Subdivision Tree Representation of Arbitrary Triangle Meshes

Wei Xu
wei@ticam.utexas.edu
TICAM
University of Texas at Austin
Austin, TX 78712

Don Fussell
fussell@cs.utexas.edu
Computer Science Department
University of Texas at Austin
Austin, TX 78712

January 15, 1998

Abstract

We investigate a new way to represent arbitrary triangle meshes. We prove that a large class of triangle meshes, called *normal triangle meshes*, can be represented by a subdivision tree, where each subdivision is one of four elementary subdivision types. We also show how to partition an arbitrary triangle mesh into a small set of normal meshes. The subdivision tree representation can be used to encode mesh connectivity information. Our theoretical analysis shows that such a coding scheme is very promising.

0.1 Introduction

Triangle meshes are popular in graphics and other application areas. One problem in dealing with meshes is to store topological information or connectivity of meshes. Recently, some methods have been presented to efficiently code meshes [5, 3, 4, 13, 6, 9]. A recent survey can be found in [12].

A triangle mesh has two different parts: topological information and geometric information. The topological information (also called connectivity) is about the relationship between vertices, edges and triangles in the mesh, while the geometrical information is about the position of vertices. Usually, the topological information is independent from geometric information, except some special cases such as regular grid, or Delaunay Triangulations, where geometric information is able to determine connectivity.

Most current methods try to generate long triangle strips [5, 3, 13, 1] to code mesh connectivity so that they can be used in some graphical library such as OpenGL. Some methods use vertex split/merge to code topological information implicitely [6, 9]. The hierarchical triangulation presented in [4] discussed the representations of meshes in tree structures. A recursive refinement of meshes naturally forms tree representations of triangle meshes. However, paper [4] uses links to code vertices, edges and triangles explicitely. It may result in large storage in storing these links (indices for vertices, edges and triangles).

It is true that a recursive refinement has a tree representation. But for a given mesh, whether we can generate its tree representation is not obvious. We have not seen any work on this *reverse problem*. The main purpose of this paper is to answer solve this problem. We are going to find the necessary and sufficient conditions for a triangle mesh to be represented by a tree.

This paper is organized as follows: In Section 2 gives basic terminologies. Our main results are presened in Section 3.

0.2 Notation and Definition

We first list some topological concepts used in our study. Some terms are defined in R^n in their general forms, although we are only interested in orientable bounded surfaces in R^3. For details about these terms and related information, readers may refer to topology textbooks, such as [8].

Definition 1 *The* Euclidean distance *between* $x = (x_1, ..., x_n)$ *and* $y = (y_1, ..., y_n)$ *in* R^n *is given by*

$$\|x - y\| = \sqrt{(x_1 - y_1)^2 + ... + (x_n - y_n)^2}$$

Definition 2 *A* topological space *is a set* X *with a collection* $\mathcal{B}$ *of subsets* $N \subseteq X$*, called* neighborhoods*, such that*

- *every point is in some neighborhood, i.e.,* $\forall x \in X, \exists N \in \mathcal{B}$ *such that* $x \in N$
- $N_1, N_2 \in \mathcal{B}$ *with* $x \in N_1 \cap N_2, \exists N_3 \in \mathcal{B}$ *such that* $x \in N_3 \subseteq N_1 \cap N_2$*.*

The set, $\mathcal{B}$*, of all neighborhoods is called a* basis *for the topology on* X*.*

Definition 3 *Two topological spaces* A *and* B *are* homeomorphic *if there is a continuous invertible function* $f : A \to B$ *with continuous inverse* $f^{-1} : B \to A$*. Such a function* f *is called a* homeomorphism*.*

Definition 4 *An* n-cell *is a set whose interior is homeomorphic to the n-dimensional disc* $D^n = \{x \in R^n : \|x\| < 1\}$ *with the additional property that its boundary must be divided into a finite number of lower-dimensional cells, called the faces of the n-cell.*

- *A 0-dimensional cell is a point.*
- *A 1-dimensional cell is a line segment.*
- *A 2-dimensional cell is a triangle.*
- *A 3-dimensional cell is a tetrahedron.*

Definition 5 *A* complex K *is a finite set of cells,* $K = \cup\{\sigma : \sigma \text{ is a cell}\}$ *such that:*

- *if* σ *is a cell in* K*, then all faces of* σ *are elements of* K*.*
- *if* σ *and* τ *are cells in* K*, then* $Int(\sigma) \cap Int(\tau) = \emptyset$*.*

where $Int(A)$ denotes the interior of a set A. The dimension of K is the dimension of its highest-dimensional cell.

Definition 6 *An n-dimensional* manifold *is a topological space such that every point has a neighborhood homeomorphic to an n-dimensional open disc*

$$D^n = \{x \in R^n : \|x\| < r\}$$

We further require that any two distinct points have disjoint neighborhoods. A 2-manifold is often called a surface.

Definition 7 *A n-dimensional* manifold with boundary *is a topological space such that every point has a neighborhood homeomorphic to either a 2-dimensional open disc or the half-disc $D^n_+ = \{x = (x_1, ..., x_n) \in R^n : \|x\| < r, x_n \geq 0\}$. Points with half-disc neighborhoods are called boundary points.*

Definition 8 *Let K be a complex. The set of all points in the cells of K is*

$$|K| = \{x : x \in \sigma \in K, \sigma \textit{ is a cell in } K\}$$

is the space underlying the complex K, or the realization *of K.*

Definition 9 *Let S_1 and S_2 be two surfaces. Remove a small disc from each of S_1, S_2 and glue the boundary circles of these discs together to form a new surface called the* connected sum *of S_1 and S_2, written as $S_1 \# S_2$.*

Definition 10 *A locally 2-dimensional topological space X is* triangulizable *if a 2-complex structure K can be found with $X = |K|$ and K has only triangle cells satisfying the additional condition that any two triangles are identified along a single edge or at a single vertex or are disjoint. A triangulated complex K is called a* simplicial complex *or a* triangulation *on X. A cell of a simplicial complex is called a* simplex.

Definition 11 *Let K be a complex. The* Euler characteristic *of K is*

$$\chi(K) = \#(\textit{0-cells}) - \#(\textit{1-cells}) + \#(\textit{2-cells}) - \#(\textit{3-cells})...$$

where $\#(S)$ denotes the number of elements in a finite set S.

Definition 12 *If S is a surface with boundary, the* associated surface *(without boundary) is S^*, where S^* is S with a disc sewn onto each of the boundary circles.*

In other words, S^* is the surface S with all its holes patched. If S has k holes or boundary components, then S^* can be considered as S with k discs or 2-cells added so $\chi(S^*) = \chi(S) + k$.

Definition 13 *The* genus g *of a closed surface of Euler characteristic χ is given by $(2-\chi)/2$ if χ is even, and $(1-\chi)/2$ if it is odd.*

The genus of a surface counts how many "handles" the surface has. For orientable surfaces, this is the number of "holes" in the surface.

Definition 14 *The genus of a compact surface with boundary S is $g(S) = g(S^*)$ where S^* is the associated surface without boundary.*

Theorem 1 *Every compact connected orientable surface is homeomorphic to a sphere or a connected sum of n tori.*

Intuitively, the orientable surfaces are the sphere and tori with any number of handles. Since a torus is a sphere with a handle, every orientable surface can be described as a sphere with some number of handles. Because an orientable surface with genus n is n tori glued together, and a torus is two closed discs glued together, an orientable surface can be generally decomposed into $2n$ closed discs. A genus 0 surface (sphere) is two closed discs glued together. Based on these facts and for the sake of simplicity, we are going to focus on 2D surfaces homeomorphic to a closed disc, i.e., surfaces in 2D with simple boundaries and no holes.

From our previous definition of triangulation, we have the following definition for 2D triangle meshes:

Definition 15 *A 2D* triangle mesh *is a collection of disjoint triangles in a 2D plane such that they cover a conencted region in the plane, and a triangle is not allowed to have a verex of another triangle in the interoir of one of its edge.*

In the following, we assume M is a 2D triangle mesh with a simple boundary and no holes. We denote ∂M as the boundary (polygon) of M. For a pair of adjacent vertices $v_1, v_2 \in M$, the edge connecting them is denoted by $\overline{v_1 v_2}$. A path in M consists of edges $\overline{v_1 v_2}, ..., \overline{v_{n-1} v_n}$ is denoted as $< v_1, .., v_n >$. For simplicity, a path connecting v_1, v_2 is also denoted as $v_1 v_2$ if there is no confusion as to which path we are talking about. The length of a path is defined as the number of vertices forming the path.

