Computational Assemblies: Analysis, Design, and Fabrication

Peng Song[1] Ziqi Wang[2] Marco Livesu[3]

[1]Singapore University of Technology and Design, Singapore [2]GCM, EPFL, Switzerland [3]CNR IMATI, Italy

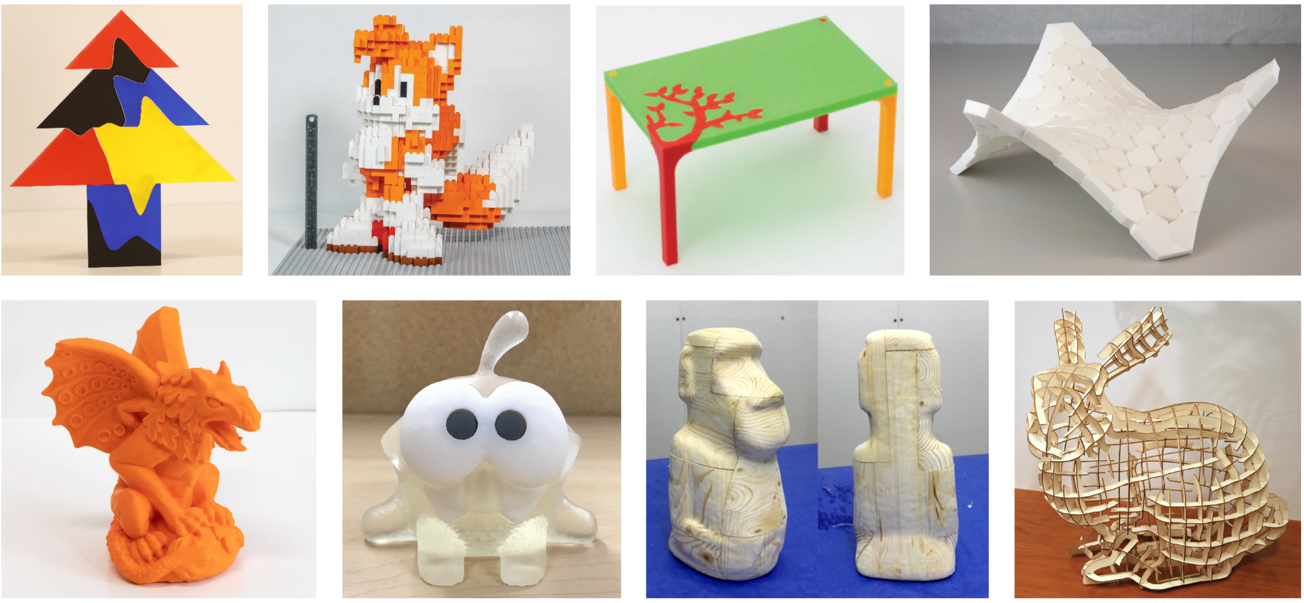

Figure 1: *In the area of computational assemblies, researchers develop computational methods (Top) to design assemblies for various applications including puzzles [WSP21a], LEGO sculptures [LYH* 15], furniture [YKGA17], and architecture [WSIP19], as well as (Bottom) to fabricate complex shapes as assemblies for addressing limitations of various digital fabrication techniques including 3D printing [FAG* 20, ACA* 19], CNC milling [MLS* 18], and laser cutting [CPMS14].*

Abstract

Assemblies are ubiquitous in our daily life, such as toys, electronic devices, furniture, and architecture. They enable to build large and complex objects by composing small yet simpler parts, facilitating fabrication, storage, maintenance, and usage. However, designing assemblies is a highly non-trivial task because one needs to consider not only the properties of each individual components, but also of the whole assembly, such as aesthetics and stability. Motivated by recent advancements in digital fabrication, various computational techniques have been developed to analyze, design, and fabricate assemblies, aiming to enable general users to easily personalize them. This tutorial will give an introduction to these computational techniques, focusing on four fundamental aspects, i.e., parts fabricability, parts joining, assembly planning, *and* structural stability. *In this tutorial, we will take a deep dive into computational methods to analyze these aspects for a given assembly as well as to design and fabricate assemblies that satisfy user-specified requirements in these aspects. This tutorial assumes knowledge of the fundamentals of computer graphics. Attendees should come away from this tutorial with a broad understanding of current work in computational assemblies, as well as familiarity with the necessary knowledge to start their own research in this area.*

