

논문 2006-43CI-6-9

# 기하학적 구조 및 위치 보간기를 이용한 3D 애니메이션 워터마킹

(3D Animation Watermarking Using Geometrical Structure and  
PositionInterpolator)

이 석 환\*, 권 기 룡\*\*

(Suk-Hwan Lee and Ki-Ryong Kwon)

## 요 약

3D 그래픽에서는 실시간 애니메이션을 위하여 이동, 회전, 스케일링 변환 노드로 구성되는 키프레임 애니메이션이 널리 이용되고 있다. 본 논문에서는 VRML 기반의 3D 애니메이션 상에서 CoordIndex 노드 내에 꼭지점 좌표를 이용한 기하학적 워터마킹 기법과 PositionInterpolator 노드 내에 키 값을 이용한 보간기 워터마킹 기법을 제안한다. 실험 결과로부터 제안한 방법이 각종 기하학적 공격 및 타임라인 공격에 대한 강인성 및 비가시성을 가짐을 확인하였다.

## Abstract

For real-time animation, keyframe animation that consists of translation, rotation, scaling transform nodes is used widely in 3D graphics. This paper presents geometrical watermarking using vertex coordinates in CoordIndex node and interpolator watermarking using keyvalues in PositionInterpolator node for 3D keyframe animation based on VRML. Experimental results verify that the proposed algorithm has the robustness against geometrical attacks and timeline attacks as well as the invisibility.

**Keywords :** 3D keyframe animation, watermarking, Geometrical Structure, PositionInterpolator

## I. 서 론

저작권 보호 및 불법 복제 추적을 위한 워터마킹 및 팽퍼프린팅 기술들은 오디오, 정지영상 및 비디오 등의 디지털 콘텐츠에 대하여 많이 적용되고 연구되어 왔다.<sup>[1],[2]</sup> 최근 웹 3D 상에서의 3D 그래픽 모델에 대한

많은 관심과 더불어, 3D 그래픽 모델에 대한 워터마킹 기술이 매우 주요한 연구 분야로 부각되고 있다. 3D 그래픽 모델은 일반적으로 꼭지점들의 좌표값 및 정렬된 꼭지점들의 연결 정보로 정의되는 메쉬로 표현된다. 그러나 정지 영상 및 비디오와는 달리 3D 그래픽 모델은 표현 기법들이 다양하고, 기하학적 및 위상학적 연산에 의하여 편집이 용이하다. 그리고 3D 형상을 그대로 유지하면서 꼭지점들의 정렬 순서 및 연결 정보들을 가변할 수 있다. 또한 웹 상에서 렌더링 속도를 높이기 위하여 메쉬 간단화에 의하여 꼭지점 데이터량을 줄일 수 있다. 이와 같은 공격에 대하여 강인성을 가지며, 원 모델없이 워터마크를 추출할 수 있는 3D 워터마킹 기술이 제안되고 있다.<sup>[3]-[9]</sup>

최근 그래픽 하드웨어의 성능이 향상되면서 보다 자

\* 정희원, 동명대학교 정보보호학과  
(Dept. of Information Security, Tongmyong University)

\*\* 정희원, 부경대학교 전자컴퓨터정보통신공학부  
(Division of Electronics, Computer and Telecommunication Eng., Pukyong University)

※ 본 논문은 2005년 정부(교육인적자원부)의 재원으로  
한국학술진흥재단의 지원을 받아 수행된 연구임  
(KRF-2005-042-D00225)

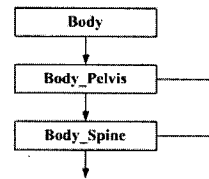
접수일자: 2006년7월20일, 수정완료일: 2006년10월30일

연스럽고 사실적인 3D 이미지를 실시간으로 렌더링할 수 있게 되었다. 따라서 3D 컴퓨터 애니메이션은 3D 모델의 사실적인 표현에 있어서 매우 중요한 요소이며, 실시간 3D 애플리케이션, 3D 애니메이션 영화, 및 3D 컴퓨터 게임에 있어서 필수 요소가 되었다. 그러나 현재 국내에 유통되는 3D 캐릭터 애니메이션의 불법 복제로 인하여 캐릭터 개발업체 및 제조업체가 피해를 입고 있다. 따라서 3D 컴퓨터 애니메이션의 저작권을 보호하는 기술이 매우 절실히 요구되고 있으나, 이에 대한 연구가 매우 미비하다. 본 논문에서는 3D 컴퓨터 애니메이션의 저작권을 보호하기 위한 워터마킹 기법을 제안한다.

3D 그래픽에서 애니메이션은 3D 공간 상에서 메쉬 또는 텍스처 등을 포함하는 움직이는 물체로 정의된다. 3D 그래픽에서 널리 사용되는 애니메이션 기법으로는 다음과 같다. 1) 꼭지점 애니메이션 : 이 기법은 모핑(morphing)과 같이, 각 프레임 상에서 애니메이트되는 꼭지점들의 위치를 저장하고, 프레임 간의 꼭지점들은 보간기에 의하여 생성한다. 2) 계층적 애니메이션 : 캐릭터 등의 관절체는 계층적 구조로 구성되어 있다. 계층적 애니메이션 기법은 하나의 캐릭터를 여러 개의 메쉬 모델들로 나눈 후, 이들을 부모-자식 관계로 상속하여 계층적 구조로 만든다. 그리고 각 프레임 상의 계층 모델들을 이동, 회전 및 크기의 변환 행렬에 의하여 표현한다. 3) 뼈대 기반 애니메이션 : 이 기법은 계층적 애니메이션의 확장 기법으로 인간 몸의 뼈대와 같이 3D 데이터를 가지는 뼈대를 만든 후 각 뼈대에 메쉬들을 붙인다. 4) 스키닝 : 이 기법은 계층적 및 뼈대 기반 애니메이션에서 발생하는 관절체의 불연속성을 각 뼈대의 가중치 기법 등을 이용하여 연속적으로 만드는 것이다. 5) 역운동학 : 역운동학(IK, Inverse Kinematics)은 자연스러운 움직임을 표현하기 위하여 계층 구조의 최하위 오브젝트의 위치를 조정함으로써 최상위 고정점까지 연결된 링크 오브젝트들의 위치에 역방향으로 영향을 미치도록 하는 방식이다.

3D 그래픽에서는 실시간 애니메이션을 위하여 위의 방법들을 적용한 키프레임 애니메이션이 널리 사용되고 있다. 이 방법은 전체 프레임 중에서 중요한 프레임들의 키값을 등록하고, 등록된 키값들을 이용한 보간기에 의하여 나머지 프레임들을 생성한다. 본 논문에서는 3D 그래픽 상에서 널리 이용되고 있는 키프레임 애니메이션에 대한 워터마킹 기술을 제안한다.

Hierarchical structure



(a)

```

DEF Body Transform
{
  translation -0.1064 74.08 22.21
  rotation 0 1 0 -1.571
  scale 0.5313 0.5313 0.5313
  children
  {
    DEF Body-TIMER TimeSensor { loop TRUE cycleInterval 5 }
    DEF Body-POS-INTERP PositionInterpolator
    {
      key [...]
      keyValue [...]
    }
    DEF Bipody-POS-INTERP OrientationInterpolator
    {
      key [...]
      keyValue [...]
    }
  }
}
DEF Body_Pelvis Transform
{
  translation -19.46 0 -5.658e-005
  rotation 0.5773 0.5774 -0.5774 -2.094
  scale 25.21 40 25.21
  children
  {
    DEF Body_Pelvis-POS-INTERP PositionInterpolator
    {
      key [...]
      keyValue [...]
    }
    Shape
    {
      geometry DEF Body_Pelvis-FACES IndexedFaceSet
      coord DEF Body_Pelvis-COORD Coordinate
      point [...]
      coordIndex [...]
    }
  }
}
DEF Body_Spine Transform
{
  ROUTE Body-TIMER.fraction_changed TO Body-POS-INTERP.set_fraction
  ROUTE Body-POS-INTERP.value_changed TO Body.set_translation
  ROUTE Body-TIMER.fraction_changed TO Body-ROT-INTERP.set_fraction
  ROUTE Body-ROT-INTERP.value_changed TO Body.set_rotation
  ROUTE Body-TIMER.fraction_changed TO Body_Pelvis-POS-INTERP.set_fraction
  ROUTE Body_Pelvis-POS-INTERP.value_changed TO Body_Pelvis.set_translation
}

```

(b)

그림 1. VRML 애니메이션 파일 상에서 (a) 계층 구조 및 (b) 변환 노드의 구조

Fig. 1. (a) Hierarchical structure and (b) transform node structure in a VRML animation file.

