스펙트랄 분해를 응용한 폴리곤 메쉬의 워터마킹 *

손정교*, 최형인*, 김태완**, 권성화***, 박상훈*, 신헌주°

*서울대학교 수리과학부

**서울대학교 조선해양공학과

***서울대학교 복잡계통계 연구소

°서울대학교 협동과정 계산과학전공

{jgsohn, hichoi, taewan, shkwon, kitewin, shj1}@snu.ac.kr

A Digital Watermarking Scheme for Polygonal Meshes using Modified Spectral Decomposition

Jung-Kyo Sohn*, Hyeong-In Choi*, Tae-wan Kim**, Song-Hwa Kwon***, Sang-Hun Park*, Heon-Ju Shin°

*Dept. of Mathematical Science, Seoul National University, Korea

**Dept. of Naval Architecture and Ocean Engineering, Seoul National University, Korea

***Statistical Research Center for Complex Systems, Seoul National University, Korea

°Interdisciplinary Program in Computational Science and Technology, Seoul National University, Korea

2004년 11월 20일

요 약

그래픽, 오디오, 텍스트 정보 등의 저작권 보호를 위해 사용되어오던 디지털 워터마킹은 이제 기하 정보의 저작권을 보호하는데에도 중요하게 쓰이게 되었다. 스펙트랄 영역에서의 다각형 모델에 대한 워터마킹은 여러 가지 공격에 대하여 강건함을 갖는다. 우리는 기존의 스펙트랄 영역에서의 워터마킹 기법을 일반화하고 이에 기반한 새로운 방법을 제시한다. 워터마크를 삽입 할 때에 라플라시안 행렬의 고유값과 고유벡터를 찾지 않는다. 대신 스케일링 함수로 부터 얻어진 선형작용소를 사용한다. 테스트 결과는 서로 상충되는 워터마크를 삽입하고 추출하는데 드는 비용과 공격에 대한 강건성을 어떻게 조정하였는지 보여준다.

1. 서 론

컴퓨터의 성능이 빠르게 발전하고 초고속통신망의 보급이 이루어짐에 따라 음성, 영상, 비디오 등과 같은 콘텐츠 데이터들이 디지털화 되고 있다. 이러한 디지털 데이터는 기존의 아날로그 데이터와 비교하여 저장과 편집이 쉽다는 장점이 있으나, 디지털콘텐츠를 복사하여 인터넷과 같은 통신매체를 통해 유통이 손쉽게 이루어짐에 따라 소유권의 보호문제가 심각하게 대두되고 있다 [15] [7].

디지털 워터마킹(Digital Watermarking)이란 영상이나 음성 매체를 사람이 인식하지 못할 정도로 미세하게 변화시켜 저작권 정보(mark)를 삽입하고 필요할 때 삽입된 정보를 추출하는 방법으로서 디지털콘텐츠의 저작권 보호문제를 해결하기 위한 하나의 기술적인 해결책이다.

수년동안 이미지, 애니메이션, 음악과 같은 콘텐츠에 대한 워터마킹 기술이 발전되어 왔다. 그러나 곡선이나 곡면과 같은기하학적 오브젝트로 구성된 2차원, 3차원 기하학적 모델에 대한 연구는 많지 않았다. 최근에서야, 이 분야에 대한 연구가 많은 학자들에 의해 시작되었다. 특별히 3차원 게임의 대중화와 인터넷 상에서의 협동적 CAD 시스템이 증가함에 따라, 3차원 폴리곤 모델과 CAD 모델에 대한 디지털 워터마킹 기술이 몇 명의 연구자들에 의해 연구되기 시작했다. Ohbuchi et al. [12] [10] [13] [11], Bredens [3], Wagner [19], Praun et al. [16] 는 3차원 폴리곤 모델에 워터마크를 삽입하는 방법을 제시하였다. Ko et al. [8]은 최근에 2개의 NURBS 곡면을 서로 매칭하여 유사성을 찾는 방법을 제시하였다. 그리고 이를 응용하여 NURBS곡면의 저작권을 보호할 수 있는 방법을 소개하였다. 조남익 et al. [9] 은 3차원 NURBS 데이터에서 2차원 가상 이미지 정보를 만들어 내어 기존의 2차원 워터마킹 방법을 사용하는 워터마킹 방법을 제시하였다. Daniel Cotting et al. [5] 은 샘플링된 점군 기하 모델에 스펙트랄 분석을 통하여 워터마크를 삽입하는 방법을 제시하였다.

Ohbuchi et al. [14]은 3차원 메쉬 모델의 공간영역 대신에 주파수 영역에 워터마크를 삽입하는 방법을 제시하였다. 메쉬의 공간 영역에서 주파수 영역으로 가기 위해서, 우리는 먼저 메쉬의 주파수 영역이 무엇인지를 먼저 정의하여야 한다. 메쉬의 주파수 영역은 라플라시안 행렬 [4] 이라고 부르는 어떤 행렬의 고유치와 그에 대응되는 고유벡터들에 의해 그 특성이 기술될 수 있다. 주어진 메쉬의 라플라시안 행렬은 그 메쉬의 위상적 성질을 포함하고 있다. 다시 말하면, 각

*본 연구는 한국과학재단 목적기초연구 R01-2001-00396 지원으로 수행되었습니다.

메쉬 에 대하여 행렬 를 대응시킬 수 있는데, 이 행렬은 그 고유치와 대응되는 고유벡터들에 메쉬의 스펙트랄 정보를 담고 있다. 그러나 고유치 문제는 그 행렬이 대칭이든 아니든 상관없이 불안정할 수가 있다 [2] [6]. 행렬의 고유치와 고유벡터를 찾는 문제는 차 방정식의 모든 해를 구하는 것이므로 일반적으로 ill-posed 문제이다. 특별히 고유벡터의 안정성은 고유치들이 서로 얼마나 가까이 있느냐에 크게 의존한다. 그러므로 두 개의 고유치가 서로 가까이 있다면, 행렬이 약간만 변하더라도 고유벡터는 크게 변할 수 있다는 것이다. 우리는 특정한 메쉬에 독립적인 워터마킹 기법을 고안하려고 하기 때문에, 우리가 다룰 모든 메쉬에 대한 라플라시안 행렬의 고유치 분포가 매우 좋으리라고 가정하는 것은 위험하다.

Taubin [18]은 단순한 선형 작용소를 사용하여 메쉬를 부드럽게 만드는 방법을 제시하였다. 이 선형 작용소는 행렬 에 대한 다항식이었다. 메쉬 신호에 이 작용소를 여러번 적용하면, 메쉬 신호에서 고주파를 가지는 부분들의 세기를 점차적으로 작게 만들 수 있다. 메쉬 신호들은 메쉬 의 공간 영역에 속한다. Taubin 의 아이디어의 핵심은 공간 영역에 적절한 선형 작용소를 적용시켜서 마치 메쉬 신호의 스펙트랄 계수를 직접 변화시킨 것과 동일한 효과를 얻을 수 있다는데 있다.

우리는 Taubin의 아이디어를 따라 워터마킹 기법을 고안하고자 한다. 먼저 Ohbuchi et al. [14]에서 기술되어 있는 방법과 같은 기존의 워터마킹 기법을 일반화 하고, 고유치와 고유벡터 문제에서 발생할 수 있는 안정성 문제를 피할 수 있는 여지를 만들 것이다. 그리고 Ohbuchi et al. [14]에 기술되어 있는 방법은 이 일반화된 방법의 특별한 예라는 것을 보일 것이다. 우리는 이러한 일반화의 틀 아래에서 새로운 기법을 제시한다.

