sie aus den Anforderungen des Bauens entwickelt wurden. Die junge Firma Triqbriq arbeitet mit kleinteiligen Verschnittresten im Holzbau, um aus diesen Mirco-Modulen trocken fügbare Holzbausteine herzustellen (triqbriq.de, 2023). Das Institut für Tragwerksentwicklung (ITE) von Prof. Harald Kloft an der TU Braunschweig, forscht an präzisen Trockenfugensystemen für dünnwandige Elemente aus UHPFRC (Ultrahochfestem faserverstärkten Beton) und entwickelt präzise neuartige Verbindungen (Zahnverbindung, Inlays etc.) für geometrisch komplexe Flächen- und Stabwerkselemente aus UHPFRC zur Übertragung von Druck-, Biege- und Scherkräften (Mainka et al., 2013). Die Trockenfugen und Verbindungen mit hoher Passgenauigkeit und hoher Tragfähigkeit wurden mit hochpräzisen CNC-gefrästen Schalungen aus Wachs oder

Kunststoff realisiert (Lehmberg et al., 2016). Die Forschung ist für dieses Projekt relevant, weil mit den entwickelten Trockenfugensystemen im architektonischen Maßstab Bauteile wie dünnwandige Segmentträger mit hoher Tragfähigkeit realisiert werden können. Roberto Naboni und Anja Kunic entwickeln komplexe elementierte Holzbauweisen, bei denen topologieoptimierte kontinuierliche Formen in modularisierte Konstruktionen übersetzt werden. Die Forschung ist relevant, weil sie die gesamte Prozesskette vom kombinatorischen Entwerfen bis zur Fügung der Bauteile als kollaborativen Prozess von Menschen und Robotern untersucht (Naboni und Kunic, 2019).

- **Robotergestützte Montage in der Architektur:** Die automatisierte Montage ist im Vergleich zu anderen Industriezweigen in der Architektur noch wenig entwickelt. Obwohl Roboter für das Bauwesen aufgrund der aktuellen sozioökonomischen Veränderungen der Gesellschaften (Fachkräftemangel, Demographie) zu einer Notwendigkeit werden (Brehm, 2019).

Aktuelle Entwicklungen in der Robotik, im maschinellen Lernen und der Sensorik eröffnen die Möglichkeit, Automatisierung für komplexe und autonome Bauaufgaben mit neuen Ansätzen zu erforschen. Es besteht Forschungsbedarf, weil die digitalen Prozessketten heutiger Bauprozesse Lücken aufweisen (Melenbrink et al., 2020; Wibranek et al., 2019). Der Einsatz von Robotern im Bereich der Fügung von modularen Bauteilen kommt derzeit in der Praxis noch nicht zum Einsatz. Das Robotic assembly system for Computer integrated CONstruction (ROCCO) wurde von 1992 bis 1996 entwickelt und war in der Lage, schwere Ziegelsteine zu heben (T. Bock, 2017). Das Forschungsprojekt hatte aufgrund der komplexen Handhabung von Maßtoleranzen und der zeitaufwendigen Neupositionierung des Roboters auf der Baustelle keinen kommerziellen Erfolg. Die Weiterentwicklung ROCCO sollte ein spezielles, für die Roboter montage optimiertes Ziegelsystem verwenden (Bonwetsch, 2015, S. 45), was bereits auf die Schwierigkeit hinweist, Roboter für die Montage von Bauelementen zu nutzen, die für die menschliche Montage konzipiert sind. Heute existiert ein kleines Spektrum an kommerziellen Projekten im Bereich der Mauerwerksroboter wie der SAM100 (Semi-Automated Masonry) (SAM100 - Construction Robotics, o.J.) und der Hadrian X ROBOT (Hadrian X® | Outdoor Construction & Bricklaying Robot von FBR, o.J.). Gleichzeitig gibt es ein umfangreiches Forschungsinteresse im Bereich der Mauerwerksautomatisierung (Dakhli & Lafhaj, 2017). An der ETHZ forscht das Team um Fabio Gramazio und Matthias Kohler daran, Bauteile aus Stein, Holz und Stahl robotergestützt zu fertigen und zu fügen (Thoma et al. 2018; Robeller et al. 2017; Parascho et al., 2017). Das ICD in Stuttgart erforscht das fliegende Fügen von Bauteilen (Aflalo et al., 2017) und Konzepte der Mensch-Roboter Interaktion (ICD, 2020). An der RWTH Aachen erforscht Prof. Sigrid Brell-Cokcan mit ihrem Team den Einsatz von Mensch-Roboter Interaktion für Bauprozesse (Stumm et al. 2017). Wu und Kilian erforschen die Nutzung von Robotern als Ersatz für temporäre Stützkonstruktionen von instabilen Bauzuständen während der Montage (Wu und Kilian, 2018). Robeller et al. untersuchen die Nutzung von Robotern für Einpressverbindungen im Holzbau (Robeller et al., 2017).

Darüber hinaus erforschen verschiedene Teams die Entwicklung von Bauelementen, die speziell auf die robotergestützte Montage abgestimmt sind, so dass auf Potentiale und Beschränkungen der Robotertechnologie und Sensorik gezielt eingegangen werden kann. Thomas Bock spricht 1988 erstmals von roboterorientiertem Design (ROD). Dabei werden roboterbezogene Parameter in den frühen Entwurfs- und Produktionsphasen von Bauteilen berücksichtigt, um definierte Bedingungen für den Robotereinsatz vor Ort zu schaffen (Bock, 1988). Rogeau et al. entwickeln Elementverbindungen für die Roboter montage von Holzplattenkonstruktionen. Die Holzverbindungen wurde abgeschrägt, um Materialtoleranzen zu berücksichtigen und eine automatische Kalibrierung zu ermöglichen (Rogeau et al., 2020). Ein Baustein speziell für die robotische Montage wurde für die Acoustic Brick Wall von Gramazio und Kohler entwickelt. Die Geometrie wird durch einen selbst korrigierenden "Fall-in-Place"-Mechanismus bestimmt, um eine horizontale Fehlstellung automatisch zu vermeiden (Vomhof et al., 2014). Ariza und Gazit definieren die Elementgröße und das Fügedetail eines pilzförmigen Daches unter Berücksichtigung der maximalen Reichweite des Montageroboters (Ariza & Gazit, 2015). Für den Einsatz von Drohnen zum Platzieren von Bausteinen entwickeln Goessens et al. leichte Elemente, die sich während der Montage selber ausrichten (Goessens et al., 2018).

Eine weitere Richtung in der Forschung zu robotergestützter Montage ist die Modularisierung von Robotern, die mit einem zu montierenden Materialsystem nahezu verschmelzen. Diese Material-Roboter-Systeme (MaRS) sind mobile Roboter, die diskrete zelluläre Strukturen zusammensetzen können (Jenett & Cheung, 2017). Der Roboter und das Material sind so konzipiert, dass sie ein kohärentes System bilden, in dem beide Einheiten aufeinander abgestimmt sind. Die Elemente sind für die robotische Montage ausgelegt, und die Roboter sind für ihre Montage konstruiert, so dass die technischen und materiellen Rahmenbedingungen beider Seiten berücksichtigt werden (N. Gershenfeld et al., 2015). Das Projekt Distributed Robotic Assembly System for In-Situ Timber Construction von Samuel Leder und Ramon Weber am ICD Stuttgart nutzt Holzbauteile temporär als Teil des Roboters. Um die Funktionalität eines Roboterarms zu erreichen, greifen mobile Gelenke die Stäbe und bilden einen 6-achsigen Roboterarm, der sich nach Montieren des Bauteils selbst demontiert und wieder neu zusammensetzt (Leder et al., 2019). An der Bartlett School of Architecture (UCL) konzipieren Dafni Katrakalidi, Martha Masli, Mengyu Huang, Man Nguyen und Wenji Zhang im Projekt AssemblerAssemble einen Roboter, der zu einem Teil des Gebäudes wird und untrennbar mit ihm verbunden bleibt (Retsin, 2019).

- **Reinforcement Learning und taktiles Feedback für robotisches Fügen:** Das Bauen modularer Strukturen durch autonome Roboter wurde von Hartmann et al. betrachtet. Die Autoren benutzen eine idealisierte Task-and-Motion-Planung zur Berechnung von Bewegungspfaden des Roboters und setzen das Greifen von Teilen ohne Schlupf voraus (Hartmann et al., 2020). Sartoretti et al. entwickeln einen Multiagenten-Ansatz für ein Montageproblem. Ein Schwarm autonomer Roboter wurde mit RL trainiert, um ein gemeinsames Ziel zu erreichen (Sartoretti et al., 2019). Beim RL werden robotische Aufgaben durch direktes Lernen der Wahrnehmungs-Handlungs-Kopplung durchgängig gelöst (Levine et al., 2016). Um taktile Sensoren zur Robotersteuerung nutzen zu können, müssen Sensordaten an einen Feedback-Controller übergeben werden (Yamaguchi and Atkeson, 2017). Diese erlauben dem Roboter wiederum, eine Reihe von taktilen Fähigkeiten für das Fügen von Bauteilen einzusetzen (Belousov et al., 2019). Die Nutzung taktiler Informationen für robotische Aufgaben ist ein aktives Forschungsgebiet (Kappassov, 2015 und Yousef, 2011). Da die Informationen von taktilen Sensoren hochdimensional und schwer zu interpretieren sind, werden häufig Lernverfahren eingesetzt, um Steuerungen zu entwickeln, die solche reichhaltigen Feedbacksignale nutzen können (Calandra et al., 2018). Yuan et al. trainieren ein neuronales Netzwerkmodell, um die Stabilität eines geplanten Griffs vorherzusagen, der mit dem taktilen Sensor GelSight (Yuan et al., 2017) erfasst wurde. Van Hoof et al. verwenden RL, um ein Objekt mit einem Roboterarm basierend auf taktilen Feedback von einem BioTacTastsensor (Fishel et al., 2012) zu stabilisieren (Van Hoof et al., 2016). Tian et al. verwenden RL, um kleine Objekte wie Murmeln und Würfel mit Roboterfingern, die mit einem modifizierten GelSight-Sensor ausgestattet sind, zu bewegen und zu drehen (Tian et al., 2019).

2.2 Forschungslücke/Entwicklungsbedarf

Heutige Bauteile sind nicht für ihre Wiederverwendung im Sinne der Kreislaufwirtschaft und für das automatisierte Bauen entworfen. Vorgefertigte, trocken gefügte, repetitive, rekonfigurierbare Baugruppen leisten einen wichtigen Beitrag zu einer produktiveren und nachhaltigen Bauwirtschaft. Für einen höheren Grad an Automatisierung und Produktivität ist der Einsatz von Robotern für das Fügen unabdingbar. Aktuell werden in der Architektur vor allem Bauteile gestapelt oder verklebt. Verzahnende Bauteile werden noch nicht automatisiert gefügt. Darüber hinaus werden robotische Prozesse im Bauen (Greifen, Platzieren, Fräsen, 3D Drucken, Laminiert etc.) vorprogrammiert und durch Sensorik korrigiert oder an den Kontext angepasst. Roboter, die ohne explizite Pfad- und Bewegungsplanung und nur auf Basis eines 3D-Modells der Konstruktionen autonom bauen, existieren noch nicht. Die Analyseverfahren zur algorithmischen Bilderkennung (Computer Vision) haben durch Themen wie autonomes Fahren und Gesichtserkennung in den letzten Jahren eine signifikante Weiterentwicklung erfahren. Die Nutzung von Computer Vision in Kombination mit visuotaktilen Sensoren findet im Bauen keine Anwendung, obwohl das Erkennen und "Begreifen" unterschiedlicher Materialien, Dimensionen und Gewichten für die oben beschriebenen Fügeprozesse von großem Nutzen sein kann.

3. Zielstellung

3.1 Projektziele

Projektziel war von Beginn an der Erkenntnisgewinn zu Prinzipien autonomer Fügeprozesse modularer, trocken gefügter Konstruktion mit Hilfe des maschinellen Lernens und visuotaktile Sensoren. Dafür mussten eine Reihe von Teilzielen erreicht werden:

- **Ziel Elemente:** Es wurden für die robotische Fügung optimierte Elemente entwickelt und getestet. Außerdem wurden algorithmische Verfahren zum teil- automatisierten Entwerfen elementierter Konstruktionen entwickelt, die mit einer hohen Anzahl an Einzelementen effizient umgehen können.
- **Ziel Aggregationen:** Es wurden eine Vielzahl von Aggregationsformen entwickelt, so dass aus den Elementen neue Bauteile und Konstruktionen entstehen. Aggregationen entstehen Top-down (eine Großform wird in kleine Elemente zerlegt) oder Bottom-up (aus den gefügten Elementen erwächst eine Großform).
- **Ziel Sensor:** Der Einsatz des visuell-taktilen Sensors für die Fügung von Elementen wurde konzipiert. Die Kalibrierung des Sensors und die Implementierung der Signalverarbeitungsalgorithmen, sowie das Erlernen eines Klassifikators für unterschiedliche Elemente und Oberflächen wurden erarbeitet.
- **Ziel Pfad- und Bewegungsplanung:** Die grundlegenden Bewegungspfade (Aufbauprimitive) wurden mittels KI in Simulationen und durch manuelles Vorführen (learning from demonstration) erlernt.
- **Ziel Demonstrator:** An verschiedenen Demonstratoren wurden die oben beschriebenen Teilziele zu einer neu entwickelten digitalen Prozesskette zusammengefasst, um Konstruktionen herzustellen und diese zu evaluieren. Die Demonstratoren zeigen verschiedene Konstruktionstypologien (Last abtragen, Spannen, Kragen). Um die Integration unterschiedlicher Eigenschaften und Funktionen zu veranschaulichen, wurde der Demonstrator aus unterschiedlichen Materialien (Beton, Holz, Gips, Schaum, Mycelium, Kunststoffe etc.) modular aufgebaut. Die reversiblen Verbindungen der Module erlauben den einfachen Rück- und Umbau (Kreislaufwirtschaft). Die trockenen Fügungen sind kraftschlüssig verbunden und gehen damit über das einfache Stapeln von Bauteilen hinaus. Die Konstruktion wurde in höherer Präzision als mit den üblichen Bauteiltoleranzen gebaut. Zur Messung kamen 3D-Scans und Traglasttests zum Einsatz.

3.2 Fragestellung

- Wie müssen Bauteile für das autonome, reversible Fügen durch Roboter gestaltet sein?
- Sind autonom agierende Roboter, bestückt mit visuotaktile Sensoren, in der Lage, eigenständig Füge-, Montage- und Umbauprozesse auszuführen?
- Kann maschinelles Lernen Roboter in die Lage versetzen, ihre Bewegungsabläufe (Pfadplanung) eigenständig, kontextsensitiv und auf der Basis von Sensorfeedback zu ermitteln?
- Können Roboter Gewicht, Dimension und Oberflächeneigenschaften von Bauteilen mit Hilfe visuotaktile Sensoren ermitteln und selber "entscheiden", welche Greifposition für den Fügeprozess am besten funktioniert?
- Können architektonisch/konstruktive Kriterien als Belohnungen bei verstärkendem Lernen eingesetzt werden? Können diese Prinzipien verallgemeinert und auf viele Anwendungsfälle übertragen werden?

3.3 Methodischer Ansatz

- **Interdisziplinär:** Die interdisziplinäre Zusammenarbeit von DDU und IAS besteht seit 2018 und hat ein gemeinsames Vokabular und eine Robot Operating System (ROS) -basierte Softwareplattform für gemeinschaftliches Arbeiten hervorgebracht. Neben der jeweiligen disziplinären Expertise sind die Nutzung von 3D-Simulations- und Modellierumgebungen und der experimentelle Einsatz von Robotern überlappende und verbindende Kompetenzen. Die gemeinsamen Herausforderungen liegen in der Skalierung der robotischen Prozesse hinsichtlich Elementdimensionen und einer hohen Anzahl an Fügeprozessen mit sehr vielen Elementen.
- **Kombinatorisch:** Dieser digitale Material-Ansatz soll für Konstruktionen genutzt werden. Die Konstruktionen werden nicht 3D gedruckt, sondern reversibel und autonom 3D assembliert. WASP, ein Grasshopper Plugin für das Entwerfen mit vielen modularen Elementen, entwickelt bei DDU, dient dabei als algorithmisches Werkzeug zum Entwurf solcher Konstruktionen.
- **Maschinelles Lernen:** IAS nutzt hierarchisches Lernen und strukturiertes Lernen von Robotersteuerungsstrategien, informationstheoretische Methoden zur Strategie, Imitationslernen und autonome Exploration, lernende Vorwärtsmodelle für Langzeitvorhersagen, autonome kooperative Systeme und biologische Aspekte autonomer Lernsysteme für den autonomen Bau modularer Konstruktionen durch Roboter.
- **Visuotaktile Sensorik:** Bildbasierte visuotaktile Sensoren erlauben neue Methoden zur Erkennung von Geometrie, Material, Gewicht und Masseschwerpunkt von Objekten, die Roboter greifen. Transparente Elastomerkissen mit aufgedrucktem Punktraster an den Greiferbacken eines Roboters verformen sich beim Anpressen an das zu greifende Objekt. Kamerabilder ermöglichen die algorithmische Rekonstruktion der Geometrie des Objekts. Die aufgezeichnete Verformung des Punktrasters kann zur Berechnung von Kontaktkräften und Drehmomenten genutzt werden. Die weit entwickelten Methoden der automatisierten Bilderkennung kommen hier in einem neuen Kontext zum Einsatz.

3.4 Projektorganisation und Kooperationspartner

Das Projekt wird von den Fachgebietsleitern Prof. Jan Peters PhD (IAS) und Prof. Dr.-Ing Oliver Tessmann (DDU) geleitet. Für das Projekt wurde bei DDU und IAS jeweils eine neue Stelle für wissenschaftliche Mitarbeit geschaffen.

Im Fachgebiet DDU wird schwerpunktmäßig das Thema digitale Modularität und modulare Konstruktionen bearbeitet. DDU hat umfangreiche Vorarbeiten zu parametrischen und kombinatorischen Modulsystemen (z. B. trocken gefügte, topologisch verzahnte Konstruktionen) und ihrer robotergestützten Fügung sowie dazugehörige Softwareentwicklung (WASP für Grasshopper) geleistet. WASP ermöglicht das regelbasierte Konstruieren mit Modulen und implementiert das Konzept des Digitalen Materials in ein Entwurfswerkzeug für Architektur. Elemente werden mit Informationen zu Geometrie, Fügeprinzipien und -regeln und Orientierung versehen. WASP ermöglicht das Fügen sehr vieler Module und das Erforschen ihrer Kombinationsmöglichkeiten mittels topologischer Graphen. Unterschiedliche Modultypen können zu vorab definierten Zielformen gefügt werden.

IAS entwickelt die autonom agierenden Roboter, die Algorithmen zum Sensorfeedback und die Robotersteuerung, die durch maschinelles Lernen kontextsensitiv und auf Basis von komplexen (high-level) Instruktionen verschiedenste Aufgaben erfüllen. Roboter lernen durch Interaktion mit der Umwelt, Menschen oder anderen Robotern. Sie erkunden autonom ihre Umgebung und bestimmen, welche Merkmale für die Entscheidungsfindung berücksichtigt werden. IAS erforscht hierarchisches Lernen und strukturiertes Lernen von Robotersteuerungsstrategien, informationstheoretische Methoden zur Strategiesuche, Imitationslernen und autonome

Exploration, lernende Vorwärtsmodelle für Langzeitvorhersagen, autonome kooperative Systeme und biologische Aspekte autonomer Lernsysteme. IAS hat eine zusätzliche Forschungsgruppe am Max-Planck-Institut.

Die wissenschaftliche Arbeit wird durch studentische Hilfskräfte unterstützt. Das Einbinden der Forschungsthemen in die Lehre hat sich in der Vergangenheit bewährt und kam auch in diesem Projekt zum Einsatz. Der Prototypenbau der Sensoren erfolgte gemeinschaftlich und durch die Nutzung von 3D-Druckern und dem Maschinenpark des Fachbereichs Architektur. Komplexere Fräsarbeiten wurden in anderen Werkstätten der TU Da erbracht. IAS und DDU nutzten ihre Roboterlabore für das Projekt.

3.5 Arbeitspakete und Meilensteine

Das Projekt ist in drei Phasen unterteilt.

■ Phase 1: Module und Sensorik.

AP1 (DDU): Entwicklung von Modulen mit unterschiedlichen Funktionen (Tragen, Feuchtigkeit regulieren, Licht lenken etc.) und Materialien, die von Robotern gefügt werden können. Die Elemente sind demontier- und rekonfigurierbar.

AP2 (IAS): Die visuoaktiven Sensoren werden für verschiedene Modularten weiterentwickelt und an den Architekturmaßstab angepasst. Die Module dienen als Trainingsdaten für ihre Kalibrierung. Die Signalverarbeitungsalgorithmen werden implementiert. Es wird ein erster Klassifikator gelernt, der Materialtypen und strukturelle Eigenschaften erkennen kann.

Meilenstein1 ist erreicht, wenn die exemplarische Entwicklung von trocken gefügten Modulen, die aus unterschiedlichen Materialien bestehen und von Robotern gefügt werden können, abgeschlossen ist. Außerdem ist der bestehende visuotaktile Sensor auf den Architekturmaßstab angepasst worden. Abbruchkriterium: Die Technologie des visuotaktilen Sensors lässt sich nicht an höhere Traglasten von Architekturbauteilen anpassen/weiterentwickeln.

■ Phase 2: Konstruktionsentwicklung und Fügesimulation

AP3 (DDU): Es werden Konstruktionen mit WASP algorithmisch entwickelt, bei denen Module mit identischen Fügedetails 3D assembliert werden. Lastabtrag, Raumbildung, Klimaregulierung, Schallabsorption, Reversibilität etc. bestimmen die Modul-/Materialdifferenzierung.

AP4 (IAS): Roboter mit visuotaktiven Sensoren lernen die verschiedenen Module durch "begreifen" zu erkennen. Nutzung von Sensordaten für die Greif- und Pfadplanung durch maschinelles Lernen. In 3D-Simulationen lernt der Roboter die grundlegenden Bewegungspfade zur Fügung (Aufbauprimitive).

Meilenstein 2: Algorithmisch generierte, modulare Konstruktionen sind als digitale 3D-Modelle vorhanden und in kleinmaßstäblichen Modellen getestet. IAS hat erste Aufbauprimitive für Fügeprozesse simuliert und erste sensorbasierte Greiftests durchgeführt. Eine Pipeline für das Erlernen von Strategien mit geschlossenem Regelkreis, die Sensordaten direkt als Eingabegrößen nutzen, ist erarbeitet worden, d. h. der Roboter lernt Materialien, Oberflächen und Gewichte zu unterscheiden und Greifpositionen zu optimieren.

Abbruchkriterien: Im Projekt wird WASP verwendet, um Konstruktionen mit vielen, komplex geformten Modulen zu generieren (500 - 10.000). Die Vielzahl an simulierten Objekten stellt eine Herausforderung für die Lernalgorithmen dar, denn die Simulation muss in jedem Lernzyklus mehrmals ausgeführt werden. In dem

Projekt werden Verfahren untersucht, die Simulation zu beschleunigen. Jedoch besteht eine Möglichkeit, dass Lernen und Bewegungsplanoptimierung mit so vielen Objekten nicht realisierbar ist.

■ **Phase 3: Autonomer Bau eines Demonstrators.**

AP5 (DDU): Generierung zunehmend abstrakter Instruktionen zum Bau eines Demonstrators aus dem CAD Modell. Traglasttests und 3D-Scans zur Evaluation der vom Roboter gebauten Strukturen.

AP6 (IAS): Fusion der Aufbauprimitive zu Bausequenzen. Vollständiges sequentielles Zusammenfügen zu einem Demonstrator aus verfügbaren Teilen, inkl. aller Montageschritte auf Basis eines CAD-Modells. Auswertung der kompletten Pipeline basierend auf den von DDU identifizierten Kriterien. IAS und DDU entwickeln gemeinsam die Transformation von Kriterien wie Bausequenz/-zustände, Fügepräzision, Modulerkennung etc. zu Belohnungen (rewards) für das verstärkende maschinelle Lernen.

Meilenstein3/Demonstrator: Eine modulare Demonstrator konstruktion, ist von autonom agierenden Robotern errichtet worden. Der Roboter hat eigenständig eine Aufgaben- und Bewegungsplanung erstellt. Visuo-taktile Sensordaten ermöglichen ein kontextsensitives Handeln und das eigenständige Ermitteln der besten Greifposition von Bauteilen, so dass eine kollisionsfreie und präzise Montage möglich ist.

4. Projektverlauf

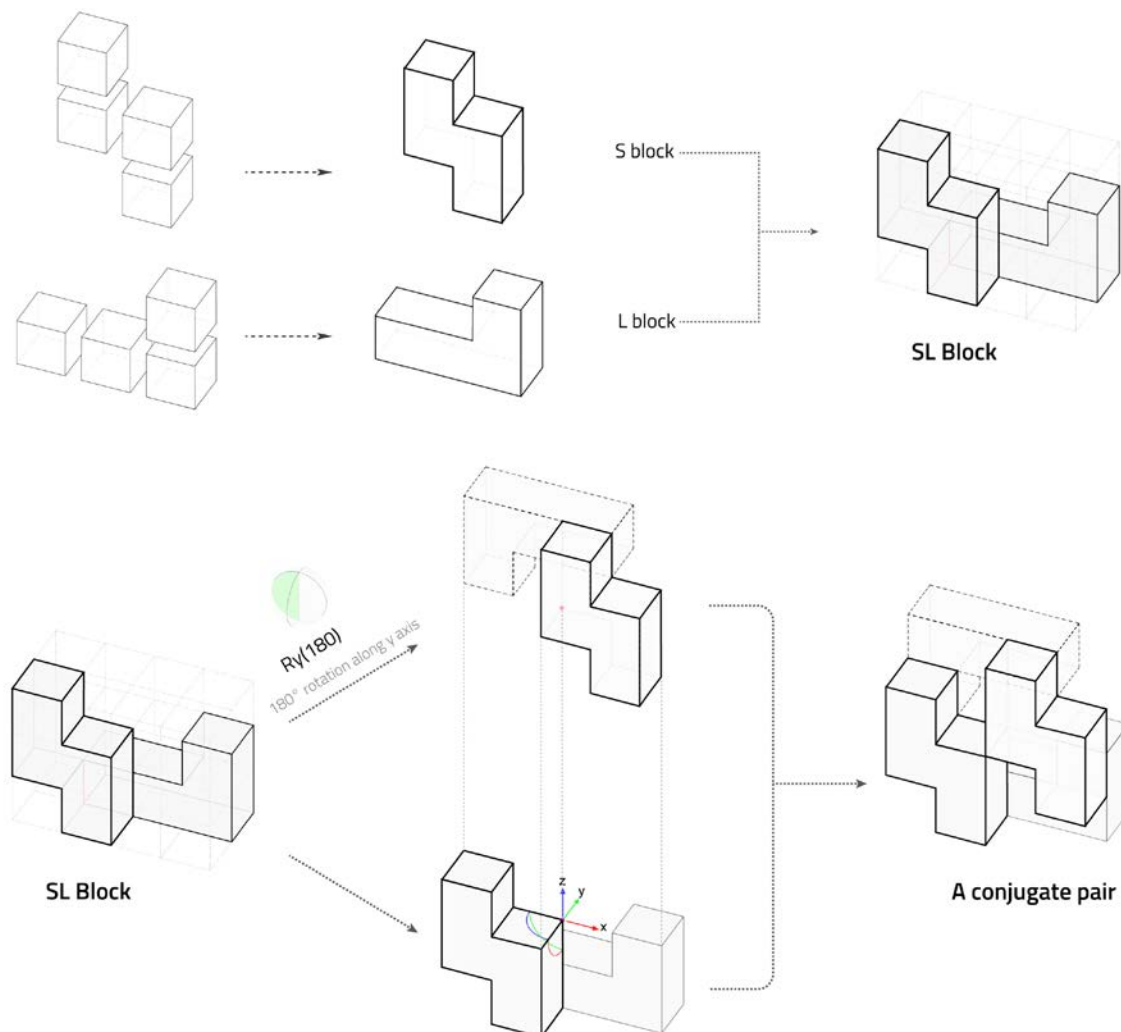
4.1 Das SL-Block System

Nach einer Analyse der Eigenschaften der verschiedenen Verzahnungssysteme wurde sich für das SL-Block-System entschieden. SL-Blöcke ermöglichen die trockene Fügung von seriellen und sich wiederholenden Elementen aus verschiedenen Materialien.

Durch Änderung der Verbindungsrichtungen und -konfigurationen können verschiedene Großformen geschaffen werden. Jede Verzahnungsvariante blockiert fünf Freiheitsgrade, so dass die meisten Bauzustände bei hoher Fertigungspräzision der Elemente, ohne temporäre Abstützung auskommen.

Abbildung 3

Geometrische Systematik des SL-Block Systems. Links: Vier Kuben bilden ein "L", vier Kuben bilden ein "S". Mitte: Zusammen entsteht ein SL-Block aus acht Kuben. Rechts: Zwei verzahnte SL-Blöcke bilden ein konjugiertes Paar.



Quelle: DDU, TU Darmstadt 2021

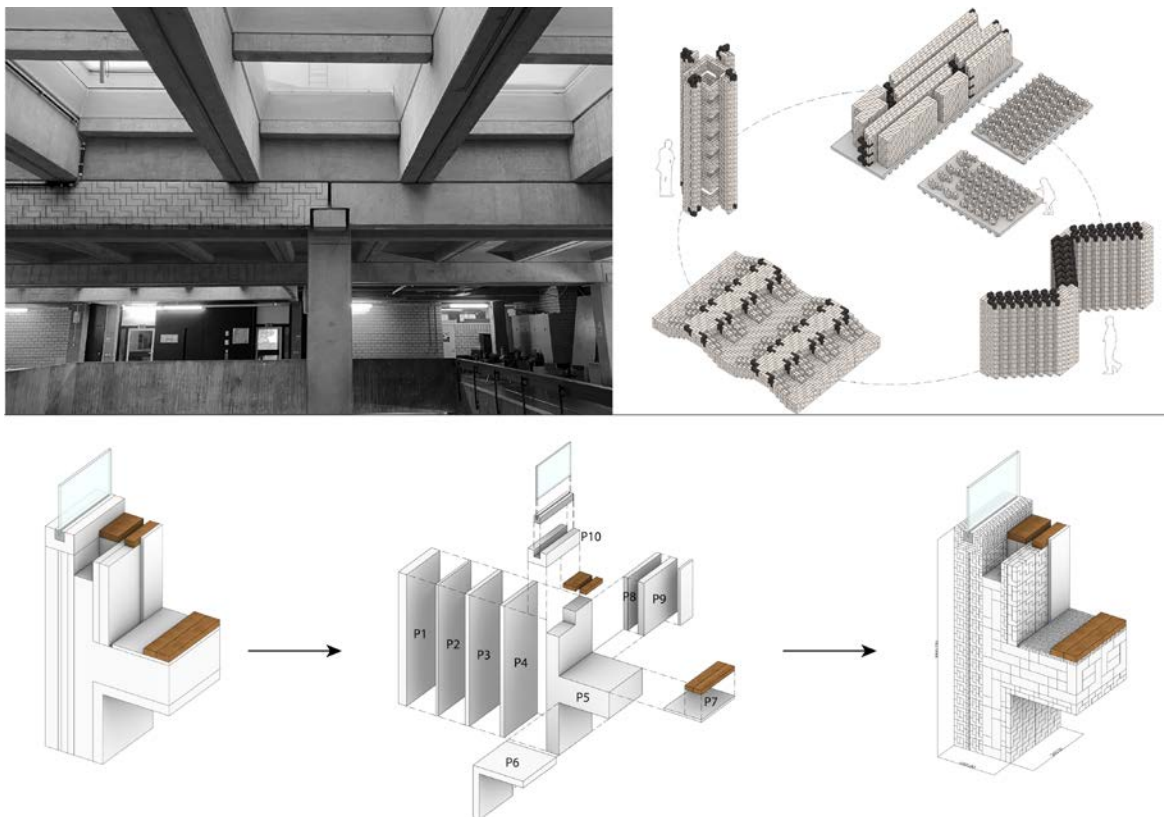
4.1.1 Geometrisches Element: Der SL-Block

Der SL-Block ist von dem taiwanesischen Wissenschaftler Shen-Guan Shih von der National Taiwan University of Science and Technology Taipei entwickelt worden (Shih, 2016). Er besteht aus einem S-förmigen und einem L-förmigen Element mit jeweils vier Kuben, die entlang ihrer Seiten miteinander verbunden sind (vgl. Abbildung 3).

Durch diese komplexe Form können zwei SL-Blöcke auf sechs unterschiedliche Arten miteinander reversibel verzahnt werden. Je nach Fügevariante entstehen unterschiedliche, reversibel gefügte Großformen. Bauelemente, die bisher als massiv und monolithisch galten, können nun durch Zusammenfügen vieler Blöcke gebildet werden (vgl. Abbildung 4). Diese kleinteiligen Elemente können zusammengebaut, zerlegt und in verschiedenen Konfigurationen wieder zusammengesetzt werden, was zu einer dauerhaften Umwandlung in eine zirkuläre Wiederverwendung der Gebäudeelemente beiträgt.

Abbildung 4

Oben links: eine Innenaufnahme aus dem Fachbereich Architektur der TU Darmstadt, die den Vergleich zwischen dem traditionellen vorgefertigten Betonbalken und einem Entwurf zur Diskretisierung eines Trägers in SL-Blöcke zeigt. Bauteile, die bisher als massiv und monolithisch betrachtet wurden, können nun durch viele kleine SL-Blöcke zusammengefügt werden. Diese kleinen Blöcke können zu verschiedenen Gebäudekonfigurationen zusammengesetzt, demontiert und wieder zusammengebaut werden, was eine temporäre oder dauerhafte Nutzung von Bauelementen ermöglicht. Oben rechts ist ein Diagramm zu sehen, das die reversible Konstruktion der SL-Blöcke veranschaulicht. Große Mengen von SL-Blöcken dienen als Materiallager, das für die Konstruktion von Wänden, Stützen und Decken verwendet werden kann. Daraus entsteht ein Kreislauf der Wiederverwendung. Unten: Entwurf zur Umgestaltung eines konventionellen Wandaufbaus in ein System aus SL-Blöcken. SL-Blöcke aus unterschiedlichen Materialien übernehmen die unterschiedlichen Funktionen einer Wand. Ihre gemeinsame Geometrie erlaubt jedoch stets die Verzahnung.



Quelle: DDU, TU Darmstadt 2021

4.1.2 Kombinatorische Prinzipien von SL-Blocks und SL-Strands

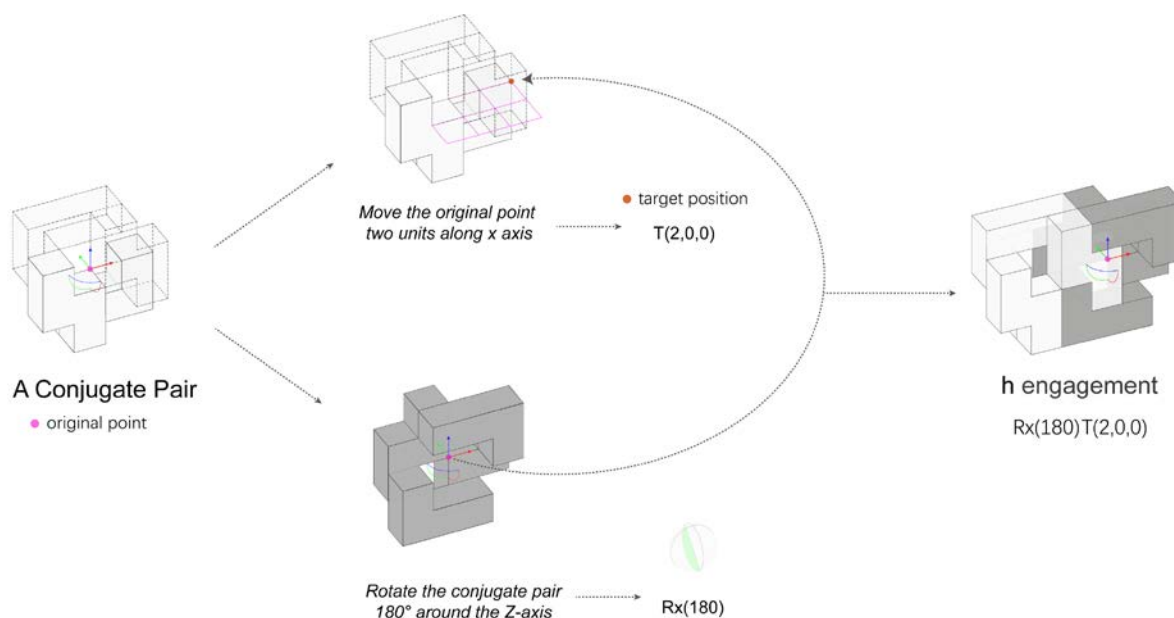
Der SL-Block ist ein Polykubus, der an 3D Puzzle oder Tetrisfiguren erinnert. Die Füge- und Verzahnungsmöglichkeiten sind nicht offensichtlich. Menschen, die zwei Blöcke in die Hand nehmen, erforschen diese intuitiv und spielerisch. Die inhärente kombinatorische Komplexität, die die Verzahnung von mehr als zwei Blöcken mit sich bringt, kann deutlich effektiver durch digitale Modelle erforscht werden, die auf Basis der kombinatorischen Regeln großen Mengen an SL-Blöcken aggregieren. Im Jahr 2016 schlug Shih sechs Verbindungsregeln vor, denen er jeweils einen Buchstaben zuordnet. Aus der Verbindung von zwei SL-Blöcken entsteht ein "SL Engagement" (Shih, 2016). So kann man mit Zeichenketten die Aggregationen der SL-Blöcke zu größeren Formen beschreiben. Diese Prinzipien bilden die Basis für die Untersuchung des SL-Block Systems und die Erforschung seiner Anwendung als modulare Konstruktion mit Trockenfugen.

Verbindungsregeln für Kompositionen von SL-Blocks

Die Fügeregeln für die Verbindung von SL-Blöcken beschreiben geometrische Transformationen und das zu transformierende Originalobjekt. Das Objekt, ein sogenanntes konjugiertes Paar, entsteht durch die Verbindung zweier SL-Blöcke. Die für die Verbindung zweier Blöcke notwendige Transformationsaktion (Drehung und Rotation) wird als Buchstabe dargestellt. Das neue konjugierte Paar, das sich aus der Transformation ergibt, wird als "Engagement" bezeichnet (vgl. Abbildung 5).

Abbildung 5

Die Kombination der SL-Blöcke basiert auf der Rotation und Translation eines konjugierten Paares. Im Rahmen der Definition von h engagement dreht sich das konjugierte Paar zunächst um 180 Grad um die Z-Achsen. Nach der Drehung wird es um zwei Einheiten entlang der x-Achse verschoben. Die sich aus der Drehung und Verschiebung ergebende Geometrie kann mit dem anfänglichen konjugierten Paar ineinandergreifen und eine Kombination bilden. Die Drehung und die Verschiebung können zu einer Transformationsmatrix zusammengefasst werden, die mit dem Buchstaben "h" bezeichnet wird.

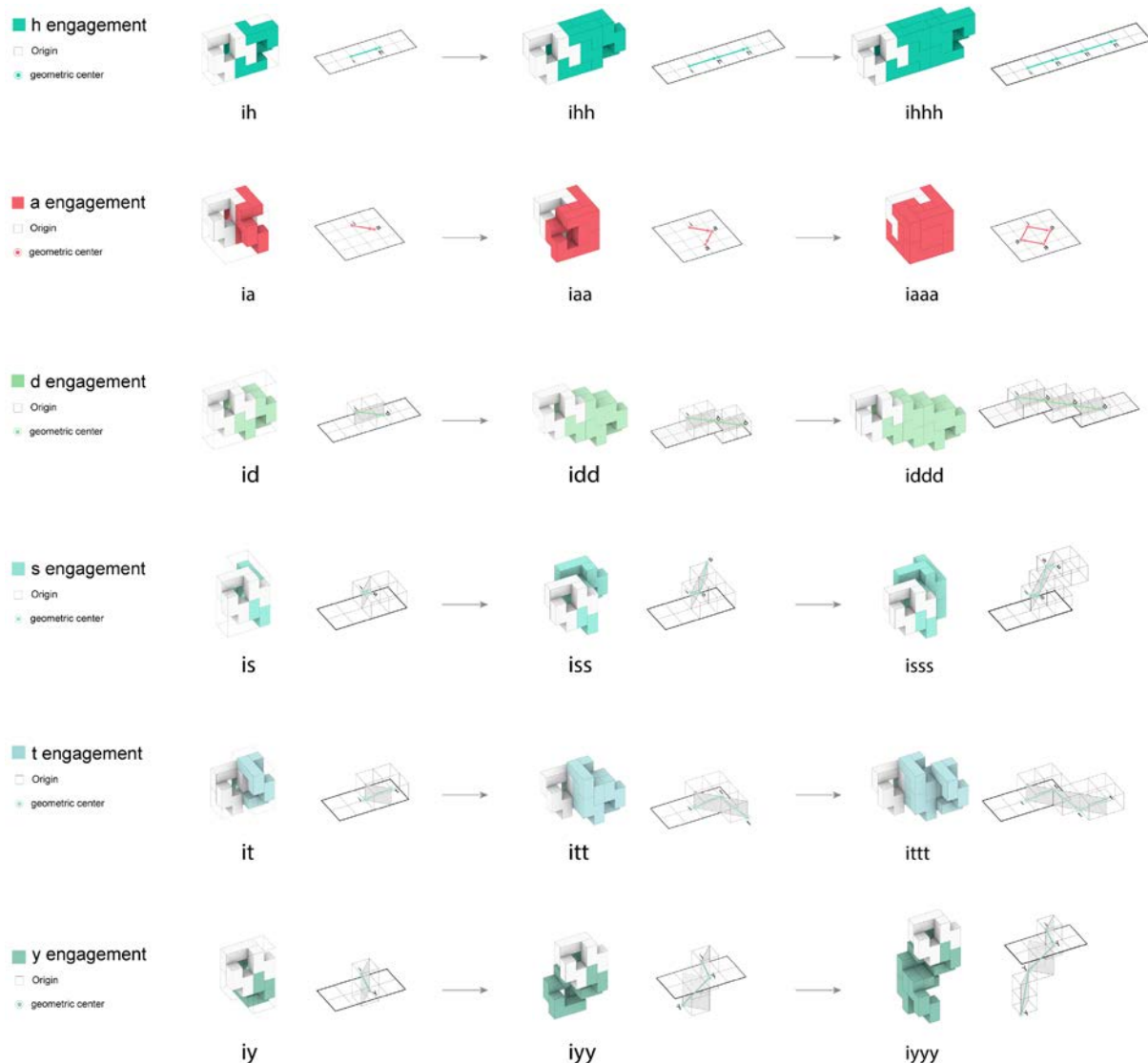


Quelle: DDU, TU Darmstadt 2021

Insgesamt gibt es sechs Arten von Verbindungsregeln, die von Shih definiert wurden. Die SL-Engagements können in längeren Ketten verbunden werden. Daraus entstehen SL-Stränge (SL- Strands). Mit Hilfe dieser Definitionen kann der komplizierte Prozess der Verbindung von SL-Blöcken als eine Abfolge von Buchstaben beschrieben werden, um verschiedene SL-Engagements zu einem SL-Strang zu kombinieren (vgl. Abbildung 6). Die generative und regelbasierte Methode für die sequentielle Kombination von SL-Blöcken zu SL Engagements und zu SL-Strands wurde im Projekt algorithmisch beschrieben und in Rhino/Grasshopper implementiert.

Abbildung 6

SL-Blöcke und die sechs verschiedenen Möglichkeiten der Verzahnung (Engagements) und die daraus entstehenden Formen: h engagement: lineare Anordnung, a engagement geschlossener Kubus aus 8 SL-Blöcken, d engagement: getrepte lineare Anordnung, s engagement: überlappend getrepte Anordnung, t engagement: diagonal, linear, stapelbar, y engagement: vertikal, spiralisierend.