## II. VRML 애니메이션

### 2.1 워터마크 삽입 대상체

VRML 기반의 애니메이션<sup>[10]</sup>은 그림 1 (a)에서와 같이 여러 개의 메쉬 모델들이 부모 자식 관계의 계층적으로 구성된다. 그리고 각 메쉬 모델들은 하나의 변환 노드들을 가지며, 변환 노드의 구조는 그림 1에서와 같이 위 절에서 언급한 보간기 및 기하학적 구조로 나타난다. 이와 같은 VRML 애니메이션 파일구조로부터 3D 애니메이션에 워터마크를 삽입하기 위한 대상체로는 다음과 같이 세 가지로 고려될 수 있다.

(1) 계층 구조 상에서 각 메쉬 모델의 변환 노드 행렬

전역 좌표계를 기준으로 각 메쉬 모델들이 얼마만큼 이동, 크기 및 회전되었는지를 표현하는 변환 행렬로 그림 1 (b) 상에서 (1)에 해당된다. 이와 같은 변환 노드의 행렬은 각 메쉬 모델들을 모델링하기 위한 지역 좌표계를 나타낸다. 변환 노드 상에서의 각 필드는 크기, 회전 및 이동의 순서로 변환된다. 워터마크는 각 변환 노드의 이동, 크기 및 회전 행렬의 값에 삽입될 수 있다.

(2) 보간기 노드 상에서의 키 또는 키 값

키 또는 키 값은 한 물체의 움직임 정보를 나타내는 것으로 워터마크 삽입 대상체로 매우 중요한 요소로 고려될 수 있는 요소이다. 보간기 노드에 대한 설명은 다음 절에 자세히 언급하기로 한다.

(3) 각 변환 노드 내에 coordIndex 필드의 기하학적 성질

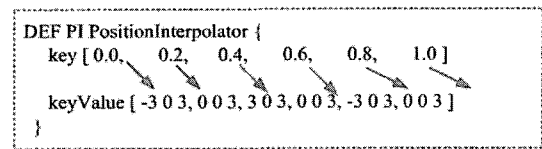
coordIndex 필드 내에 기하학적 정보는 꼭지점 좌표 값 및 꼭지점의 연결 정보가 있다. 이는 움직이는 각 메쉬 모델의 초기 기하학적 정보로 보간기의 키 값에 의하여 변환되어 렌더링된다. 3D 그래픽 정지 모델과는 유사하나, 3D 애니메이션에서는 계층 구조 내에서의 메쉬 모델을 고려하여야 한다.

변환 노드의 행렬들은 각 메쉬 모델의 초기 지역 좌표계를 나타내는 중요한 정보이나, 삽입되는 워터마크의 용량이 매우 제한적이다. 따라서 제안한 방법에서는 계층 구조를 가지는 각 메쉬 모델의 변환 노드 상에서 보간기 노드의 키 값과 coordIndex 필드 내에 꼭지점 좌표값에 워터마크를 각각 삽입한다.

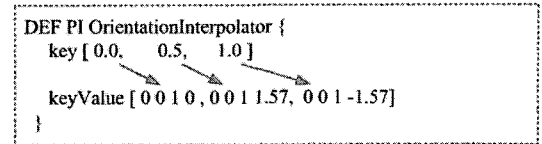
2.2 보간기

본 절에서는 3D 애니메이션 워터마크 삽입 대상체인 보간기에 대하여 간략히 살펴보기로 한다. 그러나 coordIndex 필드 내에서의 기하학적 구조는 3D 그래픽 정지 모델과 유사하므로, 이에 대한 설명은 생략하기로 한다.

VRML 및 MPEG-4 BIFS (Binary Format for Scenes)에서는 선형 키프레임 애니메이션으로 설계된 ColorInterpolator, CoordinateInterpolator, NormalInterpolator, OrientationInterpolator, PositionInterpolator 및 ScalarInterpolator의 보간 노드들이 있으며, 각 노드들은 사용자에게 따라 다양하게 적용된다.<sup>[10],[11]</sup> 이들은 각 분할된 시간동안 (eventIn) 보간



(a)



(b)

그림 2. (a) PositionInterpolator 및 (b) OrientationInterpolator의 예

Fig. 2. The example of (a) PositionInterpolator and (b) OrientationInterpolator.

노드로 보내어지는 이·벤트(eventOut)를 TimeSeonsor에 의하여 생성함으로써 애니메이션을 조정한다. 즉, 보간기의 키 값(exposedField)이 바뀔 때마다 한 이벤트는 위치, 방향 또는 색상 등과 같은 속성을 변경하기 위하여 다른 보간기로 보내어진다. 하나의 보간 노드는 분할 시간  $t$ 인 키(key)에 대한 키 값(keyValue)의 조각적 함수(piecewise function)  $f(t) [-\infty, \infty]$ 로 정의되며, 키와 키에 대한 키 값의 집합으로 구성된다.  $n$ 개의 키  $k_i, (i \in [0, n-1])$ 일 때, 조각적 선형 함수  $f(t)$ 는

$$f(t) = \begin{cases} v_0, & t < k_0 \\ v_{n-1}, & t > k_{n-1} \\ \text{linterp}(t, v_i, v_{i+1}), & k_i < t < k_{i+1} \\ v_i, & t = k_i \end{cases} \quad (1)$$

이다. 여기서  $\text{linterp}(t, v_i, v_{i+1})$ 는  $v_i$ 와  $v_{i+1}$  ( $0 \leq i < n-1$ ) 사이의 선형 보간값이다.

PositionInterpolator 및 OrientationInterpolator 노드는 키 별로 각 메쉬 모델의 위치 및 회전을 나타내므로 간단한 키프레임 애니메이션을 매우 용이하게 구현할 수 있다. 따라서 본 논문에서는 키프레임 애니메이션에서 가장 많이 이용되고 있고, 워터마크를 강인하게 삽입하기 위하여 PositionInterpolator 및 OrientationInterpolator 노드를 워터마크 삽입 대상으로 고려한다.

PositionInterpolator는 한 물체의 위치를 나타내는 키 값들 사이를 선형적으로 보간한다. 임의의 키에 대한 키 값은 직각좌표계 상에서의 좌표 (x,y,z)를 나타내는 벡터이다. 키 값들은 물체의 위치를 나타낼 뿐만 아니라, 물체의 속도를 조정할 수 있다. 즉, 키 값들 사이의 간격을 일정하게 유지하면 물체는 일정한 속도로 움직

이게 된다. 그림 2 (a)는 6개의 키와 키 값들로 구성된 PositionInterpolator 노드의 예를 나타낸다. 이 그림을 살펴보면 6개 키 값들 중 x 값들이 일정한 간격으로 변하므로, 이 물체는 x축 방향에 대하여 등속도로 운동함을 알 수 있다. OrientationInterpolator는 방향을 나타내는 키값들 사이를 보간한다. 여기서 두 방향에 사이의 보간은 단위 구 상에서 두 방향의 가장 가까운 거리를 구한 후 이를 잇는 호 (arc) 상에서 선형적으로 구해진다. OrientationInterpolator에서 임의의 키에 대한 키 값은  $xx+yy+zz$  축에 대한 방향  $\theta$ 로의 회전을 나타내는 4개의 값  $(x,y,z,\theta)$ 으로 구성된다. Fig. 2 (b)는 3개의 키에 대한 키값을 가지는 OrientationInterpolator 노드를 나타내고 있다. 이 그림을 살펴보면, 물체는 z축에 대하여 일정한 각속도로  $\pi/2$ 만큼 반시계방향으로 회전한 후에  $-\pi/2$ 만큼 시계방향으로 회전한다. 위의 OrientationInterpolator의 다른 표현으로 쿼터니언 (quaternion)  $Q$ 가 있다. 쿼터니언  $Q$ 는 4개의 성분  $\cos(\theta/2), n_x, n_y, n_z$ 에 의하여  $Q: (\cos(\theta/2), \sin(\theta/2)n_x + \sin(\theta/2)n_y + \sin(\theta/2)n_z)$ 와 같이 표현한다.  $\theta$ 는 회전각을 나타내고,  $n_x, n_y, n_z$ 는 회전축의 단위법선벡터를 나타낸다.

