점진적 메쉬의 선택적 세분화를 위한 새로운 접근법

김준호, 이승용
포항공과대학교 컴퓨터공학과

{victor, leesy}@postech.ac.kr

A New Approach to Selective Refinement of Progressive Meshes

Junho Kim and Seungyoung Lee
Dept. of Computer Science & Engineering, POSTECH

요 약

점진적 메쉬 표현법은 2-다양체 성질을 가지는 임의의 삼각메쉬에 대한 다중해상표현이 가능한 방법으로, 최근 다중해상 모델링 및 렌더링 분야로 그 응용범위를 넓혀가고 있다. 본 논문에서는 점진적 메쉬의 선택적 세분화를 위한 새로운 방법을 제안한다. 제안하는 방법은 기존방법 [4, 3]과는 달리 거의 대부분의 경우 한 정점의 세분화로 인해 주위 정점이 세분화되지 않는 최적화된 방법이다. 본 논문에서 제안하는 새로운 선택적 세분화 방법으로 인해 점진적 메쉬의 활용분야가 더욱 넓어질 것으로 기대된다.

제 1 절 서론

삼차원 스캐너 장비의 보급으로 실제로 존재하는 물체로부터 직접 기하학적 데이터를 뽑아내어 모델링하는 기법이 최근에 많이 쓰이고 있다. 이러한 기법은 복잡한 물체에 대한 모델링을 쉽게 할 수 있는 장점이 있지만, 물체의 복잡도와는 상관없이 너무 복잡한 메쉬 모델이 얻어진다는 단점이 있다. 따라서 물체의 기하적 특성을 잘 반영하면서 보다 단순한 메쉬 구조로 표현하여 저장공간이나 렌더링 시간을 줄이고자 하는 메쉬 단순화 (mesh simplification) 및 메쉬의 다중해상표현 (multiresolution analysis)에 관한 연구가 있었다. 이러한 메쉬의 다중해상 표현방법 중 점진적 메쉬 (Progressive Mesh) 표현법은 입력으로 들어온 삼각 메쉬 $\hat{M}$에 대해 에지붕괴 (edge collapse)와 정점분할 (vertex split)이라는 간단하면서도 강력한 연산자로 다중해상 분석을 할 수 있는 방법이다.

본 논문에서는 점진적 메쉬의 새로운 선택적 세분화 (selective refinement) 방법을 제안한다. 제안하는 방법은 기존의 선택적 세분화 방법 [4, 3]과는 달리 정점분할 연산 간에 상호종속 관계를 최소로 하는 방법이다.

제 2 절 관련연구

Hoppe는 2-다양체 성질을 가지는 메쉬에 대해 다중해상 정보를 구하는 점진적 메쉬 표현법을 제안하였다 [2]. 점진적 메쉬 표현법은 메쉬 상의 한 에지를 골라 그 에지 양 끝에 달린 두 정점을 하나의 정점으로 합쳐 보다 단순한 메쉬를 만드는 에지붕괴 연산자 (*ecol*)와 그 역인 정점분할 연산자 (*vsplit*)에 기반하고 있다. 입력으로 주어진 메쉬 $\hat{M}$에 대해 사용자가 원하는 에러범위 혹은 메쉬의 복잡도 내에서 계속적인 에지붕괴 연산으로 가장 성근 메쉬 모델을 얻을 수 있으며, 반대로 가장 성근 메쉬 모델에 정점분할을 계속적으로 적용하여 원래의 입력 메쉬를 복원해 낼 수 있다. 따라서 입력 메쉬 $\hat{M} = M^n$는 가장 성근 메쉬 M^0와 계속적인 정점분할 연산 $\{vsplit_0, vsplit_1, \cdots, vsplit_{n-1}\}$로 표현될 수 있으며 임의의 해상도로 표현된 메쉬 M^i는 $M^0\{vsplit_0, vsplit_1, \cdots, vsplit_{i-1}\}$로 표현된다 [그림 1].