우리의 새로운 기법은 메쉬 M의 라플라시안 행렬 Δ의 다항식으로 표현되는 선형작용소를 사용한다. 또한 선형작용소를 미리 구할 필요 없이 연속된 행렬과 벡터의 곱을 통하여 선형작용소와 신호 벡터의 곱을 얻을 수 있다. 행렬과 벡터의 곱, 그리고 스칼라와 벡터의 곱과 같은 행렬과 벡터에 대한 기본 연산을 적절히만 한다면, 빠른 시간안에 안정적으로 수행될 수 있다.

이 연구결과의 나머지 부분은 다음과 같이 구성되어있다. 2장에서는 메쉬 모델에 대한 신호 처리 모델을 소개한다. 그리고 기존의 워터마킹 기법(스펙트랄 영역에 워터마크를 삽입하는)을 일반화 한다. 3장에서는 2장에서의 일반화에 기반하여 스케일링 함수를 하용하는 새로운 워터마킹 기법을 소개한다. 4장에서는 테스트 결과를 제시한다. 끝으로 5장에서는 결론을 내린 후 향후 연구 방향을 제시한다.

2. 개요

2.1 폴리곤 모델에 대한 신호처리 모델

메쉬 M이 주어져 있다고 가정하자. 우리는 M을 추상적으로 그래프 구조를 가지고 있는 것으로 이해할 수 있다. 즉, $G_M = (V, E)$,으로 볼 수 있는데 $V = \{v_1, v_2, \ldots, v_N\}$은 꼭지점 집합이고 $E = \{e_{ij} = (v_i, v_j)\}$는 메쉬의 모서리 집합이다. 모서리 집합 E와 꼭지점 집합 V는 메쉬 M의 연결 정보(위상적 정보)를 담고 있고, 이들로부터 우리는 라플라시안 작용소(Laplacian operator)를 유도해 낼 수 있다. 이 작용소는 나중에 자세히 설명하기로 한다. 집합 $S = \{s : V \to \mathbb{R}\}$는 메쉬의 꼭지점 집합 V위에서 정의되어 있는 실함수들의 집합이라고 하자. 고정된 꼭지점 집합 V에 대하여, 우리는 각 함수 $s \in S$를 벡터 $\mathbf{s} = (s(v_1), s(v_2), \ldots, s(v_N))^T \in \mathbb{R}^N$으로 볼 수 있다. 이 벡터 $\mathbf{s}$를 메쉬 M의 (공간영역의) 신호벡터라고 부르자. 만약 꼭지점 $v \in V$가 3차원 공간에 구현되어 좌표 (x, y, z)를 가질 때, $x(v), y(v), z(v)$는 S에 있는 함수들로서 이것들은 각 $v \in V$은 x, y, z으로 각각 보낸다. 이 신호 함수들의 벡터 형태(신호벡터)를 각각 $\mathbf{x}, \mathbf{y}, \mathbf{z}$로 표시하자.

3. 일반화된 워터마킹 기법

3.1 항등 작용소의 분할

주어진 신호 벡터 $\mathbf{x}$를 다음과 같이 분할한다고 가정하자.

$$\mathbf{x} = c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + \cdots + c_n\mathbf{x}_n, \ (c_i \in \mathbb{R}, \|\mathbf{x}_i\| = 1), \qquad (1)$$

여기서 이 분할은 다음과 같은 조건을 만족한다고 하자.

1. 선형 작용소들의 집합 $\{L_1, L_2, \cdots, L_n\}$이 존재하여 $L_i(\mathbf{x}) = c_i\mathbf{x}_i$, $i = 1, 2, \ldots, n$이다,
2. $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$은 일차독립이다,
3. $i \neq j$이면 충분히 작은 $\epsilon \geq 0$에 대하여 $|\langle \mathbf{x}_i, \mathbf{x}_j \rangle| \leq \epsilon$을 만족한다,
4. $i < j$에 대하여 $\mathbf{x}_j$는 $\mathbf{x}_i$보다 $\mathbf{x}$의 더 높은 주파수에 해당되는 부분이다.

만약 집합 $\{L_1, L_2, \cdots, L_n\}$이 $\mathbf{x}$에 완전히 독립적으로 선택될 수 있다면, 우리는 집합 $\{L_1, L_2, \cdots, L_n\}$을 항등 작용소에 대한 분할이라고 부르자. 왜냐하면,

$$L_1 + L_2 + \cdots + L_n = I.$$

그러나 항등 작용소의 분할은 여전히 메쉬 모델 M의 위상적 구조에 의존한다는 것에 주의하자.

만약 $0 \leq \epsilon \leq 1/n$.이라면조건 2는 조건 3으로부터 유도됨을 증명할 수 있다. 조건 4는 갖가지 스무딩 공격(smoothing attack)을 고려하기 위하여 도입하였다. 조건 3을 도입함으로써 우리의 분석을 그렇지 않을 때보다 더욱 쉽게 만들어 주지만, 실험을 해보면 조건 3은 우리의 워터마킹 기법을 위해서 약간은 완화될 수 있다.

주어진 신호 벡터 $\mathbf{x}$를 계수들의 벡터 $(c_1, c_2, \ldots, c_n)$으로 보내는 사상은 일반화된 푸리에(Fourier) 변환으로 생각할 수 있고, $c_1, c_2, \ldots, c_n$는 의 푸리에 계수라고 생각할 수 있다. 우리의 기법에서 c_i들을 직접 계산할 필요는 없다. 우리의 목적을 달성하기 위하여, 작용소 $L_1, L_2, \cdots, L_n$만 있으면 된다.

3.2 워터마크의 삽입과 추출

$\mathbf{p} = (p_j)_{j=1,2,\ldots,n} \in \{-1, 1\}^N$를 pseudo-random 수열이라고 하자. 주어진 신호 벡터 $\mathbf{x}$에 대하여 $\mathbf{x}$가 1과 같이 분할 되었다고 가정하자. 워터마크 비트 α (1 또는 -1)를 삽입하기 위하여 워터마크 벡터를 다음과 같이 정의하자.

$$\mathbf{w} = \sum_{j=1}^{n} (\alpha \cdot \beta \cdot p_j)\mathbf{x}_j,$$

여기서 β는 크기를 조절하기 위해 도입된 파라미터이다. 그 다음 워터마크 벡터 $\mathbf{w}$를 원본 신호벡터 에 더해준 다음 그 결과를 발표한다.

$$\mathbf{x}^w = \mathbf{x} + \mathbf{w}. \tag{2}$$

워터마크를 없애기 위한 공격을 받은 것으로 의심되는 신호 벡터 $\mathbf{x}^a$가 주어졌다고 하자. 이 벡터 $\mathbf{x}^a$에서 워터마크 벡터 $\mathbf{x}^w$를 빼고 난 나머지를 공격 신호벡터 $\mathbf{a}$라고 표시하기로 하자. 워터마크 비트 α에 대한 워터마크 검출 값 $W(\mathbf{x}, \alpha)$를 다음과 같이 정의하자.

$$W(\mathbf{x}, \alpha) = \left\langle \mathbf{x}^a - \mathbf{x}, \sum_{j=1}^{n} p_j \mathbf{x}_j \right\rangle.$$

$W(\mathbf{x}, \alpha)$의 부호는 삽입된 워터마크 비트 α를 완전하게 결정한다. 이 값에 대한 식을 좀더 정리하면 다음과 같이 된다.

$$\begin{aligned} & \left\langle \mathbf{x}^a - \mathbf{x}, \sum_{j=1}^{n} p_j \mathbf{x}_j \right\rangle \\ = & \left\langle \sum_{j=1}^{n} (\alpha \cdot \beta \cdot p_j \mathbf{x}_j + \mathbf{a}), \sum_{j=1}^{n} p_j \mathbf{x}_j \right\rangle \\ = & \alpha \cdot \beta \left\langle \sum_{j=1}^{n} p_j \mathbf{x}_j, \sum_{j=1}^{n} p_j \mathbf{x}_j \right\rangle + \left\langle \mathbf{a}, \sum_{j=1}^{n} p_j \mathbf{x}_j \right\rangle \end{aligned}$$

