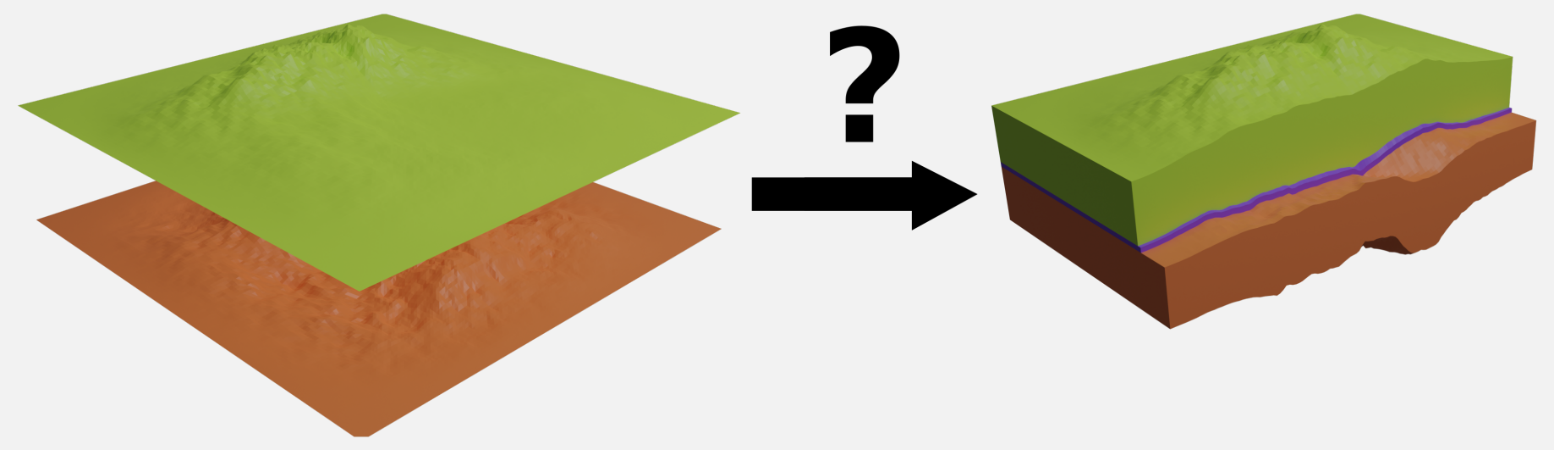
A first step towards the inference of geological topological operations

Romain Pascual[1]
Hakim Belhaouari[2], Agnès Arnould[2], Pascale Le Gall[1]
[1]MICS, CentraleSupélec, University Paris-Saclay, France; [2]XLIM UMR CNRS 7252, University of Poitiers, France

