We introduce the vector graphics complex (VGC) as a superset of SVG and planar maps. Most vector graphics tools use a data structure essentially modeling operations for vector graphics illustrations. Simple data structure to support fundamental topological modeling operations for vector graphics illustrations. Basic topological modeling, such as the ability to have several faces share a common edge, has been largely absent from vector graphics.

### Our Solution

We introduce the **vector graphics complex**:  
- 2-complex immersed in \( \mathbb{R}^3 \)
- Can represent multiway joins
- Can represent shared edges
- Can represent overlapping objects
- Editing geometry can invalidate topology

### Related Work


### Problems

- Closed or open paths
- Filled or not
- Can overlap
- Cannot represent multiway joins
- Cannot represent shared edges

### 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

### Data Structure

```cpp
class Cell { set<Cell*> star; };
class Vertex { public Cell { Point p; };
class Edge: public Cell { Vertex * start; };
class Halfedge { Edge * edge; bool b; };
class Cycle { Vertex * steiner; vector<Halfedge> halfedges; };
class Face: public Cell { vector<Cycle> cycles; };
```

### 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

### Acknowledgments

We thank Estelle Charlier, 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.