Combinatorial Maps for Cell Complex Representation

David Gu

Yau Mathematics Science Center
Tsinghua University
Computer Science Department
Stony Brook University

gu@cs.stonybrook.edu

August 27, 2020

Combinatorial Maps

Meshes

Definition (Mesh)

A mesh is a cellular decomposition of geometric objects such as curves, surfaces or volumes.

Definition (Topological Models)

Topological models provide neighborhood relations between the cells of the decomposition (vertices, edges, faces, volumes).

The data structure provide ways to:

- traverse the cells
- traverse local neighborhoods
- store data with the cells
- modify the connectivity

Combinatorial Maps

Combinatorial maps are dimension-independent and rely on a single element along with a simple set of relations. All the information about the **cells** and their **incidence** and **adjacency** relations is contained within this model. All the neighborhood queries are resolved in optimal time (linear in the number of traversed cells) without having to maintain any additional information.

Incidence Graph

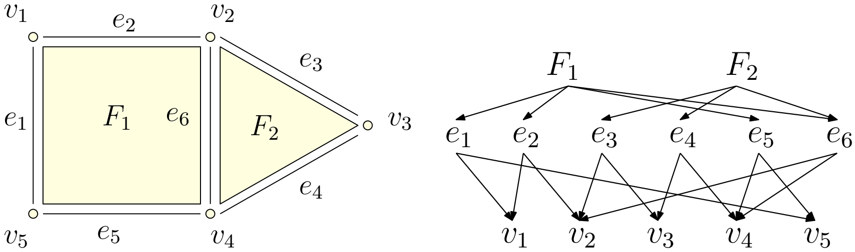


Figure: Cell decomposition and its incidence graph.

Cell-tuples

Definition (cell-tuple)

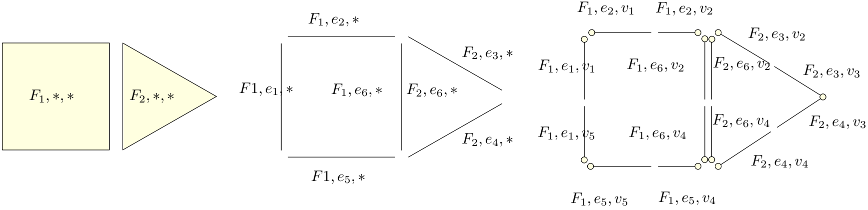
In a n-dimensional cellular decomposition, a cell-tuple is an ordered sequence of cells

$$(C_n, C_{n1}, \dots, C_1, C_0)$$

of decreasing dimensions such that $\forall i$, $0 < i \leq n$, C_i is incident to C_{i1}.

In other words, a cell-tuple corresponds to a path in the incidence graph from a n-cell to a 0-cell, i.e. a vertex.

Construction of Cell-tuples



Iterative construction of all the cell-tuples generated by the cellular decomposition, a cell-tuple is called a dart, $(face, edge, vertex)$.

i-adjacentcy

Definition (*i*-adjacency)

Adjacency relations are defined on the cell-tuples: two cell-tuples are said to be i-adjacent if their path in the incidence graph share all but the i-dimensional cell.

In the context of the cellular decomposition of a quasi-manifold, it can be shown that these $n+1$ adjacency relations put the cell-tuples in a one-to-one relation (except for the n-adjacency at the boundary of the object where cell-tuples do not have any mate).

Generalized Map

Generalized maps encode a cellular decomposition with a set D of darts (cell-tuples). A set of $n+1$ functions

$$\alpha_i : D \to D, \quad 0 \le i \le n$$

are defined based on the i-adjacency relations of the cell-tuples.
α_i functions are involutions, i.e. functions such that

$$\forall d \in D, \quad \alpha_i(\alpha_i(d)) = d.$$

α-functions

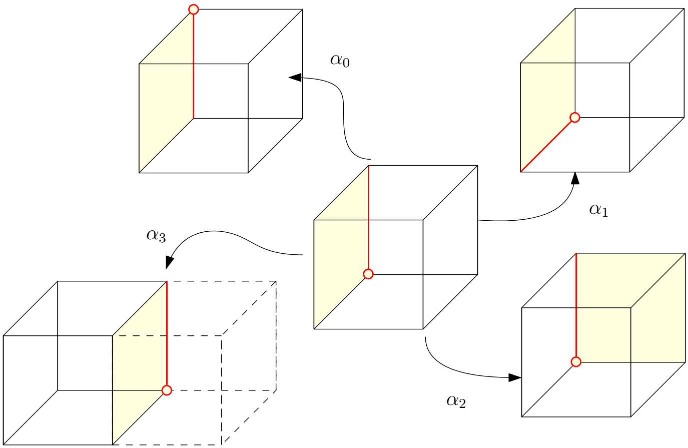


Figure: $\alpha_0, \alpha_1, \alpha_2, \alpha_3$ for a dart (V, F, E, V).