Definition 16 *: A* graph $G = (V, E)$ *consists of two sets: a finite set V of elements called* vertices *and a finite set E of vertex pairs called* edges.

Generally speaking, a 2D triangulation cannot be represented by a graph non-ambiguously. For instance, when a triangle mesh has a hole which is a single triangle, the mesh has the same graph as the mesh without the hole.

Definition 17 *: A* dual graph *of mesh M Dual(M) is a 2D graph obtained as follows. For each triangle of M we have a vertex in dual(M). If two triangles share a common edge e in M then the corresponding vertices in Dual(M) will be adjacent, i.e., connected by an edge $\tilde{e}$. We call $\tilde{e} \in$ Dual(M) the dual of $e \in M$ (Fig. 1-1).*

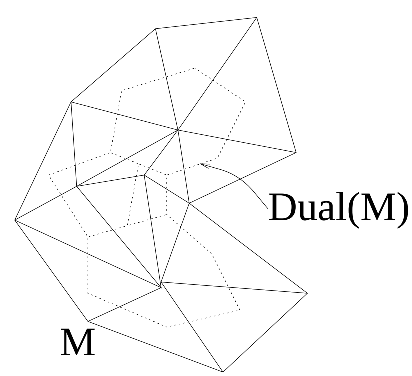

Fig. 1-1. M (in solid lines) and its dual Dual(M) (in dotted lines)

Definition 18 *: Two graphs G_1 and G_2 are said to be* isomorphic *if there exists a one-to-one correspondence between their vertex sets and a one-to-one correspondence between their edge sets such that the corresponding edges of G_1 and G_2 are incident on the corresponding vertices of G_1 and G_2.*

Definition 19 *: Two triangle meshes M_1 and M_2 are called* topologically equivalent *if their dual graphs are isomorphic.*

Definition 20 *All vertices/edges on ∂M are called* boundary vertices/edges*, the rest vertices/edges in M are called* interior vertices/edges*.*

It is easy to see that a boundary edge is contained by only one triangle in M, while an interior edge is shared by exactly two triangles in M. An edge is interior in M iff it has a dual edge in Dual(M).

Definition 21 *For a vertex v in a graph G, the number of edges incident to v is called the* valence (or degree) *of v, denoted by $\rho(v)$.*

Obviously, $\rho(v) \geq 3$ for an interior vertex v, and $\rho(v) \geq 2$ for a boundary vertex v.

Definition 22 *For a vertex $A \in \partial M$, A is called* subdivisible *if $\rho(A) > 2$.*

Definition 23 *For a point $A \in \partial M$, we define its* cone *$cone(A, M)$ as*

$$cone(A, M) = \{\triangle AV_1V_2 | \triangle AV_1V_2 \in M\}$$

i.e., $cone(A, M)$ is the collection of all triangles (in M) containing vertex A. The collection of all edges in $cone(A, M)$ excepting those containing vertex A is called the frontier *of $cone(A, M)$.*

The frontier of $cone(A, M)$ is a connected path. Fig. 1-2 shows a cone of vertex A and its frontier $< U1, U2, U3, U4, U5, U6 >$.

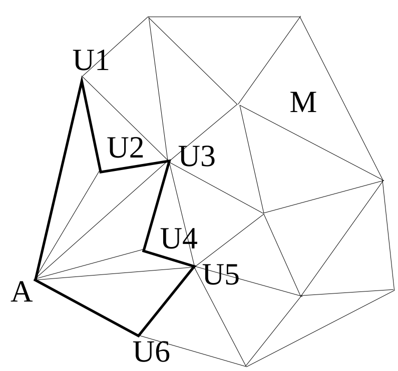

Fig. 1-2. A cone and its frontier

Definition 24 *A pair of vertices* v_1 *and* v_2 *in* M *are called* interior-connected *if there exists* $k(k \geq 0)$ *vertices* $w_1, .., w_k$ *such that* w_i*'s* $(i = 1, ...k)$ *are distinct interior vertices in* M*, and* $\overline{v_1 w_1}$*,* $\overline{w_i w_{i+1}}$ $(i = 1, ..., k-1)$*, and* $\overline{w_k v_2}$ *are interior edges in* M*. The path* $< v_1, w_1, .., w_k, v_2 >$ *is called an* interior-path *between* v_1 *and* v_2 *in* M*.*

Based on this definition, we immediately have the following property.

Property 1 *A path* $< v_1, w_1, .., w_k, v_2 >$ *is not an interior path iff there exists a vertex* $w_j(1 \leq j \leq k) \in \partial M$ *when* $k > 0$ *or edge* $\overline{v_1 v_2}$ *is on* ∂M *(when k=0).*

Because M is a single triangulated mesh without holes, all vertices are connected (not necessarily interior-connected). In addition, for any pair of vertices in M, there are at least two paths (possibly partially overlapping) in M connecting them due to the fact that any vertex has valence larger than one.

Definition 25 *For a pair of vertices* $v_1, v_2 \in M$*, an edge* e *in* M *is called a* blocking edge for v_1 and v_2 (w.r.t. M) *if the edge* e *cuts* M *into two disjoint sub-triangle meshes* $M_1, M_2 \subset M$ *such that* $v_1 \in M_1, v_1 \notin M_2, v_2 \in M_2, v_2 \notin M_1, M_1 \cap M_2 = e$*. A vertex* $v \in M$ *is said to* have no blocking edge *in* M *if there is no blocking edge for* v *and any other vertex* $w \in M$*; otherwise, we say* v *has a blocking edge in* M*.*

In the dual space Dual(M), removing the dual edge $\tilde{e}$ of the blocking edge e will result in two separated subgraphs of Dual(M).

Lemma 1 *An edge* e *is a blocking edge for some vertices* $v_1, v_2 \in M$ *iff* e *is an interior edge and both of its end points are on* ∂M*.*

Proof:

$\Rightarrow$: Assume e is a blocking edge for some vertices $v_1, v_2 \in M$.

If e is a boundary edge, cutting M along e will generate two meshes $M_1 = M$, and $M_2 = e$, thus, both $v_1, v_2 \in M_1$. This contradicts the definition of blocking edge of e. Thus, e must be an interior edge.

If at least one of the two end points of e is interior to M, cutting along e is not able to separate M into two disjoint parts. Thus, both of the two end points of e must be on ∂M.

$\Leftarrow$: If e is an interior edge and both of its endpoints u_1, u_2 are on ∂M, then e cuts M into two disjoint sub-meshes M_1, M_2; each of them has at least one triangle. Let $v_1 \notin e$ be a vertex in M_1, and $v_2 \notin e$ be a vertex in M_2, then e is a blocking edge for v_1, v_2. □

Lemma 2 *For any pair of non-adjacent vertices $v_1, v_2 \in M$, either there exists an interior-path between v_1 and v_2 or there is a blocking edge e for v_1 and v_2.*

Proof: If there is a blocking edge e for the vertices v_1 and v_2, then v_1 and v_2 are in disjoint submeshes M_1 and M_2. All paths connecting v_1 and v_2 must pass through one of the two end points of e, which are boundary points. Thus, there is no interior path for v_1, v_2.

If there is no blocking edge for non-adjacent vertices v_1 and v_2, there must be an interior path between v_1 and v_2. If not, any path connecting v_1 and v_2 has at least one vertex which is on ∂M. Assume $m > 0$ is the minimum number of boundary vertices on all such paths, and the path $\mathcal{P} =< v_1, w_1, ..., w_k, v_2 >$ has m vertices $w_{j_1}, ..., w_{j_m}$ where $(j_1 < j_2 < ... < j_m)$ are on ∂M. Without loss of generality, we assume the path $\mathcal{P}$ has the shortest length among such paths.

