

모빌의 디지털 재현 (Digital Reproduction of Mobiles)

이동춘^{*} 이남경^{**} 정대현^{**} 김창태^{***} 이동규^{***}
(Dong C. Lee) (Nam Kyung Lee) (Dae Hyun Jung) (Chang T. Kim) (Dong K. Lee)
배희정^{****} 백낙훈^{****} 이종원^{*****} 유관우^{*****}
(Hee Jung Bae) (Nakhoon Baek) (J. Won Lee) (Kwan Woo Ryu)

요약 최근에는 회화와 같은 미술 작품들을 디지털 형태로 재현하려는 시도들이 활발해지고 있다. 디지털 형태로 재현된 회화 작품들은 저장하거나 전송하기에 편리한 장점을 가진다. 반면에, 알렉산더 칼더(Alexander Calder)가 창시한 모빌, 즉 움직이는 조각은 그 특성상 기존의 방법으로는 컴퓨터에서 재현하기가 까다롭다. 모빌은 바람과 같은 외부의 힘에 의해 각각의 조각이 움직이도록 제작되어 있으므로, 사진이나 정지 영상과 같은 고정된 형태로는 작가의 제작 의도를 충분히 전달하기에 미흡하다. 본 논문에서는 컴퓨터 상에서 재현된 모빌을 사용자들이 사실적으로 감상할 수 있는, 물리 기반의 가상 모빌 시스템을 제안한다. 실제계의 모빌을 3차원 모델로 구성한 후, 가상의 바람을 생성하여 그 모빌이 움직일 수 있도록 하였다. 모빌의 움직임은 운동학 방법들을 사용하여 사실적으로 표현한다. 기존의 제약조건을 가지는 동역학(constraint dynamics)방법과 충격역학(impulse dynamics)방법들을 모빌의 구조에 적합하도록 개선하여, 모빌의 움직임을 PC 환경에서 대화형으로 재현하였다. 본 논문에서 제안하는 속도 향상 방법들은 모빌과 유사한 위상 구조를 가지는 3차원 모델들에도 적용될 수 있다.

Abstract Recently, there are many attempts to reproduce real world fine art pieces in digital forms. The digital representations are convenient to store and/or transmit. In contrast, mobiles, or moving sculptures, such as those designed by Alexander Calder cannot be reproduced realistically by usual reproduction techniques. Since mobiles are originally designed to generate motions in response to external forces applied to it, people could not fully enjoy them through photographs or static images. We present a virtual mobile system where users can easily control the mobile and can feel the impressions that the artist originally intended to provide. A real-world mobile is reconstructed in a three-dimensional physically-based model, and then virtual wind is generated to give motions to it. The motions of the mobile are generated by constraint dynamics and impulse dynamics techniques, which are modified to fully utilize the characteristics of the mobile, and finally give interactive displays on the PC platforms. The techniques presented can easily be extended to simulate other interactive dynamics systems.

1. 서론

컴퓨터 그래픽스와 가상 현실 기술의 발달에 따라 컴퓨터 화면에서 실제계의 물체들을 사실적으로 표현할 수

있게 되었다. 최근에는 기존의 회화 작품들을 디지털 형태로 재현하여 모아놓은 디지털 박물관(digital museum)까지 등장하였다[1]. 평면에 표현되는 그림의 경우에는

* 본 연구는 부분적으로 2001년도 동국대학교 신입교원 연구비의 지원을 받았습니니다.

^{*} 비 회 원 : 한국전자통신연구원 중앙현실연구팀 연구원
bluepine@etri.re.kr
^{**} 비 회 원 : 경북대학교 컴퓨터공학과
lnk@comeng.ce.knu.ac.kr
shadow@comeng.ce.knu.ac.kr
^{***} 비 회 원 : (주)KOG 기술연구소 연구원
ctkim@kogsoft.com
tpfma@dreamwiz.com

^{****} 학 생 회 원 : 경북대학교 컴퓨터공학과
hjbae@dreamwiz.com
^{****} 정 회 원 : 동국대학교 컴퓨터공학과 교수
nhbaek@dgu.ac.kr
^{*****} 정 회 원 : 미국조지워싱턴대학교 Computer Science
wonlcc@kogsoft.com
^{*****} 종 신 회 원 : 경북대학교 컴퓨터공학과 교수
kwryu@bh.knu.ac.kr
논문접수 : 2000년 7월 7일
심사완료 : 2001년 7월 2일

이미지 스캐너(image scanner)나 디지털 카메라를 사용하여 간단히 컴퓨터 화면상에 재현이 가능하다. 조각 작품들은 3차원의 기하학 정보나 3차원 스캐너에 의한 볼륨 데이터를 이용하여 재현할 수 있다[2]. 또, MCOP (multiple center of projection)과 같은 이미지 기반 렌더링 방법들을 사용하여 조각품을 재현할 수도 있다[3].

