두 다면체 모델 사이의 점진적 표현을 계산하는 휴리스틱 방법 *

윤 원영[1], 최 정주[1], 이 인권[2]

[1] 아주대학교 미디어학과

[2] 연세대학교 컴퓨터과학과

[1] {yoon1young, jungju}@ajou.ac.kr

[2] iklee@yonsei.ac.kr

Heuristic Method for Computing Progressive Mesh Representation between Two Polygonal Models

Won-Young Yoon[1] , Jung-Ju Choi[1] , In-Kwon Lee[2]

[1] Division of Media, Ajou University

[2] Department of Computer Science, Yonsei University

요 약

본 논문에서는 서로 다른 개수의 정점을 가지는 두 다면체 사이의 점진적 다면체 모델 표현(Progressive Mesh Representation)을 계산하는 휴리스틱 방법을 제시한다. 정점의 개수가 각각 n, k개인 두 다면체 모델 M^n, M^k $(n > k)$에 대하여 M^n에서 서로 다른 k개의 정점을 선택한다. 선택된 k개의 정점을 기준으로 M^n의 모든 정점에 대한 클러스터링을 수행하여 k개의 정점군(Vertex Set)을 생성한다. M^n을 간략화하여 k개의 정점만을 가지는 모델 $M^{k'}$의 위상정보(Topology)를 M^k와 동일하게 유지하기 위하여 M^n 정점군들의 위상정보를 수정한다. 수정 생성된 정점군 내에서 선분병합(Edge Collapse)을 수행하면, 위상정보를 유지하면서 M^n에서 M^k로 변화하는 점진적 다면체 모델 표현을 얻을 수 있다. $M^{k'}$과 M^k의 정점간의 기하학적 위치 차이를 선형보간하여 선분병합이 일어날때 마다 반영하면 M^n에서 M^k로 기하정보를 부드럽게 유지하면서 변화하는 점진적 다면체 모델 표현을 얻을 수 있다. 본 논문의 연구결과는 기존의 DLoD(Discrete Level of Detail)를 지원하는 게임을 CLoD(Continuous Level of Detail)를 지원하는 게임으로 확장하는 등의 다양한 컴퓨터 그래픽스 응용문제에 사용할 수 있다.

제 1 절 서론

컴퓨터 그래픽스 응용의 한 분야인 컴퓨터 게임에서는 그래픽스 하드웨어의 발전에 따라서 게임의 사실성을 높이기 위하여 정점의 개수가 많은 다면체 모델을 많이 사용하는 추세이다. 정점의 개수가 많은 다면체 모델은 필연적으로 많은 계산 시간 및 렌더링 시간을 요구하게 되는 단점을 가지고 있다. 실시간성이 중요한 게임에서는 정점의 개수가 많은 복잡한 모델을 빠르고 효율적으로 처리하기 위해서 LoD(Level of Detail)를 포함하여 다양한 기법들이 사용되고 있다. 다면체 모델에 대한 LoD 기법을 구현하기 위해서 사용될 수 있는 다면체 표현 방법 중 하나가 점진적 다면체 표현법(Progressive Mesh Representation)이 있다[2]. 점진적 다면체 표현이란 주어진 복잡한 모델의 정점 개수를 간단한 연산자를 사용하여 줄여나가면서 그 정보를 다면체의 기하정보와 함께 저장하여 원하는 순간에 원하는 정점 개수를 갖는 간략화 모델을 구성하는데 사용되는 자료구조이다. 현재 방대한 환경과 빠른 수행속도를 요구하는 게임 분야에서 CLoD (Continuous Level of Detail)를 효과적으로 구현하는데 적용할 수 있다. 하지만 기존의 점진적 다면체 모델 표현법은 간략화를 진행한 후의 얻어지는 간략 모델의 결과를 예측하기가 어려워 캐릭터 디자이너가 원하는 간략화 모델을 계산과정에서 반영하기가 매우 어렵운 단점을 가지고 있다. 본 논문에서는 점진적 표현법 계산에서 목표 간략화 모델의 결과를 반영하는 새로운 점진적 모델 표현법을 구하는 휴리스틱한 방법을 제시한다. 본 논문에서 제시한 방법은 디자이너가 원하는 간략화 모델을 반영한 CLoD의 구현에 응용할 수 있으며, 기존의 게임이 주로 사용하고 있는 몇 개의 중간단계 간략화 모델세트를 사용하는 DLoD(Discrete Level of Detail)을 CLoD로 손쉽게 확장하는데 응용할 수 있다.