For convenience, we denote $w_0 = v_1, w_{k+1} = v_2$. Consider the angle (inside M) bounded by two edges $\overline{w_{j_1-1}w_{j_1}}, \overline{w_{j_1}w_{j_1+1}}$. Assume $\overline{w_{j_1}w_{j_1-1}}, \overline{w_{j_1}u_1}, ..., \overline{w_{j_1}u_r}, \overline{w_{j_1}w_{j_1+1}}$ are all edges inside the angle, where $r \geq 0$ (See Fig. 1-3 (a)).

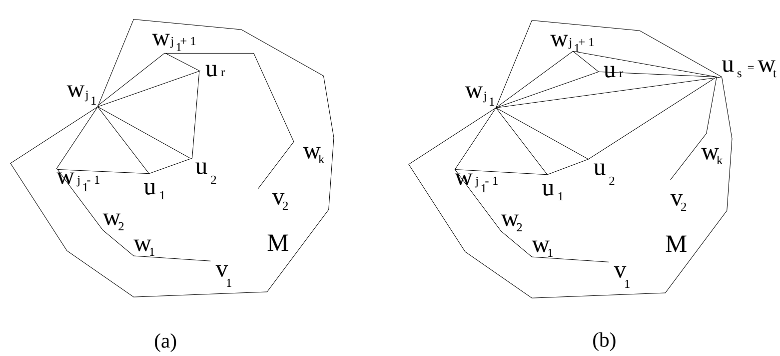

Fig. 1-3

The vertices $u_1, u_2, .., u_r$ must be interior, otherwise, say, u_s $(1 \leq s \leq r)$ is on the boundary. Then the edge $\overline{w_{j_1}u_s}$ cuts M into two separated parts M_1 and M_2.

If v_1 and v_2 are not in the same part, $\overline{w_{j_1}u_s}$ is a blocking edge for v_1, v_2. This is a contradiction.

If v_1, v_2 are in the same part (See Fig. 1-3 (b)), then the path $\mathcal{P}$ must pass through u_s and $w_t = u_s$ for some $t > j_1+1$. Therefore, we can shorten the path $\mathcal{P}$ by replacing subpath $< w_{j_1}, w_{j_1+1}, .., w_t >$ by a single edge $\overline{w_{j_1}w_t}$. This contradicts the shortest length assumption of $\mathcal{P}$.

Thus, all vertices $u_1, u_2, .., u_r$ must be interior in M. We construct a new path

$$\mathcal{P}' =< w_0, w_1, ..., w_{j_1-1}, u_1, u_2, ..., u_r, w_{j_1+1}, ..., w_{k+1} >,$$

which has only $m-1$ vertices $v_{j_2}, .., v_{j_m}$ on ∂M. This contradicts the assumption that m is minimized. Thus, v_1, v_2 are interior-connected. $\square$

Remark: If v_1 and v_2 are adjacent, it might be that neither a blocking edge nor an interior-path exists for them. An example is shown in Fig. 1-4.

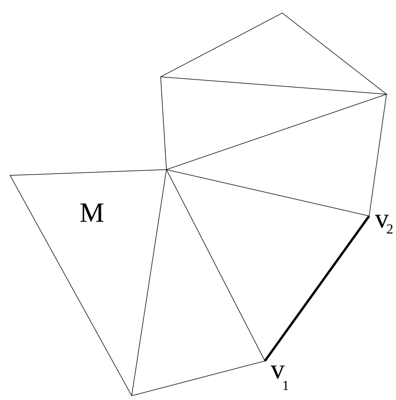

Fig. 1-4

Definition 26 *If A, B, C are three vertices on ∂M, we call M a* triangular patch, *denoted by $P_{\triangle ABC,M}$. The boundary part of ∂M between A and B is called the* patch edge *AB, denoted by $PE(A,B)$. Similarly, we have $PE(A,C), PE(B,C)$. The vertices on $PE(A,B), PE(A,C), PE(B,C)$ are called the boundary vertices of the patch. The vertices A, B, C are called the* corners *of the patch.*

Note: For simplicity, we still use M and $P_{\triangle ABC,M}$ interchangeably if no confusion is caused.

Definition 27 *A edge* $e = \overline{u_1u_2}$ *in a triangular patch* $P_{\triangle ABC,M}$ *is called* illegal *if it is interior and both of its end points are on same patch edge* $PE(A,B), PE(B,C)$, *or* $PE(A,C)$. *Otherwise, is called* legal.

Definition 28 *A triangular patch* $P_{\triangle ABC,M}$ *is* normal *if it contains no illegal edges. Otherwise, the patch is called* abnormal. *If there exists a normal triangular patch* $P_{\triangle ABC,M}$ *for a mesh* M, *we call* M normal, *otherwise,* abnormal.

Fig. 1-5 show two examples of abnormal triangular patches.

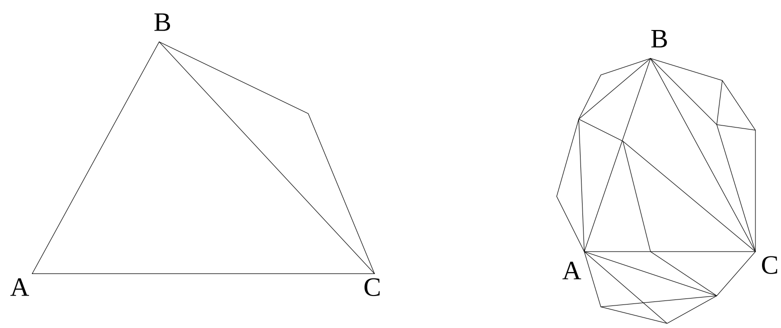

Fig. 1-5. Examples of abnormal triangular patches

It is easy to see that a sufficient (but not necessary) condition to be a abnormal triangular patch is that valence $\rho(v) = 2$ for some boundary vertex v which is not a corner.

As we mentioned earlier, a surface can be decomposed (cut along some paths on the surface) into a finite number of closed discs. For a 3D triangle mesh, we may choose the cutting paths to be paths of the mesh, i.e., we can cut a 3D triangle mesh into a number of sub-meshes, each of them topologically equivalent to a 2D disc. We should point out that we can select our cutting paths so that no illegal edges will be created by our cutting.

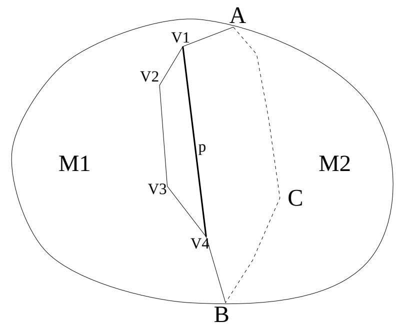

Fig. 1-6. Removing illegal edges by adjust cutting pathes

Fig. 1-6 shows the strategy for avoiding generating illegal edges. In Fig. 1-6, assume we cut a sphere-like triangle mesh into two submeshes M_1 and M_2. If there is an illegal edge $p = \overline{V_1V_4}$ (shown as a wide-solid line) on the cutting path (shown as solid lines on the front and dotted lines on the back), we can change our cutting path by replacing sub-path $< V_1, V_2, V_3, V_4 >$ by the edge p. The edge p is no longer illegal. Based on this strategy, when we cut a 3D triangle mesh without boundary into several sub-meshes, we can assume that all sub-meshes are topologically equivalent to normal triangular patches.

When a surface has boundaries, we cannot remove illegal edges simply by adjusting cutting edges if illegal edges exist on the existing boundaries. In this case, we may use a construction based on adding a vertex to get rid off all illegal edges immediately as shown in the following example.

Let M be an abnormal 2D triangular patch with the boundary $\partial M = \{V_0, V_1, ..., V_{11}\}$ and three corners $A = V_0, B = V_4, C = V_8$ (Fig. 1-7(a)). The patch $P_{\triangle ABC,M}$ has two kinds of illegal edges: those caused by degree-2-vertices – edges $\overline{V_0V_2}$,$\overline{V_2V_4}$, $\overline{V_4V_6}$, $\overline{V_6V_8}$, $\overline{V_8V_{10}}$, $\overline{V_{10}V_0}$, and those not caused by degree-2-vertices – edge $\overline{V_8V_{10}}$. We add a vertex W outside the plane of support of M and construct triangles $\triangle WV_iV_{i+1}$ for all successive boundary points V_i, V_{i+1} (shown as dotted lines in Fig. 1-7(a)). This results in a 3D triangle mesh denoted by $\tilde{M}$ which is topologically equivalent to a sphere. We consider ∂M as the path cutting $\tilde{M}$ into two sub-meshes, one is the mesh M, the other is formed by W with all vertices on ∂M. Now, we adjust the cutting path as mentioned above to remove illegal edges. In the example here, our new cutting path is shown in bold lines (Fig. 1-7(a)). Under the new cutting, the mesh $\tilde{M}$ is decomposed into two submeshes M_1 and M_2, as shown in Fig 1-7(b) and

(c) respectively. Both of them are normal patches.