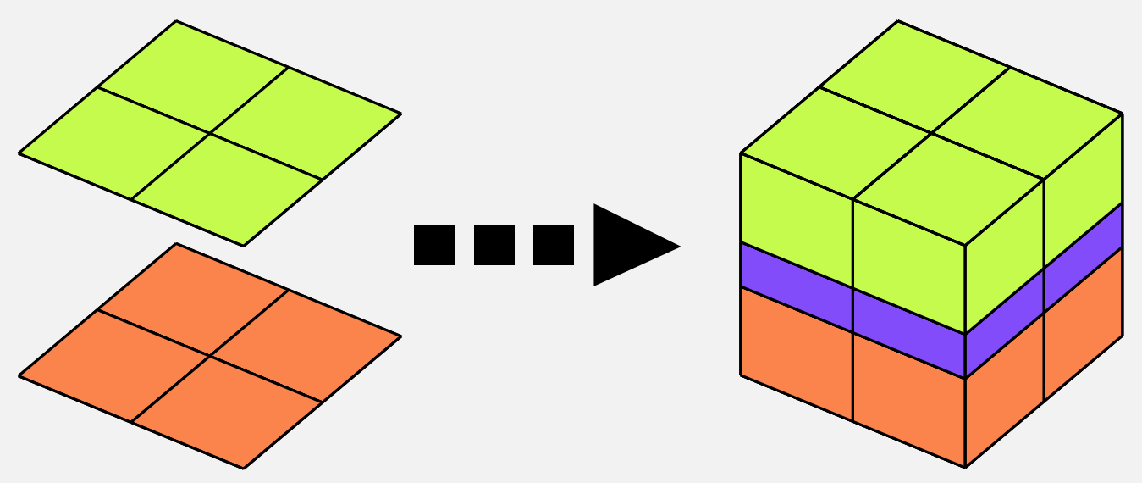
1. Problematic and motivating context

Question: How to obtain this operation ?



Before: 2 surfaces, 64516 faces, and 258064 vertices.
