Wavelet 기반 LOD 가상객체 표현 시스템

김 기 호[†]·유 황 빈[††]

요 약

3차원 Object를 LOD(Level of Detail)에 의하여 표현하기 위해서는 메쉬의 상세정도에 따른 여러 종류의 메쉬가 필요하다. 본 논문에서는 웨이블렛 변환을 이용하여 다종 메쉬(multiresolution mesh)를 생성함에 따라, 사용자가 원하는 상세 종류별로 모든 메쉬를 만들지 않고, 상세 정보에 해당하는 웨이블렛 계수(wavelet coefficient)만을 갖도록 하여, 데이터 전송 및 관리등의 효율성 증대에 기여할 수 있는 시스템을 개발하였다.

Wavelet-based Level-of-Detail Virtual Object Representation System

Ki-Ho KIM[†] · Hwang-Bin RYOU[††]

ABSTRACT

Representing 3-D objects with LOD requires a set of appropriate meshes according to the detail requirements. We have developed a system for improved geometry model data transmission and management by having ony the wavelet coefficients of the model corresponding to the detail levels, instead of generating all the meshes through wavelet transformation, when generating multiresolution meshes.

1. 서 론

정보화 사회의 발달로 가상공간(Cyber Space)에서의 디지털 정보를 이용한 사회 및 경제 활동이 높아지고 있으며. 제한된 전송속도를 갖고 있는 통신망에서는 실물기반으로 가상 물체(virtual character)를 사용하기 보다는 3차원 모델러에 의하여 인공적으로 만든 가상물체를 사용하고 있다.

그러나, 컴퓨터 사용자들은 실물과 거의 동일한 가상물체 제작을 요구함에 따라, 가상물체를 형성하고 있는 데이터량은 점점 방대해지고 있어, 현재의 통신망으로는 사용자들의 욕구를 충족할 수 없는 현실이다.

† 준 회 원 : 한국전자통신연구원 선임연구원
†† 종신회원 : 광운대학교 컴퓨터과학과 교수
논문접수 : 1999년 11월 18일, 심사완료 : 2000년 3월 3일

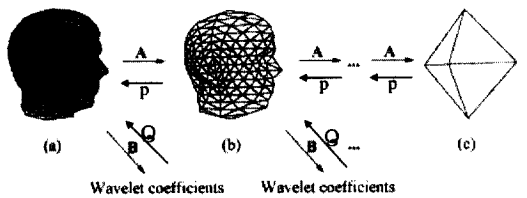
실물로부터의 모델 생성하는 실물기반 가상 물체 제작을 3차원 LOD(Level of Detail)에 의하여 표현하려면, 메쉬의 상세정도에 따른 여러 종류의 메쉬가 필요하다[1].

그러나, 본 논문에서는 웨이블렛 변환을 이용하여 다종 메쉬를 생성함에 따라, 사용자가 원하는 상세 종류별로 모든 메쉬를 만들지 않고, 상세 정보에 해당하는 웨이블렛 계수(wavelet coefficient)만을 갖도록 하여, 데이터 전송 및 관리 등의 효율성 증대에 기여할 수 있는 방법을 제안한다.

이 방법은 영상압축 효과 외에도 네트워크를 통해 영상을 전송할 때 낮은 해상도로 먼저 전송하고 차츰 영상이 또렷해 지도록 하는 등의 다양한 잇점을 제공할 수 있다.

2. 웨이블렛 변환(wavelet transform)

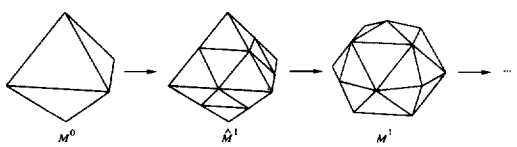
웨이블렛은 함수를 계층적으로 분해(decompose)하기 위한 수학적 도구이다. 웨이블렛을 적용함으로써 함수는 대략적인 전체 형태와 여러 단계로 이루어지는 상세 정보로 표현된다. 표현하고자 하는 함수가 영상이든지 곡선이든지 또는 3차원 표면 이든지에 상관없이 웨이블렛은 LOD(Level of Detail)을 표현하는 훌륭한 기술이다. 원래 웨이블렛은 근사이론과 신호 처리에서 시작되었으나 근래에 들어서는 컴퓨터 그래픽스의 다양한 분야에 응용되고 있다.



(그림 1) 3차원 메쉬 분석(웨이블렛 변환)

(그림 1)은 3차원 메쉬에 웨이블렛 변환과정을 보여주는 것으로서, 웨이블렛 변환은 원래의 메쉬로부터 시작하여 가장 단순한 모델로 "분석"을 하면서 상세 정보에 해당하는 웨이블렛 계수를 구한다. 이와같은 작업을 다중 해상도 분석(multiresolution analysis)라고 한다.

다중 해상도 분석을 하기 위한 첫 번째 작업은 스케일 함수(scaling function)를 결정하는 것이다. Lounsbery [2]는 메쉬에서의 스케일 함수를 결정하기 위해 세분화(subdivision)를 사용하였다. 세분화란 가장 단순한 형태의 모델을 순차적으로 잘게 나누고 변형하여 원하는 임의의 형태의 모델을 생성해내는 작업이다. 일반적으로 사용되는 메쉬 세분화에서는 한단계 세분화 할 때마다 한 개의 삼각형이 네 개의 삼각형으로 분할된다.