오른편의 첫 번째 항은 다음과 같은 부등식을 가진다.

$$\begin{aligned} & \left| \left\langle \sum_{j=1}^{n} p_j \mathbf{x}_j, \sum_{j=1}^{n} p_j \mathbf{x}_j \right\rangle_D \right| \\ = & \left| n + \sum_{i \neq j} p_i p_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \right| \\ \geq & n - \epsilon(n^2 - n). \end{aligned}$$

그리고 두 번째 항은 다음과 같은 부등식을 가진다.

$$\begin{aligned} & \left| \left\langle \mathbf{a}, \sum_{j=1}^{n} p_j \mathbf{x}_j \right\rangle \right| \\ = & \left| \left\langle \sum_{j=1}^{n} a_j \mathbf{x}_j, \sum_{j=1}^{n} p_j \mathbf{x}_j \right\rangle \right| \\ \leq & \left| \sum_{j=1}^{n} p_j a_j \right| + \epsilon(n-1)\|\mathbf{a}\|_1, \end{aligned}$$

여기서 $\|\cdot\|_1$은 일반적인 1-norm 이다. Pseudo-random 수열의 성질에 의하여 대부분의 공격 벡터 $\mathbf{a} = \sum_{i=1}^{n} a_i \mathbf{x}_i$에 대하여, 합 $\sum_{j=1}^{n} p_j a_j$은 n보다 훨씬 더 작아지리라 예상할 수 있다. 더 작은 ϵ을 택할수록 신호 벡터 에 삽입된 워터마크 비트를 정확히 검출할 확률은 더욱 높아지게 된다. 왜냐하면 첫 번째 항이 $W(\mathbf{x}, \alpha)$의 값을 결정할 가능성이 더욱 높아지기 때문이다. 만약 우리가 벡터 $\mathbf{x}_j$를 라플라시안 행렬 Δ의 고유벡터 ϕ_j로 뒀다면 우리의 방법이 [14]에 기술되어 있는 기법과 완전히 동일한 결과를 얻는다는 것을 알 수 있다. 이제 $\mathbf{q} = (q_1, q_2, \ldots, q_n)$를 또다른 pseudo-random 수열로서

$$\sum_{j=1}^{n} p_j q_j = 0.$$

을 만족한다고 하자. 그러면 내적 $\langle \sum p_j \mathbf{x}_j, \sum q_j \mathbf{x}_j \rangle$의 값은 다음과 같이 계산될 수 있다.

$$\begin{aligned} & \left| \left\langle \sum_{j=1}^{n} p_j \mathbf{x}_j, \sum_{j=1}^{n} q_j \mathbf{x}_j \right\rangle \right| \\ \leq & \left| \sum_{j=1}^{n} p_j q_j \right| + \epsilon(n^2 - n) = \epsilon(n^2 - n). \end{aligned}$$

따라서 만약 우리가 $\mathbf{x}$에 k비트 $(\alpha_1, \alpha_2, \ldots, \alpha_k)$를 심었다면, 워터마크 검출 값 $W(\mathbf{x}, \alpha_i)$은 다음과 같이 계산된다.

$$\begin{aligned} |W(\mathbf{x}, \alpha_i)| &= \left| \left\langle \mathbf{x}^a - \mathbf{x}, \sum_{j=1}^{n} p_j^{(i)} \mathbf{x}_j \right\rangle \right| \\ &\geq n - \left| \sum_{j=1}^{n} p_j^{(i)} a_j \right| - \epsilon((n-1)\|\mathbf{a}\|_1 + k(n^2 - n)), \end{aligned} \tag{3}$$

여기서 $\mathbf{p}^{(i)} = (p_1^{(i)}, p_2^{(i)}, \ldots, p_n^{(i)})$는 $\mathbf{x}$에 비트 α_i를 삽입할 때 사용되는 pseudo-random 수열이다. 부등식 3은 어떤 의미에서 정밀하지 않은 부등식이라고 볼 수 있다. 부등식 3에서 등식이 성립한다고 하더라도 우리가 ϵ이 $1/kn^2$보다 작게 되는 $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$를 택한다면 비트 수열 $(\alpha_1, \alpha_2, \ldots, \alpha_k)$를 얻을 가능성은 매우 높다.

4. 새로운 워터마킹 기법

4.1 라플라시안 행렬

곡면 신호 집합 위에 정의된 작용소 이 다음과 같은 식을 만족한다면 을 라플라시안 작용소라고 부르자.

$$\mathcal{L}s(v_i) = \sum_{(v_i, v_j) \in E} w_{ij}(s(v_j) - s(v_i)), \; s \in S, \tag{4}$$

여기서 가중치 w_{ij}는 음이 아닌 실수이고 모든 $i = 1, 2, \ldots, N$에 대하여 $\sum_{(v_i, v_j) \in E} w_{ij} = 1$ 을 만족한다. 4의 벡터 표현은 다음과 같이 주어진다.

$$\Delta \mathbf{s} = (I - W)\mathbf{s}, \tag{5}$$

여기서 $W = (w_{ij})_{1 \leq i,j \leq N} \in \mathbb{R}^{N^2}$이고 Δ는 $N \times N$행렬이다. $\mathcal{L}$에 대한 행렬 표현 Δ는 라플라시안 행렬(Laplacian Matrix)이라고 부른다. $\mathbf{s}$는 임의의 벡터이므로 Δ는

$$\Delta = I - W.$$

만약 우리가 꼭지점 $v \in V$의 valence를 $d(v)$라고 표시하기로 할 때, 가중치는 $(v_i, v_j) \in E$인 경우에 $w_{ij} = 1/d(v_i)$, 그 외의 경우에는 0으로 정의된다. 일반적으로 W는 대칭이 아니기 때문에 우리가 라플라시안 행렬 Δ의 고유치는 모두 실수라고 할 수는 없다. 가중치 행렬 W를 우리가 제시할 방식으로 둔다면 라플라시안 행렬 Δ의 스펙트럼 $\sigma(\Delta)$은 실수 공간 $\mathbb{R}$속에 들어가게 된다. 이제 $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$을 행렬 Δ의 N개(서로 다를 필요는 없다)의 고유치라고 하고 $\Phi = \{\phi_1, \phi_2, \ldots, \phi_n\}$를 대응되는 고유벡터라고 하자.

여기서 몇가지 유용한성질은 다음과 같다.

- $\Lambda \subset [0, 2]$.
- $\lambda_1 + \lambda_2 + \cdots + \lambda_n = n$.
- 첫 번째 고유치와 첫 번재 고유벡터는 $\lambda_1 = 0$, and $\phi_1 = 1 = (1, 1, \ldots, 1)^T$.와 같다.

비록 Φ는 일차 독립이고 따라서 벡터공간 $\mathbb{R}^N$의 기저이지만, 내적공간 $\mathbb{R}^N$의 표준 내적 $\langle \cdot, \cdot \rangle$에 대하여 직교기저를 이루지 못한다. 하지만 이 표준 내적을 약간만 수정을 하면, 직교기저가 되도록 할 수 있다. 다음과 같이 새로운 내적을 정의하자.

$$\langle \mathbf{x}, \mathbf{y} \rangle_D = \mathbf{x}^T D \mathbf{y}, \ \mathbf{x}, \mathbf{y} \in \mathbb{R}^N, \tag{6}$$

여기서 D는 대각 행렬로서 그 대각 원소들은 $1/d(v_j), j = 1, 2, \ldots, N$로 이루어져 있다. 이 새로운 내적 $\langle \cdot, \cdot \rangle_D$으로부터 norm $\| \cdot \|_D$이 유도된다.

$$\|\mathbf{x}\|_D = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle_D}$$

. 이제 다음과 같은 보조정리를 얻는다.