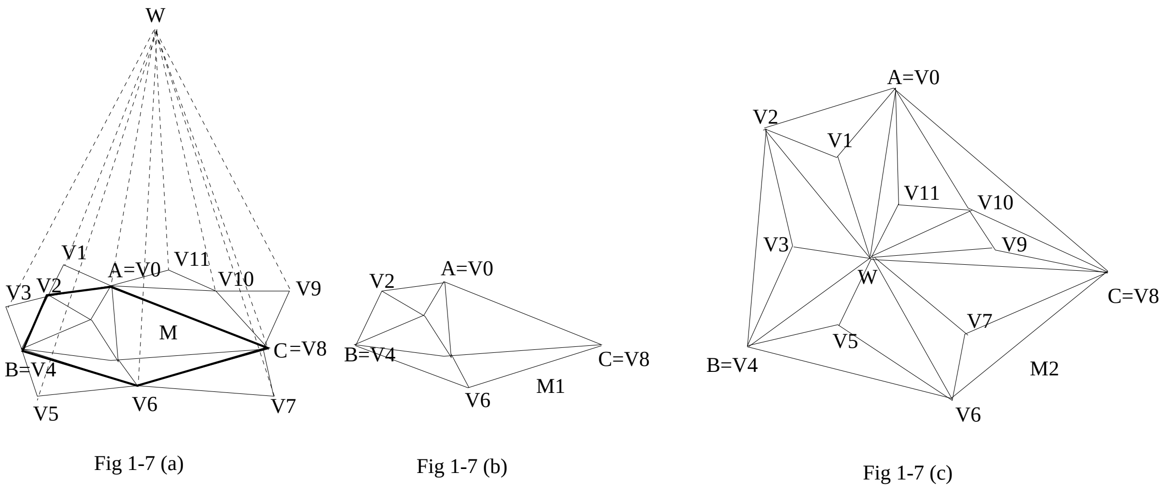

Fig. 1-7. Removing illegal edges

Note: The advantage of this technique is that we only need to code two sub-patches if there are many illegal edges. The major drawback is that it may create many new triangles in M_2, although the topological information is very cheap to store (we will address this problem later). An alternative way to normalize triangular patches is to cut a patch into many sub-patches so that each sub-patch is normal, but it may reduce the efficiency of coding.

0.3 General subdivision theorem

Theorem 2 *Let $P_{\triangle ABC,M}$ be a normal triangular patch obtained from triangle mesh M. We can reconstruct another triangle mesh $\bar{M}$ by adaptively subdividing a single triangle $\triangle\bar{A}\bar{B}\bar{C}$ such that each subdivision is one of the following four elementary types (called binary, ternary, quaternary, and mitsubishi, respectively) as shown in Fig 2-1(a)(b)(c)(d), $\bar{M}$ is topologically equivalent to M, and the vertices $\bar{A}, \bar{B}, \bar{C}$ correspond to the vertices A, B, C.*

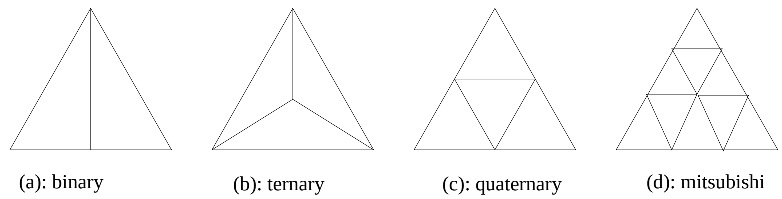

Fig. 2-1. Four different ways to subdivide a triangle

Proof: We have two steps in our proof.

In step 1, we prove that $P_{\triangle ABC,M}$ can be subdivided into several sub-triangular patches using one of the four elementary subdivisions. In the second step, we prove the sub-triangular patches generated in the first step can be normalized. Therefore, we can repeat the subdivision on the sub-patches until all sub-patches have only three vertices, i.e., a single triangle.

0.3.1 step 1

There are two cases.

Case 1: At least one vertex of A, B, C has no blocking edge in M

Assume the vertex A has no blocking edges, in other words, for any point $v \in M$, there exists an interior path from A to v.

- Case 1.1: If there is a point v on the boundary $PE(B, C)$, then we can subdivide M along the interior path from A to v using 'binary subdivision' (Fig. 2-2 (a)) and finish our proof.

- Case 1.2: Suppose the patch edge $PE(B, C)$ is a single edge $\overline{BC}$. Let $\triangle DBC$ be the triangle in M containing the edge $\overline{BC}$. The vertex D must be an interior point of M. Otherwise, either $\overline{BD}$ or $\overline{CD}$ will be a blocking edge for the vertex A, and thus it contradicts our assumption.

  Because D and A are inner-connected by a path P, we can subdivide M into three sub-triangular patches by cutting edge $\overline{BD}, \overline{CD}$ and the path P (Fig. 2-2 (b)) using the ternary subdivision.

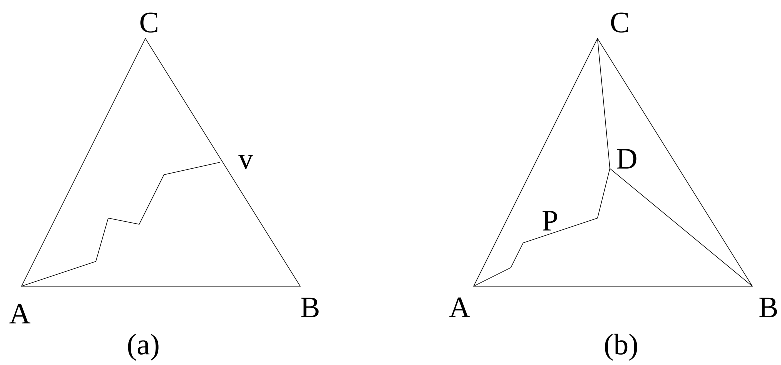

Fig. 2-2.

Case 2: All three vertices A,B,C have blocking edges in M.

Let edge A_1A_2 be a blocking edge for vertex A. We can assume that vertices A_1 and A_2 are on $PE(A,B)$ and $PE(A,C)$,respectively. If this condition is not satisfied, then one of the two vertices A_1, A_2, say, A_1, must be on the $PE(B,C)$. That leads to $A_1 = B$ and then B, A_2 are interior-connected.

We run over all such blocking edges, and assume the blocking edge A_1A_2 is selected such that the area AA_1A_2 is maximized.

Similarly, we have blocking edges $\overline{B_1B_2}$ and $\overline{C_1C_2}$ for vertices B and C, which maximize area BB_1B_2 and CC_1C_2, respectively (Fig 2-3).

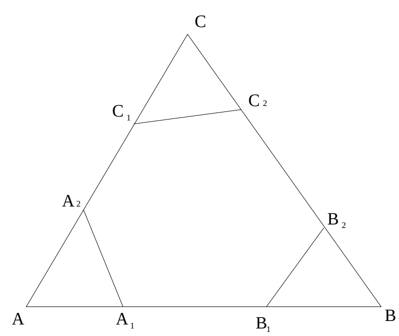

Fig. 2-3.

We call this assumption the *maximum area assumption* in the following.

We denote by M' the region bounded by $A_1B_1B_2C_2C_1A_2$.

Based on the maximum area assumption, we immediately get the following result: there are no blocking edges inside the region M', and any interior point $v \in M'$ is interior-connected to all vertices of $A_1, A_2, C_2, C_1, B_2, B_1$.