워터마크를 삽입하기 위하여 PositionInterpolator 노드에서는 각 키에 대한  $(x,y,z)$ 의 키 값 그리고 OrientationInterpolator 노드에서는  $(x,y,z,\theta)$ 의 키 값을 사용할 수 있다.

본 논문에서는 VRML 기반의 애니메이션에서 기하학적 정보인 꼭지점 좌표와 PositionInterpolator 필드의  $(x,y,z)$ 의 키 값에 각각 워터마크를 삽입하는 방법을 제안한다.

## II. 제안한 애니메이션 워터마킹

일반적으로 3D 애니메이션에 대한 공격으로는 기하학적 정보 및 움직임 정보에 대한 공격으로 나눌 수 있다. 이들 공격에 강인성을 가지기 위하여 워터마크는 이 두 가지 정보에 각각 삽입되어야 한다. 제안한 애니메이션 워터마킹 기법에서는 그림 3에서와 같이 먼저 전체 계층적인 노드 중 임의의 변환 노드들을 선택한다. 그리고 이진 워터마크 데이터를 선택된 변환 노드 상에서 PositionInterpolator 필드의 키 값과 CoordIndex 필드의 꼭지점 좌표에 각각 삽입한다.

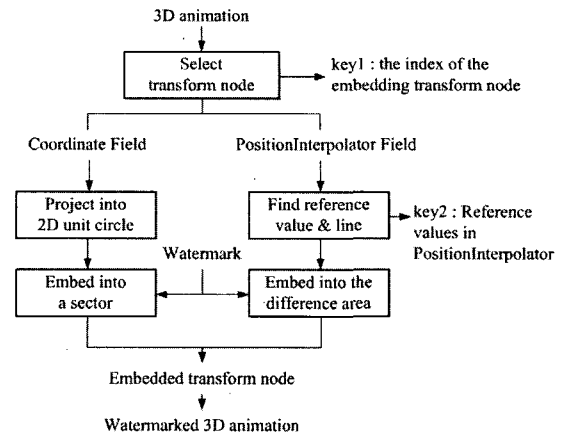


그림 3. 3D VRML 기반 애니메이션에 제안한 워터마크 삽입 과정

Fig. 3. The process of the proposed watermark embedding in 3D VRML animation.

### 1. 기하학적 워터마킹

기하학적 워터마킹에서는 그림 1에서와 같이 선택된 변환 노드 상에서 CoordIndex 필드 내의 꼭지점 좌표에 워터마크를 삽입하는 것이다. 즉, 총  $N$ 비트의 워터마크 데이터 중  $n$ 비트가 선택된 변환 노드에 삽입된다. 각 변환 노드의 CoordIndex 필드 내에 꼭지점  $v$ 의 좌표 성분들은 변환 노드의 지역 좌표계  $V_{local}$ 의 축  $X_{local}, Y_{local}, Z_{local}$ 을 기준으로 한다. 실 좌표계가  $V_{world} = V_{local} \times M_{TR}$ 이면, 지역 좌표계는  $V_{local} = V_{world} \times M_{TR}^{-1}$ 이다. 여기서  $M_{TR}$ 는  $M_{TR} = M_{xld} \times M_{parent}$ 이고,  $M_{xld}$ 는 자식 노드인 변환 노드의 변환 행렬  $M_{xld} = M_{rotation} \times M_{translation}$ 로 스케일 행렬  $M_{scale}$ , 회전 행렬  $M_{rotation}$ , 이동 행렬  $M_{translation}$ 의 곱으로 나타낸다.  $M_{parent}$ 은 선택된 변환 노드를 상속한 부모 노드인 변환 노드의 변환 행렬을 나타낸다.

선택된 변환 노드  $TR_i$  내에 모든 꼭지점  $v_{i \in [0, N_{TR_i}]}$ 들의 단위 벡터  $\overline{v_{i \in [0, N_{TR_i}]}}$ 을 2D 지역 좌표계  $\{X_{local}, Y_{local}\}$ 의 단위원으로 투영시킨다. 여기서 단위원은 하나의 변환 노드 상에 삽입될 워터마크 비트 수  $n$ 와 동일하게  $n$ 등분된다. 그림 2 (a)는 4비트의 워터마크를 삽입하기 위하여 4등분된 단위원을 보여준다. 하나의 워터마크 비트  $w_{k \in [1, n]}$ 는 등분된 구역에 투영되는 꼭지점 좌표의 중간점  $c_{k \in [1, n]}$

$$c_{k \in [1, n]} = \frac{1}{N_k} \sum_{j=1}^{N_k} (\widehat{v}_{xj} X_{local} + \widehat{v}_{yj} Y_{local}) = c_{kx} X_{local} + c_{ky} Y_{local} \quad (2)$$

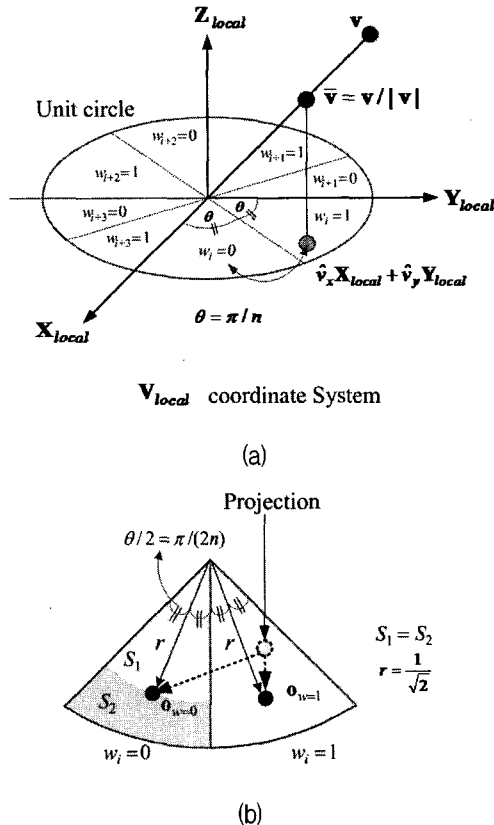


그림 4. CoordIndex 필드 내의 꼭지점 좌표에 워터마크 삽입 방법; (a) 꼭지점  $\mathbf{v}$ 를 2D 지역 좌표계의 단위 원으로 투영 (b) 분할된 구역 내에 워터마크 비트 삽입

Fig. 4. The watermark embedding into vertex coordinates in CoordIndex field: (a) Projection a vertex  $\mathbf{v}$  onto unit circle of 2D local coordinate system and (b) Embedding a watermark bit into a sector of unit circle.

에 삽입된다. 여기서  $N_k$ 는  $k$ 번째 구역에 투영되는 꼭지점 벡터  $\widehat{v_{kj \in [0, N_k]}} = \widehat{v_{xj}} \mathbf{X}_{local} + \widehat{v_{yj}} \mathbf{Y}_{local}$  들의 개수를 나타낸다.  $w_{k \in [1, n]}$ 가 1이면, 중간점  $c_{k \in [1, n]}$ 을 구역의 오른쪽 목표점  $o_{w_k=1}$ 으로 이동시키고 반대로  $w_{k \in [1, n]}$ 가 0이면, 구역의 왼쪽 목표점  $o_{w_k=0}$ 으로 그림 4에서와 같이 이동시킨다. 워터마크 비트에 따른 이동 목표점  $\{o_{w_k=1}, o_{w_k=0}\}$ 은 워터마크의 견고성을 가지기 위하여 구역을 면적이 동일하게 채등분할 수 있는 구간의 중간점으로 결정된다. 즉,  $k$ 번째 구역의 이동 목표점  $\{o_{w_k=1}, o_{w_k=0}\}$ 은

$$\begin{aligned}
 o_{w_k=0} &= \frac{1}{\sqrt{2}} \cos \frac{4k-3}{2n} \pi \mathbf{X}_{local} + \frac{1}{\sqrt{2}} \sin \frac{4k-3}{2n} \pi \mathbf{Y}_{local} \\
 &= o_{kx0} \mathbf{X}_{local} + o_{ky0} \mathbf{Y}_{local}
 \end{aligned} \tag{3}$$

$$\begin{aligned}
 o_{w_k=1} &= \frac{1}{\sqrt{2}} \cos \frac{4k-1}{2n} \pi \mathbf{X}_{local} + \frac{1}{\sqrt{2}} \sin \frac{4k-1}{2n} \pi \mathbf{Y}_{local} \\
 &= o_{kx1} \mathbf{X}_{local} + o_{ky1} \mathbf{Y}_{local}
 \end{aligned} \tag{4}$$