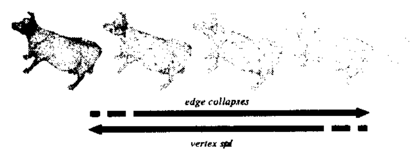


그림 1: Hoppe [2]가 제안한 점진적 메쉬 표현법

Xia와 Varshney [4], Hoppe [3]는 점진적 메쉬로 생기는 다중해상 정보를 이용하여 메쉬를 렌더링하는 경우 필요한 부분은 세밀하게 표현하고 그렇지 않은 부분은 적당히 성글게 표현하는 점진적 메쉬의 선택적 세분화 방법을 제안하였다. Xia와 Varshney [4]가 제안한 선택적 세분화 방법은 다음과 같다. 우선, 가장 성근메쉬를 만드는 과정에서 에지붕괴로 인해 생기는 각 정점의 1-고리 이웃 (1-ring neighbor) 상황을 모두 기억해 놓는다. 그런 다음 성근 메쉬 상의 한 정점에 대해 정점분할을 하려고 하면 기억된 1-고리 이웃 상황이 되도록 이웃의 정점들을 적당히 세분화하거나 단순화시킨 후 정점분할을 수행한다. 따라서 한 정점의 세분화 과정이 그 1-고리 이웃 정점들과 종속되어 있기 때문에 필요이상으로 넓은 부분이 세분화되어야 하는 단점이 있다. Hoppe [3]는 점진적 메쉬에서의 정점분할 및 에지붕괴 연산의 정의를 수정하고 정점 계층구조 (vertex hierarchy)를 제안하므로써 증가적인 방법으로 점진적 메쉬의 선택적 세분화가 가능하도록 하였다. 그러나 이 경우에 있어서도 한 정점이 분할하기 위해서는 그 1-고리 이웃 주변의 상황이 단순화 과정에서 기억된 상황과 같을 때에만 동작하기 때문에 한 정점의 선택적 세분화를 위해 그 주위의 정점까지 세분화시켜야 하는 단점을 여전히 가지고 있다.

기존 방법들은 모두 에지붕괴를 통해 새로운 정점이 생길 때, 그 정점의 정점분할을 위해 그 당시의 1-고리 이웃에 대한 정보를 기억하여 사용하기 때문에 정점분할 연산간의 상호연관 관계가 생겨 증가적인 방법에 의해서만 선택적 세분화가 가능하게 된다.

본 논문에서는 점진적 메쉬가 가지는 계층적 구획성을 이용, 점진적 메쉬의 부분적 세분화를 위한 새로운 방법을 제안한다. 제안하는 방법은 정점분할의 모양을 그 상황에 맞게 시스템 수행시 결정할 수 있으며, 대부분의 경우 선택된 정점 v_s가 분할되기 위해서 1-고리 이웃이 분할되는 일 없이 한번의 *vsplit* 연산만으로 정점 v_s에 대한 정점분할이 가능하다. 본 논문에서 제안하는 방법은 기존의 연구와는 달리 점진적 메쉬의 계층구획성으로부터 직관적으로 증명되며 보다 효율적인 방법이다.

제 3 절 점진적 메쉬의 계층적 구획성

에지붕괴 연산자의 특성으로 인한 점진적 메쉬의 계층적 구획성(hierarchical partitioning)은 이미 알려진 사실이다 [1]. 여기서는 이러한 점진적 메쉬의 계층적 구획화 특성을 보다 분석적인 방법에서 설명하고 이를 이용하여 제 4 장에서는 본 논문에서 제시하는 정점분할 및 에지붕괴 연산자가 항상 타당함을 보일 것이다.

점진적 메쉬의 계층 구획성은 Hoppe가 제시한 정점 계층구조로 쉽게 설명할 수 있다 [3]. 에지에 달려 있었던 두 정점과 그 에지의 에지붕괴로 인해 생기는 정점들은 부모-자식 관계를 가지는 계층구조를 만들게 된다. 또, 정점 계층구조에서 리프 노드(leaf node)들은 입력으로 들어온 메쉬 $\hat{M}$에 있던 정점이다. 따라서 정점 계층구조에 있는 임의의 정점 노드 v에 대해 그것을 루트로 하는 서브트리 $T(v)$를 생각한다면 그 서브트리의 리프 노드들은 원래 입력으로 들어온 메쉬상의 정점들이며, v는 $T(v)$ 내의 자식 노드들 간의 에지붕괴들에 의해 만들어졌기 때문에 $T(v)$의 리프 노드들의 집합 $S(v)$에 대해 다음이 성립한다.

