

CONTINUOUS PROPERTY GRADATION FOR MULTI-MATERIAL 3D-PRINTED OBJECTS

C. Altenhofen*†, T. H. Luu*†, T. Grasser*†, M. Dennstädt*†, J. S. Mueller-Roemer *†, D. Weber
†, and A. Stork†

*Fraunhofer Institute for Computer Graphics Research IGD, 64283 Darmstadt, Germany
†Technische Universität Darmstadt, 64283 Darmstadt, Germany

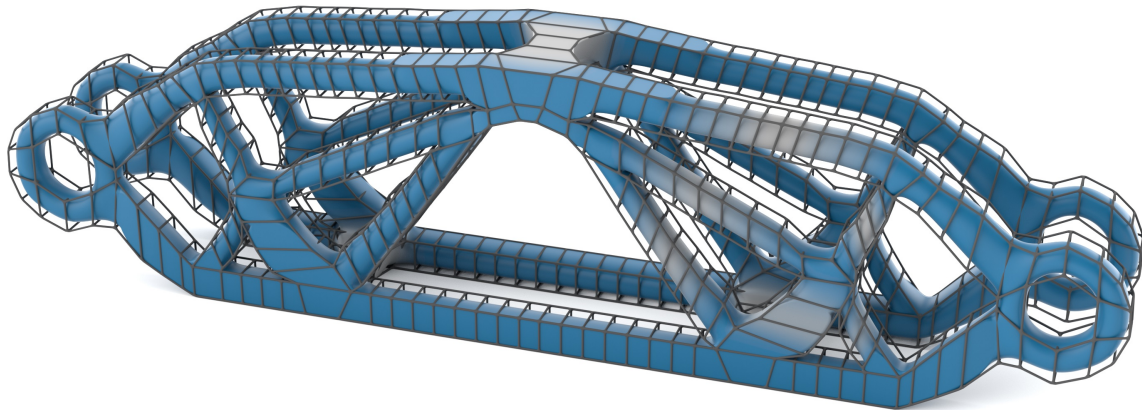


Figure 1: Multi-material 3D printable model with smooth material gradation generated with our approach.

Abstract

Modern AM processes allow for printing multiple materials. The resulting objects can be stiff/dense in some areas and soft/porous in others, resulting in distinct physical properties. However, modeling material gradients is still tedious with current approaches, especially when smooth transitions are required. Current approaches can be distinguished into a) NURBS-BReps-based and b) voxel-based. In case of NURBS-BReps, discrete material distributions can be modeled by manually introducing separate shells inside the object; smooth gradation can only be approximated in discrete steps. For voxel representations, gradation is discrete by design and comes along with an approximation error. In addition, interacting on a per-voxel basis is tedious for the designer/engineer. We present a novel approach for representing material gradients in volumetric models using subdivision schemes, supporting continuity and providing elegant ways for interactive modeling of locally varying properties. Additionally, the continuous volumetric representation allows for on-demand sampling at any resolution required by the 3D printer.

Introduction

Modern 3D printers can vary mechanical properties of objects locally, e.g. by mixing different base materials or varying process parameters – the latter is less common, still. Traditionally,

CAD systems using BRep-NURBS only allow / support assigning a material for each solid or topological lump. However, modern 3D printers, such as the Stratasys J750, allow for material variation on a per-voxel basis. Obviously, defining a lump per voxel would be an impractical approach considering the size of the individual “elements” being printed. Of course, one could consider coarsening the voxels representing the geometry of the object, but that would limit the accuracy of the shape representation and the freedom in material variation. Alternatively, it is possible to assign material information on a per-slice level, but slicing is an approximating discretization performed in a post-process to the modeling stage. Thus, the actual geometry would not carry the material information. Considering the huge number of slices created, assigning material information would again be a tedious process.

So, wouldn't it be ideal to enable the designer to define material properties on a continuous representation allowing for editing on a moderate number of control points? For geometry, CAD systems use splines to represent continuously differentiable shapes. In isogeometric analysis (IGA), the same concept is extended towards modeling and simulating on the same representation scheme (trivariate splines) and has recently been used to assign materials to volumes. However, trimmed splines are not necessarily the most easy-to-use representation for organic shapes as can be seen in the movie and game industry where subdivision schemes are much more common than in mechanical engineering.