CCS Concepts

• ***Computing methodologies*** → *Shape modeling;* • ***Applied computing*** → *Computer-aided manufacturing;*

1. Introduction

An assembly refers to a collection of parts joined together to achieve a specific form and/or functionality. Assemblies enable to build large and complex objects by assembling a set of small and simple parts, facilitating their fabrication, storage, maintenance, and usage. Due to this reason, assemblies are ubiquitous in our daily life, existing in a wide variety of scales and geometric forms; typical examples include toys, electronic devices, furniture, and architecture. While assemblies have intriguing properties, designing assemblies is a non-trivial task, even for professionals. A slight modification on an individual part could have a global impact on the aesthetical, structural, and/or functional performance of the whole assembly. Ensuring that all parts can be put together and properly joined to form the final assembly is another challenge. To address these challenges, researchers have developed various computational techniques for designing and fabricating assemblies [ALL*18, BCMP18, WSP21b].

In this tutorial, we define *computational assemblies* as an emerging research area in computer graphics that studies computational techniques for analyzing, designing, and fabricating assemblies. Computational assemblies provide new ways for general users to easily design and make their personalized assemblies. Users just need to specify their high-level specifications on assemblies and then computational tools that encapsulate the needed low-level knowledge will automatically generate and optimize designs to satisfy these specifications. For example, if a user wants to design a 6-piece equilibrium puzzle that looks like a bunny, then his/her high-level specifications on assemblies would be the number of puzzle pieces, equilibrium under gravity, and assembly appearance. The design generated from the computational tools typically consists of the final detailed shape of each component part for fabrication and the assembly instruction of parts for installation.

In general, computational assemblies address two different goals. The first goal is computational design of assemblies for a specific application, e.g., puzzles for entertainment [SFCO12] and furniture for supporting human activities [YKGA17]; see Figure 1 (top). The second goal is computational fabrication of complex shapes as assemblies for addressing limitations of a specific digital fabrication technique, e.g., approximating a 3D shape with a set of planar parts for laser cutting [CPMS14]; see Figure 1 (bottom). Besides computational design and fabrication, computational analysis of assemblies is necessary and crucial, not only for evaluating the resulting designs, but also for guiding the whole design process. This tutorial will introduce computational techniques for analysis, design, and fabrication of assemblies, respectively; see Table 1 for the tutorial outline.

Table 1: *Tutorial outline.*

Section title	Presenter	Length
Introduction	Peng Song	25 mins
Computational Analysis of Assemblies	Peng Song	45 mins
Break		10 mins
Computational Design of Assemblies	Ziqi Wang	45 mins
Computational Fabrication of Assemblies	Marco Livesu	45 mins

2. Scope and Target Audience

This tutorial requires some basic knowledge in computer graphics (e.g., shape representations, transformations) to understand how assemblies are represented, modelled, and designed in computers.

Scope. Assembly is a very broad concept. According to their functionality, assemblies can be classified as *structures* that transmit forces to carry loads and *mechanisms* that transmit motion and forces to perform mechanical work. This tutorial limits its scope to *structures with rigid parts*, focusing on four fundamental aspects, i.e., *parts fabricability*, *parts joining*, *assembly planning*, and *structural stability*. By assuming that the fabrication material is rigid, design of assemblies generally can be formulated as a geometric modeling and optimization problem, with minimal consideration of material behavior (e.g., friction).

Target Audience. This tutorial is aimed at two classes of audiences. The first class is new graduate students in computer graphics, or more experienced researchers in other fields who are looking for an entry point into this exciting area. The second class is designers and makers who are interested in making use of computational tools to create assembly designs that are impossible before. At the end of the tutorial, we hope that people in the fist class are well equipped with the necessary knowledge to start independent research of their own in this area, and people in the second class know where to look and with whom to consult if they want to make use of research results to assist their design and fabrication process.