본 논문의 구성은 다음과 같다. 2절에서는 점진적 다면체 모델 표현법과 기존의 계산방법이 임의의 중간 단계 모델을 반영하지 못하는 이유에 대해 설명하고, 3절에서는 본 논문에서 제시하는 새로운 점진적 다면체 모델 표현법을 얻기 위한 알고리즘에 대해서 설명한다. 4절에서는 구현된 연구결과를 제시하고, 마지막으로 5절에서 결론 및 향후 연구과제를 제시한다.

제 2 절 점진적 모델 표현법

1996년 Hoppe에 의해서 제시된 점진적 모델 표현법이란 정점의 개수가 많은 복잡한 모델이 주어졌을 때 정점의 개수

*본 연구는 정보통신부 대학IT연구센터(ITRC) 육성 지원사업의 연구비 지원에 의해 수행되었음.

를 점진적으로 줄여서 원하는 정점의 개수를 갖는 모델을 구성할 때 사용되는 다면체 표현방법이다 [2] (그림 1 참조).

그림 1: 점진적 모델 표현법으로 구성한 간략화 모델의 예

2.1 점진적 모델의 생성과정

점진적 모델 표현을 생성하는데 기본적인 과정으로서, 선분병합 (Edge Collapse)연산과 오차 측정(Error Metrics)이 있다. 선분병합이란 다면체에서 하나의 선분을 선택하여 선분을 삭제함으로써 선분 양쪽 끝 두 정점을 하나의 정점으로 줄여 정점의 개수를 하나 줄이는데 사용하는 단위 연산이다 (그림 2 참조).

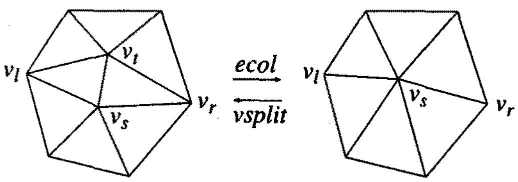


그림 2: 선분병합과 점분할 연산

오차 측정은 선분병합 이전의 다면체 모델과 선분병합 이후의 다면체 모델간의 기하학적 차이를 측정하는 것을 말한다. 이러한 기하학적 차이를 측정하는 방법에 관해 다양한 연구가 있다[1, 3]. 본 논문에서는 Garland의 QEM(Quadric Error Metric)을 오차측정 기법으로 사용하였다 [1].

주어진 다면체 모델에 대한 점진적 모델 표현을 생성하는 개략적인 과정은 다음과 같다.

1. 주어진 모델의 모든 선분에 대해서 오차 측정을 한다.
2. 오차 측정값이 가장 작은 선분을 병합한다.
3. 병합 정보를 저장한다.
4. 사용자가 원하는 정점의 개수에 도달할 때까지 위 과정을 반복한다.

다면체 모델 정점의 기하학적 정보와 함께 병합 정보를 포함하고 있는 자료구조를 점진적 모델 표현이라 하며, 일단 이상과 같은 계산을 한번만 수행하면 점진적 모델 표현을 가지고 손쉽게 필요한 정점 개수를 갖는 모델을 구성할 수 있다.

2.2 기존 방법의 문제점

모든 선분병합 과정이 종료되어 원하는 정점의 개수를 가진 간략화 모델을 얻었을 때, 간략화된 모델의 위상정보(*Topology*)와 기하정보(*Geometry*)를 결정하는 것은 선분병합의 순서이다. 선분 병합의 순서는 해당 단계에서 병합될 선분을 결정하게 되는 오차 측정법에 달려있으므로 결국 오차 측정법에 따라 최종 간략화 모델이 결정되게 된다. 모든 종류의 선분병합 순서의 개수는 병합 되어야 할 선분의 개수의 순열조합의 수와 같다. 이중에서 원하는 위상구조를 갖는 모델을 선택하여 선분병합 순서를 결정하는 오차측정법을 찾기란 매우 어려운 일이다 (그림 3 참조)

$$
\begin{array}{ccccc}
 & \overset{\mathrm{ecol}^1_{n-1}}{\nearrow} & M_1^{n-1} & \rightsquigarrow & M_\alpha^k \\
 & \overset{\mathrm{ecol}^2_{n-1}}{\rightarrow} & M_2^{n-1} & \rightsquigarrow & M_\beta^k \\
M^n & \vdots & \vdots & \vdots & \vdots \\
 & \overset{\mathrm{ecol}^{m-1}_{n-1}}{\rightarrow} & M_{m-1}^{n-1} & \rightsquigarrow & M_\psi^k \\
 & \overset{\mathrm{ecol}^{m}_{n-1}}{\searrow} & M_m^{n-1} & \rightsquigarrow & M_\omega^k
\end{array}
$$