Quelle: DDU, TU Darmstadt 2021

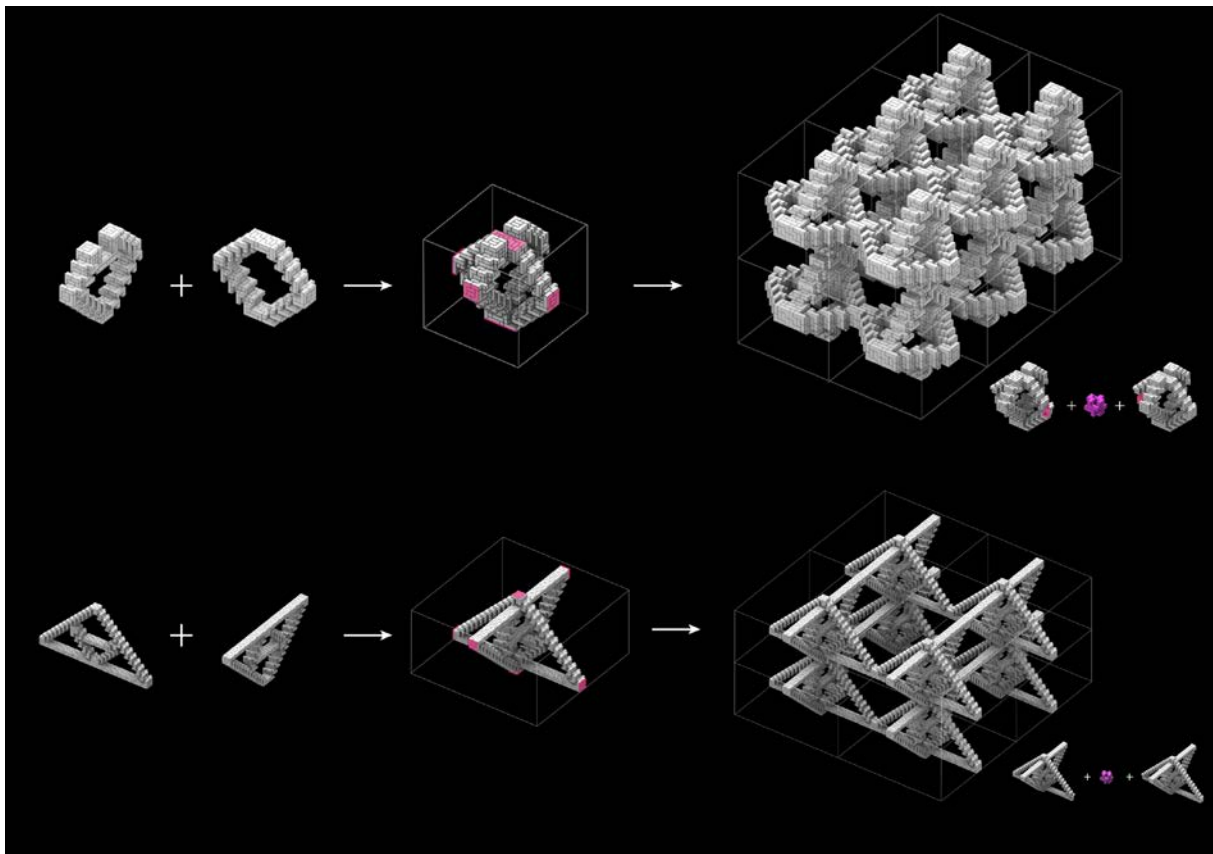
Verbindungsmethoden für Aggregationen von SL-Strands

SL-Stränge können zu größeren geometrischen Systemen und letztlich konstruktiven Elementen verbunden werden (vgl. Abbildung 7). Beispielsweise entstehen aus SL Block Aggregationen über die verschiedenen Fühierarchien Kuben, Quader und Diagonalstreben, die dann mit anderen, konventionellen Bauteilen verbunden werden können.

Bilden viele SL-Blöcke einen Quader, kann dieser als massiver Träger oder Stütze betrachtet werden. Die einzelnen SL-Blöcke und die Tatsache, dass das Objekt aus vielen kleinen Objekten besteht, spielt auf dieser konstruktiven Ebene keine Rolle mehr. So können SL-Block Aggregationen in bestehende Konstruktionen integriert werden. Wir haben ihre Verbindungen in drei Typen unterteilt: Verzahnte Verbindungen, die an traditionelle

Abbildung 7

Beispiele komplexer, algorithmisch generierter SL-Strukturen. Oben: Zwei ringförmige SL-Strands bilden zusammengesteckt einen Oktaeder, der in einem Raumraster repliziert werden kann. Unten: Lineare SL-Strands bilden fachwerkartige Strukturen, die räumlich miteinander gefügt werden können und eine Art räumliches Stabwerk aus SL-Blöcken bilden.



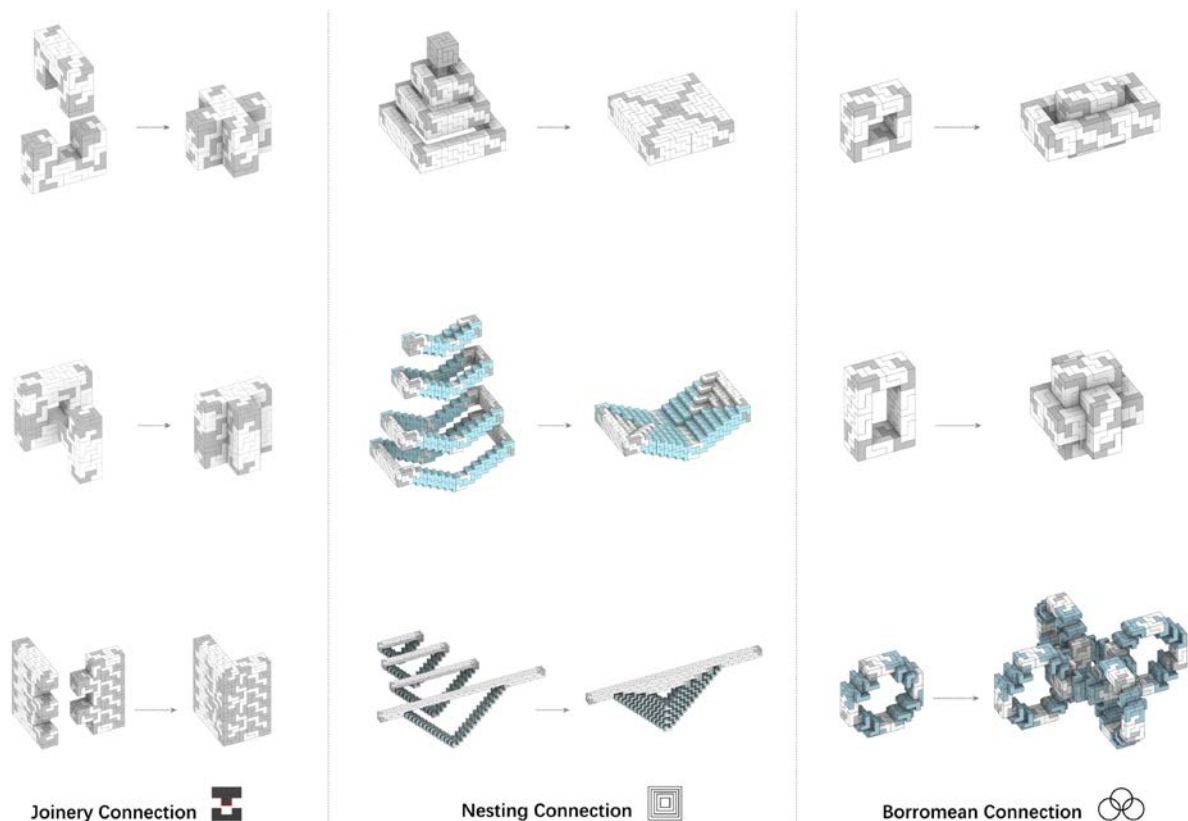
Quelle: DDU, TU Darmstadt 2021

Konstruktionen anknüpfen (Schwalbenschwanz, Zapfen, Überblattung), rekursiv-verschachtelte Verbindungen, bei dem ein Satz von SL-Strängen so angeordnet ist, dass jeder Strang von einem anderen umschlossen wird. Dieser Mechanismus erinnert an russische Matrjoschka-Puppen. Borromäische Verbindungen bilden miteinander verknotete Schleifen aus SL-Strängen. Dabei wird der Leerraum eines Schleifenstrangs durch eine andere Schleife gefüllt. Diese mathematischen Konzepte haben wir im Folgenden auf ihre Anwendbarkeit als Konstruktionselemente untersucht (vgl. Abbildung 8).

Durch die Kombination von SL-Blöcken wird ein hierarchisches und elementiertes System aufgebaut, bei dem ein einzelnes Teil zahlreich vervielfältigt werden kann, um größere Komponenten zu bilden, die sich umeinander wickeln und miteinander verflechten, um komplexe Aggregationen höherer Ordnung (Bauteile) zu bilden. Diese Aggregationen können auseinander- und wieder zusammengebaut werden.

Abbildung 8

SL-Strands, die zu größeren Aggregationen miteinander verbunden werden. Drei Arten der Verbindung von SL-Strands: Links: Verzahnte Verbindungen, Mitte: Rekursiv-verschachtelte Verbindungen, Rechts: Borromäische Verbindungen zum Einsatz.



Quelle: DDU, TU Darmstadt 2021

4.1.3 Computergestützte Generierung von SL-Strands

Die reversible Fügung von sich wiederholenden Komponenten birgt viele Vorteile, aber erfordert im Entwurfsprozess das Arbeiten mit hoher kombinatorischer Komplexität. Jede Entscheidung über SL-Block Fügung hat schwer absehbare Konsequenzen auf die nächsten Fügeschritte, so dass reines 3D-Modellieren oder das Arbeiten am physischen Modell zeitaufwendig und häufig nicht zielführend ist. Wie in einem Graphen, hat eine frühe falsche Verzweigung nachhaltige Auswirkungen auf das „Wachstum“ der Struktur. Produktiver ist es, ein System zu haben, das die gewünschten architektonischen Strukturen und Entwurfsziele als Eingangswerte benutzt, um die notwendigen Fügeprozesse der SL-Blöcke dafür zu berechnen. Für diesen Arbeitsschritt wurden im Projekt die Fügeregeln formalisiert und in Rhino/Grasshopper implementiert.

In dieser 3D-Modellierungssoftware kann das kombinatorische Potenzial von geometrischen Modulen erforscht werden. Außerdem kann das Trag- und Materialverhalten, die Verzahnung sowie Montageprozesse simuliert und bewertet werden. Die Transformationsmatrix für jedes SL-Engagement wurde in Rhino/Grasshopper erstellt. Ausgehend vom ursprünglichen Punkt des konjugierten Paares bei (0,0,0), können Rotation über

Abbildung 9

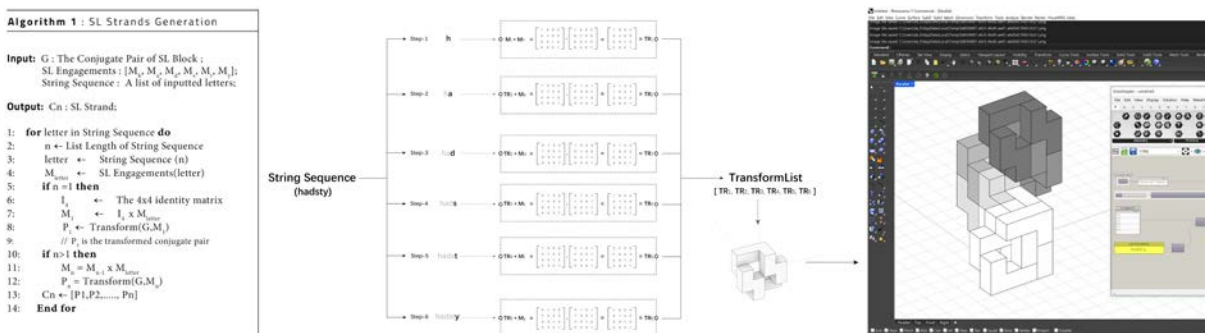
Die Tabelle zeigt die SL engagement Transformationsmatrix mit Werten für Rotationsradien und Translationsvektoren. Rechts sind konjugierte SL Block Paare und ihre in Grasshopper implementierten SL Engagement Regeln dargestellt.

Rotation	$R_z(0)$	$R_z(-90)$	$R_z(90)$
$R_z(180)$	$\begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ \downarrow $R_z(180)T_z(2,0,0)$ <h3>h</h3>	$\begin{bmatrix} 0 & -1 & 0 & 1 \\ -1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ \downarrow $R_z(180)R_z(-90)T_z(1,-1,1)$ <h3>t</h3>	$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ \downarrow $R_z(180)R_z(90)T_z(1,-1,1)$ <h3>s</h3>
$R_z(0)$	$\begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ \downarrow $T_z(2,0,-1)$ <h3>d</h3>	$\begin{bmatrix} 0 & 1 & 0 & 1 \\ -1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ \downarrow $R_z(-90)T_z(1,-1,0)$ <h3>a</h3>	$\begin{bmatrix} 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ \downarrow $R_z(90)T_z(1,1,2)$ <h3>y</h3>

Quelle: DDU, TU Darmstadt 2021

Abbildung 10

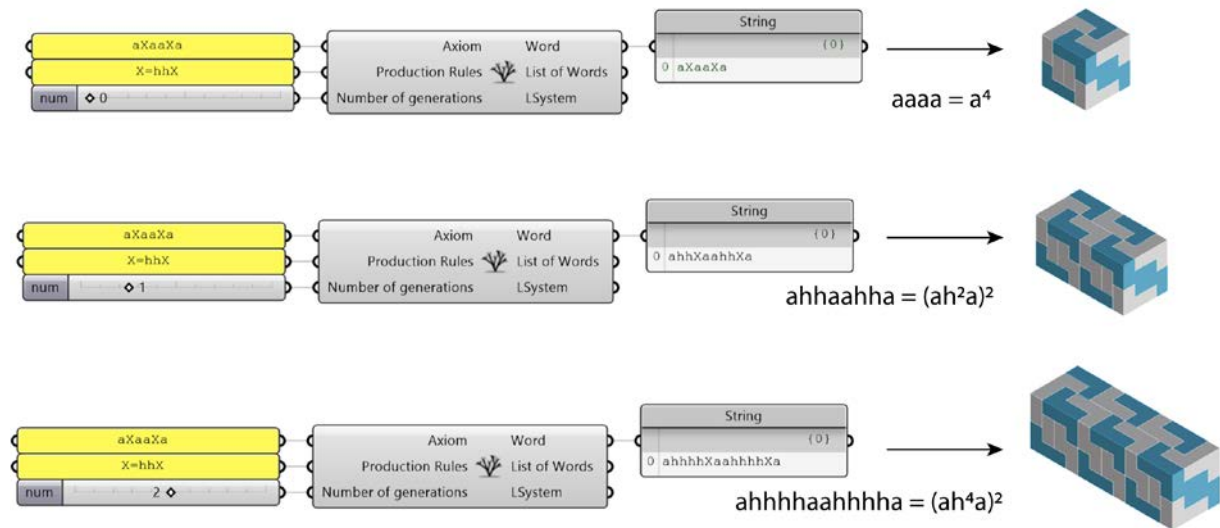
SL Strand Erzeugung dargestellt als Python Code in Grasshopper (links), Zuweisung von Zeichen zu Transformationsmatrix (mitte) und Geometrische Repräsentation (rechts). Sequentielle Anwendung der SL Engagements h, a, d, s, t, y Regeln und Grasshopper und die geometrische Repräsentation in Rhino.



Quelle: DDU, TU Darmstadt 2021

Abbildung 11

Grafische Programmieroberfläche von Grasshopper. Zeichenketten werden in einer eigens entwickelten Grasshopper Python Komponente zu SL-Block Aggregationen übersetzt.



Quelle: DDU, TU Darmstadt 2021

Radien und die Translation über Vektoren als definierte Bewegung beschrieben werden. Diese geometrischen Transformationen können in einer einzigen 4x4-Matrix dargestellt und einem SL-Engagement Buchstaben zugeordnet werden. Eine Fügesequenz aus einer Reihe von SL-Engagements kann dann über eine Zeichenkette beschrieben werden, bei der jeder Buchstabe eine zugewiesene Transformationsmatrix aufruft und durchführt (vgl. Abbildung 9 und 10).

Grammatische Regeln zur Generierung SL-Strands

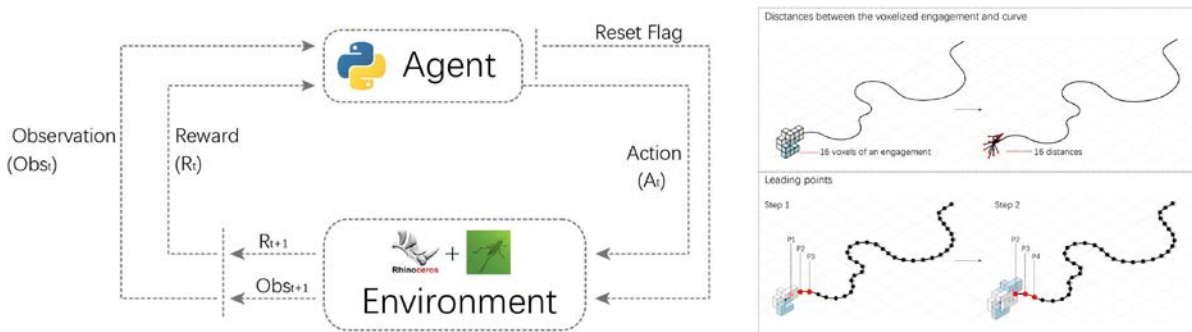
Eine Zeichenkette kann auf ein konjugiertes SL-Block Paar angewendet werden, um eine Aggregation von SL-Engagements zu erzeugen, die gemeinsam einen neuen SL-Strand erzeugen. Die Kette "aaaa" beispielsweise erzeugt einen Würfel, den kleinsten geschlossenen SL-Strang. Inspiriert von dem Lindenmayer-System, das zur mathematischen Beschreibung von Fraktalen und Pflanzenwachstum genutzt (Prusinkiewicz & Hanan 2013), kommen hier Grammatikregeln zum Einsatz, die das rekursive Ersetzen von einzelnen Zeichen mittels Produktionsregeln in komplexere Zeichenketten steuern.

Die Zeichenkette "aaaa" bildet das Startsymbol, "aXaa" die Anfangsvariable (Axiom). Das eingefügte "X" wird bei jeder Iteration durch die Produktionsregel (X→hhX) ersetzt. Mit zunehmender Anzahl von Generationen wird die Zeichenkette schnell größer und komplexer, so dass aus dem kleinsten geschlossenen SL-Strang eine größere Struktur abgeleitet werden kann (vgl. Abbildung 11). Im Rahmen des Projekts wurde dieser generative Vorgang in eine Grasshopper Definition überführt, die es erlaubt, durch die einfache Veränderung von Zeichenketten die Aggregationen zu verändern.

Verstärkendes Lernen für die Aggregation von SL-Blöcken entlang von Referenzkurven

Abbildung 12

Links: Reinforcement Learning Schleife mit Rhino/Grasshopper. Rechts: Zwei Hauptbestandteile der Beobachtung (Observation): Entfernungen von jedem der 16 Voxel des aktuellen Engagements zur Kurve (oben) und Koordinaten von 3 vorauslaufenden Punkten (unten). Anhand dieser zusätzlichen Informationen kann der Agent vorhersehen, wie sich die Kurve in Zukunft entwickeln wird.



Quelle: DDU, TU Darmstadt 2021

In einer rückgekoppelten verstärkenden Lernschleife (vgl. Abbildung 12) führt der Agent (Python-Skript) eine Aktion aus (den nächsten Block setzen), und die Umgebung (Grasshopper) liefert eine Beobachtung (z. B. die Positionen der Teile im Verhältnis zur Kurve) und eine Belohnung (numerischer Wert, der angibt, wie gut die Kurve durch SL-Blöcke abgedeckt ist). Der Agent verbessert seine Aktionen auf der Grundlage von Beobachtungen und Belohnungen mit dem Ziel, die Gesamtsumme der zu erwartenden Belohnungen zu maximieren.

Die Aktion ist in unserem Fall eine Ganzzahl zwischen null und fünf, die angibt, welcher der sechs möglichen SL-Engagements als nächstes ausgeführt werden soll. Die Aktion bezieht sich auf den zuvor platzierten Block, zu dem die neue Verzahnung hergestellt werden soll. Daher muss der aktuelle Zustand der Anschlussstelle als auch die Positionen aller bereits platzierten Teile und der Zielkurve bekannt sein und repräsentiert werden.

Unsere Beobachtung besteht aus drei Teilen: i) einer Liste normalisierter Werte, die den Abstand von jedem der 16 Voxel des aktuellen SL-Engagements zur Kurve kodieren, ii) einer abgeflachten Transformationsmatrix, die die Position und Orientierung des aktuellen SL-Engagements beschreibt, und iii) den Koordinaten von drei Punkten auf der Kurve vor dem Agenten.

Abbildung 12 veranschaulicht, wie die Beobachtung aufgebaut ist. Es ist wichtig, einige lokale Informationen über die zukünftige Richtung der Kurve bereitzustellen; daher werden der Beobachtung drei vorausliegende Punkte hinzugefügt, um eine Blockplatzierung besser bewerten zu können. Gleichzeitig erhöht sich jedoch die Dimensionalität des Beobachtungsraums, was dazu führt, dass sich der Agent zu sehr an die aktuelle Kurve anpasst. Andererseits führt die Angabe von nur einem Punkt dazu, dass der Agent kurzsichtig ist und die Richtung der Kurve oder Hindernisse, die einige Schritte vor ihm liegen, nicht berücksichtigen kann.

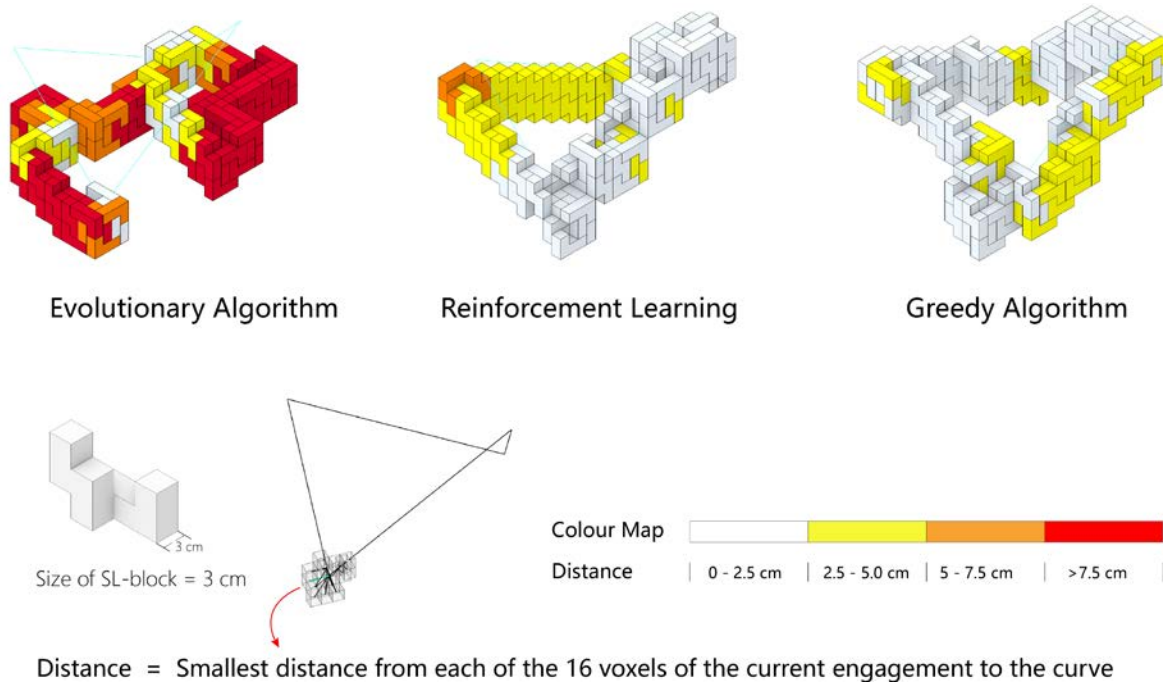
In den Experimenten wurde festgestellt, dass drei Punkte ausreichen, um die betrachteten Aufgaben zufriedenstellend zu lösen. In anderen Szenarien kann dieser Parameter angepasst werden, um die Leistung zu verbessern.

Die Belohnungsfunktion und das "Reset-Flag" werden verwendet, um den Agenten zu ermutigen, der gewünschten Kurve zu folgen. Insbesondere wird die Belohnung (definiert als eins minus dem kleinsten Abstand des aktuellen Blocks zur Kurve) maximiert, wenn die Struktur der Kurve perfekt folgt. Weicht der Agent zu sehr von der Kurve ab, wird die Umgebung auf den Ausgangszustand zurückgesetzt, wodurch die Anhäufung von Belohnungen gestoppt wird und der Agent gezwungen ist, von vorne zu beginnen.

Evaluation des Verstärkenden Lernens für SL -Block Aggregationen

Abbildung 13

Evaluation des Verstärkenden Lernens für Top Down Basisaggregationen eines geschlossenen SL-Strands entlang eines geschlossenen Polygons. Farbliche Darstellung der Distanzwerte zwischen SL-Blöcken und Kurve: Weiß: 0-2,5 cm, gelb 2,5-5 cm, orange: 5 - 7,5cm. Rot: größer 7,5 cm. Beim evolutionären Algorithmus sind ca. 50 der Blöcke rot 20% orange, 15% gelb, 15% weiß. Beim Reinforcement Learning sind ca. 50 der Blöcke weiß 45% gelb, 5% orange. Beim evolutionären Algorithmus sind ca. 50 der Blöcke rot, 20% orange, 15% gelb, 15% weiß. Beim greedy Algorithmus sind ca. 65 der Blöcke weiß, 35% gelb.



Quelle: DDU, TU Darmstadt 2021

Wir vergleichen zwei Algorithmen zum Verstärkenden Lernen, Proximal Policy Optimization (PPO) und Deep Q-Networks (DQN). PPO ist ein beliebter Algorithmus im Bereich des verstärkenden Lernens, der ein stabiles und robustes Training gewährleistet, dank der Begrenzung der Strategieaktualisierungen, die eine Balance zwischen Exploration und Exploitation aufrechterhält. DQN verwendet ein tiefes neuronales Netz, um die Q-Wert-Funktion zu approximieren, die den erwarteten Belohnungen für das Ausführen einer bestimmten Aktion in einem gegebenen Zustand entspricht. Ein Schlüsselmerkmal von DQN ist die Verwendung von Erfahrungswiederholung, bei der der Agent vergangene Erfahrungen speichert und zufällig aus ihnen auswählt, um die Korrelation zwischen aufeinanderfolgenden Lernaktualisierungen zu brechen. Dies, zusammen mit einem separaten Zielnetzwerk, hilft dabei, den Lernprozess zu stabilisieren.

Bei der Aufgabe der sequentiellen SL-Block-Fügung entlang einer gegebenen Kurve werden drei Arten von Kurven betrachtet: offene 2D-Kurven, offene 3D-Kurven und geschlossene 3D-Kurven. Für jeden Kurventyp werden zwei Kurvenlängen evaluiert: kurz (erfordert etwa 20 SL-Engagements) und lang (erfordert etwa 40 SL-Engagements). Die Auswertung dauert zwischen 10 und 30 Minuten, je nach Schwierigkeitsgrad der Kurve.

Während der genetische Algorithmus schneller konvergiert als das Verstärkende Lernen, nämlich in 5-10 Minuten, findet er bei 2D-Kurven eine recht gute Lösung, versagt aber bei offenen und geschlossenen 3D-Kurven. Der Grund für die Schwierigkeiten mit 3D-Kurven muss in der höheren Komplexität des Problems im 3D-Raum liegen (vgl. Abbildung 13), da viel mehr Möglichkeiten zur Platzierung von SL-Blöcken zur Verfügung stehen. Der Algorithmus kann eine solche exponentielle Zunahme an Kombinationen nicht bewältigen.

Im Vergleich zum evolutionären Algorithmus gelingt es, den Algorithmen des verstärkten Lernens, sequentiell ineinandergreifende SL-engagements auf Kurven unterschiedlicher Länge und Dimension zu finden. Beide vorgestellten Algorithmen, DQN und PPO, erhalten ähnliche Lösungen.

Die Qualität der generierten kombinatorischen Sequenzen verschlechtert sich leicht bei komplexen, geschlossenen 3D-Kurven. Dies könnte auf eine unzureichende Trainingszeit zurückzuführen sein, da die Trainingszeit auf 30 Minuten begrenzt wurde, um die Schnittstelle interaktiv zu halten.

Eine Erhöhung der Trainingsdauer kann die Ergebnisse bei diesen komplexen Problemfällen verbessern. Betrachtet wird nur die Standard-RL-Einstellung, bei der ein Agent für dieselbe Aufgabe bewertet wird, für die er trainiert wurde, d. h. wir trainieren einen RL-Agenten und bewerten ihn für dieselbe Kurve. Im Idealfall möchten wir, dass der Agent in der Lage ist, auf neue Kurven zu verallgemeinern, ohne dass ein vollständiges Neutrainning erforderlich ist. Die Algorithmen, die zu einer solchen Generalisierung fähig sind, werden derzeit in den Bereichen Multitasking und Transfer Learning sowie Meta-Reinforcement-Learning erforscht.

Abbildung 13 rechts zeigt die Ergebnisse des Greedy-Algorithmus. Dieser Algorithmus nutzt eine annähernd optimale Heuristik für die Wahl der nächsten Aktion, und daher können seine Ergebnisse als Ziel für evolutionäre und verstärkende Lernalgorithmen angesehen werden. Da die Heuristik bereitgestellt wird, läuft der Algorithmus sehr schnell und findet eine Lösung in weniger als zwei Minuten. In allen Fällen findet der Algorithmus kombinatorische Sequenzen, die die vorgegebenen Kurven ausreichend genau reproduzieren.

Adaption des Verstärkenden Lernens an SL-Block-Platzierung an Kurven mit zusätzlichen Hindernissen

Ein wichtiger Vorteil des Verstärkungslernens im Vergleich zur "greedy" Optimierung mit Heuristiken ist, dass RL-Algorithmen die Wertfunktion selbständig lernen. Im Gegensatz dazu muss bei der "greedy" Optimierung ein Mensch die Heuristik festlegen.

Obwohl es einfach ist, eine zielführende Heuristik für die einfache Aufgabe der Kurvenverfolgung bereitzustellen, sind wir letztendlich an abstrakteren Aufgaben interessiert, wie z. B. der Optimierung einer Struktur auf der Grundlage allgemeiner Gestaltungskriterien, und in diesem Fall ist es teilweise nicht möglich, eine robuste Heuristik zu finden.

Abbildung 14 zeigt eine Modifikation der Kurvenverfolgung durch Einführung von Hindernissen die das Platzieren von SL-Blöcken verhindern. Wie dargestellt, ist der "greedy" Algorithmus nicht in der Lage, sich an die veränderte Aufgabe anzupassen, während RL eine Strategie findet, mit der das Problem gelöst werden kann. In diesem Szenario führt das Befolgen der starren und vordefinierten Heuristik zu einer suboptimalen Lösung. Um das vom "greedy" Algorithmus erzielte Ergebnis zu verbessern, müsste der Benutzer die Heuristik manuell an das neue Szenario anpassen. Im Gegensatz dazu sind beim Verstärkenden Lernen keine weiteren Anpassungen erforderlich, und es kann genau derselbe Algorithmus auf das geänderte Problem angewendet werden.

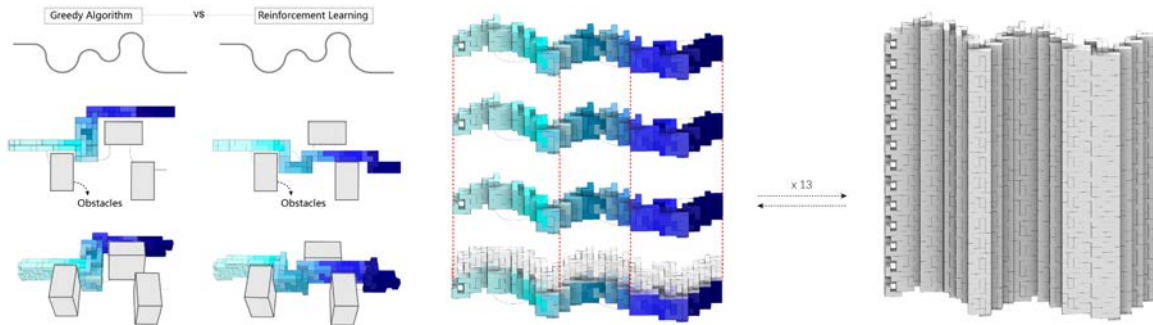
Von der Aggregationen zur Konstruktion

Der Algorithmus zur Erstellung von SL-Block-Konfigurationen entlang vorgegebener Kurven kann als Werkzeug im Entwurfsprozess eingesetzt werden. Abbildung 15 zeigt das Stapeln von Kurven, um eine Wand zu erzeugen. Geschlossene räumliche Kurven bilden die Basis für freitragende oder auskragende Strukturen (vgl. Abbildung 15). Mit physischen Modellen überprüfen wir die Verzahnung und die Tragwirkung der Aggregation. Verstärkungslernen kann also die Montagereihenfolgen und SL-Strands finden, die sich an einfach beschriebene geometrische Vorgaben anpassen.

Mit dieser Methode lassen sich architektonische Fragestellungen in ein RL-Problem übersetzen. Das mit standardmäßigen Open-Source-Python-Paketen bekannter RL-Algorithmen gelöst werden kann.

Abbildung 14

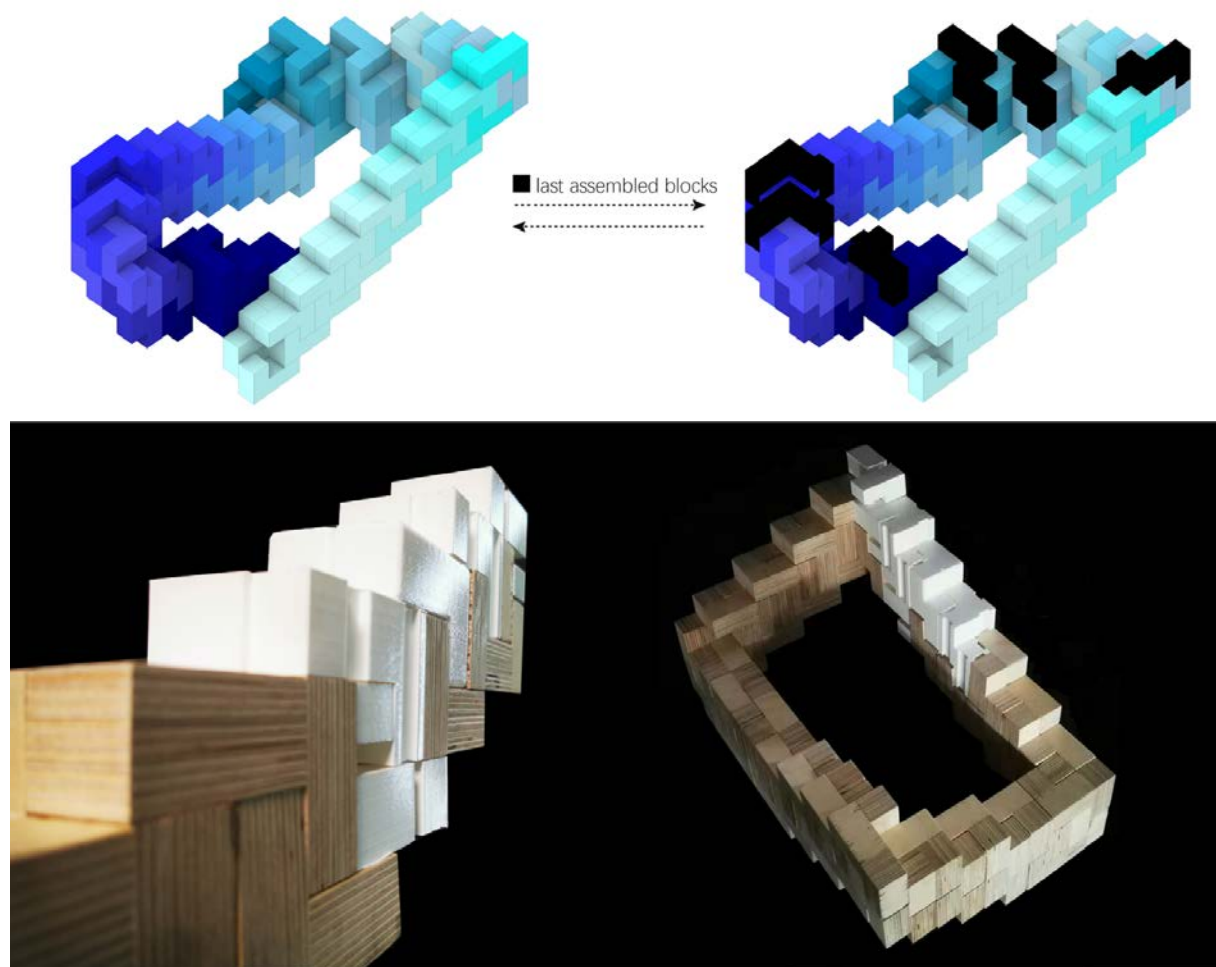
Versuchsaufbau mit Referenzkurven zur Anordnung von SL-Blöcken und Hindernissen, die das Platzieren von Blöcken nicht erlauben. Vergleich von Blockanordnung von Greedy Algorithmus und Reinforcement Learning zeigt unterschiedliche Lösungen.



Quelle: DDU, TU Darmstadt 2021

Abbildung 15

Digitales und physisches Modell einer SL-Block-Aggregation auf Basis einer dreidimensional ausgedruckten Polylinie.



Quelle: DDU, TU Darmstadt 2021

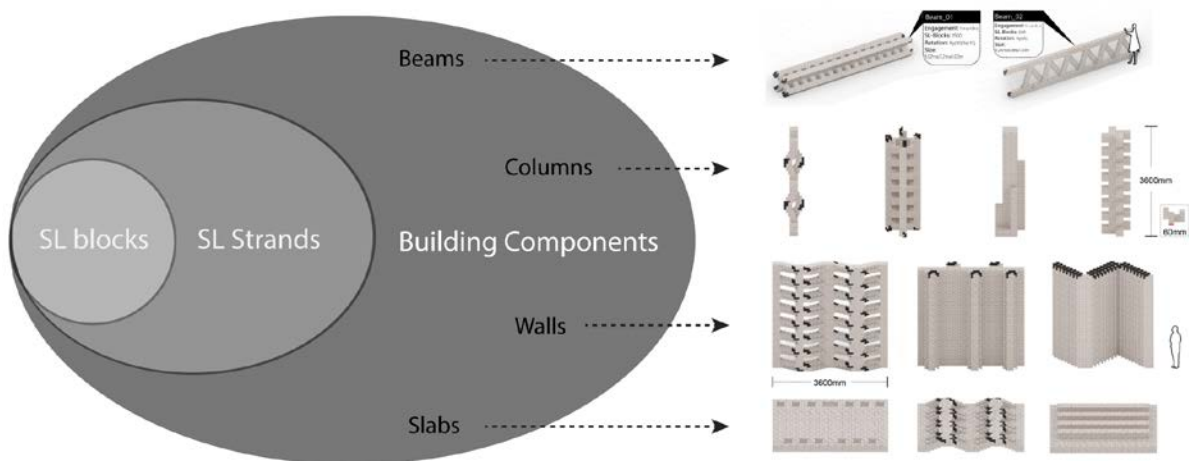
4.1.4 Bottom-Up Entwurfsexperimente

In einem kombinatorischen System sind die einzelnen Elemente gegeben. Die Berechnung im Computer befasst sich mit den herstellbaren Kombinationen solcher Elemente zu größeren Aggregationen und Konstruktionen. Erwächst die Konstruktion aus einem Bottom-up-Entwurfs-Experiment ist das Ergebnis offen und unbestimmt (Sanchez, 2014). Im Projekt haben wir untersucht, welche Formen aus SL-Blöcken mit einem Bottom-up-Ansatz entwickelt werden können. Die Experimente untersuchen, welche wertvollen Konfigurationen von SL-Strängen und deren Aggregationen entstehen können und welche Algorithmen, Methoden und Werkzeuge dafür entwickelt werden müssen.

Mit den Bottom-up-Experimenten erfolgte hinsichtlich der SL-Block Dimensionen (9x9x6cm Hüllform) auch ein konzeptioneller Umschwung im Projekt. Statt einer ursprünglich geplanten Vergrößerung des SL-Blocks als Bauteil haben wir Element-Hierarchien eingeführt: SL-Blocks > SL-Strands > Bauteil (Träger, Stütze, Wand, Decke) entwickelt (vgl. Abbildung 16).

Abbildung 16

Grafische Darstellung der Elemente-Hierarchie. Bauteile bestehen aus SL-Strands. SL-Strands bestehen aus SL-Blöcken. Bauteile können Träger, Stütze, Wand oder Decke sein.



Quelle: DDU, TU Darmstadt 2021

Abbildung 17

Links: verschiedene SL-Strands und ihre räumliche Entwicklung. Rechts: Aggregationen und Bauteile, die auf der Kombination verschiedener Fügegrammatiken für SL-Strands basieren



Quelle: DDU, TU Darmstadt 2021

Damit entstehen größere Bauteile nicht durch größer skalierte SL-Blocks, sondern durch deren hierarchische Aggregation im ursprünglichen Maßstab. Damit erhöht sich die Auflösung einer Konstruktion und auch die Flexibilität in der Wiederverwendbarkeit. Der Mehraufwand in der Montage ist wegen der Automatisierung/Autonomisierung dieses Prozesses vernachlässigbar. Ein quaderförmiger SL Strand kann eine Stütze oder einen Träger darstellen. Das hierarchische Konstruktionsprinzip ermöglicht die Montage modularer Komponenten in verschiedenen Maßstäben und auch in Kombination mit konventionellen Bauteilen. Mehrere SL-Strands können zu Konstruktionen gefügt werden. Auf dieser Bauteil-Hierarchieebene spielt die komplexe SL-Block Verzahnung konstruktiv keine Rolle mehr. So kann beispielsweise ein Quader aus 500 SL-Blöcken als monolithisches Bauteil betrachtet werden, das mit anderen Bauteilen konventionell verbunden wird.

Abbildung 18

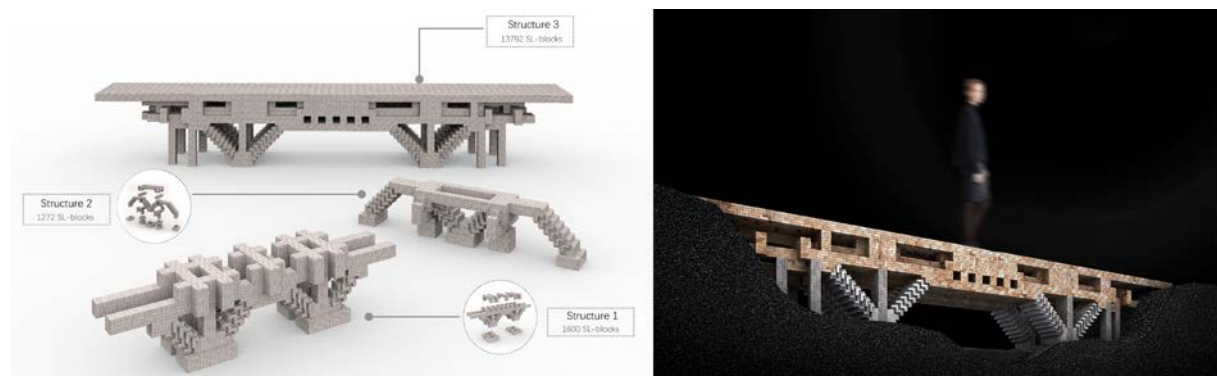
Stützen, Wände und Decken aus SL-Blöcken.



Quelle: DDU, TU Darmstadt 2021

Abbildung 19

Entwurf einer Brücke, in der verschiedene SL-Strands miteinander verbunden wurden. Rechts ist die Materialdifferenzierung erkennbar, die aus den verschiedenen Lastverteilungen resultiert. Druckbeanspruchte Bereiche werden aus Betonblöcken hergestellt. Spannende Bereiche aus hölzernen SL-Blöcken.



Quelle: DDU, TU Darmstadt 2021

Abbildung 20

Visualisierungen zu Montageszenarien. Bauteile werden von Robotern vormontiert und anschließend auf der Baustelle mit konventionellen Methoden des Fertigteilbaus verarbeitet.



