



Spatial Parameterization of Non-Semantic CAD Elements for Supporting Automated Disassembly Planning

Chris Rausch^{1*}, Ben Sanchez¹, and Carl Haas²

¹ *PhD candidate, Department of Civil and Environmental Engineering, University of Waterloo*

² *Professor, Department of Civil and Environmental Engineering, University of Waterloo*

**Corresponding author's e-mail: chris.rausch@uwaterloo.ca*

ABSTRACT

Digital data and associated semantics play a fundamental role in supporting the vision of Construction 4.0. Advancements in digitization workflows such as scan-to-BIM and automated meta-data generation are being used for data-driven decision making. A challenge with collecting and processing raw, non-semantic data is the process of integrating intelligence into and characterizing data automatically. This paper demonstrates how spatial parameterization (i.e., extracting, modifying and analysing parameters that define the spatial properties of a component) can be used as a method for automating steps in disassembly planning for buildings. The potential use cases of disassembly planning include adaptive building reuse, robotic assembly programming, reconfigurable prefabricated assemblies and selective disassembly for rehabilitation and repairs. This paper presents spatial parameterization in a framework to disassemble building components via a rule-based algorithm that comprises three dimensional Cartesian properties and clash detection between non-semantic CAD elements. Demonstration of the framework is carried out using a case study where the interior wall of a building on the University of Waterloo campus was disassembled for adaptive reuse purposes. Comparison of the case study results to the actual disassembly sequence demonstrates how spatial parameterization is effective for automating key steps in disassembly planning. A discussion is provided to identify key barriers to increased automation which relate to modelling accuracy, Level of Development (LOD) for Building Information Modelling (BIM), and global spatial constraints for disassembly.

KEYWORDS

Disassembly planning; digitalization workflows; computational algorithms; automation; Building Information Modelling; parametric analysis

INTRODUCTION

The influence of the fourth industrial revolution in the construction industry can already be seen in applications such as machine learning, artificial intelligence, additive manufacturing, big data, blockchain and high-performance computing, to name a few. Digitization – i.e., the convergence of digital and physical states – is considered to be the most powerful driver of innovation in the coming decades for the progression of the fourth industrial revolution (Kagermann 2015). However, the construction industry is among the least digitized, according to the 2015 McKinsey Global Institute industry digitization index (Agarwal et al. 2016). Developing associative data semantics and workflows for transforming data between physical and digital states is a core challenge that needs to be addressed in construction. Examples of existing workflows where digital data is extracted from physical or other non-digital entities include scan-to-BIM, scan-vs-BIM,

progress tracking, materials tracking, Internet of Things (IoT) applications, mobile computing and digital imaging analysis (Turkan et al. 2012; Bosché et al. 2015; Xu et al. 2018; Han and Golparvar-Fard 2017; Kochovski and Stankovski 2018; Kim et al. 2013). Each of these workflows produce large datasets that require analysis for integration into value-adding processes and insights. While automation can be developed to help embed intelligence into the data generated in digitization workflows, in many cases, manual processing and cognition are still required to produce working solutions. For instance, scan-to-BIM algorithms can easily detect planar and cylindrical features but struggle with more complex features, which still rely on manual modelling. As digitization continues to evolve and shape the construction industry, we need to continue to search for and develop novel data processing approaches that can be used to solve diverse challenges.

In life-cycle management of existing buildings, there is a growing interest to improve performance holistically from cradle to grave (i.e., from manufacturing to end-of-life). This is required to improve sustainability and to create the conditions necessary for a circular economy (Geissdoerfer et al. 2017), which is a model that moves away from a linear product usage (i.e., disposal at end-of-life) towards increasing levels of reuse and product recovery (Korhonen et al. 2018). One way to address end-of-life product recovery in construction is adaptive building reuse which involves restoring and or changing the use of buildings entering their disuse stage. Adaptive reuse can maximize the residual utility and value of our aging building stock, bringing significant environmental and sustainability benefits. Disassembling components according to a specific sequence of steps (i.e., “disassembly sequence planning”) is a key part of the adaptive reuse process (Sanchez and Haas 2018). Disassembly sequence planning is common in the manufacturing industry, where it is used to extract and repair defective components in an assembly or for recovering salvageable components at a product’s end-of-life. Due to a lack of updated information, disassembly sequence planning in manufacturing is performed using incomplete data between the overall assembly and sub-assemblies (Vongbunyong and Chen 2015). Similarly in construction, updated data for components is not available since as-built models are often not current nor contain the required semantic information. As a result, disassembly planning relies on manual processes for defining relationships between components. Previously, the authors developed a novel framework for adaptive reuse of buildings based on the disassembly graph model (Sanchez and Haas 2018; Sanchez et al. 2019). This framework is the first of its kind to apply disassembly planning in construction in order to target key components for selective disassembly. Although this research creates tremendous opportunities for the construction industry, a major challenge in this work was that key spatial relationships of components had to be extracted manually, which is very tedious, time consuming and prone to errors.

