INTRODUCTION

Ground-breaking *hardware* advances from the past decade or two have blurred the dividing line between the real and virtual worlds beyond recognition. Nowadays, one can scan a real-world object using a simple smartphone app and render it in a virtual scene, and even predict how a musical instrument will sound just from a 2D image of it. The inverse direction of this challenge has been even more dramatically revolutionized by the democratization of 3D printing technology, accessible today to many consumers.

However, *software* has been lagging largely behind this revolutionary change in the Geometry Processing pipeline. Most algorithms in the toolbox (be it smoothing, shape analysis, deformation or mesh reconstruction) are designed to work on certain, fixed input geometries. This fails to incorporate the uncertainties and errors associated with real-world geometric data capture, as well as the possibility of including further capturing steps as intermediate steps in the pipeline. Similarly, virtual 3D modelling tools rarely account for whether the designed object is actually fabricable or not, leading to disappointing outcomes when attempting to bring the virtual asset into the real world.

My vision is that of an accessible and seamless integration between the real and virtual worlds: virtual design must be fabrication-aware and the Geometry Processing pipeline must be revolutionized to fully exploit data captured from the real world. My previous research tackles some of the specific challenges one encounters when making this vision a reality, and sets the stage for my proposed future work.

PAST EVIDENCE OF SUCCESS

Developability of surfaces

Traditionally, 3D modelling software focuses on providing the user with the maximum possible freedom so that they can create the precise asset they want to include in a scene or animation. Indeed, most are raw user interfaces controlling the digital representation of a shape. Little to no attention is given to whether the virtual object can actually be fabricated or even exist in the real world.

A paper sphere can be modelled in Blender 2.9, but not in real life!

As an example, take a sheet of paper and attempt to fold it into a sphere. You will find that this is impossible; in other words, a paper sphere is not a fabricable shape. Mathematically, we say that a sphere is not a *developable* shape. However, existing 3D design software, unaware of this fact, allows one to build a virtual model of this sphere, and then render it with a texture that replicates that of a piece of paper to produce a realistic image (see figure above). The same applies to wood that is bent for ship construction or steel sheets for architecture. In [Sellán et al. 2020a], we show that one characterize these developable shapes mathematically and use that insight to either constrain the user to them during the design process or to project a freely designed existing surface into a developable one.

Previous works in the study of developability are very limited in the class of developable shapes they can consider [Rabinovich et al. 2018, 2019] [Rose et al. 2007] or present deep discretization and orientation dependence [Stein et al. 2018]. Our proposed method imposes no restriction on the number of developable patches nor on the distribution of ruling lines, and is discretization, resolution and orientation independent.

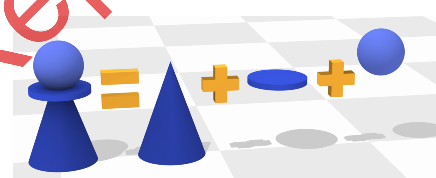
Opening and Closing Surfaces

Modelling these non-stretchable surfaces is just one example of using properties of real-world materials to guide fabrication-aimed design. Another possibility is that the latter is not constrained by the object's *material*, but by the *equipment* with which it will be fabricated. For instance, we may design an object with the intention to then build it with the aid of a CNC machine with a round end mill (see inset). Not every object can be constructed in this way; in fact, whether this is possible or not can be characterized mathematically in terms of a bound on the object's surface's curvature. This leads to an immediate research question: "Given an arbitrary object, what is the closest object that can be fabricated?"

Mathematically, this amounts to computing a morphological operation on a given shape. Unfortunately, presently available methods to answer this question (which usually amount to successive computation of isosurfaces) are either too slow or produce extreme aliasing. Furthermore, their complexity depends on the size of the whole input object, even if the vast majority is already fabricable. In our upcoming ACM SIGGRAPH Asia paper [Sellán et al. 2020b] we introduce a novel view of this problem, and show one can compute this closing as a surface-only geometric flow that is output-sensitive in complexity.