Quelle: DDU, TU Darmstadt 2021

4.2 Top-Down Entwurfsstrategien für das hierarchische Fügen von SL-Blöcken

Im Verlauf der oben beschriebenen Bottom-Up-Entwurfsstudien müssen Architekt*Innen zum Erreichen von Bauteilen aus SL-Blöcken manuell Fügeregeln erproben, um komplexere SL-Stränge und verzahnte Strukturen zu schaffen.

Im Folgenden werden Strategien gezeigt, die weniger Experimentierfreude bzw. Auseinandersetzung mit der spezifischen SL-Block Grammatik erfordern, sondern in einem automatisierten Prozess beliebige Formen (ganze Bauteile) aus den dafür erforderlichen SL-Block-Sequenzen zusammensetzen. Dafür haben wir Algorithmen entwickelt, die einfach beschriebene Formen als Input bekommen und diese automatisiert in SL-Block Konfigurationen umwandeln. Damit sollen Methoden und Werkzeuge geschaffen werden, die die Anwendung von SL-Blöcken im Entwurf vereinfachen. Der Prozess von zwei verzahnten SL-Blöcken bis zum konstruktiven Bauteil durchläuft mehrere Schritte:

- Entworfenen Großformen werden zunächst dreidimensional gerastert (Voxel) und anschließend in eine Reihe von miteinander verzahnter 3D-Polyomino (Tetris-artige Polywürfel) zerlegt.
- Aus den 3D-Polyomino werden anschließend geschlossene Polylinien abgeleitet, deren Liniensegmente die geometrischen Mittelpunkte aller enthaltenen Voxel genau einmal durchlaufen.
- Übersetzung der Polylinien in Zeichenketten, die für die automatische Generierung von verzahnten SL-Blöcken genutzt werden können.

Im Folgenden werden die einzelnen Schritte im Detail erklärt und die Algorithmen im Rahmen von Entwurfsstudien getestet.

4.2.1 Bauteilhierarchien im SL-Block System

SL-Hierarchie

Wir beschreiben eine Reihe von Elementhierarchien, um die Abhängigkeiten zwischen den Elementen des verzahnten Systems deutlich zu machen und den Möglichkeitsraum beim Entwerfen mit SL-Blöcken zu strukturieren. Den Kern bilden die Würfel, die zu "L" und "S" Formen zusammengesetzt werden. "S" und "L" aus acht Würfeln bilden den SL-Block. Wie in Abbildung 22 dargestellt, führte Shih eine SL-Block Hierarchie ein, die auf dem konjugierten SL-Block-Paar basiert (Shih 2016).

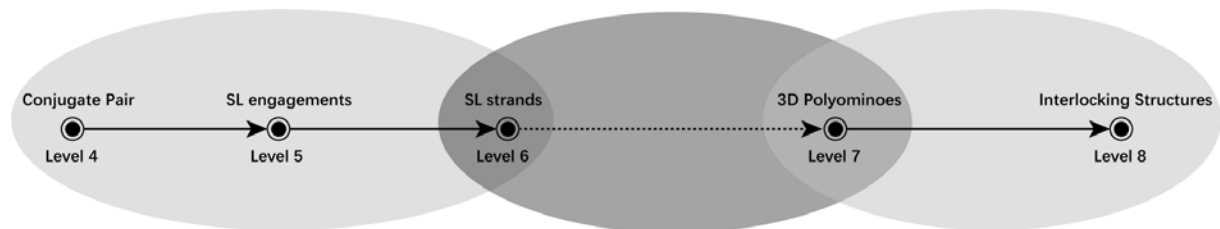
Zwei SL-Blöcke können auf sechs unterschiedliche Arten verzahnt werden. Diese verschiedenen Engagements bilden anschließend unterschiedliche SL-Strands. Auf der nächsten Ebene bilden SL-Strands Formen, die Bauelementen ähneln.

Zusätzlich zu Shih's Hierarchie zeigen wir neben den S- und L-förmigen Tetrawürfeln einen weiteren Tetrawürfeltyp, der als Paar die gleiche Form wie ein konjugiertes SL-Block-Paar bilden kann. Dieser Tetrawürfels ermöglicht beispielsweise die Verbindung von SL-Strands mit konventionellen linearen Bauteilen. Die SL-Block-Verzahnung wird dann zu einem Verbindungsdetail (vgl. Abbildung 23).

Außerdem fügen wir mit den 3D-Polyomino eine weitere Hierarchieebene hinzu (vgl. Ebene 7 in Abbildung 24). Ebene 8 in Abbildung 24 zeigt SL-Strukturen, die durch Aggregation von verschiedenen SL-Strängen

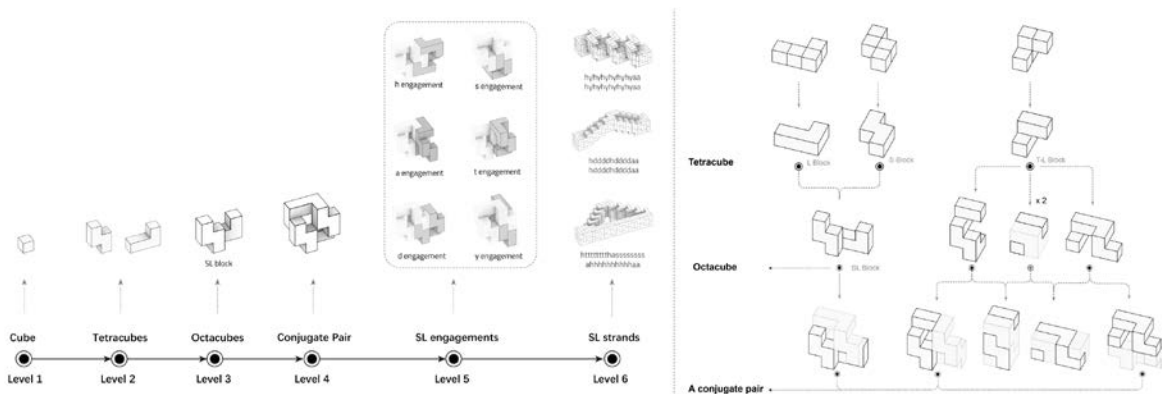
gebildet werden. SL-Stränge aus Ebene 6 werden zu größeren geometrischen und tektonischen Systemen zusammengefügt. Diese Konstruktionen beruhen nicht mehr auf der komplizierten Verzahnungslogik von SL-Blöcken. Sie werden stattdessen mit den drei in Abschnitt 4.1.2 vorgestellten Verbindungsarten montiert: klassische Holzverbindungen, verschachtelte und borromäische Verbindungen.

Abbildung 21
Grafische Darstellung der Elemente-Hierarchie.



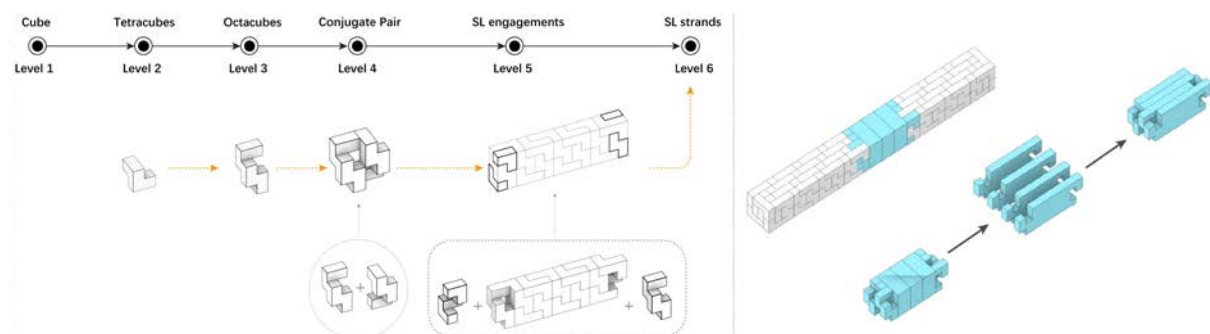
Quelle: DDU, TU Darmstadt 2022

Abbildung 22
Links: Zusammensetzung der SL-Blöcke aus Kuben. Rechts: Konjugierte SL-Block Paare.



Quelle: DDU, TU Darmstadt 2022

Abbildung 23
Links: Ein weiterer Octacube, mit einer anderen Form als die SL-Blöcke, kann einen SL-Strand abschließen, oder ihn mit linearen Bauteilen (blau) verbinden.



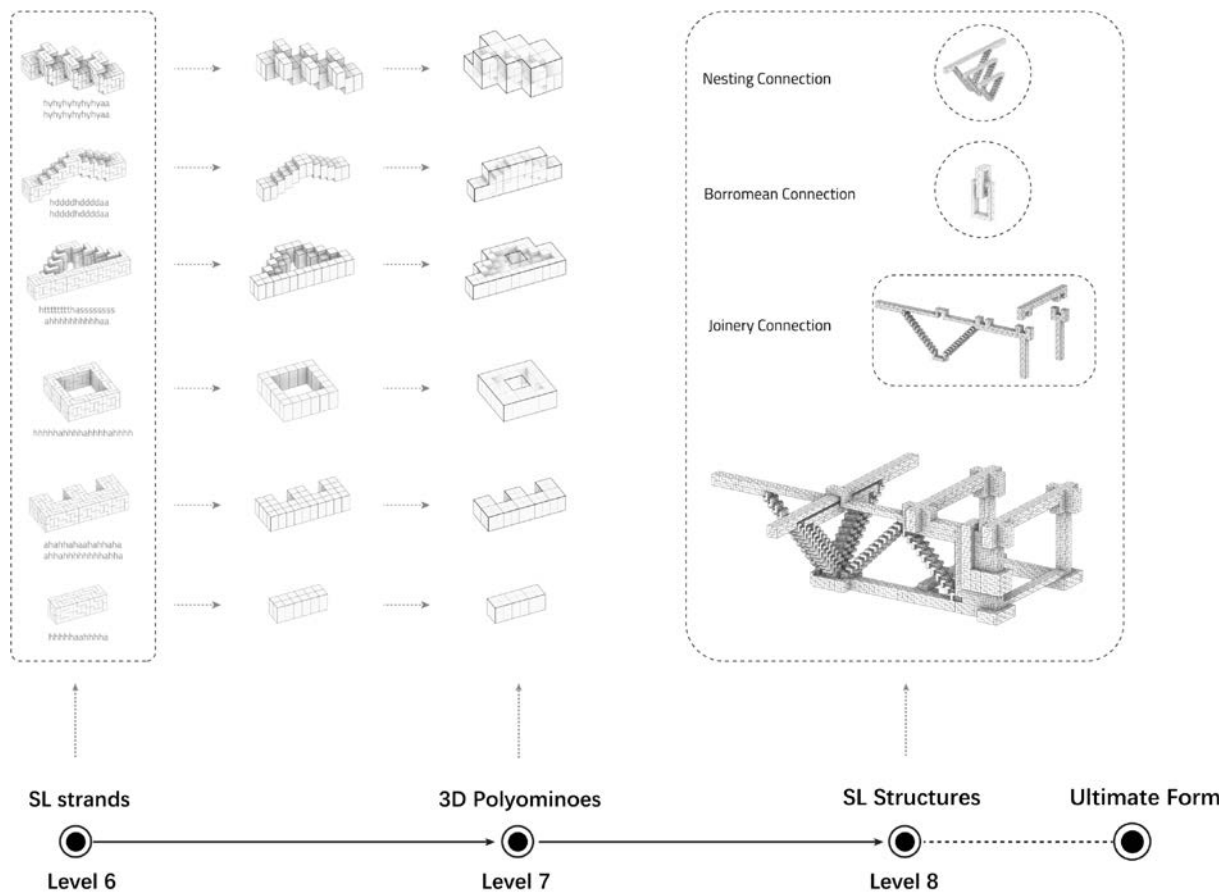
Quelle: DDU, TU Darmstadt 2022

Drei kombinatorische Optimierungsstrategien in der SL-Block-Hierarchie

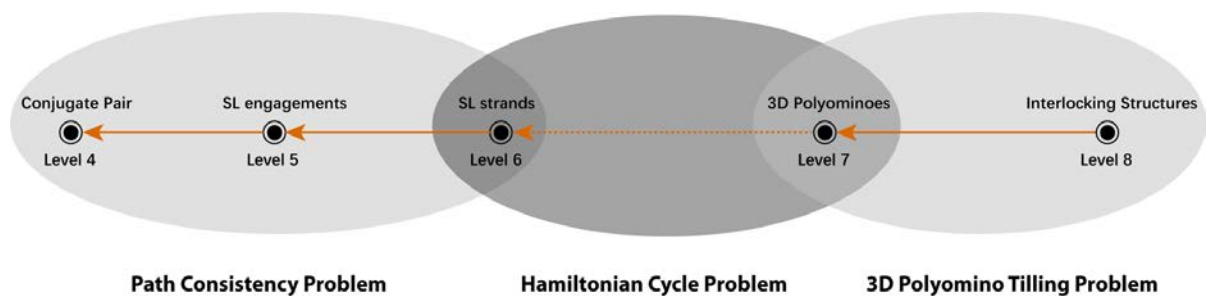
Bauteilhierarchien und drei dazugehörige Top-down Entwurfsstrategien: 1) Path Consistency Problem beschreibt wie man eine gegebene Polylinie in Zeichenketten für die automatisierte Generierung von SL-Strands überführt. 2) Das Hamiltonkreisproblem beschreibt die Suche nach geschlossenen Pfaden in gegebenen 3D-Polyominoes. 3) Das 3D-Polyomino Tiling untersucht, wie man eine in Voxel unterteilte Form in eine Gruppe verzahnter 3D-Polyominoes überführt.

Abbildung 24

Links: Ein weiterer Octacube, mit einer anderen Form als die SL-Blöcke, kann einen SL-Strand abschließen, oder ihn mit linearen Bauteilen (blau) verbinden.



Quelle: DDU, TU Darmstadt 2023



4.2.2 Graphenbasierte Analyse der SL-Hierarchie

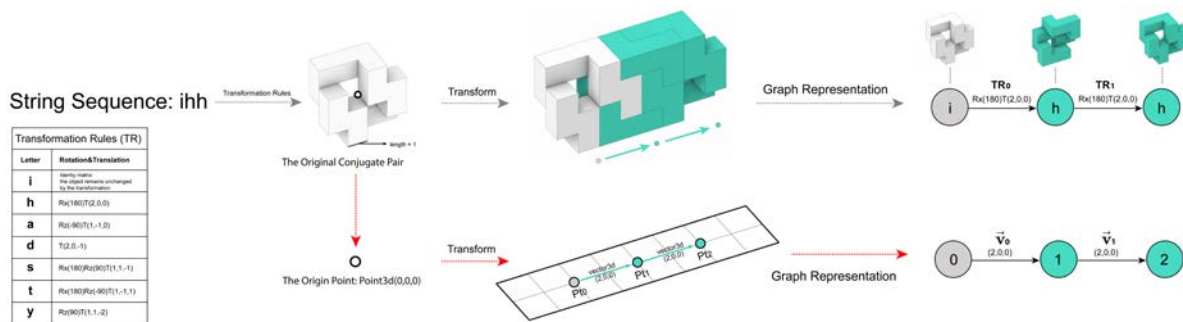
Die Erzeugung von SL-Strängen wird durch eine Zeichenkettensequenz gesteuert, die eine Liste von Transformationsregeln für die schrittweise Ausrichtung des konjugierten Paares von seiner Ausgangsposition zu bestimmten Zielen erstellen kann.

Jeder Buchstabe in der Sequenz entspricht einer vordefinierten Transformationsregel. In jedem Schritt wird diese Regel mit der vorherigen Transformation multipliziert, was zu einer neuen Transformation führt, die das ursprüngliche konjugierte Paar durch Rotation und Bewegung neu positioniert. Dieser iterative Prozess führt zu einem kumulativen Effekt, wobei jeder Schritt auf den in den vorangegangenen Schritten angewandten Transformationen aufbaut. In Fällen, in denen es keine vorherige Transformation gibt (wie beim ersten Schritt), wird die Regel mit der Identitätsmatrix der Größe 4x4 multipliziert, um sicherzustellen, dass die vordefinierte Transformation ihre Funktion beibehält. Durch die iterative Befolgung der Stringsequenz entfaltet sich eine Reihe geometrischer Transformationen, die die Anordnung der SL-Blöcke innerhalb der SL-Strand bilden (siehe auch 4.3.3)

Die SL-Strangerzeugung repräsentieren wir als einen gerichteten Graphen. Jeder Scheitelpunkt (Vertex) des Graphen symbolisiert einen Buchstaben, der die Transformationsregel angibt. Das Graphsegment zwischen Scheitelpunkten beschreibt die Transformationsregel des jeweiligen Buchstabens. Wir zeigen die Transformationsregel außerdem als Polylinie, die sich der räumlichen Figur der SL-Strands annähert (vgl Abbildung 25).

Abbildung 25

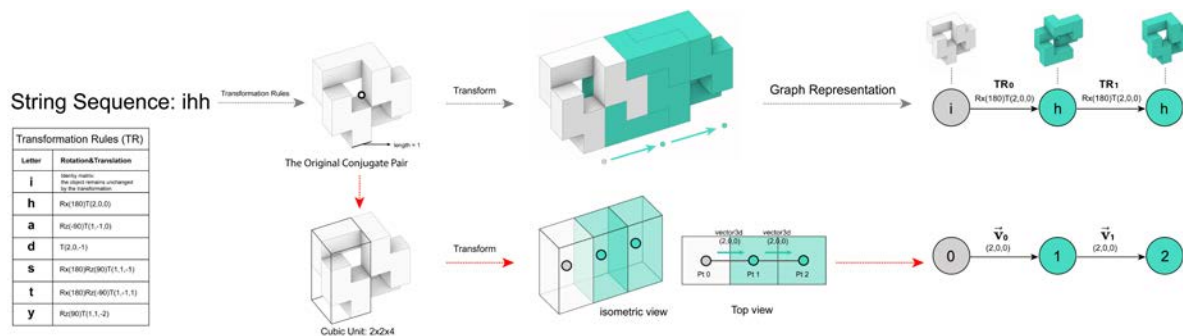
SL-Strang-Erzeugung. Links: Eine Zeichenkette für SL-Engagements. Mitte: SL-Strands-Erzeugung. Rechts: Darstellung des Prozesses als gerichteter Graph.



Quelle: DDU, TU Darmstadt 2023

Abbildung 26

Vereinfachte Darstellung eines konjugierten Paares als Voxel mit geometrischem Zentrum, an dem sich eine Polylinie ausrichten lässt, die zu einem Graphen weiterentwickelt wird.



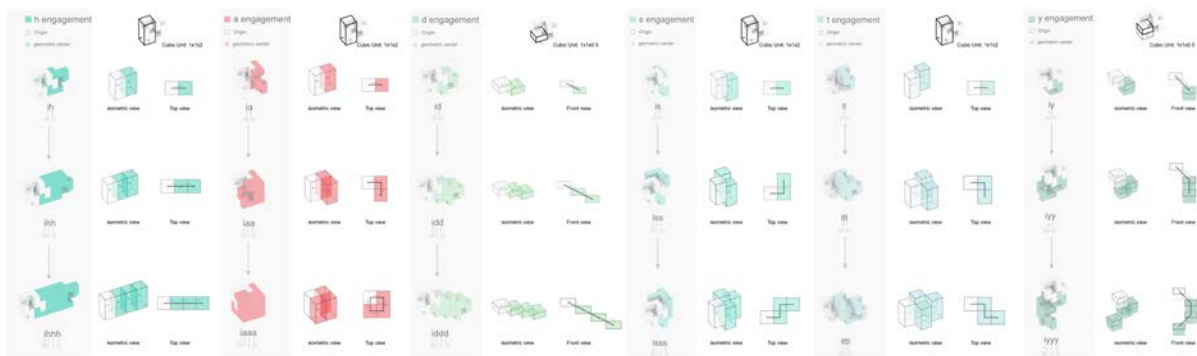
Quelle: DDU, TU Darmstadt 2023

Bei der dreidimensionalen Darstellung laufen die Pfade durch die geometrischen Schwerpunkte der Blöcke. Hier haben wir die SL Block-Darstellung vereinfacht, um die Lesbarkeit der Diagramme zu erhöhen (vgl. Abbildung 26). Die vereinfachten quaderförmigen Zellen haben keinen einheitlichen Maßstab: Die SL-Stränge, die durch die Buchstaben "h", "a", "s" und "t" erzeugt werden, können in der Mitte eines Würfels mit den Abmessungen 1x1x2 positioniert werden. Die von den Buchstaben 'd' und 'y' erzeugten Pfade sollten jedoch in der Mitte eines Würfels der Größe 1x1x0,5 liegen (vgl. Abbildung 27).

Bei der Erzeugung eines Polylinienpfades für einen neuen SL-Strang steht jeder Vektor für eine geometrische Transformationen von SL-Blöcken. Bei diesem sequentiellen Prozess muss die Aufrechterhaltung der Konsistenz zwischen der ursprünglichen Vektorliste des Pfades und der gedrehten Vektorliste, gewährleistet sein. Der transformierte Vektor muss nahtlos mit den gesamten Pfadvektoren übereinstimmen (vgl. Abbildung 28). Diese Formulierung kann als Pfadkonsistenzproblem angegangen werden. Die Methode besteht darin, iterativ den zugewiesenen Buchstaben für jeden Pfadknoten zu ermitteln und dabei die Angleichung des transformierten Vektors an den vorgegebenen Pfad zu erzwingen. Ziel ist es, den Polylinienpfad in eine gültige Zeichenkettensequenz zu dekodieren, die für die automatische Erzeugung von SL-Strängen genutzt werden kann.

Abbildung 27

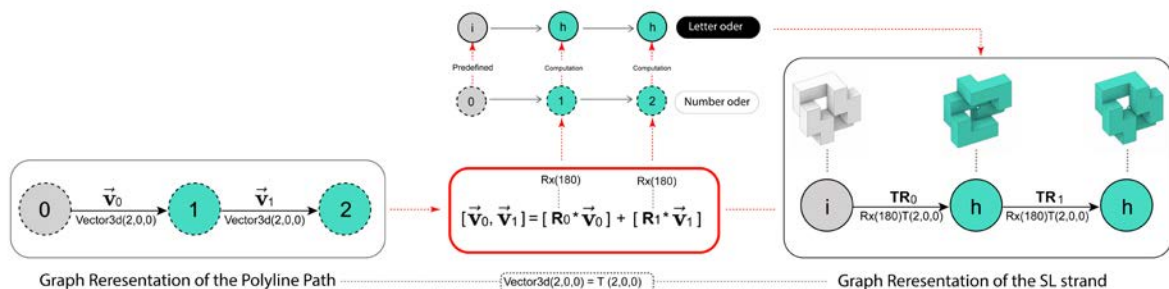
Sechs SL-Engagements reduziert als Voxeldarstellung. Die Komplexität der SL-Blöcke wird durch die Darstellung einer einfachen quader-artigen Hüllgeometrie reduziert. Wir nutzen zwei Quadergrößen, die bei der Annäherung an und Positionierung des generierten Polylinienpfades genutzt werden können.



Quelle: DDU, TU Darmstadt 2023

Abbildung 28

Eine Liste von Vektoren und ihre Drehwinkel als Zeichenketten und gerichteter Pfad.



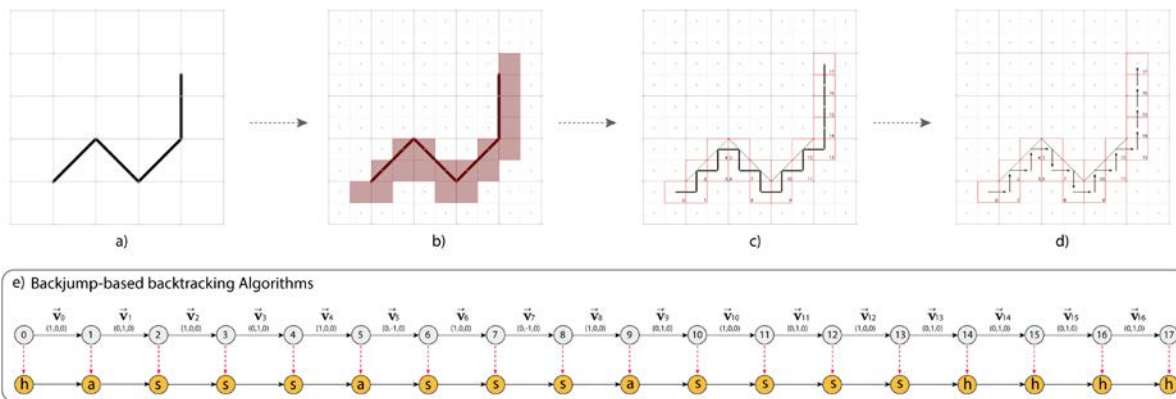
Quelle: DDU, TU Darmstadt 2023

Algorithmus zur Dekodierung eines Pfades in eine Zeichenkette

Im Folgenden zeigen wir, wie ein Pfad in eine Zeichenkette und schließlich in einen SL Strand übersetzt werden kann. Der Pfad muss auf einer Polylinie basieren, die in einem Raster mit einer Zellengröße 4x4 gezeichnet werden kann. Pfadsegmente verlaufen von einem Rasterpunkte zum nächsten in 0°, 45° oder 90° (vgl. Abbildung 29(a)). In einem zweiten Schritt wird das Raster in 2x2 Einheiten unterteilt und die Polylinie in einen Rasterpfad übersetzt (vgl. Abbildungen 29(b-d)). Auf Basis dieses Pfades können wir einen gerichteten Graphen erzeugen, in dem Punkte zu Schnittpunkten werden und Segmente zu Richtungsvektoren für die SL-Block-Anordnung.

Abbildung 29

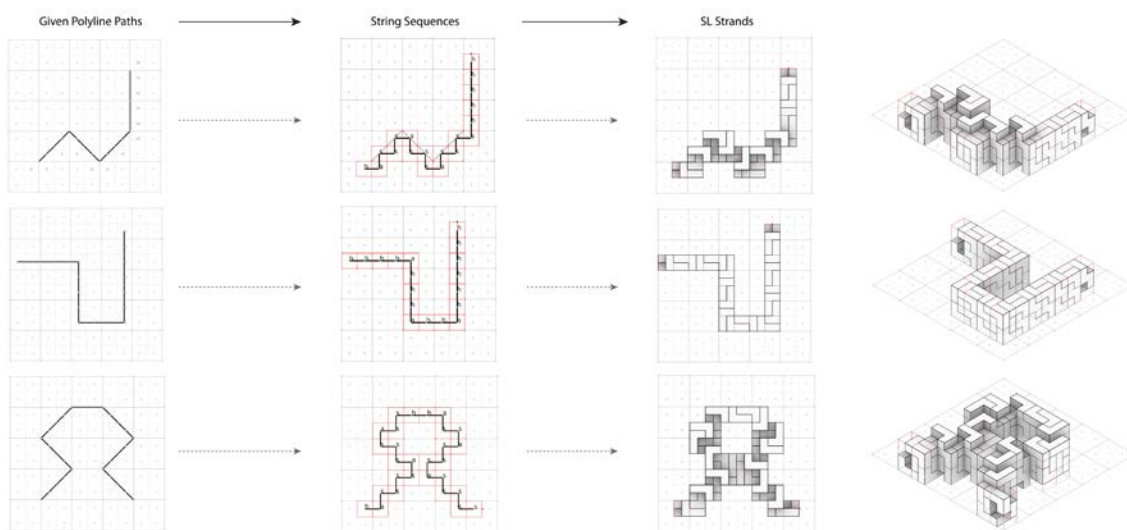
Übersetzung einer Polylinie in Richtungsvektoren für die SL-Block Anordnung. A) Quadratraster und Polylinie mit Segmenten in 0°, 45° und 90° Orientierung, die jeweils ein Rasterschnittpunkte miteinander verbinden. B) Einfärbung aller Rasterzellen, die an die Polylinie angrenzen ergibt eine flächige Darstellung der Polylinie im Raster. C) Vereinfachung der Polylinie: Die Mittelpunkte der unter b) eingefärbten Rasterzellen werden mit Segmenten in 0° oder 90° miteinander verbunden, so dass eine getreppte Polylinie entsteht. D) Ableitung der Richtungsvektoren aus den Segmenten von C). Unten im Bild: Übersetzung der zweidimensionalen Polylinie und ihre Segmentvektoren in Zeichenkette für SL-Engagements.



Quelle: DDU, TU Darmstadt 2023

Abbildung 30

Schrittweise Übersetzung einer Polylinie in einen komplexen SL Strand basierend auf den Prozess in Abbildung 29 und anschließend Übersetzung in 3D SL-Strand.



Quelle: DDU, TU Darmstadt 2023

Drittens führen wir einen Backtracking-Suchalgorithmus ein. Dieser Algorithmus ermöglicht die Umwandlung der Scheitelpunktsequenz in eine gültige Zeichenkette (vgl. Abbildung 29(e)) für die Erzeugung des entsprechenden SL-Strangs:

- i) An jedem Punkt wird ein Buchstabe ausgewählt und ihm der Wert der kumulativen Transformation zugewiesen. Für jeden Transformationswert invertiert der Algorithmus den Wert und prüft dann die Konsistenz zwischen dem resultierenden transformierten Vektor und dem ersten Vektor im gegebenen Pfad.
- ii) Ist dies der Fall, wird ein rekursiver Aufruf für den nächsten Punkt durchgeführt.
- iii) Ist dies nicht der Fall, kehrt der Algorithmus zurück.
- iv) Dieser iterative Suchprozess wird so lange fortgesetzt, bis eine gültige Zeichenkettensequenz dekodiert ist, die für die automatische Erzeugung von SL-Strängen aus verschiedenen Polylinien genutzt werden kann (vgl. Abbildung 30).

Das obige Prinzip zur Konstruktion der Polylinie funktioniert nur für die SL-Engagements "h", "a", "s" und "t". Um die Anwendung des entwickelten Algorithmus zur Dekodierung der Polylinie in die Zeichenfolge mit den SL-Engagements "d" und "y" zu erweitern (vgl. Abbildung 31), wird die Rastereinheit 2x2 weiter in 2x1 unterteilt, während die Richtung der gezeichneten Liniensegmente an die in "d" und "y" eingebettete Richtungseigenschaften angepasst werden muss (vgl. Abbildung 27).

4.2.3 Hamiltonkreisproblem

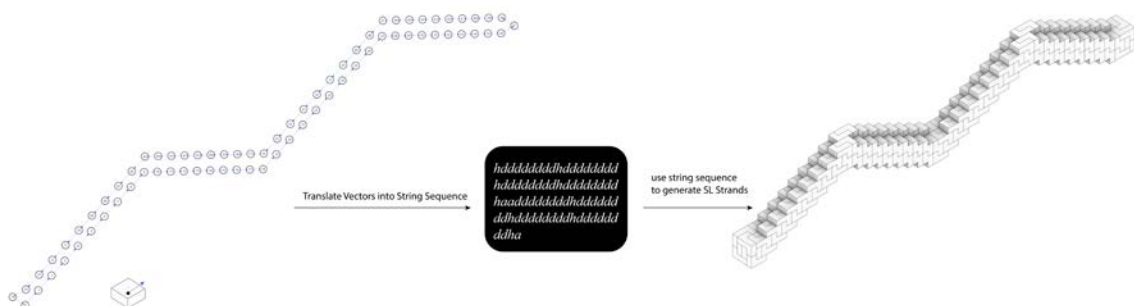
Graphenbasierte Analyse der SL-Hierarchie: Level6-Level7

(SL-Strand → Single Pfad → 3D-Polyomino)

Um SL-Strands effektiv als Bauteile mit einfachen Geometrien nutzen zu können, bevorzugen wir solche, die einen geschlossenen Pfad bilden (vgl. Abbildung 32), bei dem das letzte SL-Engagement mit dem ersten SL-Engagement verzahnt ist, so dass eine geschlossene Schleife entsteht, deren Elemente kinematisch miteinander

Abbildung 31

Die Umwandlung eines Pfades in SL-Blöcke basiert auf den engagements "h", "a" und "d". Ein Polylinienpfad aus Voxeln der Größe 2x2x1 wird berechnet und anschließend in eine Zeichenkette zur Erzeugung des entsprechenden SL-Strands umgewandelt.



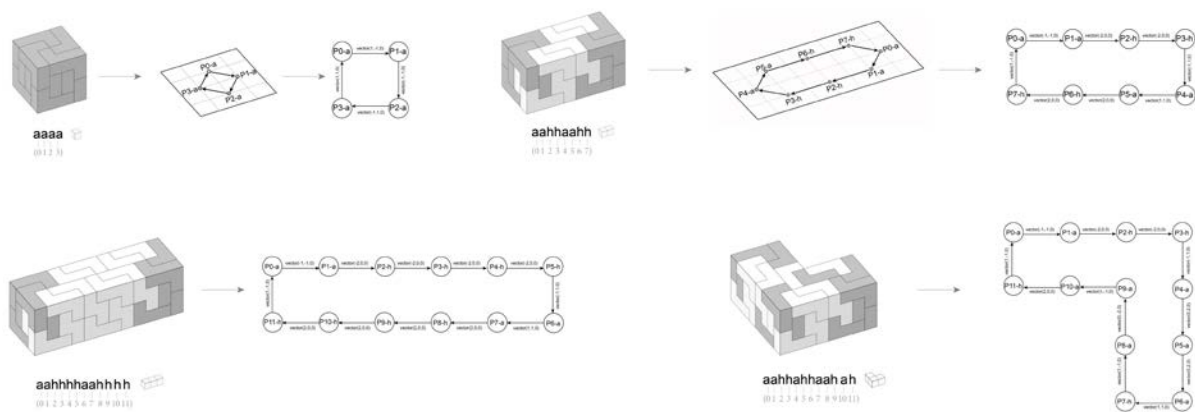
Quelle: DDU, TU Darmstadt 2023

verzahnt sind. Auf der Grundlage der in Abschnitt 4.2.2 vorgestellten Analyse kann der Polylinienpfad jedes SL-Strangs innerhalb eines 2D-Gitterraums positioniert werden. Diese quadratischen Zellen verbinden sich Kante für Kante miteinander und bilden die als Polyomino bekannte ebene geometrische Figur. Durch die Erweiterung der quadratischen Zellen in der Höhe kann das Polyomino zu einem 3D-Polyomino verallgemeinert werden, was eine vereinfachte geometrische Darstellung des SL-Strangs ergibt (vgl. Abbildung 33).

Die Suche nach geschlossenen Pfaden in den gegebenen 3D-Polyominos kann als Hamiltonkreisproblem formuliert werden, bei dem das Ziel darin besteht, einen Pfad zu finden, der jeden Knoten (hier: Mittelpunkt einer Rasterzelle) genau einmal besucht und zum Startknoten zurückkehrt, um die Schleife zu schließen. Im folgenden Abschnitt stellen wir die von uns entwickelten Algorithmen vor und zeigen, wie sie das Hamiltonkreisproblem im Kontext von 3D-Polyominos effektiv nutzen.

Abbildung 32

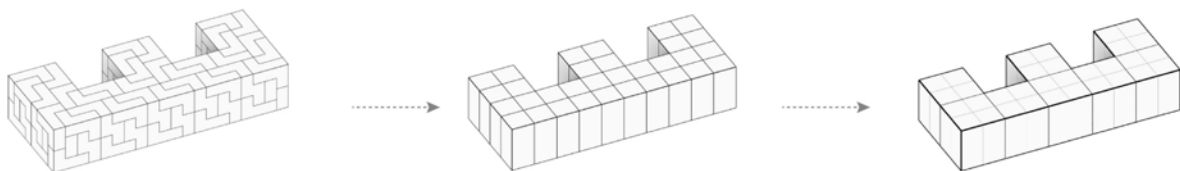
Graphen-basierte Analyse eines geschlossenen SL-Strands auf Basis von "h" und "a" Engagements.



Quelle: DDU, TU Darmstadt 2023

Abbildung 33

Ein geschlossener SL-Strands erzeugt von "h" und "a" Engagements zunächst übersetzt in ein Voxelraster und anschließend gegliedert in Kuben, die aus vier Quadrern gebildet werden.



Quelle: DDU, TU Darmstadt 2023

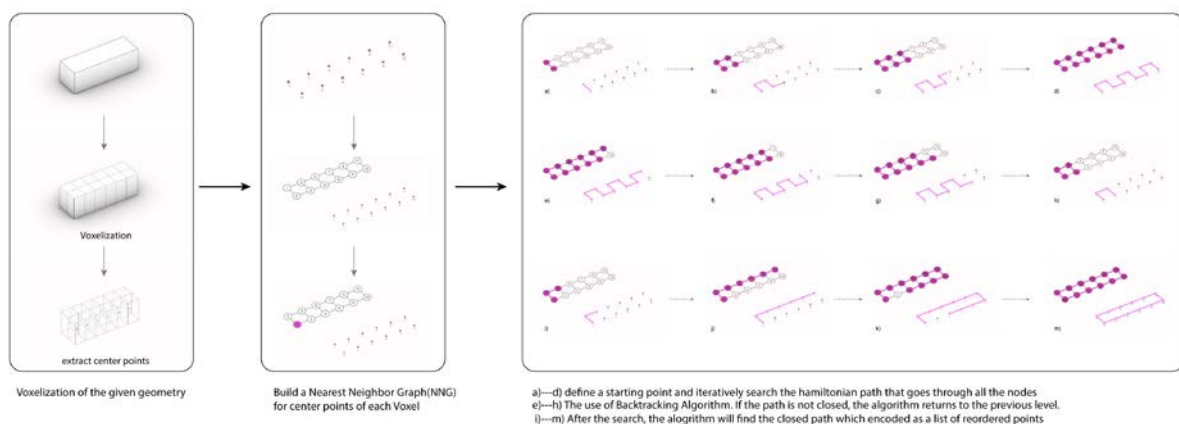
Algorithmus zum Findes des Hamiltonkreises in einem Quadratraster

3D Polyomino → Einzelpfad → SL Strand

Zur Identifizierung eines Hamiltonkreises in dem 3D-Polyomino Graphen nutzen wir einen Backtracking-Algorithmus. Dieser Algorithmus geht von einem ausgewählten Knoten aus und versucht immer, den Pfad zu verlängern, indem er nach und nach den besuchten Knoten und einen seiner nicht besuchten Nachbarn hinzufügt.

Abbildung 34

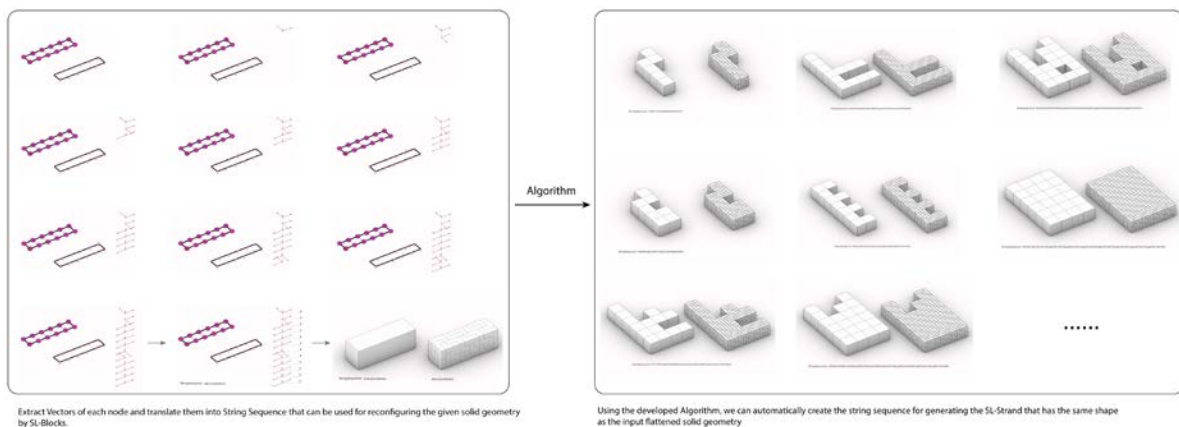
Von der Graphendarstellung zum Volumen. Die Vektoren werden an jedem Knoten extrahiert und in eine Zeichenfolge überführt, die für eine Generierung des Volumens aus SL-Blocks genutzt werden kann. Mit diesem Algorithmus können voxel-basierte Volumen (Polycubes) in verzahnte SL-Blocks überführt werden.



Quelle: DDU, TU Darmstadt 2022

Abbildung 35

Von der Graphendarstellung zum Volumen. Die Vektoren werden an jedem Knoten extrahiert und in eine Zeichenfolge überführt, die für eine Generierung des Volumens aus SL-Blocks genutzt werden kann. Mit diesem Algorithmus können voxel-basierte Volumen (Polycubes) in verzahnte SL-Blocks überführt werden.



Quelle: DDU, TU Darmstadt 2022

Wenn der erweiterte Pfad den Hamiltonkreis nicht zulässt, kehrt der Algorithmus zurück, indem er den zuletzt hinzugefügten Knoten entfernt und den Suchprozess mit einem anderen, nicht besuchten Nachbarn wiederholt. Der Backtracking-Algorithmus kehrt erst zurück, wenn er einen Hamiltonkreis gefunden hat oder klar ist, dass kein Hamiltonkreis im 3D-Polyomino existiert (vgl. Abbildung 34).

Obwohl die Implementierung des Backtracking-Algorithmus die Erstellung eines geschlossenen Pfades ermöglicht, der anschließend mit den in Abschnitt 4.2.2 entwickelten Algorithmen in den entsprechenden SL-Strang entschlüsselt werden kann (vgl. Abbildung 35), ergeben sich bei einem größeren Suchraum große Herausforderungen. Der Backtracking-Algorithmus mit einer Brute-Force-Suchstrategie verifiziert alle möglichen Nachbarn des Scheitelpunkts.

Mit einem Algorithmus, der schrittweise den Folgezustand auswählt, der zum Zeitpunkt der Wahl das beste Ergebnis verspricht (Greedy Algorithmus), suchen wir nach den Hamiltonkreis mit den folgenden Schritten (vgl. Abbildung 36):

- 1) Unterteilung der Rasterzellen: Jede Zelle des 3D Polynomino wird in vier kleinere Einheiten unterteilt.
- 2) Erstellen eines quadratischen Gittergraphen: Erstellung eines ungerichteten Graphen aus den Mittelpunkten der kleineren Quadrate. Der Scheitelpunkt bezeichnet dabei den Punktindex, und die Kante verbindet den Scheitelpunkt mit seinen engsten Nachbarn.
- 3) Prune Edges Using MST: Verwenden des generierten Pfades, um die Kanten des Graphen zu beschneiden, die sich mit dem Pfad kreuzen.
- 4) Gewichtung der verbleibenden Kanten: Basierend auf dem Abstand zwischen ihren Mittelpunkten und dem Pfad wird jeder verbleibenden Kante ein normalisierter Abstandswert als Gewicht zugewiesen.
- 5) Greedy Suche des Hamiltoniankreis : Definition des Startpunkts und Integration des Backtracking in den Suchprozess, indem der Pfad schrittweise um den nicht besuchten Nachbarn erweitert wird, dessen Kante einen höheren Gewichtswert hat.
- 6) Rückgabe einer Liste von geordneten Punktindizes: Diese Liste kann verwendet werden, um die Mittelpunkte der ursprünglichen kleinen Quadrate neu anzuordnen und so einen geschlossenen Polylinienpfad innerhalb des 3D-Polyominos zu konstruieren (vgl. Abbildung 36(2)).

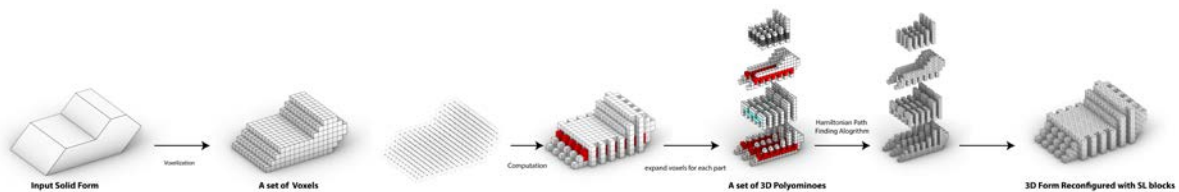
Die resultierende Polylinie wird mit Hilfe des in Abschnitt 4.2.2 entwickelten Algorithmus in eine Stringsequenz dekodiert, die die SL-Strands erzeugt. Der von uns entwickelte Algorithmus zum Auffinden eines Hamiltonkreise kann effizient den optimalen geschlossenen Pfad aus einer Vielzahl von 3D-Polyominos berechnen, unabhängig vom Maßstab, selbst wenn die Anzahl ihrer zusammengesetzten Voxel Tausende übersteigt (vgl. Abbildung 36(3-6)).