While extracting geometric information from 3D objects in BIM is a relatively trivial endeavour, the development of algorithms for higher order semantics (such as complex spatial relationships) is a challenge and remains elusive in many spheres of the construction industry (Lu et al. 2017). Traditional automated workflows in construction require groups of experienced programmers and software companies to develop custom-made solutions. This comes with high cost, long turnaround and requires a highly coherent software team who understands construction processes deeply. However, the advent of visual programming languages (VPL) for computational algorithm development (Erwig et al. 2017) has fostered a new way of developing custom software solutions

for the construction industry. While many VPLs were designed to only handle intuitive design tasks (Preidel and Borrmann 2016), recent developments and contributions from the open-source community have increased functionality for being able to address highly complex tasks. This paper demonstrates how intelligence can be extracted from non-semantic CAD elements (i.e., elements lacking parametric properties of spatial relationships with other elements) through a rule-based VPL algorithm that comprises three dimensional Cartesian properties and clash detection. Collectively, this process provides a way to analyze spatial parameters of components automatically which is demonstrated in this paper using a real adaptive reuse project. In doing so, several key challenges are outlined which need to be addressed in order to utilize other existing digitization workflows (e.g., scan-to-BIM) within the emerging Construction 4.0 ecosystem.

METHODS

This paper develops key spatial relationships between objects in a 3D model automatically for the purpose of supporting automated disassembly planning, and specifically in the novel framework previous introduced by the authors (Figure 1). This framework has five key matrices that define the underlying mechanics in terms of the spatial relationships between components in an assembly (information regarding how to obtain the 3D BIM model and how to target the disassembly components can be found in Sanchez and Haas 2018). The first matrix (component contact constraint matrix) defines the direct connectivity between components c_n . The second matrix (component motion constraint matrix) defines the first object(s) that will obstruct an object of interest from being disassembled in a given direction. The third and fourth matrices define the contact and motion constraints for fasteners between objects. The fifth matrix (component projection constraint matrix) is an extension of the motion constraint matrix, except it includes all obstructions for an object of interest in a given disassembly direction. Most approaches simplify the procedure by considering only principle Cartesian directions i.e., x, y, z (Smith et al. 2012). This paper does not consider fastener constraints since most 3D models in construction are not developed to the level of fastener details such as nails, screws, bolts, etc.