3. Tutorial Outline

Introduction (25 mins)
Presenter: Peng Song
Peng will first introduce the presenters. He will then introduce the definition of assemblies, ubiquitous of assemblies in many applications, and advantages of assemblies compared with monolithic objects. The technical content of his talk will start by explaining the importance of computation in analysis, design, and fabrication of assemblies, outlining the key challenges, and motivating why computer graphics would play a central role in addressing these challenges. He will then define *computational assemblies* as an emerging research area that studies computational techniques for analyzing, designing, and fabricating assemblies. After that, he will introduce the learning goal of the tutorial: attendees should understand the main motivations for computational assemblies, feel equipped to start their own research work in this area, or to make use of existing research results to assist their design and fabrication process. Finally, he will define scope for the tutorial, and introduce the tutorial outline as well topics to be covered by each presenter.

Computational Analysis of Assemblies (45 mins)
Presenter: Peng Song
In this section, Peng will present and explain computational methods to analyze different aspects of assemblies. Peng will first give an overview of various aspects of assemblies that are relevant to design and fabrication (e.g., aesthetics, packing efficiency), and then take a deep dive into four fundamental aspects of assemblies (i.e., parts fabricability, parts joining, assembly planning, and structural stability). One of the key insights that Peng will share is that there are many different ways to abstract assemblies in computers, and a

suitable abstraction/representation of assemblies can significantly simplify and/or speed-up the computational analysis algorithm.

Peng will start with computational analysis of parts fabricability. He will introduce the most widely used digital fabrication techniques such as 3D printing and CNC milling, emphasize the geometric constraints associated with each fabrication technique, and explain computational methods that analyze to what degree the geometric constraints are satisfied by a given part shape. Secondly, Peng will classify existing parts joining approaches and focus on those based on integral joints. He will explain that the goal of an integral joint is to restrict the relative movement of two associated parts, and the ability of joining parts can be quantified by the allowed motion space of one part relative to the other. Thirdly, Peng will classify assembly plans according to their inherent features, illustrate the assembly plans with intuitive visual examples, and introduce computational algorithms to find these assembly plans [WSP21b, GM15]. He will introduce *assembly-by-disassembly* as an important strategy for assembly planning since parts in an assembled state have far more precedence and motion constraints than in a disassembled state. Fourthly, Peng will introduce the equilibrium method to analyze whether a given assembly is structurally stable under known external forces. Due to the static-kinematic duality, structural stability can be analyzed either in the force space [WOD09] or the motion space [WSP21a]. However, the motion-based approach has a significant advantage of identifying if a given assembly is globally interlocking (i.e., equilibrium under arbitrary external forces) [WSIP19], which cannot be achieved by the force-based approach. Finally, Peng will outline directions for the future and examine the open problems around computational analysis of assemblies such as functionality analysis.

BREAK (10 mins)

Computational Design of Assemblies (45 mins)
Presenter: Ziqi Wang
In this section, Ziqi will introduce the computational methods to design assemblies with various forms and functionalities. He will first summarize the challenges of manual design to emphasize the importance of computational approaches. He will give a brief overview of typical assembly design problems and their primary objectives (e.g., assemblability, structural stability, aesthetics). His presentation then will mainly focus on designing assemblies that are structurally stable and assemblable. Typical examples include assemblies in equilibrium [WOD09], assemblies with lateral stability [WSIP19], and globally interlocking assemblies [SFCO12, WSP18]. One of the main algorithmic design principles is to use proper data structures to represent assemblies in different scenarios. A well-chosen representation can reduce design variables' redundancy and improve the algorithm's performance.

He will begin with the classic trial-and-error algorithms for creating structurally stable assemblies. This method keeps generating new designs until one of them passes the stability analysis explained in the first part of this course. However, this approach could be highly inefficient, especially when the set of feasible solutions is only a fraction of the entire design space, e.g., when designing globally interlocking assemblies. Rule-based algorithms are introduced to address this challenge [SFCO12, SFLF15]. When adding a new part to the assembly, strict rules must be followed to maintain the structural stability of the assembly. These rules could eliminate potential solutions. To allow the algorithm to explore a larger design space, He will present the gradient-based algorithm [WSW*12, WSP21a], which quantitively measures the structural stability and optimize the parts' shape to improve such measurement. In particular, he will demonstrate one effective strategy, the motion-guided design algorithms [WSP21a]. Instead of optimizing the assemblies' geometry, the motion-guided method optimizes the motion restriction imposed by contacts between parts in a conceptual way. The method then finds corresponding contacts' geometry to satisfy the motion restriction requirements. This method decouples the motion and geometry, improving the algorithmic efficiency, and is flexible in handling various applications.