Lemma 4.1 *라플라시안행렬 Δ는 위에서 정의한 내적 $\langle \cdot, \cdot \rangle_D$에 대하여 자기수반작용소이다. 즉 임의의 $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$에 대하여*

$$\langle \mathbf{x}, \Delta \mathbf{y} \rangle_D = \langle \Delta \mathbf{x}, \mathbf{y} \rangle_D. \tag{7}$$

따라서 벡터 공간 $\mathbb{R}^N$에 새로운 내적 $\langle \cdot, \cdot \rangle_D$을 부여하면 다음과 같은 정리를 얻는다.

Theorem 4.2 *라플라시안 행렬 Δ의 고유벡터들의 집합 $\Phi = \{\phi_j | j = 1, 2, \ldots, N\}$은 벡터공간 $\mathbb{R}^N$의 직교기저(표준화하면 정규직교기저)를 이룬다.*

정리 4.2에 의하여 임의의 벡터 $\mathbf{x} \in \mathbb{R}^N$는 다음과 같이 Φ의 원소들의 일차결합으로 유일하게 표현될 수 있다.

$$\begin{aligned} \mathbf{x} &= \mathbf{x}_1 + \mathbf{x}_2 + \cdots + \mathbf{x}_N \\ &= \langle \mathbf{x}, \phi_1 \rangle_D \phi_1 + \langle \mathbf{x}, \phi_2 \rangle_D \phi_2 + \cdots + \langle \mathbf{x}, \phi_N \rangle_D \phi_N, \end{aligned}$$

여기서 $\mathbf{x}_i = \langle \mathbf{x}, \phi_i \rangle_D \phi_i, i = 1, 2, \ldots, N$이다. 이 분해는 분명히 앞에서 제시한 조건 2, 3, 4를 만족한다. 다음 정리에 의하여, 이 분해는 조건 1도 만족함을 알 수 있다.

Theorem 4.3 (스펙트랄 정리) *$i (1 \leq i \leq N)$, W_i에 대하여 W_i를 고유치 λ_i에 대응되는 Δ의 고유공간이라고 하자. Δ_i는 $\mathbf{x} \in \mathbb{R}^N$을 고유공간 W_i에 내리는 직교 사영이라고 하자. 이 때, 다음이 성립한다.*

1. $\mathbb{R}^N = W_1 \oplus \cdots \oplus W_N$.
2. $1 \leq i \leq N$에 대하여, $\Delta_i = \phi_i \phi_i^T D$ 이고, 내적 $\langle \cdot, \cdot \rangle_D$에 대하여 자기수반작용소이다.
3. $1 \leq i, j \leq N$에 대하여, $\Delta_i \Delta_j = \delta_{ij} \Delta_i$이다.
4. $I = \Delta_1 + \cdots + \Delta_N$이다.
5. $\Delta = \lambda_1 \Delta_1 + \cdots + \lambda_N \Delta_N$이다.

4.3에 의하여, 각 $\mathbf{x}_i = \Delta_i(\mathbf{x})$이고 따라서 이 분해는 조건 1을 만족한다. 왜냐하면 우리는 선형 작용소들의 집합 $\{\Delta_i\}$을 찾았기 때문이다.

4.2 선형 작용소

임의의 다항함수 f와 정사각행렬 M에 대하여, 행렬 $f(M)$은 잘 정의되어 있다. 우리의 경우, 행렬 $f(\Delta)$는 신호 벡터를 또 다른 신호벡터로 보내는 선형작용소이다.

4.2.1 밴드패스 필터

f가 구간 $[0, 2]$위에 정의된 다항함수라면, 임의의 라플라시안 행렬 Δ 와 신호벡터 $\mathbf{x}$에 대하여 다음이 성립한다.

$$\begin{aligned} f(\Delta)\mathbf{x} &= f(\Delta)(\langle \mathbf{x}, \phi_1 \rangle_D \phi_1 + \cdots + \langle \mathbf{x}, \phi_N \rangle_D \phi_N) \\ &= \langle \mathbf{x}, \phi_1 \rangle_D f(\Delta)\phi_1 + \cdots + \langle \mathbf{x}, \phi_N \rangle_D f(\Delta)\phi_N \\ &= \langle \mathbf{x}, \phi_1 \rangle_D f(\lambda_1)\phi_1 + \cdots + \langle \mathbf{x}, \phi_N \rangle_D f(\lambda_N)\phi_N. \end{aligned}$$

라플라시안 행렬 Δ의 고유값 중 λ에서는 함수값이 1이고, 다른 고유값에서는 함수값이 0인 함수 f를 생각하자. 이런 다항식은 항상 찾을 수 있는데 라그랑즈(Lagrange) 보간 공식을 사용하면 된다. 예를 들면 f를 다음과 같이 정하면 된다.

$$f_\lambda(x) = \frac{\prod_{\substack{\mu \in \sigma(\Delta) \\ \mu \neq \lambda}} (x - \mu)}{\prod_{\substack{\mu \in \sigma(\Delta) \\ \mu \neq \lambda}} (\lambda - \mu)}.$$

이 다항식 f에 대하여 다음이 성립한다.

$$f_\lambda(\Delta)\mathbf{x} = c\phi(\lambda), \tag{8}$$

여기서 $\phi(\lambda)$는 고유치 λ에 대응되는 고유벡터이고 $c = \langle \mathbf{x}, \phi(\lambda) \rangle$이다. 각 신호 벡터는 이산 주파수를 가지므로, 선형 작용소 $f(\Delta)$는 오직 주파수가 λ에 대응되는 신호만 통과시키는 좁은 밴드패스 필터로 간주할 수 있다. 더구나, 우리가 양변에 벡터 $\mathbf{x}^T$를 곱한다면, 다음을 얻는다.

$$c^2 = \langle f_\lambda(\Delta)\mathbf{x}, \mathbf{x} \rangle_D. \tag{9}$$

이 결과의 의미는 스펙트랄 계수의 제곱 (주파수 λ에 해당되는 부분의 에너지)을 구하기 위해서 행렬 Δ의 고유벡터를 구할 필요가 없다는 것이다.대신에 행렬과 행렬간의 곱, 스칼라와 행렬간의 곱과 같은 기본 연산만을 통하여 같은 결과를 얻을 수 있다는 것이다.

그러나 다항식 f_λ은 특정한 스펙트럼 $\sigma(\Delta)$에 많이 의존한다. 앞에서도 논의가 되었듯이, 우리의 목표는 어떤 주파수 밴드에 대응되는 신호만 통과시키는 선형작용소를 구하

되, 행렬 Δ의 고유치를 직접 구하지 않는 것이다. 또한 우리의 기법이 최대한 다양한 종류의 라플라시안 행렬을 다룰 수 있기를 원한다. 따라서 우리는 특정한 행렬 Δ의 스펙트럼에 독립인 선형작용소를 찾으려고 한다. 이 때 우리는 고유치를 직접 구하지 않고도 라플라시안 행렬의 스펙트럼은 구간 $[0,2]$에 속한다는 것을 알고 있다.