이다. 여기서 워터마크 비트  $w_{k \in [1, n]}$ 에 따라 중간점  $c_{k \in [1, n]}$ 을 목표점으로 이동하기 위하여 투영된 꼭지점 벡터  $\widehat{v_{kj \in [0, N_k]}} = \widehat{v_{xj}} \mathbf{X}_{local} + \widehat{v_{yj}} \mathbf{Y}_{local}$  들의 좌표를

$$\widehat{v}'_{xj} = \widehat{v}_{xj} + \delta_{xj}, \widehat{v}'_{yj} = \widehat{v}_{yj} + \delta_{yj} \tag{5}$$

와 같이 각각 변경한다. 각 좌표값을 비가시적으로 변경하기 위하여  $\delta_{xj}, \delta_{yj}$ 는

$$\delta_{xj} = \begin{cases} \Delta x_j & \text{if } \alpha(o_{kx} - c_{kx}) > \Delta x_j \\ -\Delta x_j & \text{if } \alpha(o_{kx} - c_{kx}) < -\Delta x_j \\ \alpha \times (o_{kx} - c_{kx}) & \text{else} \end{cases} \tag{6}$$

$$\delta_{yj} = \begin{cases} \Delta y_j & \text{if } \alpha(o_{ky} - c_{ky}) > \Delta y_j \\ -\Delta y_j & \text{if } \alpha(o_{ky} - c_{ky}) < -\Delta y_j \\ \alpha \times (o_{ky} - c_{ky}) & \text{else} \end{cases} \tag{7}$$

와 같이 제한 범위  $[\Delta x_j, \Delta y_j]$  내에  $\alpha \times (o_{kx} - c_{kx})$  및  $\alpha \times (o_{ky} - c_{ky})$ 으로 결정된다. 제한 범위  $[\Delta x_j, \Delta y_j]$ 는 각각

$$\Delta x_j = \min |\widehat{v_{xj}} - v_{val(x)}|, \tag{8}$$

$$\Delta y_j = \min |\widehat{v_{yj}} - v_{val(y)}|$$

와 같으며, 이는 투영된 꼭지점 벡터  $\widehat{v_{kj}}$ 에 연결된 벡터  $\widehat{v_{val}} = \widehat{v_{val(x)}} \mathbf{X}_{local} + \widehat{v_{val(y)}} \mathbf{Y}_{local}$ 의 좌표를 벗어나지 않는다.

기하학적 워터마킹에서는 단위원의 구역 개수  $n$  및 이동 인자  $\alpha$ 의 변수가 존재한다. 먼저 단위원의 구역 개수  $n$ 은 견고성 및 용량성에 따라 선택하여야 한다. 하나의 변환 노드 상에서는 약 수십 개 정도의 꼭지점이 있다. 단위원의 구역 개수  $n$ 이 크면, 각 구역에 투영되는 꼭지점의 개수가 작아진다. 즉, 단위원의 구역 개수  $n$ 가 증가할수록, 워터마크의 삽입 용량이 증가되나, 각종 공격에 강인하지 못하다. 제한한 방법에서는 그림 4 (a)에서와 같이 단위원의 구역 개수  $n$ 을 4로 하여, 각 변환 노드 상에서 CoordIndex 필드 내에 4 비트를 삽입한다. 이동 인자  $\alpha$ 는 견고성 및 비가시성을 고려하여 실험적으로 0.5로 결정되었다.

## 2. 보간기 워터마킹

PositionInterpolator 필드는 각 키 시간에서 물체의

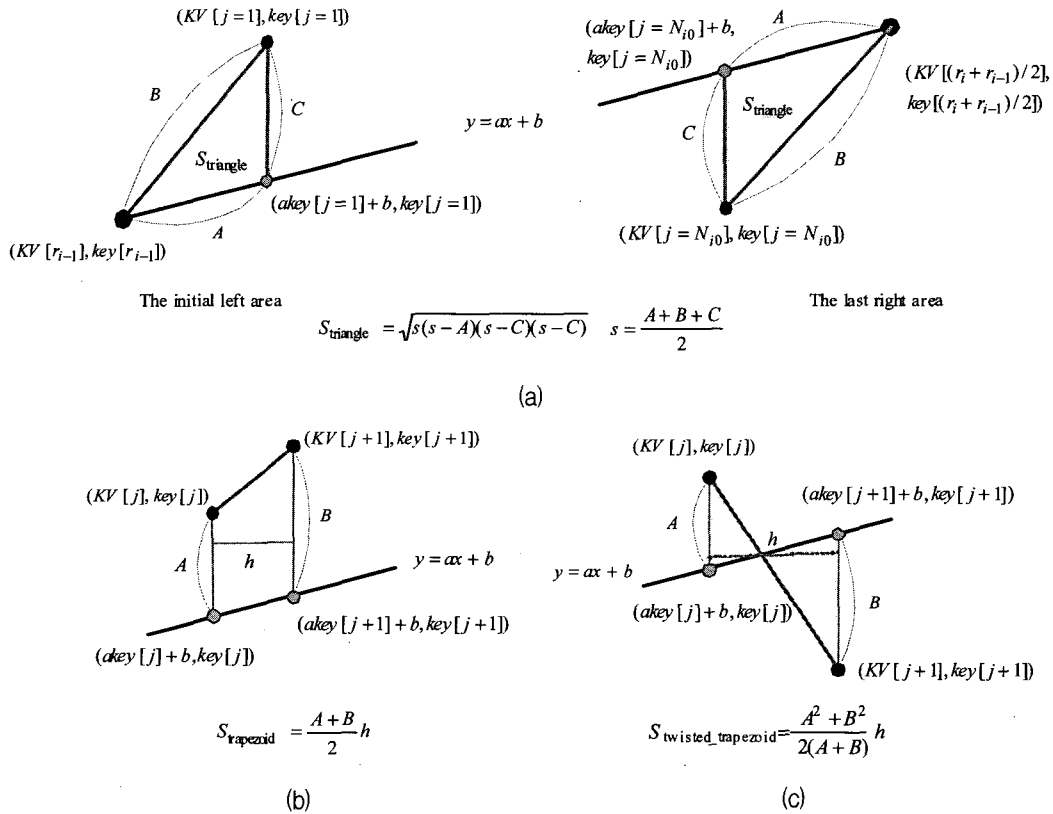


그림 6.  $S_{i0}$ 에서 기준선 및 이동선과의 (a) 첫 번째 및 마지막 구간에서 삼각형 면적  $S_{\Delta,first}$  및  $S_{\Delta,last}$ , (b) 사다리꼴 면적  $S_{trapezoid}$  및 (c) 꼬인 사다리꼴 면적  $S_{twisted-trapezoid}$

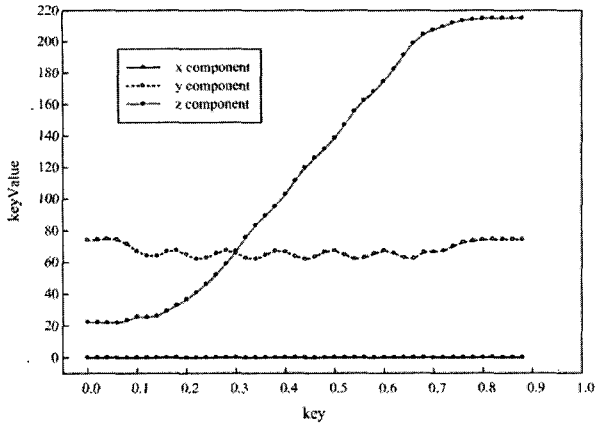
Fig. 6. (a) Triangle area  $S_{\Delta,first}$  and  $S_{\Delta,last}$  within the first and the last range, (b) trapezoid area  $S_{trapezoid}$  and (c) twisted trapezoid area  $S_{twisted-trapezoid}$  between the reference line and the moving line in  $S_{i0}$ .