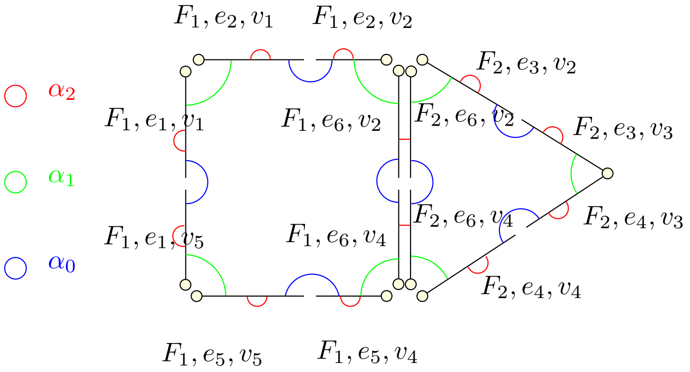


Figure: $\alpha_0, \alpha_1, \alpha_2$ for a dart (F, E, V).

α-functions

Consistency Condition

Combinatorial constraints express the correct assembly of cells along their boundary. For α_i functions, these constraints are expressed as follows:

$$\forall i, j, \quad 0 \leq i < i + 2 \leq j \leq n, \alpha_i \circ \alpha_j$$

is an involution.

Dart - Cell

1. each dart identifies a set of n cells of each dimension, i.e. those contained in the corresponding cell-tuple;
2. each k-cell is represented by a set of darts, i.e. all the darts whose corresponding cell-tuple contains this cell;

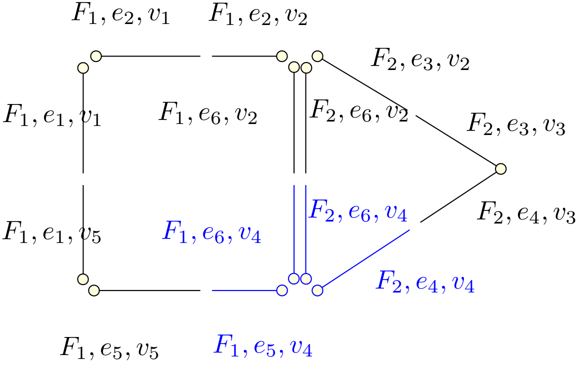


Figure: $\langle \alpha_1, \alpha_2 \rangle(d)$.

Orbit

1. $\alpha_i(d)$ is the dart that represents the same cells as d except from the i-dimensional cell;
2. All the other α_j, $j \neq i$ functions will lead to darts that share the same i-cell as d;
3. The set of darts representing the same i-cell can be obtained by applying successively all the functions that maintain the i-dimensional cell unchanged, i.e.

$$\{\alpha_j, j \in \{0, 1, \dots, i-1, i+1, \dots, n\}\}.$$

Such sets of darts are formally defined as orbits, noted:

$$\langle \alpha_0, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_n \rangle.$$

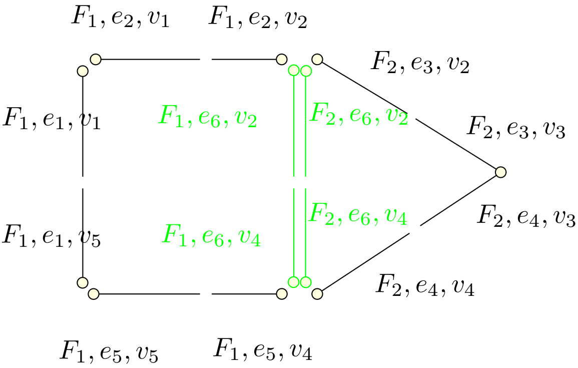


Figure: $\langle \alpha_0, \alpha_2 \rangle (d)$.

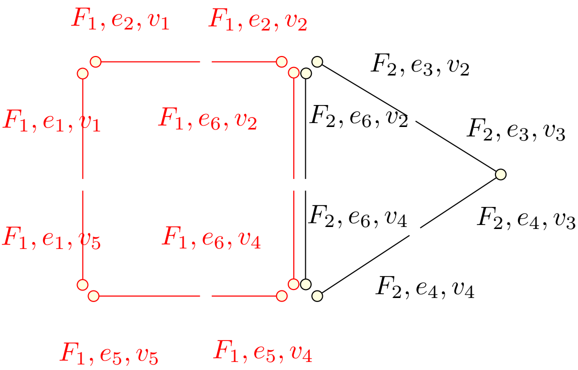


Figure: $\langle \alpha_0, \alpha_1 \rangle(d)$.