$$S(v) = \{\hat{v}_1, \hat{v}_2, \cdots, \hat{v}_k\}$$
$$\forall i, j, \exists \text{ a path } W = \hat{v}_i \hat{v}_{m_1} \hat{v}_{m_2} \cdots \hat{v}_j$$
$$(\text{where } (\hat{v}_{m_p}, \hat{v}_{m_{p+1}}) \in \hat{M} \text{ and } \hat{v}_{m_p}, \hat{v}_{m_{p+1}} \in S(v))$$

또한 점진적 메쉬로 만들어지는 임의의 메쉬 M^i 상의 한 정점들은 서로들 간에 정점 계층구조에서 부모-자식 관계를 가지지 않을 뿐 아니라, 그들이 가지는 리프 노드 집합의 합집합은 항상 원래 메쉬 상의 모든 정점 포함하므로 다음이 성립한다.

$$S(v_i) \cap S(v_j) = \phi, (\text{where } i \neq j)$$
$$S(v_1) \cup \cdots \cup S(v_m) = \text{the set of 0-simplexes in } \hat{M}$$
$$(\text{where } \{v_1, \cdots, v_m\} \text{ is the set of 0-simplexes in } M^i)$$

위의 성질로부터 점진적 메쉬로 만들어지는 M^i 상의 각 정점들은 $\hat{M}$ 상의 정점들을 적절히 구획화하고, 또한 이러한 메쉬들을 계층적으로 만들 수 있기 때문에 점진적 메쉬 표현법은 입력 메쉬를 계층적으로 구획화하는 특성이 있다.

메쉬 상의 한 에지는 양 끝에 달려 있는 정점들이 연결되어 있다는 사실을 말해 주듯이, M^i 상의 한 에지는 에지 양 끝에 달린 두 정점이 대표하는 $\hat{M}$ 상의 두 구획 간의 연결 상태를 말해 준다. 따라서 점진적 메쉬 표현법에 의한 한 임의의 메쉬 M^i는 M^i 상의 각 정점들이 대표하는 $\hat{M}$의 구획 상황 그래프(partitioning configuration graph)로 볼 수 있다. 또한, 기존의 정점분할 및 에지붕괴 연산자는 구획 연산자로서의 측면에서 각각 하나의 구획을 인접한 두 구획으로 나누는 연산자, 그리고 인접한 두 구획을 하나의 구획으로 합치는 연산자로 볼 수 있다 [그림 2].

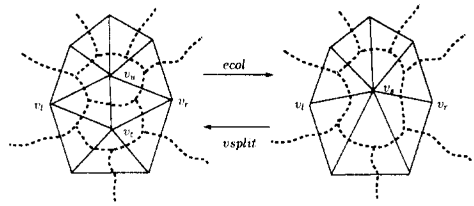


그림 2: 구획 연산자로서의 *vsplit*과 *ecol* 연산자

제 4 절 최적화된 정점분할

정점분할 $vsplit(v_s, v_t, v_u, v_l, v_r)$은 [그림 2]에서 보듯이 정점 v_s가 대표하는 $\hat{M}$ 상의 영역 $S(v_s)$를 영역 $S(v_u)$와 $S(v_t)$로 나눈다. 이 때 영역 $S(v_s)$가 영역 $S(v_u)$, $S(v_t)$로 나누어지는 형상이 마치 에지 (v_l, v_s)와 (v_s, v_r)로 영역 $S(v_s)$를 자른 듯한 모양을 띄게 된다. 즉, 영역 $S(v_s)$가 두 영역 $S(v_u), S(v_t)$로 쪼개지는 형상은 이미 정해져 있으며 그 쪼개지는 꼴에 대한 정보를 정점 v_l, v_r이 가지고 있는 셈이다. 그런데 듀얼 그래프로부터 알 수 있듯이 정점분할의 파라메터로 들어가는 v_l과 v_r이 [2]에서와 같이 v_s가 만들어질 당시에 v_u와 v_t 둘 모두에 에지로 연결되어 있던 1-고리 이웃 상의 두 정점이 아니더라도, 파라메터 v_l과 v_r이 나타내는 두 영역 $S(v_l), S(v_r)$이 $S(v_s)$의 분할된 두 영역 $S(v_u)$와 $S(v_t)$에 모두 연결된 구획이기만 하면 정점분할 연산자는 올바르게 작동하게 된다 [그림 3]. 따라서 [그림 3]의 성질