운동 위치를 나타내는 키 값으로 구성된다. 이 때 키 값은 3D 직각 좌표계의 (x,y,z)으로 나타낸다. 제안한 보간기 워터마킹에서는 먼저 기하학적 워터마킹에서와 같이 계층적 구조 상에서 변환 노드를 랜덤하게 선택한다. 그리고 이진 워터마크는 선택된 변환 노드 상에서 PositionInterpolator 필드 내 키 값의 각 성분 에 각각 삽입된다.

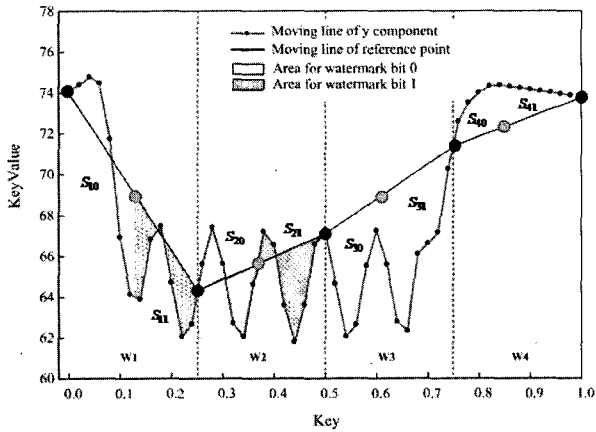
그림 5 (a)는 하나의 변환 노드 내 PositionInterpolator 필드의 각 성분을 보여주며, 이 때 각 성분은 0과 1 사이 45개의 키 시간 동안의 키 값을 나타낸다. 그림에서와 보는바와 같이, x 축 상의 값이 거의 변화가 없으므로, x 값에 대한 속도는 거의 0이다. 그러나 y 값에 대한 속도는 매우 크게 변하고 있고, z 축에 대한 속도는 서서히 커지나 속도 변화는 크지 않다. 제안한 보간기 워터마킹에서는 비지각적인 운동 변화를 위하여 운동 변화가 있는 성분에 워터마크를 삽입한다. 즉, 0과 1 사이 각 성분의 속도 분산이

$\sigma = \sqrt{\sum_{i=0}^{N_{key}} (v_i - \bar{v})^2 / N_{key}} > 0.5$ 을 만족하는 성분에 워터마크를 삽입한다. 여기서  $\bar{v}$ 는 평균 속도이며,  $N_{key}$ 는 키의 개수이다. 기하학적 워터마킹에서와 같이  $N$  비트의 워터마크 중  $n$  비트씩 각 변환 노드의 PositionInterpolator 필드에 삽입한다.

각 성분에  $n$  비트를 삽입하기 위하여 키  $k(0 \leq k \leq 1)$ 를  $n+1$ 개의 기준점  $r_{i \in [0,n]}$ 에 의하여  $n$  등분한다. 여기서  $r_0 = 0$ 이고  $r_n = 1$ 이다. 본 논문에서는 등분된 영역을  $W_{i \in [1,n]}$ 로 하고,  $i$ 번째 키  $k_i$ 에 대한 키 값을  $k_i = (x_i, y_i, z_i)$ 로 한다. 주어진 키 값의 각 성분에서 기준점  $r_i$ 이 없으면, 가장 인접한 두 키 값들의 선형 보간에 의하여  $r_i$ 을 생성한다. 이 때  $n+1$ 개의 기준점  $r_{i \in [0,n]}$ 은 워터마크 키로 사용된다. 그림 5 (b)에서는 그림 5 (a)의 PositionInterpolator 필드 중 속도 변화가 큰 y 성분이 5개의 기준점  $r_{i \in [0,4]}$ 에 의하여 4 등분



(a)



(b)

그림 5. (a) 3DS-MAX에 제공된 Wailer 애니메이션의 Bip 변환 노드 상에서 45개의 키 값을 가지는 PositionInterpolator 및 (b) y 성분에 면적 차이를 이용한 4비트의 워터마크 삽입

Fig. 5. (a) PositionInterpolator in Bip transform node of Wailer animation that provided in 3DS-MAX. The number of keys is 45 and (b) Embedding 4bit watermark into y component by using area difference.

된 후, 각 등분된 영역에 4 비트 워터마크를 삽입하기 위한 면적을 보이고 있다.

제안한 방법에서는  $r_{i-1} \leq k \leq r_i$ 인 영역  $W_i$ 을  $[r_{i-1}, (r_i + r_{i-1})/2]$  및  $[(r_i + r_{i-1})/2, r_i]$ 의 두 구간으로 분할한 후, 분할된 영역  $S_{i0}$  및  $S_{i1}$ 의 면적 차이  $\Delta S_i = S_{i1} - S_{i0}$ 에 임의의  $i$ 번째 워터마크 비트  $w_i$ 를 삽입한다. 이때 영역  $W_i$ 에서의 면적  $S_i$ 는 두 점  $(r_{i-1}, k_{r_{i-1}})$  및  $(r_i, k_{r_i})$ 을 잇는 기준선과 원래 키 값의 이동선  $k_j (r_{i-1} < j < r_i)$ 와의 차이에 의하여 구하여진다. 여기서 기준선은

$$y = ax + b, \tag{9}$$

$$\text{where } a = \frac{k_i - k_{i-1}}{k_i - k_{i-1}}, b = \frac{k_i k_{i-1} - k_{i-1} k_i}{k_i - k_{i-1}}$$

이며,  $x, y$ 는 키  $k$ 와 키 값  $k$ 의 변수들이다. 분할된 영역의 면적  $S_{i0}$  및  $S_{i1}$ 는 구간  $[r_{i-1}, (r_i + r_{i-1})/2]$  및  $[(r_i + r_{i-1})/2, r_i]$ 내에서의 키  $j \in [i, N_{i0}]$  및  $j \in [N_{i0} + 1, N_{i1}]$  동안의 기준선과 이동선과의 면적이다. 즉,  $S_{i0}$  또는  $S_{i1}$ 의 면적은

$$S_{i0} (\text{or } S_{i1}) = S_{\Delta, first} + S_{\Delta, last} + \sum_i S_{trapezoid} + \sum_j S_{twisted-trapezoid} \tag{10}$$

와 같이 두 직선의 첫 번째 및 마지막 구간에서의 삼각형 면적  $S_{\Delta, first}$  및  $S_{\Delta, last}$ 와 중간 구간에서의 사다리꼴 면적  $S_{trapezoid}$  및 꼬인 사다리꼴 면적  $S_{twisted-trapezoid}$ 들의 합이다.  $S_{i0}$ 에서의  $S_{\Delta, first}$ ,  $S_{\Delta, last}$ ,  $S_{trapezoid}$  및  $S_{twisted-trapezoid}$ 의 면적은 그림 6에서와 같이 구하여진다.

영역  $W_i$ 에 삽입될 워터마크 비트  $w_i$ 가 0이면,  $\Delta S_i = S_{i1} - S_{i0} < 0$ 이 되도록  $[1, N_{i0}]$  구간 내의 키 속도를 증가시켜

$$\begin{aligned} v'_{i0,j} &= \alpha \times v_{i0,j}, j \in [1, N_{i0}] \\ \frac{k'_{i0,j} - k'_{i0,j-1}}{k_{i0,j} - k_{i0,j-1}} &= \alpha \times \frac{k_{i0,j} - k_{i0,j-1}}{k_{i0,j} - k_{i0,j-1}} \end{aligned} \tag{11}$$

와 같이  $S_{i0}$ 의 면적을 넓히는 반면,  $[N_{i0} + 1, N_{i1}]$  구간 내의 키 속도를 줄여

$$\begin{aligned} v'_{i1,j} &= \frac{1}{\alpha} \times v_{i1,j}, j \in [N_{i0} + 1, N_{i1}] \\ \frac{k'_{i1,j} - k'_{i1,j-1}}{k_{i1,j} - k_{i1,j-1}} &= \frac{1}{\alpha} \times \frac{k_{i1,j} - k_{i1,j-1}}{k_{i1,j} - k_{i1,j-1}} \end{aligned} \tag{12}$$

$S_{i1}$ 의 면적을 줄인다. 이를 다시 요약하면,

$$k'_{i0,j} = \alpha \times k_{i0,j} + (1 - \alpha) \times k_{i0,j-1}, j \in [1, N_{i0}] \tag{13}$$

$$k'_{i1,j} = \frac{1}{\alpha} \times k_{i1,j} + (1 - \frac{1}{\alpha}) \times k_{i1,j-1}, j \in [N_{i0} + 1, N_{i1}] \tag{14}$$

와 같다. 영역  $W_i$ 에 삽입될 워터마크 비트  $w_i$ 가 1이면,  $\Delta S_i = S_{i1} - S_{i0} > 0$ 이 되도록 한다.