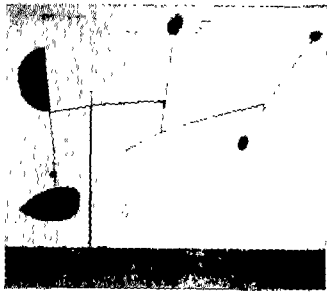


그림 1 실존 모델 "Steel fish"

반면에, 움직이는 조각으로 알려져 있는 모빌(mobile)의 경우에는 이들과는 다른 새로운 방법이 필요하다. 그림 1에 제시된 알렉산더 칼더의 "Steel Fish"라는 작품에서와 같이, 모빌은 하나의 독립적인 동역학 시스템(dynamics system)이다. 즉, 여러 개의 물체들이 공중에 매달려 있고, 바람과 같은 외부의 힘에 의해 조각 자체가 움직이는 과정이 모빌의 제작 의도이다. 작품을 감상하는 입장에서는 실시간으로 이루어지는 모빌과의 교감이 가장 중요한 요소가 된다.

본 논문에서는 물리 기반의 가상 모빌 시스템을 제안한다. 컴퓨터 상에서 모빌의 움직임을 재현하고자 하는 이러한 시도는 아직까지는 없었던 것으로 보인다. 실제의 모빌을 컴퓨터 상에서 재현하기 위해서는 우선 기하학적 모형을 만들어야 한다. 실시간으로 모빌을 화면에 표시하여야 하는 필요성 때문에 모빌의 외형을 삼각형의 집합으로 모델링하고, 모빌 내의 각 물체에 대한 질량(mass)과 관성 모멘트(moment of inertia) 등의 물리적 성질을 설정한다. 모빌의 움직임을 기하학적 정보와 물리적 성질들을 이용하여 제약조건을 가지는 동역학 방법(constraint dynamics method)으로 재현한다. 즉, 가상의 바람을 생성한 후, 이 바람이 모빌의 각 물체에 가하는 힘을 계산하고, 제약조건을 가지는 동역학 계산기(constraint dynamics solver)가 각 물체들의 움직임을 생성한다. 이 과정에서 생길 수 있는 물체를 간헐 충돌은 충격역학(impulse dynamics) 방법으로 해결한다.

본 논문은 일반적인 PC 환경에서도 대화형으로 모빌의 움직임을 재현하는 데에 목적을 둔다. 이를 위해서 공기역학(aerodynamics)에 기초하여 단순화된 가상 바람 모델을 사용한다. 또, 운동학 방법들에서 필연적으로 풀어야 하는 대형 연립방정식을 모빌의 위상 정보를 이용하여 선형 시간에 풀 수 있는 방법을 제시한다. 본 논문이 제시한 방법들은 단순히 모빌에만 한정된 것이 아니라 물리 기반의 실시간 시뮬레이션 분야에서 응용할 수 있다.

본 논문의 전체 구성은 다음과 같다. 2절에서 모빌 시스템의 전체 구성에 대해서 설명하고, 3절에서는 실제의 모빌로부터 가상의 모빌을 모델링하는 방법에 대해 기술한다. 4, 5, 6 절은 각각 본 논문에서 사용한 가상 바람 모델, 제약조건을 가지는 동역학 계산기, 충격역학 계산기(impulse dynamics solver)에 대해서 설명한다. 7절에서는 가상 모빌의 실제 움직임에 대한 예제들을 보이고, 마지막으로 8절에서 결론과 향후 과제를 제시한다.

2. 전체 구성

본 논문의 가상 모빌 시스템은 펜티엄 급의 PC에서 OpenGL 라이브러리를 이용하여 C++ 언어로 구현하였다. 그림 2는 가상 모빌 시스템의 전체 흐름도이다. 전처리 과정에서는 모델링하고자 하는 실존 모빌의 사진들로부터 3차원 기하학 정보를 유추하고, 이로부터 물리적인 제약조건(constraint)들과 각 물체의 물리적 성질들을 결정한다. 이 때, 실측한 자료를 사용하는 것도 한 방법이다. 그러나 모빌의 물리적 성질들은 대부분의 경우 모빌 자체를 파괴하지 않고는 측정이 불가능한 경우가 많기 때문에, 실존 모빌의 외형으로부터 추정하는 방법이 더 유용하다.