If one of the following happens: $A_1 = B_1$ or $A_2 = C_1$ or $B_2 = C_2$, we say there is a degeneracy. We can deal with the degeneracy as follows. If, say, $A_1 = B_1$ (Fig. 2-4), then Lemma 2 implies that A_2 and B_2 must be interior-connected by a path $\mathcal{P}$ in the region bounded by $A_1A_2CB_2$. Otherwise, there will be a blocking edge A_1Q such that $Q \in A_2C$ or $Q \in B_2C$ (shown by a dotted line in Fig. 2-4), which contradicts the maximum area assumption. Thus, we can subdivide M along path P and two blocking edges $\overline{A_1A_2}$ and $\overline{A_1B_2}$ using quaternary subdivision.

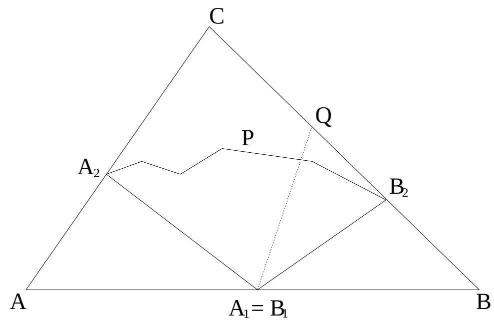

Fig. 2-4.

From now on, we assume there is no degeneracy, i.e., $A_1 \neq B_1$, $A_2 \neq C_1$ and $B_2 \neq C_2$.

Consider the triangles $\triangle A_1A_2A_3$, $\triangle B_1B_2B_3$ and $\triangle C_1C_2C_3$ in the region M'. These triangles exist because of the maximum area assumption and the triangulation of M'. In addition, the maximum area assumption guarantees that all three points A_3, B_3, C_3 are interior points in M'. Based on Lemma 2, it follows that there exist interior-paths $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$ inside M' connecting A_3 and B_3, A_3 and C_3, B_3 and C_3, respectively (Fig 2-5).

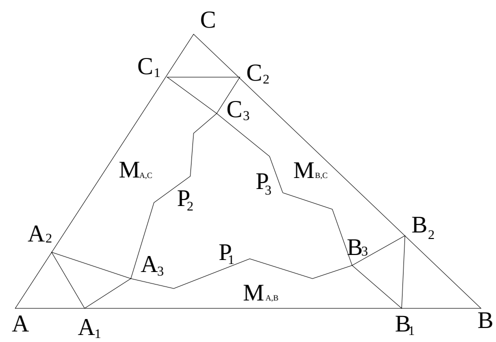

Fig. 2-5

We let $M_{A,B}$ denote the region bounded by $\overline{A_1A_3}, \mathcal{P}_1, \overline{B_3B_1}$ and A_1B_1; $M_{A,C}$ bounded by $\overline{A_2A_3}, \mathcal{P}_2, \overline{C_3C_1}$ and A_2C_1; $M_{B,C}$ bounded by $\overline{C_2C_3}, \mathcal{P}_3, \overline{B_3B_2}$ and B_2C_2 (Fig 2-5).

We assume paths $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$ do not cross, although in general they may overlap. (If the paths cross, we can always re-organize the paths to remove the crossing.) We denote the three possible overlapping sub-paths by A_3A_4, B_3B_4 and C_3C_4 (Fig 2-6). We assume the lengths of these overlapping sub-pathes are minimized.

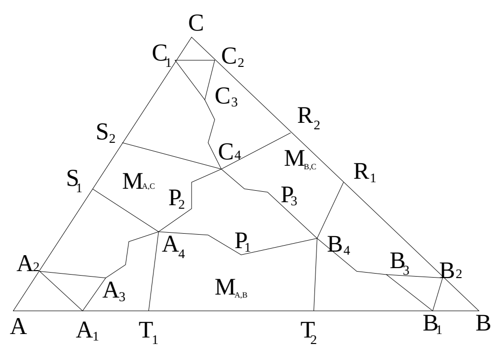

Fig. 2-6

Claim: There exist six edges (Fig. 2-6) $\overline{A_4S_1}$, $\overline{C_4S_2}$, $\overline{B_4R_1}$, $\overline{C_4R_2}$, $\overline{A_4T_1}$, $\overline{B_4T_2}$ such that

- $S_1, S_2 \in A_2C_1$, $T_1, T_2 \in A_1B_1$, $R_1, R_2 \in B_2C_2$,
- $\overline{A_4S_1}, \overline{C_4S_2} \in M_{A,C}$,
- $\overline{B_4R_1}, \overline{C_4R_2} \in M_{B,C}$,
- $\overline{A_4T_1}, \overline{B_4T_2} \in M_{A,B}$.

We now prove the existence of the edge $\overline{A_4S_1}$. If $A_3 = A_4$, we immediately get $S_1 = A_2$ and finish the proof. If $A_3 \neq A_4$, the angle A_4 in the region $M_{A,C}$ must be subdivisible. Otherwise, we can reduce the length of path A_3A_4, which contradicts our shortest length assumption on A_3A_4 above. Now build the cone of A_4 inside the region $M_{A,C}$. The frontier of the cone must have a vertex S_1 on the boundary A_2C_1, otherwise, the frontier can be used to reduce the length of A_3A_4.

Similarly, we can prove the existence of the remaining five edges.

It is easy to see that the regions $M_{A,B}, M_{A,C}, M_{B,C}$ are disjoint, but share parts of their boundaries.

Based on Fig.2-6, we will find subdivision paths in M. The problem can be solved when we classify M into the following three cases.

- Case 2.1: $A_4 \neq B_4$, $A_4 \neq C_4$, $C_4 \neq B_4$. (Fig. 2-7 (a)).

  We claim that there exist six paths $U_1U_2, U_1U_3, V_1V_2, V_1V_3, W_1W_2, W_1W_3$ such that

  – vertex $U_1 \in S_1S_2$, vertices $U_2, U_3 \in A_4C_4$, $U_2 \neq U_3$,

  – vertex $V_1 \in T_1T_2$, vertices $V_2, V_3 \in A_4B_4$, $V_2 \neq V_3$,

  – vertex $W_1 \in R_1R_2$, vertices $W_2, W_3 \in B_4C_4$, $W_2 \neq W_3$,

  – paths U_1U_2, U_1U_3 are interior paths in region $A_4C_4S_2S_1$,

  – paths V_1V_2, V_1V_3 are interior paths in region $A_4T_1T_2B_4$,

  – paths W_1W_2, W_1W_3 are interior paths in region $B_4R_1R_2C_4$,

  – all six paths are interior to M.

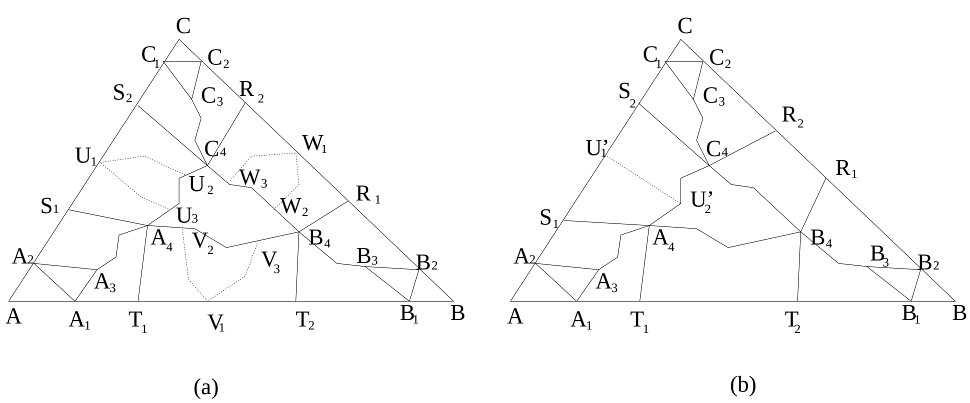

Fig. 2-7

It suffices to prove the existence of the paths U_1U_2 and U_1U_3. We can use the following recursive algorithm to find them.

