An Implementation of Halfedge Data Structure in Catmull-Clark Subdivision for 2-Manifold Single-sided Surface

Yu Wang

August 2015

1 Introduction

Catmull-Clark subdivision is widely applied to construct a smooth surface from an initial mesh of polygons. It is independent of the topology of initial mesh.

2 Halfedge Data Structure

An object in the 3D Euclid space can be modeled as several meshes of polygons. For a single mesh, it comprises three types of geometry elements: vertex, edge, and face. Adjacency data structure is need to store the topological information (adjacency and connectivity) between these elements.
Several adjacency structures have been fully developed, including simple data structure, winged edge data structure (Baumgart, 1975), halfedge data structure (Eastman, 1982), QuadEdge Data structure (Guibas and Stolfi), and FaceEdge Data Structure (Dobkin and Laszlo, 1987).
Among all these data structures, the author chooses halfedge data structure in this project to realize Catmull-Clark subdivision, because 1) the storage size is independent of the mesh topology, and 2) it has a simple implementation. The author also extends its definition to add the ability in dealing with single-sided surfaces (or non-orientable object).

2.1 Vertex, Halfedge, and Face

The definitions and assumptions of vertex, halfedge and face follow the assumption of 2-manifold, as shown in Table 1. The element ID is unique. When two elements fall into the same group of element, they can not have the same ID.
A quadrilateral face made with four halfedges and four vertices is shown in Figure 1.

Elements store two categories of information: self-information and adjacency information, where the adjacency information include links in a face and links between faces. As shown in Table 2.

	Definition	Assumption
Vertex	A 3-dimensional point.	No overlapping vertices exits in a mesh. But overlapping vertices can exist in different meshes.
Halfedge	An edge that starts from one vertex and ends at another vertex.	A halfedge connects exactly two non-overlapping vertices and it has a direction. Less than two halfedges start from the same vertex and end at the same vertex in a single mesh.
Face	A polygon that contains a loop of vertices and halfedges.	A face has at least three non-overlapping vertices so it makes a polygon. The face has to be constructed with a complete loop of halfedges with on openings.

Table 1: Definitions and assumptions of vertex, halfedge, and face

2.2 Face Connections

There are two types of face connections, the normal connection and the mobius connection, as shown in Figure 2. In a typical halfedge data structure, with the assumption of double-sided surface, the pair of halfedges between two faces are defined with opposite direction. The author extends this idea to single-sided surface and adds another type of connection, named as mobius connection. In a mobius connection, the pair of halfedges are in same direction. In the example of Figure 2, on the top, e_1 and e_1' are siblings to each other. On the bottom, e_1 and e_1' are mobius siblings to each other.

2.3 Traversals

Two basic traversals are necessary to implement Catmull-Clark subdivision: 1) traversal around a face, and 2) traversal around a vertex.

do
{

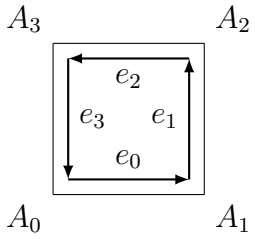

Figure 1: A quadrilateral face made with four halfedges

Element	Self-Information	Adjacency Information
Vertex	1. vertex ID 2. vertex position 3. vertex normal	1. one outgoing halfedge 2. on mobius connection?
Halfedge	1. edge ID	1. start and end vertex 2. link to parent face 3. predecessor and successor in the parent face 4. sibling links to adjacent face 5. boundary links to adjacent face
Face	1. face ID 2. face normal	1. one side halfedge

Table 2: Definitions and assumptions of vertex, halfedge, and face

}

2.3.1 Traversal Around Face

The traversal around a face lead to all the edges and vertices belong to this face. It starts from one side halfedge, follows the halfedge flow, and ends at the halfedge that the face traverse starts. Traversal of all faces in a mesh takes O(F) time, where F is the total number of faces in the mesh.