그림 3: 오차측정 법에 따른 수많은 선분병합의 순서

본 논문에서는 서로 다른 정점 개수를 갖는 임의로 주어진 두 개의 다면체 모델에 대하여 하나의 다면체 모델로부터 얻어지는 많은 선분병합 순서 중에서 다른 하나의 다면체 모델에 이르는 점진적 모델 표현법을 계산하는 휴리스틱 방법을 제시한다.

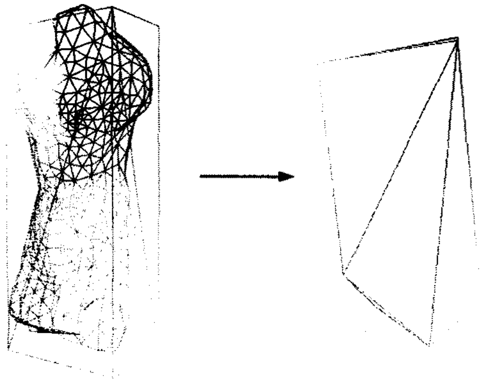
제 3 절 알고리즘

2.2절에서 주어진 문제를 해결하기 위해서는 선분병합을 적용한 간략화된 결과 모델과 주어진 다면체 모델의 위상정보를 동일하게 하고, 점진적 과정의 중간 다면체 모델들에 주어진 간략화 모델의 기하정보를 반영하는 두가지 문제를 해결해야 한다. 2.2절에서 설명한 것처럼 중간단계 모델의 위상정보를 반영하는 선분병합순서를 결정하는 오차측정법을 찾는 것은 매우 어려운 일이다. 본 논문에서는 이러한 오차측정법을 찾기보다는 선분병합 순서를 제한하는 방법을 제시하였다.

3.1 정점 집합

그림 4(a)와 같이 어떤 모델을 구성하고 있는 정점들을 특정한 기준에 의해 집합으로 구분했다고 가정하자.

만약 정점 집합 내부에서만 선분병합이 일어나도록 제한한다면 모든 선분병합 연산이 끝난 후 생성되는 간략화 모델의 정점 수는 정점 집합의 수와 동일하게 되고, 위상구조는 선분병합 연산을 적용하기 전 정점 집합들의 위상구조(정점 집합 사이의 연결 관계)와 동일하게 된다는 사실을 알 수 있다 (그림 4(b) 참조). 즉, 정점 집합 사이의 위상구조를 조정할 수 있다면 원하는 위상구조를 가지는 간략화 모델을 얻을 수 있게 된다.



(a) (b)

그림 4: (a)정점 집합으로 나누어진 정점들, (b)정점 집합 내에서 선분병합을 수행한 결과

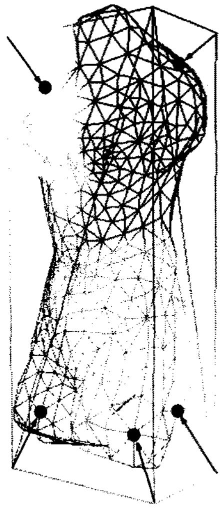


그림 5: 대표점을 중심으로 정점 집합의 구성

3.1.1 정점 집합의 구성

주어진 간략화 모델과 동일한 위상구조를 갖는 정점 집합을 구성하기 위해서 다음과 같이 정점 집합을 만든다.

1. 주어진 단순한 모델의 각각의 정점과 거리가 가장 작은 정점을 복잡한 모델에서 찾는다. 이 정점을 대표점이라고 하자 (그림 5 참조).
2. 각 대표점을 중심으로 연결거리를 기준으로 하는 Voronoi 집합을 생성하여 정점들을 클러스터링한다. 연결거리란 특정 정점에서 해당 정점으로 도달하기 위해 거치는 선분의 개수의 최소값이다.
3. 생성된 정점 집합의 위상관계를 주어진 간략화 모델과 동일하게 하기 위해서 정점 집합을 수정한다.