4.2.2 다항 함수족에 의해 성성되는 선형 작용소

구간 $[a,b]$에서 정의되어 있는 함수 족 $\{f_j\}_{j\in J}$가 다음 조건을 만족한다고 하자.

$$\sum_{j\in J} f_j \equiv 1.$$

이 경우, 함수족 $\{f_j\}_{j\in J}$을 단위원(unity)에 대한 분할이라고 부르자. 이러한 예는 많이 찾을 수 있다. 한가지 예를 들자면, Bernstein 다항함수들

$$\left\{\frac{n!}{j!(n-j)!}t^j(1-t)^{n-j}\right\}_{j=0,\ldots,n}$$

은 구간 $[0,1]$에서 단위원에 대한 분할이다. 특별히 다항 함수족 $\{f_j\}_{j\in J=\{1,2,\ldots,N\}}$이 오직 다항함수들로만 이루어져 있다면, 라플라시안 행렬 Δ와 신호 벡터 $\mathbf{x}$에 대하여 $f_j(\Delta)$는 잘 정의되어있고 다음을 만족한다.

$$\begin{aligned}\mathbf{x} &= \sum_{j=1}^{N} f_j(\Delta)\mathbf{x} \\ &= f_1(\Delta)\mathbf{x} + f_2(\Delta)\mathbf{x} + \cdots + f_N(\Delta)\mathbf{x}. \qquad (10)\end{aligned}$$

만약 f_j가 앞에서 나온 f_λ와 같이 $f_j(\lambda_k) = \delta_{jk}$를 만족한다면, 다음과 같은 식을 만족한 다.

$$\begin{aligned}f_j(\Delta)\mathbf{x} &= f_j(\Delta)(c_1\phi_1 + c_2\phi_2 + \cdots + c_n\phi_n) \\ &= c_j\phi_j.\end{aligned}$$

그러므로 10은 $\mathbf{x}$를 다음과 같이 분할한다.

$$\begin{aligned}\mathbf{x} &= f_1(\Delta)\mathbf{x} + f_2(\Delta)\mathbf{x} + \cdots + f_N(\Delta)\mathbf{x} \\ &= c_1\phi_1 + c_2\phi_2 + \cdots + c_N\phi_N.\end{aligned}$$

4.3 특별한 함수족: 스케일링 함수(scaling functions)

여기서는 스케일링 함수라는 특별한 함수를 소개한다. 이 함수들은 웨이블릿 이론에 자주 등장한다.

4.3.1 직교 다항함수

V_n을 차수가 $n+1$미만인 다항식들로 이루어진 벡터 공간이라고 하자. $n+1$차원 벡터공간 V_n은 기저로서 $\{1,x,x^2,\ldots,x^n\}$를 가진다. 임의의 $f,g\in V_n$에 대하여, 다음과 같이 V_n위의 내적을 정의하자.

$$\langle f,g\rangle = \int_{-1}^{1} f(x)g(x)w(x)dx, \qquad (11)$$

여기서 $w(x)$는 구간 $[-1,1]$에서 정의된 음이 아닌 가중치 함수이다. 각 $w(x)$에 대하여 대응되는 내적은 서로 직교하는 다항식들을 생성해낸다. 특별히 $w(x) = 1/\sqrt{1-x^2}$, $-1 \le x \le 1$인 경우에 이를 체비세프(Chebyshev) 다항식이라 한다. 체비세프 다항식은 근사 이론(approximation theory)에서 매우 중요한 역할을 한다 [2].

$$T_n(x) = \cos(n\arccos x),\ n \ge 0.$$

체비세프 다항식들에 대해서는 많은 참고 서적들이 있으며 이 다항식에 대한 여러 가지 공식들은 [17]에서 찾아볼 수 있다.

$T_n(x)$의 모든 해는 구간 $[-1,1]$에 속해있고, 그 해를 삼각함수를 이용하여 구해보면 다음과 같이 주어짐을 알 수 있다.

$$x_j = \cos\left(\frac{2j\pi+\pi}{2n}\right), j = 0,1,\ldots,n-1.$$

$n+1$개의 다항식 $T_0(x),T_1(x),\ldots,T_n(x)$은 내적 (11)에 대하여 서로 직교하기 때문에 $\mathcal{T} = \{T_0(x),T_1(x),\ldots,T_n(x)\}$은 내적 공간 V_n의 기저를 이룬다. 이제 주어진 $\xi\in\mathbb{R}$에 대하여, 다음 함수를 정의하자.

$$K_n(x;\xi) := \sum_{k=0}^{n} T_k(x)T_k(\xi).$$

다항식 $K_n(x;\xi)$은 내적 $\langle\cdot,\cdot\rangle$과 ξ에 대한 커널 다항식이라고 불리운다. 커널 다항식들은 다음과 같은 재생(reproducing) 성질을 가지는 것으로 알려져 있다. 즉, 임의의 $p\in V_n$에 대하여,

$$\langle K_n(\cdot;\xi),p\rangle = \int_{-1}^{1} K_n(x;\xi)p(x)w(x) = p(\xi).$$

n번째 커널 다항식 $K_n(x;\xi)$은 다음과 같은 제한 조건이 있는 최적화 문제의 유일한 해가 된다.

$$\left\|\frac{K_n(\cdot;\xi)}{K_n(\xi;\xi)}\right\| = \min\{\|p\|\in V_n,\ p(\xi)=1\}. \qquad (12)$$

식(12)은 커널 다항식 $K_n(x;\xi)$이 한 점 $x=\xi$주위로 국소화되어 있다는 것을 말해준다. 그림(1)에서 보는 것과 같이 $K_n(x;\xi)$은 $x=\xi$에서 최대값을 가짐을 알 수 있다.

4.3.2 스케일링 함수

파라미터 $x_0 < x_1 < \cdots < x_n$에 대하여, 스케일링 함수를 다음과 같이 커널 다항식으로 정의하자.

$$\varphi_{n,k}(x) = \varphi_n(x;x_k) := K_n(x;x_k),\ k = 0,1,\ldots,n.$$

적절하게 파라미터 집합을 잡아주면 다음과 같은 정리를 얻는다.

Theorem 4.4 *만약 파라미터 $x_0 < x_1 < \cdots < x_n$을 $T_{n+1}(x) = 0$의 $n+1$개의 해로 잡아주면*

$$\begin{aligned}\langle\varphi_{n,k},\varphi_{n,l}\rangle &= \sum_{j=0}^{n} T_j(x_k)T_j(x_l) \\ &= \varphi_{n,k}(x_k)\cdot\delta_{k,l} \\ &= \varphi_n(x_k;x_k)\cdot\delta_{k,l}.\end{aligned}$$

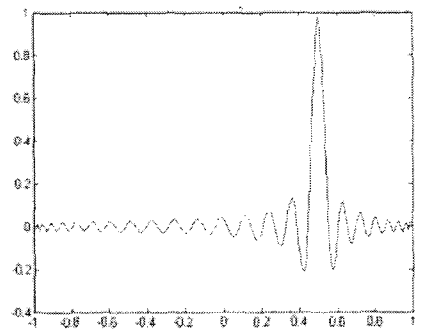

그림 1: 스케일링 함수, $x = 0$에 국소화 되어있음

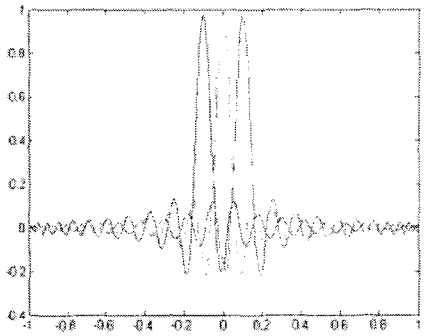

그림 2: 실효 통과 영역 $B_j(\delta)$, $\delta = 0.3$

위의 정리에 의하여, 스케일링 함수 $\varphi_{n,0}, \varphi_{n,2}, \ldots, \varphi_{n,n}$은 내적 벡터 공간 V_n의 또다른 직교기저를 이룬다. 따라서 특별히 $T_0 \in V_n$은 $\varphi_{n,0}, \varphi_{n,2}, \ldots, \varphi_{n,n}$들의 일차결합으로 다음과 같이 유일하게 표현된다.