In this paper, we propose subdivision schemes to model shape and material property distribution for 3D-printed objects, because 3D printing is particularly well suited for generating organic (bionic) shapes that cannot be manufactured with traditional subtractive production technology. Leveraging subdivision allows us to enable the designer to define the shape and properties he or she wants to achieve at certain regions in space on the same representation scheme. The continuous representation can be evaluated at any point in space with the resolution of the 3D printing machine – no discretization or conversion required. The properties defined on a control mesh are smoothly interpolated into the interior of the object. The interpolated property is matched as closely as possible by mixing the materials provided by the 3D printer according to their properties / behavior. Properties can also be assigned to internal control points giving the needed freedom to the user to specify continuously graded material properties throughout the object.

Related Work

Modeling for Additive Manufacturing: In engineering, the shape of a part is usually defined using computer aided design (CAD). For this purpose, trimmed NURBS patches, which bound the part to be produced as a continuous surface, have been established in industrial practice as the standard representation for the description of free-form geometry. Subdivision surfaces generated by methods such as Catmull-Clark [1] or Loop subdivision [2] are an alternative modeling technique for geometric objects. Algorithms for efficient and accurate representation of subdivision surfaces [3] and methods for sketch-based generation of subdivision models [4, 5] have already been developed. These forms of surface representations are now established as a standard in computer graphics and are gaining increasing acceptance in industrial surface design. However, although closed surface meshes implicitly describe a volumetric object, they are not considered to be volumetric meshes, since they do not contain any information about the interior of the volume

they enclose. Common tools for polygonal modeling used in design and computer animation, such as Maya [6], Blender [7], Rhino [8], and SolidWorks [9] only support surface modeling.

For fully volumetric models, different representations have to be used. In the context of IGA [10], trivariate NURBS volumes are used as a representation for volumetric models. In 2016, Massarwi and Elber presented a framework for modeling volumetric B-spline models in their so-called “VRep” (“Volume Representation” analogously to the commonly used “BRep”) [11]. Volumetric subdivision methods, such as the ones presented by Joy and MacCracken [12], Schaefer et al. [13], or Chang et al. [14], transfer the concept of subdivision surfaces to volumetric meshes. In addition to vertices, edges, and faces, a volumetric subdivision mesh consists of cells and has inner faces to separate them.

When it comes to assigning materials to volumetric models, two main concepts exist: image-based and voxel-based. For image-based material assignment, a 2D image is created for every layer, carrying the material values in its color channels. With voxel-based material modeling, material values are explicitly assigned to individual cells in a 3D voxel grid. GrabCAD Print [15] and Autodesk Monolith [16] are just two tools on the market that allow volumetric material editing with the above techniques. However, for both concepts, material gets assigned to the model after slicing/voxelization. In contrast, our approach operates on the original model before slicing/voxelization takes place and is therefore independent from the final printing resolution and orientation.

Exchange Formats for Volumetric Models: Currently, the primary exchange format for 3D-printed models is the stereolithography (STL) file. Standard STL files only contain a set of triangles describing the outer surface of a mesh, without any information about color, material or scale. While more advanced formats exist, they all describe the model only by its boundary, i.e., as a boundary representation. For example, the 3D Manufacturing Format (3MF), developed by a consortium large CAD, CAM, and 3D-printing companies, adds colors, thumbnail images, and digital signatures but is still based on triangulated surface meshes (see, e.g., the 3MF specification [17]).

While these advanced formats allow assigning multiple surface colors via vertex colors or even using texture images, actual material types can only be specified by name, turning mapping to an actual material into a manual process:

<base> attribute name: Specifies the material name, intended to convey design intent, for the purpose of aiding users in mapping to print materials. [17]

Furthermore, due to the BRep nature of current formats, material gradations or mixes within the volume cannot be described directly, even though current 3D printers can already produce material mixes. Approximations such as multiple thin shells are used and often created using a tedious, manual process.

Modeling

As mentioned earlier, we investigate the use of subdivision schemes to model shape and material property distribution. Subdivision models create continuous shapes while supporting complex and unstructured topologies.