3.1.2 정점 집합의 위상정보 수정

주어진 다면체 모델 M^n과 M^k $(n > k)$에 대해서 3.1.1절과 같은 방법으로 M^n의 정점 집합을 구성했을 때 정점 집합 사이의 연결관계가 그림 6(b)와 같다고 하자.

그림 6에서 $V^*(v)$는 v를 대표점으로 갖는 정점 집합을 나타내며, $v_{i,k}$(또는 $v_{i,n}$)는 M^k(또는 M^n)에 존재하는 i번째 정점을 나타낸다. 그림 6(b)의 (1)번과 같은 경우 왼쪽에 주어진 간략화 모델 M^k에는 연결이 존재하지만 오른쪽 M^n에서 정점 집합간의 연결은 끊어져있다. 이를 왼쪽 간략화 모델의 위상구조처럼 연결해 주어야 한다. 이 문제를 해결하기 위해 그림 7(a)와 같이 단순히 최단경로를 따라서 두 정점 집합 $V^*(v_{i,n})$과 $V^*(v_{j,n})$을 연결한다면 연결 경로 상의 정점 집합 $V^*(v_{l,n})$이 분할되는 문제가 발생할 수 있다. 따라서 그림 7(b)와 같이 연결이 통과하는 정점 집합 $V^*(v_{l,n})$의 경계를 따라 연결해야 한다.

연결이 통과하는 정점 집합은 연결을 해야 하는 두 정점 집합을 이루고 있는 모든 정점 사이에 최단경로를 구해서 그 중에 연결거리가 가장 작은 경로가 지나가는 정점 집합이다. 연결이 관통하는 정점 집합의 경계를 따라 경로를 설정할 때 경로가 설정되는 정점 집합의 경계는 주어진 간략화 모델의 위상구조를 참고로 결정된다.

그림 8[B]에서처럼 M^n의 정점 집합이 구성되었다고 하자. 그림 8[A]에서 주어진 간략화 모델 M^k의 위상구조를 보면 그림 8[B]에서처럼 집합 $V^*(v_{l,n})$과 집합 $V^*(v_{m,n})$은 접하고 있지 않아야 한다. 이때 두 집합간의 최단경로가 관통하는 정점 집합이 $V^*(v_{l,n})$라고 한다면 접하지 말아야 하는 집합 $V^*(v_{m,n})$과 접하고 있는 집합 $V^*(v_{l,n})$상의 경계를 따

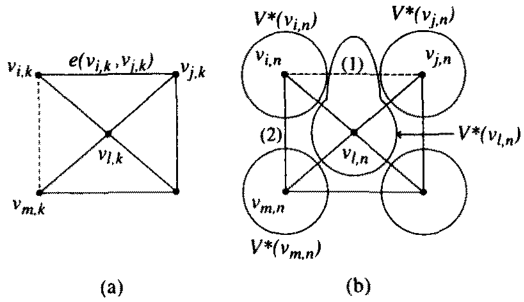


그림 6: (a) M^k의 위상구조, (b) M^n 정점 집합의 위상구조

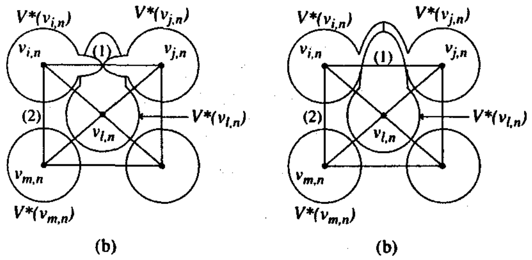


그림 7: 두 정점 집합 $V^*(v_{i,n})$과 $V^*(v_{j,n})$을 $V^*(v_{l,n})$을 통하여 연결하는 방법. (a) 최단경로 연결, (b) 우회경로 연결

라 경로를 구성하게 되는 결과를 가져온다.