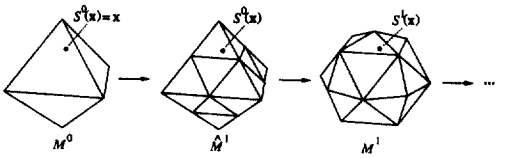


(그림 2) 3차원 메쉬의 세분화

(그림 2)는 이러한 세분화 과정의 예를 보이고 있다. 세분화 과정의 각 단계는 다시 두 과정으로 구성되는데 하나의 삼각형을 네 개로 분할하는 과정을 "분리(split)"라 하고 새로 생긴 꼭지점들의 위치를 옮겨서 모델을 변형하는 과정을 "변형(averaging)"이라 한다. 만약 V^j를 j단계의 메쉬를 구성하는 꼭지점들이라 하면 다음을 만족하는 행렬 P^j가 존재하는데 이 행렬을 "세분 행렬"이라 한다. 이 행렬이 세분화의 특징을 결정짓기 때문이다.

$$V^{j+1} = P^j V^j$$

세분화를 이용해 스케일 함수를 정의하기 위해 먼저 메쉬를 매개화(paramiterize)하는 함수 S(x)를 정의한다. 메쉬의 매개화는 서로 다른 단계의 메쉬에 속하는 점들의 대응관계를 구하는 것이다. 이는 (그림 3)에 나타나 있다.



(그림 3) 메쉬의 매개화

이러한 메쉬 매개화 함수 S(x)는 무한 프로세스(limiting process)에 의해 다음과 같이 세단계로 정의될 수 있다.

① $S^0(x) := x, x \in M^0$

② $S^{s-1}(x)$가 $\hat{M}^s$의 삼각형 ($\hat{v}_a^s$, $\hat{v}_b^s$, $\hat{v}_c^s$)내부에서 중심 좌표계 (α, β, γ)에 있다고 할 때 $S^s(x) := \alpha v_a^s + \beta v_b^s + \gamma v_c^s$이다. 이때 (v_a^s, v_b^s, v_c^s)는 $\hat{M}^s$의 ($\hat{v}_a^s$, $\hat{v}_b^s$, $\hat{v}_c^s$)에 대응되는 M^s에서의 삼각형이다.

③ $S(x) := \lim_{s\to\infty} S^s(x)$

위의 ②식에서 $b^s(x) = (0\cdots0\alpha0\cdots0\beta0\cdots0\gamma0\cdots0)$을 대입하여 행렬식으로 나타내면 아래 (1)식과 같다.

$$S^s(x) = b^s(x)\, V^s \qquad (1)$$

그런데 V^s는 세분화에 의해 생성된 메쉬이므로

$$V^s = P^{s-1}P^{s-2}\cdots P^0 V^0 \qquad (2)$$

를 만족하고 결과적으로

$$S^s(x) = b^s(x) P^{s-1} P^{s-2} \cdots P^j V^j \tag{3}$$

가 된다. 이때,

$$\Phi^{s-j}(x) := b^{s(x)} P^{s-1} P^{s-2} \cdots P^j \tag{4}$$

라고 정의하면

$$\Phi^{s-j}(x) = \Phi^{s-j+1}(x) P^j \tag{5}$$

가 성립한다.

앞에서의 정의에 의하여

$$S(x) = \left(\lim_{s \to \infty} \Phi^{s-j}(x) \right) V^j \tag{6}$$

이고,

$$\Phi^j(x) := \left(\lim_{s \to \infty} \Phi^{s-j}(x) \right) \tag{7}$$

로 정의하면

$$S(x) = \Phi^j(x)\, V^j \tag{8}$$

가 되며,

$$\Phi^j(x) = \Phi^{j+1} P^j \tag{9}$$

를 만족한다. 따라서 $\Phi^j(x)$가 이루는 벡터 공간은 내포 관계를 갖게 되고 이를 스케일 함수로 정의할 수 있다.

다중 해상도 분석에서 두 번째 단계는 내적을 정의하는 것이다. 3차원 메쉬에 대해서는 다음과 같이 정의할 수 있다.

$$\langle f, g \rangle := \int_{x \in M^0} f(x) g(x) dx \tag{10}$$

이제 마지막으로 웨이블렛을 정의한다. 스케일 함수 $\Phi^{j+1}(x)$는 $(O^{j+1} N^{j+1})$과 같은 블록 행렬로 나타낼 수 있다. 여기에서 O^{j+1}은 $\Phi^j(x)$에서 파생된 함수이고 N^{j+1}은 새로 발생된 부분으로 웨이블렛 $\Psi^j(x)$와 연관성을 갖는다. 따라서 N^{j+1}은 다음과 같이 구할 수 있다.

$$N^{j+1}(x) = \Psi^j(x) + \Phi^j(x)\, \alpha^j \tag{11}$$

양쪽에 $\Phi^j(x)$와 내적을 취하면 $\Psi^j(x)$는 직교하므로 없어지고 결과적으로

$$\begin{aligned} \langle \Phi^j(x), \Phi^j(x) \rangle \alpha^j &= \langle \Phi^j(x), N^{j+1}(x) \rangle \\ &= (P^j)^T \langle \Phi^{j+1}(x), N^{j+1}(x) \rangle \end{aligned} \tag{12}$$

와 같이 되고, 이 연립 방정식을 풀면 α^j를 구할 수 있다.