Oriented Combinatorial Maps

A Generalized map is able to represent orientable or non-orientable quasi-manifolds.

The orientability of a given G-map can be determined with a binary coloring process of its darts following this rule: a dart of a given color can only be linked to darts of the other color.

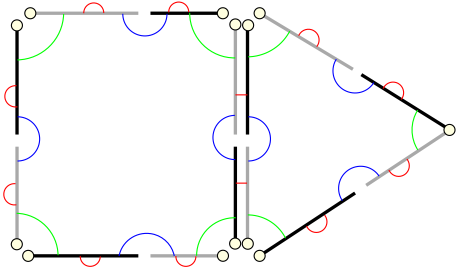


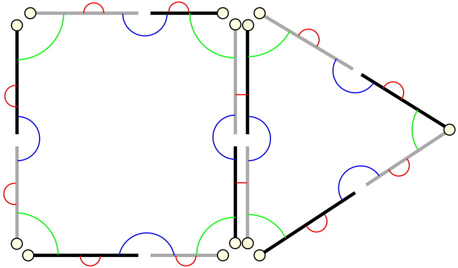
Figure: Orientable manifold.

Oriented Combinatorial Maps

For orientable manifold, the darts of the G-map are partitionned in two sets D-black and D-white of equal cardinality, each one representing one of the two orientations of the object. For any dart $d \in D$,

$$\langle \varphi_1, \ldots, \varphi_n \rangle (d)$$

with $\varphi_i = \alpha_i \circ \alpha_0$ is the set of darts corresponding to the orientation yielded by d.



Oriented Combinatorial Maps

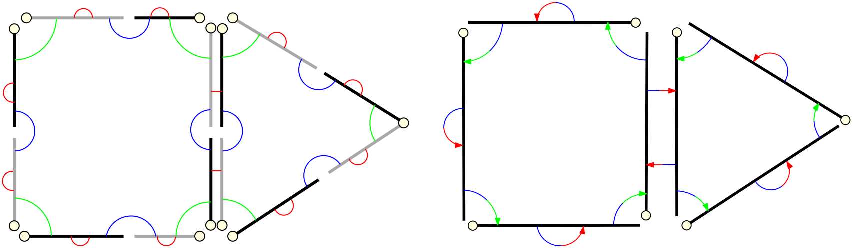


Figure: The oriented combinatorial map yielded by dart d, $\varphi_1 = \alpha_1 \circ \alpha_0$ and $\varphi_2 = \alpha_2 \circ \alpha_0$.

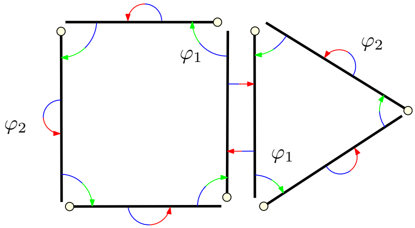
Oriented Combinatorial Maps

Definition (Oriented Combinatorial Maps)

One orientation of an orientable G-map is actually a combinatorial map, defined as a set of darts D along with n functions

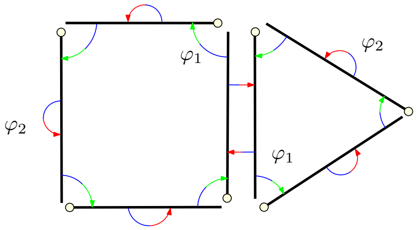
$$\varphi_i : D \to D, \quad 1 \leq i \leq D,$$

with $\varphi_i = \alpha_i \circ \alpha_0$.



Oriented Combinatorial Maps

- The φ_1 function is a permutation that links the ordered vertices around oriented faces;
- The φ_i, $i \leq 2 \leq n$ functions are involutions, as stated by the constraint expressed above on the α_i involutions;
- Each of these involutions allows to glue pairs of i-dimensional cells along their common $(i-1)$-dimensional boundary cell.



Oriented Combinatorial Maps

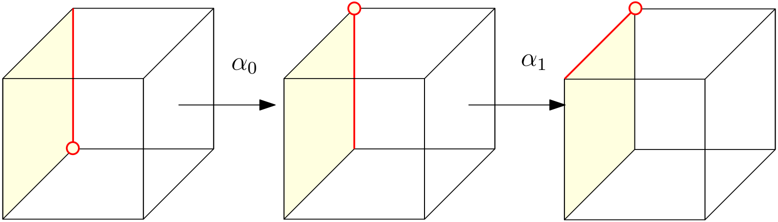


Figure: φ_1.