연결경로를 구성할 때 연결경로가 관통하는 정점 집합의 모든 정점을 포함하는 경우가 있다. 그림 8[C]가 이와 같은 경우인데 이 경우 관통하는 정점 집합을 바꿔서 문제를 해결할 수 있다. 그림 8[B]에서 집합 $V^*(v_{l,n})$이 경로에 완전히 포함 되는 집합이라고 하면 $V^*(v_{l,n})$과 접하지 말아야 할 $V^*(v_{m,n})$을 관통하는 집합으로 바꾸어도 문제가 발생하지 않는다 (그림 8[D]). 이 경우 $V^*(v_{i,n})$에서 $V^*(v_{j,n})$에 이르는 경로는 $V^*(v_{m,n})$에서 $V^*(v_{l,n})$에 접하는 경계를 따라 구성되게 된다. 이와 같이 경로설정을 위하여 정점 집합을 바꾸어도 모든 문제가 해결되지는 않는데, 정점 집합을 바꾸었을때 공교롭게도 바꾼 정점 집합의 모든 정점이 경로에 포함 되어버리는 경우가 발생할 때이다. 이 경우에 대해서는 아직까지 효과적인 방법을 찾지 못하였다.

또 한가지 문제가 되는 경우는 정점 집합의 분할을 피하기 위해 관통 정점 집합의 경계를 따라 경로를 구성해도 정점 집합이 분할되는 경우이다. 그림 9과 같이 정점 집합 내에 병목지점이 있을 경우인데 이 부분을 경로가 지나간다면 관통 정점 집합이 분할된다.

이 문제는 '구멍 메우기'라는 방법을 통해 해결하였다. 모든 정점에는 0부터 시작하는 정수 값으로 해당 정점이 속하는 정점 집합이 표시되어있다. 경로 구성이 끝난 후 관통 정점 집합의 정수번호를 갖고 있는 모든 정점의 집합번호를 -1로 설정한 후 대표점을 중심으로 다시 한번 해당 번호로 클러스터링을 수행한다. 그 후 모든 정점의 집합번호를 검사해서 -1을 가지고 있는 정점이 발견된다면 해당 정점 집합이 분할 된 경우이다. -1로 설정되어있는 부분을 구멍이라고 부른다. 구멍의 정점들은 구멍을 접하고 있는 정점 집합에 적당히 분배한다. 이전에 현단계의 관통 집합의 대표점이 다른 경로에 의해서 다른 정점 집합에 분배되어 버렸을 경우 '구멍 메우기'는 정상적으로 시행될 수 없으므로 대표점을 정점 집합 내의 다른 점으로 대체한다.

그림 6(b)의 (2)으로 표시된 연결의 경우 해당 선분에 대하여 선분병합이 수행된 후 이상상태(Degenerate Case)를 만들게 된다. M^k에서의 두 점 $v_{i,k}$와 $v_{m,k}$의 연결 거리가 1보다 크지만 M^n에서 두 정점 $V^*(v_{i,k})$와 $V^*(v_{m,k})$의 연결 거리가 1이 되는 경우 발생한다. 실험적인 결과에 의하면 이 경우는 앞서 언급한 그림 6(b)의 (1)번 문제를 해결하는 과정에서 경로를 재설정 하면서 정점 집합을 수정하기 때문에 더불어 해결되는 경우가 많이 발생하였다. 만일 해결되지 않고 남아 있는 경우에는 두 정점 $v_{i',n} \in V^*(v_{i,n})$와 $v_{j',n} \in V^*(v_{j,n})$가 존재하여 $d(v_{i',n}, v_{j',n}) = 1$인 점 $v_{i',n}$, $v_{j',n}$중 하나를 선택하여 $V^*(v_{i,n})$에서 $V^*(v_{j,n})$에 이르는 경로 중에서 직접 연결되지 않는 최단경로상의 집합 $V^*(v_{l,n})$에 삽입한다. 이때 $d(a,b)$는 정점(혹은 정점 집합) a에서 b에 이르는 연결 거리를 나타낸다. 이렇게 되면 $d(V^*(v_{i,n}), V^*(v_{m,n})) > 1$이 되기 때문에 선분 병합 후 $v_{i,k}$와 $v_{m,k}$사이에 선분이 생기는 것을 막을 수 있다.