다중 해상도 분석을 합성 필터에 의해 표시하면

$$(\Phi^j(x)\ \Psi^j(x)) = \Phi^{j+1}(x) (P^j Q^j) \tag{13}$$

와 같이 나타낼 수 있고, 분석 필터는 역행렬 관계에 의해

$$\begin{pmatrix} A^j \\ B^j \end{pmatrix} = (P^j Q^j)^{-1} \tag{14}$$

과 같이 구할 수 있다. 이식을 앞에서 웨이블렛을 정의한 식에 대입하면

$$(P^j\ Q^j) = \begin{pmatrix} O^j & -O^j \alpha^j \\ N^j & 1 - N^j \alpha^j \end{pmatrix} \tag{15}$$

를 만족하고, 다면체 세분화(polyhedral subdivision)의 경우에는 O^j는 항등 행렬이므로

$$(P^j\ Q^j) = \begin{pmatrix} 1 & -\alpha^j \\ N^j & 1 - N^j \alpha^j \end{pmatrix}, \quad \begin{pmatrix} A^j \\ B^j \end{pmatrix} = \begin{pmatrix} 1 - \alpha^j N^j & \alpha^j \\ -N^j & 1 \end{pmatrix} \tag{16}$$

와 같이 구할 수 있다.

여기에서 구한 필터들을 이용해서 다중 해상도 분석을 표현하면 다음과 같다.

$$\begin{aligned} V^j &= A^j V^{j+1} \quad W^j = B^j V^{j+1} \\ V^{j+1} &= P^j V^j + Q^j W^j \end{aligned} \tag{17}$$

3. 세분화를 통한 정규 메쉬 생성

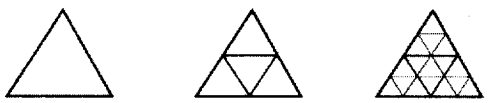
웨이블렛을 구하기 위해서는 세분화를 통해서 정규 메쉬를 생성하는 과정이 필요하다. 웨이블렛에 관련된 처리 과정은 필터 뱅크 분해(Filter bank decomposition), 선택(Selection), 복원(Reconstruction)의 세 과정으로 나눌 수 있는데 필터 뱅크 분해 과정은 이 절에서 설명하고 실제 웨이블렛으로 메쉬를 생성하는 선택과 복원 과정은 다음 절에서 설명하기로 한다.

필터 뱅크 분해 과정에서 정규 메쉬를 생성하기 위해서는 다음 두 과정을 반복적으로 거치게 된다. 하나의 삼각형을 네 개로 분할하는 분리(split) 과정과 새로 생긴 꼭지점들의 위치를 옮겨서 모델을 변형하는 변형(averaging) 과정으로 이루어진다.

3.1 분리 과정 (Split)

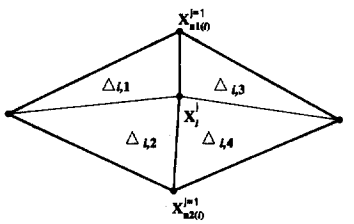
(그림 4)에서와 같이 하나의 삼각형을 네 개의 부삼

각형으로 분할하는 분리 과정에서는 각 변의 어떤 점으로 삼각형을 나눌 것인가가 문제가 된다. 그리고 나누는 기준이 되는 점을 정하면 그 점을 바탕으로 초기 메쉬의 각 삼각형에서 너비 우선 방문(Breadth first traversal) 방식으로 정규 메쉬를 생성하게 된다. 이 때 분리 과정은 각 단계에 상관없이 일정한 방법을 사용하게 되므로 그 결과를 저장할 필요가 없고 다음 과정인 변형 과정에서 추출되는 웨이블렛 만을 저장하면 되는 것이다.



(그림 4) 분리 과정

분리의 기준이 되는 점을 선정하는 방법은 다음의 두 가지가 있다. 파라메터 평균 추출법(Parametrically uniform resampling)과 기하학적 평균 추출법(Geometrically uniform resampling)이 있다.



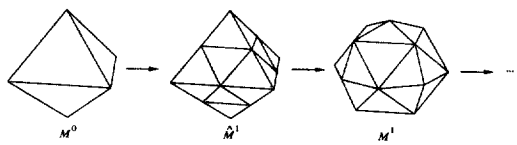
(그림 5) 기하학적 평균 추출법
(Geometrically uniform resampling)

파라메터 평균 추출법은 변의 중점을 기준점이 되게 하는 방법인데 삼각형이 정규적으로 분할된다는 단점이 있지만 속도도 빠르다. 기하학적 평균 추출법은 (그림 5)와 같이 원점 대칭이 되는 삼각형 두 개의 넓이의 합이 비슷해 지도록 기준점을 잡는 방법이다. 이 방법은 삼각형의 분할이 자연스럽지만 계산에 따라 속도가 느리다는 단점이 있다. 실제로 어느 방법을 써도 결과는 큰 차이가 없다는 결과가 있다. 본 시스템에서는 파라메터 평균 추출 방법을 사용하였다.

분리 과정에서 선행되어야 할 처리로써 인접한(adjacent) 삼각형의 정보를 구해야 한다. 인접한 삼각형이 이미 분리 과정을 거쳤다면 공유한 변에서 다음 단계의 삼각형의 꼭지점이 될 기준 점이 이미 구해졌고 그것을 반복할 필요가 없기 때문이다. 그리고 꼭지점과 웨이블렛을 중복되게 등록할 가능성을 제거하기 위해서도 인접한 삼각형의 정보를 구해야 한다.