$$T_0 \equiv 1 = \frac{1}{n+1}\left(\varphi_{n,0} + \varphi_{n,1} + \cdots + \varphi_{n,n}\right).$$

4.4 신호 벡터의 분해

이제 주어진 신호 벡터 $\mathbf{x}$를 분해해보자. $\tilde{\Delta} := \Delta - I$라고 놓으면 $\tilde{\Delta}$의 스펙트럼은 스케일링 함수들의 정의역 $[-1, 1]$ 안에 들어가게 된다. 각 $\mathbf{x}_j$에 대하여

$$\mathbf{x}_j = \frac{1}{n+1}\varphi_{n,j}(\tilde{\Delta})\mathbf{x}.$$

이라고 놓자. 그러면 $\mathbf{x}$는 다음과 같이 분해된다.

$$\begin{aligned}\mathbf{x} &= \mathbf{x}_1 + \cdots + \mathbf{x}_n \\ &= \frac{1}{n+1}\varphi_{n,1}(\tilde{\Delta})\mathbf{x} + \cdots + \frac{1}{n+1}\varphi_{n,n}(\tilde{\Delta})\mathbf{x}.\end{aligned}$$

분명히 이 분해는 2.2.1에서 제시된 분해 조건 1을 만족을 한다. 그러나 이 분할은 행렬 라플라시안 행렬 $\tilde{\Delta}$의 고유치들과 신호 벡터 $\mathbf{x}$의 스펙트랄 분포에 의존한다. 이 분해가 앞에서 제시한 나머지 조건들 2,3,4를 만족하는지는 분명하지 않다. 그림1에서 볼 수 있듯이 이 분할에 사용된 스케일링 함수들은 특정한 범위의 주파수를 가지는 벡터들은 크게 변화시키지 않고 통과시키는 반면, 이 범위 밖의 주파수에 대한 벡터는 거의 통과시키지 않음을 알 수 있다. $1 > \delta \geq 0.2$에 대하여, 선형작용소 $\varphi_{n,j}(\tilde{\Delta})$에 대한 실효 통과 영역, 즉 $\{x \in [-1, 1] \mid |\varphi_{n,j}(x)| \geq \delta\}$을 $B_j(\delta)$로 표시하자. 그림(2)보면 분명히 $\delta \to 1$이면 $B_j(\delta) \to \emptyset$이다. 고정된 n에 대하여, $\{B_j(\delta)\}_{j=1,\ldots,n}$은 거의 구간 $[-1, 1]$에 대한 분할처럼 작용한다. 즉,

$$\bigcup_{1 \leq j \leq n} B_j(\delta) = [-1, 1],\ m(B_k(\delta) \cap B_l(\delta)) \leq \eta, k \neq l, \quad (13)$$

여기서 m은 르벡(Lebesgue)측도이고 η는 δ에 의존한다.

식(13)은 조건 3의 완화된 버전이라고 볼 수 있다. 조건 3은 매우 강한 조건이라고 볼 수 있다. 대부분의 경우 비록 이 분할이 조건 3을 완전히 만족시키지는 못하더라도, 식(13)을 보면 이 분할은 여전히 우리의 목적에 잘 부합될 것이라고 기대할 수 있다.

이 분할을 위해서는 선형작용소 $\varphi_{n,j}(\tilde{\Delta})$를 계산해야 한다. 그러나 이것은 행렬을 다항식 $\varphi_{n,j}$의 차수만큼 곱해 나가야 한다. 일반적으로 두 개의 $N \times N$행렬을 곱하는데 걸리는 시간은 $O(N^3)$이므로, 모든 $\varphi_{n,j}(\tilde{\Delta})$을 계산하는데 걸리는 시간은 대략 모두 $O(nN^3)$이 될 것이므로, 이것은 $N \times N$행렬의 고유치와 고유벡터들을 구하는데 걸리는 시간보다 많다. 그러나 우리의 워터마킹 기법은 $\varphi_{n,j}(\tilde{\Delta})$을 개별적으로 구할 필요가 없다. 우리가 필요한 것은 $\varphi_{n,j}(\tilde{\Delta})$가 아니라 주어진 $\mathbf{x}$에 대한 $\varphi_{n,j}(\tilde{\Delta})\mathbf{x}$이다. 실제로 $\varphi_{n,j}(\tilde{\Delta})\mathbf{x}$을 계산하는데 걸리는 시간이 $O(nN^2)$정도가 되게 할 수 있음을 다음 두 가지 방법을 제시하여 보여주도록 한다.

4.4.1 방법 1

$\xi_j, j = 0, 1, \ldots, n$를 다항함수 $T_{n+1}(x) = 0$의 해라고 하자. 앞에서 설명되었듯이 이 해들은 다음과 같이 쓸 수 있다.

$$\xi_j = \cos\left(\frac{2j\pi + \pi}{2(n+1)}\right), j = 0, 1, \ldots, n.$$

다항식 $\varphi_{n,j}(x) = \varphi_n(x; \xi_j)$, $j = 0, 1, \ldots, n$은 n개의 서로 다른 해를 가지고 그 해들은 다음과 같이 됨을 쉽게 증명할 수 있다.

$$\xi_0, \xi_1, \ldots, \xi_{j-1}, \xi_{j+1}, \ldots, \xi_n.$$

다항함수 $\varphi_{n,j}(x)$의 차수는 모두 n이므로 각 $\varphi_{n,j}(x)$은 다음과 같이 인수분해가 된다.

$$\varphi_{n,j}(x) = A_j(x - \xi_0)\cdots(x - \xi_{j-1})(x - \xi_{j+1})\cdots(x - \xi_n)$$

이제 $\mathbf{x}_j = \varphi_{n,j}(\tilde{\Delta})\mathbf{x}$를 다음과 같은 단계로 계산해보자.

1. Input:
$\mathbf{x}_j \leftarrow \mathbf{x}$

2. From $j = 2$ to n except j
$\mathbf{x}_j \leftarrow (\tilde{\Delta} - \xi_i I)\mathbf{x}_j$

3. Output:
$\Psi \leftarrow 1/(n+1)\Psi$

위 계산법에서는 행렬과 벡터간의 곱이 n번 일어나므로 전체 걸리는 시간은 $O(nN^2)$이 된다. 차수가 n과 비교해서 작으면 작을수록 계산시간이 $O(nN^3)$인 행렬과 행렬의 곱보다 계산 속도가 그만큼 더 빨라지게 된다.

4.4.2 방법 2

이 방법은 체비세프 다항식에 대한 3항 점화식을 사용한다. 체비세프 다항식을 생성시키는 방법으로 이 3항 점화식이 쓰이기도 한다. 정의에 의하여 $T_0(x) = 1$, $T_1(x) = x$이다. 3항 점화식은 다음과 같이 주어진다.

$$T_{n+2}(x) = 2xT_{n+1}(x) - T_n(x), n = 0, 1, \ldots.$$

이 관계식을 이용하여, $\mathbf{x}_j = \varphi_{n,j}(\tilde{\Delta})\mathbf{x}$를 계산해보자.