After: 48387 volumes, 290322 faces, and 2322576 vertices.

Intuition: It is a layering operation !



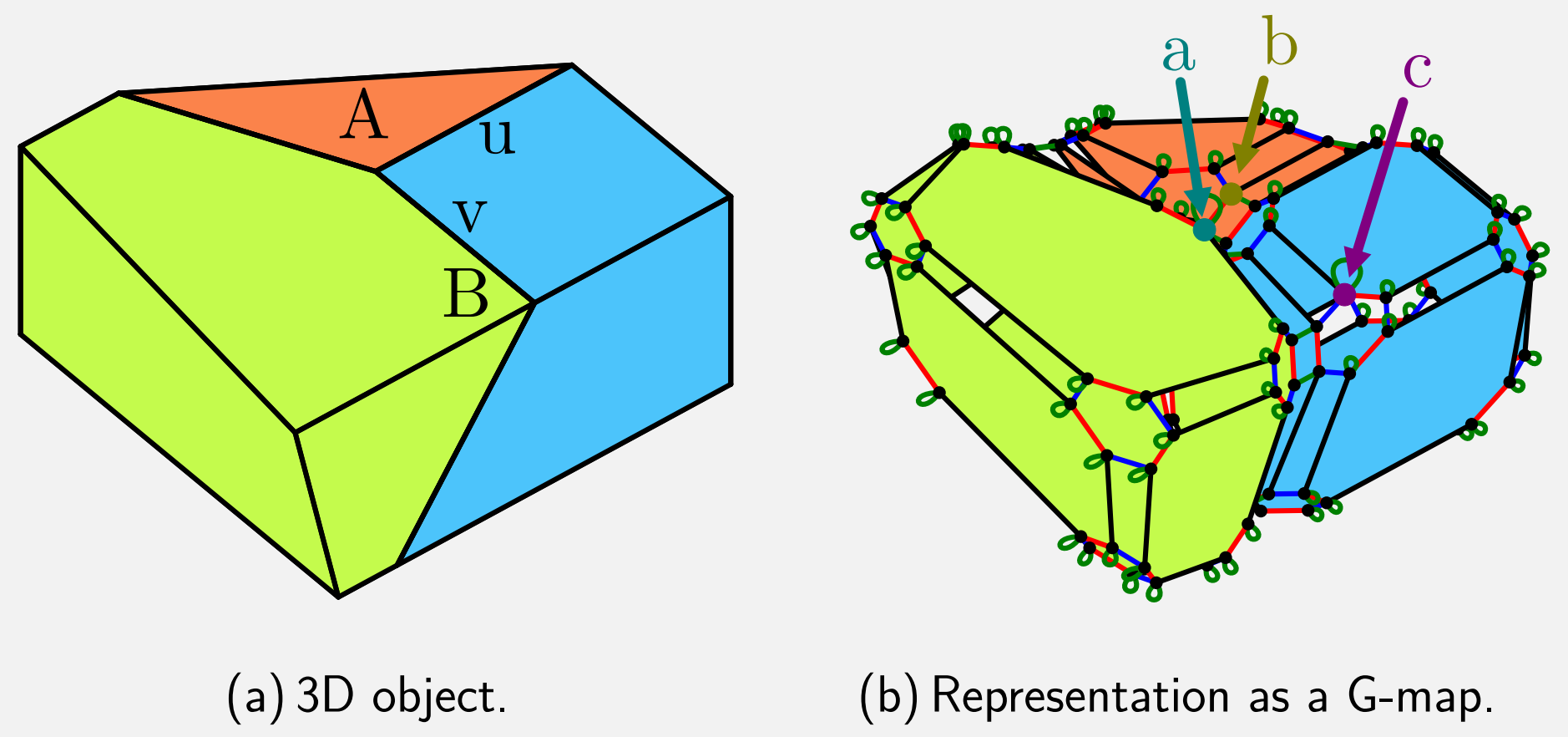
Before: 2 surfaces, 8 faces, and 32 vertices.
After: 12 volumes, 72 faces, and 288 vertices.