4.2.4 3D Polyomino Kachelung

In dem hier vorgestellten Top-Down Ansatz, können Entwerfende die grobe Zielform entwickeln und die nachfolgende Zerlegung der Form in verzahnte SL-Blöcke einem automatisierten algorithmischen Prozess überlassen, der sicherstellt, dass alle zerlegten Teile ineinandergreifen (Song, 2022). Im Rahmen des Projekts haben wir weitere Algorithmen entwickelt, um in 3D Rastern 3D-Polyominoe zu erzeugen. Diese Polyomino werden dann automatisch in SL-Strands umgewandelt, wodurch monolithische Bauteile durch SL-Block Aggregationen nachempfunden werden können. (vgl. Abbildung 38). Diese Algorithmen können nahtlos in den digitalen Entwurf verschiedener Bauelemente oder Strukturen unter Verwendung von SL-Blöcken integriert werden und wurden mit Studierenden der TU Darmstadt getestet (vgl. Abbildung 39).

Abbildung 38

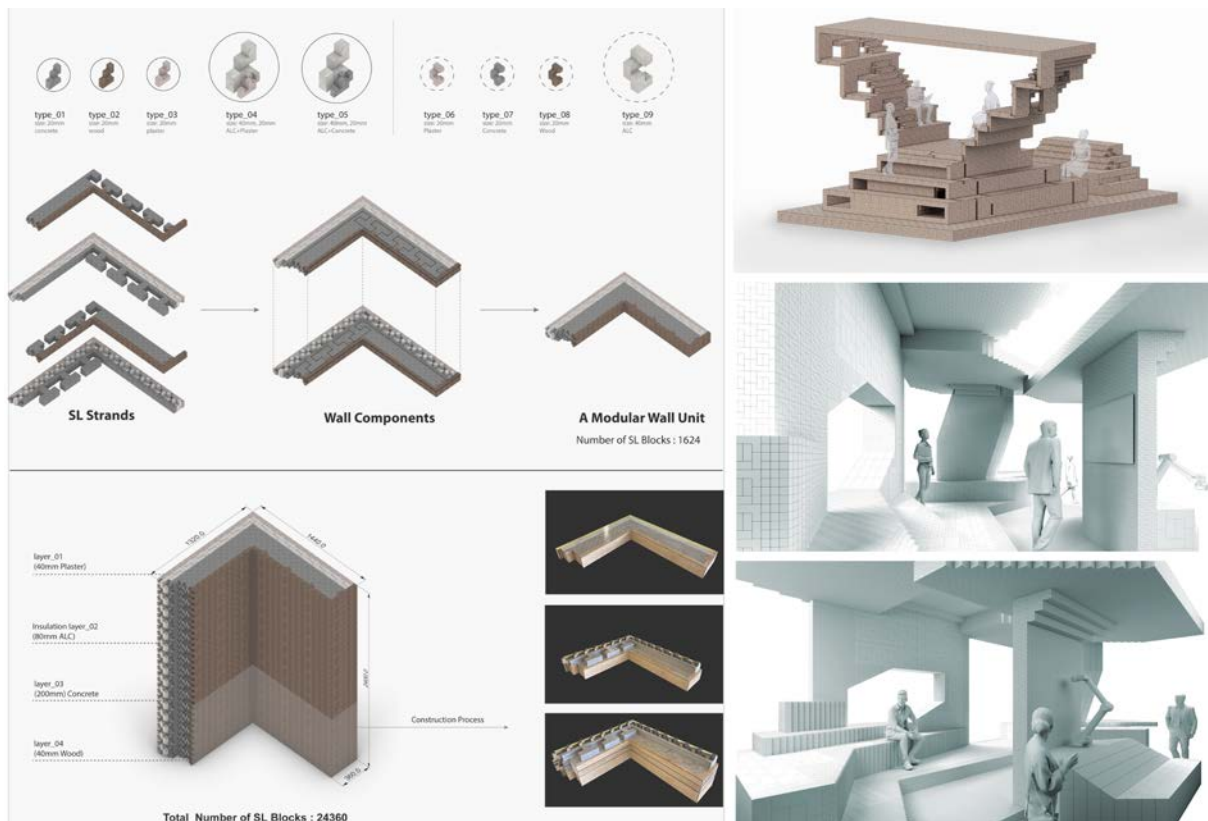
Bildsequenz zur Überführung eines Volumens in eine SL Block Aggregation. 1. Startvolumen zeigt eine Art massiven Sessel. 2. Übersetzung des Volumens in Voxel. 3. Übersetzung der Voxel in ein Raumraster. 4. Zerlegung des Raumrasters einzelne, verzahnte Voxelformen. 5. Generierung der geschlossenen Polylinien in den Voxelformen mittels Hamiltonkreis. 6. Generierung der SL-Strands entlang der Polylinien



Quelle: DDU, TU Darmstadt 2023

Abbildung 39

Entwurfsstudien von Studierenden der TU Darmstadt unter Verwendung der im Projekt entwickelten Algorithmen.



Quelle: DDU, TU Darmstadt 2023

4.3 Materialisierung von SL-Blöcken

Wie haben umfangreiche Untersuchungen zur Herstellung von SL-Blocks aus unterschiedlichen Materialien und mit unterschiedlichen Herstellungsverfahren durchgeführt. SL-Blocks wurden aus Beton, Modellbaugips, Acryl-Gießharz in Schalungen gegossen.

Hölzerne SL-Blocks wurden aus linearen Multiplex-Hölzern zunächst zu langen S- und L-Profilen verleimt und anschließend wie Salamischeiben zu einzelnen Teilen geschnitten. Anschließend wurden beide „Buchstaben“ miteinander verleimt. In einem anderen Verfahren wurden die Blöcke aus einzelnen flachen Teilen mittels Laserschneiden und CNC-Fräsen hergestellt, was eine modifizierte Geometrie der Teile erforderte (vgl. Abbildung 40). Weitere Teile wurden mittels 3D-Druck hergestellt.

Auch hier kam ein Verfahren zum Einsatz, bei dem die Bauteile aus einzelnen Teilen zusammengesteckt wurden. Dadurch konnte der 3D Druckprozess beschleunigt werden. Andere Teile wurden gedruckt, um komplexere Geometrien zu testen, die mit konventionellen Herstellungsmethoden zu aufwändig gewesen wären. Die Bauteile erhielten ausgerundete Kanten, um die Montage zu erleichtern. Andere Teile erhielten Vertiefungen oder Riefen, deren unterschiedliche Nutzungen unten beschrieben werden. Außerdem wurden erste Testblöcke aus Mycel in tiefgezogenen Formen hergestellt, um perspektivisch SL-Block-Dämmschichten für Wände zu erzeugen.

Abbildung 40

SL-Blocks aus unterschiedlichen Materialien, hergestellt mit unterschiedlichen Fabrikationstechniken und variiert Geometrie. Reihe 1.: Beton, Gips, Vollholz, Schichtholz. Reihe 2: Schichtholz, elementiert und CNC gefräst, Lindenholz gelasert, 3D-Kunststoffdruck. 3. Reihe: Acrylglas verschiedene Farben. Reihe 4/5: 3D-Druck unterschiedliche Formen.



Quelle: DDU, TU Darmstadt 2023

4.4 Visuelle taktile Sensorsysteme



Foto: Team DDU&IAS TU Darmstadt 2021.

Visuelle taktile Sensorsysteme installiert auf einem Robotergrifer ermöglichen Fingerspitzengefühl für die Roboter.

Taktile Sensoren sind Sensoren, die Informationen über physische Kontaktpunkte eines Roboters mit seiner Umgebung liefern. Darunter fallen z.B. Informationen über die an Kontaktpunkten wirkenden Kräfte oder die Oberflächenbeschaffenheit und Form von Objekten. Somit reproduzieren taktile Sensoren die Fähigkeiten menschlicher Haut, insbesondere den biologischen kutanen Tastsinn, der es ermöglicht, isolierte Berührungen und Drücke, die über interaktive Bereiche räumlich aufgelöst sind, zu spüren. Insbesondere in Situationen, in denen visuelle Sensorik keine oder unzureichende Informationen liefern, z. B. wenn der Roboter kritische Teile der Szene mit seinen Manipulatoren verdeckt oder wenn eine Aufgabe Informationen über Kontaktkräfte erfordert, ist der Einsatz taktile Sensoren für Mensch und Maschine unabdingbar.

Die Anwendung taktile Sensoren in technischen Systemen eröffnet neue Möglichkeiten für die Mensch-Maschine-Interaktion, die Sicherheit von Systemen und die Überwachung von Prozessen. Potenzielle Anwendungsbereiche reichen von der klassischen industriellen Automation und fortgeschrittenen Robotik-Anwendungen bis zur Medizintechnik. Zurzeit gibt es verschiedene Implementierungsansätze für taktile Sensoren. Ein Design, das aufgrund seiner hohen spatialen Auflösung und gleichzeitiger Kostengünstigkeit zurzeit hohe Popularität in der Wissenschaft genießt, sind visuelle taktile Sensoren. Visuelle taktile Sensoren zeichnen sich dadurch aus, dass sie eine Kamera im Inneren des Roboters nutzen, um Daten aufzunehmen. Üblicherweise sitzt diese Kamera in den Fingerspitzen des Roboters hinter einem Gel und nimmt dessen durch Kontakt induzierte Verformung auf. Dadurch ist es möglich, mittels Methoden der visuellen Datenverarbeitung, Informationen über die Textur und lokale Form von Objekten zu extrahieren.

4.4.1 DIGIT Sensor

Abbildung 41

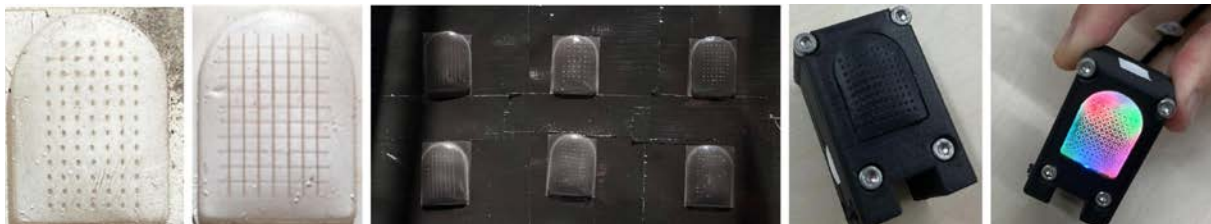
Standarddesign des taktilen Sensors Digit (<https://digit.ml/>), der als Grundlage für unsere Arbeit diente. Die Bilder auf der rechten Seite zeigen die Rohdaten des Sensors, wenn kein bzw. ein scheibenförmiges Objekt berührt wird.



Quelle: Team DDU&IAS, TU Darmstadt 2021

Abbildung 42

Links sind die Gele mit lasergraviierten Punkt- und Streifenmustern auf der Oberfläche des Sensors zu sehen, die eindeutige lokale Merkmale aufweisen, welche zur Quantifizierung der Verformung des Gels genutzt werden können. In der Mitte sind die schwarz beschichteten Gele zu sehen, die Lichtreflexionen reduzieren. Rechts wird das Gesamtdesign des Sensors mit einem quadratischen Gehäuse und dickeren modifizierten Gelpads vorgestellt.



Quelle: Team DDU&IAS, TU Darmstadt 2022

Ein Beispiel für einen visuellen taktilen Sensor ist der DIGIT Sensor. Wie in Abbildung 41 zu sehen ist, besteht der Digit Sensor aus einer RGB-Kamera hinter einem Plexiglas-Element, das von drei verschiedenfarbigen LEDs beleuchtet wird. Auf dem Plexiglas-Element ist eine Gelschicht aufgebracht, die durch eine Farbbeschichtung von außen lichtundurchlässig gemacht wird. Ist der Sensor in Kontakt mit einem Objekt, verformt sich das Gel, was sich mittels der Kamera beobachten lässt. Siehe Abbildung 41 für ein Beispiel der durch die Kamera des DIGIT Sensors aufgenommenen Verformung des Gels. Die Vorteile des DIGIT Sensors sind sein relativ simples Design und die hohe spatiale Auflösung, die dieser bietet. Jedoch wurden mehrere Nachteile des Sensordesigns in Bezug auf dessen Fähigkeit, Objekte zu erkennen und Kräfte und Drehmomente abzuschätzen, identifiziert: i) das Silikonkugelpolster ist zu hart und zu dünn, daher ist die Verformung des Gels bei Kontakt mit Objekten nicht ausreichend; ii) das Fehlen von Merkmalen auf der Oberfläche des Gels erschwert die Abschätzung von Scherkräften; iii) die weiße Oberfläche des Gels führt zu Lichtreflexionen und macht die Abschätzung der Normalkraft unzuverlässig. Daher wurde eine Reihe von Verbesserungen am Sensordesign vorgenommen, um eine zuverlässige Objekterkennung und Kraft-/Drehmomentabschätzung zu ermöglichen: 1) Erstens wurden die Dicke des Gels und die Silikonformel angepasst, um das Gel weicher und damit empfindlicher für Berührungen zu machen; 2) Zweitens wurden Muster aus Punkten und Streifen per Laser in die Oberfläche des Gels eingraviert, um eine bessere Schätzung der Scherkraft zu ermöglichen; 3) Drittens wurden die Gele schwarz beschichtet, um Interferenz von externem Licht auszuschließen und interne Reflexionen zu verringern, was zu einer zuverlässigeren Schätzung der Normalkraft führte. Abbildung 42 zeigt die verbesserten Gele.

4.4.2 Erkennung von Materialien durch Roboter mit visuell-taktilen Sensoren mittels KI

Taktile Sensoren erlauben es Menschen, die Materialbeschaffenheit unbekannter Objekte zu bestimmen. Wissen über die Materialbeschaffenheit ist für die Manipulation unbekannter Objekte hochrelevant, da deren korrekte Behandlung häufig vom Material abhängig ist. So muss z.B. eine Glasplatte mit höherer Vorsicht behandelt werden als eine Metallplatte. Da ein zentraler Indikator für Materialbeschaffenheit die Oberflächentextur ist, wurde in diesem Projekt ein Klassifikator für Oberflächentexturen trainiert. Dazu wurde ein Datensatz von Objekten mit unterschiedlichen Oberflächeneigenschaften erstellt (vgl. Abbildung 43) und ein Modell darauf trainiert.

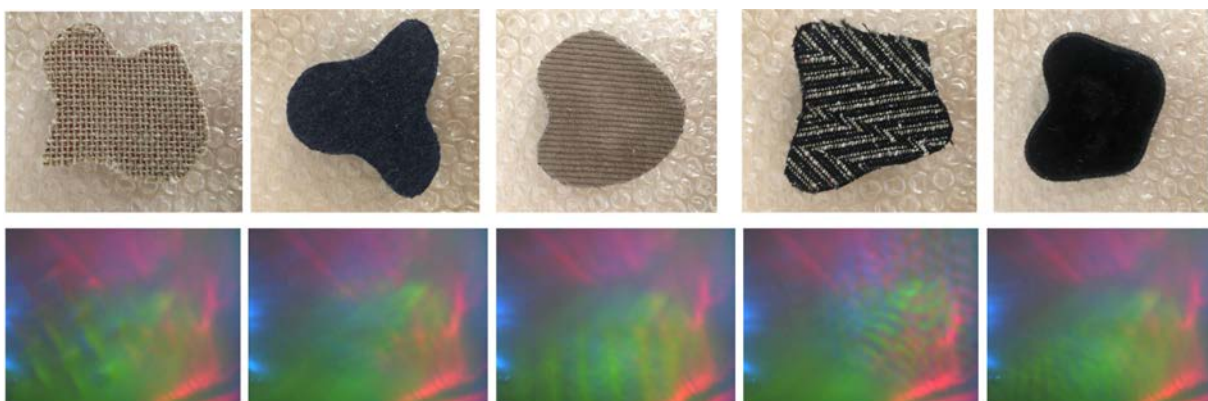
Eine zentrale Erkenntnis dieses Projekts ist, dass das Design des DIGIT Sensors die Klassifikation von Texturen stark begünstigt. So lag die Wiedererkennungsrates von Texturen bei menschlichen Versuchsteilnehmern bei gerade mal 67%, während sich mittels des DIGIT Sensors

4.4.3 Berechnung der Kontaktkräfte und Drehmomente für die Greif- und Pfadplanung

Wie bereits zuvor erwähnt, ist ein zentraler Vorteil von taktilen Sensoren gegenüber rein kamerabasierten Systemen, dass diese den Roboter mit Informationen über wirkende Kräfte an den Kontaktpunkten liefern können. Für die Schätzung von Kontaktkräften, Druck und Drehmomenten wurden daher zwei Ansätze untersucht. Einerseits wurde ein modellbasierter Ansatz entwickelt, der auf photometrischer Nahlicht-Stereoanalyse beruht, um die Normalkraft zu schätzen, basierend auf dem Frankot-Chellappa-Algorithmus (vgl. Abbildung 44). Dieser wurde mit einer auf optischer Strömung basierenden Scherkraftschätzung gekoppelt (vgl. Abbildung 44). Der Algorithmus ist effizient und läuft problemlos mit einer Rate von 30 Hz. Er erfordert jedoch eine Kalibrierung. Weiterhin wurde eine lernbasierte Lösung entwickelt, bei der ein modifiziertes U-Netz trainiert wurde, um eine Kraftverteilung auf der gesamten Oberfläche des Sensors vorherzusagen. Dazu wurde ein gelabelter Datensatz verwendet, der mit einem vordefinierten Satz von Objekten mit bekannten Formen erarbeitet wurde (vgl. Abbildung 45). Das in der Abbildung 46 gezeigte Robotersetup verfügt über einen konventionellen Kraft-Drehmoment-Sensor, der am Tisch befestigt ist und als Basiswert für die Gesamtkraft dient, die der visuell-taktile Sensor erfährt. Ein Vergleich zwischen der modell- und der lernbasierten Kraft-/Drehmomentschätzung ist Gegenstand laufender Arbeiten. Vorläufige Ergebnisse deuten auf Vorteile der lernbasierten Lösung hin, die bei einer sehr geringen Menge von Demonstrationen schon eine Genauigkeit von 90% erreichen ließ.

Abbildung 43

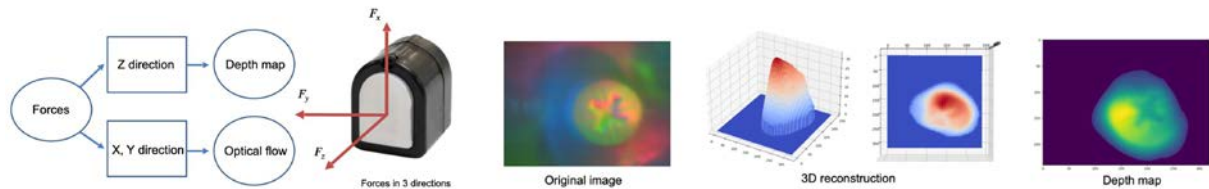
Ein Klassifikator für Materialoberflächeneigenschaften wird anhand von Sensor-Rohbildern trainiert. Die obere Reihe zeigt Beispielobjekte mit unterschiedlichen Oberflächentexturen, die untere Reihe zeigt die entsprechenden Tastbilder. Ein Klassifikator, der auf vortrainierten ResNet18-Merkmalen basiert, erreicht eine Genauigkeit von 80 %, wenn er auf nur 6 taktilen Bildern pro Klasse trainiert wird.



Quelle: Team DDU&IAS, TU Darmstadt 2022

Abbildung 44

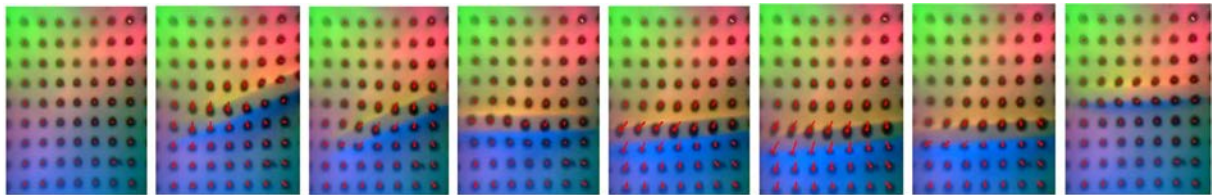
Die Normalkraft (z-Richtung) wird auf der Grundlage einer rekonstruierten Tiefenkarte geschätzt, während die Scherkraft (x- und y-Richtung) auf der Grundlage des optischen Flusses geschätzt wird. Auf der rechten Seite ist ein Beispiel für ein taktiles Bild zusammen mit einer 3D-Rekonstruktion der erfassten Objektform und der entsprechenden Tiefenkarte dargestellt. Die Tiefenkarte wird mit Hilfe des entwickelten Algorithmus geschätzt.



Quelle: Team DDU&IAS, TU Darmstadt 2023

Abbildung 45

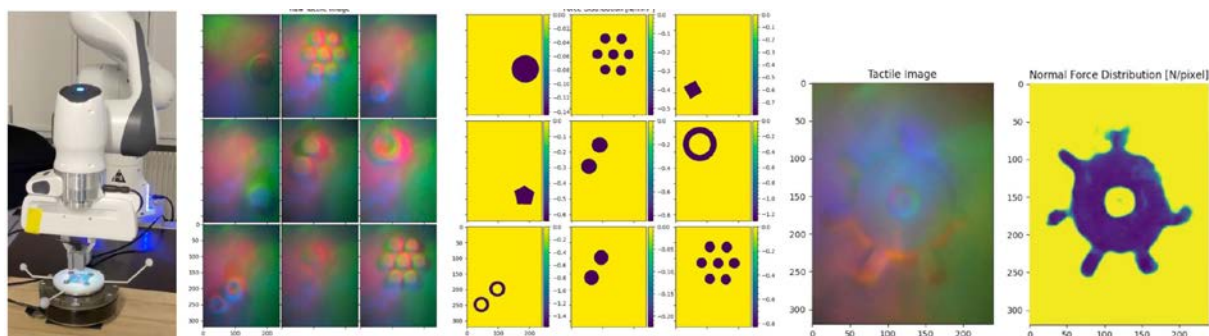
Die Scherkraft ist proportional zur Verformung des Gels in dem Kraftbereich, für den der Sensor geeignet ist. Daher werden die Abweichungen der Punkte im Gel verfolgt, und die Werte der Punktabweichungen werden als proportional zur Scherkraftverteilung angenommen.



Quelle: Team DDU&IAS, TU Darmstadt 2023

Abbildung 46

Der entwickelte lernbasierte Ansatz zur Schätzung der Normalkraft. Ein Datensatz mit taktilen Bildern und den entsprechenden Kräften wird durch den Kontakt des Roboters mit einem vordefinierten Satz von 3D-gedruckten Objekten mit Formen und flachen Oberflächen gesammelt (linke Seite der Abbildung). Die Methode wird an nicht-konvexen Objekten evaluiert (zwei Bilder auf der rechten Seite). Die geschätzte Normalkraftverteilung stimmt visuell mit der Form des Testobjekts überein und kann eine Genauigkeit von bis zu 1 g bei der Gesamtnormalkraft erreichen.

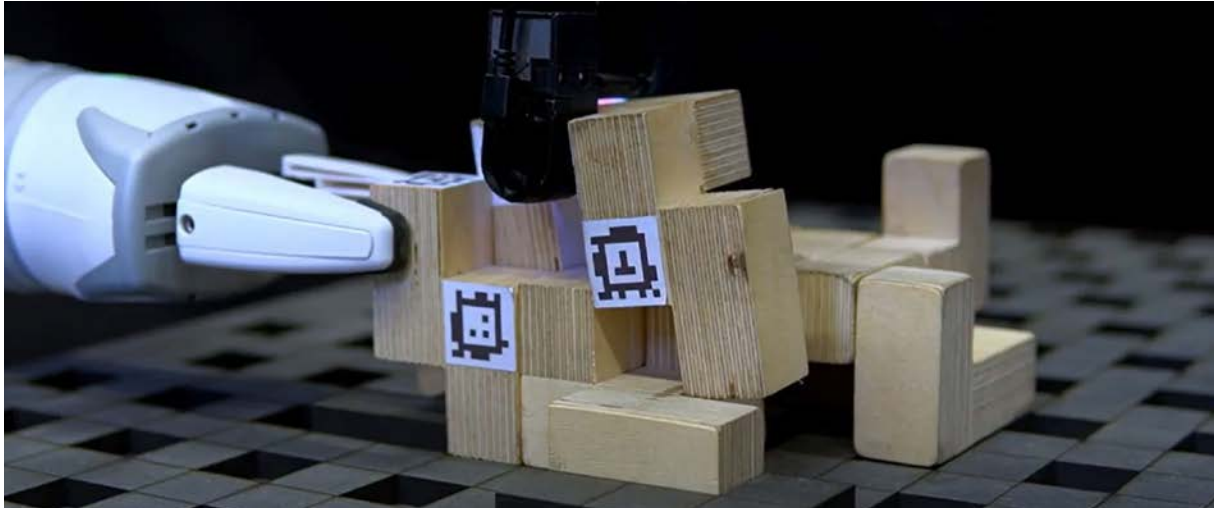


Quelle: Team DDU&IAS, TU Darmstadt 2023

4.5 Low-Level-Robotersteuerung für das Einsetzen von SL-Blöcken unter Verwendung von taktilen Feedback und Verstärkungslernen

Abbildung 47

Eine Montageaufgabe, die von einem Team aus zwei Robotern durchgeführt wird: Während ein Greifer die Bauelemente an Ort und Stelle hält, fügt der zweite Greifer, der mit taktilen Sensoren ausgestattet ist, einen SL-Block ein.



Quelle: Team DDU&IAS, TU Darmstadt 2022

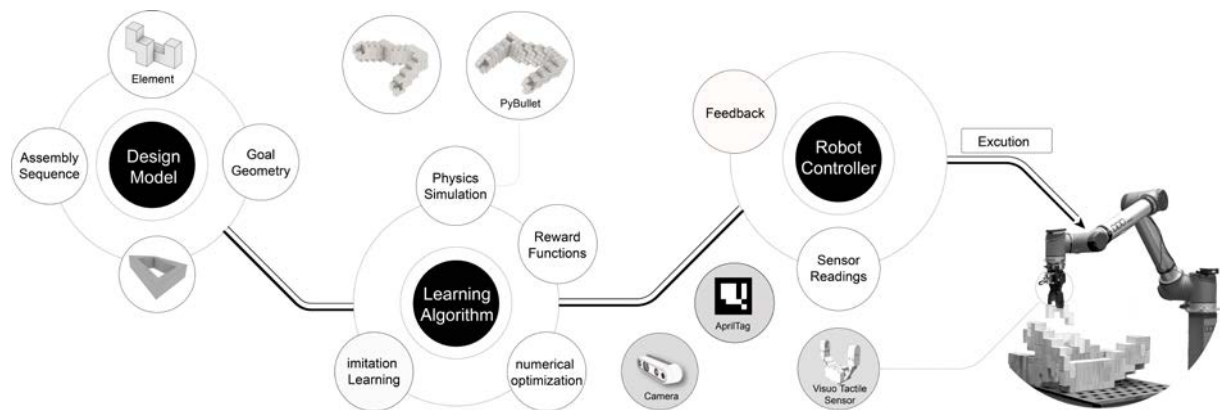
Taktiler Sensorik ist besonders wichtig im Kontext von manipulations- und montageintensiven Aufgaben. Im vorherigen Abschnitt wurden die grundlegenden Fähigkeiten der taktilen Sensorik für visuelle taktile Sensorik erforscht und entwickelt. In diesem Abschnitt wird ein Anwendungsszenario betrachtet, in dem das Tastgefühl erforderlich ist, um eine kontaktreiche Konstruktionsaufgabe zu bewältigen, bei der eine Struktur aus hölzernen SL-Blöcken zusammengesetzt wird.

Die Manipulation der SL-Blöcke wird von einem sogenannten Low-Level-Controller gehandhabt. Dieser Controller ist reaktiv und verantwortlich für das direkte Senden der Befehle an den Roboterarm und das Empfangen von Rückmeldungen. Im Gegensatz dazu ist ein High-Level-Controller für die Planung der gesamten Abfolge der Roboterbewegungen für die Montage verantwortlich, d.h. welcher Block aufgenommen und wo er platziert wird. Außerdem werden Kollisionen vermieden. Der High-Level-Controller wird in einem nachfolgenden Abschnitt beschrieben.

Zunächst wird das unüberwachte Repräsentationslernen von visuellen taktilen Sensoren für die Robotersteuerung bei komplexen Montageaufgaben untersucht. Wir betrachten die Verfügbarkeit von sofort einsatzbereiten physischen Lernumgebungen und Daten aus physischen Systemen als einen Schlüsselfaktor für die

Abbildung 48

Die Gesamtarchitektur des entwickelten autonomen modularen Montageansatzes. Im Design Modell (links) ist die SL-Block Geometrie, die Fügesequenzen und die Bauteilhierarchien hinterlegt. Der Lernalgorithmus (Mitte) umfasst Bauteil- und Fügesimulationen, Belohnungsfunktionen für verstärkendes Lernen, Imitationslernen und numerische Optimierung. Die Robotersteuerung (rechts) beinhalten die physische Ausführung der robotischen Fügung, Sensoren und Feedback.



Quelle: Team DDU&IAS, TU Darmstadt 2022

Forschung im Bereich der taktilen Steuerung. Daher wurde ein Versuchsaufbau entwickelt, der die schnelle Implementierung von Gym RL-Umgebungen auf einem UR10-Roboter mit Robotis-Greifer und DIGIT-Tastensensoren, kombiniert mit OptiTrack und/oder Apriltag-Tracking ermöglicht.

Der Versuchsaufbau sowie eine anspruchsvolle kontaktreiche taktiler robotische Konstruktionsaufgabe werden im Folgenden vorgestellt. Mit einem umfangreichen Datensatz aus dem zuvor beschriebenen Aufbau lernte das System, taktile Repräsentationen auf eine unüberwachte Weise zu extrahieren. Ein CNN-Autoencoder durchlief ein Training, um die Bilder des Sensors zu kodieren.

Durch zahlreiche Lernexperimente wurden wichtige Erkenntnisse über die Schlüsselaspekte des erfolgreichen Erlernens taktiler Repräsentationen, wie Netzwerkarchitektur und Hyperparameter, gewonnen. Diese Repräsentationen wurden dann verwendet, um das Transitionsmodell unserer multimodalen Umgebung zu lernen. Verschiedene Sequenzvorhersagemodelle für die Planung in Bezug auf ein- und mehrschrittige Vorhersagen wurden verglichen. Abschließend wurde unser erlerntes Modell mit einem CEM-Planer kombiniert, um modellprädiktive Steuerung durchzuführen.

4.5.1 Das Assembly Gym Framework

Das Einrichten der Laborumgebung für Robotik-Lernexperimente ist eine umfangreiche Aufgabe. Eine physische Umgebung mit einem echten Roboter muss mit mehreren Hardwarekomponenten verbunden werden, die jeweils individuelle Treiber und Einrichtungen benötigen.

Infolgedessen neigen Forscher dazu, in der Simulation zu bleiben oder nur eine minimale Aufgabe einzurichten, die schnell hartkodiert werden kann. Um dieses Problem zu überwinden und die Forschung im Bereich der robotergestützten Montage zu ermöglichen, wurde das Assembly Gym eingerichtet.

Das Assembly Gym Framework ist eine Softwareebene, die die modulare Montage als ein Reinforcement Learning Problem darstellt, und einerseits ermöglicht, die Anwendung existierender RL Algorithmen auf konkrete Montageprobleme anzuwenden, und andererseits die Verwendung eines beliebigen Simulators oder realen Roboters als "Umgebung", in der der RL Algorithmus agiert.

Das Assembly Gym Framework ermöglicht die schnelle Definition neuer Gym RL-Umgebungen, mit denen anschließend in der Entwicklung von Steuerungslösungen experimentiert werden kann. Zur Ermöglichung einer Übernahme dieses Setups werden die Architektur und die Schlüsseldesignkonzepte vorgestellt.

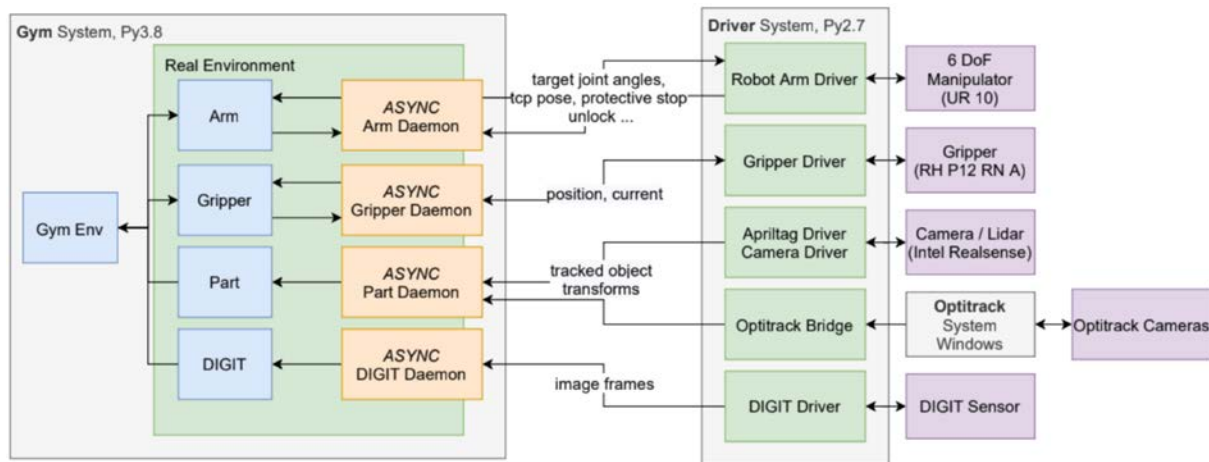
Hardware-Komponenten

Das physische Setup besteht aus dem Roboter-Manipulator, den taktilen Sensoren und dem Trackingsystem:

- **Roboter-Manipulator:** Ein 6DoF UR10 Roboterarm mit einem Robotis RH-P12-RN-A Greifer am Endeffektor. Der Robotis RH-P12-RN-A wurde verwendet, da er eine Low-Level-Control-API bietet, die das Aktivieren des Greifers mittels Strom ermöglicht, was beim Anbringen taktiler Sensoren daran notwendig ist. Die Greifstärke muss entsprechend eingestellt werden, damit die Silikongreiffläche des taktilen Sensors nicht beschädigt wird.
- **Taktile Sensoren:** Für das DIGIT-Design wurde sich entschieden, da es Open Source ist und kostengünstig hausintern hergestellt werden kann. Da es hausintern hergestellt wird, ist es möglich, Verbesserungen vorzunehmen und es an spezifische Bedürfnisse anzupassen.
- **Trackingsystem:** Die Montage, welche die Manipulation mehrerer Objekte zur Zusammenstellung einer Struktur beinhaltet, erfordert in der Regel eine Form der Objektlokalisierung. Für diesen Zweck bietet das Setup zwei Trackingsysteme: eine RGB-Kamera in Kombination mit AprilTag-Fiducial-Markern sowie OptiTrack – ein präzises Bewegungserfassungssystem, basierend auf Infrarotkameras und Reflektormarkern.

Abbildung 49

Software-Design der Assembly Gym Umgebung. Komponenten kommunizieren asynchron mit den Gerätetreibern.



Quelle: Team DDU&IAS, TU Darmstadt 2022

Softwarearchitektur

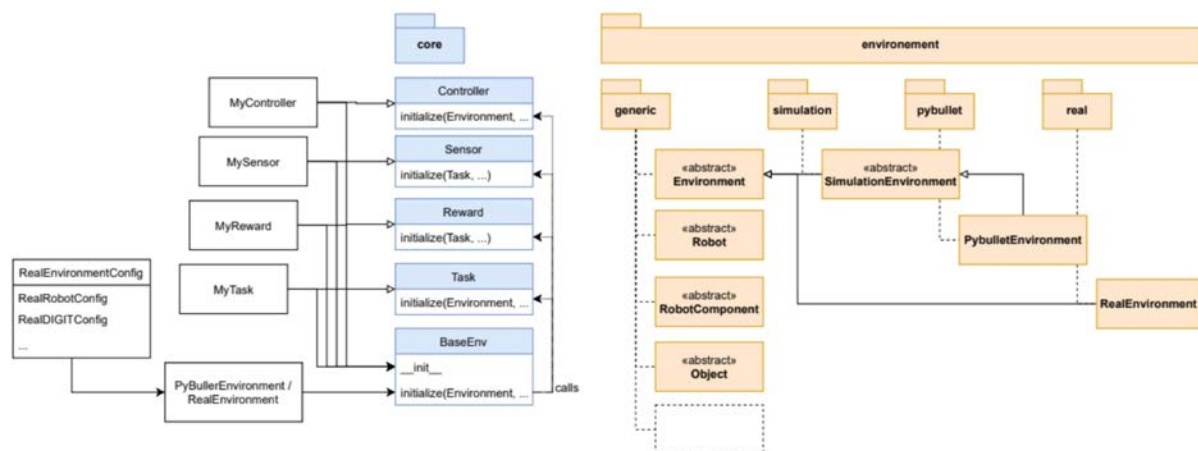
Aus technischen Gründen wurde die Implementierung in zwei Codebasen aufgeteilt, die in diesem Fall auch auf unterschiedlichen Maschinen laufen (vgl. Abbildung 49). Eine Codebasis umfasst den gesamten Python-Code, der die Benutzeroberflächen implementiert, und eine zweite enthält den gesamten Back-End- und Treiberbene-Code im ROS-Framework.

Das Gym Framework, der benutzerorientierte Teil des Assembly Gyms, kann von Forschern genutzt werden, um schnell eine neue Gym RL-Umgebung zu implementieren. Bei diesem Gym-Stil wird nicht strikt die OpenAI Gym API nachgebildet, sondern es orientiert sich an der gängigen Step: Aktion → (Beobachtung, Belohnung, Fertig, Info) und Reset: → Beobachtungsschnittstelle. Da das Framework auf einem physischen Setup basiert, existiert kein Anwendungsfall als Benchmark-Umgebung.

Die Umgebung (Gym Env) bietet die Schnittstellen und das Verhalten einer typischen RL-Umgebung, während die Umgebung das Back-End ist, auf dem dieses Setup basiert. Ein solches Back-End könnte ein Simulator (z. B. PyBullet oder Mujoco) oder Schnittstellen zu einem tatsächlichen physischen Laboraufbau sein.

Abbildung 50

Kernkomponenten des Assembly Gym Frameworks. Weiße Kästen sind Objekte, die im Benutzercode instanziiert werden.



Quelle: Team DDU&IAS, TU Darmstadt 2022

Das Software-Design folgt dem Kompositionsmuster. Die Umgebung wird als Hierarchie von Komponenten instanziiert, wodurch die Struktur der Komponenten auf der höchsten Ebene sichtbar wird. Die BaseEnv wird in die folgenden abstrakten Komponenten zerlegt (vgl. Abbildung 50):

- Controller, Sensoren, Belohnung, Aufgabe:** Controller, Sensor, Belohnung und Aufgabe werden von der BaseEnv aufgerufen und erhalten die Umgebung, in der sie operieren. Der Controller empfängt eine Aktion und die Umgebung führt diese Aktion aus, indem sie die Umgebung aufruft. Die Sensoren und die Belohnung erhalten die Aufgabe und extrahieren und geben Beobachtungen und eine Belohnung zurück. Die Aufgabe ist die komplexeste Komponente, da sie das aufgabenspezifische Verhalten definiert, wie das Beenden der aktuellen Episode oder das Halten eines Zustands, der zur Berechnung der Belohnungsfunktion benötigt wird, indem sie eine Schritt- und Reset-Funktion implementiert.
- Serialisierbare Globale Top-Level-Konfiguration:** Alle wichtigen Konfigurationsparameter werden auf der höchsten Ebene im Benutzercode definiert und dann durch die Komponentenhierarchie weitergegeben. Das Konzept einer globalen zentralisierten Konfiguration ist insbesondere im Forschungsbereich wichtig. Oft ist der Code Teil des Experiments und Parameter müssen angepasst werden, um Ergebnisse zu erzielen. Wir möchten sicherstellen, dass der Benutzer alle Parameter, die das Verhalten der Umgebung beeinflussen, an einem Ort überblicken kann. Indem unbeabsichtigte Fehlkonfigurationen verhindert werden, kann der Prozess der Parameteranpassung beschleunigt werden. Während des Experimentierens möchte man die Parameter der Umgebung zusammen mit den Experimentergebnissen protokollieren, um Kontext und Reproduzierbarkeit zu gewährleisten. Unsere globale Top-Level-Konfiguration ist daher serialisierbar.
- Reale und Simulationsumgebung:** Als abstrakte Komponente konzipiert, kann die Umgebung mehrere Implementierungen haben, die austauschbar verwendet werden. In diesem Fall wurde eine RealEnvironment basierend auf dem realen physischen Setup im Labor und eine SimulationEnvironment basierend auf PyBullet implementiert. Aufgrund der Herausforderungen in der Simulation der DIGIT-Sensoren wurde zunächst das physische Setup genutzt. Während die SimulationEnvironment mit der PyBullet-Physik-Engine interagiert, verbindet sich die RealEnvironment über ROS mit den Treibern der verschiedenen Hardwarekomponenten.

4.5.2 Robotische Konstruktionsaufgabe

Eine vereinfachte Aufstellung der robotergestützten Montageaufgabe von SL-Blöcken wird betrachtet. Eine bereits montierte Struktur ist auf dem Tisch befestigt und ein neuer SL-Block wird vom Roboter in die Struktur eingefügt. Die Umgebung liefert die verfolgte Position des SL-Blocks.

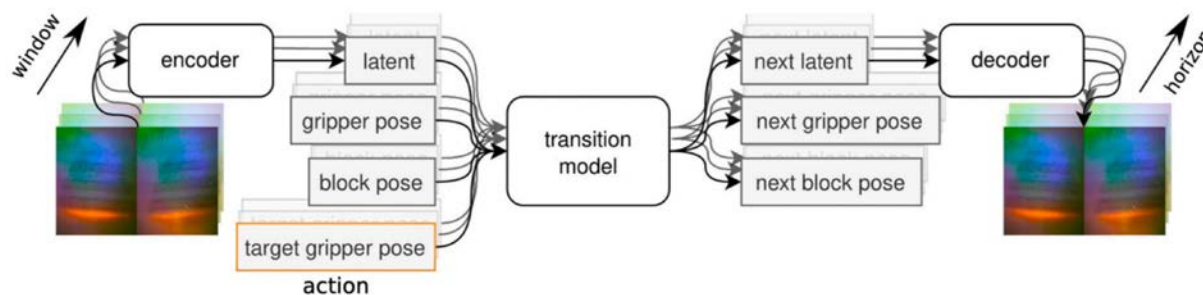
Alle Positionen in der Umgebung, die Position des Greifers sowie die Position des SL-Blocks, werden relativ zur Zielposition, der montierten Position des SL-Blocks gegeben, sodass die Aufgabe als Bewegen des SL-Blocks zum Ursprung dargestellt wird.

Die Belohnungsfunktion ist der Abstand des SL-Blocks zum Ursprung entworfen, den es zu minimieren gilt.