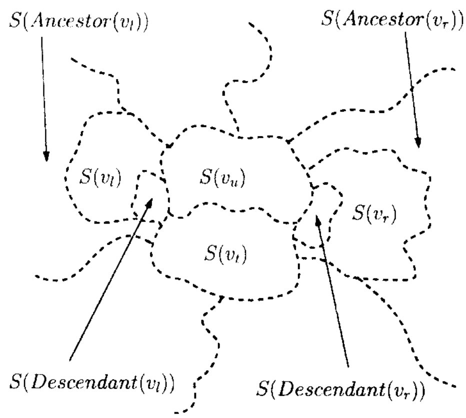


그림 3: 확장된 정점분할 연산

과 정점 계층구조 [3]의 성질을 이용하여 확장된 정점분할 연산자를 제안한다.

- $vsplit(v_s, v_t, v_u, v_{ll}, v_{rr})$은 다음 조건을 만족하면 올바른 연산자이다.

1. i) $v_{ll} = Ancestor(v_l)$ 이거나, ii) $v_{ll} = v_l$ 이거나, iii) $v_{ll} = Descendant(v_l)$이고 $S(v_{ll})$는 $S(v_u)$와 $S(v_t)$ 모두에 인접해 있다.
2. i) $v_{rr} = Ancestor(v_r)$ 이거나, ii) $v_{rr} = v_r$ 이거나, iii) $v_{rr} = Descendant(v_r)$이고 $S(v_{rr})$은 $S(v_u)$와 $S(v_t)$ 모두에 인접해 있다.
3. $v_{ll} \neq v_{rr}$

확장된 정점분할 연산자를 효율적으로 처리하기 위해 [그림 4]과 같이 에지 (v_u, v_t)에 대한 에지붕괴 연산을 할 때, $\hat{M}$ 상의 정점 중 $S(v_u)$와 $S(v_t)$에 연속적으로 모두 인접한 정점을 찾아 $v_s.\hat{v}_l, v_s.\hat{v}_r$로 설정한다. 따라서 최적화된 정점

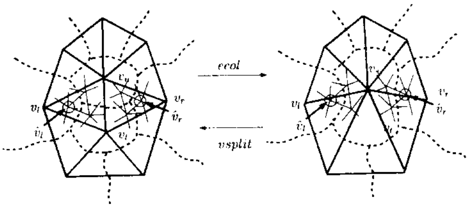


그림 4: $v_s.\hat{v}_l$과 $v_s.\hat{v}_r$의 설정

분할 연산자 및 에지붕괴 연산자는 다음과 같이 수행된다.

$$\begin{gathered} ecol(v_s, v_t, v_u, v_{al}, v_{ar}) \\ vsplit(v_s, v_t, v_u, v_{al}, v_{ar}) \\ (\text{where } v_{al} = ActiveAncestor(v_s.\hat{v}_l), \\ v_{ar} = ActiveAncestor(v_s.\hat{v}_r)), \text{ and } v_{al} \neq v_{ar}) \end{gathered}$$

만일 $v_{al} = v_{ar}$인 경우 정점분할 $vsplit(v_s, \cdots)$를 수행하기에 앞서 v_{al}에 대한 정점분할 $vsplit(v_{al}, \cdots)$를 먼저 수행한다. 여기서 생기는 정점분할 연산 간의 종속관계는 점진적 메쉬 연산자가 가지는 한계에 비롯된 것이며, 이와 같은 종속 관계가 생기는 경우는 아주 드문 경우여서 거의 대부분의 경우 한번의 정점분할 연산만으로 원하는 정점에 대한 정점분할을 수행할 수 있다.