We offer to infer operations from a representative example.

Applicability: Should produce a valid operation.
Robustness: Should yield an operation usable in a broader context.
Simplicity: Should not require user knowledge of the data structure.
Time efficiency: Should infer in an acceptable amount of time.

Exploiting Jerboa's platform, we can use objects created in various software.

2. Topological structure



(a) 3D object. (b) Representation as a G-map.

Figure 1: Generalized maps (G-maps) [DL14] as an arc-labeled graph encode the object's topological cells.

3. Modeling operations

We developed a dedicated tool with Jerboa's platform [BALGB14] supporting imports from common file formats.

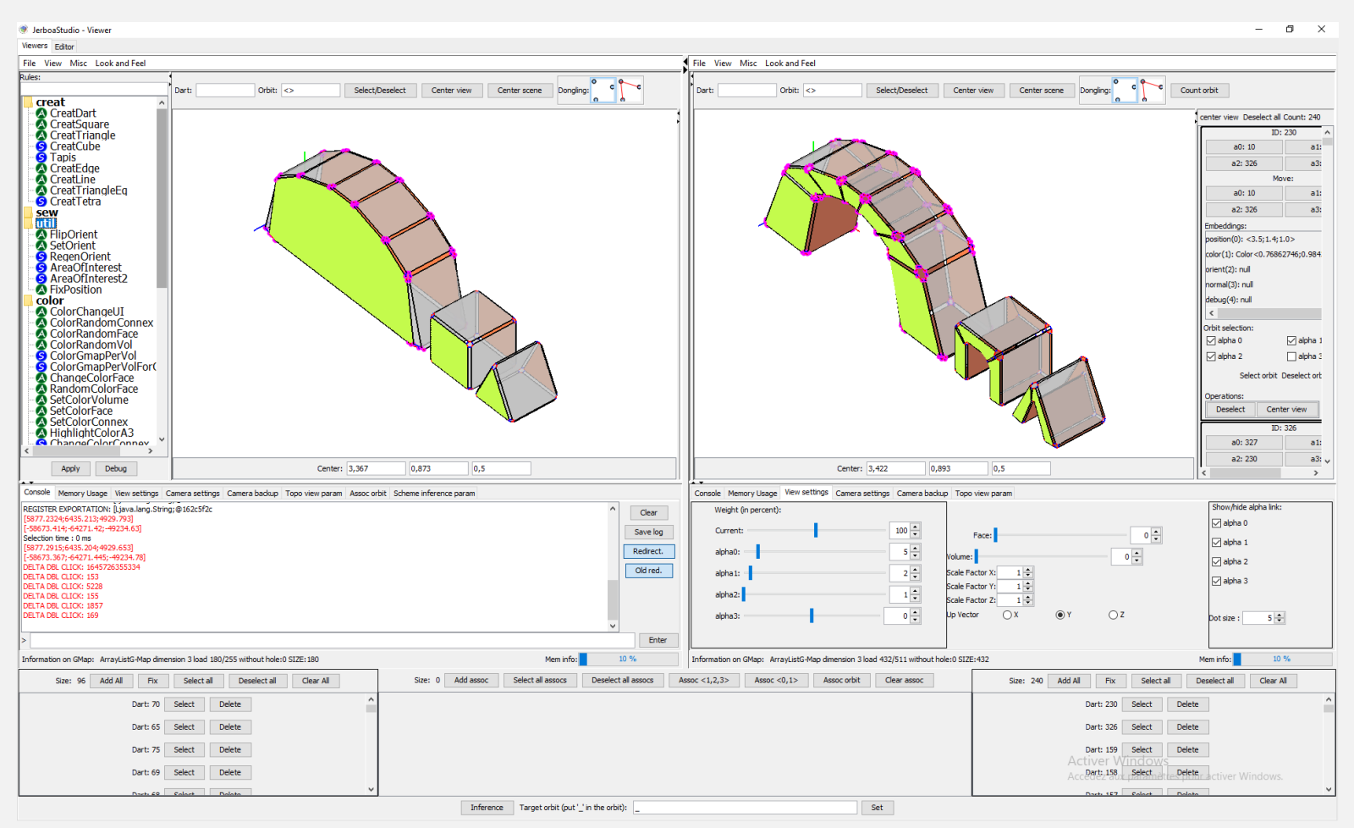


Figure 2: Viewer tab.