모빌의 움직임은 가상의 바람에 의해서 생성된다. 본 논문의 가상 모빌 시스템에서는 사용자가 실시간 대화

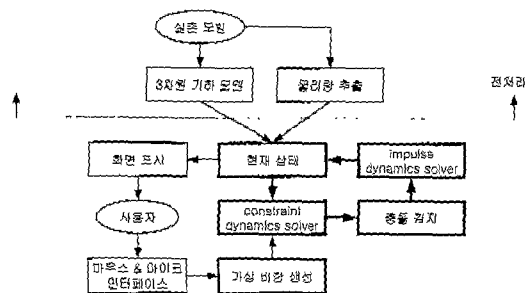


그림 2 전체 시스템의 순서도

형으로 가상의 바람을 만들어 낼 수 있도록 하고, 바람의 세기는 마이크를 이용하여 제어할 수 있도록 한다. 생성된 가상 바람은 모빌을 구성하는 각 물체에 힘으로 작용하게 되고, 각 물체에 대해서는 물체들 간의 연결 상태를 제약조건으로 가지는 제약 조건을 가지는 동역학(constraint dynamics) 방법과, 물체들 간의 충돌을 처리하기 위한 충격역학 방법을 이용하여 최종적으로 모빌의 움직임을 생성한다. 즉, 내부 물체들 간의 연결 상태에 따라 제약조건힘(constraint force)들이 계산되고, 제약조건을 가지는 동역학 계산기가 가상 바람에 의한 힘과 제약조건힘들로부터 각 물체의 새로운 위치를 계산한다. 이 때, 물체들 간의 충돌이 감지되면 충격역학 계산기가 이러한 충돌을 처리하기 위해 사용된다. 가상 모빌 시스템은 바람의 생성에서부터 물체의 새로운 위치를 계산하기까지의 과정을 반복적으로 수행하고, 매 반복시마다 가상 모빌의 움직임을 화면에 표시한다.

3. 물리량의 유추

본 논문에서는 가상 모빌 시스템에 사용할 실존 모빌로 알렉산더 칼더가 제작한 "Steel Fish"라는 작품을 사용하였다. 이 실존 모빌은 독립적으로 움직일 수 있는 9개의 물체들과 이들을 연결하는 자유도 3을 가지는 8개의 조인트들로 구성되어 있다. 운동학의 관점에서는 이들 8개의 조인트들이 제약조건으로 작용한다. 그림 3은 각각의 물체들과 조인트들을 나타내고 있다.

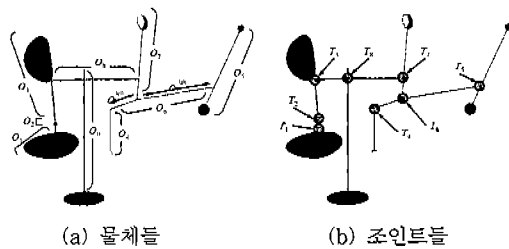


그림 3 가상 모빌의 물체들과 조인트들

운동학 방법으로 이 모빌의 움직임을 표현하려면, 가상 모빌을 구성하는 각 물체의 무게(mass), 무게 중심(center of mass), 관성 모멘트(moment of inertia) 등의 물리적 성질들이 필요하다. 이들은 실존하는 모빌로부터 직접 측정할 수도 있겠지만, 대부분의 실존 모빌은 분해가 불가능하다. 또, 실존하지 않는 모빌을 컴퓨터에서 표현하여야 할 경우도 있다. 따라서, 본 논문에서는 외부 힘이 가해지지 않을 경우에 모빌이 평형 상태에

있다는 사실과 각 물체의 기하학적 형태로부터 각 물체의 물리적 성질을 유추하는 방식을 취한다.

가상 모빌 시스템에서 사용할 모빌이 트리(tree) 형태이므로, 각 물체들간의 무게 비는 상향(bottom-up) 방식으로 계산할 수 있다. 예를 들어, 물체 O_1 와 O_5 의 무게 비는 이들을 연결하는 물체 O_6 에서의 길이 비, O_6^{left} 와 O_6^{right} 의 비례 관계로 계산할 수 있다. 이러한 기하학정보들로는 무게의 비례 관계만을 계산할 수 있기 때문에, 전체 모빌의 무게는 사용자가 조절할 수 있는 변수(controllable parameter)로 설정한다.