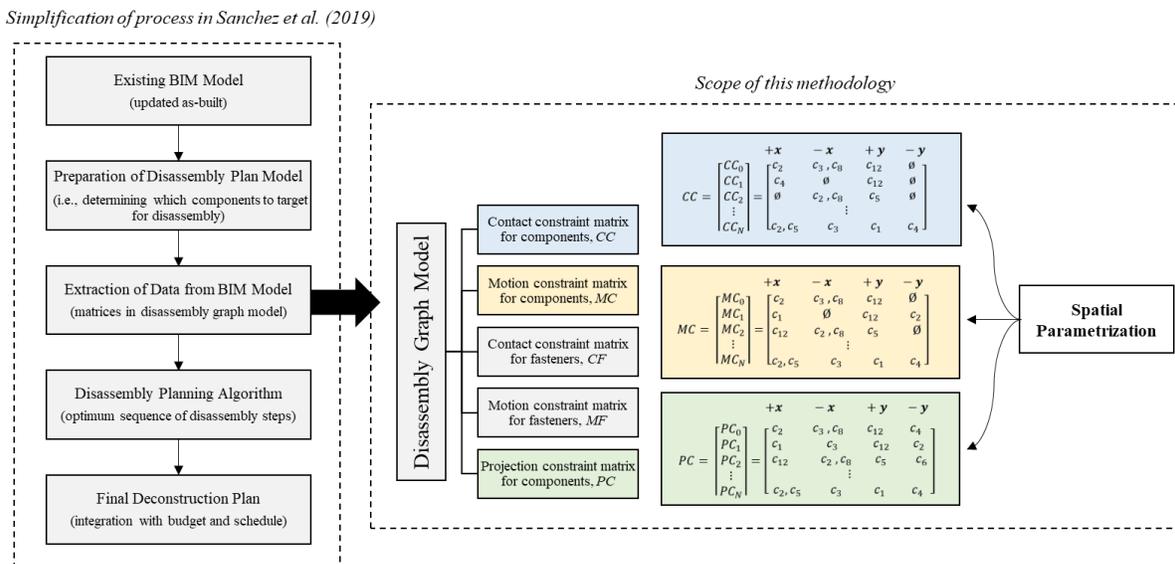


Figure 1. Summary of previous work by authors for disassembly planning and scope of this paper as it relates to automating the extraction of key matrices.

Spatial Parameterization Framework

Model initialization. The first step involves grouping CAD elements together according to their object definition. This enables all raw CAD elements (e.g., curves, surfaces, meshes, etc) to be assigned a unique object ID. If the existing BIM model was created using a parametric software such as Autodesk® Revit, raw CAD data is often already sorted into unique groups. Apart from manually grouping elements, emerging workflows are being developed to automatically recognize and semantically segment objects (Czerniawski and Leite 2018). The second part of model initialization is determining which objects are in direct contact with each other. Throughout this framework, clash detection is used as an event to trigger binary responses for contact between object groups (“1” for positive clash event, “0” for no clash). All binary clash events are initially stored in an n -dimensional array, where n is the total number of objects.

Removing self-intersections from binary contact array. Since the process of clash detection between groups of objects will inevitably create self-intersections (each group can contain numerous CAD elements that are in contact with each other that will return positive clash events), these need to be removed from the binary contact array.

Constraint matrices for components. For extracting the contact constraint matrix, each object is moved iteratively in each of the \pm principle Cartesian directions (+x, -x, +y, -y, +z, -z) until it is brought into a negative clash condition (no contact, binary response of “0”). The distance in each direction to bring the objects out of contact is recorded, and the direction with the smallest distance is the disassembly direction. The motion constraint matrix is derived by iteratively moving each object in the principle Cartesian directions until the first clash event is detected. The contact object(s) in each direction are recorded in the motion constraint matrix. Finally, the projection constraint matrix is derived from the motion constraint matrix, with the only difference being that all objects in a given direction, and inside of the working-space, are recorded. For all matrices, if a given search (e.g., motion constraints for component c_l in the +x direction) turns up with a null response (i.e., no objects found), then by default a value of “-1” is inserted into the matrix.

The spatial parameterization framework was programmed using Rhinoceros® software and a visual programming plugin called Grasshopper™. As a NURBS (non-uniform rational B-spline) based software, Rhinoceros® does not intrinsically have meta-data and thus all CAD elements rely on an algorithmic approach to parametrically embed intelligence. A high-level depiction of the algorithm (with over 800 script components) is shown in Figure 2.

Algorithm parameters. The distance for each iterative move is a parameter in the algorithm, but by default it is set at 10mm. Decreasing this threshold will not yield a more accurate result for object matrices but will significantly increase the computing demand (note that a smaller step value may be required for deriving fastener constraint matrices). Another algorithm parameter is tied to the motion constraint and project constraint matrices in the form of a maximum distance. In previous work (Sanchez et al. 2019), the authors have introduced a “working-space” of 1500mm which functions as a spatial envelope within which a construction worker can remove a given component from the assembly. This parameter is applied to both the motion and projection constraint matrices in order to define an upper bound on the iterative moves for each object.