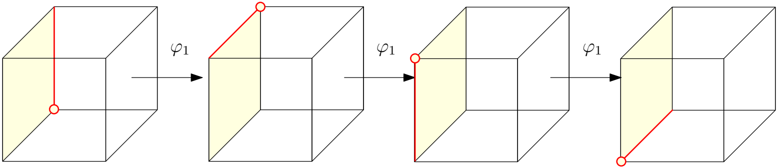


Figure: φ_1^n.

Oriented Combinatorial Maps

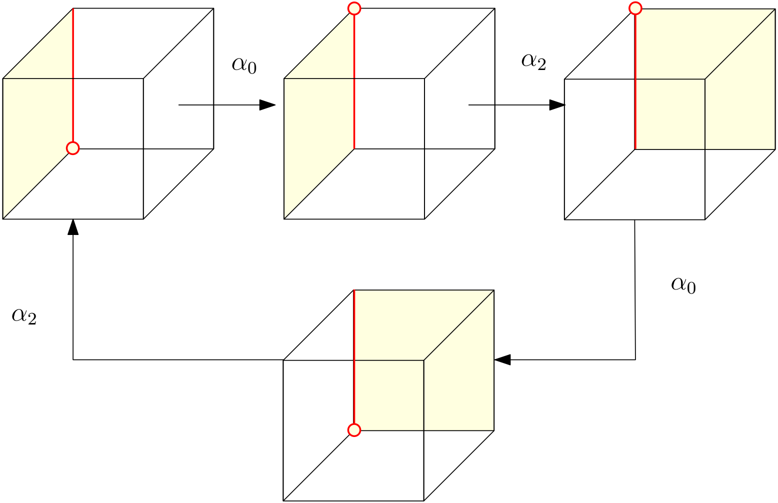


Figure: φ_2, $\varphi_2^2 = id$.

Oriented Combinatorial Maps

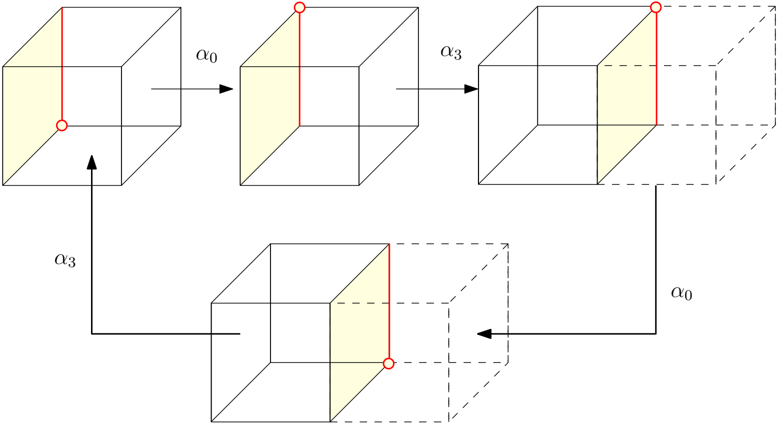


Figure: φ_3, $\varphi_3^2 = id$.

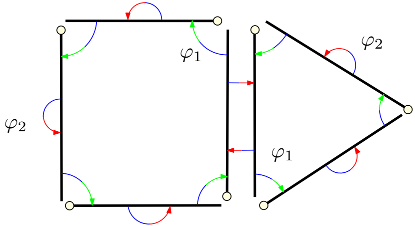
Orbits

1. For cells of dimension $i \geq 1$, the sets of darts that represent the cells are defined by the orbit

$$\langle \varphi_1, \dots, \varphi_{i-1}, \varphi_{i+1}, \dots, \varphi_n \rangle.$$

starting from any dart, all the functions that maintain the i-dimensional cell unchanged are applied.