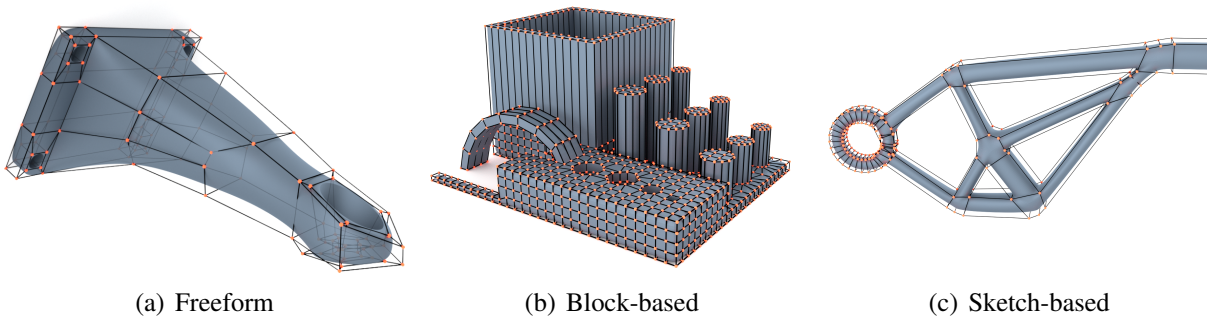


Figure 2: Three subdivision models created with different modeling techniques. (a) has been created with freeform polygonal modeling, while (b) has been created with a block-based approach and (c) is the result of sketching.

Subdivision Modeling: Different techniques exist to create subdivision models. Although most tools implement polygonal modeling (block-based modeling), procedural modeling with analytical functions as well as sketch-based modeling are also possible. Sketch-based approaches are particularly useful for modeling strut-heavy structures that are most often the result of topology optimization techniques. Figure 2 shows three subdivision models created with different modeling techniques.

Material Definition using Subdivision Models: In addition to the geometric modeling, the modeling of material distributions constitutes an important extension for 3D printing. When applying the subdivision rules to a given control mesh, a refined set of points is created by interpolating the original ones. When representing geometry, every control point only has a 3D position (x,y,z) . However, subdivision rules and the underlying interpolation scheme can be applied to points of arbitrary dimension. Therefore, we introduce higher-dimensional control points, e.g. 4D, 5D or 6D, with a geometric position in 3D space as well as material parameters such as density, Young’s modulus, or Poisson’s ratio. Figure 3 shows an example of control points with attached stiffness values. Note how the stiffness values are interpolated over the subdivided geometry, creating a smooth gradient. Following the subdivision scheme, material properties can also be defined volumetrically, when creating models with inner control points. This way, models with a soft outer shell over stiff internal structures can be created and very interesting physical behaviors can be achieved.

Subdivision schemes – both for surfaces, as well as volumes – converge towards a smooth limit surface/volume when applying a theoretically infinite number of subdivision steps. For most schemes, the limit can be evaluated mathematically, without performing explicit subdivision steps at all. By “attaching” additional parameters directly to the control points, the limit values of these parameters can be evaluated everywhere in the volume with the same algorithm that is used for

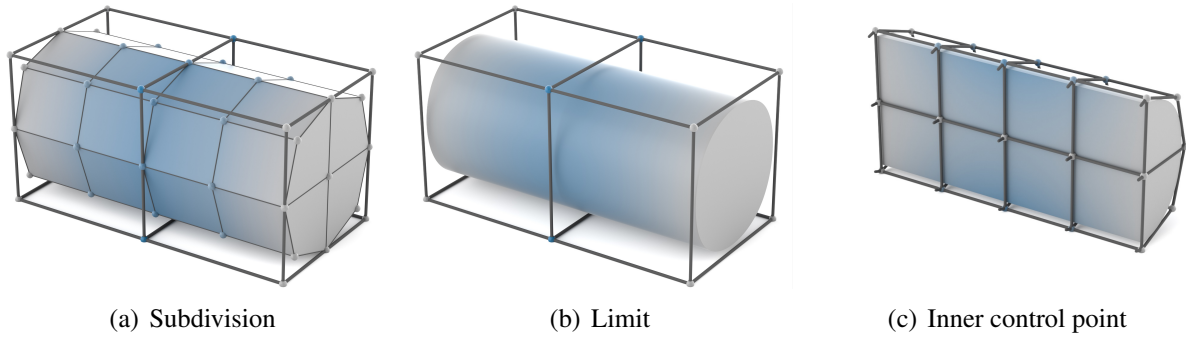


Figure 3: Material properties assigned at the control points and their effect on the subdivided model and the limit. (a) shows an example of a subdivision cylinder, showing the effects on the subdivided surface, while (b) shows the corresponding limit surface. (c) shows a cut through the object, showing inner control points. In this example, white color represents a low stiffness, while blue color represents a high one.

evaluating the limit positions. Figure 3(b) shows an example of evaluating stiffness values on the limit surface as well as in the limit volume.