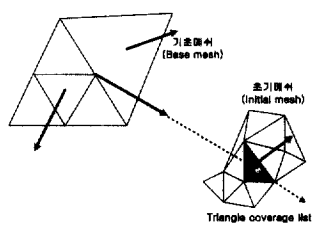
3.2 변형 과정(averaging)

분리 과정을 거친 후에는 새로 생성된 꼭지점을 이동시켜 주는 변형 과정이 필요하다. 이 때 주변의 삼각형들에 의한 가중치(weight)를 어떻게 고려하는가는 정의된 웨이블렛을 따르게 된다. 이 시스템에서는 가장 간단한 형태인 주변의 가중치를 최소로 하고 중심 꼭지점의 가중치를 최대로 하는 형태의 웨이블렛을 적용하였다.



(그림 6) 변형 과정(Averaging)

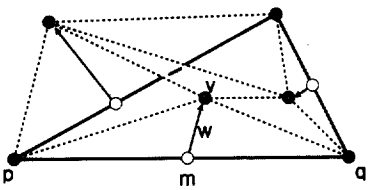
실제 꼭지점을 변형 시키는 방법은 (그림 6)에 나온 것처럼 구한다. 초기 메쉬의 삼각형과 교차하는 점을 구해서 그 점까지 변형시키는 것이다. 초기 메쉬에서 기초 메쉬를 구하는 과정에서 만든 삼각형 커버 리스트(Triangle coverage list)를 사용하여 초기 메쉬 전체를 탐색하는 것보다 빠르게 할 수 있다.



(그림 7) 삼각형 커버 리스트를 이용한 교차점 구하기

먼저 나누고자 하는 삼각형과, 옮기려는 꼭지점을 공유하는 인접한 삼각형의 법선 벡터를 구한다. 이 두 벡터의 평균을 구해서 이것과 평행하고 옮기려는 꼭지점을 지나는 직선의 방정식을 구한다. 이 직선과 삼각형 커버 리스트의 삼각형과 교차하는 지를 검사해서 교차하면 그것이 구하고자 하는 꼭지점이 변형될 위치가 된다.

검사하는 시간을 단축하기 위해서 삼각형 커버 리스트 상의 삼각형의 법선 벡터와 90도 이상 차이가 나면 그냥 제외하는 방법을 쓴다. 이렇게 구한 점과 원래 꼭지점의 차를 웨이블렛 계수로 저장하게 된다. 저장한 다음에는 (그림 7)과 같이 변경된 위치로 꼭지점을 이동해서 삼각형을 만들고 다음 단계로 넘어간다. (그림 8)에서와 같이 한 삼각형에서 웨이블렛 계수를 구할 때 인접한 삼각형에서 이미 구해졌으면 그것을 그대로 사용하면 된다. 꼭지점을 변경하는 방향이 인접한 삼각형의 법선 벡터와의 평균으로 정해졌기 때문이다.

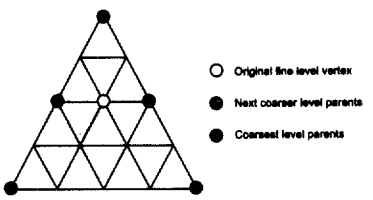


(그림 8) 웨이블렛 계수

4. 웨이블렛 변환을 이용한 LOD 메쉬 생성

분리 과정과 변형 과정으로 세분화를 해서 정규 메쉬를 생성한 결과로 웨이블렛을 구했으면 이제 이것으로 원하는 LOD(Level of detail)의 메쉬를 생성할 수가 있다. 이 과정은 먼저 기준치(Threshold)를 입력 받아서 그 이상이 되는 확장할 꼭지점들을 구하고 그것으로 삼각형들을 만들어서 메쉬를 만드는 알고리즘으로 구성된다.

4.1 꼭지점 확장(Expansion)

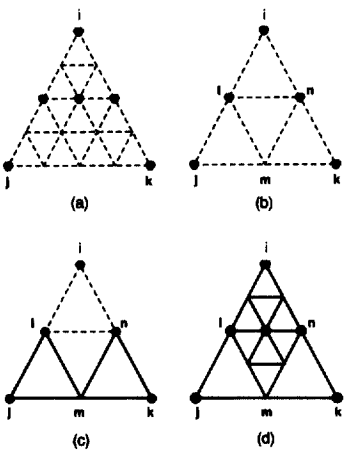


(그림 9) LOD 메쉬를 만드는 차례

기초 메쉬의 각 삼각형에 대해서 깊이 우선 방문(Depth first traversal) 형식으로 삼각형을 나눠가면서 웨이블렛 계수로 확장되는 꼭지점이 기준치 이상인지 비교하여 웨이블렛으로 확장을 할 것인지 말 것인지를 정한다. 이 때 부분적으로 확장된 메쉬에 웨이블렛을 적용하면 원래 모델에서 비틀어지게 되므로 기초 메쉬에서 웨이블렛을 적용하여 확장된 위치를 구하는 함수가 필요하게 된다. 기초 메쉬에 이 방법이 적용되는 차례는 (그림 9)와 같다.

먼저 확장될 꼭지점을 다 구한 다음에 그것을 바탕으로 삼각형을 만들 수도 있지만 저장하는데 메모리가 더 필요하게 된다. 이 시스템에서는 재귀적으로 삼각형을 분할(subdivision)하면서 확장될 꼭지점의 조합에 따라 삼각형을 다르게 만들어 가는 방법을 사용했다.

4.2 삼각형 생성(Triangulation)



(그림 10) 삼각형 생성(Triangulation)

확장될 꼭지점을 선택했으면 이제 그 꼭지점들을 확장하여서 삼각형을 생성해야 한다. 고려해야할 것은 한 삼각형의 모든 꼭지점이 확장되지 않을 경우도 있다는 것이다.