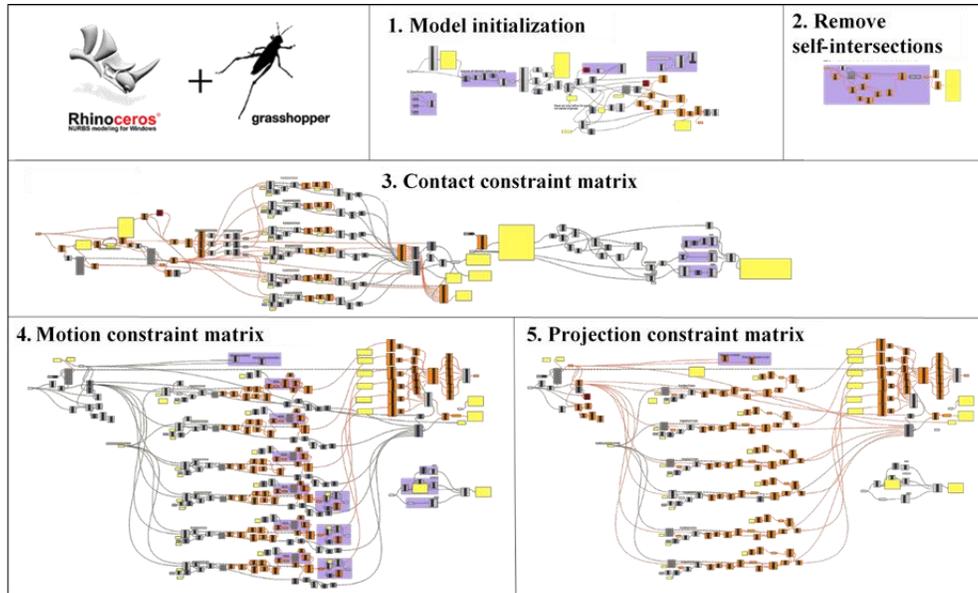


Figure 2. Depiction of visual programming script for spatial parameterization framework. Script created using Grasshopper™ algorithmic software in Rhinoceros® modelling platform.

RESULTS

The spatial parameterization framework was used on data from an actual construction project, where an entire corridor of a building located at the University of Waterloo was redeveloped and repurposed. An important part of this project was disassembling the construction assembly so that objects such as sinks, cabinets, and some structural members could be either reused or properly recycled. For the purpose of determining a suitable disassembly plan, a 3D model of an interior wall was generated using Autodesk® Revit, and then subsequently imported as raw CAD elements (curves, surfaces and meshes) into Rhinoceros®, as shown in Figure 3.

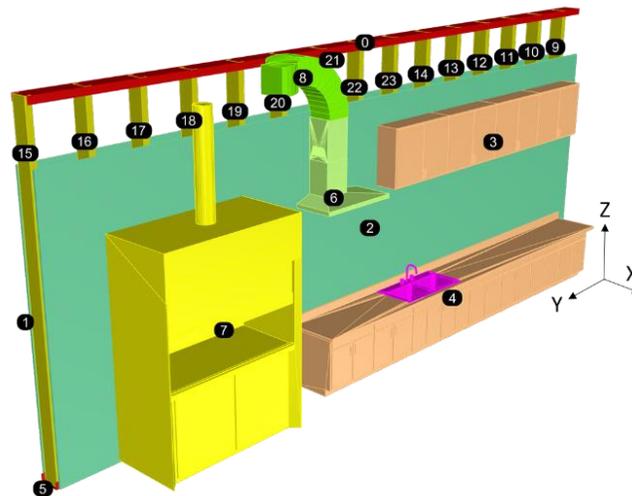


Figure 3. Construction assembly for adaptive reuse case study. 3D CAD elements categorized into groups, numerically annotated and coloured for visualization purposes. Note: the numbers in this figure only represent object labelling, and do not reflect the disassembly sequence.

Once the elements were imported, they were grouped according to their object definition (e.g., all elements corresponding to the upper cabinet – object 3 – were grouped together). Then, using the

algorithm depicted in Figure 2, a contact constraint matrix (Eqn. 1), motion constraint matrix (Eqn. 2) and project constraint matrix (Eqn. 3) were automatically produced. The output from the algorithm is a text file, delineated by commas to represent principle Cartesian directions, which can be imported into the disassembly planning workflow. For comparison, a manual approach was employed for generating this data. The authors manually isolated each component in the model and recorded the objects in direct contact, and the objects constraining motion and projection.

$$CC = \begin{bmatrix} CC_0 \\ CC_1 \\ CC_2 \\ \vdots \\ CC_{23} \end{bmatrix} = \begin{bmatrix} -1, -1, -1, -1, -1, (9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23) \\ 5, -1, -1, -1, -1, -1 \\ (3, 4, 6, 7, 8), 5, -1, -1, -1, -1 \\ \vdots \\ -1, -1, -1, -1, 0, 5 \end{bmatrix} \quad (1)$$

$$MC = \begin{bmatrix} MC_0 \\ MC_1 \\ MC_2 \\ \vdots \\ MC_{23} \end{bmatrix} = \begin{bmatrix} 8, -1, -1, -1, -1, (9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23) \\ 5, -1, -1, -1, 0, -1 \\ (3, 4, 6, 7, 8), 5, -1, -1, 0, -1 \\ \vdots \\ 2, 1, 22, 14, 0, 5 \end{bmatrix} \quad (2)$$