Geometry Processing on Deconstructed Domains

My research vision is not limited to a world where tools exist to seamlessly transition between these two worlds. I firmly believe these tools must be accessible, and not restricted to a privileged, technologically trained minority. The struggle to make this a reality must begin with 3D virtual modelling, which today lags very much behind 2D in terms of ease-of-use for the general population.



One of the most intuitive ways of designing solid objects is by adding and subtracting simple primitive shapes (like spheres, cones or cubes) to create very complex ones (see inset). While simple, this approach often leads to disappointing results, given that the resulting shapes can be completely unusable within the Geometry Processing pipeline due to the lack of a unified tetrahedralization of the shape. For instance, I may want to design an object that vibrates to produce a specific sound frequency, or an object with certain heat or sound insulation properties, or one which has a given degree of elasticity.

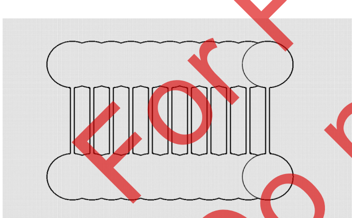
Mathematically, responding to these challenges amounts to solving a Partial Differential Equation (PDE) defined in the solid object. However, in practice, this requires knowing a three-dimensional grid of the solid object or tetrahedralization, and building an object by adding and substracting simpler ones does not provide us with this tiling: putting together a tetrahedralization of a sphere with a tetrahedralization of a cube does not produce a tetrahedralization of the union of a sphere and a cube. In [Sellán et al. 2019], presented at the Eurographics Symposium on Geometry Processing, however, we circumvent this problem and introduce a first-of-its-kind way of solving PDEs defined on unions of simple shapes by utilizing only the grids of these primitives and coupling them algebraically in the form of linear equality constraints.

MY PLAN FOR THE FUTURE

My vision of *integrating the real and virtual worlds* has short and mid term deliverables. On the shorter term end of the spectrum, there are many improvements of my previously published works that would increase their impact and relevance even further. For instance, the method in [Sellán et al. 2020a] is restricted to work on depth images of surfaces, while these are more often represented as triangular meshes. The work in [Sellán et al. 2019] is built to work only on unions of primitive shapes, while a full modelling pipeline can include other set operations like intersections and subtractions. Removing limitations like these would certainly make their contributions to the research field and their applications to industry software even richer.

Efficient and robust Swept Volumes

A common way of constructing virtual objects is by sweeping an input shape across 3D space, and outputting the solid region of space covered by the shape as it moves (see [Abdel-Malek et al. 2006]). An exciting application of this design framework is to Virtual Reality sculpting (see inset image, from Adobe Oculus Medium). Currently existing methods for computing these swept volumes lack efficiency and/or robustness (as a sample, consider [Rossignac et al. 2007; Peternell et al. 2005]): if they are forced to run near to real time, the outputs present stroboscopic aliasing (see inset below, showing aliasing from sweeping a dumbbell shape from left to right).



By drawing insights from physics, I believe the problem can be rephrased with a change in the frame of reference. Instead of a surface moving rigidly across space, the surface can be thought of as fixed and every particle in space moving around it. Therefore, computing the boundary of the swept volume can be rephrased in terms of a particle-surface collision detection problem, which I expect to solve efficiently and robustly by drawing from the fields of non-linear ray tracing and importance sampling. Furthermore, once a point in the boundary of the swept volume is identified, local operations can be carried out to learn the direction in which said boundary continues, avoiding expensive computations on the inside of the shape.

Understanding Signed Distance Fields

Newly possible virtual geometry capture methods like 3D drawing in Virtual Reality and many Machine Learning Computer Vision frameworks have drawn attention to a type of implicit geometric representation called *Signed Distance Fields (SDFs)*. While this representation presents many advantages in terms of ease of use and modelling speed, many understudied challenges remain when integrating these representations into the Computer Graphics pipeline, which traditionally prefers explicit representations like triangular meshes. For instance, what does it mean to deform an SDF rigidly? How can one interpolate between two existing SDFs while ensuring that every step of the interpolation is also an SDF? What does it mean to define a geometric surface flow on an SDF, and how can one compute it? Given a noisy or erroneous SDF-like function, how can one project to the "closest" real SDF? In the future, I will research the answers to these questions, which will have a tremendous impact in the practicalities of losslessly using SDF data for traditional Computer Graphics applications.