2.3.2 Traversal Around Vertex

The traversal around a vertex lead to all the edges and faces that contains this vertex. The traversal of a vertex need to consider two issues: 1) is this

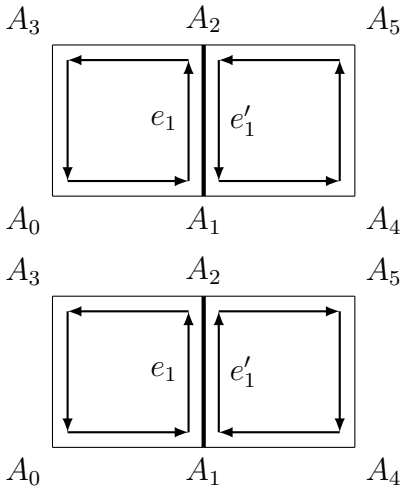

Figure 2: Normal connection (up) and mobius connection (bottom) between two faces

vertex on a boundary, and 2) is this vertex on a mobius connection. This makes four different types of vertex traversals. See Figure 3 - 6 for examples.

In the vertex traversal without boundary and mobius issue, we start from one outgoing halfedge of this vertex. We continue to go to the next outgoing halfedge by going to the successor of its sibling until we hit the first outgoing halfedge. In the example of Figure 3, if start and end at halfedge e_1, the sequence of traversal is: e_1 (sibling link to) e_1' (successor link to) e_2 (sibling link to) e_2' (successor link to) e_3 (sibling link to) e_3' (successor link to) e_4 (sibling link to) e_4' (successor link to) e_1.

In order to address the issue of a vertex on boundary, instead of using sibling links, we use boundary links. In the example of Figure 5, we have a boundary e_4' to e_2. If start and end at halfedge e_1, the sequence of traversal is: e_1 (sibling link to) e_1' (successor link to) e_2 (boundary link to) e_4' (successor link to) e_1.

To address the issue of vertex on a mobius connection, instead of using normal links, we use mobius links. At the same time, we switch between the successor and predecessor to the sibling every time we hit a mobius connection. In the example of Figure 3, e_1 to e_1' and e_3 to e_3' have mobius siblings rather than normal sibling. If start and end at halfedge e_1, the sequence of traversal is: e_1 (mobius sibling link to) e_1' (predecessor link to)

e_2 (sibling link to) e'_2 (predecessor link to) e_3 (mobius sibling link to) e'_3 (successor link to) e_4 (sibling link to) e'_4 (successor link to) e_1.

If boundary and mobius connection both occur, we use a combination of the two methods above. In the example of Figure 3, e_1 to e'_1 and e_3 to e'_3 have mobius siblings rather than normal sibling and we have a mobius boundary connection between e'_4 and e_2. If start end end at halfedge e_1, the sequence of traversal is: e_1 (mobius sibling link to) e'_1 (predecessor link to) e_2 (mobius boundary link to) e'_4 (successor link to) e_1.

We can combine the four cases together as shown in Algorithm 1.

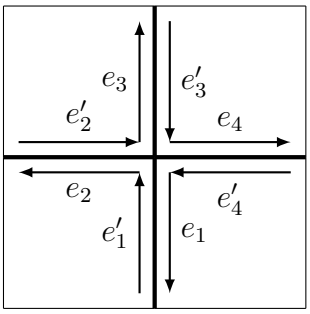

Figure 3: Vertex traversal without boundary and without moibus connection

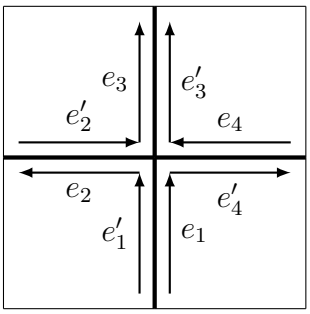

Figure 4: Vertex traversal without boundary and with moibus connection

2.4 Mesh Construction

The input of Catmull-Clark is a polygon mesh with arbitrary topology.

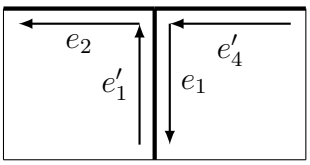

Figure 5: Vertex traversal withboundary and without moibus connection

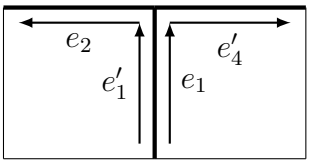

Figure 6: Vertex traversal with boundary and with moibus connection