`Algorithm for searching pathes` U_1U_2, U_1U_3 `in the region` $A_4C_4S_2S_1$:

```
init:
    U1 ← S2, U2 ← C4, U3 ← A4,
    path U1U2 = S2C4 (overlined),
    done = FALSE
while done = FALSE  do {
    If there is an interior path P  for U1, U3 in A4C4S2S1
        path U1U3 ← P
        done = TRUE
    else
        if there is a blocking edge U1'U2' in the region A4C4S2S1
        such that U1' ∈ S1S2 and U2' ∈ C4A4 (Fig 2-7 (b))
            U1 ← U1',  U2 ← U2'.
        else /* U1 = S1  */
            U2 ← A3 ,  path U1U3 = S1A4 (overlined),  done = TRUE
}
```

The algorithm will terminate because there are only a limited number of vertices on S_1S_2.

Now, we are able to subdivide M along three pathes: $U_1U_2, U_2C_4, C_4W_3, W_3W_1, W_1W_2, W_2B_4, B_4V_3, V_3V_1, V_1V_2, V_2A_4, A_4U_3, U_3U_1$ using the quaternary subdivision (Fig. 2-7 (a)).

- Case 2.2: $A_4 = B_4 = C_4$.

  We find an edge $\overline{U_{11}U_{12}}$ and a path A_4U_{11} such that $U_{11} \in A_1T_1, U_{12} \in A_3A_4$, and $\overline{U_{11}U_{12}}, A_4U_{11}$ are interior to M as follows.

  - If $A_1 = T_1$, we simply select $U_{11} = A_1, U_{12} = A_3$, path $A_4U_{11} =$ path A_4T_1, and $\overline{U_{11}U_{12}} = \overline{A_1A_3}$. Obviously, these two paths are interior to M.

– If $A_1 \neq T_1$ and there is an interior path $\mathcal{P}$ inside the region $A_1T_1A_4A_3$ (Fig. 2-8(a)), we select $U_{11} = A_1, U_{12} = A_3, \overline{U_{11}U_{12}} = \overline{A_1A_3}$, and path A_4U_{11} = path $\mathcal{P}$.

– Suppose $A_1 \neq T_1$ and A_1, A_4 are not interior-connected in the region $A_1T_1A_4A_3$. Based on Lemma 2, there is a blocking edge $\overline{u_1u_2}$ for A_1, A_4. Because $\overline{A_1A_3}, \overline{A_4T_1}$ are edges, the vertices u_1 and u_2 must be on A_1T_1 and A_3A_4, respectively (Fig. 2-8 (b)). There might be more than one such blocking edge. We assume $\overline{u_1u_2}$ is the one closest to A_4, i.e., there is a path $\mathcal{P}$ connecting A_4 and u_1, and $\mathcal{P}$ is an interior path in M. Now we assign $U_{11} = u_1, U_{12} = u_2, \overline{U_{11}U_{12}} = \overline{u_1u_2}$, and path A_4U_{11} = path $\mathcal{P}$.

Similarly, we can get an edge $\overline{U_{21}U_{22}}$ and a path A_4U_{21} such that $U_{21} \in A_2S_1, U_{22} \in A_3A_4$, and $\overline{U_{21}U_{22}}, A_4U_{21}$ are interior to M (Fig. 2-8 (c)). Together, we have constructed a sub-triangular patch $A_4U_1U_2$ (we rename $U_1 = U_{11}, U_2 = U_{21}$, and the triangular patch is shown by the shadowed area in Fig. 2-8 (c)) inside the region $AT_1A_4S_1$. The patch $A_4U_1U_2$ intersects ∂M only at U_1, U_2, and its three boundary edges are all interior to M.

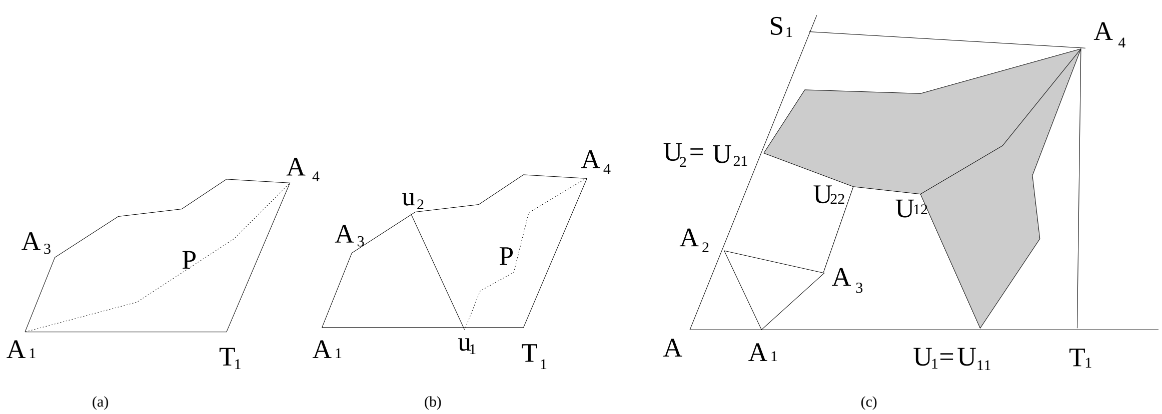

Fig. 2-8 Mesh near vertex A

In the same way, we can obtain two other sub-triangular patches $A_4V_1V_2$ inside $BR_1A_4T_2$, and $A_4W_1W_2$ inside $CS_2A_4R_2$ (Fig. 2-9 (a)). All three sub-triangular patches $A_4U_1U_2$, $A_4V_1V_2$, $A_4W_1W_2$ are disjoint, because they are separated by six edges $\overline{A_4R_1}, \overline{A_4R_2}, \overline{A_4S_1}, \overline{A_4S_2}, \overline{A_4T_1}, \overline{A_4T_2}$.

If $U_1 \neq V_1, V_2 \neq W_2, W_1 \neq U_2$ (Fig. 2-9 (a)), we can subdivide M along the boundaries of these three sub-triangular patches using mitsubishi subdivision.

If one of the following happens: $U_1 = V_1, V_2 = W_2, W_1 = U_2$, and, say, $U_1 = V_1$ (Fig. 2-9 (b)), we can subdivide M along paths U_1U_2, V_1V_2, U_2A_4 and V_2A_4 using quaternary subdivision.

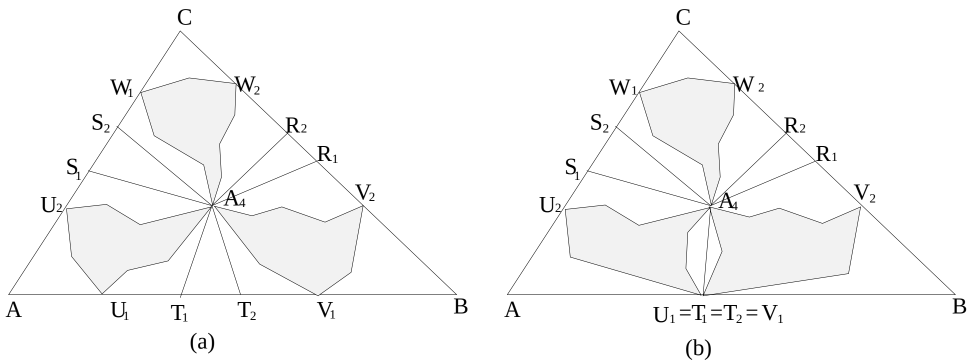

Fig. 2-9

- Case 2.3: Exactly one of the following is true: $A_4 = B_4$, $A_4 = C_4$, $C_4 = B_4$.

  Assume $A_4 = C_4$, $A_4 \neq B_4$ (Fig. 2-10 (a)). Using the exact same methods as in Case 2.1 and Case 2.2, we can get sub-triangular patches $A_4U_1U_2$, $A_4X_1X_2$, and paths V_1V_2, V_1V_3, W_1W_2, W_1W_3 (Fig. 2-10 (b). We obtain a third sub-triangular patch bounded by the following eight paths: A_4V_3, V_3V_1, V_1V_2, V_2B_4, B_4W_3, W_3W_1, W_1W_2, W_2A_4. Therefore, we get a result similar to Case 2.2. The rest of the argument is exactly same as in Case 2.2.