Uncertain Captured Geometry

Input (real scan) Our output

Many challenges remain in the question of transferring real world geometry into the virtual world. Often, this is done via 3D scanning, and many works in the literature have been proposed for smoothing or post-processing the outputs of these scanners (among them, our [Sellán et al. 2020a], a figure of which is shown as inset). Dealing with this scan data, which is often noisy and filled with artifacts, often means making deterministic, final decisions to questions like "Is this depth data point an outlier, or does it really represent the real-world geometry?" or "Is this a hard crease present in the scanned object, or is it noise provoked by the user while scanning?"

Today, the answers to these questions are final and can dramatically impact what we conclude about the object further down the Geometry Processing pipeline. If we wrongly identified some extreme depth data points as outliers and discarded them, we may wrongly conclude the object to be more smooth than it really is, leading to very different material properties. Similarly, if we identified one-directional noise to be a crease in the real geometry instead of the result of noise, our elasticity or resistance calculations about the object may be completely off. In the future, I will experiment with adding a measure of uncertainty or stochasticity to the capture of real-world geometry. If we can measure the *possibility* of a certain data point being an outlier or a product of noise instead of making a deterministic choice, we should be able to make *probabilistic* conclusions about the properties of the object. Ideally, this would even inform the user to re-scan or put more emphasis on certain parts of the shape, leading to more precise results.

In summary, my plan for the future is to tackle specific challenges associated with the interaction between real-world and virtual-world geometry, while keeping an open mind and also taking on more broad ideas with the potential of revolutionizing the way we think of transporting geometry from one to the other. My previous work and publications in the field show a promising start in this direction, and I find myself thrilled to fulfill that promise in the following years.

REFERENCES

[Abdel-Malek et al. 2006] Abdel-Malek, K., Yang, J., Blackmore, D., & Joy, K. (2006). Swept volumes: fundation, perspectives, and applications. *International Journal of Shape Modeling*, *12*(01), 87-127.

[Peternell et al. 2005] Peternell, M., Pottmann, H., Steiner, T., & Zhao, H. (2005). Swept volumes. *Computer-Aided Design and Applications*, *2*(5), 599-608.

[Rabinovich et al. 2018] Rabinovich, M., Hoffmann, T., & Sorkine-Hornung, O. (2018). Discrete geodesic nets for modeling developable surfaces. *ACM Transactions on Graphics (ToG)*, *37*(2), 1-17.

[Rabinovich et al. 2019] Rabinovich, M., Hoffmann, T., & Sorkine-Hornung, O. (2019). Modeling curved folding with freeform deformations. *ACM Transactions on Graphics (TOG)*, *38*(6), 1-12.

[Rose et al. 2007] Rose, K., Sheffer, A., Wither, J., Cani, M. P., & Thibert, B. (2007, July). Developable surfaces from arbitrary sketched boundaries. *Computer Graphics Forum*.

[Rossignac et al. 2007] Rossignac, J., Kim, J. J., Song, S. C., Suh, K. C., & Joung, C. B. (2007). Boundary of the volume swept by a free-form solid in screw motion. *Computer-Aided Design*, *39*(9), 745-755.

[Sellán et al. 2019] Sellán, S., Cheng, H. Y., Ma, Y., Dembowski, M., & Jacobson, A. (2019, February). Solid Geometry Processing on Deconstructed Domains. In *Computer Graphics Forum* (Vol. 38, No. 1, pp. 564-579).

[Sellán et al. 2020a] Sellán, S., Aigerman, N., & Jacobson, A. (2020). Developability of heightfields via rank minimization. *ACM Transactions on Graphics (TOG)*, *39*(4), 109-1.

[Sellán et al. 2020b] Sellán, S.,Yan Sheng, A., Kesten, J. & Jacobson, A. (2020). Opening and Closing Surfaces. *ACM Transactions on Graphics (TOG), to appear*.

[Stein et al. 2018] Stein, O., Grinspun, E., & Crane, K. (2018). Developability of triangle meshes. *ACM Transactions on Graphics (TOG)*, *37*(4), 1-14.