### 3. 워터마크 추출

제안한 방법에서는  $N$  비트 길이를 가지는 워터마크 중  $n$  비트를 임의의 변환 노드 상의 꼭지점 좌표 및 PositionInterpolator의 키 값에 각각 삽입한다. 따라서 워터마크 추출 과정에서는 선택된 변환 노드의 주소 및 PositionInterpolator의 기준점이 필요하다.

먼저 기하학적 워터마킹에서의 워터마크 추출 과정에서는 선택된 변환 노드 상의 모든 꼭지점 좌표들을 2D 단위 원 내에 투영한다. 그런 다음 단위 원 내에 등분된 구역의 중간점  $\widehat{c_{k \in [1, n]}} = \widehat{c_{kx}} X_{local} + \widehat{c_{ky}} Y_{local}$  들을 구한다. 각 구역에 삽입된 워터마크 비트  $w_k$ 는 중간점의 위상  $\theta_k$

$$\theta_k = \tan^{-1}\left(\frac{\widehat{c_{ky}}}{\widehat{c_{kx}}}\right), \frac{2(k-1)\pi}{n} \leq \theta_k \leq \frac{2k\pi}{n} \quad (15)$$

에 의하여

$$w'_k = \begin{cases} 0, & \text{if } \frac{2(k-1)\pi}{n} \leq \theta_k \leq \frac{2k\pi}{n} \\ 1, & \text{else } \frac{(2k-1)\pi}{n} \leq \theta_k \leq \frac{2k\pi}{n} \end{cases} \quad (16)$$

와 같이 추출된다.

보간기 워터마킹의 워터마크 추출 과정에서는 먼저 기준점  $r_{i \in [0, n]}$ 을 잇는 기준선과 공격된 애니메이션의 변환 노드 내에 기준점  $r'_{i \in [0, n]}$ 을 잇는 기준선과 비교한다. 이들 두 선들이 일치하면, 키의 재배열 과정없이 워터마크를 추출한다. 그러나 키 타임의 확대 및 축소 또는 키의 삽입 및 삭제의 경우에는 이들 두 선들이 일치하지 않는다. 이 때 이들 두 선들이 일치하도록  $r'_{i \in [0, n]}$ 를 변경하는 키의 재배열 과정을 수행한다. 그런 다음 선택된 변환 노드의 영역  $W_k$ 내에 분할된 두 구간의 면적  $S_{k0}$  및  $S_{k1}$ 를

$$w'_k = \begin{cases} 0, & \text{if } S_{k0} \geq S_{k1} \\ 1, & \text{else } S_{k0} < S_{k1} \end{cases} \quad (17)$$

와 같이 비교하여 워터마크 비트  $w'_k$ 를 추출한다.

### III. 실험 결과

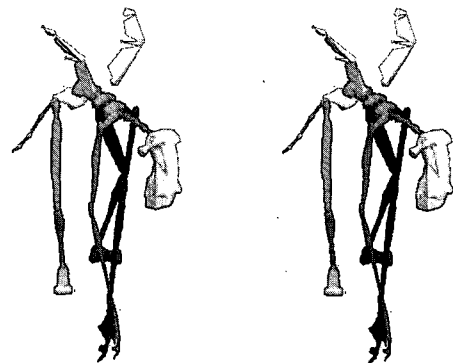
본 논문에서는 제안한 방법의 성능 평가를 위하여 3DS-MAX 테스트 애니메이션으로 제공되는 Wailer를 VRML 데이터로 변환하여 이를 테스트 모델로 사용하였다. Wailer는 76개의 변환 노드와 100개 프레임들로 구성되어 있고, 각 변환 노드 내에 키는 [0 1] 사이에

다양한 개수들로 이루어져 있다. 본 실험에서는 CoordIndex 노드를 가지는 변환 노드 중 25개를 랜덤하게 선택한 후, 100 비트의 워터마크를 4비트씩 각 변환 노드의 꼭지점 좌표 및 PositionInterpolator의 키 값에 각각 삽입하였다. 본 실험에 사용된 성능 평가 척도로는 3D 애니메이션 공격에 대한 견고성 및 워터마크의 비가시성이다.

#### 1. 비가시성

최근 3D 그래픽 모델에 대한 화질 척도에 대한 연구가 진행되고 있으며, 대표적인 화질 척도로 Hausdorff 거리를 이용한 MESH가 있다. 그러나 MESH는 키 프레임 기반의 3D 애니메이션에서는 적용되지 못한다. 따라서 본 실험에서는 간단한 비가시성의 화질 척도로 꼭지점 좌표 및 키 값들에 대한 SNR (Signal-to-Noise Ratio)을 사용하였다.

여기서 SNR은  $SNR = 10 \log_{10} \frac{\text{var}(\|a - \bar{a}\|)}{\text{var}(\|a - a'\|)}$  이고,  $a$  및  $a'$ 는 원 애니메이션의 변환 노드 및 워터마크된 애니메이션의 변환 노드 상에 꼭지점 좌표값 또는 PositionInterpolator 내 키 값의 좌표값을 나타낸다.  $\bar{a}$ 는  $a$ 의 평균이며,  $\text{var}(a)$ 는  $a$ 의 분산이다. 워터마크된 Wailer 애니메이션의 변환 노드 상에 꼭지점 좌표에 대한 평균 SNR이 38.8 dB이고, PositionInterpolator 내 키 값의 좌표값에 대한 평균 SNR은 39.1 dB이다. 이는 워터마크가 삽입된 변환 노드 상에 대한 평균 SNR이다. 전체 애니메이션을 구성하는 모든 변환 노드의 평균 SNR은



(a)

(b)

그림 7. (a) Wailer의 첫 번째 프레임 (0번째 키 프레임 모델) 및 (b) 워터마크가 삽입된 Wailer의 첫 번째 프레임

Fig. 7. (a) The first frame (0th. key-frame model) of original Wailer animation and (b) watermarked Wailer animation.



꼭지점 좌표에 대하여 39.5 dB이고 PositionInterpolator 내 키 값의 좌표값에 대하여 42 dB이다. 이는 객관적 화질 척도로 우수한 비가시성을 가지고 있다고 판단된다. Wailer의 원 애니메이션의 첫 번째 프레임 모델 및 워터마크가 삽입된 첫 번째 프레임을 모델은 그림 7에서와 같다. 이 그림들을 살펴보면, 워터마크가 비가시적으로 삽입됨을 알 수 있다.

2. 견고성

3DS-MAX, Maya 등 3D 애니메이션을 편집할 수 있는 범용 도구들이 많이 개발되고 있다. 본 논문에서는 범용 도구들 상에서 사용되는 기능을 크게 기하학적 공격, 타임라인 공격 및 파일 형식 변환으로 분류하여 이를 견고성 실험에 사용하였다. 기하학적 공격은 IndexFaceSet 노드 상에 꼭지점 좌표 또는 연결 정보를 바꾸는 것으로 구부림 (Bend), Taper, 잡음, 패치 변형 (Patch Deform), 메쉬 뭉롱화 (Mesh smooth), HSDS 변경, 다각형 절단(cutting)/분배(divider)/분출(extrude), 꼭지점 제거, 회전, 스케일링, 이동 등이 대표적 기하학적 공격이다. 타임라인 공격은 타임 축 또는 키의 수를 확장 및 축소하는 것이다. 파일 형식 변환에는 3D 압축 알고리즘 또는 ASE, DWG, DXF, M3G, STL, W3D 등과 같은 많은 파일 형식의 변환이 있다. 본 실험에서는 3DS-MAX 툴을 이용하여 기하학적 공격 및 타임라인 공격에 대한 견고성 성능을 평가하였다.