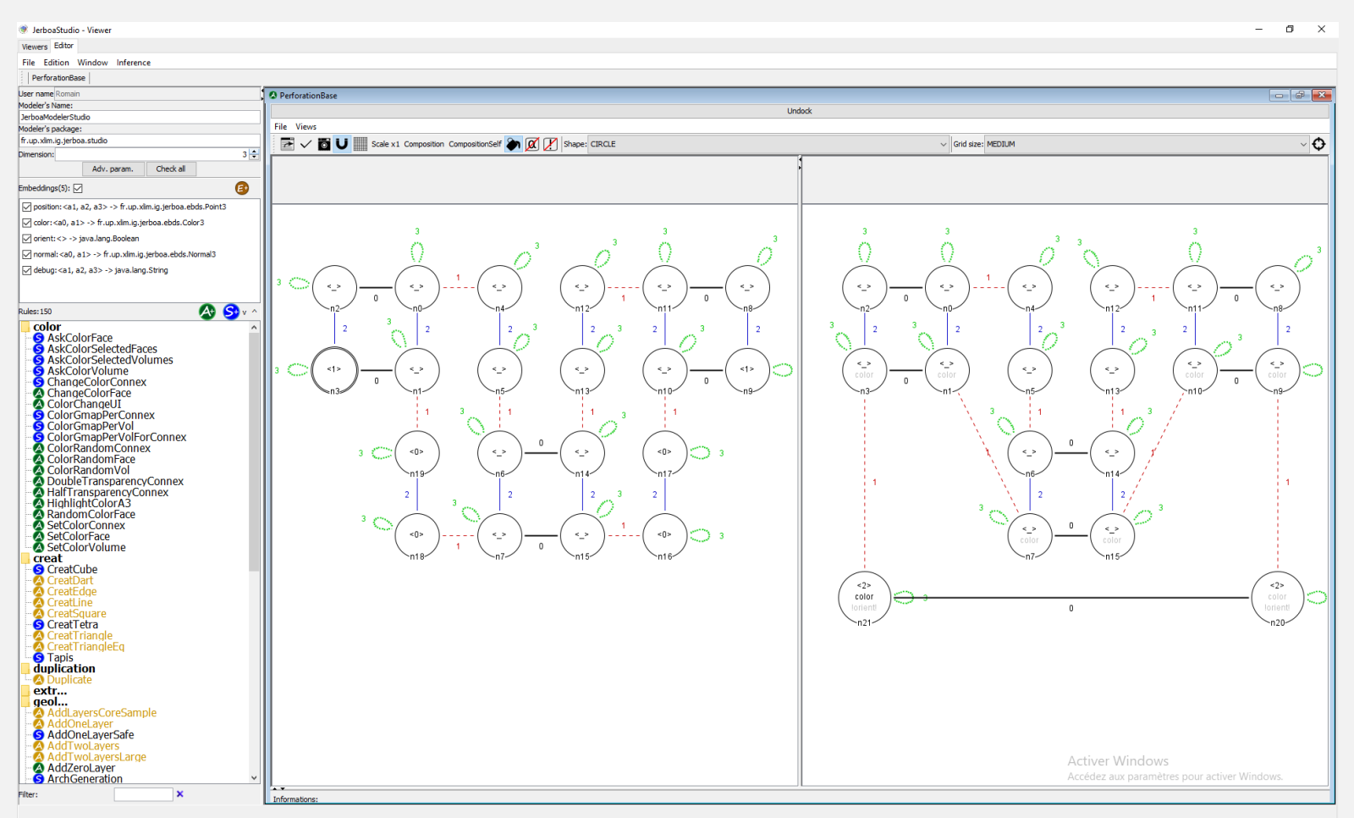


Figure 3: Editor tab.

The viewer tab (Fig. 2) contains both instances used for the inference. The editor tab (Fig. 3) allows visualizing the inferred operations and adding missing geometric computations.