Interaction Techniques: For evaluating our approach, we implemented a prototype application. It is written as a C++ desktop application and provides two different interaction techniques for applying material properties on the control polygon of a given subdivision model. One way is to directly select individual control points to allow a targeted, local modification. The other way is to use brushing to quickly and intuitively edit a larger number of nodes that are within the brush's radius. Figure 4 shows examples of both techniques. In our examples, geometry colored in blue corresponds to a high stiffness, whereas white areas depict a lower one.

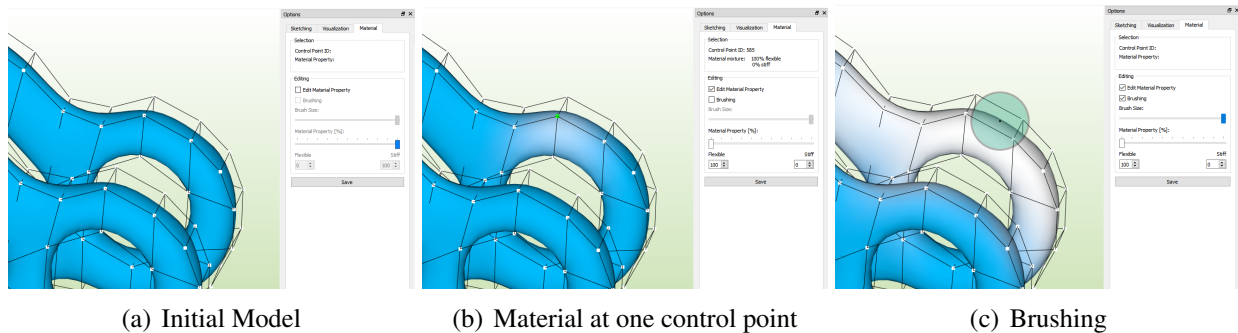


Figure 4: Process of material parameter definition by a user. (a) shows the initial model with a uniform distribution of material parameters. In (b), a different material parameter has been assigned to one control point. Finally, (c) shows a brushing operation, changing the material parameters for multiple control points at once. Blue color corresponds to very stiff and white color to a very flexible material parameters.

Export and Storage of Volumetric Multi-Material Models

Now that our subdivision model has been created and the distribution of its material parameters has been defined, the model has to be prepared for 3D printing. As available slicers cannot process subdivision geometry directly, but require tessellated input, we discretize our model into a tetrahedral mesh and store it using a volumetric attribute-based data format.

Implicit Tetrahedrization: As presented in our previous paper [18], we exploit the already existing volumetric structure given by the subdivision scheme to create the tetrahedral cells out of the hexahedral subdivision cells resulting from n explicit subdivision steps. The number of subdivision steps n can be increased or decreased in order to control the resolution of the resulting tetrahedral mesh. As the entire meshing process can be expressed as a set of pre-computable matrix-matrix multiplications and a final matrix-vector multiplication, the tetrahedral mesh can be re-generated very efficiently, when modifying the geometry or the material parameters on the original control mesh – as long as the topology of the control mesh remains unchanged. All the calculations to get from the set of original control points to the tetrahedral mesh are performed independently on the x , y and z coordinate of every three-dimensional control point. Therefore, it seems natural to apply the same calculations also on the material properties when dealing with 4D, 5D or 6D control points described in Section 4.2. The material parameters attached to every control point are “subdivided” and “meshed” in the same way as the control point’s geometric position.

Export and Storage Format: Since current exchange formats cannot represent volumetrically varying material parameters efficiently, we propose a volumetric, attribute-based model exchange format. Instead of describing a volume’s boundary using a triangle mesh, we use a tetrahedral mesh to describe it’s volume. Furthermore, instead of only storing material colors and names, we also store physical properties such as density, Young’s modulus and Poisson’s ratio. It is then up to the printer driver to approximate these properties as well as possible using techniques such as material mixes (potentially via dithering) or metamaterials (mesostructures). This mirrors how 2D color printers receive a color specification which they approximate using halftoning or contoning, without the user having to select specific inks or dot patterns. Furthermore, by storing physical properties with the model, it becomes directly simulatable.