(그림 10)과 같은 경우를 살펴보자. (a)에서 확장될 꼭지점이 진한 점으로 표시되어 있다. (b)는 i,j,k 삼각형부터 재귀적 삼각형 생성 과정이 시작되고, l,n,m 꼭지점 중에 l,n 꼭지점이 확장될 것으로 판별되었다. 그래서 (c)에서 삼각형 ijk는 4개의 삼각형으로 나눠지게 되고 이 때 삼각형 생성 과정이 각 삼각형에 대해서 재귀적으로 불리게 된다. 삼각형 ljm과 nmk는 확장될 꼭지점을 포함하고 있지 않으므로 여기서 더 이상 나눠지지 않게 된다. m의 위치는 웨이블렛이 적용되지 않은 j와 k의 중점이 된다. (d)에서 l과 m의 중간에 확장될 꼭지점이 있으므로 이에 접한 삼각형들은 더 나눠지게

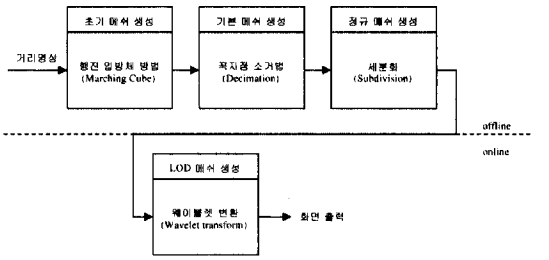
된다. 이 때 웨이블렛으로 확장된 꼭지점은 기초 메쉬부터 웨이블렛이 적용되어 만들어진 점이므로 삼각형 생성의 파라메터로 넘어온 위치에서 웨이블렛 계수를 적용하면 안 된다. 그래서 이러한 과정이 필요하다. 나머지 확장되지 않지만 삼각형 생성에 필요해서 생기는 꼭지점은 파라메터로 넘어온 위치로 계산을 해도 된다. (그림 10)에서 l과 m이나 n과 m 사이에 생성되는 꼭지점은 두 점의 중점을 계산하면 된다.

5. 프로그램구성 및 실험결과

5.1 프로그램 구성

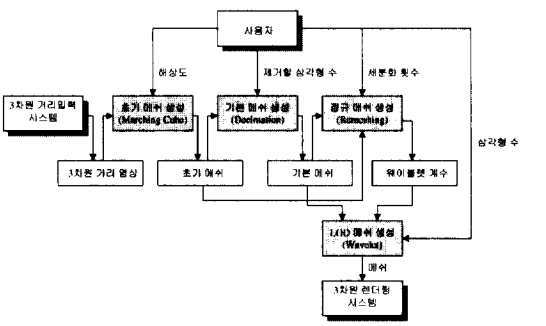
본 시스템은 Visual C++ 6.0을 이용하여 구현하였으며, 사용자 인터페이스는 MS Windows를 사용하였다. 3차원 렌더링을 위하여 OpenGL 그래픽 라이브러리를 사용하였으며, 개발한 3차원 객체생성 프로그램은 크게 4부분으로 구성된다. 입력된 거리 영상으로부터 초기 메쉬를 생성하는 부분이 첫 번째이고, 웨이블렛을 이용하여 임의의 상세 수준에 해당하는 메쉬를 생성하는 메쉬를 생성하는 부분이 두 번째이다. 그리고, 입력된 거리 영상 또는 생성된 메쉬를 몇 개의 의미단위로 분할하거나 몇 개의 객체를 계층적으로 모으는 작업이 있다. 웨이블렛을 적용하는 모듈은 다시 가장 단순한 형태의 기본 메쉬를 생성하는 "메쉬 간략화" 모듈과 기본 메쉬를 세밀하게 분할함으로써 "세분 연결성(subdivision connectivity)"을 만족하면서 초기 메쉬와 유사한 형태의 메쉬를 생성하는 모듈, 마지막으로 웨이블렛을 적용하는 모듈로 구성된다. 이렇게 하는 이유는 웨이블렛을 적용하기 위해서는 메쉬가 세분 연결성을 만족하여야 하는데 거리 영상으로부터 생성된 초기 메쉬는 이러한 성질을 만족하지 못하기 때문이다.

(그림 11)은 거리 영상을 입력으로 받아 LOD 메쉬를 생성하는 과정을 도식화 한 것이다. 네 개의 모듈로 구성되는데 이중 전반부의 세 모듈은 오프라인(offline)으로 처리되고 나머지 웨이블렛 변환은 온라인으로 처리된다. 이는 웨이블렛 상수들을 오프라인으로 구해서 저장해 놓으면 응용 프로그램에서 화면에 출력할 때에는 저장해 놓은 데이터를 사용하여 그때 그때 원하는 LOD의 메쉬를 생성할 수 있도록 개발하였다.



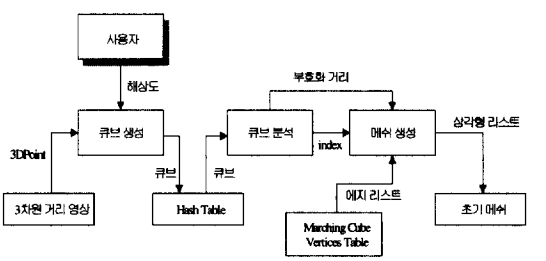
(그림 11) 프로그램 구성도

(그림 12)에서 볼 수 있듯이, 웨이블렛 변환도 웨이블렛 상수를 뽑는 부분과 이를 적용해서 LOD 메쉬를 생성하는 부분으로 구분되며, 웨이블렛 상수를 뽑는 부분은 오프라인으로 수행하고, 온라인으로 LOD 메쉬를 표현한다. (그림 12)는 (그림 11)의 프로그램 구성도를 좀 더 상세히 기술하여 프로그램 전체를 한 눈에 알아볼 수 있는 블록 다이어그램이다.