Computational Fabrication of Assemblies (45 mins)
Presenter: Marco Livesu
In this section Marco will discuss algorithms that aim to decompose a complex shape in order to meet geometric or semantic restrictions imposed by fabrication hardware. Two prominent examples that fall in this category are methods that split multi-color or multi-material objects into an assembly of uniform sub-objects, and algorithms that decompose geometrically intricate objects into parts with simpler geometry, such as height fields (e.g. for subtractive CNC milling or molding) or quasi height fields (e.g. for 3D printing with minimal overhangs).

The section will start with an introduction on the most widely used fabrication techniques, such as 3D printing, CNC milling and molding. Particular emphasis will be given to the geometric and material constraints that are peculiar of each fabrication paradigm, because these constraints define the space of all and only shapes that can be fabricated with a certain hardware.

Methods that operate by splitting an object into simpler parts always strive for *minimal* decompositions, that is, they want to create the minimum number of sub-pieces such that all input constraints are fulfilled. As for many other domain decomposition problems, this often leads to NP-Hardness, which is mitigated by the adoption of heuristic approaches that do not offer theoretical guarantees, and that mostly differentiate to each other for their ability to land in *good* local minima that correspond to practically useful solutions.

Marco will firstly discuss the most typical constraints considered by current methods, connecting them with practical problem and specific features of fabrication hardware. Secondly, the most prominent decomposition heuristics will be analyzed and compared to each other, putting side by side surface and volumetric methods, methods based on boolean operators and methods based on clustering. Interesting cross relations between decomposition strategies and the type of constraints they support will also be discussed and emphasized.

4. Presenters

Peng Song is an Assistant Professor of Computer Science at Singapore University of Technology and Design (SUTD). Prior to joining SUTD in 2019, he was a research scientist at EPFL, Switzerland. He received his PhD from Nanyang Technological University, Singapore in 2013. Peng's research interest is in computer graphics, with a particular focus on geometry modeling, computational design, and digital fabrication. He has published over 30 research papers, including 10 SIGGRAPH/SIGGRAPH Asia technical papers and 1 Eurographics STAR paper on Computational Assemblies [WSP21b]. He has served as a co-organizer of a weekly web series on Computational Fabrication [BSS*21], and a program committee member of several leading conferences including SIGGRAPH Asia and Pacific Graphics. Peng's research contribution is mainly in the field of Computational Assemblies. He has proposed original computational solutions to understand, model, and design complex assemblies, including interlocking assemblies [SFCO12, FSY*15, WSP18], reconfigurable assemblies [SFJ*17, TSW*19], architectural assemblies [SFG*13, MSY*14, WSIP19], and mechanical assemblies [SWT*17, XFS*20, CSSL21].

Ziqi Wang recently finished his Ph.D. at the School of Computer and Communication Sciences at EPFL. He received his bachelor degree in Mathematics in 2017 from the University of Science and Technology of China. His research focuses on geometry processing, architecture geometry, and digital fabrication. During his Ph.D., he studied the computational analysis and design of structurally stable assemblies with rigid parts. His proposed methods are used to generate globally interlocking assemblies [SDW*16, WSP18], topological interlocking assemblies [WSIP19], and assemblies with cone-joints [WSP21a]. Together with Prof. Peng Song and Prof. Mark Pauly, he has published a Eurographics STAR paper [WSP21b] that reviews the recent advancement in computational assemblies.

Marco Livesu is a tenured researcher at the National Research Council of Italy, where he works at the Institute for Applied Mathematics and Technological Information (CNR-IMATI). Prior to joining CNR, he was a postdoctoral researcher at the University of British Columbia (Canada) and University of Cagliari (Italy), where he also obtained his PhD in 2014. His research is mostly focused on geometry processing for manufacturing and engineering applications. He regularly publishes in top tier computer graphics journals and conferences, including SIGGRAPH, SIGGRAPH Asia, Eurographics and Symposium on Geometry Processing. He also serves as program committee member and reviewer for these and many other conferences and journals in the field. In 2015 he was the recipient of the Alain Bensoussan PostDoctoral Fellowship (ERCIM) and in 2021 he obtained the SGP Dataset Award for the HexaLab project [BTP*19]. He is also the creator and lead developer of CinoLib [Liv19], a C++ library for the processing of surface and volumetric meshes. He co-authored an EG STAR on 3D printing [LEM*17], a book that covers the whole digital manufacturing pipeline from design to fabrication [ALL*18] and various papers on related topics [ACA*19, MLS*18, LCA19]. In these works, he explored both additive and subtractive manufacturing techniques, as well as non-trivial combinations of those [TCA*20].