The storage format is based on HDF5 2.0 [19], a self-describing, hierarchical, binary file-format that supports n -dimensional datasets, stores metadata, has free cross-platform implementations, and supports compression. Extensibility is achieved through the use of universally unique identifiers (UUID) [20], 128-bit identifiers out of which 122 are random for approximately $5.3 \cdot 10^{36}$ possibilities. As this effectively rules out collisions, no central administration of extensions is required. When adding a new physical property, a new UUID is chosen and used from that point on to describe that property’s semantic meaning. Additionally, we store a user-readable name and SI-unit exponents to clarify intent if the UUID is not yet known by the tool used.

To be able to describe sharp edges as well as gradients, we use a multi-indexed approach. Each of the four vertices of a tetrahedron is given an index into the position property that also describes the model’s topology. Additional indices are attached to each of these four indices for

every physical property. Therefore, neighboring tetrahedra can use the same position but different property indices to create a sharp transition.

We explicitly do not store any slicing information, resolution or specific materials, as all of these would closely couple a model to a specific printer model, vendor, or technology.

Volumetric Slicing

To print the 3D objects with a 3D printer that supports multiple materials, the objects have to be converted into the according input format for that particular printer. In case of the Stratasys J750 printer that we used for the experiments in this paper, as well as for other printers, the input format is a stack of 2D images that determine, which material is placed at which position in a three-dimensional voxel grid. Every image represents one layer. These layers are finally printed on top of each other. Creating the layer information from the input 3D object is called “slicing”.

The slicing process is performed by the software Cuttlefish [21] and consists of two steps: First the surface of the exported tetrahedral mesh is rasterized and after that the voxelization of the interior takes place.

Surface rasterization: During the surface rasterization step, the input 3D object’s surface is tested against the three dimensional voxel grid and all voxels that intersect with the surface of the object are marked as occupied surface voxels. The implemented surface rasterization method is described in Section 3.4 of [22].

Interior voxelization: The interior of the 3D object is then filled by evaluating the winding number of the voxels as described in [23]. To speed up the process, a subset of all voxels is selected as seed points, for which the winding number gets evaluated. The information of these seed points is then propagated throughout the volume by using a floodfill approach. An overview of this method can be found in Paragraph 108-110 of the patent [24].

After the voxelization process, the application has performed an inside-outside segmentation of the set of voxels and now materials must be assigned to create the output for the 3D printer. For this we interpret the material values per vertex of every tetrahedron as tonal values – probabilities that a material is placed. These attributes are assigned to the surface voxels during the rasterization phase by interpolating between the three vertices that form a face of the tetrahedron. As described in Section 4 of [25], the material values are then interpolated from the surface into the volume of a voxelized tetrahedron which results in the assignment of tonal values to each voxel. These tonal values are then halftoned using a 3D blue noise mask, which results in a concrete assignment of materials.

The described implementation is missing a linearization step that takes the per vertex color, interprets it as stiffness, and converts it into a material probability that correspond to that stiffness. One possible approach to obtain a linearization is to print targets with different material mixtures and measure their stiffness. The resulting data set will plot the non-linearity of the stiffness and interpolating between the samples will allow us to obtain a material probability for stiffness that can

be achieved with this material. In case more materials are used, the sampling dimension increases and the interpolations have to be performed in higher dimensions. This described missing workflow is implemented in Cuttlefish although sample data is missing.

Results

To print our example models, we used a Stratasys J750 printer and a variety of photopolymer materials. As a stiff material we decided on a cyan Vero photopolymer. As a flexible material we chose Agilus30, a PolyJet photopolymer with high tear-resistance, capable of withstanding repeated flexing and bending. To represent our smooth gradation of material properties, both materials are blended together with the corresponding ratio for every voxel. Some of the printed parts alongside their multi-material digital counterparts are shown in Figure 5. The printing process for a single model takes approximately 1 hour and 45 minutes and uses about 110 grams of material. The exact numbers depend on the actual material distribution.