1. Input:
$t_0 \leftarrow 0$, $t_1 \leftarrow \xi_j$
$T_0 \leftarrow$ zero matrix, $T_1 \leftarrow \tilde{\Delta}$
$\Psi \leftarrow t_0 T_0 + t_1 T_1$

2. From $j = 2$ to n
$t \leftarrow 2\xi t_1 - t_0$, $t_1 \leftarrow t$, $t_0 \leftarrow t_1$
$T \leftarrow 2\tilde{\Delta}T_1 - T_0$, $T_1 \leftarrow T$, $T_0 \leftarrow T_1$
$\Psi \leftarrow \Psi + t \cdot T$

3. Output:
$\Psi \leftarrow 1/(n+1)\Psi$

방법 1에서와 같이 행렬과 벡터간의 곱이 n번 일어난다. 방법 1은 하나의 $\mathbf{x}_j = \varphi_{n,j}(\tilde{\Delta})\mathbf{x}$를 계산하는데 편리한 방법이다. 그러나 방법 2를 약간만 고치면, 모든 $\varphi_{n,j}(\tilde{\Delta})\mathbf{x}, j = 1, 2, \ldots, n$를 한꺼번에 계산할 수 있다. 왜냐하면 이 벡터들의 계산에서 같은 벡터들 $\{T_0(\tilde{\Delta})\mathbf{x}, T_1(\tilde{\Delta})\mathbf{x} \ldots, T_n(\tilde{\Delta})\mathbf{x}\}$을 공유하기 때문이다.

5. 테스트 결과

우리는 테스트에서 체비세프 다항식을 이용한 스케일링 함수를 이용하였다. 우리가 제시한 워터마킹 방법의 강건성을 랜덤 노이즈, 아핀 변환, 리메쉉(remeshing) 공격에 대하여 테스트 하였다. 우리의 워터마킹 방법의 계산 시간과 강건성을 분석하였다.

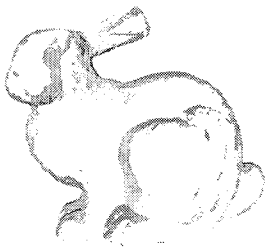

그림 3: 원본 Stanford bunny model

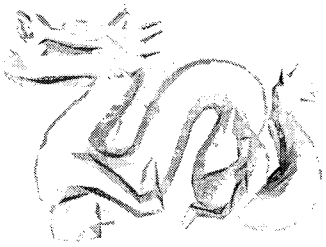

그림 4: 원본 Dragon model

5.1 테스트 환경

인텔 펜티엄4 2.6GHz 프로세서와 512메가 바이트의 메인 메모리를 갖는 PC에서 테스트 하였다. 사용한 모델은 Stanford Bunny [1], a Dragon [1], Skeleton Hand [1]이고 각각의 꼭지점의 개수는 1494, 1257, 1055개이다. 그림(3), (4), (5)는 워터마크가 들어가지 않은 Stanford Bunny, Dragon, Skeleton Hand의 형상이다.

5.2 방법

각각의 워터마크가 삽입된 모델에 대하여 랜덤 노이즈, 아핀 변환, 리메쉉 공격을 하였다. 랜덤 노이즈 공격을 위한 벡터는 두 가지 방법으로 구현하였다. 첫 번째는 공격 벡터의 각각의 원소값들이 -1에서 1사이에 분포하도록 하였다. 두 번째는 공격 벡터의 크기를 워터마크 벡터의 크기의 상수배가 되도록 하는 방법을 사용하였다. 아핀 변환 공격을 하기 위하여 워터마크가 들어간 모델에 스케일링과 회전 행렬, 그리고 평행이동 벡터를 적용시켰다. 마지막으로 리메쉉 공격은 워터마크가 삽입된 모델의 위상을 수정하는 방법으로 구현하였다. 이 때에 리메쉉 공격은 워터마크가 삽입된 모델의 형상은 조금 변화시키지만 모서리의 개수는 변화시키지 않

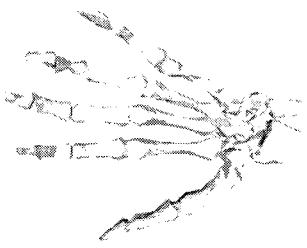

그림 5: 원본 Skeleton hand model

그림 6: 랜덤 노이즈 공격을 받은 Stanford Bunny

그림 7: 아핀 변환 공격을 받은 Dragon

는다.

그림(6), (7), (8)은 공격 받은 후의 Stanford Bunny, Dragon, Skeleton Hand모델의 형상이다.

또한 워터마킹의 계산 속도를 높이기 위하여 다음의 메쉬 파티션 방법을 BFS(Breadth First Search)를 이용하여 구현하였다.

1. 서브메쉬의 중심이 되는 꼭지점(feature vertex)을 선택한다.

2. 중심점으로부터 모서리로 연결된 자손꼭지점들을 구하여 서브메쉬에 추가한다.

3. 2단계에서 구해진 자손꼭지점들은 각각 새로운 중심점이 된다. 1단계과 2단계를 반복한다.

그림(9), (10), (11)은 서브메쉬에 워터마크가 삽입된 Stanford Bunny, Dragon, Skeleton Hand의 형상이다.