(그림 12) 프로그램 블록 다이어그램

5.2 초기메쉬 생성모듈



(그림 13) 초기메쉬 생성모듈 블록 다이어그램

(그림 13)은 구현된 초기 메쉬 생성 모듈의 동작 모

습을 보여주고 있다. 그림에서 왼쪽에 있는 입력자료가 3차원 거리 영상 자료이고, 오른쪽에 있는 출력자료가 초기메쉬이다.

임의의 3차원 거리 정보를 사용하여 초기 메쉬를 생성하기 위해서는 우선 거리 영상이 저장된 파일을 로드 한다. 일반적으로 거리 영상은 한 면의 정보만을 제공하는데 3차원 물체를 완전히 복원하기 위해서는 다양한 각도에서 측정한 거리 영상이 모두 필요하다[4].

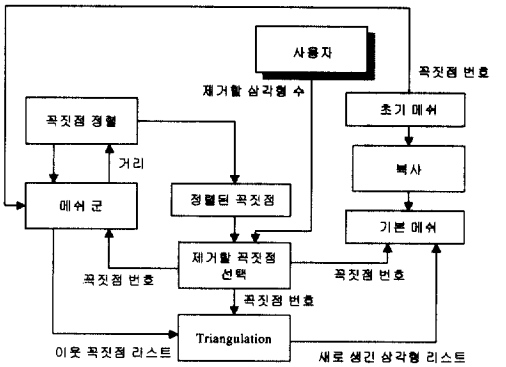
따라서 구현된 모듈에서도 이러한 기능을 제공하는데 각 면에서 측정한 거리 영상은 각각의 파일에 따로 저장된다. 여러 면에서 측정한 거리 영상을 합쳐서 모델을 만들기 위해서는 각각의 파일을 연속적으로 로드 하면 된다. 그리하면 메쉬생성 모듈에서 그 정보들을 기억하고 있다가 사용자가 "Create Mesh"의 메뉴를 선택하였을 때 각 방향에서 측정한 거리 영상을 차례로 적용하여 메쉬를 생성한다.

각 방향에서 측정된 거리 영상들이 로드 되면 점이 없는 부분은 측정값이 없는 부분이지만, 복원되는 메쉬에서 빈 공간을 없애기 위해서는 모델 내부에 빈 공간이 없어야 한다[5].

거리 영상이 모두 로드된 후에는 메쉬를 생성하게 되는데 이를 위해서는 "Create Mesh" 메뉴를 선택하면 된다. 그런데 이전에 해상도를 정의해 주어야 한다. 해상도는 본 연구에서 사용한 행진입방체 방법[6]에서 각 큐브들의 크기를 결정하는 파라미터이다. 거리 영상이 로드된 후에 프로그램에서는 입력된 값들을 둘러싸는 물체의 대략적인 크기를 계산한다. 이 정보를 이용하면 물체가 차지하는 3차원 공간을 구할 수 있는데 이 공간을 격자 형태로 분할 한 것이 큐브가 된다. 이때 몇 개의 격자를 사용할 것인지를 결정하는 것이 해상도이다. 즉 해상도를 256으로 결정하면 물체가 차지하는 공간을 256*256*256의 격자로 구분한다. 따라서 해상도가 커질수록 큐브의 크기가 작아지고 따라서 물체를 좀더 세밀하게 표현할 수 있게 된다. 그러나 이 경우에 세밀하게 표현될 수 있는 반면 측정값이 세밀하지 못한 경우에는 빈 공간이 많이 생길 가능성이 있다. 따라서 해상도는 측정값의 해상도를 넘을 수 없으며 실험에 의하면 측정값의 해상도가 메쉬 생성 해상도의 2배정도 되는 것이 적당하다.

5.3 기본메쉬 생성 모듈

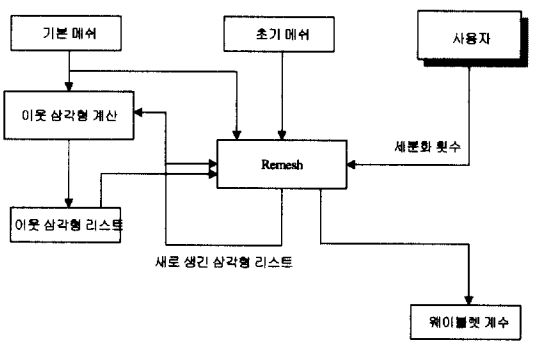
기본 메쉬를 생성하기 위해서는 사용자가 입력자료로



(그림 14) 기본메쉬 생성모듈 블록 다이어그램

제거할 삼각형의 개수 또는 소거량(decimation rate)을 입력하는 작업[7]으로 기본메쉬 생성모듈은 동작한다, 초기메쉬는 각 꼭지점들을 각각의 평면에서 평면까지의 거리에 따라 정렬하여 정렬된 꼭지점을 갖도록 한다. 가장 거리가 짧은 꼭지점을 선택하는 작업으로 꼭지점의 수를 줄인다. 각 꼭지점에 둘러싸인 이웃 꼭지점들을 분석하여 꼭지점의 형태를 분석하고 이웃 꼭지점 리스트, 평균평면, 평균평면까지의 거리등을 계산하는 모듈인 메쉬군을 거치면 이웃하는 꼭지점 리스트를 얻을 수 있으며, 이를 이용하여 삼각 분할을 거치면 새로운 기본 메쉬가 만들어진다. 이를 초기메쉬와 비교 분석하기 위하여 한 화면에 동시에 출력시킨다[8].

