VESPA: VTK Enhanced with Surface Processing Algorithms

C. Gueunet[1] and T. L. Chhim[1]

[1]Kitware SAS

Abstract

This work introduces the VESPA project, a bridge between geometry processing with the Computational Geometry Algorithms Library (CGAL) and scientific visualization with the Visualization Tool Kit (VTK) and ParaView. After a brief description of these tools, we motivate the use of VESPA through the example of a full processing pipeline detailing the construction of a mold from an initial 3D surface model. This paper illustrates the use of several robust geometry operations as well as the benefits of added interactivity and visualization. As an open source project, VESPA is already publicly available and open to contributions.

CCS Concepts

• ***Mathematics of computing*** → ***Geometric topology;*** • ***Computing methodologies*** → ***Scientific visualization;***

1. Introduction

The fields of scientific visualization and geometry processing are often associated together, for instance when studying and manipulating meshes. While geometry processing algorithms typically transform or generate meshes, their detailed analysis is made possible with scientific visualization through visual displays of the structure, as well as various data processing routines such as mesh clipping, thresholding, contouring, etc.

Most workflows for real life applications or, e.g., topological data analysis usually require conforming 2 or 3-manifold meshes and a strict robustness of the encountered algorithms. Since geometry processing is not at the center of scientific visualization, widespread visualization software such as the Visualization Tool Kit (VTK) library and its derived application ParaView rarely include advanced meshing techniques.

To answer this need, a suitable solution is to rely on specialized tools like the Computational Geometry Algorithms Library (CGAL). This lead to the creation of the VESPA project: VTK Enhanced with Surface Processing Algorithms, which integrates some of CGAL processing power into VTK and ParaView. This paper briefly introduces these tools as well as a complete application example of a processing pipeline using VESPA.

2. Existing Tools

2.1. VTK

In the field of scientific visualization, one of the most popular tools is the VTK library [SAH00] (under BSD-3 licensing) which boasts strong rendering capabilities and proposes a vast number of algorithms, readers, writers, etc. to let users create their own customized processing pipelines and applications. VTK is organized into different modules for IO, data processing, rendering, interaction, etc. and is designed to be extensible through the addition of new modules, making it flexible and adaptable to different types of applications. Due to its nature, coding knowledge is however necessary for efficient usage of the library.

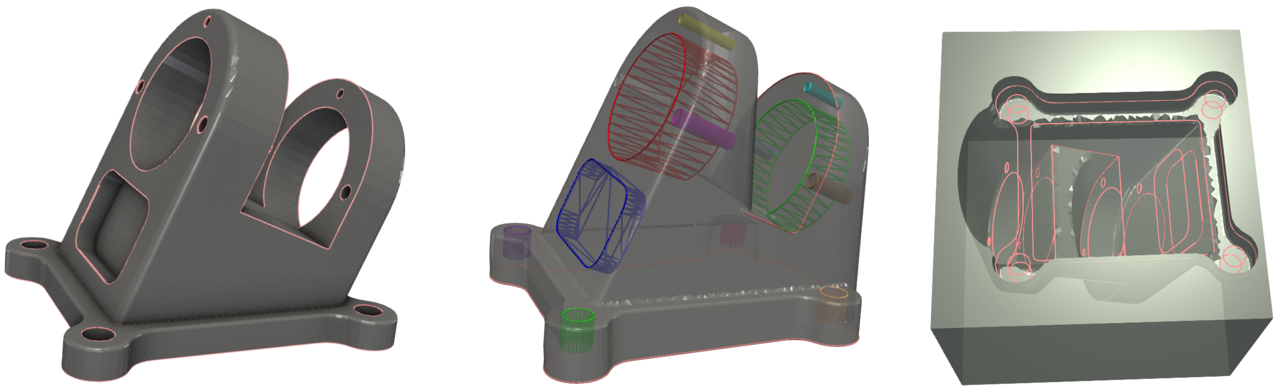
2.2. ParaView

ParaView [AGL05] is a leading open source software (under BSD-3 licensing) based on VTK that offers a great variety of filters for data analysis and scientific visualization. The software is much more user friendly with a graphical user interface that allows users to easily explore and interact with their data. Many file formats are supported and can be readily imported in ParaView for instant manipulation. More features such as new filters, visual representations, toolbars, etc. can also be added via plugins.

