Vector Graphics Complexes

Boris Dalstein | **University of British Columbia, Canada**
Rémi Ronfard | **Laboratoire Jean Kuntzmann, University of Grenoble & Inria, France**
Michiel van de Panne | **University of British Columbia, Canada**

Graphisme, animation et nouveaux médias
grand
Graphics, Animation and New Media

Project Website **www.dalboris.com/research/vgc**
Contact **dalboris@cs.ubc.ca**

Abstract

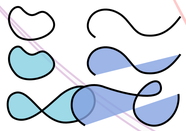
Basic topological modeling, such as the ability to have several faces share a common edge, has been largely absent from vector graphics.

We introduce the vector graphics complex (VGC) as a simple data structure to support fundamental topological modeling operations for vector graphics illustrations.

Problem

Most vector graphics tools use a data structure essentially following the *Scalable Vector Graphics* (***SVG***) specification:

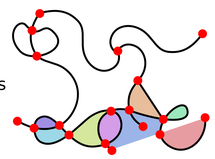
- Closed or open paths
- Filled or not
- Can overlap



Issues:
- Cannot represent multiway joins
- Cannot represent shared edges

A few other vector graphics tools use ***planar maps***:

- 2-complex embedded in $\mathbb{R}^2$
- Can represent multiway joins
- Can represent shared edges

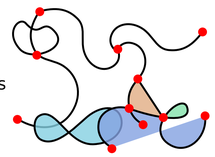


Issues:
- Cannot represent overlapping objects
- Editing geometry can invalidate topology

Our Solution

We introduce the ***vector graphics complex***:

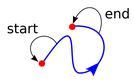
- 2-complex **immersed** in $\mathbb{R}^2$
- Can represent multiway joins
- Can represent shared edges
- Can represent overlapping object
- Editing geometry never invalidates topology



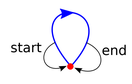
Idea:
- **Superset of SVG and planar maps**
- Overlap *when you want to*
- Intersect *when you want to*

Data Structure

```
class Cell { set<Cell*> star; };

class Vertex: public Cell { Point p; };

class Edge:   public Cell { Vertex * start;
                            Vertex * end;
                            Curve curve;    };

struct Halfedge { Edge * edge; bool b; };

struct Cycle { Vertex * steiner;
               vector<Halfedge> halfedges; };

class Face: public Cell { vector<Cycle> cycles; };
```

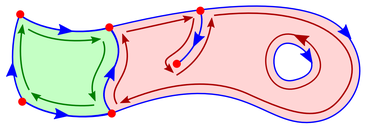


open edge



open edge (special case)

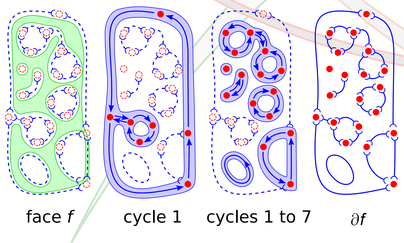
closed edge



Faces and Cycles

A *face* is defined by one or several *cycles*, where each cycle is one of:

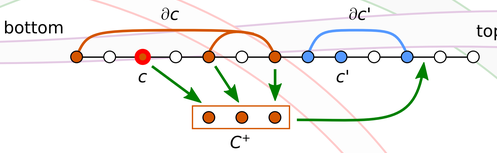
- **Non-simple cycle**
  Looping sequence of consecutive oriented open edges
- **Simple cycle**
  Oriented closed edge
- **Steiner cycle**
  Single vertex



Depth Ordering

All cells are globally ordered via a doubly-linked list, and are rendered back to front.

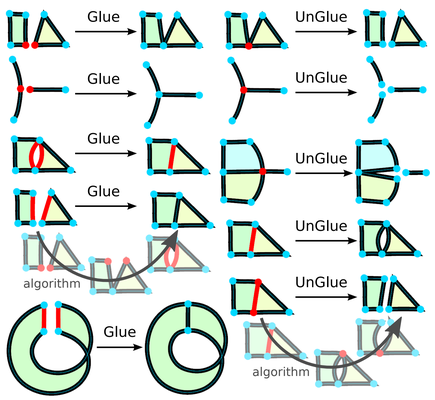
Users can use **raise** or **lower** to alter this ordering.