- **Belohnung:** Die Belohnung r ist die negative Distanz d zwischen der Position des bewegenden Blocks p und der Zielposition p^* . Da alles in den Rahmen der Zielposition transformiert wird, ist das Ziel der Ursprung $p^* = (t_0, r_0)$. Der Abstand zwischen den Positionen ist eine gewichtete Summe der Übersetzung und Rotationsdistanz $d(p, p^*) = \alpha d(p, p^*) + (1 - \alpha) d(p, p^*)$, die zwei Werte mit unterschiedlichen Einheiten kombiniert. Da Übersetzungsdistanzen im Bereich von Zentimetern und Rotationen in Radianten liegen, setzen wir $\alpha = 0,9$.
- **Aktionsraum:** Ein 6DoF-Roboter-Manipulator bietet mehrere Steuerungsmodi. Die Steuerung im kartesischen Raum wird über die Gelenkraumsteuerung gewählt. Das Beschränken und Erkunden des Aktionsraums wird durch die Steuerung im kartesischen Raum erleichtert, da diese einfacher zu visualisieren und zu verstehen ist als die Gelenksteuerung. Ursprünglich wurde eine Kraft- oder Geschwindigkeitssteuerung im kartesischen Raum angestrebt, jedoch konnte keine Echtzeitsteuerungsleistung mit dem vorhandenen Aufbau erreicht werden, sodass auf die Positionssteuerung zurückgegriffen werden musste. Intuitiv erscheint die Kraftsteuerung im kartesischen Raum oder die Drehmomentsteuerung als der geeignetste Steuerungsmodus für die Aufgabe. Die Drehmomente und Kräfte im kartesischen Raum, die vom Manipulator ausgeübt werden, korrelieren direkt mit dem taktilen Feedback.
- **Beobachtungsraum:** Der Beobachtungsraum ist multimodal, da er aus Bildern der DIGITs und Positionen des Greifers und des SL-Blocks besteht. Alle Positionen, die aus der Umgebung kommen, sind 7-dimensionale Vektoren $p_x, p_y, p_z, \omega_x, \omega_y, \omega_z, \omega_w$. Die Bilder vom DIGIT-Sensor sind dreikanaliges RGB mit einer Auflösung von (w: 240, h: 320).
- **Zurücksetzen:** Ein vollständig autonomes Zurücksetzen der Umgebung wird gewünscht, wenn ein großer Datensatz gesammelt wird. Das Verfahren zum Zurücksetzen der Aufgabe beinhaltet das Aufheben des SL-Blocks, entweder durch Herausziehen aus der Struktur, wenn die letzte Episode erfolgreich verbunden wurde, oder durch Aufheben vom Tisch. Da keine Umsetzung des Neugreifens vorhanden ist, wird der Mensch, der den Prozess überwacht, benachrichtigt, falls der SL-Block in eine unerreichbare Position gerollt ist, sodass eine manuelle Neuausrichtung erfolgen kann. Bei den meisten wiederholten Trajektorien kommt es zu einem erfolgreichen Verbinden des Blocks, wodurch die menschliche Interaktion minimal bleibt. Die Übergangsfunktion $t: (o_n, a_n) \rightarrow o_{n+1}$ wird durch ein neuronales Netzwerk approximiert, basierend auf Datenpaaren (o_n, a_n, o_{n+1}) , die aus der n -ten Beobachtung o_n , der Aktion a_n und der Beobachtung des nächsten Zeitpunkts o_{n+1} die aus dem System abgetastet wurden, bestehen. Da das Einspeisen eines vollständigen Bildes in ein vollständig verbundenes Netzwerk nicht machbar ist, müssen die Bilder durch ein Encoder-Netzwerk auf eine geringere Dimensionalität komprimiert werden, wofür ein konvolutionelles neuronales Netzwerk verwendet wird. Die resultierende Netzwerkarchitektur und das Trainingsszenario sind in Abbildung 51 dargestellt.

Abbildung 51

Die Architektur des Modells, das aus den vom Roboter gesammelten realen Daten gelernt wurde, ist so gestaltet, dass das Modell als Eingabe ein Fenster vergangener Beobachtungen erhält, welches in der Abbildung durch den in die Tiefe des Diagramms gehenden Schatten dargestellt wird. Jede Beobachtung ist multimodal und setzt sich aus drei Teilen zusammen: (i) zwei taktile Bilder (linker und rechter Greiferfinger), (ii) Greiferpose, (iii) Posen der manipulierten Blöcke. Rohsensorbeobachtungen durchlaufen einen lernbaren Encoder, bevor sie in das Übergangsmodell gelangen. Gegeben durch ein Fenster solcher Beobachtungen zusammen mit einem Fenster von Aktionen, gibt das Übergangsmodell ein Fenster zukünftiger Beobachtungen zurück, das in diesem Fall als Horizont bezeichnet wird.



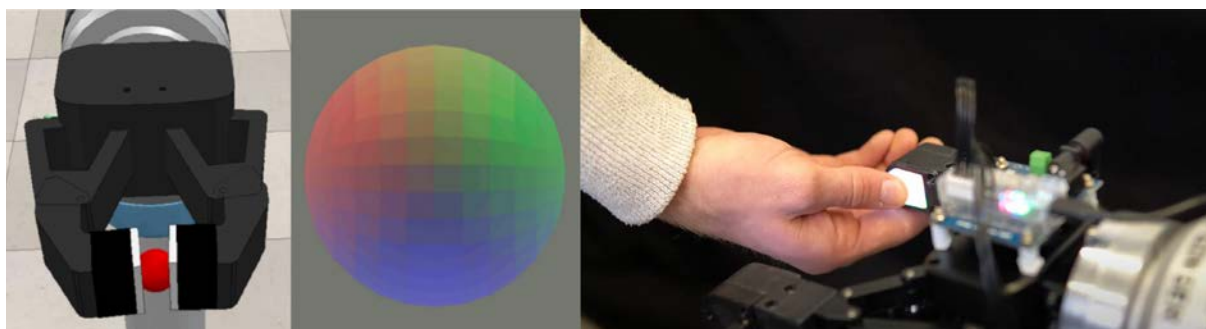
Quelle: Team DDU&IAS, TU Darmstadt 2022

4.5.3 Repräsentation taktiler Daten des DIGIT Sensors

Um Reinforcement Learning in der Simulation zu ermöglichen, muss der DIGIT-Sensor in PyBullet integriert werden. Es wurden zwei Ansätze in Betracht gezogen. Der erste basiert auf der Simulation von Lichtausbreitung und Reflexion. Während diese Methode für kleine Objekte wie die kleine Kugel in Abbildung 52 oder die Würfel in ansprechend ist, funktioniert sie nicht gut für Objekte, die die gesamte Oberfläche des Sensors bedecken, wie es bei unseren Blöcken der Fall ist. Wenn sich große Objekte im Greifer verschieben, kann diese Art der Simulation kein Signal erfassen, während der echte Sensor ein Signal erzeugt, weil sich das Silikonengel an der Fingerspitze verformt. Der andere Simulationsansatz basiert auf einer physikalischen Simulation mit weichen Körpern.

Abbildung 52

Links: Simulation des DIGIT Sensors in PyBullet. Die synthetisch generierten Bilder sind nicht realistisch genug, um eine Transition gelernter Modelle auf das echte System zu ermöglichen. Rechts: das reale System. DIGIT Sensoren sind an den Fingerspitzen des Roboters angebracht.



Quelle: Team DDU&IAS, TU Darmstadt 2022

Moderne Physiksimitoren unterstützen noch keine zuverlässige und realistische Weichkörpersimulation, wie sie für die Robotersteuerung erforderlich ist. Zwar war man in diesem Projekt in der Lage, die Weichkörpersimulation in PyBullet so abzustimmen, dass sie dem realen Sensor ähnelt, aber dies erforderte sehr kleine Zeitschritte und folglich einen sehr hohen Rechenaufwand, was den Ansatz für die Verwendung mit RL unpraktisch macht. Aufgrund der oben erwähnten Schwierigkeiten bei der Simulation von visuellen taktilen Sensoren kann eine in der Simulation erlernte Strategie nicht direkt auf dem realen Roboter ausgeführt werden. Um die Simulation mit der Realität in Einklang zu bringen, wurde ein auf einem neuronalen Netz basierendes Simulationsmodell direkt mit den Bilddaten des Sensors trainiert, die der reale Roboter beim Zusammenbau der SL-Blöcke erhielt.

Verarbeitung taktiler Daten mittels eines vortrainierten Autoencoder CNN Modells

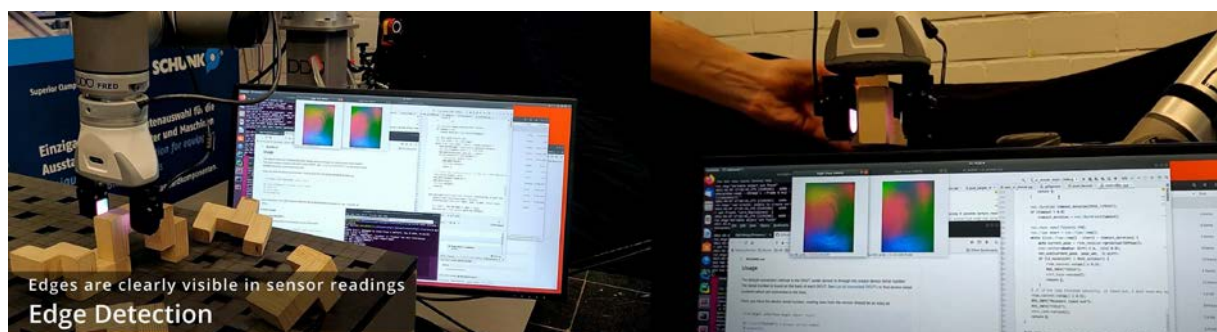
Der Kodierer und der Dekodierer der taktilen Bilder werden gemeinsam als statischer Autoencoder trainiert. Die linken und rechten DIGIT-Bilder werden in eine latente Repräsentation mit einem Autoencoder kodiert, der anhand der SSIM-Verlustgleichung trainiert wird. Der Encoder besteht aus mehreren CNN-Blöcken und einem vollvernetztem Neuronalem Netz, das die Merkmale des letzten CNN-Blocks aggregiert und in die latente Dimension komprimiert.

Der Decoder ist spiegelbildlich zum Encoder aufgebaut und verwendet transponierte CNN-Blöcke. Ein CNN-Block ist eine Faltung, gefolgt von einer Batch-Normalisierung und einer Aktivierungsfunktion. Da die beiden DIGIT-Sensorbilder tatsächlich eine Messung zu einem bestimmten Zeitpunkt darstellen, ist es intuitiv sinnvoll, sie gemeinsam zu kodieren. Aufgrund von Herstellungsfehlern sehen die Bilder der DIGITs jedoch unterschiedlich aus, selbst wenn sie den gleichen Zustand messen. Daher ist es auch sinnvoll, spezifische Parameter für linke und rechte DIGITs zu lernen.

Wir betrachten mehrere Modelle der getrennten und gemeinsamen automatischen Kodierung von DIGIT-Bildern, die in Abbildung 53 dargestellt sind und die sich in der Menge der gemeinsam genutzten Parameter unterscheiden. Wir stellen die Hypothese auf, dass die gemeinsame Kodierung der Bilder zu einer kosteneffizienteren Kodierung führt, die auch für die Aufgabe besser geeignet ist.

Abbildung 53

Fotografie des experimentellen Montagesetups. Taktile Messungen des Kontakts des DIGIT Sensors mit hölzernen SL-Blöcken. Die resultierenden Bilder sind auf dem Bildschirm dargestellt.



Quelle: Team DDU&IAS, TU Darmstadt 2022

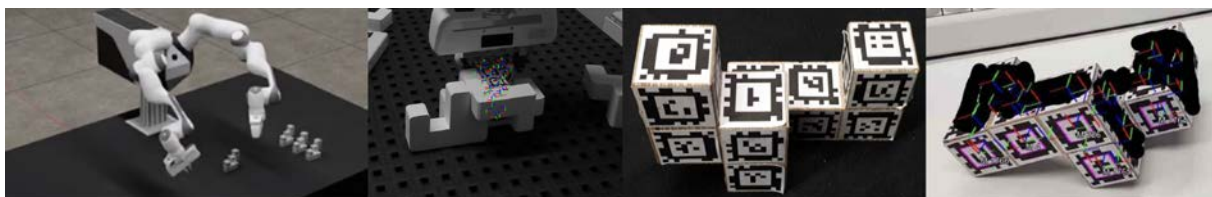
4.5.4 Datenerfassung aus menschlichen Demonstrationen

Die anfänglichen Daten für das Training einer Montagestrategie werden aus menschlichen Demonstrationen gesammelt. Für das Verfolgen der SL-Blöcke werden AprilTags verwendet. AprilTag ist ein visuelles Referenzsystem. Die Tags können mit einem gewöhnlichen Drucker erstellt werden, und die AprilTag-Erkennungssoftware berechnet die genaue 3D-Position, Ausrichtung und Identität der Tags relativ zur Kamera. Die Positionen mehrerer Tags werden gemittelt, um die Positionsschätzung des SL-Blocks zu verbessern. Die Verfolgung und Erkennung von Objekten ist ein aktives Forschungsgebiet mit vielen verschiedenen Ansätzen. Diese Ansätze können hauptsächlich nach der Art der Daten und der daraus resultierenden Dimension der zur Hypothesenbildung über das Objekt verwendeten Daten kategorisiert werden. Je mehr Dimensionen, desto mehr Informationen stehen zur Verfügung, um eine qualifizierte Schätzung über den Standort und die mögliche Orientierung des untersuchten Objekts zu bilden. Es gibt Ansätze, die auf Computervision basieren, sowie nicht-visuelle Ansätze. In [Pfanne et al, 2018] wurde ein nicht-visueller Ansatz verwendet, um die Objektposition allein durch die Auswertung der Gelenkmessungen einer Roboterhand, die das Objekt hält, zu verfolgen. Dabei wurde jedoch festgestellt, dass die Verwendung nur der Gelenkmessungen zu signifikanten Abweichungen bei der Schätzung der Objektposition führt. Daher integrierten sie ein auf Computervision basierendes Erkennungssystem, um es mit den zuvor gewonnenen Gelenkwinkeln zu kombinieren und so eine gute Schätzung der Objektposition zu erzielen. Dies zeigt, dass für präzise Vorhersagen der Positionen manipulierter Objekte mehr als nur Roboter-Gelenkinformationen benötigt werden. Ein auf Computervision basierender Objektverfolgungsansatz wird von Pauwels et al. [Paulwels and et al, 2015] verwendet. Sie nutzen eine RGB-D-Kamera, um Tiefeninformationen zu extrahieren und eine 3D-Simulation der Szene zu aktualisieren. Die Simulation wird dann verwendet, um die Positionsschätzung zu bestimmen.

Eine einfache, aber robuste Alternative besteht darin, die zu erfassenden Objekte mit fiduzialen Markern zu versehen. Aufgrund ihrer hohen Erkennungsraten selbst bei schlechten Lichtverhältnissen, der eingebauten Positionsschätzung für die Marker und des fehlerresistenten Designs sind fiduziale Marker wie AprilTag beliebte Methoden zur Objektverfolgung und werden daher auch für unser Projekt verwendet. Kürzlich haben markerbündelbasierte Objektverfolger eine bemerkenswerte Genauigkeit bei der Positionsschätzung gezeigt. In Sarmadi et al. [Sarmadi et al., 2019] wurde ein gemeinsamer Ansatz für die Kamerakalibrierung, die Schätzung der relativen Transformationen der Marker und die Schätzung der Referenzperspektivtrajektorie der Marker vorgestellt. Sie verwendeten ein Multi-Kamera-Setup mit teilweise überlappenden Sichtfeldern (FOVs), Objekten mit angebrachten Markerbündeln und Minimierung des Reprojektionsfehlers, um dies zu erreichen. In [Trinh et al., 2022] wird eine ähnliche Technik verfolgt. Anstatt mehrere Kameras und allgemeine Mehrmarker-Objektverfolgung zu verwenden, konzentrieren sie sich auf die Verfolgung einer einzelnen dodekaedrischen Manipulatoranbringung. Marker werden auf dessen Oberfläche platziert, um sicherzustellen, dass mehrere gleichzeitig aus dem Sichtfeld der Kamera sichtbar sind. Sie kalibrieren die Kameras, erkennen dann die Transformationen zwischen den Markern und optimieren diese. Während des Betriebs werden die Kalibrierungs- und Transformationsschätzungen verwendet, um die Position des gewählten Referenzmarkers von einer einzelnen Kamera

Abbildung 54

Die Position der SL-Blöcke wird mit Hilfe von AprilTags verfolgt. Experimente mit unterschiedlichen Mengen an AprilTags wurden durchgeführt. Das Mitteln über mehrere Tags verbessert die Schätzgenauigkeit, ist jedoch in der Praxis einschränkend, da jeder Block mit vielen Tags versehen werden muss. Daher wurden zwei bis vier Tags pro Block als ausreichend erachtet, um eine stabile Verfolgungsleistung zu erreichen.



Quelle: Team DDU&IAS, TU Darmstadt 2023

aus zu verfolgen. Beide Arbeiten erreichen die Verfolgung von Markern, auch wenn sie zu diesem Zeitpunkt nicht sichtbar sein mögen, indem sie andere Markerpositionen aus den sichtbaren Markerpositionen unter Verwendung ihrer optimierten paarweisen Transformationen schätzen. Um eine einheitliche Schätzung aller gewünschten Markerpositionen zu verfeinern, minimieren beide Arbeiten einen Reprojektionsfehler - den mittleren quadratischen Fehler über die Unterschiede der geschätzten Markertransformationen zu den erkannten Markertransformationen. Eine Multi-View, Multi-Marker-Positionsbestimmungstechnik wurde entwickelt, aufbauend auf den Erkenntnissen aus der jüngsten Forschung. Dieser Ansatz konzentriert sich auf die Minimierung des Reprojektionsfehlers, um eine genaue und zuverlässige Schätzung der Position zu erreichen. Bei dieser Methode wird die Optimierung der geschätzten Position eines Referenzmarkers zu einem bestimmten Zeitpunkt erreicht, indem der quadratische Fehler zwischen den in Bildern erkannten Markerpositionen und den unter Verwendung bekannter Transformationen vom Referenzmarker zu anderen Markern berechneten Positionen minimiert wird. Dieser Optimierungsprozess wird über alle verfügbaren Kameraperspektiven durchgeführt, wobei die resultierenden Fehler aus jeder Ansicht aggregiert werden, um eine Gesamtschätzung der Position zu bilden. Im Gegensatz zu früheren Arbeiten werden Reprojektionsfehler bei dieser Methode auf der Bildebene der Kameras berechnet, anstatt direkt auf den 3D-Schätzungen. Dieser Ansatz entspricht der traditionellen "Bundle-Adjustment"-Fehlerfunktion, die in der Photogrammetrie verwendet wird. Intuitiv bietet diese Methode den Vorteil, dass Änderungen in der Entfernungsschätzung für eine bestimmte Kameraperspektive einen geringeren Einfluss auf den Gesamtfehler haben, verglichen mit Korrekturen der in-plane-Übersetzung in anderen senkrechten Ansichten.

Für jeden Verbindungstyp gibt der Mensch einige Demonstrationen, die dann vom Roboter als Initialisierung des Selbstverbesserungsprozesses verwendet werden. Während der Demonstrationen wird nur die Trajektorie des SL-Blocks aufgezeichnet. Der Roboter versucht anschließend, derselben Trajektorie zu folgen, jedoch muss sich der Roboter, sobald die Blöcke in Kontakt kommen, auf die taktilen Signale verlassen, um die Politik anzupassen. Daher entstehen bei mehrmaligem Versuch derselben Aufgabe jedes Mal neue Bedingungen, und so wird ein Datensatz für das Training gesammelt.

Um die Übergangsfunktion des Systems zu erlernen, muss ein umfangreicher Datensatz für das Offline-Lernen gesammelt werden. Diese Daten sollten so viel wie möglich des relevanten Zustandsraums abdecken. In unserem Fall kann ein solcher Datensatz nicht durch zufällige Aktivierungen des Systems gesammelt werden, da wir sonst keine Daten von den Zuständen während des Fügens erhalten würden, wenn der Block beginnt, in seine Einbauposition zu gleiten. Die Daten werden aus einer Reihe von aufgezeichneten Handdemonstrationen gesammelt (vgl. Abbildung 55), die dann vom Roboter nachgefahren werden. Um die Varianz der Daten zu erhöhen, ohne zusätzliche Demonstrationen aufzeichnen zu müssen, werden die Aufnahmen verrauscht und auch interpoliert, um neue Trajektorien zu erhalten. Eine handaufgezeichnete Trajektorie ist eine Abfolge von SL-Block-Positions-Messungen.

Abbildung 55

Eine manuelle Demonstration, wie zwei SL-Blöcke verbunden werden müssen. Die Positionen der Blöcke werden mit AprilTags und Optitrack verfolgt.



Quelle: Team DDU&IAS, TU Darmstadt 2023

4.5.5 Steuerung auf dem realen System (Implementierung)

Modellfreies verstärkendes Lernen wurde bereits erfolgreich zur Lösung von taktilen Montageaufgaben eingesetzt [Spector und Castro, 2021]. Dabei kamen in der Forschung jedoch bisher selten hochauflösende visuell taktile Sensoren zum Einsatz. Gebräuchlicher ist modellbasiertes verstärkendes Lernen in Kombination mit tiefer modellprädiktiver Steuerung. Hier kommen taktile Steuerungsstrategien zum Einsatz, die auf Basis der Daten aus hochauflösenden visuellen taktilen Sensoren lernen [Lambeta, et al., 20205, Tian et al., 201910].

Im Projekt nutzen wir deshalb die Methode der tiefen modellprädiktiven Steuerung. Durch die Zustandsübergangs- oder Dynamikfunktion eines Systems kann eine Trajektorie zukünftiger Aktionen gemäß einer Kostenfunktion optimiert werden. Die tiefe modellprädiktive Steuerung wendet dieses Konzept auf das verstärkende Lernen an.

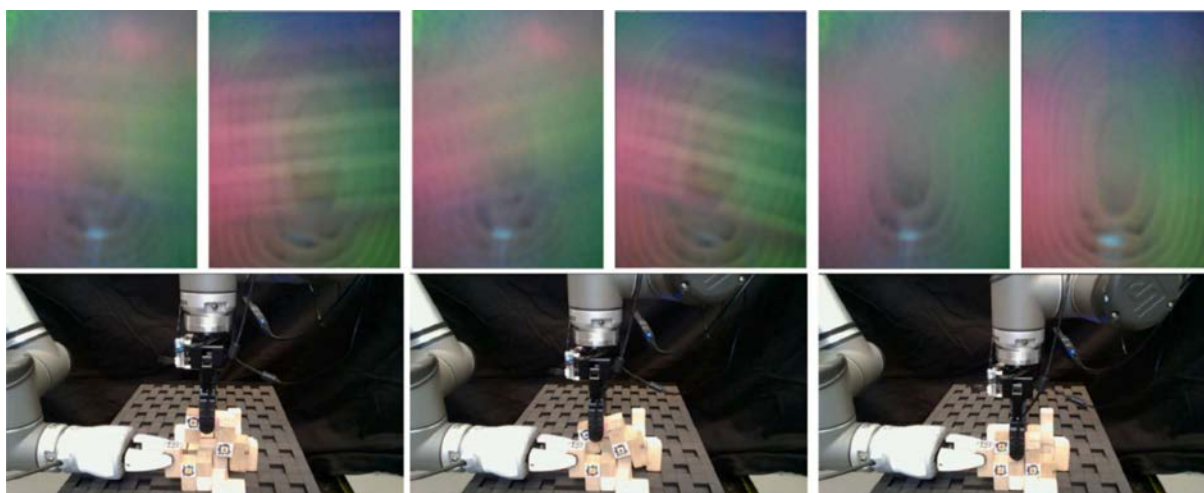
Sie lernt eine Annäherung an die Zustandsübergangsfunktion aus Datenpaaren (s_t, a_t, s_{t+1}) , die aus der Umgebung gesammelt werden. Unter Verwendung dieses Übergangsmodells und unter der Annahme, dass die Belohnungsfunktion bekannt ist, kann der Ertrag oder die Kosten einer Trajektorie zukünftiger Aktionen (a_t, \dots, a_{t+h}) geschätzt und optimiert werden, was zu einer optimalen Steuerung führt. Falls die Belohnungsfunktion unbekannt ist, kann sie ebenfalls aus Daten aus der Umgebung gelernt werden.

Während der vom Roboter durchgeführten Versuche werden alle Datenströme aufgezeichnet: taktile Bilder, Objektpositionen, Greiferposition und Roboterzustand. All diese Signale werden als Zeitreihendaten verwendet, um ein prädiktives Modell des Montageprozesses zu erlernen.

Von Datenpaaren (o_n, a_n, o_{n+1}) , die aus der n -ten Beobachtung o_n , der Aktion a_n und der Beobachtung des nächsten Zeitpunkts o_{n+1} , gesammelt aus dem System, besteht, wird die Übergangsfunktion $t: (o_n, a_n) \rightarrow o_{n+1}$ durch ein neuronales Netzwerk angenähert. Da das Einlesen eines vollständigen Bildes in ein vollständig verbundenes Netzwerk nicht durchführbar ist, müssen die Bilder zu einer niedrigeren Dimensionalität durch ein Kodierernetzwerk komprimiert werden, für das ein konvolutionales neuronales Netzwerk verwendet wird. Das resultierende Trainingsszenario ist in Abbildung 56 dargestellt.

Abbildung 56

Während des Trainings der autonomen Montage aufgezeichnete Signale: taktile Bilder, Zeitreihendaten des Roboterzustands und des Blockzustands.



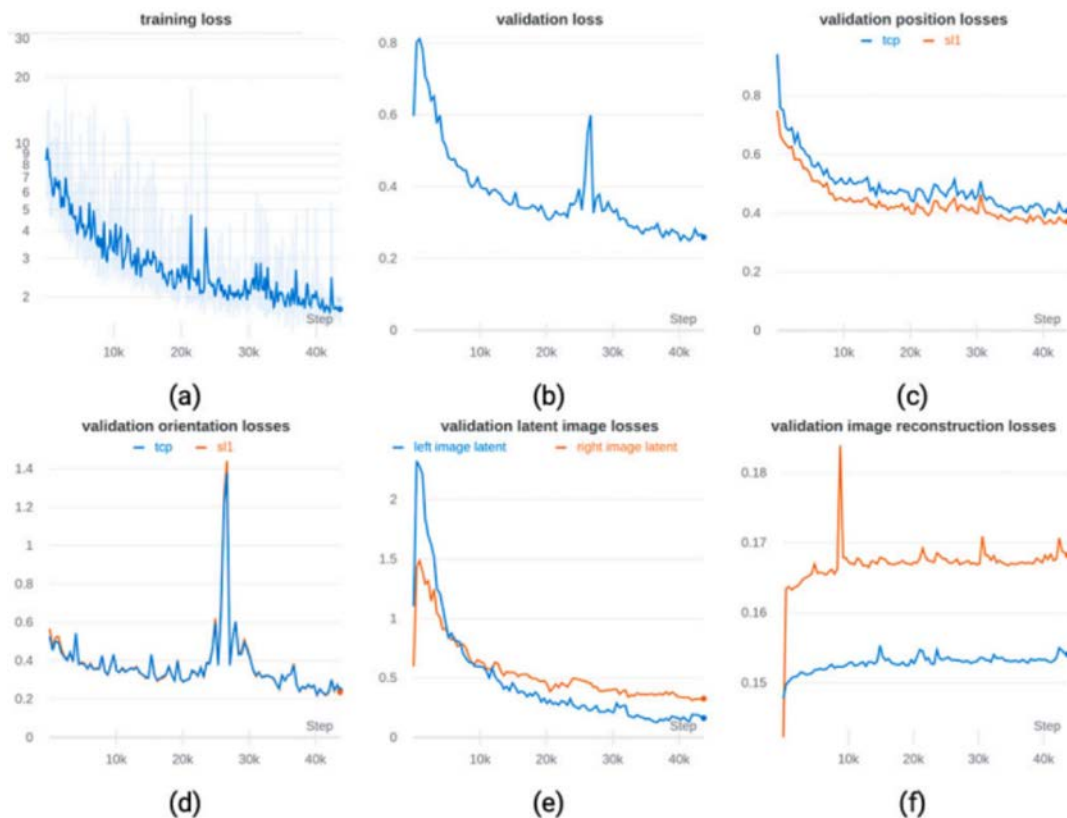
Quelle: Team DDU&IAS, TU Darmstadt 2023

In Abbildung 57 werden die Lernkurven für das Training des Übergangsmodells gezeigt. Ein Fenster von 7 Beobachtungen wurde als Eingabe in das Übergangsmodell eingespeist, und ein Horizont von 3 Zeitschritten wurde für die Vorhersage verwendet. Ein Autoencoder mit ReLU-Aktivierungen und einem 16-dimensionalen latenten Raum wurde in Chargen der Größe 16 mit einer Lernrate von 0,0002 trainiert. Obwohl es dem Modell gelingt, den Zustand des Roboters und der Objekte recht gut vorherzusagen, wie aus den Positionsverlusten in Abbildung 57 hervorgeht, ist der Fehler in der Vorhersage der taktilen Bilder sowohl im latenten Raum als auch nach der Rekonstruktion hoch. Dies könnte darauf zurückgeführt werden, dass der Agent nicht das gesamte Bild rekonstruieren muss, um aufgabenrelevante Informationen daraus zu extrahieren. Die zusätzlichen Informationen der anderen Sensoren konnten nicht genutzt werden, um Bilder gut vorherzusagen, obwohl alle anderen Vorhersagen der Realität ziemlich präzise entsprechen.

Die Steuerung auf dem realen System wird mit einem kartesischen Geschwindigkeitsregler implementiert. Das gelernte Modell der Umgebungsdynamik wird verwendet, um die Ausführung der Trajektorien vorherzusagen

Abbildung 57

Lernkurven während des Trainings des Übergangsmodells. Trainingsverlust (a) und verschiedene Validierungsverluste (b-f) sind in der Abbildung dargestellt. Die Konvergenz des Trainingsverlustes (a) zeigt, dass das Modell die Daten aus dem Trainingsdatensatz rekonstruieren kann. Die Validierungsverluste für Position (c) und Orientierung (d) zeigen ebenfalls einen konvergierenden Trend, sowie in der Vorhersage für „Tool Center Point (TCP, blaue Kurve)“ als auch für SL-Block-1 (SL1, orange Kurve). Der Anstieg in Abb. (d) gegen 25k Trainingsschritte deutet auf kurzzeitig schlechten Fit auf dem Validierungsdatensatz. Gleich danach springt der Validierungsverlust auf die abneigende Kurve und nimmt weiter ab. Die letzten zwei Teilabbildungen zeigen die Validierungsverluste für Bildrekonstruktion im latenten Raum (e) und im Bildraum (f). Es ist bemerkenswert, dass nur der Bildraumverlust (f) nicht konvergiert, und zusätzlich unterscheiden sich die Verluste für den rechten (orange) und den linken (blau) Sensoren. Jedoch im latenten Raum (e) nimmt der Verlust schnell ab. Das deutet auf die Komplexität der genauen Rekonstruktion der kompletten Bilder von dem Sensor hin. Doch für die Steuerung ist nur der latente Raum von Bedeutung, deswegen ist die insgesamt Konvergenz des Validierungsverlustes (b) ausreichend.



Quelle: Team DDU&IAS, TU Darmstadt 2023

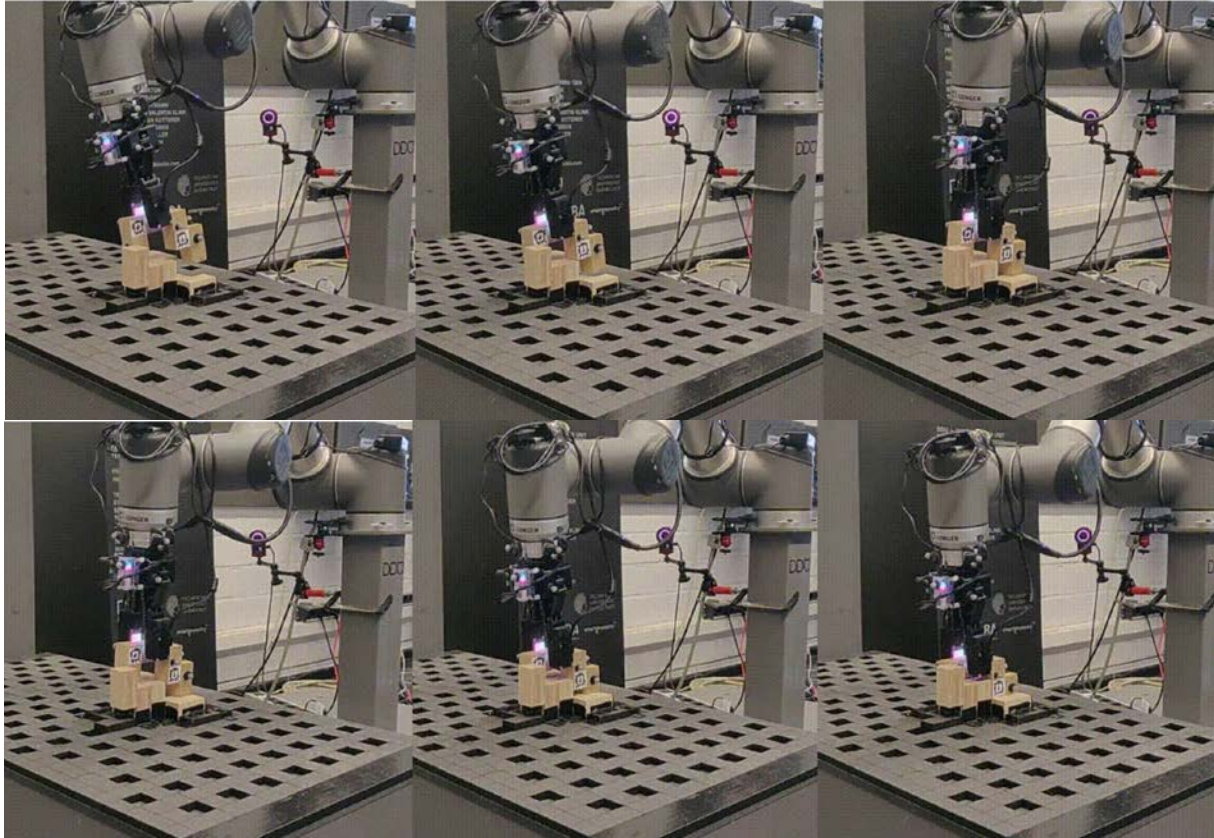
und zu planen. Die Pläne werden nach jeder ausgeführten Aktion neu generiert, wodurch sich ein modellprädiktiver Steuerkreislauf ergibt. Dank des gelernten Modells, das die taktilen Ablesungen mit den Positionen der Blöcke und den Roboterbefehlen in Verbindung bringt, ist der eingesetzte Cross-Entropy-Method-Planer in der Lage, Aktionssequenzen zu finden, die zur höchsten Belohnung führen, d.h. eine erfolgreiche Teileinfügung erreichen.

Der komplexe experimentelle Aufbau bestehend aus einem 6-Achs Roboter, visuell taktile Sensoren, Objektverfolgung, SL-Block Lager und Montageaufgaben ermöglichte das Sammeln von zahlreichen Episoden taktiler Montageprozesse als Bilddaten, die als Simulation nicht generierbar gewesen wären. Das System erlernte in einem vollständig unbeaufsichtigten Modus die Repräsentation der Bilder taktiler Sensoren. Dafür nutzten wir Autoencoder.

Diese künstlichen neuronalen Netze sind in der Lage, die Repräsentation von Daten auf komprimierte Weise zu erlernen und damit wesentliche Bildmerkmale und Informationen für die taktile Steuerung zu extrahieren und zu kodieren. Diese Repräsentation ist in einem gewissen Rahmen auf unterschiedliche taktile Sensoren generalisierbar. Damit erzielten wir im Projekt einen ersten Schritt zur Erforschung von hochgradig generalisierbaren taktilen Repräsentationen, für die breite Anwendung von visuell taktile Sensoren in Architektur und Bauwesen (vgl. Abbildung 58).

Abbildung 58

Ausgestattet mit zwei DIGIT-Sensoren, realisiert der Roboter die Einfügung eines SL-Blocks in eine existierende Struktur. Eine besondere Herausforderung dabei stellt die erforderliche hohe Präzision dar, bedingt durch die geringen Spaltmaße der SL-Blöcke.

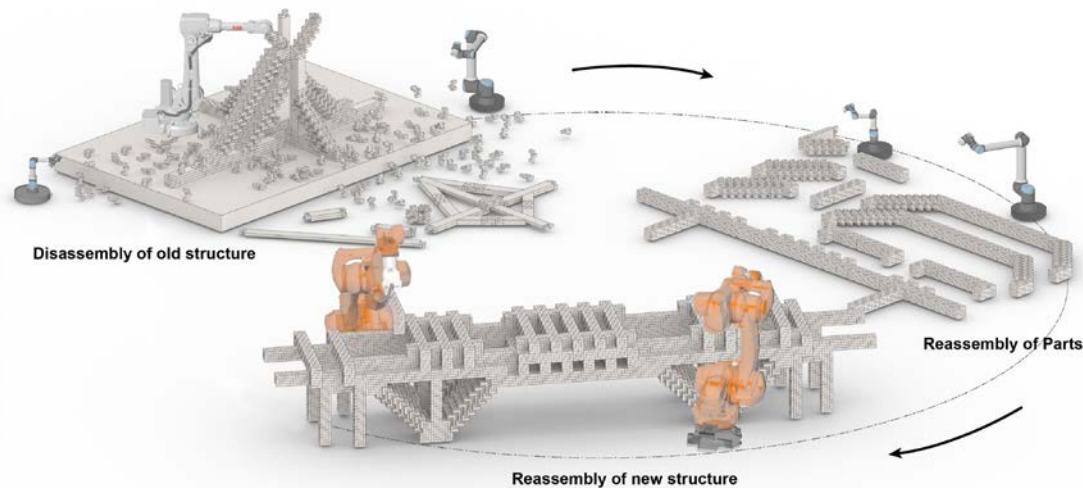


Quelle: Team DDU&IAS, TU Darmstadt 2023

4.6 SL-Blöcke, optimiert für die robotische Fügung

Abbildung 59

Konzeptbild für einen zukünftigen robotergestützten Bauprozesses, bei dem Roboter autonom Strukturen aus SL-Blöcken montieren/ demontieren und wieder zusammensetzen.



Quelle: DDU, TU Darmstadt 2023

Parallel zu dem oben beschriebenen Ansatz des Fügens von SL-Blöcken mit Hilfe von modellprädiktiver Steuerung und visuell taktiler Sensorik haben wir die SL-Blöcke so weiterentwickelt, dass ihre Geometrie auf die robotische Fügung angepasst wird (vgl. Abbildung 60). Bei diesem roboter-orientierten Entwerfen strebt man an, Bauteile so zu entwerfen, dass sie einfach von einem Roboter verarbeitet, gegriffen und montiert werden können. Der Verarbeitungsprozess bildet ein Entwurfskriterium. Somit nähern wir uns dem Thema der robotischen Fügung im Projekt von zwei Seiten an: *i) Einerseits durch die Nutzung hochentwickelter Sensorik und dem Verstärkenden Lernen, um möglichst autonom Aufgaben erledigen zu können, die nicht für Roboter optimiert sind; ii) Auf der anderen Seite haben wir "blinde" Roboter genutzt, die ohne Sensorik komplett vorprogrammierte Fügeprozesse abarbeiten.* Hier haben wir versucht, die SL-Blöcke so zu entwerfen, dass sie Eigenschaften bekommen, die das Fügen erleichtern. Der "dumme" Roboter fügt "intelligente" Bauteile. Ziel war es, beide Ansätze miteinander zu vereinen, um bestmögliche autonome Montageergebnisse zu erzielen: Intelligente Roboter fügen Bauteile, die für die robotische Fügung optimiert sind. Im Folgenden wird die Arbeit an roboter-orientierten Elementen und ihrer Fügung beschrieben.

Abbildung 60

Ein Universal Robot UR 10 ausgestattet mit einem elektronischen Parallelgreifer von Schunk und 3D gedruckten Fingern, die geometrisch präzise für die speziellen SL-Blöcke mit linearen Nuten entwickelt wurden. Weiße Würfel dienen als Stützkonstruktionen oder verhindern das Verrutschen von SL-Blöcken während der Montage.

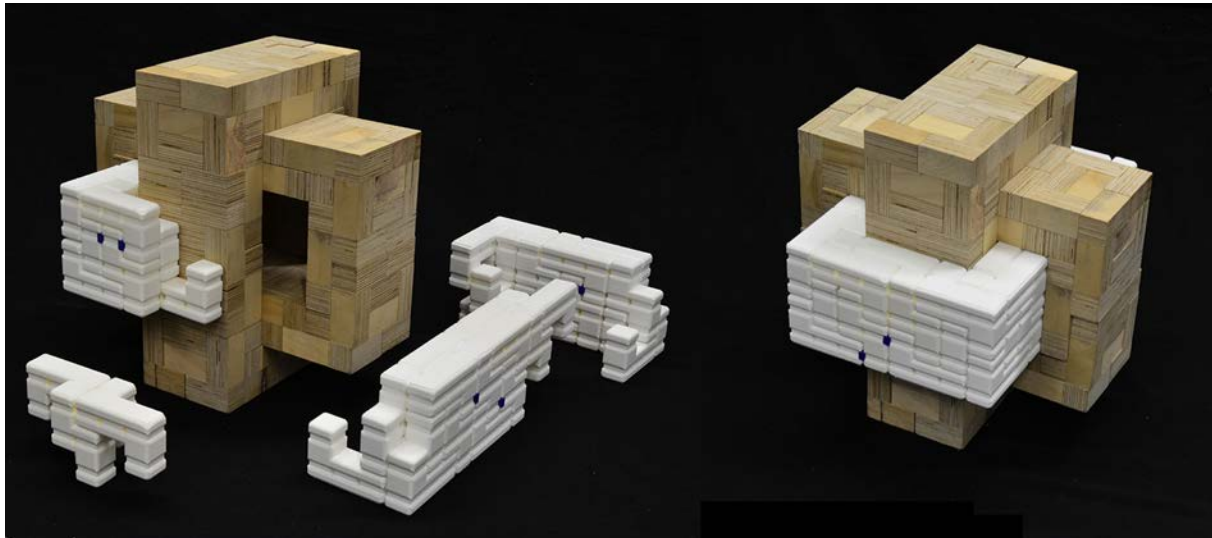


Quelle: DDU, TU Darmstadt 2023

4.6.1 Entwurf und Fertigung von selbstkalibrierenden SL-Blöcken

Abbildung 61

Drei SL-Strands verschlungen in einem borromäischen Knoten. Die weißen, 3D gedruckten SL-Blöcke verfügen über abgerundete Ecken für eine leichtere Montage. Außerdem bieten Nuten Angreifpunkte für einen gabelförmigen Greifer.



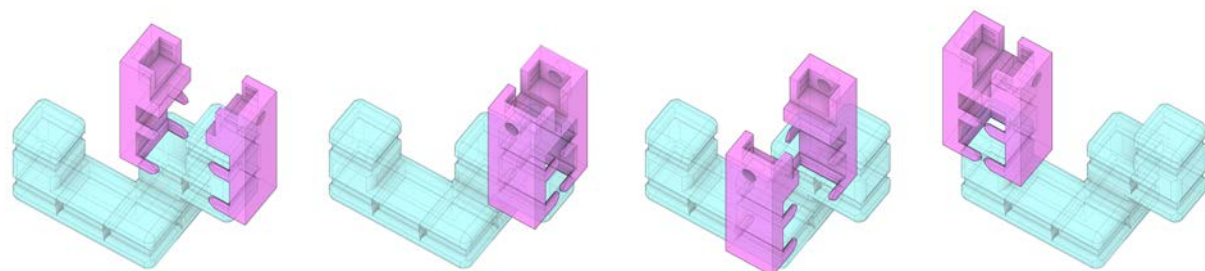
Quelle: DDU, TU Darmstadt 2023

SL-Blöcke verzahnen sich zu Konstruktionen, indem die Bewegung zwischen den Blöcken in bestimmte Richtungen gesperrt wird oder große Kontaktflächen Reibung erzeugen. Die Anzahl der gesperrten Freiheitsgrade hängt von der Komplexität der Aggregation, der Fügeregeln und dem Verriegelungsmechanismus ab (vgl. Abbildung 61). Gleichzeitig benötigt die robotergestützte Montage Spielraum bzw. eine verminderte Reibung, um Elemente platzieren zu können. Berührungssensitive Verbindungen stellen eine Herausforderung für die Montage dar. Um eine Balance zwischen diesen widersprüchlichen Anforderungen zu finden, wurden die Formen der SL-Blöcke wie folgt im Sinne des roboter-orientierten Entwerfens (Bock, 2015) weiterentwickelt.

Abgerundete Ecken vermindern die Reibung und erleichtern das Einsetzen der Teile. Lineare Nuten sind passend zu den Robotergreifern ausgebildet (vgl. Abbildung 62). Ihre Form hilft dabei, das Bauteil beim Greifen möglichst präzise auszurichten.