제안한 방법은 워터마크를 선택된 변환 노드 상에 CoordIndex 노드의 꼭지점 좌표 분포 및 Position-Interpolator 노드의 키 값 분포에 각각 삽입한다. 즉, 워터마크가 삽입된 애니메이션이 기하학적 공격에 의하여 변형되더라도 PositionInterpolator 노드의 키 값에 삽입된 워터마크를 오류없이 추출이 가능하다. 이와는 반대로 타임라인 공격에 의하여 물체의 움직임 위치가 변형되었을 때에는 CoordIndex 노드의 꼭지점 좌표에 삽입된 워터마크를 오류없이 추출할 수 있다. 기하학적 공격 및 타임라인 공격에 대한 견고성 실험 결과는 표 1에서와 같다. 실험 결과에 설명 및 표에 나타난 변수에 대한 설명은 다음 절에 자세히 나타내었다.

2.1 기하학적 공격

워터마크된 애니메이션의 모든 변환 노드 상에서는 여러 가지 기하학적 공격에 의하여 편집되어질 수 있다. 기하학적 공격된 애니메이션에서는 변환 노드의 CoordIndex 필드 내의 값들이 바뀌어진다. 따라서 기하

표 1. 기하학적 공격 및 타임라인 공격에 대한 견고성 실험 결과

Table 1. Experimental results for the robustness against geometrical and timeline attacks.

공격	변수	BER(Bit Error Rate)		
		CoordIndex 노드	Position-Interpolator 노드	
기하학적 공격	구부림 (Bend)	(90,22,z)	0.07	-
	점감 (Taper)	(1.2,0.5,z,xy)	0.06	-
	잡음	(29,200,1,6,2,2,)	0.05	-
	메쉬 뭉롱화 (Mesh Smooth)	(1,1,0)	0.05	-
	다각형 절단 (Polygon Cutting)	40%	0.21	-
	다각형 분출 (Polygon Extrude)	50%	0.18	-
	꼭지점 제거	50%	0.25	-
타임라인 공격	프레임수 감소 (Down-scaling time)	50 frame	-	0.10
	프레임수 증가 (Up-scaling time)	200 frame	-	-
	키 삽입	20 keys	-	0.03
	키 제거	20 keys	-	0.03
	운동 변화 (Motion change)		-	0.30

학적 공격 실험에서는 CoordIndex 필드 내에 삽입된 워터마크의 BER에 의하여 견고성을 확인하였다.

구부림 (Bend)은 구부림 축 (Bend axis)을 기반으로 방향 (direction) 및 물체의 각도 (angle)의 변수들 (angle, direction, bend axis)에 의하여 애니메이션의 모델을 구부린다. 구부림 공격 실험에서는 그림 8 (a)에서와 같이 워터마크된 Wailer를 임의의 변수 (90, 22, z)에 의하여 구부린 후 표 1에서와 같이 CoordIndex 필드 상에서 워터마크를 추출하였다. 이 때 추출된 워터마크의 BER은 0.07이다. 점감 (Taper)은 (amount, curve, primary, effect)의 변수에 의하여 점감 축 (Taper axis)을 기준으로 블록 또는 오목 형태로 만든다. 변수 amount 및 curve는 모델의 점감되는 양 및 형태를 나타낸다. 변수 primary는 점감되는 중심축이며, 변수 effect는 점감되는 형태인 curve의 방향을 나타낸다. (1.2, 0.5, z, xy)에 의하여 점감된 Wailer는 그림 8 (b)에서와 같으며, 이 때 CoordIndex 필드 상에서 추출된 워터마크의 BER은 0.06이다. 잡음 (Noise)은 (seed, scale, roughness, iterations, strength x, y, z) 변수에 의하여 각 seed 축 상에서의 잡음 세기 (strength), 스케일 (scale) 및 거친 정도 (roughness)에 따라 반복적

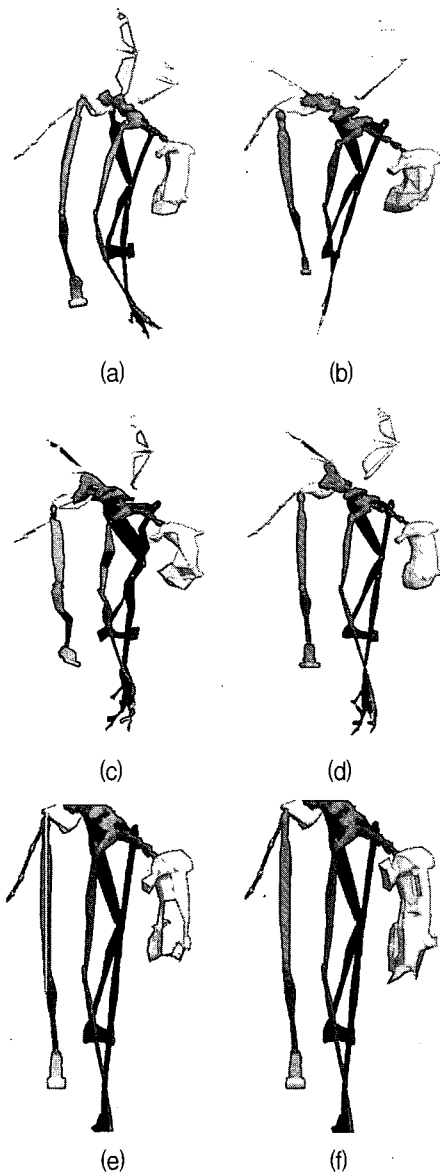


그림 8. 모든 변환 노드 상에서 (a) 구부림, (b) 점감 및 (c) 잡음이 가해진 그리고 머리 변환 노드 상에서 (e) 다각형 절단 및 (f) 다각형 분출이 가해진 워터마크된 Wailer

Fig. 8. Watermarked Wailers attacked by (a) Bend, (b) Taper and (c) Noise in all transform nodes and attacked by (e) polygon cutting and (f) polygon extrude.

으로 애니메이션 모델의 형상을 불규칙하게 만든다. (29, 200, 1.6, 2.2, 2)에 의하여 잡음이 첨가된 Wailer는 그림 8 (c)에서와 같으며, 이 때 CoordIndex 필드 상에서 추출된 워터마크의 BER은 0.06이다. 메쉬 뭉뚱화 (Mesh smooth)는 (iteration, smoothness) 변수에 의하여 다각형 표면을 분할함으로써 애니메이션 모델의 형상을 뭉뚱하게 만든다. 모든 변환 노드 상에서 (1, 1.0)으로 메쉬 뭉뚱화된 Wailer는 그림 8 (d)에서와 같으며,

이 때 CoordIndex 필드 상에서 추출된 워터마크의 BER은 0.05이다. 임의의 변환 노드 상에서 40-50% 정도의 다각형 또는 꼭지점들을 분출 및 절단하였다. 이 때 추출된 워터마크의 BER은 0.18-0.25이다. 머리 변환 노드 상에서 다각형 절단 및 분출된 Wailer는 그림 8 (e) 및 (f)에서와 같다.

## 2.2 타임라인 공격

타임라인 공격된 애니메이션에서는 보간기의 키와 키 값이 바뀌어진다. Re-scale 타임에서는 프레임 수와 함께 타임 길이를 조정한다. 애니메이션 타임이 원래의 타임 길이에 비하여 반으로 줄어지면, 그림 9 (a)에서와 같이 보간기의 키 수가 반으로 줄어지고, 키가 [0 1] 내에 재정렬된다. 제안한 워터마크 추출 방법에서는 워터마크 삽입 과정에서 저장된 타임라인 상의 기준점에 의하여 워터마크를 추출한다. 그러나 키의 수가 줄어든 변환 노드의 PositionInterpolator 상에서는 몇몇 기준점들이 없어질 수 있다. 제안한 방법에서는 이들 없어진 기준점들을 주위의 존재하는 기준점들의 보간에 의하여 생성하는 전처리 과정을 수행한 후 워터마크를 추출한다. 키 수가 반으로 줄어든 애니메이션 상에서 추출된 워터마크의 BER은 표 1에서와 같이 0.10이다. 이는 각 변환 노드의 PositionInterpolator 상에 랜덤 치환된 4개의 워터마크 비트를 각각 x,y,z 축의 키 값에 삽입하였으므로, 약 90%의 워터마크를 추출할 수 있다. 애니메이션 타임이 원래의 타임 길이보다 두 배로 확장된 PositionInterpolator의 각 성분들에서 워터마크 추출을 위한 기준점들이 모두 존재하므로, 전처리 과정이 필요 없으며 표 1에서와 같이 워터마크가 오류없이 추출된다. 키 삭제 및 첨가 실험에서는 모든 변환 노드의 보간기 상에서 임의의 20개 키를 삭제 또한 첨가하였다. 이 때 이동선의 변경 정도에 따라 면적 차이 또한 바뀌어지므로, 전처리 과정을 수행한 후 추출된 워터마크의 BER은 약 0.03이다. 운동 변경 실험에서는 그림 9 (b)에서와 같이 랜덤하게 생성된 35개의 키를 가지는 PositionInterpolator  $M'_{TR}$ 를 기준으로 애니메이션의 운동 위치를 변경하였다.