각 물체의 무게를 구한 후에는 무게 중심과 관성 모멘트를 계산하여야 한다. 본 논문에서는 각 물체가 균일한 밀도를 가지는 강체(uniform density rigid body)라고 가정하였고, 이 경우는 기하 정보를 이용하여 순차적으로 계산할 수 있다[4]. 먼저 각 물체의 기하학적 형태로부터 부피를 계산할 수 있고, 무게와 부피로부터 각 물체의 밀도가 구해진다. 밀도를 아는 경우에는 Mirtich가 제안한 적분 방정식들[4]을 이용하여 무게 중심과 관성 모멘트가 계산된다. 이들 물리량들과 기하 정보들은 가상 모빌의 운동학에 기초한 움직임을 만들어 내기 위해 사용된다.

4. 가상 바람의 생성

실세계에서는 자연 발생적인 바람에 의해서 모빌이 움직이지만, 가상 모빌 시스템에서는 가상의 바람으로 모빌을 움직여야 한다. 가상 바람의 생성에는 전통적인 입력 도구들인 키보드와 마우스가 사용될 수도 있지만, 가상 모빌 시스템에서는 추가로 마이크를 사용할 수도 있도록 설계하였다. 사용자가 마이크에 바람을 불면, 이때 생성되는 소리의 강도에 따라 가상 바람의 풍속이 제어되는 방식을 취하였다. 가상 바람의 방향은 마우스를 이용하여 설정한다.

마이크를 이용한 사용자 인터페이스는 몇 가지 장점을 가지는데, 우선 사용자가 실제로 바람을 부는 동작으로 바람을 생성하는 방식이 좀더 직관적이다. 또, 마이크를 통한 입력 방식은 풍속의 시간에 따른 변화를 비교적 쉽게 흉내낼 수 있다. 마이크 자체가 쉽게 구할 수 있고, 저가의 장비이기 때문에, 개인 사용자에게 적합하다는 점도 있다.

사람의 입으로 가상 바람을 생성하기 때문에, 가상 바람은 그림 4에서와 같이, 그 근원으로부터 원뿔 모양으로 퍼져나가도록 설계하였다. 바람의 근원은 반지름 r_0 의 원반 형태 S_0 로 설정하였고, 이 원반의 중점으로부터

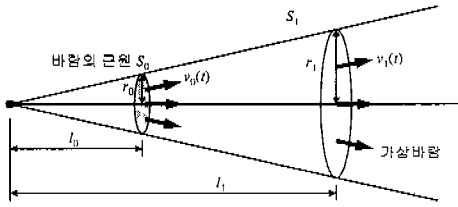


그림 4 가상 바람 모델

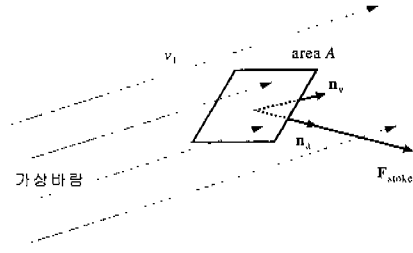


그림 5 바람에 영향 받는 면 추출

터 l_0 거리에 원뿔의 꼭지점을 위치시킨다. 원뿔의 형태는 r_0 와 l_0 의 값을 조정함으로써 변화시킬 수 있다. 생성되는 바람은 S_0 에서 꼭지점의 반대 방향으로 퍼져 나간다. 마이클을 이용한 인터페이스는 S_0 에서의 풍속 $v_0(t)$ 를 조절하는 역할을 한다.

바람을 시뮬레이션하기 위해서는 원뿔의 꼭지점으로부터 거리 $l_1 > l_0$ 에서의 풍속을 계산산하여야 한다. 컴퓨터 그래픽스 분야에서는 공기 역학 방법으로 풍속을 계산하는 방법들[5, 6, 7]이 제안되어 있지만, 가상 모델 시스템에서는 실시간 처리를 행하여야 한다는 조건 때문에 단순화된 형태로 풍속을 계산하였다. 우선 공기는 비점착성의 압축되지 않는 유체이고, 이 유체의 흐름은 주어진 원뿔 내로 폐쇄된다고 가정한다. 이 가정은 정상 풍속의 바람에 대해서는 유효하다[6]. 유체 역학에서의 연속 방정식(continuity equation)은 아래의 식으로 표현할 수 있다.

$$A_0 v_0 = A_1 v_1 \tag{1}$$

이 때, A 와 v 는 각각 해당 단면의 면적과 그 단면에서의 유체의 속도를 나타낸다. 또, 첨자 0와 1은 각각 거리 l_0 와 l_1 에 대응되는 물리량을 의미한다.