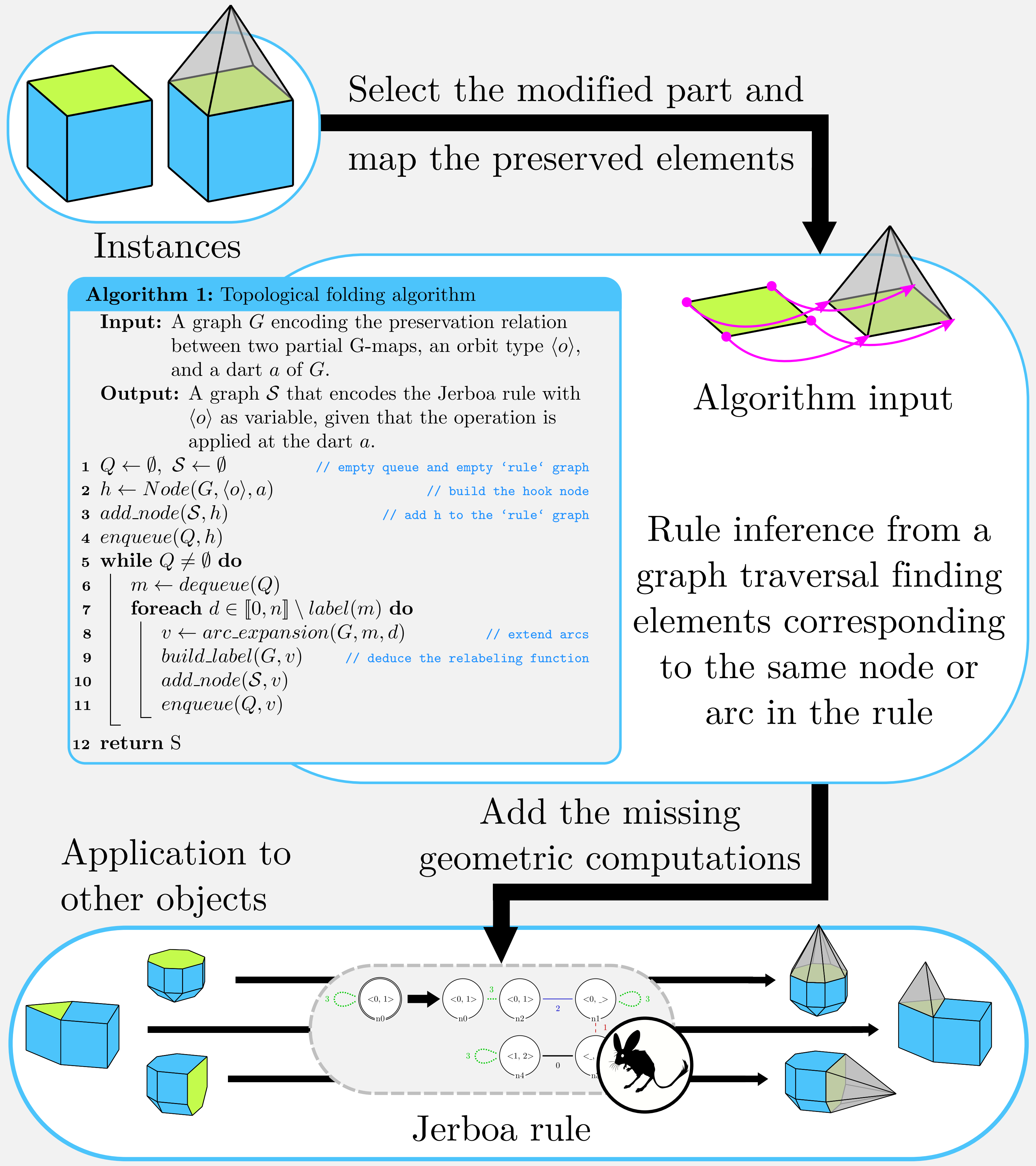
Take a picture to access the demo

[BALGB14] Belhaouari H., Arnould A., Le Gall P., Bellet T.: Jerboa: A Graph Transformation Library for Topology-Based Geometric Modeling. In **Graph Transformation (ICGT)** (2014), Giese H., König B., (Eds.).

[DL14] Damiand G., Lienhardt P.: **Combinatorial Maps: Efficient Data Structures for Computer Graphics and Image Processing.** 2014.

4. Method

We reverse the instantiation process by folding a graph that encodes the elements modified by the operation.



Instances

Select the modified part and map the preserved elements

Algorithm 1: Topological folding algorithm

Input: A graph G encoding the preservation relation between two partial G-maps, an orbit type $\langle o \rangle$, and a dart a of G.
Output: A graph $\mathcal{S}$ that encodes the Jerboa rule with $\langle o \rangle$ as variable, given that the operation is applied at the dart a.

```
Q ← ∅, S ← ∅                          // empty queue and empty 'rule' graph
h ← Node(G, ⟨o⟩, a)                   // build the hook node
add_node(S, h)                        // add h to the 'rule' graph
enqueue(Q, h)
while Q ≠ ∅ do
    m ← dequeue(Q)
    foreach d ∈ ⟦0, n⟧ \ label(m) do
        v ← arc_expansion(G, m, d)    // extend arcs
        build_label(G, v)             // deduce the relabeling function
        add_node(S, v)
        enqueue(Q, v)
return S
```

Algorithm input

Rule inference from a graph traversal finding elements corresponding to the same node or arc in the rule

Add the missing geometric computations

Application to other objects

Jerboa rule

Figure 4: How to obtain a topological operation from an example.

5. Applications for procedural modeling (geology)

Layering: (a) no inter-layer, (b) one inter-layer, (c) two inter-layers, (d) six inter-layers and their inferred operations.