이때  $V'_{world} = V_{local} \times \hat{M}'_{TR}$ 이고,  $\hat{M}'_{TR} = M_{TR} \times M'_{TR}$ 이다. 여기서 원래의 PositionInterpolator  $M_{TR}$ 는 알지 못하므로, 워터마크는 변경된 PositionInterpolator  $\hat{M}'_{TR}$  내에서 추출되어야 한다. 추출된 워터마크의 BER은 표 1에서와 같이 약 0.30으로 약 70%정도 워터마크가 오류

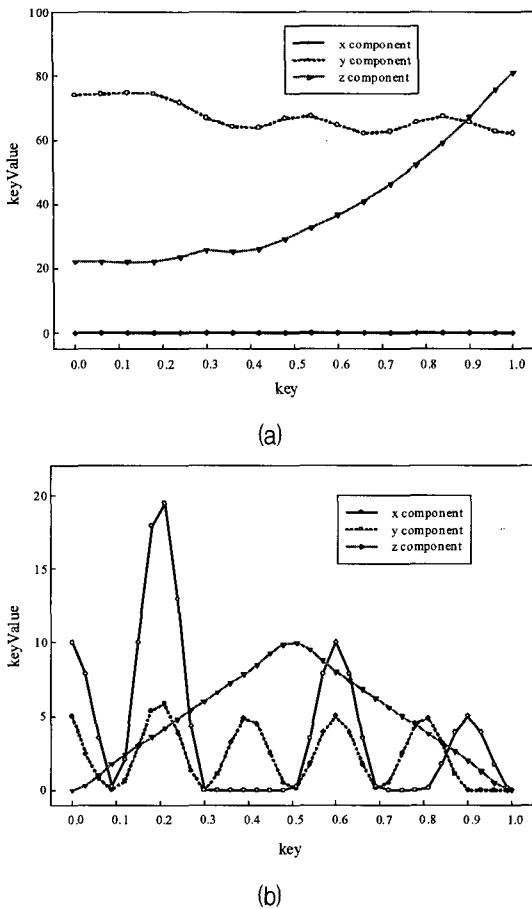


그림 9. (a) 50개의 키 및 (b) 운동 변화를 가지는 Bip 변환 노드 상에서 PositionInterpolator의 x,y,z 성분들

Fig. 9. x,y,z components in PositionInterpolator of Bip transform node with (a) 50 keys and (b) PositionInterpolator for motion change.

없이 추출된다.

이상과 같이 본 실험에서는 제안한 방법에 의하여 삽입된 워터마크는 우수한 비가시성을 가지며, 대표적인 기하학적 및 타임라인 공격에 대하여 우수한 견고성을 가짐을 확인하였다.

#### IV. 결 론

VRML 기반의 3D 키프레임 애니메이션은 계층 구조 상에서 각 메쉬 모델의 변환 노드, 변환 노드 상에서의 보간기 내에 키 또는 키 값, 그리고 coordIndex 필드의 기하학적 성질로 구성되어 있다. 본 논문에서는 CoordIndex 필드 내에 꼭지점 좌표값을 이용한 기하학적 워터마킹과 PositionInterpolator의 키 값을 이용한 보간기 워터마킹 기법을 제안하였다. 기하학적 워터마킹에서는 임의의 선택된 변환 노드 상 CoordIndex 필드

내에 모든 꼭지점 좌표를 4개로 분할된 2D 단위원 내에 투영시킨 후, 분할된 영역 내에 워터마크를 1비트씩 삽입한다. 보간기 워터마킹에서는 임의의 선택된 변환 노드 상 PositionInterpolator 필드 내에 키 값의 x,y,z 성분을 4개의 타임으로 등분한 후, 등분된 영역 내에 기준선과 이동선과의 차이 면적에 워터마크를 1비트씩 각각 삽입한다. 실험 상에서 구부림, 점감, 잡음, 메쉬 뭉통화, 다각형 편집 등과 같은 각종 기하학적 공격 및 프레임 수 증가 및 감소, 키 삭제 및 제거 등 타임라인 공격에 대하여 워터마크가 70%-80%이상 오류없이 검출함을 확인하였다. 또한 워터마크 삽입에 의하여 변경된 좌표 값 및 키 값에 대한 SNR이 40dB 이상임을 확인하였다.

#### 참 고 문 헌

- [1] J. Cox, J. Kilian, T. Leighton, and T. Shamoan, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. on Image Processing*, vol. 6, no. 12, pp. 1673-1687, Dec. 1997.
- [2] W. Zhu, Z. Xiong, and Y.-Q. Zhang, "Multi-resolution watermarking for image and video," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 545-550, June 1999.
- [3] R. Ohbuchi, H. Masuda, and M. Aono, "Watermarking Three-Dimensional Polygonal Models Through Geometric and Topological Modification," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 551-560, May 1998.
- [4] O. Benedens, "Geometry-Based Watermarking of 3D Models," *IEEE Computer Graphics and Applications*, vol. 19, no. 1, pp. 46-55, Jan./Feb. 1999.
- [5] K.-R. Kwon, S.-G. Kwon, and S.-H. Lee, "3D Watermarking Shape Recognition System Using Normal Vector Distribution Modelling," *Lecture Notes in Computer Science*, vol. 3397-9743, pp. 481-490, Feb. 2005.
- [6] S.-H. Lee and K.-R. Kwon, "Watermarking for 3D Mesh Model Using Patch CEGIs," *Lecture Notes in Computer Science*, vol. 3481, pp. 557-566, May 2005.
- [7] 전정희, 호요성, "3차원 메쉬 모델의 적응형 워터마킹 방법," 대한전자공학회논문지, 제40권 SP편 제6호, pp. 41-50, 2003년 11월.
- [8] 이석환, 권기룡, "패치 CEGI를 이용한 메쉬 워터마킹," 대한전자공학회논문지, 제42권 CI편 제1호, pp. 67-78, 2005년 1월.
- [9] 이석환, 권기룡, "볼록 집합 투영 기법을 이용한

3D 메쉬 워터마킹," 대한전자공학회논문지, 제43권 CI편 제2호, pp. 81-92, 2005년 3월.

- [10] ISO/IEC 14772-1, The virtual reality modeling language.
- [11] E.S. Jang, James D.K.Kim, S.Y. Jung, M.-J. Han, S.O. Woo, and S.-J. Lee, "Interpolator Data Compression for MPEG-4 Animation," *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 14, no. 7, pp. 989-1008, July 2004.

---

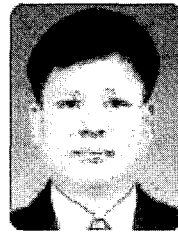
저 자 소 개

---



이 석 환(정회원)  
1999년 경북대학교  
전자공학과 학사 졸업.  
2001년 경북대학교  
전자공학과 석사 졸업.  
2004년 경북대학교  
전자공학과 박사 졸업.

2005년~현재 동명대학교 정보보호학과 전임강사  
<주관심분야 : 워터마킹, DRM, 영상신호처리>



권 기 룡(정회원)-교신저자  
1986년 경북대학교  
전자공학과 학사 졸업.  
1990년 경북대학교  
전자공학과 석사 졸업.  
1994년 경북대학교  
전자공학과 박사 졸업

2000년~2001년 Univ. of Minnesota, Post-Doc  
1996년~2005년 부산외국어대학교 디지털정보  
공학부 부교수  
2006년~현재 부경대학교 전자컴퓨터정보통신  
공학부 부교수  
<주관심분야 : 멀티미디어 정보보호/통신>