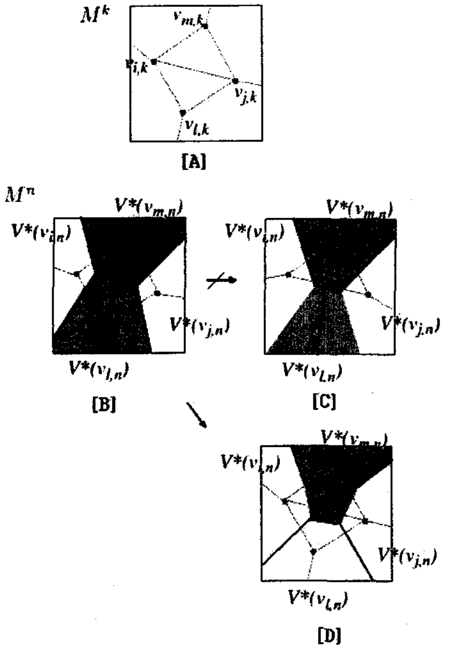


그림 8: 연결경로가 지나가는 정점 집합 바꾸기

지금까지 설명한 두 가지 문제는 다면체 모델 M^k가 M^n의 선분병합 연산만으로 구성되는 모델이 아닐 가능성이 있기 때문에 발생한다. 다면체 모델의 경우 임의의 두 정점 간의 연결 경로가 매우 많기 때문에 대부분의 경우 경계를 따라서 두 정점 집합을 연결하거나, 두 정점 집합을 분리하는 경우의 수가 대단히 많아 대개의 경우는 지금까지 설명한 휴리스틱한 접근으로 문제를 해결할 가능성이 높다. 실험 결과 k가 n보다 충분히 작은 수일수록 생성된 각각의 정점 집합을 구성하고 있는 정점수가 많아져 위에 언급한 문제들을 해결하는데 유리했다. 극단적으로 $n = k+1$와 같은 경우, M^n에서 단 한개를 제외한 모든 정점 집합은 오로지 하나의 정점만으로 구성된다. 이때 정점 집합간의 위상 정보가 불일치 하여 정점 집합을 수정하려는 경우 주변의

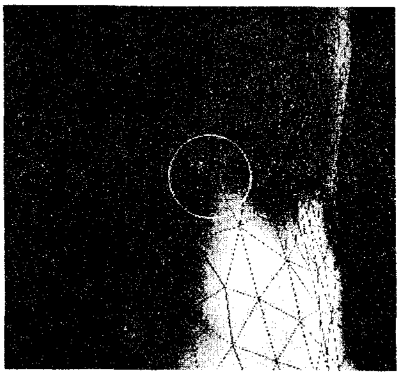


그림 9: 정점 집합 내 병목지점

어떠한 정점 집합을 선택하여도 우회 경로가 존재하지 않는 가능성이 대단히 높으며, 단 하나의 정점으로만 정점 집합이 구성되어 있으므로 연결관계를 분리할 수 있는 방법도 존재하지 않는다.

3.2 선분병합 순서 결정

집합간의 위상수정 문제가 모두 해결되면 선분 병합의 문제는 기존의 방법을 그대로 사용하면 된다[3, 4].

이때 선분 병합을 할 선분은 서로 다른 두 정점 집합에 걸쳐 있는 선분이 선택되지 않도록 한다. 두 정점 집합에 걸쳐 있는 선분을 병합했을 때 집합간의 위상관계가 변하지 않고 M^k의 정점들에 해당하는 M^n의 정점들만 유지된다면 해당 선분을 병합해도 문제가 되지 않는다. 하지만 두 정점 집합에 걸쳐있는 선분이 병합되었을 경우 어느 집합으로 남아 있는 한 점을 포함할 것인지 결정하는 문제를 피하고, 간략화 후 정점 집합의 위상을 그대로 보존할 수 있기 때문에 정점 집합 사이에 걸쳐 있는 선분은 선분병합의 대상에서 제외하였다. 간략화 모델인 M^k의 점들은 M^n에 있는 정점들 중에서 비교적 중요하기 때문에 캐릭터 디자이너에 의해서 남아있을 가능성이 크고, 이러한 중요한 점으로부터 거리가 먼 점들은 시각적으로 덜 중요한 점이라고 생각할 수 있다. 이러한 의미에서 두 집합 사이에 걸쳐있는 선분을 먼저 병합하는 것이 시각적으로 우수한 효과를 가질 가능성도 있다. 하지만 남아 있는 정점의 집합포함 결정 문제와 집합간의 위상관계를 유지하는지를 검증하는 문제를 피하기 위해서 본 논문의 실험에는 적용하지 않았다.