그림 8: 랜덤 노이즈 공격을 받은 Skeleton Hand

그림 9: 워터마크가 들어간 Stanford Bunny

그림 10: 워터마크가 들어간 Dragon

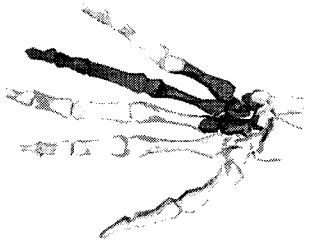

그림 11: 워터마크가 들어간 Skeleton Hand

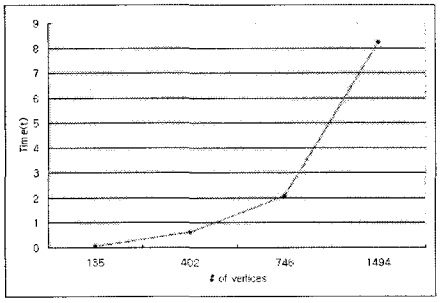

그림 12: 꼭지점 개수에 대한 계산 시간

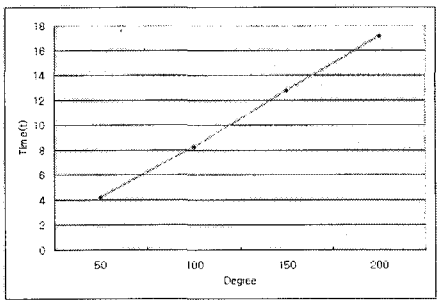

그림 13: 스케일링 함수 차수에 대한 계산 시간

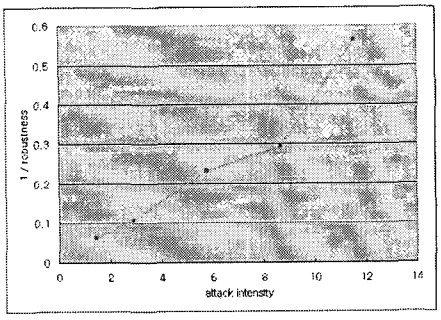

그림 14: 공격 강도에 대한 강건성

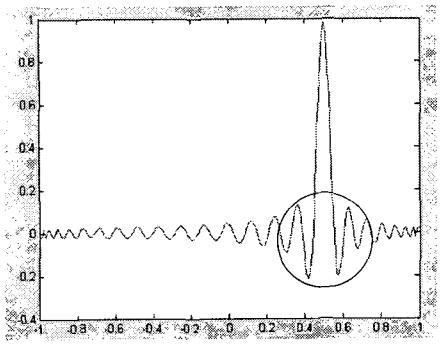

그림 15: Gibbs현상

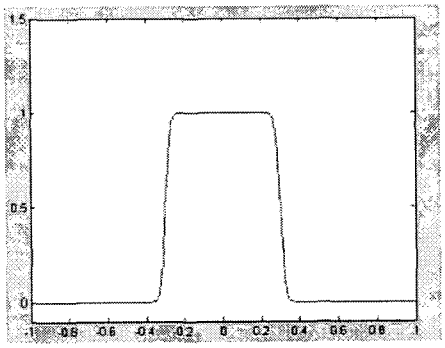

그림 16: 모자 함수

5.3 결과

5.3.1 계산 시간

계산 시간은 워터마크를 만들어서 원본모델에 삽입하는데 까지 걸리는 시간이다. 먼저, 스케일링 함수의 차수는 고정한 후 모델의 점의 개수에 대한 계산 시간을 측정하였다. 그림(12)는 계산 시간이 $O(N^2)$임을 보여준다. 이것은 메쉬 파티션을 이용하여 작은 메쉬를 만들면 큰 모델도 효율적으로 워터마킹 할 수 있음을 의미한다. 또한 점의 개수는 고정한 후 스케일링 함수의 차수에 대한 계산 시간을 측정하였다. 그림(13)은 계산 시간이 $O(N)$임을 보여준다. 이것은 높은 차수의 스케일링 함수를 사용하면 그 만큼 많은 워터마크 비트를 넣을 수 있으나 반비례하여 계산 시간은 길어지게 되는 것을 의미한다.

5.3.2 강건성

우리의 워터마킹 방법에 대한 강건성을 분석하였다. 이 테스트에서는 랜덤 노이즈 공격에 대한 강건성만 분석하였다. 왜냐하면 아핀 변환과 리메쉥 공격은 적당한 전처리를 거치고 난 후에는 랜덤 노이즈 공격의 일부로 볼 수 있기 때문이다. 그림(14)의 x축은 공격의 강도를 나타내고, y축은 강건성의 역수값을 나타낸다. 이 결과는 강건성이 공격의 강도에 반비례함을 보여준다.

6. 결론 및 고찰

우리의 워터마킹 방법은 기존의 스펙트랄 영역에 워터마크를 삽입하는 방법들의 안정성과 계산 시간을 줄이려고 하였다. 이 때에 계산 속도는 빨라지더라도 강건성은 약해질 수 있으리라 예상했다. 그러나 실험 결과가 보여주듯이 우리의 방법이 랜덤 노이즈 공격에 대한 강건성이 상당히 높음을 알 수 있다. 우리의 목적에 맞는 선형작용소를 생성해내기 위하여 웨이블릿 이론에 자주 나오는 스케일링 함수들을 사용하였다. 선형 작용소를 생성해내는 스케일링 함수들의 차수가 낮을수록 계산 속도는 그만큼 줄어들었다. 이 경우 계산 시간은 약 $O(nN^2)$이었으며, N은 행렬의 차수이고, n은 사용된 다항식의 차수이다. n이 커지면 그 만큼 많은 정보를 심어 넣을 수 있지만 계산 시간은 반비례하여 더 많이 걸리게 된다.

$x = \xi$에 국소화되어 있는 스케일링 함수는 다항함수로서 피할 수 없이 $x = \xi$근처에서 점프(overshoot)를 유발한다. 이 점프는 우리가 제시한 방법의 강건성을 약화시키는 원인일 것 으로 예상된다. 그림15와 같이 점프가 일어나는 현상은 푸리에 분석(Fourier analysis)에서 나타나는 Gibbs 현상이라고 불리우는 것으로서, Gibbs 현상은 삼각함수들의 일차결합으로 함수를 근사할 때 불연속인 점에서 나타난다. Gibbs 현상의 특징은 차수를 아무리 높여 근사를 시키더라도 이 현상은 사라지지 않는다는 것이다. 이런 단점에도 불구하고 우리가 선형작용소들을 생성하는데 사용했던 스케일링 함수들을 구성하는데 있어서 직교 다항식들은 오랜 기간 학자들에 의해 연구되어 왔다. 따라서 스케일링 함수로 생성한 선형작용소들을 빠르고 쉽게 계산할 수 있다.

향후, 우리는 선형작용소들에 대한 함수들을 선택할 때, 스케일링 함수들에 대한 대안 함수들을 찾아보고자 한다. 그림16에서 보는 바와 같이 국소성을 가지는 모자 모양의 함수를 고른 근사(Uniform approximation)를 하는 다항함수들로 생각해 볼 수 있다. 고른 근사를 하므로 Gibbs 현상은 많이 줄어 들 것으로 예상된다. 또는 대안 함수로서 유리함수를 사용하면 Gibbs현상은 사라지게 된다.

참고 문헌

[1] Available at georgia institute of technology. In *http // www.cc.gatech.edu/projects/large_ models/.*

[2] K. Atkinson. *An Introduction to Numerical Analysis*. John Wiley, 2nd edition, 1989.

[3] O. Benedens. Geometry-based watermarking of 3d models. *IEEE Computer Graphics and Applications*, 19(1):46–55, 1999.

[4] F. R. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[5] D. Cotting, T. Weyrich, M. Pauly, and M. Gross. Robust watermarking of point-sampled geometry. In *International Conference on Shape Modeling and Applications 2004*, pages 234–242, Genova, Italy, June 2004.

[6] G. Golub and C. V. Loan. *Matrix Computations*. John Hopkins Univ. Press, 3rd edition, 1996.

[7] C.-T. Hsu and J.-L. Wu. Hidden digital watermarking in images. *IEEE Trans. Image Process*, 8(1):58–68, 1999.

[8] K. H. Ko, T. Maekawa, N. M. Patrikalakis, H. Masuda, and F.-E. Wolter. Shape intrinsic fingerprints for free-form object matching. In *Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications*, pages 196–207, Seattle, USA, 2003.

[9] J. J. Lee, N. I. Cho, and J. Kim. Watermarking for 3d nurbs graphic data. In *International Workshop on Multimedia Signal Processing 2002*, pages 304–307, 2002.

[10] R. Ohbuchi, H. Masuada, and M. Aono. Data embedding algorithms for geometrical and non-geometrical targets in three-dimensional polygon models. *Computer Communications*, 21:1344–1354, 1998.

[11] R. Ohbuchi and H. Masuda. Managing cad data as a multimedia data type using digital watermarking. In *IFIP WG5.2 Fourth Workshop on Knowledge Intensive CAD*, May 2000.

[12] R. Ohbuchi, H. Masuda, and M. Aono. Watermarking three-dimensional polygon models. In *Proceedings of the ACM Multimedia '97*, pages 261–272, Seattle, Washington, USA, November 1997.

[13] R. Ohbuchi, H. Masuda, and M. Aono. Watermarking three-dimensional polygon models through geometric and topological modifications. *IEEE Journal on Selected Areas in Communications*, pages 551–559, 1998.

[14] R. Ohbuchi, S. Takahashi, T. Miyazawa, and A. Mukaiyama. Watermarking 3d polygonal meshes in the mesh spectral domain. In *Proc. Graphics Interface 2001*, pages 9–17, 2001.

[15] I. Pitas. A method for watermark casting on digital images. *IEEE Trans. Circuit Syst. Video Technol.*, 8(6):775–780, 1998.

[16] E. Praun, H. Hoppe, and A. Finkelstein. Robust mesh watermarking. *Computer Graphics (Proceedings of SIGGRAPH'99)*, 33:49–56, 1999.

[17] G. Szegö. *Orthogonal polynomials*. AMS Colloquium Publications XXIII, American Mathematical Society, New York, USA, revised edition, 1959.

[18] G. Taubin. A signal processing approach to fair surface design. *Computer Graphics (SIGGRAPH'95 Proceedings)*, pages 351–358, 1995.

[19] M. Wagner. Robust watermarking of polygonal meshes. In R. Martin and W. Wang, editors, *Proceedings of the Conference on Geometric Modeling and Processing*, pages 201–208, 2000.