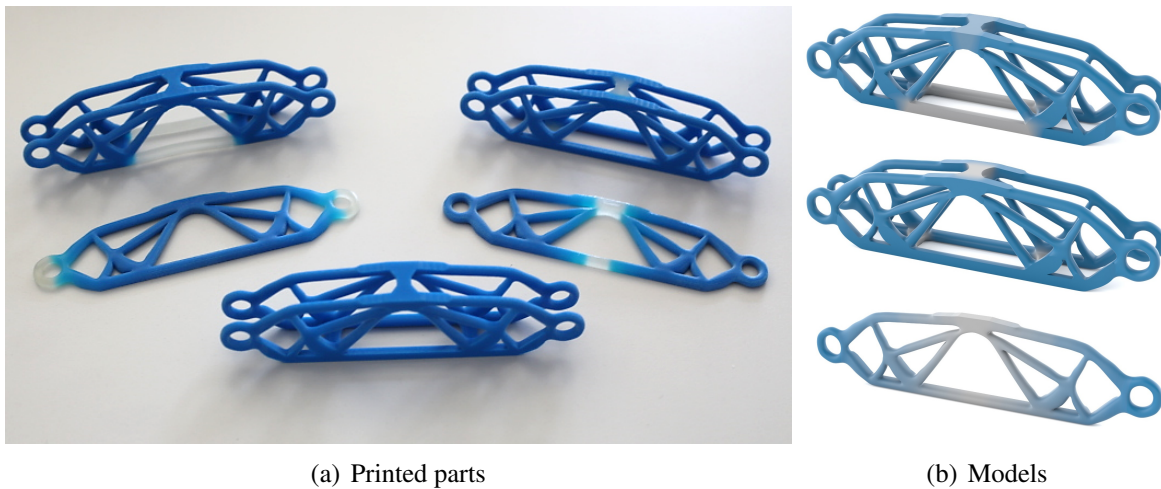


Figure 5: Printouts of multiple example models with the Stratasys J750 (a) and the corresponding digital models created with our approach (b). Blue areas indicate a high stiffness, while white areas indicate a low one.

Conclusion and Future Work

In this paper, we present a novel approach for defining volumetrically varying material parameters in 3D models. By attaching properties such as density, Young's modulus, or Poisson's ratio to the control points of a subdivision control mesh, we can represent smooth gradients of parameters across and inside the model when applying explicit subdivision steps or direct limit evaluation. The approach offers great flexibility in geometric modeling and property definition with a relatively low number of degrees of freedom. The ability to directly work with a volumetric representation enables great exploitation of modern AM possibilities.

With our approach, we define smooth variations of material properties. However, even though the chosen subdivision scheme supports crease edges and sharp features for the geometry, discrete boundaries and discontinuous transitions are not possible for material properties.

For Catmull-Clark subdivision models, the property gradation follow higher-order basis functions similar to the ones of cubic B-splines. When converting the subdivision model into a tetrahedral mesh for slicing, we discretize the smooth material properties for every tetrahedron. Instead of calculating constant properties for every element by averaging the values at its corners, the value at a given position inside the tetrahedron could be approximated using the same subdivision basis functions as we do for the modeling. To speed up the subdivision and tetrahedrization process, we plan a parallel GPU implementation using a highly efficient GPU data structure [26]. In order to eliminate the tetrahedrization completely, direct slicing of the subdivision geometry is envisaged.

Acknowledgements

This work was supported by the EU-project CAxMan (project number 680448) which is co-funded by the *Horizon 2020* Framework Program H2020-FoF-2015 of the European Union. More information can be found at <https://www.caxman.eu/>. We thank Alan Brunton for his implementations of the voxelizer and the distance transform as well as André Schulz for the implementation of the halftoner.

References

- [1] E. Catmull and J. Clark, “Recursively generated B-spline surfaces on arbitrary topological meshes,” *Computer-Aided Design*, vol. 10, no. 6, pp. 350–355, 1978.
- [2] C. Loop, “Smooth subdivision surfaces based on triangles,” Ph.D. dissertation, University of Utah, 1987.
- [3] V. Settgast, K. Müller, C. Fünfzig, and D. Fellner, “Adaptive tessellation of subdivision surfaces,” *Computers & Graphics*, vol. 28, no. 1, pp. 73–78, 2004.
- [4] M. Bein, S. Havemann, A. Stork, and D. Fellner, “Sketching subdivision surfaces,” in *Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling*. ACM, 2009, pp. 61–68.
- [5] Z. Ji, L. Liu, and Y. Wang, “B-mesh: a modeling system for base meshes of 3d articulated shapes,” in *Computer Graphics Forum*, vol. 29, no. 7. Wiley Online Library, 2010, pp. 2169–2177.
- [6] Autodesk. (2018, June) Autodesk Maya. [Online]. Available: <http://www.autodesk.de/products/maya/overview>
- [7] B. Foundation. (2018, June) Blender. [Online]. Available: <https://www.blender.org/>
- [8] R. McNeel and Associates. (2018, June) Rhino. [Online]. Available: <https://www.rhino3d.com/>
- [9] D. Systems. (2018, June) SolidWorks. [Online]. Available: <http://www.solidworks.de/>