$$PC = \begin{bmatrix} PC_0 \\ PC_1 \\ PC_2 \\ \vdots \\ PC_{23} \end{bmatrix} = \begin{bmatrix} 8, -1, -1, -1, -1, (9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 1, 2) \\ (5, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 2, 3, 7, 8, 4, 6), -1, -1, -1, 0, -1 \\ (3, 4, 6, 7, 8), (5, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 1), -1, -1, 0, -1 \\ \vdots \\ (2, 3, 4), 1, (22, 21), (14, 13), 0, 5 \end{bmatrix} \quad (3)$$

Algorithm performance. Minor discrepancies between the automated and manual approaches were detected. After analysis, these differences were found to be missing constraint components in the matrices produced in manual approach, demonstrating that the automated approach is suitable of catching human errors. The automated approach was found to be 100% accurate for extracting the contact, motion and project constraints. This accuracy does however depend on having a properly configured BIM model, e.g., no overlapping objects, no gaps between objects that should be in contact, etc. Of course, the accuracy suffers with bad input and an incorrectly configured BIM model. Furthermore, the process of iterating principle movements, checking for collisions and cross-referencing components was found to be computationally heavy. This is likely based on the distance threshold used for iterating movements (set at 10mm), the working space parameter (set at 1500mm) and the level of model complexity (this 3D model contained 182 curves and 3364 mesh elements). The automated approach took 243min to extract the contact constraint matrix, 734min for the motion constraint matrix and 1076min for the projection constraint matrix. With a total run time of 2053 mins, there were no direct time savings for using this algorithm over a manual approach (which took 145 mins in total). However, the benefit of an automated approach is that this process can be run in the background of other processes at very little cost. In addition, the accuracy of the extracted data can be improved. Moreover, with higher powered processors, the algorithm run time could be reduced to become faster than the manual approach.

CONCLUSION

Considering the unique demands of the fourth industrial revolution, the construction industry needs to continue developing innovative approaches to digitization. The benefits of automated digitization workflows far outweigh using manual approaches, notwithstanding the practical limitations that need to be overcome. As evidenced in this paper, even if automated digitization workflows (while in their infancy) may not bring direct time savings, reduced labour costs, parallel processing and improved data accuracy can still be realized. With improvements to computing

capabilities expected in the future, even computationally heavy algorithms can be faster than manual approaches. Previously, the authors developed a novel application of disassembly planning for buildings that supports adaptive reuse. A challenge in this work was the manual steps involved for initialization of required matrices. This paper demonstrates how spatial parameterization and clash detection can be used to automatically define geometric relationships between components in an assembly with non-semantic CAD data. This method was found to catch human errors and enable parallel processing, despite the long runtime. Lessons learned in previous studies indicate that due to the large quantity of semantic data to retrieve from the BIM model, manual input mistakes are highly probable. In fact, in each of the previous case studies, the authors detected initial mistakes with the manual process of creating the initial matrices, pointing to the reliability of the proposed method in this paper for catching these inevitable costly mistakes.

Challenges processing construction assembly data. In the case study, many model components were not initially in direct contact that should have been (e.g., the upper horizontal frame member – component c_0 – was not in direct contact with vertical frame members – components c_9 to c_{23}). As a result, manual processing was required to improve this modelling accuracy. While this issue can be addressed upfront within parametric modellers such as Autodesk® Revit, it is much more likely to exist in non-semantic datasets where there is no active parametric control for ensuring contact between components (such as for mesh models or in NURBS-based models). Another major challenge with analysis of as-built data relates to uncertainty in assembly composition. For instance, if no updated model exists for a construction assembly, it is difficult to determine hidden components (e.g., not knowing what lies behind a wall before disassembly; this results from not having up-to-date BIM information). In general, data uncertainty is also very challenging for the success of disassembly planning since it can be difficult to gage component quality, and to detect defective connections that reduce ease of disassembly.

Limitations. As described in the methods section, this paper does not automatically extract constraint matrices for fasteners. This is due to the lack of granularity for fastener details in most BIM models. Another limitation to the proposed algorithm is that highly interconnected (and multi-system) construction assemblies such as pipe assemblies embedded in walls cannot be analyzed since the removal of certain components requires some aspect of deconstruction or “cutting”. As such, the proposed algorithm can only be used for linear extraction processes.