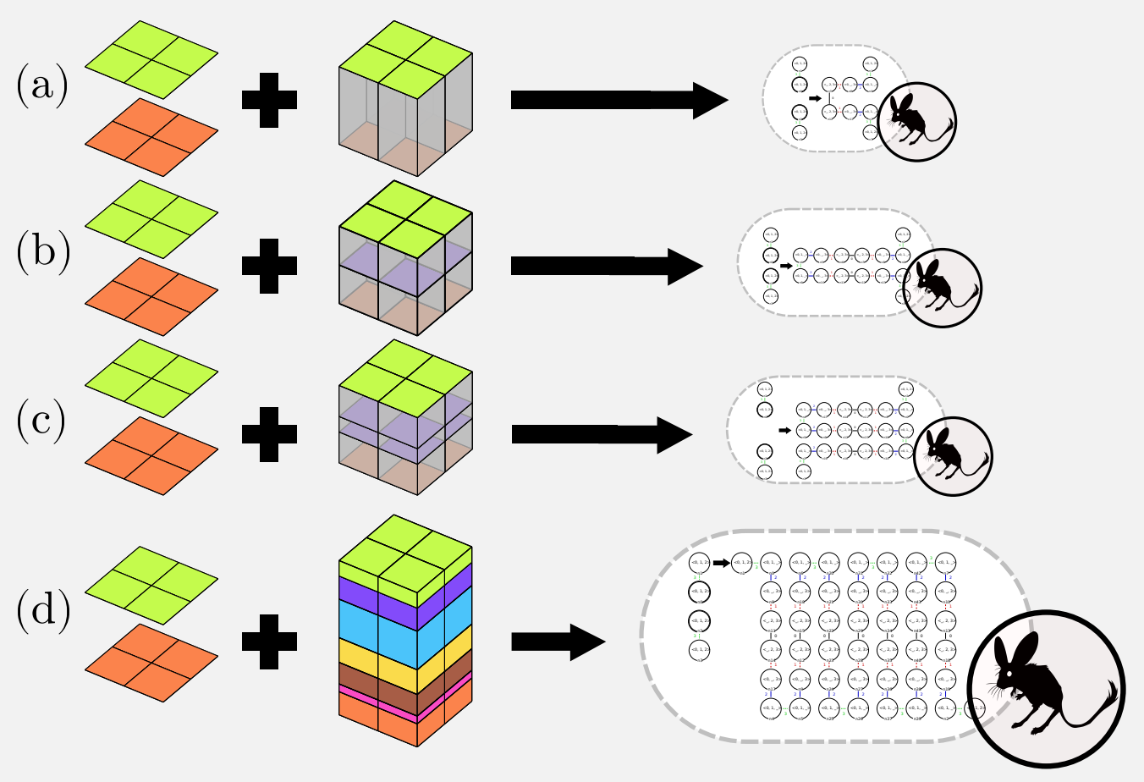


Figure 5: Layering inference.

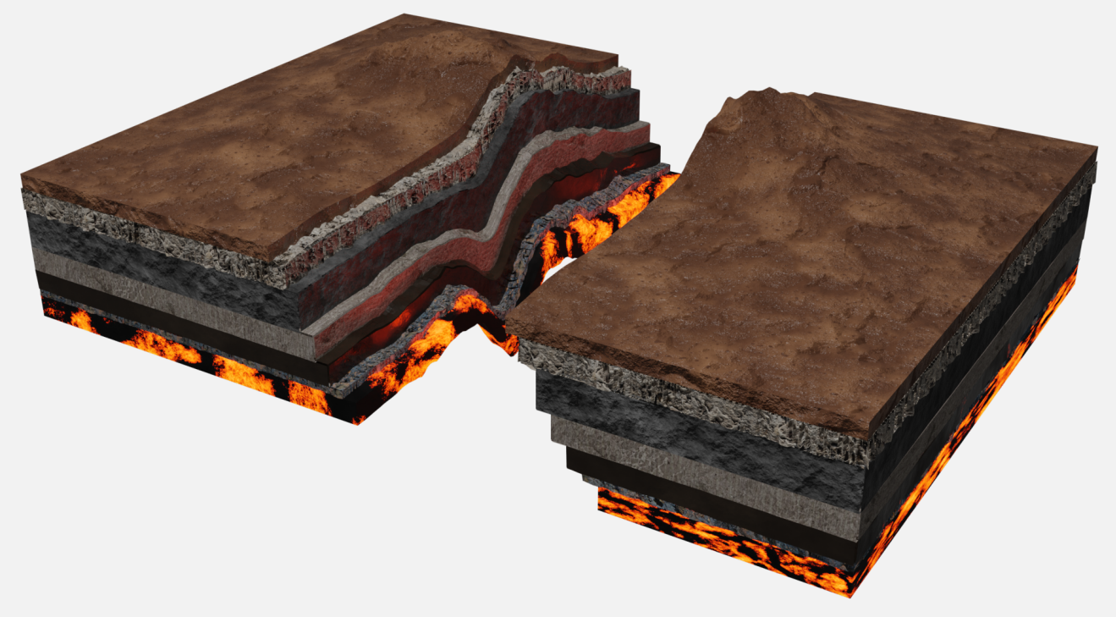


Figure 6: Final rendered subsoil.

Arch generation: (a) three-step and (b) one-step procedure by removing the base of a prism, and (c) two-step procedure by hole perforation.