Abbildung 62

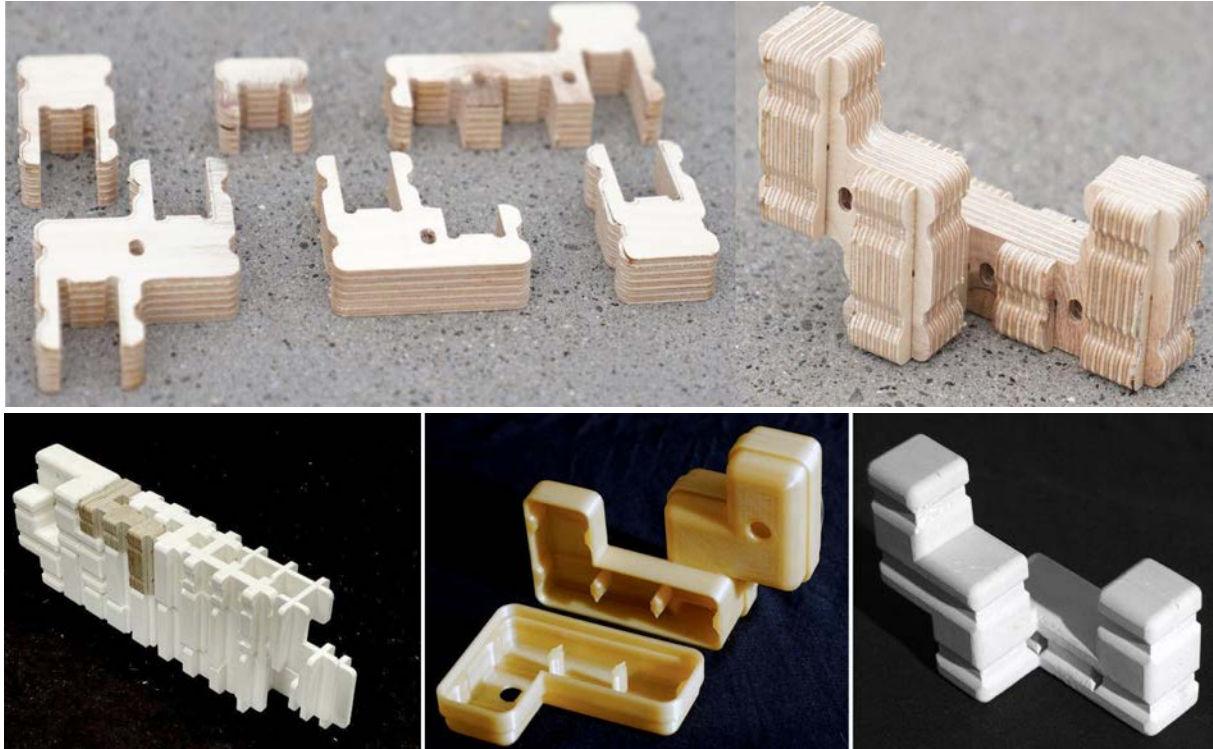
SL-Block und Gabelgreifer als integral entwickeltes System. Kerben in den SL-Blöcken komplementieren die Gabelquerschnitte der Greifer. Die konisch zulaufenden Greifer zentrieren und positionieren den Block beim Zugreifen.



Quelle: DDU, TU Darmstadt 2023

Abbildung 63

SL-Blöcke aus CNC gefrästen, steckbaren Furniersperrholz-Elementen (oben), 3D gedrucktem, weißen Kunststoff (unten links) und aus Beton, gegossen in 3D gedruckte Schalungen (unten rechts).



Quelle: DDU, TU Darmstadt 2023

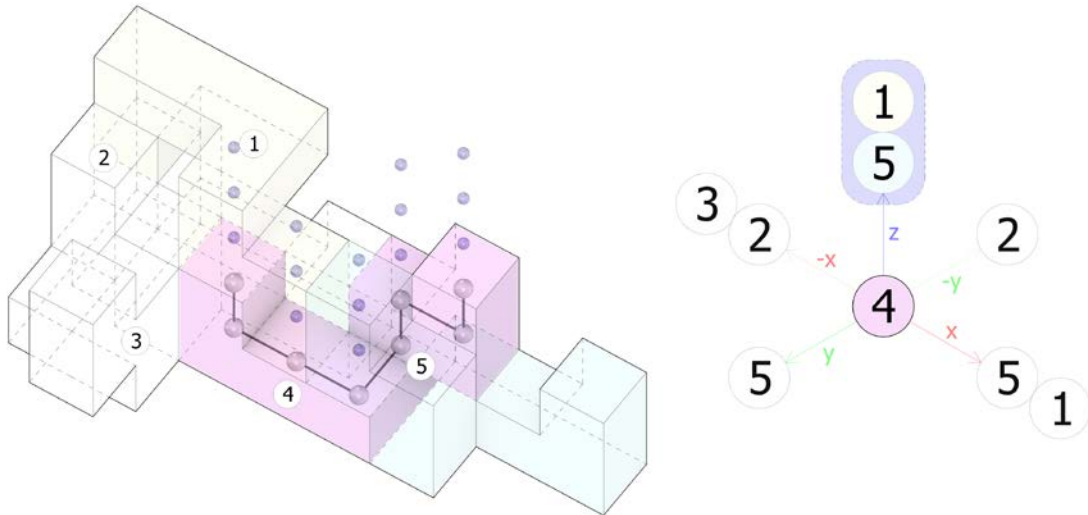
Für die Herstellung der komplexer geformten SL-Blockgeometrie testeten wir FDM 3D-Druck, Laserschneiden und CNC-Fräsen. Die Produktion von SL-Blöcken aus kreuzverleimten Furniersperrholzplatten erforderte eine Zerlegung des SL-Blocks in Elemente, die aus Plattenmaterial gefräst werden können. Die Geometrie der selbstzentrierenden Kerben wurde an die Werkzeuge der CNC-Fräse angepasst. Um das Zusammenkleben von Teilen zu vermeiden, erstellten wir Teile, die durch Nut- und Feder-Verbindungen passgenau zu einem SL-Block zusammengefügt werden konnten (vgl. Abbildung 63 oben). Diese Teile können aus Plattenmaterialien unterschiedlicher Dicke gefertigt werden. Die Verwendung dünnerer Holzplatten kann eine leichtere Struktur schaffen, die Porositäten erzeugt und das Potenzial hat, auf unterschiedliche Lasten und Belastungen in einer Struktur zu reagieren (vgl. Abbildung 63 unten links).

Andere SL-Blöcke wurden auf zwei verschiedene Arten aus Beton gegossen: zusammenklappbare Formen und dauerhafte Schalung. Doyle et al. (Doyle, 2019) schlagen vor, auflösbare Schalung und flexible Stahlbewehrung zu verwenden. Wir testeten den 3D-Druck von Formen mit wasserlöslichen Filamenten für zusammenklappbare Formen und erreichten präzise SL-Blöcke in einem zuverlässigen Prozess (vgl. Abbildung 63 unten rechts); jedoch kann jede 3D-gedruckte Form nur einmal verwendet werden, was dem serienmäßigen Charakter der SL-Blöcke widerspricht. Die Verwendung der 3D-gedruckten Hülle als dauerhafte Schalung erzeugt eine dünne Kunststoffschicht um den Betonkern und zwischen den Elementen. Dies erleichtert die Montage der Elemente und schützt den spröden Beton während der Montage und Demontage. Eine solche Schicht könnte die Haltbarkeit verbessern, führt aber zu SL-Blöcken aus mehreren Materialien, eine unerwünschte Eigenschaft in einem System, bei dem ein Block idealerweise aus einem Material besteht.

Algorithmen für robotische Montagesequenzen von SL-Blöcken

Abbildung 64

SL-Block und seine Repräsentation als Graph zur Analyse der Einbausituation und möglicher Freiheitsgrade für die Montage. SL-Block 4 wird von seinen Nachbarn 1, 2, 3, und 5 in all seinen Bewegungsrichtungen eingeschränkt. Er kann nicht demontiert werden.



Quelle: DDU, TU Darmstadt 2023

Für die automatisierte Montage von SL-Blöcken muss die Montagesequenz so entwickelt werden, dass alle Blöcke einbaubar sind, ohne dass die bereits hergestellte Konstruktion ein Hindernis für das nächste Bauteil oder den Roboter darstellt. Kollisionen während der Montage gilt es zu vermeiden. Unsere Montagesequenzen sind inspiriert von Songs rekursiven Algorithmus zum Entwerfen von verzahnten 3D-Puzzles (Song, 2012) und Wang et al.'s Forschung zur Beschreibung von Demontagesequenzen und deren Umkehrung zu Montagesequenzen (Wang, 2021).

Neben Reihenfolge und Richtung des Zusammenbaus mussten geeignete Greifpunkte für den Roboter gefunden werden, die Kollisionen verhindern und ein stabiles Fügen der Blöcke ermöglichen. Jeder SL-Block ist in der Aggregation von mehreren anderen SL-Blöcken umgeben, die seine translatorischen Freiheitsgrade einschränken.

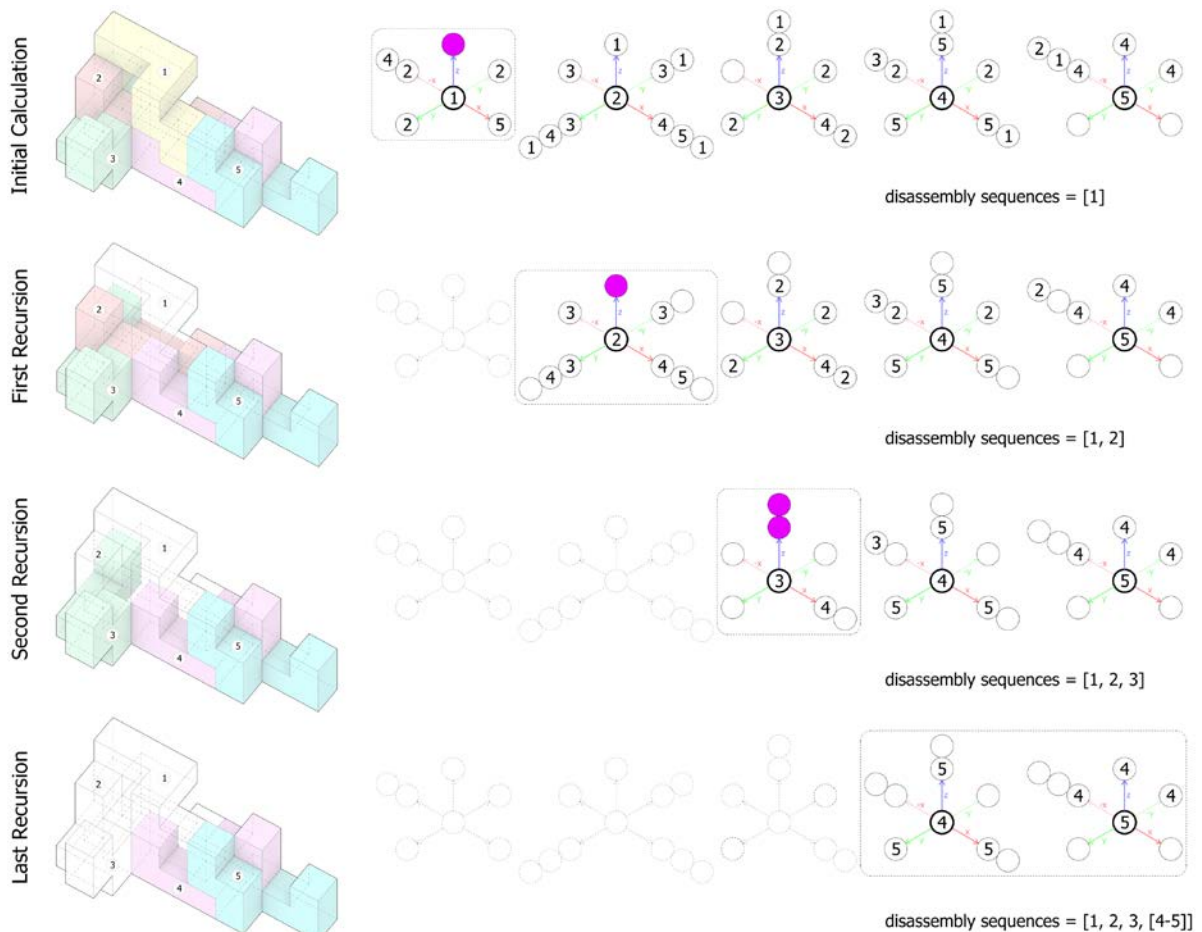
Im digitalen Modell kennen wir die Lage jedes einzelnen Blocks und nutzen diese Informationen, um mögliche Bewegungsrichtungen zur Montage zu finden. Dabei werden die SL-Blöcke als Graphen dargestellt, die aus der Verbindung der Mittelpunkte der acht Basiswürfel jedes SL-Blocks entstehen (vgl. Abbildung 64/65).

Wir nennen SL-Blöcke, die die Bewegung ihrer Nachbarn einschränken, "Blockees". Um die Blockees um jeden SL-Block zu bestimmen, wird eine Liste von sechs möglichen Richtungen (z , $-z$, x , $-x$, y , $-y$) für den Zusammenbau des SL-Blocks analysiert. Die Sequenzen und Richtungen für die Demontage können mithilfe eines rekursiven Algorithmus bestimmt werden:

- i) In jeder rekursiven Ebene gibt es eine Hauptschleife, die über jeden SL-Block iteriert, und eine weitere Schleife in jedem Objekt, die Richtungspunkte überprüft. Der Algorithmus vergleicht die X-, Y- und Z-Koordinaten dieser Punkte, um Übereinstimmungen in anderen Blöcken oder Montageumgebungen zu finden.

Abbildung 65

Grafische Darstellung der Demontagesequenz von SL-Blöcken. Links ist jeder der fünf SL-Blöcke mit einer eigenen Farbe und einer Zahl dargestellt. Rechts veranschaulicht ein Graph für den jeweils zu entfernenden SL-Block die blockierenden Nachbarn als Zahl. Die Richtung, in der der Block entfernt werden kann, ist mit einem lila Punkt markiert. Ist ein SL-Block entfernt, startet die Analyse für die aktualisierte Aggregation von neuem. Der rekursive Prozess wird fortgesetzt, bis alle Blöcke entfernt sind.



Quelle: DDU, TU Darmstadt 2023

ii) Wenn die Koordinaten übereinstimmen, bedeutet dies, dass der Block ein Hindernis [nicht beweglich] ist; andernfalls kann er sich frei in diese Richtung bewegen.

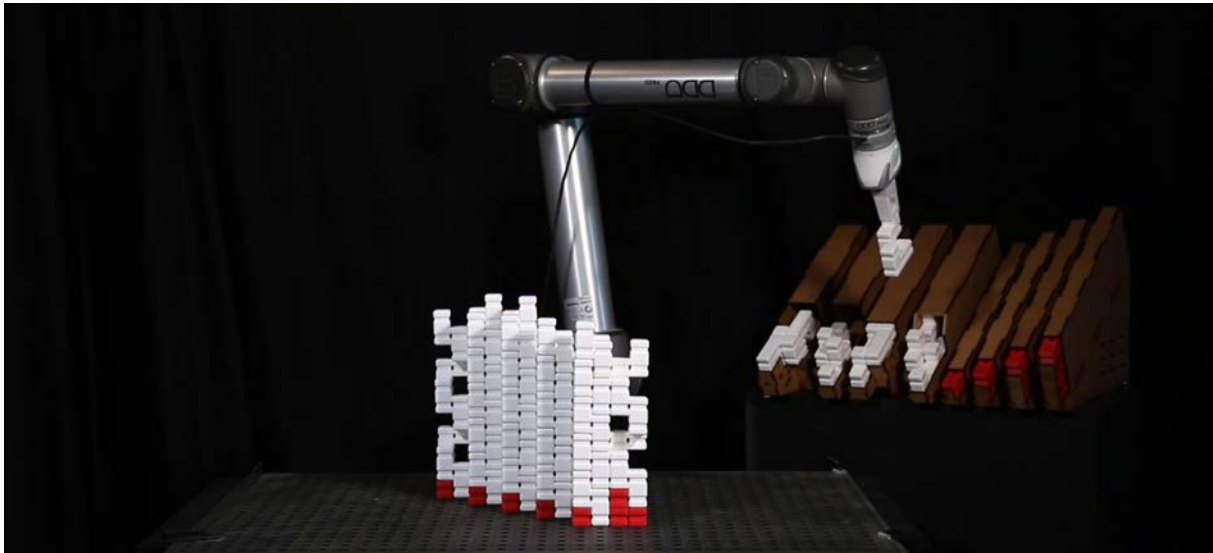
iii) Wenn es keine Übereinstimmung gibt, können wir den Block in dieser Richtung demontieren und das Objekt und die Bewegungsrichtung des freien Blocks zur Demontageliste hinzufügen, während wir ihn aus der Hauptaggregationsliste entfernen.

iiii) Schließlich muss die Liste der SL-Block-Objekte aktualisiert werden, um die Indexnummer des freien SL-Blocks aus allen Blocklisten zu entfernen und den Aggregationsraum für die nächste Rekursion freizugeben.

4.6.2 Aufbau der Montageumgebung

Abbildung 66

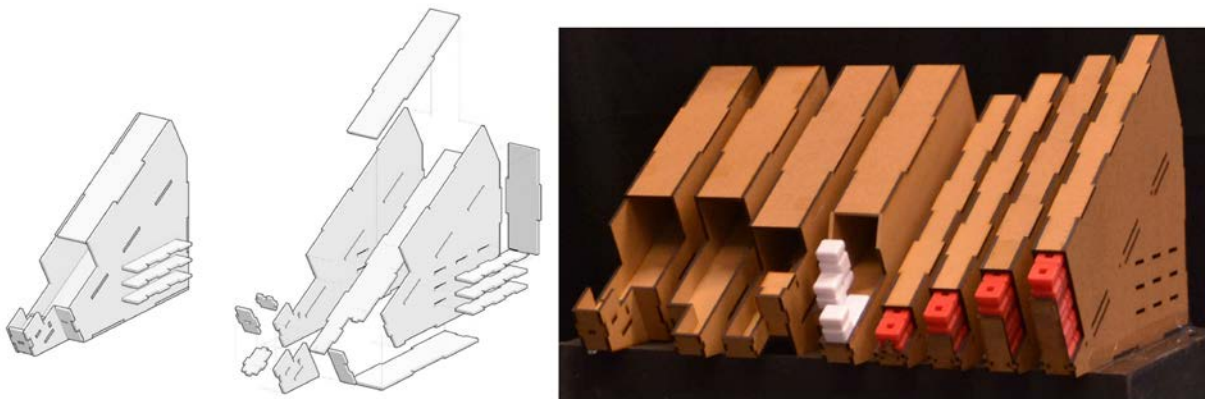
Automatisierte robotische Aggregation von weißen SL-Blöcken, die durch rote Würfel gestützt werden. Das Bild zeigt den Roboterarm beim Herausgreifen eines SL-Blocks aus dem Materialzuführungssystem.



Quelle: DDU, TU Darmstadt 2023

Abbildung 67

3D Modell, Explosionszeichnung und Fotografie des Materialzuführungssystems. In einem geneigten Schacht werden die Blöcke hintereinander aufgereiht. Entnimmt der Roboter unten einen Block, rutscht der nächste automatisch nach. Das System wird am oberen Ende befüllt.



Quelle: DDU, TU Darmstadt 2023

Strategien zur Fixierung der laufenden Montage

Zu Beginn der Montage muss sichergestellt werden, dass die SL-Blöcke beim Montieren nicht verrutschen und ihre gewünschte Position nicht verändern. Einige SL-Block Aggregationen starten nicht als planare Konfigurationen, sondern werden sofort räumlich angeordnet. Dies erfordert zusätzliche temporäre Stützstrukturen, bis ein stabiler Montagezustand erreicht ist. Hierfür wurden Fixierungswürfel für den Montageprozess hinzugefügt (vgl. Abbildung 66). Diese roten Würfel können während des robotergestützten Montageprozesses nach Bedarf angeordnet werden. Ihre Größe entspricht den Würfeln, aus denen die SL-Blöcke bestehen. Sie können über

Zapfen an einer perforierten Montageplatte oder anderen Würfeln befestigt werden. Die Blöcke unterstützen die Selbstkalibrierung der SL-Blöcke, sichern die SL-Blöcke während des Montageprozesses und stützen auskragende Abschnitte der Struktur, bis die Aggregation durch das Platzieren von Schlüsselsteinen stabil ist.

Materialzuführungssystem für die robotische Montage

Für die präzise und ortsgenaue Bereitstellung der SL-Blöcke für den Roboter, der ohne Sensorik arbeitet, haben wir ein Materialzuführungssystem entwickelt. Derzeit können acht Elemente eingesetzt werden, das System ist aber erweiterbar. Einzelne SL-Blöcke, aber auch SL-Block-Paare können im System bereitgestellt werden. Nach Entnahme eines Blocks, rutschen die folgenden Elemente automatisch an die Greifposition nach. Auch die roten Stützwürfel werden so bereitgestellt (vgl. Abbildung 67).

4.6.3 Aufgaben- und Bewegungsplanung für den Roboter

Bei der Aufgaben- und Bewegungsplanung für den Roboter wird eine Abfolge von Aktionen und Bewegungen bestimmt, die ein Roboter ausführen soll, um eine gegebene Aufgabe effizient zu erfüllen. Bei der robotergestützten Montage beinhaltet die Aufgabenplanung die Bestimmung, welche Teile montiert werden sollen, in welcher Reihenfolge und wie jede Montageoperation durchgeführt wird. Die Aufgabenplanung in der robotergestützten Montage lässt sich in zwei Hauptphasen unterteilen: High-Level- und Low-Level-Planung.

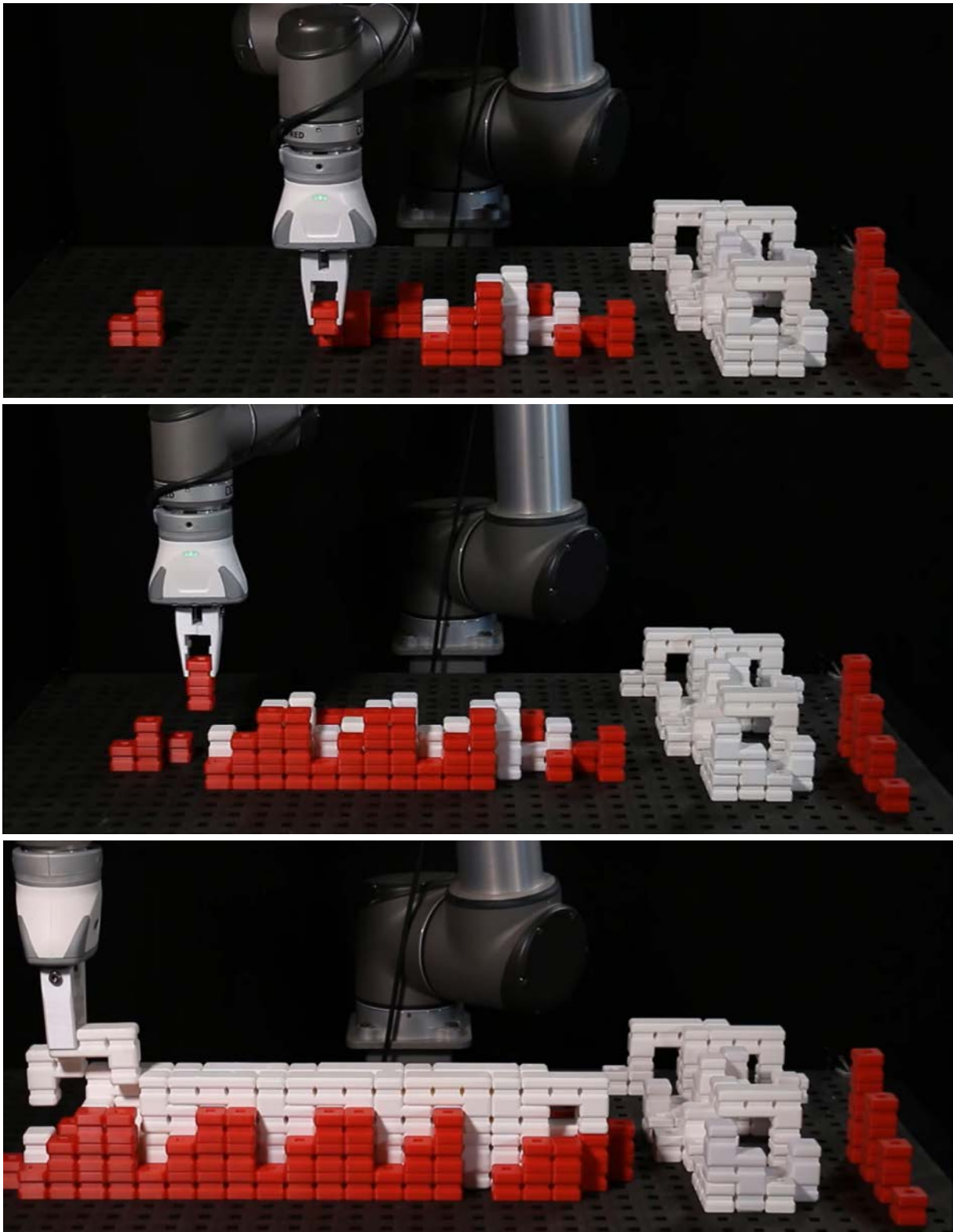
Bei der High-Level-Planung wird die gesamte Abfolge der Montageoperationen bestimmt, die zur Erfüllung der Aufgabe erforderlich sind (vgl. Abbildung 68). Dies beinhaltet die Aufteilung des Montageprozesses in Teilprozesse, die Bestimmung der Reihenfolge dieser Teilprozesse und die Zuweisung von Ressourcen (z. B. Roboterarme, Greifer usw.) zu jedem Teilprozess. Der Montagevorgang wurde in drei Schritte unterteilt: das Ergreifen eines Teils am Abholort, das Transportieren in die Nähe des Zielorts und schließlich das Platzieren des Teils. Bei der High-Level-Aufgabenplanung werden zwei Hauptszenarien berücksichtigt: das Aufnehmen/Entfernen von SL-Blöcken, deren Transfer und das Platzieren an ihrem vorgesehenen Ort.

Die Low-Level-Planung umfasst die Bestimmung der spezifischen Aktionen, die der Roboter ausführen sollte, um jeden Teilprozess abzuschließen. Dies beinhaltet die Bestimmung des optimalen Weges für den Roboter, um sein Endeffektor zu bewegen, um ein Teil aufzunehmen, es an den gewünschten Ort zu bringen und es mit anderen Elementen zu montieren.

In unserem Projekt befasst sich die Low-Level-Aufgabenplanung mit den richtigen Montage- und Demontagesequenzen für SL-Blöcke, der Bestimmung der Richtung, in der sie montiert oder demontiert werden sollten, der Suche nach den Greifpunkten für die SL-Blöcke und der Erstellung von Anordnungen für Fixierblöcke. Zusätzlich muss der Algorithmus die korrekten Montage-, Demontage- oder Umordnungssequenzen für Fixierblöcke sowie deren optimale Greifpunkte und -richtungen identifizieren. Wir entwickelten einen Algorithmus, der den Stützbedarf einer SL-Block Aggregationen und ihrer Montagezustände ermittelt und die Positionen der Fixierblöcke festlegt. Damit werden unerwünschte Bewegungen während der Platzierung vermieden (vgl. Abbildung 68). Zusätzlich ist es notwendig, die Greifpunkte dieser Fixierblöcke zu identifizieren. Der Algorithmus arbeitet in zwei Stufen. Zuerst erkennt er potenzielle Orte, an denen Unterstützung oder Fixierung um die gegenwärtige Platzierung des SL-Blocks herum benötigt wird. Dann überprüft er jeden Ort, um sicherzustellen, dass keine Kollisionen zwischen dem Greifer, dem positionierten SL-Block und anderen Blöcken in der Montageumgebung auftreten. Wenn keine Kollision vorliegt, wird der Greifpunkt verzeichnet. Diese Strategie garantiert, dass die Konfiguration der Fixierblöcke mit kollisionsfreien Greifpunkten in späteren Schritten entfernbar ist. Anschließend berechnet der Algorithmus die potenziellen Greifpunkte für die aktuelle Platzierung des SL-Blocks unter Berücksichtigung der Position der Fixierblöcke. Da es verschiedene mögliche Greifpunkte geben kann, wählt er den höchsten aus.

Abbildung 68

Bildsequenz der robotischen Fügung von Stützwürfeln (rot) und SL-Blöcken (weiß). Eine lineare Struktur wird hergestellt.



Quelle: DDU, TU Darmstadt 2023

4.6.4 Autonome robotische Montage

Auf Basis der Erkenntnisse aus der Erforschung automatisierter Strategien für die Montage von trocken gefügten Elementen haben wir den Einsatz autonomer Roboter in der Montage untersucht.

Die oben gezeigten Methoden basieren auf manuellen Heuristiken. Das Ziel der autonomen Montage wäre jedoch eine vollständig autonome Prozesskette, die als Eingabe eine vom Benutzer definierte, gewünschte Form und die verfügbaren Bausteine nimmt und direkt in Aktionen umsetzt, die vom Roboter ausgeführt werden können.

Als Ergebnis sollte die gewünschte Form optimal durch die autonomen Aktionen des Roboters erzeugt werden. Um dies zu erreichen, muss ein Ansatz entwickelt werden:

- i) Dieser Ansatz sollte den kombinatorischen Suchraum der Montagevarianten verfügbarer Teile handhaben kann.
- ii) Berücksichtigen die Einschränkungen des Roboters, wie zum Beispiel seinen begrenzten Arbeitsbereich.
- iii) Erstellen einen durchführbaren Plan, der die Stabilität der Struktur zu jedem Zeitpunkt im Bauprozess gewährleistet.
- iv) Vermeiden Kollisionen zwischen dem Roboter und der Struktur.
- v) Im Fall von SL-Blöcken versucht sicherzustellen, dass die Gesamtstruktur den Fügeregeln entsprechend an jeder Stelle korrekt ineinandergreift und die oben beschriebenen Lernergebnisse des visuell taktilen Greifers nutzt.

Um all diese verschiedenen Ebenen der Komplexität zusammen zu bewältigen, haben wir Q-Learning (Mnih et al. 2013) genutzt. Ein Ansatz auf der Basis von modellfreiem verstärkendem Lernen. Bei dieser Methode agieren die Roboter in einer Situation und führen Aktionen aus, ohne über ein komplettes analytisches Modell dieses Kontextes (model-based) zu verfügen. Die Roboter lernen, indem sie die Konsequenzen ihrer eigenen Handlungen beobachten, wahrnehmen und bewerten.

Dieser Ansatz erschien uns für die Aufgabe adäquat, da eine vollständige Perzeption von Baustellenkontexten oft zu rechenintensiv oder gar nicht möglich ist. Beim modellfreien verstärkenden Lernen erlernen Roboter die Zuordnung zwischen eigenen Aktionen und den Auswirkungen, die sie auf die Umgebung haben. Der Algorithmus besitzt kein Modell der Umgebung und benötigt nur das Standard-Interface des Roboters für die Steuerung. Er strebt lediglich die Minimierung der Fehlerfunktion an, z.B. die Lagerposition eines SL-Blocks und seine gewünschte Position in der SL-Block Aggregation.

Das Lernen erfordert einen Trainingsprozess, der in einer Simulationsumgebung durchgeführt wurde (siehe Abbildung 69). Hier wird die Qualität der Aktion a , gegeben den aktuellen Zustand s , durch $Q(s,a) = r(s,a) + \max(a')Q(s',a')$ gegeben, wobei $r(s,a)$ die unmittelbare Belohnung für die Ausführung der Aktion a im Zustand s ist (d.h. eine positive Belohnung, wenn die stabile Füllung der gewünschten Form erhöht wurde und die Aktion erfolgreich war, andernfalls eine negative Belohnung), und $\max(a')Q(s',a')$ den Wert der besten verfügbaren Aktion im Zustand s' darstellt, den wir erreichen. Wichtig ist, dass der langfristige Horizont der Aufgabe berücksichtigt wird, da wir erst am Ende beurteilen können, ob alle Aktionen die gewünschte Zielstruktur optimal nachbilden. Um die Qualität der Aktionen vorherzusagen, wird ein graphbasiertes neuronales Netzwerk verwendet (Scarselli et al. 2008). Wie in der Abbildung 69 gezeigt, wird die gewünschte Struktur, die gebaut werden soll, in Voxel unterteilt. Dies ermöglicht es, einen Graphen zu definieren, mit Knoten, die die Mittelpunkte der jeweiligen Voxel darstellen.

Um die Aktionswerte vorherzusagen, wird zunächst Information zwischen allen Knoten im Graphen ausgetauscht, um ein besseres Verständnis der Montageszene zu erhalten. Alle notwendigen Werte für das Bewegen eines noch nicht platzierten Blocks zu einer leeren Position im Raster, werden vorhergesagt.

Graphbasierte Darstellungen von Montageszenen funktionieren für sehr unterschiedliche Problemgrößen, da der Informationsaustausch lokal zwischen den Knoten definiert ist. Daher kann dieselbe gelernte Graphendarstellung verwendet werden, um Probleme mit unterschiedlichen Mengen von verfügbaren Bausteinen und gewünschten Zielstrukturen zu bewältigen.

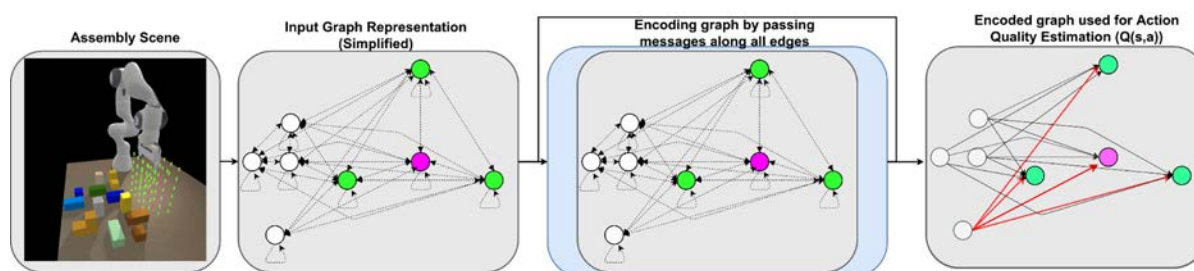
Auf Basis der graphbasierten Darstellung wurde der gelernte Prozess, für den Bau einer Struktur aus Kuben und Polykuben in der Simulation und in einem physischen Experiment getestet (vgl. Abbildung 70). Der Algorithmus agiert dabei vollständig autonom, d. h., er erhält nur als Eingabe die gewünschte Zielstruktur und die verfügbaren Bausteine. Die Bauteilplatzierung und die robotischen Aktionen werden ausschließlich durch den zuvor vorgestellten Algorithmus gehandhabt.

Dieser learn2assemble-Algorithmus bietet einen flexiblen, autonomen robotergesteuerten Montageagenten für den Bau von 3D-Formen mit mehr als zehn Blöcken. Wir erzielen hier drei wichtige Ergebnisse:

- 1) Wir schlagen eine auf Multi-Head-Attention basierende Graphendarstellung für das 3D-Montageproblem vor, die flexibel genug ist, um beliebige, stabile Strukturen darzustellen.
- 2) Wir entwickeln einen integrierten Langzeit-Manipulationsalgorithmus, der durch Exploration lernt, der in Kombination mit einer modellbasierten Suche zu generalisierbaren Fähigkeiten führt. Unsere Methode berücksichtigt den Roboter im Prozess und integriert high-level Aktionsplanung mit low-level Bewegungsgenerierung, um Steuerungsprozesse zu erlernen, die die kinematische Machbarkeit und Stabilität der konstruierten Struktur gewährleisten.
- 3) Schließlich wurde eine neuartige Benchmark-Umgebung für die 3D-Roboter-Montage entwickelt, die modular ist für Tests mit und ohne den Roboter, für beliebige Zielentwürfe und mit einstellbaren Typen und Anzahlen von Bauteilen.

Abbildung 69

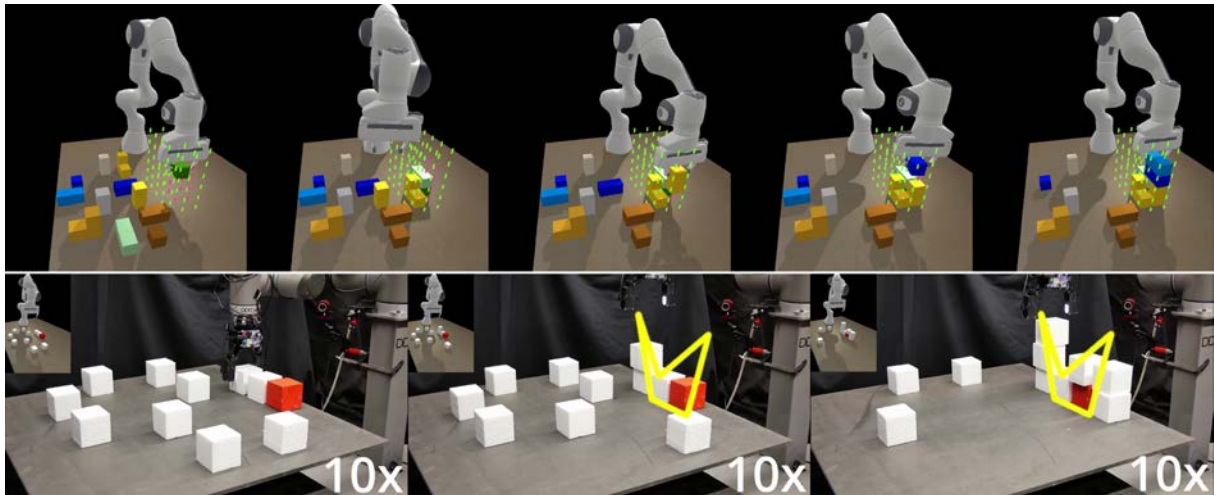
Montagesituation mit Roboter, Bauteilen und Zielhülle für eine Konstruktion in der 3D Simulation (links). Graphenbasierte Darstellung der Ausgangssituation und Berechnung der auszuführenden Aktionen.



Quelle: Team DDU&IAS, TU Darmstadt 2023

Abbildung 70

Oben: Simulationsumgebung mit einem 7-achsigen Roboter, auf einer Fläche verteilten Polykuben und einer Zielfigur als Hüllgeometrie. Ziel ist es, dass der Roboter eigenständig die Bausteine in die grün markierte Zielfigur setzt, so dass eine dreidimensionale Figur entsteht, die der Silhouette möglichst nahe kommt. Unten: Drei Fotos des physischen Experiments zur robotischen Stapelung von auf einer Platte zufällig verteilten Kuben zu einer Figur, die einer vorgegebenen räumlichen Polylinie folgt.



Quelle: Team DDU&IAS, TU Darmstadt 2023

Unsere empirischen Ergebnisse einer Reihe von repräsentativen Experimenten zeigen die Generalisierungsfähigkeit der vorgeschlagenen algorithmischen Lösung auf und liefern interessante Einblicke in die Kombination von integriertem Lernen und Planen für langfristige Manipulationen, die auf eine Reihe von robotergestützten Anwendungen anwendbar sind, weil Aufgaben erledigt werden können, ohne dass der Roboter vorab eine Kenntnis des Aufgabenplans besitzt. Stattdessen erfasst das System die räumlichen Beziehungen zwischen Zielkonstruktionen und unplatzierten Blöcken.

Unser hierarchischer Controller führt High-Level-Lernen von Handlungssequenzen und Zielspezifikationen (Konstruktion) zusammen mit Low-Level-Pfadplanung durch. Damit können Aufgaben mit langem Zeithorizont ausgeführt werden, bei denen viele Aktionen (kleine Teile fügen) zum Abschluss einer großen Aufgabe (Gesamtkonstruktion) beitragen. Die Demontage der gebauten Struktur gestaltet sich um ein Vielfaches einfacher, da nach der Platzierung der Bauteile ihre exakte Lage und auch die Fügereihenfolge aus dem Zusammenbau bekannt ist.

5. Ergebnisse

Die folgenden Ergebnisse wurden für die Themen Elemente, Konstruktionen, visuell taktiles Greifen, robotisch autonomes Montieren und den Demonstrator erreicht:

- **Elemente:** Eine umfangreiche Recherche, Analyse und Dokumentation zu bestehenden, reversibel gefügten, verzahnten Systemen wurde durchgeführt. Das SL-Block System wurde anschließend auf Basis der Forschung von Shih hinsichtlich der Übersetzung von einem geometrischen Prinzip in ein Materialsystem für Konstruktionen weiterentwickelt. Die Entwicklung umfasste die Herstellung von Elementen aus unterschiedlichen Materialien und mit verschiedenen Herstellungsmethoden. Die Geometrie der SL-Blöcke wurde für die Montage mit Roboter optimiert (Selbstkalibrierung, bessere Greifbarkeit, leichteres Montieren, Entwicklung neuer Stützkuben für die Montagephase).
- **Konstruktionsentwicklung:** Für die Konstruktionsentwicklung wurden Zerlegungsprozesse (Top-Down) und Wachstumsprozesse Strategien (Bottom-Up) erarbeitet. Bei der Bottom-up Strategie entstehen SL-Block Aggregationen aus der sequentiellen Aneinanderreihung von Blöcken mit Hilfe von spezifischen Fügeregeln (Engagements). Mit Hilfe von Grasshopper Definitionen wurde dieser Prozess für Benutzer extrem vereinfacht. Lediglich die Eingabe von Zeichenketten ist für die Generierung erforderlich. Der Prozess ist sehr ergebnisoffen und funktioniert in einer explorativen und suchenden Entwurfsphase, wobei verschiedene Engagements tektonischen Prinzipien (vertikaler Lastabtrag, Überspannen, Aussteifen) zuzuordnen sind. Außerdem wurden Top-Down Strategien entwickelt, bei denen SL-Block Aggregationen Referenzgeometrien folgen oder große Volumen zunächst in Raumraster/Voxel und anschließend in regelgerechte/verzahnende SL-Block Aggregationen verwandelt werden. Dabei kommen topologische Graphen zum Einsatz. Der Top-Down Ansatz ist dort zielführender, wo konventionell modellierte, einfache Geometrien mit Hilfe von SL-Blöcken realisiert werden sollen. Auch diese Unterteilung/3D-Kachelungsprozesse wurden in einer Grasshopper Definition implementiert.
- **Visuell taktiles Greifen:** Im Projekt wurde für den visuell taktilen Sensor DIGIT ein Klassifikator für Oberflächentexturen trainiert, um Materialbeschaffenheiten von Bauteilen zu erkennen. Dazu wurde ein Datensatz von Objekten mit unterschiedlichen Oberflächeneigenschaften erstellt und ein Modell darauf trainiert. Die Wiedererkennungsrates von Texturen durch den DIGIT Sensors liegt nun bei 90 %, während die Wiedererkennungsrates von Texturen bei menschlichen Versuchsteilnehmern bei gerade mal 67% lag. Außerdem wurden modellbasierte und lernbasierte Algorithmen zur Schätzung von Kontaktkräften, Druck und Drehmomenten entwickelt. Der Sensor wurde bei der kontaktreichen SL-Block Montage eingesetzt. In physisch durchgeführten Robotikexperimenten lernt der Roboter SL-Blöcke zu fügen. Mit diesen Montageprozessen erlernte das System in einem vollständig unbeaufsichtigten Modus die Repräsentation der Bilder taktiler Sensoren aus denen Kontaktkräfte, Druck und Drehmomente abgeleitet werden können. Diese Repräsentation ist ein erster Schritt zur Erforschung von generalisierbaren taktilen Repräsentationen, für die breite Anwendung von visuell taktiler Sensorik in Architektur und Bauwesen. Jenseits von Montageprozessen ist die Kontrolle von Oberflächenqualitäten und die Sortierung von Baumaterialien und -elementen denkbar. Weiterer Forschungsbedarf besteht bei der Skalierung von taktilen Greifern zum Heben größerer Lasten. Hier muss die Shore-Härte und Materialdicke der Silikonschichten angepasst werden. Die visuelle Interpretation der Verformungen ist skalier- und generalisierbar.
- **Autonome Montageprozesse:** Mit Hilfe des modellfreien verstärkenden Lernens haben Roboter in der Simulation Aggregationen aus verfügbaren Bauteilen hergestellt. Dabei war die gewünschte Zielform als abstrakte Hüllform vom Benutzer definiert. Auf dieser Basis hat der Roboter eigene Aktionen zur Montage autonom geplant und durchgeführt. Der Algorithmus arbeitet ohne konkretes Umgebungsmodell. Er strebt lediglich die Minimierung der Fehlerfunktion an, z. B. die Lagerposition eines SL-Blocks und seine gewünschte Position in der SL-Block Aggregation. In Simulationen und einer physischen Testumgebung mit

22 Bauteilen konnte dieser Prozess erfolgreich durchgeführt werden. Für einen Praxisübertrag ist weitere Forschung notwendig, um die Kluft zwischen Grundlagenforschung und Praxis zu überwinden. Bei vorangegangenen Arbeiten und dem Stand der Technik wurden deutlich einfachere Experimente durchgeführt. Hier sind die genauen Zielpositionen eines jeden Blocks vorab bekannt. In unserem Fall haben wir lediglich eine Zielform, die gefüllt werden muss. Der Algorithmus kennt nicht a priori die Reihenfolge und die Positionen der Blöcke, und es kann sogar sein, dass nicht alle Blöcke benötigt werden, um die Struktur zu konstruieren. Wenn wir zum Beispiel einen Turm bauen müssen, muss der Algorithmus herausfinden, dass er die Blöcke übereinander legen muss. Stand der Forschung ist aber, dass die Reihenfolge und die Zielposition der Blöcke, bekannt ist.