5.4 정규메쉬(Remesh) 생성 모듈

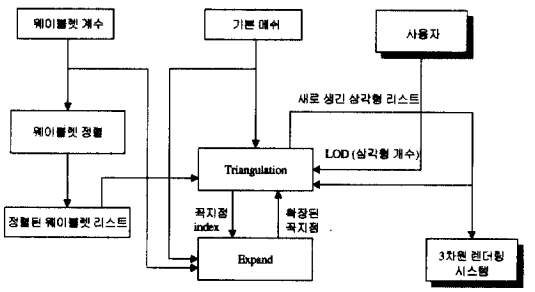


(그림 15) 정규메쉬 생성모듈 블록 다이어그램

정규메쉬 생성모듈은 사용자가 세분화 횟수를 지정하면서 모듈이 구동한다. 기본메쉬 생성모듈에 의하여 얻

어진 기본메쉬를 근간으로 이웃하는 삼각형의 상태를 계산하면서 사용자가 지정한 세분화 횟수만큼 메쉬를 소거하면서 정규화 작업[9]을 반복 수행한다. 이때, 얻어지는 웨이블렛계수는 웨이블렛에 의한 다중메쉬 생성 모듈에서 사용한다.

5.5 다중메쉬 생성 모듈



(그림 16) 다중메쉬 생성모듈 블록 다이어그램

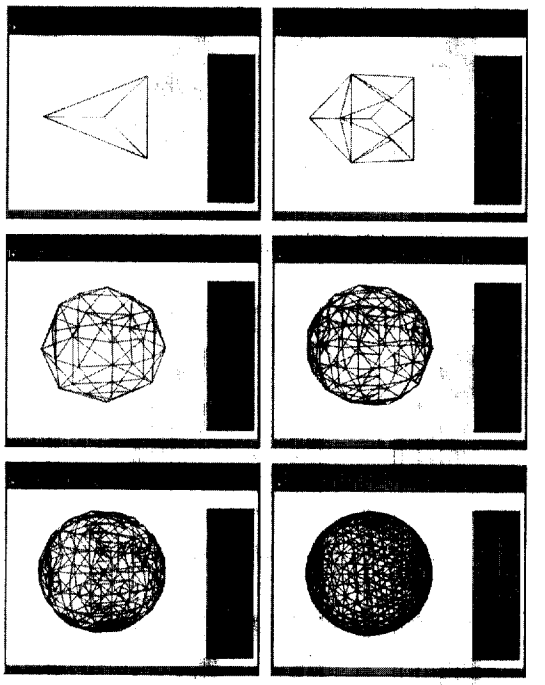
다중메쉬 생성 모듈은 정규메쉬 생성 모듈에서 얻어진 웨이블렛 계수를 이용하여, 실시간으로 LOD 다중 메쉬를 얻어내는 모듈이다.

웨이블렛 계수를 크기 순으로 정열하여, 정렬된 웨이블릿 리스트를 만들어 내고, 사용자가 입력한 수만큼의 웨이블릿 계수를 뽑아서 메쉬를 구성하고, 삼각형을 더 분할해야 할 필요가 있을 경우 이를 더 분할한다. 그리고, 실시간으로 화면에 출력하여 다중 메쉬의 변화를 비교 검토할 수 있도록 개발하였다.

5.6 실험결과

앞 장에서 기술하였듯이 오프라인(offline)으로 세분화 과정에서 추출한 웨이블렛 계수로 삼각형을 생성한 결과가 (그림 17)의 첫 번째 그림이다. 이 상태가 LOD 최소 상태인 기초 메쉬 형태이며, 이 상태에서 LOD를 점차적으로 높혀 완벽한 구의 형태를 갖음을 볼수 있다.

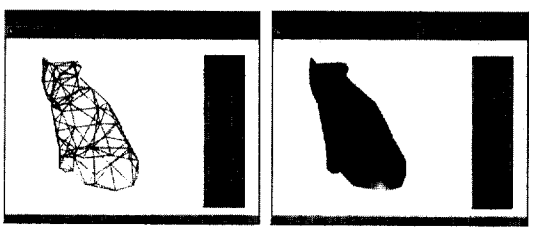
본 LOD 객체표현 시스템은 실시간으로 이루어지지만, 이와같은 실시간 표현을 위해서 웨이블렛 계수를 생성하는데 소요되는 작업시간은 입력되는 자료에 따라 증가하며, 본 웨이블렛 계수 생성작업에서 기초메쉬, 기본메쉬, 정규메쉬등을 함께 생성하면서 진행되기 때문에 정확한 웨이블렛 계수 생성시간을 추출하기 불가능하며, 또한 웨이블렛 계수 추출은 오프라인으로 이루어지기 때문에 본 LOD 객체표현 시스템 성능을 평가하는 중



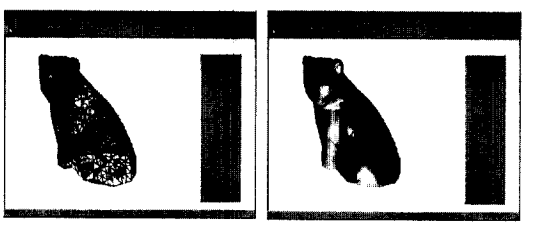
(그림 17) 구의 실험결과

요한 척도로 활용되지 않는다.