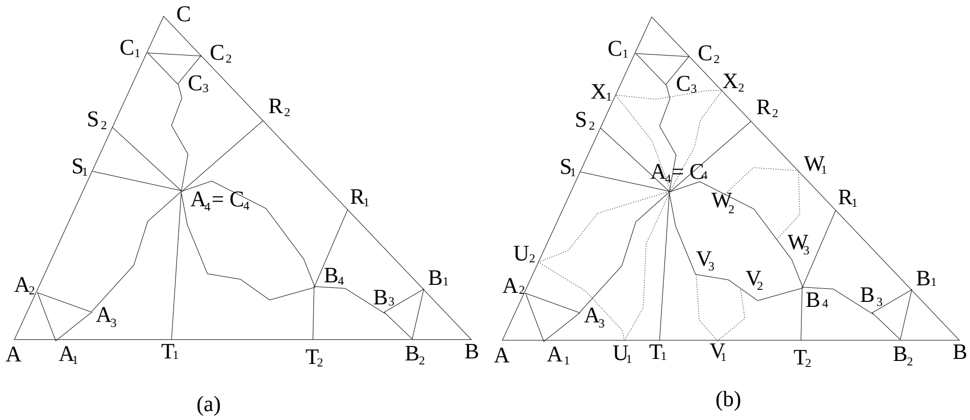

Fig. 2-10

0.3.2 step 2

In step 1, we proved that any normal triangular patch M can be subdivided into several sub-triangular patches. But these sub-triangular patches may be abnormal, so we cannot apply our general subdivision theorem on the sub-patches. Fortunately, we can normalize abnormal sub-patches in the way we described earlier.

Consider the result of using binary subdivision (Fig. 2-11). Assume we subdivide M along the path CD (in solid lines). There are three new boundaries AD, DB, CD in the two sub-triangular patches ACD and BCD. Obviously, paths AD and BD do not have any blocking edges because M is normal. If path CD has a blocking edge, say $\overline{UV}$ (shown in a dotted line), we can subdivide M using a new path formed by path CU, edge $\overline{UV}$ and path VD. Thus, the blocking edge is removed in the new subdivision.

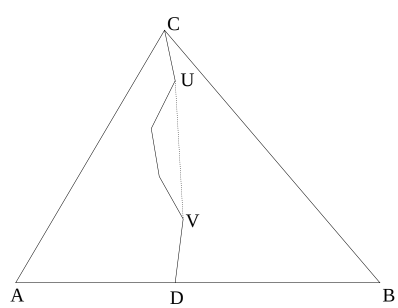

Fig. 2-11

It is easy to see that this technique applies to all abnormal sub-triangular patches generated by the other three subdivisions. □

Remark: From our arguments before, we know that the 'normality' of a triangle mesh is a weak condition: all closed 3D triangle meshes can be subdivided into normal triangular patches, and an abnormal triangular patch with an arbitrary number of illegal edges can always be decomposed into two normal triangular patches if we add one point.

In CAGD and computer graphics, it is common that a single triangular surface defined on a triangle domain, say, a triangular Bezier surface or B-spline surface, is subdivided into many triangular sub-patches by a 'divide and conquer' technique. No illegal edges are created in the subdivision procedure. Therefore

Corollary 3 *If triangle mesh M is generated by arbitrarily subdividing a singular triangular surface defined on a triangle domain, then M can be represented as a subdivision tree where each subdivision is one of the four elementary types defined above.*

Corollary 4 *A necessary and sufficient condition for a triangle mesh to be represented by a single subdivision tree is that the mesh be normal.*

0.4 Efficiency analysis

0.4.1 About entropy coding

Before we analyze the efficiency of our subdivision tree representation for topological coding, we need to briefly introduce some basic concepts from information theory.

Information theory uses *entropy* as a measure of how much information is contained in a message [10]. The higher the entropy of a message, the more information it contains. For a message consisting of symbols $S = \{s_1, ..., s_m\}$, each s_i with proability p_i, the entropy of the information is defined as

$$\sum_{i=1}^{m} p_i \log_2 \frac{1}{p_i} \tag{0.1}$$

If we use l_i bits to encode each symbol s_i (i=1,..,m), the average codeword length is lower bounded by the entropy, i.e.,

$$l_{average} = \sum_{i=1}^{m} p_i l_i \geq \sum_{i=1}^{m} p_i \log_2 \frac{1}{p_i}$$

When $log_2 \frac{1}{p_i}$ are integers, we can code each s_i using $l_i = log_2 \frac{1}{p_i}$ bits with a Huffman coding scheme. Otherwise, a more complicated scheme called arithematic coding can be used so that the average coding length $l_{average}$ is as close to the entropy bound as possible [2]. Huffman and arithmetic coding are both often called entropy coding techniques.

When we analyze efficiency of a compression method theoretically, we can use the entropy of its output message without actually encoding the message.

It can be verified that the entropy defined by Eq.(0.1) has the maximum value of $log_2 m$ when $p_i = \frac{1}{m}$ for all $i = 1, ..., m$. Therefore, we can always use an average $log_2 m$ bits to encode each symbol in a message which consists of m symbols $s_1, ..., s_m$.

0.4.2 Efficiency analysis

Our general subdivision theory tells us that the topological structure of a normal triangular patch can be represented (thus encoded) as a tree. Each node represents a triangle and each node has only four different ways to generate its children. Because the first elementary subdivision (binary subdivision) can involve adding a point to any of three edges, it has three cases. In addition, we have to include a 'leaf' case which has no subdivision. Thus we have seven subdivision cases to represent. Based on information theory, we can roughly estimate that each case can be coded by at most $log_2 7 \approx 2.81$ bits.

When a normal triangular patch is represented by a subdivision tree, the representation is usually not unique. Fig 2-12 (b)(c) are two different subdivision tree representations for the same triangular patch shown in Fig 2-12(a). Excluding leaf nodes, the tree in Fig. 2-12(b) has 6 subdivisions while Fig 2-12(c) has 11 subdivisions for the total 14 triangles. That brings up the question: how can we get the most efficient subdivision?

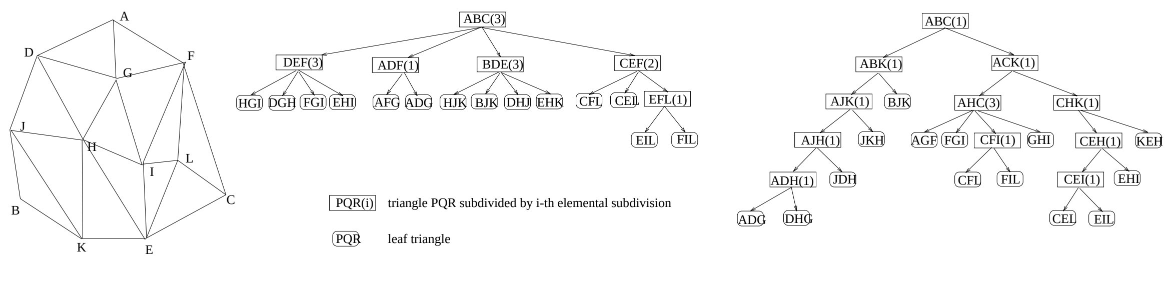

Fig 2-12 (a): mesh

Fig 2-12 (b): subdivision tree

Fig 2-12 (c): another subdivision tree

Fig. 2-12 a triangular patch and its subdivision tree representations

We define *BT rate* as 'bits/triangle (BT)' to measure the coding efficiency. The BT rate for each of our four elementary subdivisions is $\frac{log_2 7}{2}(\approx 1.40)$, $\frac{log_2 7}{3}(\approx 0.94)$, $\frac{log_2 7}{4}(\approx 0.70)$ and $\frac{log_2 7}{9}(\approx 0.31)$, respectively. Obviously, the first elementary (binary) subdivision has the least efficiency, the last (mitsubishi) has the greatest.

The subdivision tree for a normal triangular patch with n triangles has n leaf nodes and at most $n-1$ non-leaf nodes. Therefore, if we use $log_2 7$ bits for each subdivision case, the BT rate is upper bounded by $\frac{(2n-1)\log_2 7}{n} \leq 2\log_2 7 \approx 5.62$(BT). That happens when all non-leaf nodes use binary subdivisions. However, taking the probabilities into account, the BT rate of entropy coding is much smaller than the 5.62. Actually, we have the following theorem.