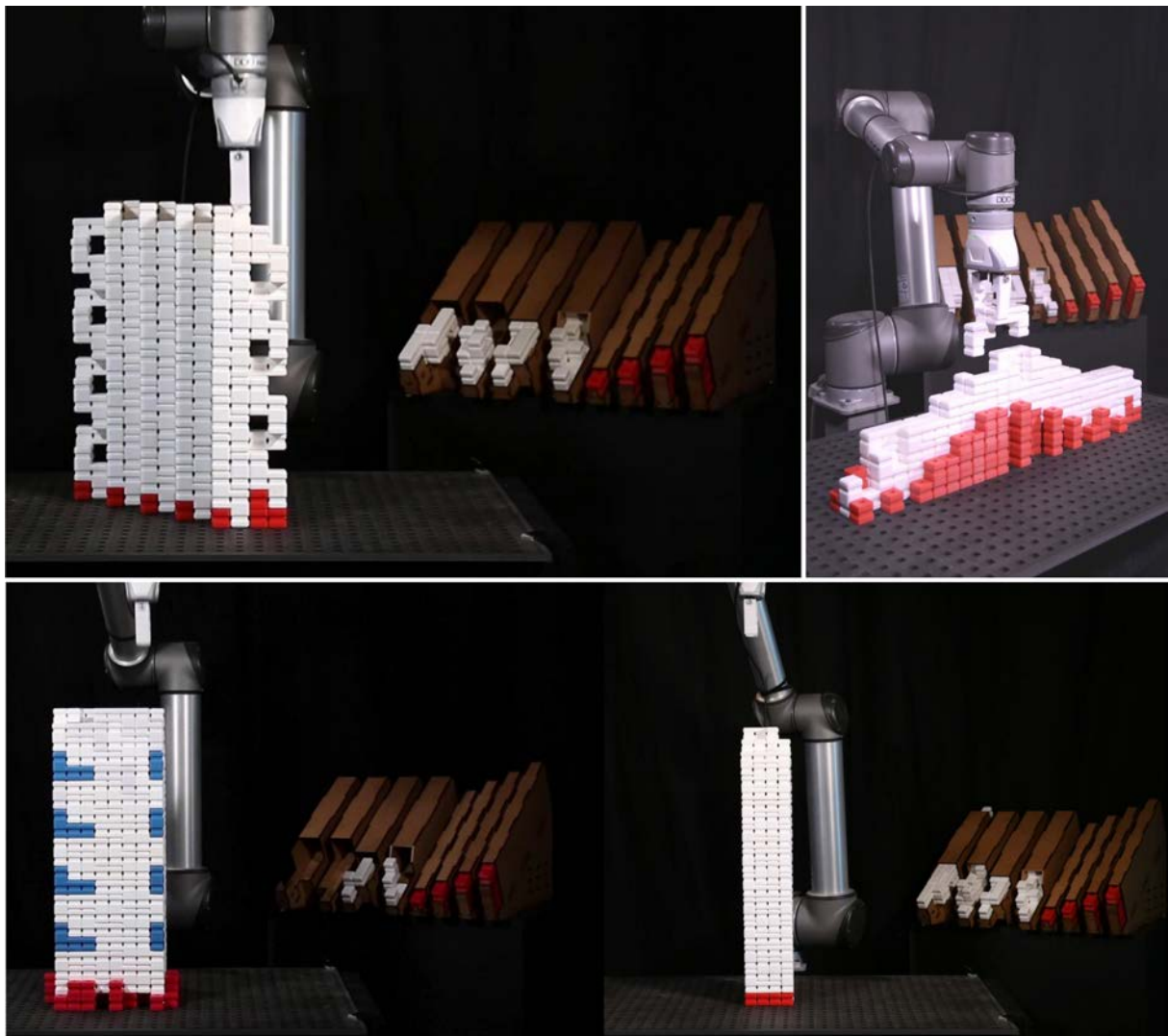
5.1 Demonstratoren

Die Forschungen zu Elementen, Konstruktionen, visuell taktilem Greifen und automatisiertem und autonomen robotischen Montieren wurden beim Entwurf und Bau von Demonstratoren als Prozesskette getestet. An verschiedenen Demonstratoren wurden jeweils unterschiedliche Fragestellungen untersucht.

Die ersten Demonstratoren sind SL-Block Aggregationen, die Wände, Stützen und Hohlquerschnitte bilden (vgl. Abbildung 71/72). Sie wurden mit Hilfe eines automatisierten, robotischen Prozesses hergestellt und auch wieder demontiert. Die Montageschritte waren vorprogrammiert. Mit diesem Ansatz konnten die Arbeiten zu Algorithmen für robotische Montagesequenzen von SL-Blöcken (Siehe 4.6) getestet werden. Außerdem kamen 3D gedruckte SL-Blöcke zum Einsatz, die mit geometrischen Eigenschaften ausgestattet waren, die das Montieren erleichtern und beim Greifen und Einbauen selbstkalibrierende Wirkung entfalten. Die Verminderung der Reibung zwischen den Blöcken machte eine größere Anzahl an Stützwürfeln erforderlich, um das Herausrutschen der Blöcke zu vermeiden. Hier ist weitere Entwicklungsarbeit notwendig, um eine gute Balance

Abbildung 71

SL-Block Aggregationen aus 3d gedruckten weißen Blöcken bilden Wände, Stützen, Träger und Hohlquerschnitte. Rote Würfel stützen die Konstruktionen. Blaue Blöcke markieren jeweils die Startseite eines SL-Strands, der auf den darunterliegenden SL-Strands gestapelt wird.



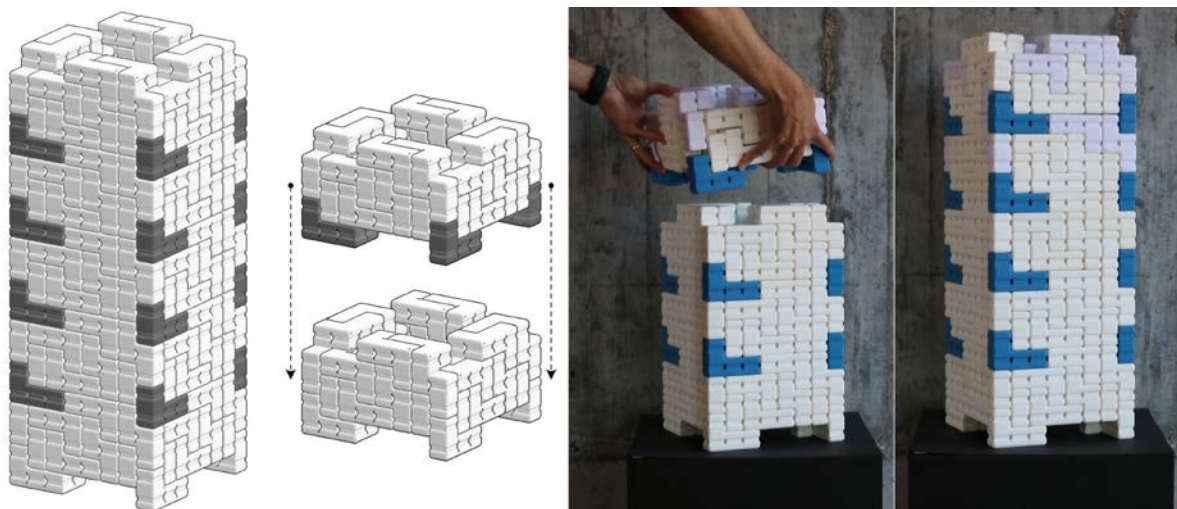
Quelle: Team DDU&IAS, TU Darmstadt 2023

zwischen notwendiger Reibung und Einbaubarkeit sicherzustellen. Die Erarbeitung der Montagesequenzen erbrachte außerdem wichtige Ergebnisse hinsichtlich geometrisch/topologischer Prüfung von Bauteilverzahnungen. Mit den entwickelten digitalen Werkzeugen kann neben der Herstellungsreihenfolge auch die Stabilität von Bauteilaggregationen und Bauzuständen ohne Strukturanalyse schnell und übersichtlich ermittelt werden. Für diese Serie an Demonstratoren wurde außerdem ein Materialzuführungssystem konzipiert und prototypisch umgesetzt.

In einem weiteren Demonstrator haben wir verschiedene Entwurfsstrategien, Elementhierarchien, Tragwerkstypologien, SL-Block Materialien und robotische Montageprozesse getestet. Der Demonstrator erinnert an eine Baumstütze und setzt sich aus vier Diagonalen zusammen, die zwei V-förmige Figuren bilden und einem zentralen soliden Sockel entspringen. In der Draufsicht bilden die Diagonalen vier gleich große Quadranten. Ein fachwerkartiger Träger und ein Zuggurt kreuzen sich und verbinden jeweils zwei gegenüberliegende obere Enden der Diagonalen. SL-Blöcke werden zu SL-Strands aggregiert, die dann wiederum mit den drei verschiedenen Fügeprinzipien für größere Bauteile (Verzahnte Verbindungen, rekursiv-verschachtelte Verbindun-

Abbildung 72

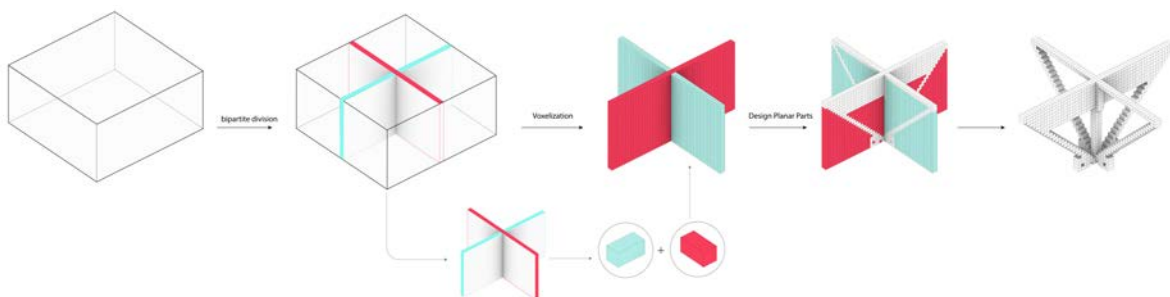
Hohlprofil aus SL-Block Aggregationen als digitales Modell und physischer Demonstrator. Das Bild in der Mitte zeigt, dass das Objekt abschnittsweise stapelbar ist, bzw. eine ganze Gruppe von SL-Blöcke nach oben abhebbar ist. Die blauen Steine markieren jeweils einen SL Strand Bauabschnitt. Die Fügestellen benötigen Eigengewicht oder hohe Reibung, da die Verzahnung hier nicht alle Freiheitsgrade sperrt.



Quelle: Team DDU&IAS, TU Darmstadt 2023

Abbildung 73

Bildsequenz zur Entwicklung des Demonstrators. Ein Quader wird durch zwei lotrecht platzierte Wände in vier gleiche Teile unterteilt. Die kreuzförmigen Wände werden in Voxel zerlegt. Aus beiden Voxelwänden werden V-förmige Elemente herausgezogen und jeweils an ihren oberen Enden horizontal verbunden.

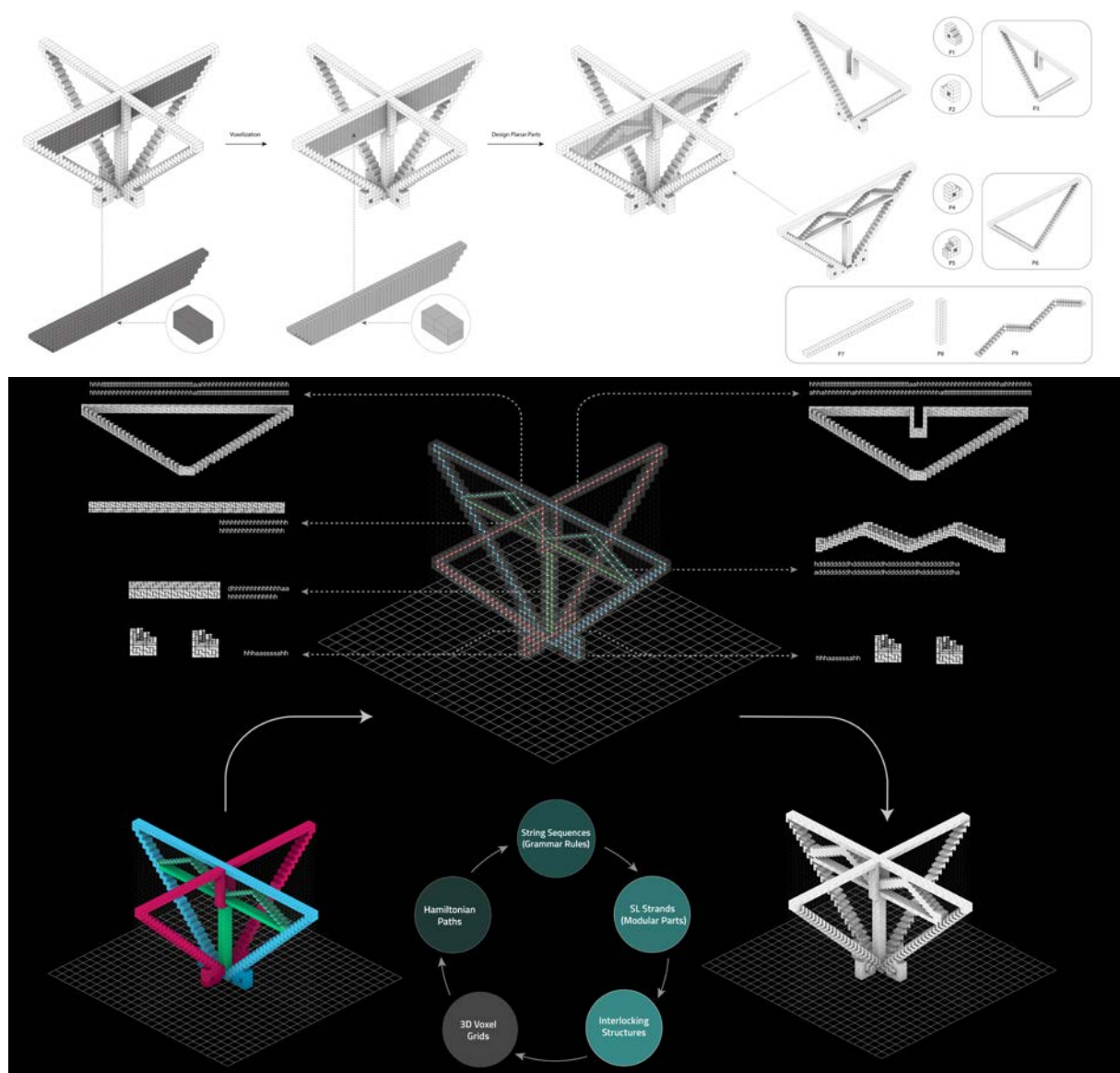


Quelle: Team DDU&IAS, TU Darmstadt 2023

gen und borromäische Knoten) zur Gesamtkonstruktion gefügt werden. Der Demonstrator wurde zunächst durch Voxel in einem Raumraster beschrieben, das je nach lokalen Bedarfen in seiner Auflösung erhöht wurde (vgl. Abbildung 73/74). Nachdem die einzelnen Bauteile als diskrete Kuben beschrieben wurden haben wir alle voxelbasierten Teile in eine SL-Block-Baugruppe übersetzt. Mit Hilfe des Algorithmus zur hamiltonschen Pfadfindung können wir durch das Verbinden der geometrischen Mittelpunkte der Voxel eine geschlossene Polylinie bilden. Anschließend wird die Polylinie in eine Stringsequenz dekodiert, um die entsprechende Zeichenkette für einen SL-Strang zu erzeugen. Die einzelnen Bauteile (Fachwerkträger, Diagonalen, Stütze, Zugurt) wurden mit Robotern automatisiert und autonom aus SL-Blöcken zusammengebaut. Die Fügung dieser großen Bauteile erfolgte manuell (vgl. Abbildung 75).

Abbildung 74

Bildsequenz zur Entwicklung des Demonstrators. Fortsetzung des Prozesses aus Abbildung 73. Die oberen horizontalen Verbindungen werden ausdetailliert. Eine Verbindung wird zu einem Fachwerkträger aus Voxel. Die andere Verbindung verzahnt sich mit dem Fachwerkträger mit einem U-förmigen Details im mittleren Kreuzungspunkt. Im unteren Teil der Abbildung werden die einzelnen Elemente des Demonstrators dargestellt: Diagonalen aus den V-Stützen, Fachwerkträger, Sockel.

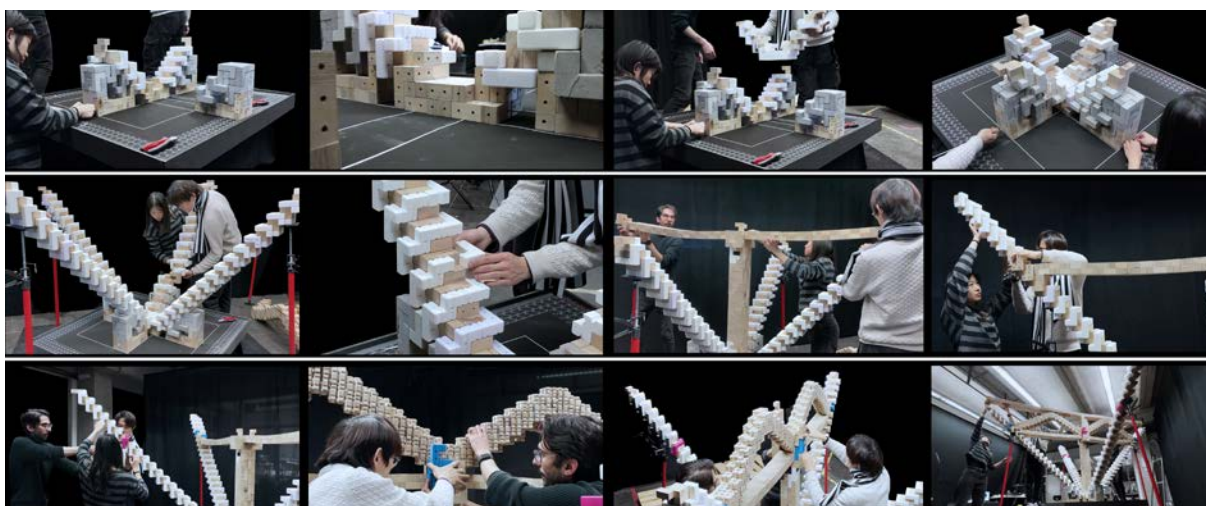


Quelle: Team DDU&IAS, TU Darmstadt 2023

Bei der Konstruktion werden die SL-Blocks-Bauteile Druck-, Zug- und Biegekräften ausgesetzt, um die Leistung des ineinandergreifenden Systems zu testen. Der massive Sockel nimmt Druckkräfte auf, V-förmige Säulen sind auf Biegung belastet und werden durch ein modulares Fachwerk zurückgespannt. Der Fachwerkträger wurde mit unterschiedlichen Ansätzen vorgespannt. Ein durchgängiges Spannkabel im Untergurt verbindet beide Enden des Trägers. Dafür mussten die entsprechende hölzernen SL-Blöcke mit Bohrungen versehen werden. In den Diagonalen des Trägers kamen kurze Spannbänder zum Einsatz, die kleinere Baugruppen miteinander verbinden. Dies war notwendig geworden, weil Fertigungstoleranzen den Formschluss zwischen SL-Blöcken teilweise nicht vollständig und mit ausreichender Reibung hergestellt haben. Es wurde deutlich, dass bestimmte Herstellungsmethoden für diese Konstruktionsform nicht geeignet sind. Im Demonstrator spiegelt die lokale Differenzierung der SL Block-Materialien die unterschiedlichen lokalen Kräfte wider: Druckbeanspruchte Bereiche sind aus Beton gefertigt. Spannende Bereiche bestehen aus leichten SL-Blöcken.

Abbildung 75

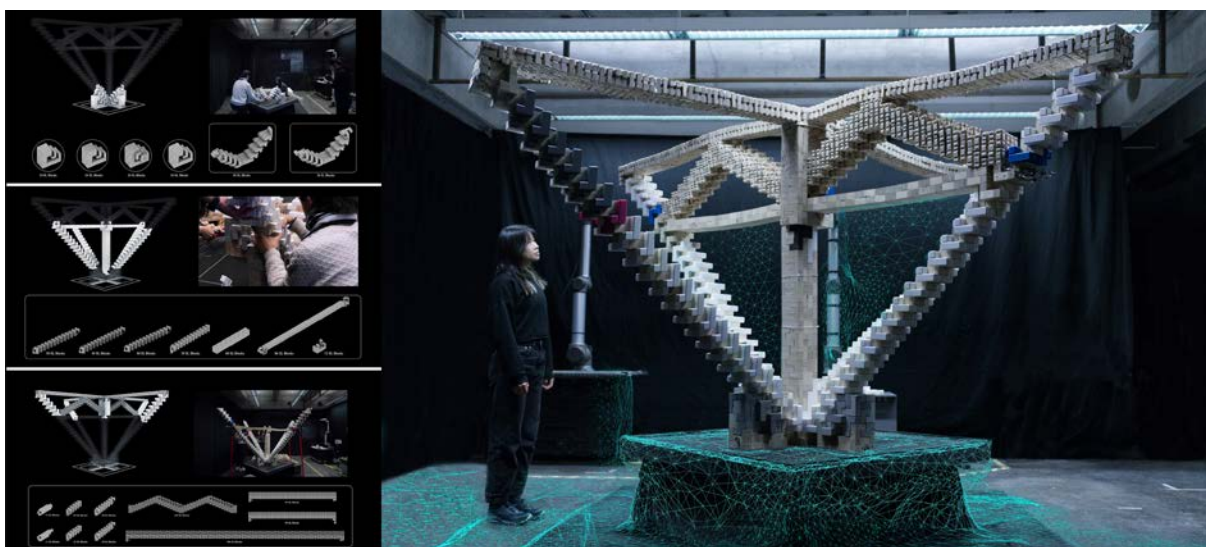
Die Bildsequenz zeigt die manuelle Montage der einzelnen robotisch gefügten Bauteile (Fachwerkträger, Diagonalen, Stütze, Zuggurt).



Quelle: Team DDU&IAS, TU Darmstadt 2023

Abbildung 76

Links werden einzelne Bauteile (Fachwerkträger, Diagonalen, Stütze, Zuggurt, Sockel) als digitale Modelle und als Fotografie während der Montage gezeigt. Rechts ist der fertig montierte Demonstrator zu sehen.



Quelle: Team DDU&IAS, TU Darmstadt 2023

5.2 Bewertung/Diskussion/Zielerreichung

Folgende Ziele konnten im Projekt erreicht werden:

- **Ziel 1:** Wir haben ein prototypisches modulares System entwickelt, das trocken und reversibel durch Roboter montierbar ist. Die Module sind zu unterschiedlichen Großformen assemblierbar (Prinzip Lego, keine parametrischen Unikate). Die Fügeprinzipien ermöglichen die Kombination von Modulen aus unterschiedlichen Materialien. Die Module sind für die robotergestützte Montage optimiert. Die Konstruktion ist im Sinne der Kreislaufwirtschaft reversibel gefügt und durch Roboter (de-)montierbar. Die Demontierbarkeit wurde durch die Mehrfachnutzung von SL-Blöcken in einer Reihe von kleineren Demonstratoren erfolgreich getestet.

Bewertung/Diskussion:

Ziel 1 wurde vollumfänglich erreicht. Im Projekt wurde eine ausführliche Recherche, Analyse und Dokumentation bestehender trocken und reversibel gefügter verzahnter Systeme und ihrer Fügeprinzipien durchgeführt. Auf dieser Basis haben wir uns für den SL-Block als zentrales Grundelement für das Projekt entschieden.

Die multidirektionale Verzahnung der SL-Blöcke ist sehr hilfreich, um trocken und reversibel gefügte Konstruktionen und ihre Bauzustände mit Robotern zu bearbeiten. Die bereits formalisierte Fügegrammatik und die hohe kombinatorische Vielfalt bildeten eine Voraussetzung, um algorithmisch und rechnerbasiert eine hohe Vielfalt und Varianz herzustellen.

Die SL-Blöcke konnten mit unterschiedlichen Verfahren und aus unterschiedlichen Materialien hergestellt werden. Dabei bilden die Geometrie der Teile und die Topologie der Fügung einen gemeinsamen Nenner, der multi-material Aggregationen möglich macht. Wir haben verschiedene Herstellungsmethoden getestet, dabei sind die hohen notwendigen Herstellungspräzision und die komplex verwinkelte Form der Blöcke eine Herausforderung für fast alle Verfahren. Der 3D-Druck kann die Präzision sicherstellen, ist jedoch nicht das optimale Verfahren für ein repetitives Bauteil.

Industrielle, serielle Herstellungsverfahren wurden im Projekt nicht im Detail untersucht, weil sie nicht im Fokus der Forschungsfragen standen. Im Projekt haben wir eine Reihe von Geometrievarianten für die SL-Blöcke entwickelt. Die Varianten verfügten alle über die gleichen Verzahnungsmöglichkeiten, erhielten aber geometrische Veränderungen, die das Greifen und Fügen und Kalibrieren der Bauteile vereinfacht.

Weitere Fragen für zukünftige Forschung entstanden in der Projektbearbeitung:

- i) Welche Fertigungstechnologien sorgen bei vertretbarem Aufwand für die notwendige Präzision von SL-Blöcken?
- ii) Wie kann die Verzahnung von SL Blöcken so weiterentwickelt werden, dass die gesperrten Freiheitsgrade an die jeweilige Einbauposition und die damit verbundenen Lastfälle angepasst ist?
- iii) Wie kann die notwendige Balance zwischen einfacher (De-)montierbarkeit und kontaktreicher Verzahnung optimiert werden?
- iv) Wie kann die SL-Block Geometrie vereinfacht werden, um einen Transfer in die Praxis zu erleichtern?

Während der Projektbearbeitung tauchten erste Unternehmen auf, die aus hölzernen Micro-Modulen größere Bauteile herstellen, die reversibel gefügt werden. Dieses Praxisbeispiel zeigt konstruktive Parallelen

zum Projekt, verfolgt aber auf der Ebene der Micro-Module (hier SL-Blöcke) keine Reversibilität. Dennoch zeigt das Sart-up die Praxisrelevanz solcher Konstruktionsformen. Die Erkenntnisse zur taktilen, robotergestützten Fügung aus dem Projekt bieten hier eine relevante thematische Erweiterung. Weiter bestehen bauphysikalische und mechanische Herausforderungen wie Feuchte- und Temperatureinflüsse, die im Rahmen des Projekts kein Untersuchungsgegenstand waren.

- **Ziel 2:** Wir haben Roboter in die Lage versetzt, modulare architektonische Strukturen autonom zu montieren. Das derzeit übliche Programmieren von Bewegungspfaden sowie der kontaktintensiven Modulplatzierung bei der robotischen Fügung wurde mit Hilfe von Reinforcement Learning und der Integration von visuellen, taktilen und anderen sensorischen Informationen durch autonom agierende Roboter ergänzt.

Bewertung/Diskussion:

Die im ersten Arbeitsschritt zur robotischen Montage entwickelten Algorithmen zur präzisen Beschreibung von Montagesequenzen und der Vorprogrammierung von Aufgaben- und Pfadplanung waren für die Montage von einzelnen Demonstratoren und zum Testen der formoptimierten SL-Blöcke notwendig und eine wichtige Vorarbeit zur autonomen Montage.

Außerdem dienten sie einem vertiefenden Verständnis der tektonischen Eigenschaften von SL-Strands und größeren Aggregationen. Als ersten wichtigen Schritt für die autonome Montage haben wir für den visuell taktilen Greifer einen Klassifikator für Oberflächentexturen trainiert, der in der Lage ist, unterschiedliche SL-Block Materialien zu erkennen. Für die autonome Aufgaben- und Bewegungsplanung haben wir ein graphbasiertes neuronales Netzwerk trainiert. Das Netzwerk konnte alle notwendigen Werte für das Bewegen eines noch nicht platzierten Blocks zu einer leeren Einbauposition vorhersagen.

- **Ziel 3:** Verallgemeinerbarkeit (Generalization) des Reinforcement Learning Im Projekt sollen architektonisch/konstruktive (high-level) Kriterien zu Belohnungen (Rewards) für das Reinforcement Learning (RL) werden. Ziel 3 ist es, den projektspezifischen Erkenntnisgewinn zur effektiven Ausgestaltung der Belohnungen (Reward shaping) beim RL zu verallgemeinern, so dass er für vielfältige Anwendungen des Entwerfens und Bauens nutzbar wird.

Bewertung/Diskussion:

RL kam an drei Stellen im Projekt zum Einsatz:

i) beim Anordnen verzahnter SL-Block-Baugruppen entlang von Referenzkurven. RL konnte im Vergleich mit anderen Optimierungsalgorithmen die gestellte Aufgabe effizient lösen. Wichtiger ist jedoch, dass nach der Modifikation der Kurvenverfolgung durch Einführung von Hindernissen, die das Platzieren von SL-Blöcken verhindern, RL eine Strategie gefunden hat, mit den veränderten Gegebenheiten umzugehen, statt starre und vordefinierte Heuristiken einzusetzen. Es kann genau derselbe Algorithmus auf das geänderte Problem angewendet werden.

ii) bei der Repräsentation, Komprimierung und Extraktion von Bildinformationen aus den visuell taktilen Greifprozessen. Der komplexe experimentelle Aufbau bestehend aus einem 6-Achs Roboter, visuell taktile Sensoren, Objektverfolgung, SL-Block Lager und Montageaufgaben ermöglichte das Sammeln von zahlreichen Episoden taktiler Montageprozesse als Bilddaten, die als Simulation nicht generierbar gewesen wären. Das System erlernte in einem vollständig unbeaufsichtigten Modus die Repräsentation der Bilder taktiler Sensoren. Dafür nutzten wir Autoencoder. Diese künstlichen neuronalen Netze sind in der Lage, die Repräsentation von Daten auf komprimierte Weise zu erlernen und damit wesentliche Bildmerkmale und Informationen für die taktile Steuerung zu extrahieren und zu kodieren. Diese Repräsentation ist in einem gewissen Rahmen auf unterschiedliche taktile Sensoren generalisierbar. Damit

erzielten wir im Projekt einen ersten Schritt zur Erforschung von hochgradig generalisierbaren taktilen Repräsentationen, für die breite Anwendung von visuell taktilen Sensoren in Architektur und Bauwesen.

iii) bei der autonomen Aufgaben- und Bewegungsplanung für die simulierte Montage von Bauteilen zu einer abstrakt beschriebenen Großform. Die Belohnungsfunktionen unterstützen Aktionen, die zu einer dichteren Elementpackung führen. Der Bauprozess ist beendet, wenn das Hüllvolumen bis zu einem Schwellenwert mit Elementen befüllt ist oder keine unplatzierten Blöcke mehr verfügbar sind oder wenn eine ungültige Aktion ausgeführt wird, d.h. eine Aktion, die zu einer instabilen Konstruktion führt, die aktuelle Struktur zerstört oder kinematisch nicht durchführbar ist. Der Ansatz des modellfreien verstärkenden Lernens bei der Aggregation von Bauteilen zu einer gewünschten Zielform ist generalisierbar. Der Algorithmus arbeitet ohne konkretes Umgebungsmodell. Er strebt lediglich die Minimierung der Fehlerfunktion an, z.B. die Lagerposition eines SL-Blocks und seine gewünschte Position in der SL-Block Aggregation. Graphbasierte Darstellung von Montageszenen funktioniert unabhängig von der Anzahl der Bauteile, so dass dieselbe gelernte Graphendarstellung verwendet wird, um Montageaufgaben mit unterschiedlichen Mengen von verfügbaren Bauteilen und verschiedenen Zielstrukturen zu bewältigen. Damit wurde ein erster Schritt in Richtung einer Generalisierung des Gelernten erreicht.

Alle Teilziele sind durch den Bau eines ca. 2 m hohen und 3 m langen Demonstrator evaluiert worden. Verschiedene Konstruktionstypologien (Last abtragen, Spannen, Kragen) und die Integration unterschiedlicher Eigenschaften und Funktionen wurden veranschaulicht. Der Demonstrator wurde aus unterschiedlichen Materialien modular aufgebaut. Roboter bauten aus SL-Blöcken SL-Strands, die als Bauteile wie Stützen, Trägergurte und Diagonalen formten. Die Montage dieser größeren Elemente erfolgte manuell (siehe auch Abbildung 76).

Die in 4.2 erläuterten Bauteilhierarchien, bei denen nicht, wie ursprünglich geplant, die SL-Blöcke größer dimensioniert werden, sondern große Bauteile aus vielen kleinen SL-Blöcken geformt werden, hatten auch Einfluss auf die robotische Montage.

Im Projekt haben wir uns darauf konzentriert, die arbeitsaufwendige Montage vieler kleiner Elemente zu automatisieren, während wir davon ausgehen, dass die daraus entstehenden größeren Einheiten in der Praxis wieder über konventionelle Hebetekniken bewegt und verbaut werden. Die reversiblen Verbindungen der Module erlaubten später den einfachen Rück- und Umbau. Die trockenen Fügungen werden kraftschlüssig verbunden und gehen damit über das einfache Stapeln von Bauteilen hinaus.

5.3 Ergebnistransfer

Während der Projektlaufzeit wurde Zwischen- und Endergebnisse auf Konferenzen, in fachbegutachteten Zeitschriften und auf Vorträgen und Präsentation vorgestellt:

Konferenzbeiträge/-präsentationen:

- Designing for Robotic (Dis-)Assembly, Advances in Architectural Geometry 2023, University of Stuttgart, 06-07.10.2023.
- Auto(mated)nomous Assembly. International Conference on Trends on Construction in the Post-Digital Era, Guimarães, Portugal, 06.-09.09.2022.
- Building a Library of Tactile Skills Based on FingerVision. 19th International Conference on Humanoid Robots (Humanoids), Toronto.

- Reinforcement Learning for Sequential Assembly of SL-Blocks - Self-interlocking combinatorial design based on Machine Learning. The 39th eCAADe Conference, Novi Sad.

Beiträge in Fachzeitschriften:

- Robotic architectural assembly with tactile skills: Simulation and optimization, In: Automation in Construction.

Vorträge:

- Human-Robot Interaction for Post-Carbon Architecture (HRI4PCA), LMU Hannover 29.4.2022.
- digitalBAU Köln, 2.6.2022.
- 20. Projektetage der Bauforschung, 22.6.2022.
- ENSA Versaille, Vortragsserie digital Holzbau, 27.6.2022.
- Symposium Robots that Built - The Extension of Men (Dutch Design Week 2022), 20.10.2022.
- SEA City Conversations (Vortragsreihe), Mumbai (Online Vortrag), 20.01.2023.
- BAU München Heinze/Bauwelt, 18.04.2023.
- CITA (KADAK), Copenhagen, Research Metting, 06.06.2023.
- Symposium La construction hors site face à l'anthropocène, Paris, 19.10.2023.
- Festvortrag, Verbandstreffen baugewerblicher Unternehmer Hessen e.V. (VbU) , 03.11.2023.
- UDK Berlin, Entwurfskurz Gast-Prof. G. Betti, 15.11.2023.
- Ringvorlesung der Universität Stuttgart Fachbereich Architektur, 22.11.2023.

Posterpräsentation:

- The Future of Construction – Symposium an der ETH Zurich.

Bücher:

- Print! Architecture.
- Building Better - Less - Different: Clean Energy Transition and Digital Transformation.

6. Fazit & Ausblick

Tactile Robotic Assembly konnte die gesteckten Ziele erreichen und hat als Grundlagenforschungsprojekt eine Reihe von neuen Fragen für zukünftige Projekte aufgeworfen.

■ Elemente:

SL-Blöcke sind in vielerlei Hinsicht geeignet für trocken gefügte reversible Konstruktionen. Ihre kombinatorische Vielseitigkeit und die komplexe Verzahnungslogik ermöglichen gestalterische Vielfalt und stabile Bauzustände, die fast ohne temporäre Stützkonstruktionen auskommen. Diese Eigenschaften resultieren aus der komplexen Geometrie der Blöcke, die ihre Herstellung aufwändig macht. Eine industrielle, serielle Produktion der SL-Blöcke ist möglich, da keine parametrische Ausdifferenzierung der Teile vorliegt. Es könnten somit wenige Formen/Schalungen wiederverwendet werden oder andere Herstellungsprozesse auf eine Geometrie standardisiert und optimiert werden.

Jedoch ist eine hohe Ausdifferenzierung unterschiedlicher Materialien gewünscht. SL-Block-Konstruktionen sollen die unterschiedlichen architektonischen und konstruktiven Anforderungen an ein Gebäude durch Blöcke erreichen, die mit unterschiedlichen Materialien unterschiedliche Eigenschaften mit sich bringen. Somit müsste die komplexe Geometrie in unterschiedlichsten Verfahren hergestellt werden. Bei allen Herstellungsverfahren ist auf hohe Präzision zu achten, so dass sich Toleranzen nicht kumulieren und die Kraftweiterleitung zwischen den Blöcken flächig funktioniert.

Zukünftige Forschung könnte darauf abzielen, die geometrische Komplexität der Blöcke zu reduzieren und gleichzeitig Verzahnungen zu gewährleisten, die für architektonische Konstruktionen notwendig sind. Parallel zur Projektbearbeitung konnten wir beobachten, wie Ideen zur trockenen und reversiblen Fügung von "micro-modularen" Bauteilen nicht nur in der Forschung, sondern auch in der Praxis auftauchen. Reversibles Klinkermauerwerk und die Verwendung von Holzverschnitt zum Bau von Holzbausteinen sind neue Systeme, für die zukünftig robotergestützte Montage relevant werden könnte.

■ Konstruktion:

Die intensive Auseinandersetzung mit bottom-up und top-down Entwurfsstrategien und die Implementierung von Fügeregeln in digitale Werkzeuge erlaubt die Nutzung von SL-Blöcken für neue Konstruktionen und für die Integration von SL-Strands in bestehende Konstruktionen.

Die Automatisierung im Entwurf ist dabei ein wichtiges Hilfsmittel, um das System für Entwerfende zugänglich zu machen, die sich nicht mit den komplexen Details der Verzahnung auseinandersetzen möchten, aber an kreislaufgerechten, reversiblen Konstruktionen interessiert sind.

Die geometrische und topologische Methodik in Graphen abzubilden war eine entscheidende Vorarbeit für die automatisierte und autonome Montage von Bauteilen. Das Repräsentieren und Analysieren von Bauteil-Freiheitsgraden während der Montage bildet eine wichtige Vorarbeit, um in zukünftigen Projekten die Verzahnungslogik der Blöcke weiterzuentwickeln und praxistauglicher zu gestalten. Weitere Forschung hinsichtlich des mechanischen Verhaltens von Blöcken und den daraus resultierenden Konstruktionen ist notwendig.

■ Visuell taktile Sensorik/Autonome Robotik:

Wir konnten zeigen, dass der Tastsinn zum Erlernen von Montagestrategien verwendet werden kann. Bauteiloberflächen verschiedener Materialien konnten zuverlässig identifiziert werden. Der gesamte Montageprozess erfordert jedoch große Mengen an zeitaufwändig zu erzeugenden Versuch-und-Irrtum-Daten, die nicht aus Simulationen, sondern auch aus physischen Experimenten gewonnen werden müssen. Die Simulation spielt jedoch eine entscheidende Rolle bei der Entwicklung von Roboterlösungen. Eine große Herausforderung bei der autonomen Montage ist die Schwierigkeit der Kontaktsimulation (Gelpad - SL-Block).

Weitere Forschung in der generativen Modellierung ist notwendig, um detailliertere und realistischere Modelle der Objektinteraktion zu erstellen. Die Verwendung solcher Simulationen für das Training von Roboterstrategien wird den Forschungs- und Entwicklungsprozess erheblich beschleunigen. Die Montageprozesse lassen sich derzeit nur mit zusätzlichem Training auf verschiedene Verbindungstypen übertragen. Eine Verallgemeinerung bleibt eine Herausforderung. Zukünftig werden modularere und hierarchischere Lernmethoden benötigt, um über verschiedene Blöcke und Montagearten hinweg zu lernen.

Grundlegende Vision-Language-Action-Modelle (VLA) bieten eine vielversprechende Richtung, um eine Generalisierung über verschiedene Aufgaben hinweg zu erreichen, indem sie Schlussfolgerungen auf Objektebene ziehen, anstatt - wie beim DIGIT Sensor - eine Steuerung durchzuführen, die auf der Übersetzung von Pixelbild in Drehmoment basiert. In ähnlicher Weise müssen Ansätze für die Einbeziehung der taktilen Wahrnehmung in Basismodelle weiter untersucht werden.

Eine wichtige zukünftige Forschungsfrage ist das kollaborative Arbeiten von Menschen und Robotern. Die Basis dafür bilden autonom agierende Roboter, weil sie ihre Umwelt "wahrnehmen" und eigene Entscheidungen fällen können. Damit ist Kollaboration "auf Augenhöhe" erst möglich.

7. Danksagung und Mitwirkende

Diese Arbeit und die ihr zugrundeliegende Forschung wäre ohne die außerordentliche Unterstützung der "Zukunft Bau Forschungsförderung" des Bundesministeriums für Wohnen, Stadtentwicklung und Bauwesen (BMWSB) nicht möglich gewesen. Ein großer Dank richtet sich außerdem an die Studierenden der TU Darmstadt, die das Projekt tatkräftig unterstützt haben

Autorinnen und Autoren

Prof. Dr.- Ing. Oliver Tessmann (Fachgebiet Digitales Gestalten, TU Darmstadt)
Prof. Jan Peters, Ph.D. (Intelligente Autonome Systeme, TU Darmstadt)
M.Sc. Yuxi Liu (Fachgebiet Digitales Gestalten, TU Darmstadt)
M.Sc. Mehrzad Esmaeili Charkhab (Fachgebiet Digitales Gestalten, TU Darmstadt)
Dr. Boris Belousov (DFKI GmbH)
M.Sc. Tim Schneider (Intelligente Autonome Systeme, TU Darmstadt)

Weitere Mitwirkende

M.Sc. Niklas Funk (Intelligente Autonome Systeme, TU Darmstadt)
Prof:in Georgia Chalvatzaki, Ph.D (Intelligente Robotersysteme für Assistenz (iROSA), TU Darmstadt)

Fachliche Betreuung

Dr.-Ing. Michael Brüggemann
BRÜGGEMANN KISSELER INGENIEURE, Mettingen



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Gefördert durch:



aufgrund eines Beschlusses
des Deutschen Bundestages



Kurzbiographien



Prof. Dr.- Ing. Oliver Tessmann

Oliver Tessmann ist Architekt und seit 2015 Professor für Digitales Gestalten am Fachbereich Architektur der Technischen Universität Darmstadt. Oliver Tessmann forscht und lehrt im Bereich des computerbasierten Entwerfens, der digitalen Fabrikation, und der Robotik. Nach seinem Diplom an der Universität Kassel arbeitete er für Coop Himmelblau und andere Architekten in Mexico, Wien und Frankfurt.



Prof. Jan Peters, Ph.D.

Professor Jan Peters lehrt und forscht sowohl am Fachbereich Informatik der Technischen Universität Darmstadt als auch am Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI). Er gehört im Bereich des Maschinellen Lernens für autonome, intelligente Roboter zu den weltweit führenden Forschern.



M.Sc. Yuxi Liu

Yuxi Liu ist wissenschaftlicher Mitarbeiter an der Digital Design Unit (DDU). Im Jahr 2020 begann er mit der Erforschung des Einsatzes von Interlocking-Systemen in der Architektur. Seine derzeitige Forschung konzentriert sich auf die autonome Roboter- montage mit Hilfe von künstlicher Intelligenz sowie auf die Entwicklung von topologisch verzahnten Systemen für modulare Konstruktion und Fügung.