정점 집합 내에서만 선분병합을 수행하여도 결과가 이상상태가 되는 위상을 만들어내는 선분이 있는데 이 선분 또한 선분병합을 피해야 한다. 그림 10에서 두 정점 v_3와 v_4를 연결하는 선분을 병합할 경우 새로운 정점의 위치에 따라 주변 평면들이 교차하는 등 해당 다면체가 비다양체(non-manifold) 상태가 되는 경우가 발생하므로 선분병합 연산을 수행해서는 안된다. 일반적으로 동일한 정점 집합에 존재하는 두 정점 v_i와 v_j를 병합하려는 경우, v_i와 v_j를 연결하는 선분을 공유하는 두개의 삼각형에서 v_i와 v_j를 제외한 또 다른 꼭지점 v_k, v_l의 연결개수(Valence)가 4보다 작은 경우는 선분 병합후에 항상 이상상태를 만들게 된다.

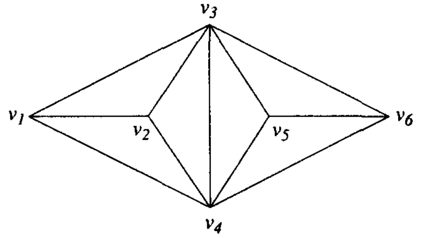


그림 10: $\overline{v_3v_4}$ 선분병합 결과 이상상태가 발생하는 경우

M^n의 독립된 정점 집합 내에서 선분 병합을 진행하면 진행과정에서 자연스럽게 두 정점 집합 사이에 걸쳐있는 선분의 개수가 줄어들게 되고 결과적으로 모든 선분 병합연산을 수행하면 연결된 두 정점 집합 사이에는 단 하나의 연결 성분만 남게 되어 M^k의 동일한 위상 구조를 얻을 수 있다.

3.3 기하정보의 반영

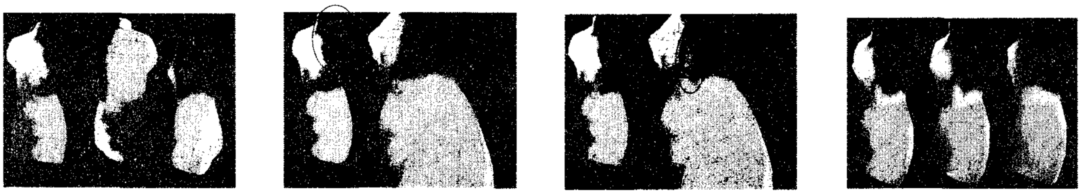
두 다면체 모델 사이의 기하정보는 M^k의 정점 v_k와 이에 해당하는 M^n상의 대표점 $v_{k'}$의 기하학적 차이를 선형보간(Linear Interpolation)하여 반영한다. 복구 시 본래 위치와 현 정점 집합에 해당하는 주어진 단순화 모델의 정점간의 차이 벡터를 구한 후, 현재 모델의 정점 수와 주어진 복잡한 원래 모델, 간략화 모델의 정점 수를 이용해서 보간 비율을 구한다. 보간 비율을 구한 후에는 차이벡터상에 보간 비율에 해당하는 위치로 해당 정점을 이동한다.

제 4 절 연구 결과

다면체 모델 M^n으로는 711개의 정점을 가지고 있는 Venus모델을 사용하였으며 M^k에 해당하는 모델은 결과의 명확성을 위해 간단한 직육면체 모델을 사용하였다 (그림 11, 12 참조). 다면체 모델의 표현을 위하여 사선분(Quad Edge) 자료구조를 사용하였으며, DirectX 8.1 라이브러리를 사용하여 구현하였다.