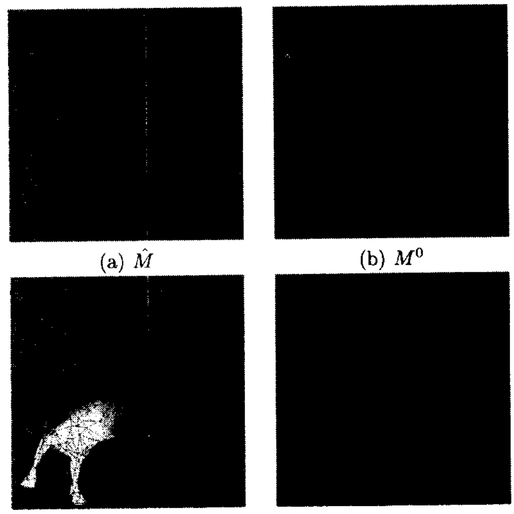
제 5 절 결과

[그림 5]는 본 논문에서 제안한 방법을 이용하여 점진적 메쉬의 선택적 세분화를 수행한 결과이다. [그림 5. (a)]는 입력 메쉬 $\hat{M}$인 소 모델이며, 계속적인 에지 붕괴연산으로 만든 가장 성근 메쉬 M^0는 [그림 5. (b)]와 같다. [그림 5. (c)]는 M^0 상의 각 정점을 서로 다른 색으로 칠해 그에 대응하는 $\hat{M}$ 상의 구획 상황을 보여 주고 있다. M^0의 정점 중, 소의 가슴 쪽에 해당하는 정점 자신과 모든 자식 노드들에 대해 본 논문에서 제시한 선택적 분할방법으로 세분화한 결과가 [그림 5. (d)]이다. [그림 5. (d)]와 같은 극단적인 경우에 대해서도 한 부분의 세분화로 인해 인접한 부분의 세분화가 연쇄적으로 일어나지 않고, 원하는 부분에 대해서만 $\hat{M}$까지 세분화됨을 알 수 있다.

제 6 절 결론 및 향후 연구과제

본 논문에서는 점진적 메쉬의 선택적 정점분할을 위해 중가적 계산이 필요하지 않은 최적의 방법을 제안하였다. 제안된 방법은 점진적 메쉬의 계층구획성을 이용하여 직관적으로 증명이 되며 기존 방법에 비해 쉽게 구현할 수 있는 장점이 있다. 또한 기존의 점진적 메쉬를 이용한 시점종속적 렌더링 시스템 [4, 3]에 곧바로 적용할 수 있는 방법이다.

본 연구에서 제안한 최적의 정점분할 연산으로 인해 다중해상표현 능력을 가지는 점진적 메쉬의 활용 범위가 더욱 넓어질 것으로 기대된다. 보다 효율적인 시점종속적인 렌더링 시스템의 디자인 뿐만 아니라 임의의 삼각메쉬에 대한 다중해상 에디팅(multiresolution editiong), 삼차원 메쉬 몰핑(3D mesh morphing) 등 다중해상 메쉬 응용 시스템 개발에 널리 쓰일 수 있을 것이다.



(a) $\hat{M}$ (b) M^0

(c) M^0에 의해 구획화된 $\hat{M}$ (d) 선택적 세분화의 결과

그림 5: 점진적 메쉬의 선택적 세분화의 예

참고 서적

[1] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. *ACM Computer Graphics (Proc. of SIGGRAPH '97)*, pages 209-216, 1997.

[2] Hugues Hoppe. Progressive meshes. *Computer Graphics (Proc. SIGGRAPH '96)*, pages 99-108, 1996.

[3] Hugues Hoppe. View-dependent refinement of progressive meshes. *ACM Computer Graphics (Proc. of SIGGRAPH '97)*, 1997.

[4] Julie C. Xia and Amitabh Varshney. Dynamic view-dependent simplification for polygonal models. *IEEE Visualization '96*, pages 327-334, 1996.