Theorem 5 *For a normal triangular patch with n triangles ($n \gg 1$), the entropy of BT rate for any subdivision tree representation of the patch is lower bounded by* $\frac{9}{8}\log_2 9 - 3 \approx 0.57$ *and upper bounded by 2.52.*

Proof: Let's assume there are n_i nodes using the i-th elementary subdivision (i=1,2,3,4) and the total number of triangles is n. Notice that the four elementary subdivisions increase the number of triangles by 1, 2, 3, and 8, respectively. Therefore, we have the following relations:

$$\sum_{i=1}^{4} k_i * n_4 = n - 1 \approx n \tag{0.2}$$

where $k_1 = 1, k_2 = 2, k_3 = 3, k_4 = 8$.

With entropy coding, the BT rate is expressed as

$$E = -\frac{\sum_{i=1}^{4}(n_i \log_2 \frac{n_i}{N}) + n \log_2 \frac{n}{N}}{n} \tag{0.3}$$

where $N = n + \sum_{i=1}^{4} n_i$. Maximizing E under the constraint Eq. (0.2), we immediately get the following equations using Langrange multiplers:

$$\frac{\log_2 \frac{n_i}{N}}{k_i} = \alpha = constant, \text{ for } i = 1, 2, 3, 4$$

Denoting $x = 2^\alpha$, we have

$$\sum_{i=1}^{4} n_i = \sum_{i=1}^{4} x^{k_i}(n + \sum_{i=1}^{4} n_i)$$

i.e.

$$\sum_{i=1}^{4} n_i = n\frac{\sum_{i=1}^{4} x^{k_i}}{1 - \sum_{i=1}^{4} x^{k_i}}$$

So,

$$n_i = x^{k_i}(n + \sum_{i=1}^{4} n_i) = n\frac{x^{k_i}}{1 - \sum_{i=1}^{4} x^{k_i}}$$

¿From Eq (0.2), we get

$$\frac{\sum_{i=1}^{4} k_i x^{k_i}}{1 - \sum_{i=1}^{4} x^{k_i}} = \frac{n-1}{n} \approx 1$$

i.e.,

$$\sum_{i=1}^{4} (k_i + 1)x^{k_i} = 1 \qquad (0.4)$$

Eq. (0.4) has a unique positive solution $x \approx 0.30435$, and thus $\alpha = \log_2 x \approx -1.7162$.

Based on the values of x and α, we also have the following $\sum_{i=1}^{4} n_i \approx 0.7399n$, $n_1 = x(n + \sum_{i=1}^{4} n_i) \approx 0.5295n$, $n_2 = x^2(n + \sum_{i=1}^{4} n_i) \approx 0.1612n$, $n_3 = x^3(n + \sum_{i=1}^{4} n_i) \approx 0.0490n$, $n_4 = x^8(n + \sum_{i=1}^{4} n_i) \approx 0.000128n$.

The maximum value of the BT rate can be calculated from Eq. (0.3): $E_{max} = 2.52$.

Because Eq (0.4) has only one positive root, which leads to the maximum value of E, the minimum value of E must be reached at the boundary of the region: $\{n_1 \geq 0, n_2 \geq 0, n_3 \geq 0, n_4 \geq 0, n_1 + 2 * n_2 + 3 * n_3 + 8 * n_4 = n - 1\}$. It is easy to verify that E gets its minimum value when $n_1 = n_2 = n_3 = 0, n_4 = n - 1 \approx n$, i.e., the minimum value is $E_{min} \approx \frac{9}{8}\log_2 9 - 3 \approx 0.57$ □

The proof above suggests us that a smaller BT rate can be obtained when fewer binary subdivisions are used and more quaternary and mitsubishi subdivisions are used.

Although topological information is only a small part of mesh representation, the efficiency of topological coding is still an important problem, especially for hardware supported rendering.

Assume there are n vertices in a big mesh. According to the Euler's theory, there are roughly $2n$ triangles. The simplest way to code the topological information is to store each triangle separately by indexing its three vertices. That costs $3\log_2 n$(BT). OpenGL allows triangles to be coded in strips. A triangle of k triangles can be coded by indexing only $k+2$ vertices. That reduces costs to $\frac{(k+2)}{k} log_2 n$ (BT). A generalized triangle strip method includes swap commands in a triangle strip [5]. Its BT rate is roughly $\frac{1}{2} log_2 n + c$ for $c \approx 2$ (BT). Deering developed the notion of *generalized triangle mesh* to reuse a limited number previously appearing vertices stored in a buffer [3]. The BT rate is about $\frac{1}{16} log_2 n + 4$ (BT). PM in [6] uses about $\frac{1}{2} log_2 n + 2.5$ (BT) for topological coding. And modified PM in [11] uses $\frac{1}{2} log_2 n + 3.5$(BT). Li's method [9] has an average BT rate of $\frac{1}{2} log_2 n + 4.5$ (BT).

We notice that all coding schemes above have a term $log_2 n$ in their BT rates. The reason is that they access triangles randomly. Random accessing is useful in rendering, especially in view-dependent rendering.

One big improvement made by Taubin and Rossignac is to arrange all vertices and triangles in a fixed way so that no indexing term $log_2 n$ appears [13]. An average cost of 2 (BT) is reported in their testing. More importantly, when all vertices are arranged in a fixed order, their geometric information can be coded efficiently using a bit-plane coding scheme. However, the method cannot generate hierarchical topology, and is not good for rendering.

Our method uses subdivision trees to put all vertices in a fixed positions. Once a tree is constructed, the topological information coded by the tree is complete. We need not spend $\log_2 n$ bits for indexing a vertex. The BT rate in our scheme is upper bounded by 2.52 (BT) for a normal triangular patch. The BT rate is lower than all existing methods, except [13]. Taubin et al. reported 2 (BT) in their tests, but didn't give theoretical analysis of their coding efficiency.

The efficiency of our method will be reduced when we cut a big mesh into many normal triangular patches. We mentioned earlier that we can always cut a mesh with genus g and k boundaries into $2n+k$ normal triangular patches with a penalty of adding n_k triangles and k vertices, where n_k is the total number of vertices in k boundaries. Thus, our efficiency reduces at most by $\frac{2n_k \log_2 7}{2n} = \frac{n_k \log_2 7}{n}$(BT). We should point out that for most cases, $n_k \ll n$. In addition, the actual cost may much less than the upper bound we estimated here. Therefore, our coding efficiency is believed to be very high.

The efficiency of topological coding is not the main goal of our compression. Most storage is spent on geometry coding, not topology coding. We use subdivision tree representation because it helps us to construct wavelets on an arbitrary mesh, and wavelets are expected to have excellent performance in geometry coding.

Bibliography

[1] Chow, M., Optimized Geometry Compression for Real-time Rendering, Proc. of Visualization'97, Oct., 1997

[2] Barnsley, M. and Hurd, L.P., *Fractal Image Compression*, AK Peters, Ltd., 1993

[3] Deering, M., Geometry Compression, SIGGRAPH'95, Aug., 1995

[4] De Floriani, L. and Puppo, E., Hierarchical Triangulation for Multiresolution Surface Description, ACM Trans. on Graphics, Vol.14, N.4, pages 363-411, 1995

[5] Evens, F., Skiena, S., and Varshney, A., Optimizing Triangle Strips for Fast Rendering, Proc. IEEE Visualization'96, pp 319-326

[6] Hoppe, H., Progressive Meshes, SIGGRAPH'96, pp 99-108, 1996

[7] Hoppe, H., View-Dependent Refinement of Progressive Meshes, SIGGRAPH'97, pp 189-198, 1997

[8] Kinsey, L., *Topology of Surfaces*, Springer-Verlag, 1993

[9] Li, J.K., Li, L., and Kuo, C.C., Progressive Compression of 3D Graphics Models Proc. of IEEE International Conf. on Multimedia Computing and System, Ottawa, Canada, 1997

[10] Nlson, M. and Gailly, J.L., *The Data Compression Book*, (2nd Ed), M&T Books, 1996

[11] Popovic, J., and Hoppe, H., Progressive simplicial complexes, SIGGRAPH'97, pp 217-224, 1997

[12] Rossignac, J., Geometric Simplification and Compression, in *Multiresolution Surface Modeling*, Course Notes #25, SIGGRAPH'97.

[13] Taubin, G., and Rossignac, J., Geometry compression through topological surgery, Research Report RC-20340, IBM, January, 1996