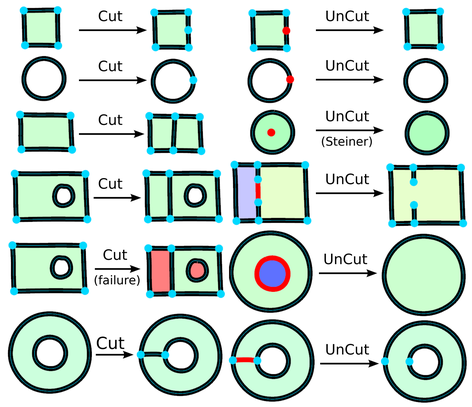


Raise(selected cell c):

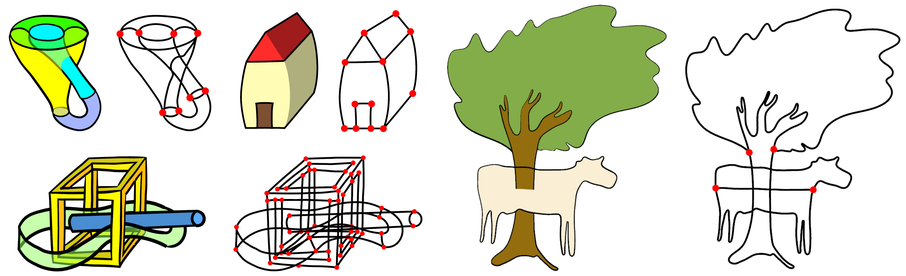
- Search from bottom to find c.
- Compute C^+ = subset of ($c \cup \partial c$) that is above c.
- Search up from c for the first cell c' satisfying:
  (c' not in ∂c) AND (c' overlaps with c)
- Move C^+ above the highest element of ($c' \cup \partial c'$).

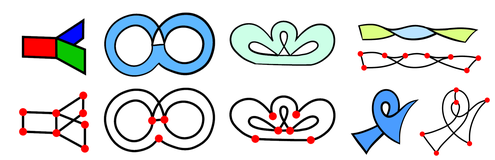
Topological Operators

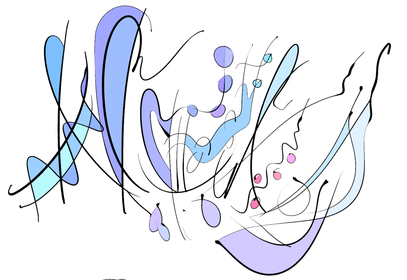


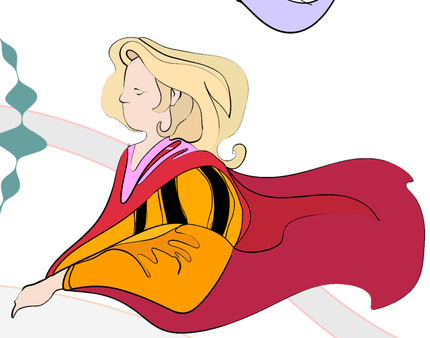


Results









Limitations

- Creating a face is tedious: the user must select all boundary edges.
- Multiway joins are not correctly rendered.

Future Work

- Extension to animation, including when changes in topology occur
- "Click-to-fill" tool

Related Work

- ASENTE, P., SCHUSTER, M., AND PETTIT, T. 2007. Dynamic Planar Map Illustration. ACM Transactions on Graphics (TOG) 26, 3.
- BAUDELAIRE, P., AND GANGNET, M. 1989. Planar Maps: an Interaction Paradigm for Graphic Design. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, New York, NY, USA, CHI '89, 313–318.
- DE FLORIANI, L., MORANDO, F., AND PUPPO, E. 2003. Representation of Non-manifold Objects Through Decomposition into Nearly Manifold Parts. In Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications, ACM, New York, NY, USA, SM '03, 304–309.
- McCANN, J., AND POLLARD, N. 2009. Local Layering. In ACM Transactions on Graphics (TOG), vol. 28, ACM, 84.
- ROSSIGNAC, J., AND O'CONNOR, M. 1989. SGC: A Dimension-independent Model for Pointsets with Internal Structures and Incomplete Boundaries. Research report. IBM T.J. Watson Research Center.
- WEILER, K. 1985. Edge-Based Data Structures for Solid Modeling in Curved-Surface Environments. IEEE Computer Graphics and Applications 5, 1, 21–40.

Acknowledgments We thank Estelle Charleroy, Etienne Colas, and other anonymous artists for their invaluable user feedback. Part of this work was supported by ERC Advanced Grant "Expressive", by GRAND, and by NSERC.