References

[ACA*19] Araújo C., Cabiddu D., Attene M., Livesu M., Vining N., Sheffer A.: Surface2Volume: Surface segmentation conforming assemblable volumetric partition. *ACM Trans. on Graph. (SIGGRAPH) 38*, 4 (2019), 1:1–1:16. 1, 4

[ALL*18] Attene M., Livesu M., Lefebvre S., Funkhouser T., Rusinkiewicz S., Ellero S., Martínez J., Bermano A. H.: *Design, Representations, and Processing for Additive Manufacturing*. Morgan & Claypool Publishers, 2018. 2, 4

[BCMP18] Bickel B., Cignoni P., Malomo L., Pietroni N.: State of the art on stylized fabrication. *Comp. Graph. Forum 37*, 6 (2018), 325–342. 2

[BSS*21] Bickel B., Schulz A., Song P., Piovarci M., Mei Y., Chen R.: Computational fabrication seminar, 2021. https://computational-fabrication.org/. 4

[BTP*19] Bracci M., Tarini M., Pietroni N., Livesu M., Cignoni P.: Hexalab.net: an online viewer for hexahedral meshes. *Computer-Aided Design 110* (2019). Bracci and Tarini joint first authors. doi:10.1016/j.cad.2018.12.003. 4

[CPMS14] Cignoni P., Pietroni N., Malomo L., Scopigno R.: Field-aligned mesh joinery. *ACM Trans. on Graph. 33*, 1 (2014), 11:1–11:12. 1, 2

[CSSL21] Cheng Y., Sun Y., Song P., Liu L.: Spatial-temporal motion control via composite cam-follower mechanisms. *ACM Trans. on Graph. (SIGGRAPH Asia) 40*, 6 (2021). 4

[FAG*20] Filoscia I., Alderighi T., Giorgi D., Malomo L., Callieri M., Cignoni P.: Optimizing object decomposition to reduce visual artifacts in 3D printing. *Comp. Graph. Forum (Eurographics) 39*, 2 (2020), 423–434. 1

[FSY*15] Fu C.-W., Song P., Yan X., Yang L. W., Jayaraman P. K., Cohen-Or D.: Computational interlocking furniture assembly. *ACM Trans. on Graph. (SIGGRAPH) 34*, 4 (2015), 91:1–91:11. 4

[GM15] Ghandi S., Masehian E.: Review and taxonomies of assembly and disassembly path planning problems and approaches. *Computer-Aided Design 67-68* (2015), 58–86. 3

[LCA19] Livesu M., Cabiddu D., Attene M.: slice2mesh: a meshing tool for the simulation of additive manufacturing processes. *Computers & Graphics 80* (2019), 73 – 84. doi:10.1016/j.cag.2019.03.004. 4

[LEM*17] Livesu M., Ellero S., Martínez J., Lefebvre S., Attene M.: From 3D models to 3D prints: an overview of the processing pipeline. *Comp. Graph. Forum (Eurographics STAR – State of The Art Report) 36*, 2 (2017), 537–564. 4

[Liv19] Livesu M.: cinolib: a generic programming header only c++ library for processing polygonal and polyhedral meshes. *Transactions on Computational Science XXXIV* (2019). https://github.com/mlivesu/cinolib/. doi:10.1007/978-3-662-59958-7_4. 4

[LYH*15] Luo S.-J., Yue Y., Huang C.-K., Chung Y.-H., Imai S., Nishita T., Chen B.-Y.: Legolization: Optimizing LEGO designs. *ACM Trans. on Graph. (SIGGRAPH Asia) 34*, 6 (2015), 222:1–222:12. 1

[MLS*18] Muntoni A., Livesu M., Scateni R., Sheffer A., Panozzo D.: Axis-aligned height-field block decomposition of 3D shapes. *ACM Trans. on Graph. 37*, 5 (2018), 169:1–169:15. 1, 4

[MSY*14] Mellado N., Song P., Yan X., Fu C.-W., Mitra N. J.: Computational design and construction of notch-free reciprocal frame structures. In *Proc. Advances in Architectural Geometry* (2014), pp. 181–197. 4