M.Sc. Mehrzad Esmaeili Charkhab

Mehrzad Esmaeili war wissenschaftlicher Mitarbeiter auf dem Gebiet des Computational Design und der digitalen Fabrikation an der Technischen Universität Darmstadt. Er arbeitet jetzt als wissenschaftlicher Mitarbeiter im Fachgebiet von Roboter- gestützte Fertigung der gebauten Umwelt an der TU Berlin.



Dr. Boris Belousov

Boris Belousov hat sein Studium abgeschlossen, bleibt dem IAS aber als Senior Rese- archer am Deutschen Forschungszentrum für KI (DFKI), Forschungsbereich, erhalten: System-KI für Roboter Lernen. Er hat sich mit Maximum-Entropie-Verstärkungslernen, risikosensitiver Politiksuche, aktivem Lernen, verteilungsmäßig robuster Trajektori- enoptimierung, Domänen-Randomisierung und visuotakter Manipulation beschäft- igt.



M.Sc. Tim Schneider

Tim Schneider ist seit dem 1. November 2021 als Doktorand im Labor für Intelligente Autonome Systeme tätig. Während seiner Doktorarbeit forscht Tim an der Schnittstel- le zwischen intrinsischer Motivation und Roboter- manipulation. Er stellt sich vor, dass Roboter, ähnlich wie Kinder, in der Lage sein sollten, ihre Umgebung haptisch zu er- kunden und durch physische Interaktion ein Verständnis für sie zu entwickeln.

8. Literaturverzeichnis

Aaauanastas (2021, 25. April). The Flat Vault. <https://aauanastas.com/project/the-flat-vault/>.

Ariza, I., & Gazit, M. (2015a). On-site Robotic Assembly of Double-curved Self-supporting Structures. SIGRADI 2015 [Proceedings of the 19th Conference of the Iberoamerican Society of Digital Graphics - Vol. 2, 746–753. <https://doi.org/10.5151/despro-sigradi2015-110316>.

Augugliaro, F., Lupashin, S., Hamer, M., Male, C., Hehn, M., Mueller, M. W., Will, J.S., Gramazio, F., Kohler, M., D'Andrea, R. (2014). The flight assembled architecture installation: Cooperative construction with flying machines. *IEEE Control Systems Magazine*, 34(4), 46-64.

Belousov, B., Sadybakasov, A., Wibranek, B., Veiga, F., Tessmann, O., & Peters, J. (2019). Building a Library of Tactile Skills Based on FingerVision. In *Proceedings of the 2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*. IEEE.

Bock, T. (2015). The future of construction automation: Technological disruption and the upcoming ubiquity of robotics. *Automation in Construction*, 59, 113–121. <https://doi.org/10.1016/j.autcon.2015.07.022>.

Bock, T. (2017). Robotic Assembly System for Computer Integrated Construction. *Proceedings of the 13th International Symposium on Automation and Robotics in Construction*. <https://doi.org/10.22260/ISARC1996/0022>.

Bock, T.-A. (1988). Robot-Oriented Design. 5th ISRC. <https://doi.org/10.1017/CBO9781139924146>.

Bonwetsch, T. (2015). Robotically assembled brickwork: Manipulating assembly processes of discrete elements (Doctoral dissertation, ETH Zurich).

Brehm, E. (2019). Robotertechnik für den Mauerwerksbau – Internationaler Status und Überlegungen für den deutschen Markt. *Mauerwerk*. <https://doi.org/10.1002/dama.201900004>.

Calandra, R., Owens, A., Jayaraman, D., Lin, J., Yuan, W., Malik, J., ... & Levine, S. (2018). More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robotics and Automation Letters*, 3(4), 3300-3307.

Dakhli, Z., & Lafhaj, Z. (2017). Robotic mechanical design for brick-laying automation. *Cogent Engineering*. <https://doi.org/10.1080/23311916.2017.1361600>.

Doyle, S. E.; Hunt, E. L. (2019). Dissolvable 3D Printed Formwork, doi: <https://doi.org/10.52842/conf.acadia.2019.178>.

Estrin, Y (2014). Topological Interlocking and Osteomorphic Blocks. In G. Fallacara & V. Minenna (Eds.), *Stereotomic Design—Catalogo della mostra omonima per Inside Marmomacc and Abitare il Tempo Veronafiere* (Edizioni G, pp. 11–13). Verona: Maglie.)

Estrin, Y., Dyskin, A. V., & Pasternak, E. (2011). Topological interlocking as a material design concept. *Materials Science and Engineering: C*, 31(6), 1189-1194.

Ewerton, M., Maeda, G., Koert, D., Kolev, Z., Takahashi, M., & Peters, J. (2019). Reinforcement learning of trajectory distributions: Applications in assisted teleoperation and motion planning. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*.

Fallacara, G., & Barberio, M. (2018). Parametric Morphogenesis, Robotic Fabrication, and Construction of Novel Stereotomic Hypar Morphologies: Hypar Gate, Hypar Wall, and Hypar Vault. In *Handbook of Research on Form and Morphogenesis in Modern Architectural Contexts* (pp. 329-353). IGI Global.

Fascetti, A., Latteur, P., & Lim, S. H. (2021). Ground-based automated construction of droxel structures: An experimental approach. *Automation in Construction*, 131, 103899.

Fishel, J. A., & Loeb, G. E. (2012, June). Sensing tactile microvibrations with the BioTac—Comparison with human sensitivity. In *2012 4th IEEE RAS & EMBS international conference on biomedical robotics and biomechatronics (BioRob)* (pp. 1122-1127). IEEE.

Funk, N., Chalvatzaki, G., Belousov, B., & Peters, J. (2022, January). Learn2assemble with structured representations and search for robotic architectural construction. In *Conference on Robot Learning* (pp. 1401-1411). PMLR.

Funk, N., Menzenbach, S., Chalvatzaki, G., & Peters, J. (2022, October). Graph-based Reinforcement Learning meets Mixed Integer Programs: An application to 3D robot assembly discovery. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 10215-10222). IEEE.

Gershenfeld, N. (2012). How to make almost anything: The digital fabrication revolution. *Foreign Aff.*, 91, 43.

Gershenfeld, N., Carney, M., Jenett, B., Calisch, S., & Wilson, S. (2015). Macrofabrication with digital materials: Robotic assembly. *Architectural design*, 85(5), 122-127.

Goessens, S., Mueller, C., & Latteur, P. (2018). Feasibility study for drone-based masonry construction of real scale structures. In *Automation in Construction*. <https://doi.org/10.1016/j.autcon.2018.06.015>.

Goessens, S., Mueller, C., & Latteur, P. (2018). Feasibility study for drone-based masonry construction of real-scale structures. *Automation in Construction*, 94, 458-480.

Hartmann, V. N., Oguz, O. S., Driess, D., Toussaint, M., & Menges, A. (2020). Robust task and motion planning for long-horizon architectural construction planning. *arXiv preprint arXiv:2003.07754*.

Hiller, J.; Lipson, H. (2009) Design and analysis of digital materials for physical 3D voxel printing. *Rapid Prototyping Journal*.

Hogan, N. (1984, June). Impedance control: An approach to manipulation. In *1984 American control conference* (pp. 304-313). IEEE.

Jenett, B., Abdel-Rahman, A., Cheung, K., & Gershenfeld, N. (2019). Material-robot system for assembly of discrete cellular structures. *IEEE Robotics and Automation Letters*, 4(4), 4019-4026. ISO 690.

Kappassov, Z., Corrales, J. A., & Perdereau, V. (2015). Tactile sensing in dexterous robot hands. *Robotics and Autonomous Systems*, 74, 195-220.

Latteur, P., Goessens, S., Mueller, C., T., *Masonry construction with drones*, Conference: International Association for Shell and Spatial Structures (IASS) Symposium 2016 At: Tokyo.

Leder, S., Weber, R., Bucklin, O., Wood, D., & Menges, A. (2019). Towards Distributed In-Situ Robotic Timber Construction. In *Research Culture in Architecture*. Birkhäuser Verlag.

Lehmberg, S., Ledderose, L., Wirth, F., Budelmann, H., & Kloft, H. (2016). Light-Weight Structures made of thin walled, dry jointed surface elements made of UHPFRC. *BETON-UND STAHLBETONBAU*, 111(12), 806-815.

Levine, S., Finn, C., Darrell, T., & Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1), 1334-1373.

Liu, Y., Belousov, B., Funk, N., Chalvatzaki, G., Peters, J., & Tessmann, O. (2022). Auto (mated) nomous Assembly. In *Trends on Construction in the Digital Era: Proceedings of ISIC 2022* (pp. 167-181). Cham: Springer International Publishing.

Mainka, J., Lehmberg, S., Budelmann, H., & Kloft, H. (2013). Non-Standard Fügeprinzipien für leichte Bauteile aus UHPFRC. *Beton-und Stahlbetonbau*, 108(11), 763-773.

Mather, A., Cipra, R., & Siegmund, T. (2012). Structural integrity during remanufacture of a topologically interlocked material. *International Journal of Structural Integrity*.

Melenbrink, N., Werfel, J., & Menges, A. (2020). On-site autonomous construction robots: Towards unsupervised building. *Automation in Construction*, 2020, 119. Jg., S. 103312 119(June). <https://doi.org/10.1016/j.aut-con.2020.103312>.

Murphy, S. O. (1967). Interlocking block building construction (Patent No. US3305982A).

Naboni, R., & Kunic, A. (2019). A computational framework for the design and robotic manufacturing of complex wood structures. In *Architecture in the age of the 4th industrial revolution: proceedings of the 37th eCAADe and 23rd SIGraDi Conference, Porto* (pp. 189-196).

Osa, T., Pajarinen, J., Neumann, G., Bagnell, J. A., Abbeel, P., & Peters, J. (2018). An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2), 1-179.

Ossoinack, A. (1955). Interlocking Hollow Building Block (Patent No. U.S. Patent No. 2,703,487).

Parascho, S., Gandia, A., Mirjan, A., Gramazio, F., & Kohler, M. (2017). Cooperative Fabrication of Spatial Metal Structures.

Petersen, K., Nagpal, R., *Complex Design by Simple Robots: A Collective Embodied Intelligence Approach to Construction*, 05 July 2017, <https://doi.org/10.1002/ad.2194>.

Retsin, G. (2016). Discrete assembly and digital materials in architecture. In *Complexity & Simplicity Proceedings of the 34th eCAADe Conference* (Vol. 1, pp. 143-151).

Retsin, G. (2019). Discrete architecture in the age of automation. *Architectural Design*, 89(2), 6–13. <https://doi.org/10.1002/ad.2406>.

Robeller, C., Helm, V., Thoma, A., Gramazio, F., Kohler, M., & Weinand, Y. (2017). Robotic Integral Attachment. *FABRICATE 2017*.

Rogeanu, N., Tiberghien, V., Latteur, P., & Weinand, Y. (2020). Robotic Insertion of Timber Joints using Visual Detection of Fiducial Markers. *Proceedings of the 37th International Symposium on Automation and Robotics in Construction (ISARC)*. <https://doi.org/10.22260/isarc2020/0068>.

Sanchez, J. (2014). Polyomino: Reconsidering serial repetition in combinatorics. In *ACADIA* (Vol. 14, pp. 91-

100).

Sanchez, J. (2016). Combinatorial design: Non-parametric computational design strategies. In *ACADIA // 2016: POSTHUMAN FRONTIERS: Data, Designers, and Cognitive Machines Proceedings of the 36th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA) Ann Arbor 27-29 October, 2016*, pp. 44-53.

Sanchez, J. (2019). Architecture for the commons: Participatory systems in the age of platforms. *Architectural Design*, 89(2), 22–29. <https://doi.org/10.1002/ad.2408>.

Sartoretti, G., Wu, Y., Paivine, W., Kumar, T. S., Koenig, S., & Choset, H. (2019). Distributed reinforcement learning for multi-robot decentralized collective construction. In *Distributed autonomous robotic systems* (pp. 35-49). Springer, Cham.

Shih, S. G. (2018). The Art and Mathematics of Self-Interlocking SL Blocks. In *Bridges 2018 Conference Proceedings* (pp. 107-114). Tessellations Publishing.

Shih, S. G.: On the Hierarchical Construction of SL blocks—A Generative System That Builds Self-interlocking Structures. In: *Advances in Architectural Geometry 2016*, Adriaenssens, S., Gramazio, F., Kohler, M., Menges, A., & Pauly, M. (Eds.), p124-136 (2016).

Shih, Shen-Guan. "Grammars of Interlocking SL blocks." In *Proceedings of Advances in Architectural Geometry 2020*.

Shih, Shen-Guan. "The art and mathematics of self-interlocking SL blocks." *Proceedings of Bridges 2018: Mathematics, Art, Music, Architecture, Education, Culture*. 2018.

Song, P., Fu, C. W., & Cohen-Or, D. (2012) Recursive interlocking puzzles. *ACM Transactions on Graphics (TOG)*, 31(6), 1-10.

Song, P., Fu, C. W., Jin, Y., Xu, H., Liu, L., Heng, P. A., & Cohen-Or, D. (2017). Reconfigurable interlocking furniture. *ACM Transactions on Graphics (TOG)*, 36(6), 1-14.

Song, Peng, Chi-Wing Fu, and Daniel Cohen-Or. "Recursive interlocking puzzles." *ACM Transactions on Graphics (TOG)* 31.6 (2012): 1-10.

Stumm, S., Braumann, J., von Hilchen, M. v, and Brell-Cokcan, S. (2017). On-site robotic construction assistance for assembly using a-priori knowledge and human-robot collaboration. *Advances in Robot Design and Intelligent Control* (p. 583–592).

Tian, S., Ebert, F., Jayaraman, D., Mudigonda, M., Finn, C., Calandra, R., & Levine, S. (2019, May). Manipulation by feel: Touch-based control with deep predictive models. In *2019 International Conference on Robotics and Automation (ICRA)*(pp. 818-824). IEEE.

Van Hoof, H., Chen, N., Karl, M., van der Smagt, P., & Peters, J. (2016, October). Stable reinforcement learning with autoencoders for tactile and visual data. In *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 3928-3934). IEEE.

Vomhof, M., Vasey, L., Gramazio, F., Kohler, M., Zurich, E. T. H., Bräuer, S., & Ag, R. (2014). Robotic Fabrication of Acoustic Brick Walls. *Acadia, Figure 1*, 555–564.

- Weizmann, M., Amir, O., & Grobman, Y. J. (2021). The effect of block geometry on structural behavior of topological interlocking assemblies. *Automation in Construction*, 128, 103717.
- Wells, A. M., Dantam, N. T., Shrivastava, A., & Kavraki, L. E. (2019). Learning feasibility for task and motion planning in tabletop environments. *IEEE Robotics and Automation Letters*, 4(2), 1255-1262.
- Wibranek, B., Tessmann O., *Interfacing architecture and artificial intelligence, The Routledge Companion to Artificial Intelligence in Architecture*, London, P. 486 (2021).
- Wibranek, B.; Belousov, B. ; Sadybakasov, A. ; Peters, J. ; Tessmann, O. (2019): Interactive Structure - Robotic Repositioning of Vertical Elements in Man-Machine Collaborative Assembly through Vision-Based Tactile Sensing. In: *Architecture in the Age of the 4th Industrial Revolution - Proceedings of the 37th eCAADe and 23rd SIGraDi Conference - Volume 2*, S. 705-713.
- Wu, K., & Kilian, A. (2018). Robotic equilibrium: Scaffold free arch assemblies. *Recalibration on Imprecision and Infidelity - Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture, ACADIA 2018*.
- Yamaguchi, A.; Atkeson, C. G. (2017) Implementing tactile behaviors using fingervision. In *IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 241–248.
- Yousef, H., Boukallel, M., & Althoefer, K. (2011). Tactile sensing for dexterous in-hand manipulation in robotics—A review. *Sensors and Actuators A: physical*, 167(2), 171-187.
- Youssef, H., Erler, M., & Jäger, W. (2019). Dry laying of masonry to build demountable, highly energy-efficient prototype houses—First findings from a currently running research project. *Mauerwerk*, 23(4), 265-275.
- Yuan, W., Dong, S., & Adelson, E. H. (2017). Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12), 2762.
- Zhang, Y., Koga, Y., & Balkcom, D. (2021). Interlocking Block Assembly With Robots. *IEEE Transactions on Automation Science and Engineering*, 18(3), 902-916.
- Ziqi Wang, Peng Song, Mark Pauly, "State of the Art on Computational Design of Assemblies with Rigid Parts" 04 June 2021.

9. Abbildungsverzeichnis

Abbildung 1 Ausschnitt aus einer Recherchetabelle zu trocken gefügten Konstruktionsprinzipien und ihren Eigenschaften. Spalte 1: Wachsmann Kubus, bestehend aus 12 identischen verzahnten Kuben. Spalte 2: Offene Fingerzinkung. Spalte 3: Dreidimensionale Überblattung. Spalte 4: Dreidimensionale Hakenblattung. Spalte 5: Klemmbausteine. Spalte 6: Topologisch verzahnte Tetraeder. Spalte 7: Räumliche Packung von vier verschiedenen verzerrten Tetraedern (Danzer projectiles). Spalte 8: Räumliche Packung von Polyedern. Spalte 9: Aggregierte, discrete Elemente. Spalte 10 SL-Blöcke. **10**

Abbildung 2 Links: Tetraeder, die sich gegenseitig stützen. Je mehr Objekte miteinander verzahnt werden, umso weniger Freiheitsgrade hat der mittlere Tetraeder. Rechts: Ein 3D gedrucktes Modell einer topologisch verzahnten Struktur mit parametrisch ausdifferenzierten Elementen. **11**

Abbildung 3 Geometrische Systematik des SL-Block Systems. Links: Vier Kuben bilden ein "L", vier Kuben bilden ein "S". Mitte: Zusammen entsteht ein SL-Block aus acht Kuben. Rechts: Zwei verzahnte SL-Blöcke bilden ein konjugiertes Paar. **18**

Abbildung 4 Oben links: eine Innenaufnahme aus dem Fachbereich Architektur der TU Darmstadt, die den Vergleich zwischen dem traditionellen vorgefertigten Betonbalken und einem Entwurf zur Diskretisierung eines Trägers in SL-Blöcke zeigt. Bauteile, die bisher als massiv und monolithisch betrachtet wurden, können nun durch viele kleine SL-Blöcke zusammengefügt werden. Diese kleinen Blöcke können zu verschiedenen Gebäudekonfigurationen zusammengesetzt, demontiert und wieder zusammengebaut werden, was eine temporäre oder dauerhafte Nutzung von Bauelementen ermöglicht. Oben rechts ist ein Diagramm zu sehen, das die reversible Konstruktion der SL-Blöcke veranschaulicht. Große Mengen von SL-Blöcken dienen als Materiallager, das für die Konstruktion von Wänden, Stützen und Decken verwendet werden kann. Daraus entsteht ein Kreislauf der Wiederverwendung. Unten: Entwurf zur Umgestaltung eines konventionellen Wandaufbaus in ein System aus SL-Blöcken. SL-Blöcke aus unterschiedlichen Materialien übernehmen die unterschiedlichen Funktionen einer Wand. Ihre gemeinsame Geometrie erlaubt jedoch stets die Verzahnung. **19**

Abbildung 5 Die Kombination der SL-Blöcke basiert auf der Rotation und Translation eines konjugierten Paares. Im Rahmen der Definition von h engagement dreht sich das konjugierte Paar zunächst um 180 Grad um die Z-Achsen. Nach der Drehung wird es um zwei Einheiten entlang der x-Achse verschoben. Die sich aus der Drehung und Verschiebung ergebende Geometrie kann mit dem anfänglichen konjugierten Paar ineinandergreifen und eine Kombination bilden. Die Drehung und die Verschiebung können zu einer Transformationsmatrix zusammengefasst werden, die mit dem Buchstaben "h" bezeichnet wird. **20**

Abbildung 6 SL-Blöcke und die sechs verschiedenen Möglichkeiten der Verzahnung (Engagements) und die daraus entstehenden Formen: h engagement: lineare Anordnung, a engagement geschlossener Kubus aus 8 SL-Blöcken, d engagement: getreppte lineare Anordnung, s engagement: überlappend getreppte Anordnung, t engagement: diagonal, linear, stapelbar, y engagement: vertikal, spiralisierend. **21**

Abbildung 7 Beispiele komplexer, algorithmisch generierter SL Strukturen. Oben: Zwei ringförmige SL-Strands bilden zusammengesteckt einen Oktaeder, der in einem Raumraster repliziert werden kann. Unten: Lineare SL-Strands bilden fachwerkartige Strukturen, die räumlich miteinander gefügt werden können und eine Art räumliches Stabwerk aus SL-Blöcken bilden. **22**

Abbildung 8 SL-Strands, die zu größeren Aggregationen miteinander verbunden werden. Drei Arten der Verbindung von SL-Strands: Links: Verzahnte Verbindungen, Mitte: Rekursiv-verschachtelte Verbindungen, Rechts: Borromäische Verbindungen zum Einsatz. **23**

Abbildung 9 Die Tabelle zeigt die SL-engagement Transformationsmatrix mit Werten für Rotationsradien und Translationsvektoren. Rechts sind konjugierte SL-Block Paare und ihre in Grasshopper implementierten SL-Engagement Regeln dargestellt. **24**

Abbildung 10 SL-Strand Erzeugung dargestellt als Python Code in Grasshopper (links), Zuweisung von Zeichen zu Transformationsmatrix (mitte) und Geometrische Repräsentation (rechts). Sequentielle Anwendung der SL-Engagements h, a, d, s, t, y Regeln und Grasshopper und die geometrische Repräsentation in Rhino. **24**

Abbildung 11 Grafische Programmieroberfläche von Grasshopper. Zeichenketten werden in einer eigens entwickelten Grasshopper Python Komponente zu SL-Block Aggregationen übersetzt. **25**

Abbildung 12 Links: Reinforcement Learning Schleife mit Rhino/Grasshopper. Rechts: Zwei Hauptbestandteile der Beobachtung (Observation): Entfernungen von jedem der 16 Voxel des aktuellen Engagements zur Kurve (oben) und Koordinaten von 3 vorauslaufenden Punkten (unten). Anhand dieser zusätzlichen Informationen kann der Agent vorhersehen, wie sich die Kurve in Zukunft entwickeln wird. **26**

Abbildung 13 Evaluation des Verstärkenden Lernen für Top Down Basisaggregationen eines geschlossenen SL-Strands entlang eines geschlossenen Polygons. Farbliche Darstellung der Distanzwerte zwischen SL-Blöcken und Kurve: Weiß: 0-2,5 cm, gelb 2,5-5 cm, orange: 5 – 7,5cm. Rot: größer 7,5 cm. Beim evolutionären Algorithmus sind ca. 50 der Blöcke rot 20% orange, 15% gelb, 15% weiß. Beim Reinforcement Learning sind ca. 50 der Blöcke weiß, 45% gelb, 5% orange. Beim evolutionären Algorithmus sind c. 50 der Blöcke rot 20% orange, 15% gelb, 15% weiß. Beim greedy Algorithmus sind c. 65 der Blöcke weiß, 35% gelb. **27**

Abbildung 14 Versuchsaufbau mit Referenzkurven zur Anordnung von SL-Blöcken und Hindernissen, die das Platzieren von Blöcken nicht erlauben. Vergleich von Blockanordnung von Greedy Algorithmus und Reinforcement Learning zeigt unterschiedliche Lösungen. **29**

Abbildung 15 Digitales und physisches Modell einer SL-Block-Aggregation auf Basis einer dreidimensional ausgebildeten Polylinie. **29**

Abbildung 16 Grafische Darstellung der Elemente-Hierarchie. Bauteile bestehen aus SL-Strands. SL-Strands bestehen aus SL-Blöcken. Bauteile können Träger, Stütze, Wand oder Decke sein. **30**

Abbildung 17 Links: verschiedene SL-Strands und ihre räumliche Entwicklung. Rechts: Aggregationen und Bauteile, die auf der Kombination verschiedener Fügegrammatiken für SL-Strands basieren. **30**

Abbildung 18 Stützen, Wände und Decken aus SL-Blöcken. **31**

Abbildung 19 Entwurf einer Brücke in der verschiedene SL-Strands miteinander verbunden wurden. Rechts ist die Materialdifferenzierung erkennbar, die aus den verschiedenen Lastverteilungen resultiert. Druckbeanspruchte Bereiche werden aus Betonblöcken hergestellt. Spannende Bereiche aus hölzernen SL-Blöcken. **31**

Abbildung 20 Visualisierungen zu Montageszenarien. Bauteile werden von Robotern vormontiert und anschließend auf der Baustelle mit konventionellen Methoden des Fertigteilbaus verarbeitet. **31**

Abbildung 21 Grafische Darstellung der Elemente-Hierarchie. **33**

Abbildung 22 Links: Zusammensetzung der SL-Blöcke aus Kuben. Rechts: Konjugierte SL-Block Paare. **33**

Abbildung 23 Links: Ein weiterer Octacube, mit einer anderen Form als die SL-Blöcke, kann einen SL-Strand abschließen, oder ihn mit linearen Bauteilen (blau) verbinden. **33**

Abbildung 24 Links: Ein weiterer Octacube, mit einer anderen Form als die SL Blöcke, kann einen SL-Strand abschließen, oder ihn mit linearen Bauteilen (blau) verbinden. **34**

Abbildung 25 SL-Strangerzeugung. Links: Eine Zeichenkette für SL-Engagements. Mitte: SL-Strands Erzeugung. Rechts: Darstellung des Prozesses als gerichteter Graph. **35**

Abbildung 26 Vereinfachte Darstellung eines konjugierten Paares als Voxel mit geometrischem Zentrum, an dem sich eine Polylinie ausrichten lässt, die zu einem Graphen weiterentwickelt wird. **35**

Abbildung 27 Sechs SL-Engagements reduziert als Voxeldarstellung. Die Komplexität der SL-Blöcke wird durch die Darstellung einer einfachen quader-artigen Hüllgeometrie reduziert. Wir nutzen zwei Quadergrößen, die bei der Annäherung an und Positionierung des generierten Polylinienpfades genutzt werden können. **36**

Abbildung 28 Eine Liste von Vektoren und ihre Drehwinkel als Zeichenketten und gerichteter Pfad. **36**

Abbildung 29 Übersetzung einer Polylinie in Richtungsvektoren für die SL-Block Anordnung. A) Quadratraster und Polylinie mit Segmenten in 0° , 45° und 90° Orientierung, die jeweils ein Rasterschnittpunkte miteinander verbinden. B) Einfärbung aller Rasterzellen, die an die Polylinie angrenzen ergibt ein flächige Darstellung der Polylinie im Raster. C) Vereinfachung der Polylinie: Die Mittelpunkte der unter b) eingefärbten Rasterzellen werden werden mit Segmenten in 0° oder 90° miteinander verbunden, so dass eine getreppte Polylinie entsteht. D) Ableitung der Richtungsvektoren aus den Segmenten von C). Unten im Bild: Übersetzung der zweidimensionalen Polylinie und ihre Segmentvektoren in Zeichenkette für SL-Engagements. **37**

Abbildung 30 Schrittweise übersetzung einer Polylinie in einen komplexen SL-Strand basierend auf den Prozess in Abbildung 29 und anschließend Übersetzung in 3D SL-Strand. **37**

Abbildung 31 Die Umwandlung eines Pfades in SL-Blöcke basiert auf den engagements "h", "a" und "d". Ein Polylinienpfad aus Voxeln der Größe $2 \times 2 \times 1$ wird berechnet und anschließend in eine Zeichenkette zur Erzeugung des entsprechenden SL-Strands umgewandelt. **38**

Abbildung 32 Graphen-basierte Analyse eines geschlossenen SL-Strands auf Basis von "h" und "a" Engagements. **39**

Abbildung 33 Ein geschlossener SL-Strands erzeugt von "h" und "a" engagements zunächst übersetzt in ein Voxelraster und anschließend gegliedert in Kuben, die aus vier Quadern gebildet werden. **39**

Abbildung 34 Von der Graphendarstellung zum Volumen. Die Vektoren werden an jedem Knoten extrahiert und in eine Zeichenfolge überführt, die für eine Generierung des Volumens aus SL-Blocks genutzt werden kann. Mit diesem Algorithmus können voxel-basierte Volumen (Polycubes) in verzahnte SL-Blocks überführt werden. **40**

Abbildung 35 Von der Graphendarstellung zum Volumen. Die Vektoren werden an jedem Knoten extrahiert und in eine Zeichenfolge überführt, die für eine Generierung des Volumens aus SL-Blocks genutzt werden kann. Mit diesem Algorithmus können voxel-basierte Volumen (Polycubes) in verzahnte SL-Blocks überführt werden. **40**

Abbildung 36 Bebilderung der Einzelschritte zur Erzeugung von SL-Strands auf Basis von geschlossenen Polylinien und Raumrastern. 42

Abbildung 37 Konzeptdiagramm zeigt, wie komplexe SL-Strands aus vielen SL-Blöcken vereinfacht betrachtet als monolithische Großbauteile betrachtet werden können, die über konventionelle Fügeregeln miteinander montiert werden. 42

Abbildung 38 Bildsequenz zur Überführung eines Volumens in eine SL-Block-Aggregation. 1. Startvolumen zeigt eine Art massiven Sessel. 2. Übersetzung des Volumens in Voxel. 3. Übersetzung der Voxel in ein Raumraster. 4. Zerlegung des Raumrasters einzelne, verzahnte Voxelformen. 5. Generierung der geschlossenen Polylinien in den Voxelformen mittels Hamiltonkreis. 6. Generierung der SL-Strands entlang der Polylinien 43

Abbildung 39 Entwurfsstudien von Studierenden der TU Darmstadt unter Verwendung der im Projekt entwickelten Algorithmen. 43

Abbildung 40 SL-Blocks aus unterschiedlichen Materialien, hergestellt mit unterschiedlichen Fabrikationstechniken und variierter Geometrie. Reihe 1.: Beton, Gips, Vollholz, Schichtholz. Reihe 2: Schichtholz, elementiert und CNC gefräst, Lindenholz gelasert, 3d Kunststoffdruck. 3. Reihe: Acrylglas verschiedene Farben. Reihe 4/5: 3D Druck unterschiedliche Formen. 45

Abbildung 41 Standarddesign des taktilen Sensors Digit (<https://digit.ml/>), der als Grundlage für unsere Arbeit diente. Die Bilder auf der rechten Seite zeigen die Rohdaten des Sensors, wenn kein, bzw. ein scheibenförmiges Objekt berührt wird. 47

Abbildung 42 Links sind die Gele mit lasergravierten Punkt- und Streifenmustern auf der Oberfläche des Sensors zu sehen, die eindeutige lokale Merkmale aufweisen, welche zur Quantifizierung der Verformung des Gels genutzt werden können. In der Mitte sind die schwarz beschichteten Gele zu sehen, die Lichtreflexionen reduzieren. Rechts wird das Gesamtdesign des Sensors mit einem quadratischen Gehäuse und dickeren modifizierten Gelpads vorgestellt. 47

Abbildung 43 Ein Klassifikator für Materialoberflächeneigenschaften wird anhand von Sensor-Rohbildern trainiert. Die obere Reihe zeigt Beispielobjekte mit unterschiedlichen Oberflächentexturen, die untere Reihe zeigt die entsprechenden Tastbilder. Ein Klassifikator, der auf vortrainierten ResNet18-Merkmalen basiert, erreicht eine Genauigkeit von 80 %, wenn er auf nur 6 taktilen Bildern pro Klasse trainiert wird. 48

Abbildung 44 Die Normalkraft (z-Richtung) wird auf der Grundlage einer rekonstruierten Tiefenkarte geschätzt, während die Scherkraft (x- und y-Richtung) auf der Grundlage des optischen Flusses geschätzt wird. Auf der rechten Seite ist ein Beispiel für ein taktilen Bild zusammen mit einer 3D-Rekonstruktion der erfassten Objektform und der entsprechenden Tiefenkarte dargestellt. Die Tiefenkarte wird mit Hilfe des entwickelten Algorithmus geschätzt. 49

Abbildung 45 Die Scherkraft ist proportional zur Verformung des Gels in dem Kraftbereich, für den der Sensor geeignet ist. Daher werden die Abweichungen der Punkte im Gel verfolgt, und die Werte der Punktabweichungen werden als proportional zur Scherkraftverteilung angenommen. 49

Abbildung 46 Der entwickelte lernbasierte Ansatz zur Schätzung der Normalkraft. Ein Datensatz mit taktilen Bildern und den entsprechenden Kräften wird durch den Kontakt des Roboters mit einem vordefinierten Satz von 3D-gedruckten Objekten mit Formen und flachen Oberflächen gesammelt (linke Seite der Abbildung). Die Methode wird an nicht-konvexen Objekten evaluiert (zwei Bilder auf der rechten Seite). Die geschätzte

Normalkraftverteilung stimmt visuell mit der Form des Testobjekts überein und kann eine Genauigkeit von bis zu 1 g bei der Gesamtnormalkraft erreichen. **49**

Abbildung 47 Eine Montageaufgabe, die von einem Team aus zwei Robotern durchgeführt wird: Während ein Greifer die Bauelemente an Ort und Stelle hält, fängt der zweite Greifer, der mit taktilen Sensoren ausgestattet ist, einen SL-Block ein. **50**

Abbildung 48 Die Gesamtarchitektur des entwickelten autonomen modularen Montageansatzes. Im Design Modell (links) ist die SL-Block Geometrie, die Fügesequenzen und die Bauteilhierarchien hinterlegt. Der Lernalgorithmus (mitte) umfasst Bauteil- und Fügesimulationen, Belohnungsfunktionen für verstärkendes Lernen, Imitationslernen und numerische Optimierung. Die Robotersteuerung (rechts) beinhalten die physische Ausführung der robotischen Fügung, Sensoren und Feedback. **51**

Abbildung 49 Software-Design der Assembly Gym Umgebung. Komponenten kommunizieren asynchron mit den Gerätetreibern. **53**

Abbildung 50 Kernkomponenten des Assembly Gym Frameworks. Weiße Kästen sind Objekte, die im Benutzercode instanziiert werden. **54**

Abbildung 51 Die Architektur des Modells, das aus den vom Roboter gesammelten realen Daten gelernt wurde, ist so gestaltet, dass das Modell als Eingabe ein Fenster vergangener Beobachtungen erhält, welches in der Abbildung durch den in die Tiefe des Diagramms gehenden Schatten dargestellt wird. Jede Beobachtung ist multimodal und setzt sich aus drei Teilen zusammen: (i) zwei taktile Bilder (linker und rechter Greiferfinger), (ii) Greiferpose, (iii) Posen der manipulierten Blöcke. Rohsensorbeobachtungen durchlaufen einen lernbaren Encoder, bevor sie in das Übergangsmodell gelangen. Gegeben durch ein Fenster solcher Beobachtungen zusammen mit einem Fenster von Aktionen, gibt das Übergangsmodell ein Fenster zukünftiger Beobachtungen zurück, das in diesem Fall als Horizont bezeichnet wird. **56**

Abbildung 52 Links: Simulation des DIGIT Sensors in PyBullet. Die synthetisch generierten Bilder sind nicht realistisch genug, um eine Transition gelernter Modelle auf das echte System zu ermöglichen. Rechts: das reale System. DIGIT Sensoren sind an den Fingerspitzen des Roboters angebracht. **56**

Abbildung 53 Fotografie des experimentellen Montagesetups. Taktile Messungen des Kontakts des DIGIT Sensors mit hölzernen SL-Blöcken. Die resultierenden Bilder sind auf dem Bildschirm dargestellt. **57**

Abbildung 54 Die Position der SL-Blöcke wird mit Hilfe von AprilTags verfolgt. Experimente mit unterschiedlichen Mengen an AprilTags wurden durchgeführt. Das Mitteln über mehrere Tags verbessert die Schätzgenauigkeit, ist jedoch in der Praxis einschränkend, da jeder Block mit vielen Tags versehen werden muss. Daher wurden zwei bis vier Tags pro Block als ausreichend erachtet, um eine stabile Verfolgungsleistung zu erreichen. **58**

Abbildung 55 Eine manuelle Demonstration, wie zwei SL-Blöcke verbunden werden müssen. Die Positionen der Blöcke werden mit AprilTags und Optitrack verfolgt. **59**

Abbildung 56 Während des Trainings der autonomen Montage aufgezeichnete Signale: taktile Bilder, Zeitreihendaten des Roboterzustands und des Blockzustands. **60**

Abbildung 57 Lernkurven während des Trainings des Übergangsmodells. Trainingsverlust (a) und verschiede-

ne Validierungsverluste (b-f) sind in der Abbildung dargestellt. Die Konvergenz des Trainingsverlustes (a) zeigt, dass das Model die Daten aus dem Trainingsdatensatz rekonstruieren kann. Die Validierungsverluste für Position (c) und Orientierung (d) zeigen ebenfalls einen konvergierenden Trend, sowie in der Vorhersage für „Tool Center Point (TCP, blaue Kurve)“ als auch für SL-Block-1 (SL1, orange Kurve). Der Anstieg in Abb. (d) gegen 25k Trainingsschritte deutet auf kurzzeitig schlechten Fit auf dem Validierungsdatensatz. Gleich danach springt der Validierungsverlust auf die abneigende Kurve und nimmt weiter ab. Die letzten zwei Teilabbildungen und zeigen die Validierungsverluste für Bildrekonstruktion im latenten Raum (e) und im Bildraum (f). Es ist bemerkenswert, dass nur der Bildraumverlust (f) nicht konvergiert, und zusätzlich unterscheiden sich die Verluste für den rechten (orange) und den linken (blau) Sensoren. Jedoch im latenten Raum (e) nimmt der Verlust schnell ab. Das deutet auf die Komplexität der genauen Rekonstruktion der kompletten Bilder von dem Sensor hin. Doch für die Steuerung ist nur der latenter Raum von Bedeutung, deswegen ist die insgesamte Konvergenz des Validierungsverlustes (b) ausreichend. **61**

Abbildung 58 Ausgestattet mit zwei DIGIT-Sensoren, realisiert der Roboter die Einfügung eines SL-Blocks in eine existierende Struktur. Eine besondere Herausforderung dabei stellt die erforderliche hohe Präzision dar, bedingt durch die geringen Spaltmaße der SL-Blöcke. **62**

Abbildung 59 Konzeptbild für einen zukünftigen robotergestützten Bauprozesses, bei dem Roboter autonom Strukturen aus SL-Blöcken montieren/demontieren und wieder zusammensetzen. **63**

Abbildung 60 Ein Universal Robot UR 10 ausgestattet mit einem elektronischen Parallelgreifer von Schunk und 3D gedruckten Fingern, die geometrisch präzise für die speziellen SL-Blöcke mit linearen Nuten entwickelt wurden. Weiße Würfel dienen als Stützkonstruktionen oder verhindern das Verrutschen von SL-Blöcken während der Montage. **63**

Abbildung 61 Drei SL-Strands verschlungen in einem borromäischen Knoten. Die weißen, 3D gedruckten SL-Blöcke verfügen über abgerundete Ecken für eine leichtere Montage. Außerdem bieten Nuten Angriffspunkte für einen gabelförmigen Greifer. **64**

Abbildung 62 SL-Block und Gabelgreifer als integral entwickeltes System. Kerben in den SL-Blöcken komplementieren die Gabelquerschnitte der Greifer. Die konisch zulaufenden Greifer zentrieren und positionieren den Block beim zugreifen. **64**

Abbildung 63 SL-Blöcke aus CNC gefrästen, steckbaren Furniersperrholz-Elementen (oben), 3D gedrucktem, weißen Kunststoff (unten links) und aus Beton, gegossen in 3D gedruckte Schalungen (unten rechts). **65**

Abbildung 64 SL-Block und seine Repräsentation als Graph zur Analyse der Einbausituation und möglicher Freiheitsgrade für die Montage. SL-Block 4 wird von seinen Nachbarn 1, 2, 3, und 5 in all seinen Bewegungsrichtungen eingeschränkt. Er kann nicht demontiert werden. **66**

Abbildung 65 Grafische Darstellung der Demontagesequenz von SL-Blöcken. Links ist jeder der fünf SL-Blöcke mit einer eigenen Farbe und einer Zahl dargestellt. Rechts veranschaulicht ein Graph für den jeweils zu entfernenden SL-Block die blockierenden Nachbarn als Zahl. Die Richtung, in der der Block entfernt werden kann, ist mit einem lila Punkt markiert. Ist eine SL-Block entfernt, startet die Analyse für die aktualisierte Aggregation von neuem. Der rekursive Prozess wird fortgesetzt, bis alle Blöcke entfernt sind. **67**

Abbildung 66 Automatisierte robotische Aggregation von weißen SL-Blöcken, die durch rote Würfel gestützt werden. Das Bild zeigt den Roboterarm beim Herausgreifen eines SL-Blocks aus dem Materialzuführungssystem. **68**

Abbildung 67 3D Modell, Explosionszeichnung und Fotografie des Materialzuführungssystems. In einem geneigten Schacht werden die Blöcke hintereinander aufgereiht. Entnimmt der Roboter unten einen Block, rutscht der nächste automatisch nach. Das System wird am oberen Ende befüllt. **68**

Abbildung 68 Bildsequenz der robotischen Fügung von Stützwürfeln (rot) und SL-Blöcken (weiß). Eine lineare Struktur wird hergestellt. **70**

Abbildung 69 Montagesituation mit Roboter, Bauteilen und Zielhülle für eine Konstruktion in der 3D Simulation (links). Graphenbasierte Darstellung der Ausgangssituation und Berechnung der auszuführenden Aktionen. **72**

Abbildung 70 Oben: Simulationsumgebung mit einem 7-achsigen Roboter, auf einer Fläche verteilten Polykuben und einer Zielfigur als Hüllgeometrie. Ziel ist es, dass der Roboter eigenständig die Bausteine in die grün markierte Zielfigur setzt, so dass eine dreidimensionale Figur entsteht, die der Silhouette möglichst nahe kommt. Unten: Drei Fotos des physischen Experiments zur robotischen Stapelung von auf einer Platte zufällig verteilten Kuben zu einer Figur, die einer vorgegebenen räumlichen Polylinie folgt. **73**

Abbildung 71 SL-Block Aggregationen aus 3D gedruckten weißen Blöcken bilden Wände, Stützen, Träger und Hohlquerschnitte. Rote Würfel stützen die Konstruktionen. Blaue Blöcke markieren jeweils die Startseite eines SL-Strands, der auf den darunterliegenden SL-Strands gestapelt wird. **76**

Abbildung 72 Hohlprofil aus SL-Block Aggregationen als digitales Modell und physischer Demonstrator. Das Bild in der Mitte zeigt, dass das Objekt abschnittsweise stapelbar ist, bzw. eine ganze Gruppe von SL-Blöcken nach oben abhebbar ist. Die blauen Steine markieren jeweils einen SL-Strand Bauabschnitt. Die Fügestellen benötigen Eigengewicht oder hohe Reibung, da die Verzahnung hier nicht alle Freiheitsgrade sperrt. **77**

Abbildung 73 Bildsequenz zur Entwicklung des Demonstrators. Ein Quader wird durch zwei lotrecht platzierte Wände in vier gleiche Teile unterteilt. Die kreuzförmigen Wände werden in Voxel zerlegt. Aus beiden Voxelwänden werden V-förmige Elemente herausgezogen und jeweils an ihren oberen Enden horizontal verbunden. **77**

Abbildung 74 Bildsequenz zur Entwicklung des Demonstrators. Fortsetzung des Prozesses aus Abbildung 73. Die oberen horizontalen Verbindungen werden ausdetailliert. Eine Verbindung wird zu einem Fachwerkträger aus Voxel. Die andere Verbindung verzahnt sich mit dem Fachwerkträger mit einem U-förmigen Detail im mittleren Kreuzungspunkt. Im unteren Teil der Abbildung werden die einzelnen Elemente des Demonstrators dargestellt: Diagonalen aus den V-Stützen, Fachwerkträger, Sockel. **78**

Abbildung 75 Die Bildsequenz zeigt die manuelle Montage der einzelnen robotisch gefügten Bauteile (Fachwerkträger, Diagonalen, Stütze, Zuggurt). **79**

Abbildung 76 Links werden einzelne Bauteile (Fachwerkträger, Diagonalen, Stütze, Zuggurt, Sockel) als digitale Modelle und als Fotografie während der Montage gezeigt. Rechts ist der fertig montierte Demonstrator zu sehen. **79**