2. For vertices, the orbit is $\langle \varphi_1 \circ \varphi_2, \dots, \varphi_1 \circ \varphi_n \rangle$.



Oriented Combinatorial Maps

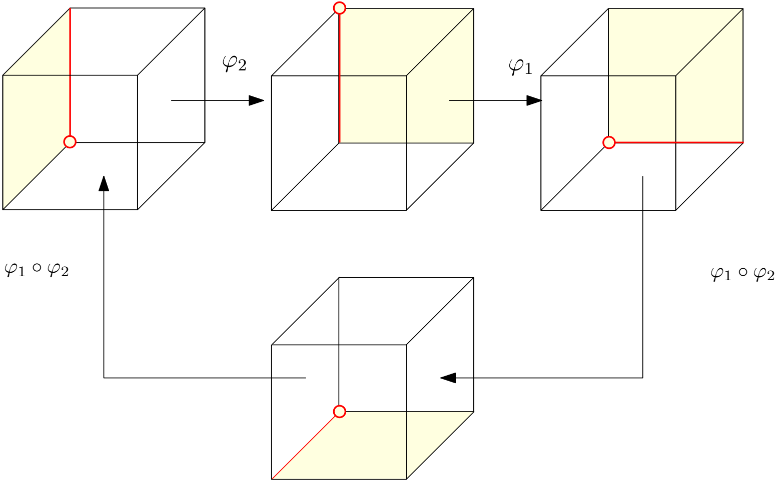


Figure: $\varphi_1 \circ \varphi_2$, $(\varphi_1 \circ \varphi_2)^2 = id$.

Oriented Combinatorial Maps

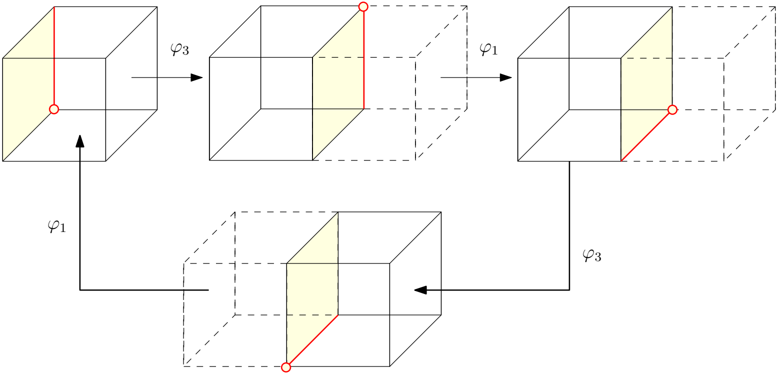


Figure: $\varphi_1 \circ \varphi_3$, $(\varphi_1 \circ \varphi_3)^2 = id$.

Oriented Combinatorial Maps

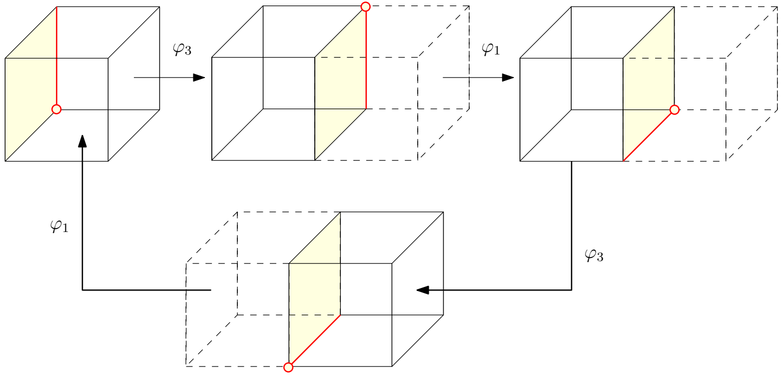


Figure: φ_2, φ_3, $(\varphi_2, \varphi_3)^2 = id$.

Volumetric Mesh

Mesh

A volumetric mesh data structure includes

1. a lits of darts;
2. a list of vertices;
3. a list of edges;
4. a list of faces;
5. a list of volumetric cells;

Dart

A dart $d = (v, e, f, c)$ includes

1. pointers to (vertex,edge,face,cell)
2. pointers to $\varphi_1(d), \varphi_2(d)$ and $\varphi_3(d)$

Volumetric Mesh

Vertex

A vertex v data structure includes

1. a pointer to one dart d, with the form $d = (v, e, f, c)$
2. attributes of the vertex
3. the neighboring darts $\langle \varphi_1 \circ \varphi_2, \varphi_1 \circ \varphi_3 \rangle (d)$

Edge

A edge e data structure includes

1. a pointer to one dart d, with the form $d = (v, e, f, c)$
2. attributes of the edge
3. the neighboring darts $\langle \varphi_2, \varphi_3 \rangle (d)$

Volumetric Mesh

Face

A face f data structure includes

1. a pointer to one dart f, with the form $d = (v, e, f, c)$
2. attributes of the face
3. the neighboring darts $\langle \varphi_1, \varphi_3 \rangle (d)$

Cell

A cell c data structure includes

1. a pointer to one dart d, with the form $d = (v, e, f, c)$
2. attributes of the cell
3. the neighboring darts $\langle \varphi_1, \varphi_2 \rangle (d)$