2.3. CGAL

CGAL [FP09] is a C++ library that offers a large amount of geometry processing algorithms such as mesh generation, mesh simplification, boolean operations, and more. The library supports a wide range of input meshes through the Boost Graph Library [SLL01]. Despite containing heavily templated C++ classes and requiring solid C++ knowledge, the high performing algorithms and extensive documentation available online make it a library of choice for geometry processing. It should be noted that CGAL is under dual licensing (GPLv3 + Commercial). Therefore, binaries compiled against this library (thus applying to VESPA as well) retain the GPLv3 license unless the commercial license is in use.

DOI: 10.2312/evp.20231074



(a) (b) (c)

Figure 1: *3D mold construction (sharp edges are highlighted in orange): (a) Initial 3D mesh, (b) 3D mesh after hole filling (hole outlines meshed in color), linear extrusion along the vertical unmolding axis and remeshing via alpha wrapping, (c) Final 3D mold obtained after a difference boolean operation with a rectangular box.*

3. VESPA

3.1. Project Description

Although VTK does include several mesh processing filters, some of them only work under specific circumstances or exhibit a suboptimal performance. Moreover, many state-of-the-art algorithms are not currently available either, such as the alpha wrapping method for remeshing, or the mesh deformation. To remedy this gap, the VESPA project was created [VES] (under BSD-3 licensing) to form a bridge between CGAL and VTK by wrapping CGAL algorithms as VTK filters.

By extension, these algorithms can also be brought to the end user via a ParaView plugin. Therefore, the robustness and efficiency of CGAL operations become accessible with all of the benefits from using ParaView (visualization, interactive selection, dynamically changing options, export, advanced rendering, etc.). For now, input meshes need to be triangulated, which can be done easily using triangulation filters available in VTK and ParaView. This limitation is due to the current implementation in charge of converting VTK meshes into CGAL data models back and forth, but a more generic version could be easily implemented. Additionally, CGAL algorithms often require their input to be manifold.

Moreover, VESPA was devised as an open source project that is easy to expand by wrapping more CGAL operations, thus becoming more versatile and further increasing its reach as a user-friendly solution to geometry processing. In the future, verification algorithms aimed at checking for mesh conformity should be included as well.

3.2. Processing Pipeline Example

To illustrate the usage of VESPA filters, this section describes a complete geometry processing pipeline with the goal of creating a mold for a given object based on its 3D mesh. This type of pipeline is adapted to real life applications, such as the creation of medical prostheses, dental impression, mold casting, etc.

Starting with the initial mesh in Figure 1a, tunnel-like openings and cavities are removed from the initial mesh using the "Patch Filling" filter. The mesh outlines of these holes are shown in Figure 1b in color. To achieve this, interactive selection in ParaView is particularly useful as it allows the user to quickly identify, select, and fill in these sections.

In the next step, we must ensure that the mold can slide smoothly during removal along the unmolding axis. This is accomplished with a linear extrusion performed on all cells along that axis, resulting in a highly non-conforming mesh with self intersections. Using the "Alpha Wrapping" filter with a small offset, a valid mesh is recomputed, producing smoothed edges for easier unmolding, as illustrated in Figure 1b.

The last operation consists in the creation of the mold itself by computing the difference between a rectangular block and the transformed mesh from the previous steps. The "Boolean Operation" filter can take these two input meshes and perform the difference to produce the negative of the transformed mesh corresponding to the casting die as shown in Figure 1c. The resulting mesh is perfectly conforming and could be further improved with filters for smoothing, refinement, and so on.

4. Conclusion

In this work we introduced the open source VESPA project which extends the data visualization tools VTK and ParaView with powerful mesh processing algorithms from the CGAL library. VESPA has the benefit of being very user friendly, both for quick and easy mesh transformation and visualization in ParaView, as well as integration into existing VTK applications. Furthermore, the project was designed to make external contributions as simple as possible. As a result, the list of available geometry processing filters is expected to keep growing in the future.

References

[AGL05] AHRENS J., GEVECI B., LAW C.: ParaView: An end-user tool for large data visualization. *The Visualization Handbook 717*, 8 (2005). 1

[FP09] FABRI A., PION S.: CGAL: The computational geometry algorithms library. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems* (2009), pp. 538–539. 1

[SAH00] SCHROEDER W. J., AVILA L. S., HOFFMAN W.: Visualizing with VTK: a tutorial. *IEEE Computer graphics and applications 20*, 5 (2000), 20–27. 1

[SLL01] SIEK J. G., LEE L.-Q., LUMSDAINE A.: *The Boost Graph Library: User Guide and Reference Manual, The*. Pearson Education, 2001. 1

[VES] The VESPA Project. https://gitlab.kitware.com/vtk-cgal/vespa. Accessed: 2023-04-11. 2