Future work. The authors are exploring methods to decrease the run time through the following aspects: reducing the model complexity (i.e., filtering only CAD elements that pertain to the geometric envelop of an object), exploring what level of detail is required for objects, and optimizing parameter presets in the algorithm (iterative distance value and working-space). Within the algorithm, other improvements can be made so that once a direction has been defined for a given component constraint, redundant processing in elapsed for-loops can be terminated. To automate the process of extracting fastener constraint matrices, the authors are exploring the use of heuristics to automatically embed this detail into the model, as well as to solve geometric discrepancies between components (e.g., components that should be in contact which are not). Finally, the authors aim to explore use cases for the proposed algorithm in the following contexts: scan-to-BIM (through a mesh model medium), and performance comparisons between various construction sectors (e.g., residential, commercial, industrial).

ACKNOWLEDGEMENTS

The authors would like to acknowledge the Natural Sciences and Engineering Research Council of Canada (NSERC) for the financial support of this research.

REFERENCES

- Agarwal, R., Chandrasekaran, S., and Sridhar, M. (2016). "Imagining construction's digital future." McKinsey Insights Report, McKinsey & Company and McKinsey Global Institute, Jun, 2016.
- Bosché, F., Ahmed, M., Turkan, Y., Haas, C. T., and Haas, R. (2015). "The value of integrating Scan-to-BIM and Scan-vs-BIM techniques for construction monitoring using laser scanning and BIM: The case of cylindrical MEP components." *Autom.Constr.*, 49 201-213.
- Czerniawski, T., and Leite, F. (2018). "3DFacilities: annotated 3D reconstructions of building facilities." *Workshop of the European Group for Intelligent Computing in Engineering*, Springer, 186-200.
- Erwig, M., Smeltzer, K., and Wang, X. (2017). "What is a visual language?" *Journal of Visual Languages & Computing*, 38 9-17.
- Geissdoerfer, M., Savaget, P., Bocken, N. M., and Hultink, E. J. (2017). "The Circular Economy—A new sustainability paradigm?" *J.Clean.Prod.*, 143 757-768.
- Han, K. K., and Golparvar-Fard, M. (2017). "Potential of big visual data and building information modeling for construction performance analytics: An exploratory study." *Autom.Constr.*, 73 184-198.
- Kagermann, H. (2015). "Change through digitization—Value creation in the age of Industry 4.0." *Management of permanent change*, Springer, 23-45.
- Kim, C., Park, T., Lim, H., and Kim, H. (2013). "On-site construction management using mobile computing technology." *Autom.Constr.*, 35 415-423.
- Kochovski, P., and Stankovski, V. (2018). "Supporting smart construction with dependable edge computing infrastructures and applications." *Autom.Constr.*, 85 182-192.
- Korhonen, J., Honkasalo, A., and Seppälä, J. (2018). "Circular economy: the concept and its limitations." *Ecol.Econ.*, 143 37-46.
- Lu, W., Webster, C., Chen, K., Zhang, X., and Chen, X. (2017). "Computational Building Information Modelling for construction waste management: Moving from rhetoric to reality." *Renewable and Sustainable Energy Reviews*, 68 587-595.
- Preidel, C., and Borrmann, A. (2016). "Towards code compliance checking on the basis of a visual programming language." *Journal of Information Technology in Construction (ITcon)*, 21(25), 402-421.
- Sanchez, B., and Haas, C. (2018). "A novel selective disassembly sequence planning method for adaptive reuse of buildings." *J.Clean.Prod.*, 183 998-1010.
- Sanchez, B., Rausch, C., and Haas, C. (2019). "Deconstruction programming for adaptive reuse of buildings." *Automation in Construction*, (under review).
- Smith, S., Smith, G., and Chen, W. (2012). "Disassembly sequence structure graphs: An optimal approach for multiple-target selective disassembly sequence planning." *Advanced Engineering Informatics*, 26(2), 306-316.
- Turkan, Y., Bosche, F., Haas, C. T., and Haas, R. (2012). "Automated progress tracking using 4D schedule and 3D sensing technologies." *Autom.Constr.*, 22 414-421.
- Vongbunyong, S., and Chen, W. H. (2015). "General Disassembly Process." *Disassembly Automation*, Springer, 9-24.
- Xu, G., Li, M., Chen, C., and Wei, Y. (2018). "Cloud asset-enabled integrated IoT platform for lean prefabricated construction." *Autom.Constr.*, 93 123-134.