- [10] T. J. Hughes, J. A. Cottrell, and Y. Bazilevs, “Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement,” *Computer methods in applied mechanics and engineering*, vol. 194, no. 39, pp. 4135–4195, 2005.
- [11] F. Massarwi and G. Elber, “A B-spline based framework for volumetric object modeling,” *Computer-Aided Design*, vol. 78, pp. 36–47, 2016.
- [12] K. I. Joy and R. MacCracken, “The refinement rules for Catmull-Clark solids,” Citeseer, Tech. Rep., 1999.
- [13] S. Schaefer, J. Hakenberg, and J. Warren, “Smooth subdivision of tetrahedral meshes,” in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*. ACM, 2004, pp. 147–154.
- [14] Y.-S. Chang and H. Qin, “A framework for multi-dimensional adaptive subdivision objects,” in *Proceedings of the ninth ACM symposium on Solid modeling and applications*. Eurographics Association, 2004, pp. 123–134.
- [15] Stratasys. (2018, June) GrabCAD Print. [Online]. Available: <https://grabcad.com/print>
- [16] Autodesk. (2018, June) Monolith. [Online]. Available: <http://www.monolith.zone/>
- [17] 3MF Consortium, *3D Manufacturing Format: Core Specification & Reference Guide*, 2018, version 1.2.1. [Online]. Available: <https://3mf.io/3d-manufacturing-format/>
- [18] C. Altenhofen, F. Schuwirth, A. Stork, and D. Fellner, “Implicit mesh generation using volumetric subdivision,” in *Workshop on Virtual Reality Interaction and Physical Simulation*, ser. VRIPHYS’17, F. Jaillet and F. Zara, Eds., Lyon, France, 2017. [Online]. Available: <http://dx.doi.org/10.2312/vriphys.20171079>
- [19] The HDF5 Group, *HDF5 File Format Specification*, 2017, version 2.0. [Online]. Available: <https://support.hdfgroup.org/HDF5/doc1.8/H5.format.html>
- [20] P. Leach, M. Mealling, and R. Salz, *RFC 4122, A Universally Unique Identifier (UUID) URN Namespace*, 2005. [Online]. Available: <https://www.ietf.org/rfc/rfc4122.txt>
- [21] F. IGD, <https://www.cuttlefish.de/>, accessed: 2018-06-29.
- [22] M. Schwarz and H.-P. Seidel, “Fast parallel surface and solid voxelization on GPUs,” *ACM Trans. Graph.*, vol. 29, no. 6, pp. 179:1–179:10, Dec. 2010. [Online]. Available: <http://doi.acm.org/10.1145/1882261.1866201>
- [23] A. Jacobson, L. Kavan, and O. Sorkine-Hornung, “Robust inside-outside segmentation using generalized winding numbers,” *ACM Trans. Graph.*, vol. 32, no. 4, pp. 33:1–33:12, Jul. 2013. [Online]. Available: <http://doi.acm.org/10.1145/2461912.2461916>
- [24] A. Brunton, C. A. Arıkan, P. Urban, and T. M. Tanksale, “Method for joint color and translucency 3d printing and a joint color and translucency 3d printing device,” Patent US 20180117849A1, 10 24, 2016. [Online]. Available: <https://patents.google.com/patent/US20180117849A1/en>

- [25] A. Brunton, C. A. Arikian, T. M. Tanksale, and P. Urban, “3d printing spatially varying color and translucency,” *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 37, no. 4, pp. 175:1–157:13, Aug. 2018.
- [26] J. S. Mueller-Roemer, C. Altenhofen, and A. Stork, “Ternary sparse matrix representation for volumetric mesh subdivision and processing on GPUs,” *Computer Graphics Forum*, vol. 36, no. 5, pp. 59–69, 2017.