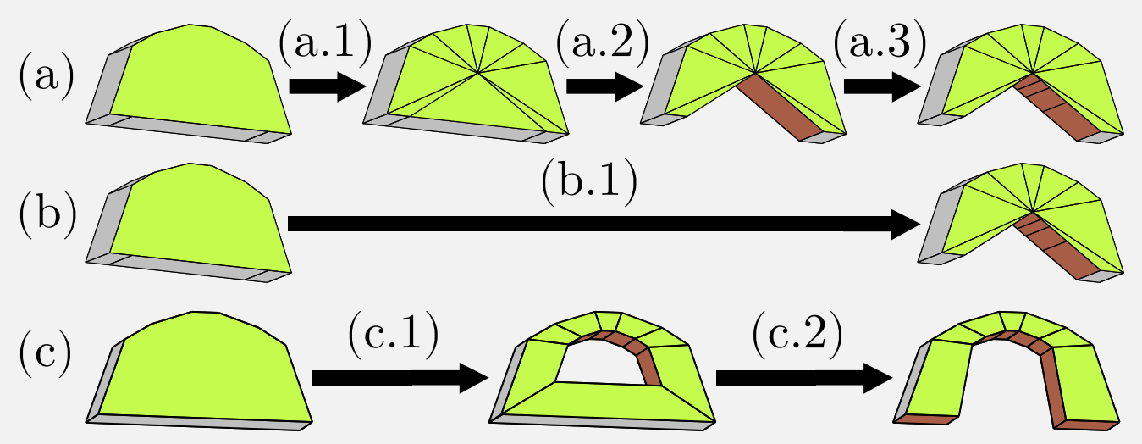


Figure 7: Arch generation inference.

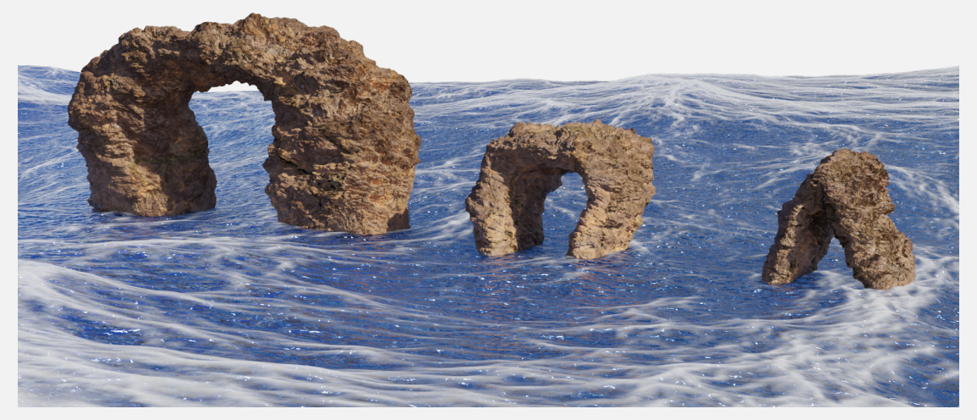


Figure 8: Rendered arches in water.

The operations presented here are part of a larger procedural workflow and were obtained through our inference mechanism. We can either refine the generation by inferring simpler operations (line (a)) or realize a single step generation with the threat of inferring an operation that completely fixes the topology (line (b)).

6. Results and conclusion

Inference time:

Time in ms for the inference of the operations from applications of section 5.

	Figure 5				Figure 7					
Operation	a	b	c	d	a.1	a.2	a.3	b.1	c.1	c.2
Time (ms)	3	4	4	4	2	5	2	28	4	5

Limitations:

- The operation should modify a connected part of the object for the algorithm to work.
- When the algorithm does not find regularities, it produces an operation tailored to the instance given as input (e.g. operation (b) from Fig. 7).

Conclusion:

We infer any constructive or destructive topological operations from their description on a representative example (facilitated by the regularity of generalized maps and the genericity of graph-based rules).

EUROGRAPHICS 2022, April 25 - 29, 2022, University of Reims Champagne-Ardenne, France