[SDW*16] Song P., Deng B., Wang Z., Dong Z., Li W., Fu C.-W., Liu L.: CofiFab: Coarse-to-fine fabrication of large 3D objects. *ACM Trans. on Graph. (SIGGRAPH) 35*, 4 (2016), 45:1–45:11. 4

[SFCO12] Song P., Fu C.-W., Cohen-Or D.: Recursive interlocking puzzles. *ACM Trans. on Graph. (SIGGRAPH Asia) 31*, 6 (2012), 128:1–128:10. 2, 3, 4

[SFG*13] Song P., Fu C.-W., Goswami P., Zheng J., Mitra N. J., Cohen-Or D.: Reciprocal frame structures made easy. *ACM Trans. on Graph. (SIGGRAPH) 32*, 4 (2013), 94:1–94:10. 4

[SFJ*17] SONG P., FU C.-W., JIN Y., XU H., LIU L., HENG P.-A., COHEN-OR D.: Reconfigurable interlocking furniture. *ACM Trans. on Graph. (SIGGRAPH Asia) 36*, 6 (2017), 174:1–174:14. 4

[SFLF15] SONG P., FU Z., LIU L., FU C.-W.: Printing 3D objects with interlocking parts. *Comp. Aided Geom. Des. 35-36* (2015), 137–148. 3

[SWT*17] SONG P., WANG X., TANG X., FU C.-W., XU H., LIU L., MITRA N. J.: Computational design of wind-up toys. *ACM Trans. on Graph. (SIGGRAPH Asia) 36*, 6 (2017), 238:1–238:13. 4

[TCA*20] TAMELLINI L., CHIUMENTI M., ALTENHOFEN C., ATTENE M., BARROWCLOUGH O., LIVESU M., MARINI F., MARTINELLI M., SKYTT V.: Parametric shape optimization for combined additive–subtractive manufacturing. *JOM 72*, 1 (2020), 448–457. doi:https://doi.org/10.1007/s11837-019-03886-x. 4

[TSW*19] TANG K., SONG P., WANG X., DENG B., FU C.-W., LIU L.: Computational design of steady 3D dissection puzzles. *Comp. Graph. Forum (Eurographics) 38*, 2 (2019), 291–303. 4

[WOD09] WHITING E., OCHSENDORF J., DURAND F.: Procedural modeling of structurally-sound masonry buildings. *ACM Trans. on Graph. (SIGGRAPH Asia) 28*, 5 (2009), 112:1–112:9. 3

[WSIP19] WANG Z., SONG P., ISVORANU F., PAULY M.: Design and structural optimization of topological interlocking assemblies. *ACM Trans. on Graph. (SIGGRAPH Asia) 38*, 6 (2019), 193:1–193:13. 1, 3, 4

[WSP18] WANG Z., SONG P., PAULY M.: DESIA: A general framework for designing interlocking assemblies. *ACM Trans. on Graph. (SIGGRAPH Asia) 37*, 6 (2018), 191:1–191:14. 3, 4

[WSP21a] WANG Z., SONG P., PAULY M.: MOCCA: Modeling and optimizing cone-joints for complex assemblies. *ACM Trans. on Graph. (SIGGRAPH) 40*, 4 (2021), 181:1–181:14. 1, 3, 4

[WSP21b] WANG Z., SONG P., PAULY M.: State of the art on computational design of assemblies with rigid parts. *Comp. Graph. Forum (Eurographics) 40*, 2 (2021). 2, 3, 4

[WSW*12] WHITING E., SHIN H., WANG R., OCHSENDORF J., DURAND F.: Structural optimization of 3D masonry buildings. *ACM Trans. on Graph. (SIGGRAPH Asia) 31*, 6 (2012), 159:1–159:11. 3

[XFS*20] XU H., FU T., SONG P., ZHOU M., FU C.-W., MITRA N. J.: Computational design and optimization of non-circular gears. *Comp. Graph. Forum (Eurographics) 39*, 2 (2020), 399–409. 4

[YKGA17] YAO J., KAUFMAN D. M., GINGOLD Y., AGRAWALA M.: Interactive design and stability analysis of decorative joinery for furniture. *ACM Trans. on Graph. 36*, 2 (2017), 20:1–20:16. 1, 2