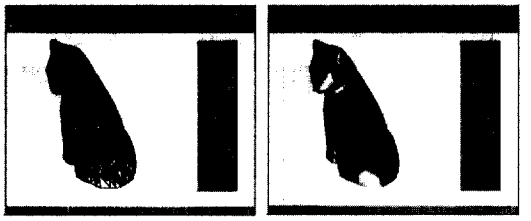
실물기반 모델에 5단계로 0%, 20%, 40%, 60%, 100%로 LOD를 갖도록 하여, wireframe model과 shading model로 실시간으로 표현함을 보였다(그림 18)~(그림 22).



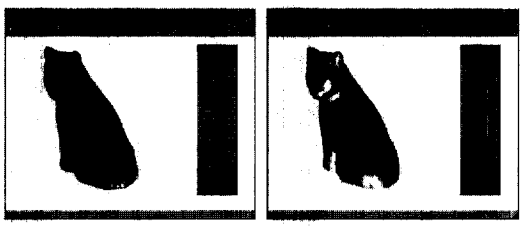
(그림 18) 고양이 실험결과(LOD 0%)



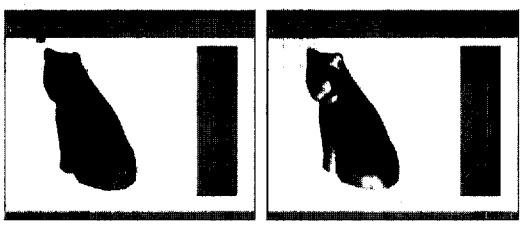
(그림 19) 고양이 실험결과(LOD 20%)



(그림 20) 고양이 실험결과(LOD 40%)



(그림 21) 고양이 실험결과(LOD 60%)



(그림 22) 고양이 실험결과(LOD 100%)

6. 결 론

웨이블릿 기반으로 LOD 메쉬를 표현함에 있어서 (그림 18)과 같이 가장 적은 량의 메쉬로 표현되는 가상물체의 경우에는 텍스추어 맵핑등의 기법을 추가적으로 활용하여야만 실물과 비슷한 효과를 갖을 수 있을 것이다.

그리고, 웨이블릿 계수를 획득하기 위해서 소요되는 컴퓨팅 처리시간을 줄이기 위한 연구가 계속 진행되어, 향후 다중메쉬 실 시간 생성 장비의 개발이 이루워져야, 실시간으로 LOD 메쉬를 사용할 수 있도록 발전되어야 할 것이다.

참 고 문 헌

[1] Ki-Ho KIM, Hwang-Bin RYOU, "3-D Level-of-Detail Object Constructed Based on Decimation," KIPS(Korea Information Processing Society) April 1998.

[2] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle, "Multiresolution Analysis of Arbitrary Meshes," Computer Graphics (SIGGRAPH 95 Proceedings), pp.173-182, August 1995.

[3] Eric J. Stollnitz, Tony D. DeRose, David H. Salesin, "Wavelets for Computer Graphics : Theory and Applications," Morgan Kaufmann Publishers, 1996.

[4] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle, "Surface Reconstruction from Unorganized Points," Computer Graphics (SIGGRAPH 92 Proceedings), 26(2):71-78, July 1992.

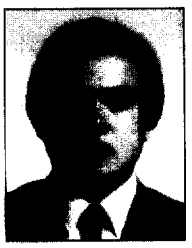
[5] Greg Turk and Mark Levoy, "Zippered Polygon Meshes from Range Images," Computer Graphics (SIGGRAPH 94 Proceedings), 1994.

[6] William E. Lorensen and Harvey E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," Computer Graphics (SIGGRAPH 87 Proceedings), 21(3):163-169, July 1987.

[7] William J. Schroeder, Jonathan A. Zarge and William E. Lorensen, "Decimation of Triangle Meshes," Computer Graphics (SIGGRAPH 92 Proceedings), 26(2): 65-70, July 1992.

[8] Algorri, M.-E. and Schmitt, F., "Mesh simplification," In Proceedings of EuroGraphics'96, Computer Graphics Forum, Vol.15, pp.77-86, Futuroscope, Poitiers, France. 1996.

[9] Paul Hinker and Charles Hansen, "Geometric Optimization," Proceedings of Visualization, pp.189-195, October 1993.



김 기 호

e-maill : khkim@etri.re.kr

1990년 광운대학교 전자계산학과 공학석사취득

1996년 광운대학교 전자계산학과 박사과정수료

1998년~현재 한국전자통신연구원 선임연구원

관심분야 : Medical VR, 입체영상, 입체음향, 실물모델링, 멀티미디어통신 및 응용, 네트워크 보안, 네트워크 게임

유 황 빈

e-maill : ryou@cs.kwangwoon.ac.kr

1975년 인하대학교 전자공학과 학사취득

1977년 연세대학교 대학원 공학석사취득

1989년 경희대학교 대학원 공학박사취득

1981년~현재 광운대학교 컴퓨터과학과 교수

1994년~1995년 美 UCSD 교환교수

1995년~1997년 광운대학교 전자계산소장

2000년~현재 광운대학교 전자계산원장

관심분야 : 멀티미디어통신 및 응용, 네트워크 보안