제 5 절 결론 및 향후 연구 과제

본 논문에서는 서로 다른 개수의 정점을 갖는 임의의 두 다면체 모델 M^n, M^k $(n > k)$이 주어졌을 때, M^n으로 부터 M^k로 전이하는 다면체 모델 M^n의 점진적 표현법을 구하는 몇 가지 휴리스틱에 기초한 방법을 제안하고 구현결과를 제시하였다. M^n으로 부터 k개의 대표점을 선택한 후, 대표점을 기준으로 M^n을 k개의 정점 집합으로 분할한다. M^n의 정점 집합 위상정보를 M^k의 위상정보와 동일하도록 정점 집합을 수정한다. 정점 집합 내에서 선분병합 연산을 수행하면 M^n으로 부터 위상정보를 유지하면서 M^k에



(a) (b) (c) (d)

그림 11: (a) 6개의 정점 집합으로 분할된 M^n모델, (b) M^k의 위상정보와 비교결과 접해 있어야할 두 정점 집합이 접하지 않는 경우, (c) 경로의 재설정을 통해서 중간 정점 집합의 경계를 따라서 두 정점 집합을 연결한 경우, (d) 경로를 모두 재설정 후 정점 집합 내에서 선분 병합 연산을 수행하는 중간 과정의 결과

이르는 다면체 모델 표현을 얻을 수 있다. M^n의 대표점과 M^k의 정점과의 차이를 선형보간하여 중간 단계의 모델에 적용하면 기하정보를 동시에 유지하는 점진적 다면체 모델 표현을 얻을 수 있다. 제시된 방법은 게임과 같이 이미 중간 단계 모델세트를 가지고 있는 DLoD를 CLoD가 가능하게 확장된 점진적 모델 표현법을 쉽게 얻을 수 있다.

본 논문에서 M^k의 위상정보를 유지하기 위해서 M^n의 정점 집합을 수정하는 여러가지 방법을 제시하였지만, 모든 문제를 해결한 것은 아니다. 특히 위상정보를 동일하게 유지하기 위해서 접하고 있지 않은 두 개의 정점 집합을 연결할 때 연결 경로가 사이에 놓여있는 정점 집합의 모든 경로를 포함하는 경우와, 우회할 수 있는 조건을 만족하는 정점 집합이 없을 경우는 해결방법을 찾지 못하였다. 이렇게 정점 집합간의 위상정보가 주어진 간략화 모델과 일치하지 않을 경우는 간략화 모델이 복잡한 모델로 부터 선분병합 연산에 의해 얻어진 모델이 아니기 때문일 가능성이 크며, 이 경우 제한적으로 점 분할(Vertex Split)연산을 수행한 후 선분 병합을 수행하는 방법이나, 선분교체(Edge Swap)를 사용해 다면체 모델의 위상정보를 변경해 해결하는 방법에 대한 연구가 필요하다.

참고 문헌

[1] M. Garland and P. Heckbert. Surface simplification using quadric error metrics. In *Proc. of SIGGRAPH '97*, pages 209–216, 1997.

[2] H. Hoppe. Progressive meshes. In *Proc. of SIGGRAPH '96*, pages 99–108, 1996.

[3] H. Hoppe. View-dependent refinement of progressive meshes. In *Proc. of SIGGRAPH '97*, pages 189–198, 1997.

[4] P. Sander, J. Snyder, S. Gortler, and H. Hoppe. Texture mapping progressive meshes. In *Proc. of SIGGRAPH 2001*, pages 409–416, 2001.

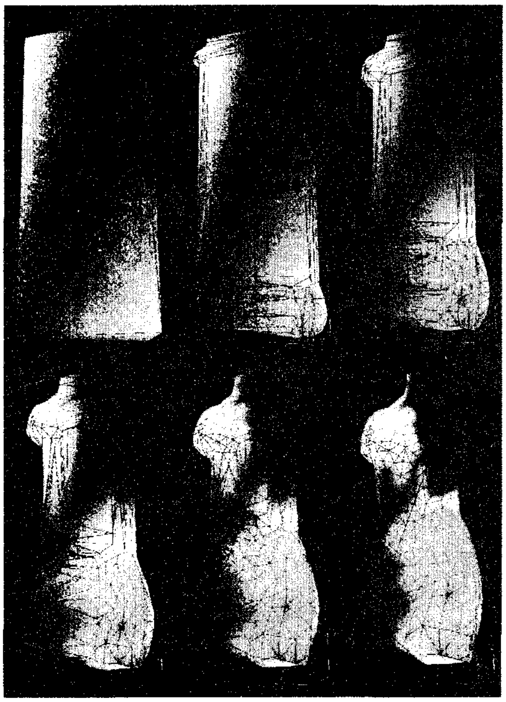


그림 12: 다면체 모델의 점진적 표현을 사용해 계산한 주어진 두 다면체 모델 사이의 중간 모델들