원뿔에서의 비례 관계와 식 (1)로부터 다음의 비례식이 구해진다.

$$\frac{v_1}{v_0} = \frac{A_0}{A_1} = \frac{\pi r_0^2}{\pi r_1^2} = \frac{\pi l_0^2}{\pi l_1^2} = \frac{l_0^2}{l_1^2} \tag{2}$$

위와 같이 모델링된 바람이 물체에 부딪혔을 때, 해당 물체에 작용하는 힘은 다음과 같이 스토크 항력 방정식(Stoke drag equation)으로 계산할 수 있다[8,9].

$$\mathbf{F}_{stoke} = \rho A v_1^2 (\mathbf{n}_v \cdot \mathbf{n}_a) \mathbf{n}_a \tag{3}$$

여기서, \mathbf{n}_v 와 \mathbf{n}_a 는 그림 5에서와 같이, 각각 유체의 흐름을 나타내는 단위 벡터와 물체 표면의 단위 법선 벡터이다. ρ 는 유체의 밀도를 나타내는 상수 값이다. 식 (2)와 (3)으로부터 단면적 A 를 가지는 평면에 작용하는 힘을 아래와 같이 계산할 수 있다.

$$\mathbf{F}_{stoke} = \alpha A \frac{v_0^2}{l_1^4} (\mathbf{n}_v \cdot \mathbf{n}_a) \mathbf{n}_a \tag{4}$$

여기서 α 는 상수이다.

실세계에서의 바람은 물체와 부딪치면서 굴절하거나 회절할 수 있다. 또, 바람 자체가 가지는 무질서성을 반영할 필요도 있다. 본 논문에서는 이러한 점들을 고려하기 위해, 식 (4)에서 계산한 힘에 무질서한 항을 더하여 물체에 작용하는 바람에 의한 힘은 최종적으로 다음과 같이 계산한다.

$$\mathbf{F}_{wind} = \mathbf{F}_{stoke} + \mathbf{F}_{rand} \tag{5}$$

무질서하게 더해지는 \mathbf{F}_{rand} 는 대개변수 β 를 이용하여, $|\mathbf{F}_{rand}| < \beta |\mathbf{F}_{stoke}|$ 의 관계를 만족시키도록 설정하고, 그 방향은 무작위로 설정된다.

식 (5)에서 계산한 바람에 의한 힘을 적용시키기 전에 물체의 각 면에 실제로 바람이 작용하는 지를 검사하여야 한다. 가상 모델 시스템에서는 가상 바람의 실시간 처리를 위하여, 바람이 물체에 직접 부딪히는 경우에만 힘이 가해진다고 가정한다. 이 경우에는 물체의 해당 부분이 원뿔의 꼭지점으로부터 직접 가시(directly visible)적이고, 원뿔의 내부에 포함된 경우에만 가상 바람의 영향을 받는다.

즉, 가상 바람의 영향을 받는 면들은 가시면 추출 방법(visible surface detection method)으로 찾을 수 있다. 본 논문에서는 수행 속도의 향상을 위해서 깊이 버퍼(depth-buffer) 방법을 사용하였다. 물체 표면이 일부만 가려진 경우도 처리하기 위해서, 물체의 각 면들은 우선 조그만 영역들로 분할되고, 각 영역에는 대응되는 표본점(sampling point)을 설정한다. 다음에, 원뿔의 꼭지점을 카메라의 위치로 하여 단면 S_0 를 포함하는 평면상으로 투영시킨 영상을 만들어 낸다. 각 표본점에는 일련 번호를 부여하고, 이것이 그래픽스 파이프라인의 알파-버퍼에 따로 저장되도록 한다. 최종적으로 각 표본점이 원뿔의 꼭지점으로부터 직접 가시적인가는 알파-버

$\mathbf{J}\mathbf{W}\mathbf{J}^T$ 행렬을 간단히 $\mathbf{A} = [a_{ij}]$ 라고 하면, 행렬 \mathbf{A} 는 아래와 같은 특성들을 가진다.

- (1) 대각선 상의 원소 a_{ii} 들은 모두 0이 아니다.
- (2) a_{ij} 가 0이 아니면, a_{ji} 도 0이 아니다.
- (3) 하나의 물체가 i 번째, j 번째 제약조건에 영향을 받는 경우에만 a_{ij} 가 0이 아니고, 그 역도 성립한다.

앞에서도 언급했듯이, 몇몇 수치 해석 방법들에서는 식 (6)의 해를 선형 시간에 구할 수도 있지만, 그 처리 과정은 상당히 복잡하다[11, 12]. 반면에, 대부분의 모빌은 트리 형태이기 때문에, 행렬 \mathbf{A} 가 위와 같은 특성들을 가진다. 이 경우에는 아래에 설명한 것과 같이, 가우스 소거법만으로도 간단히 선형 시간에 그 해를 구할 수 있다.

가우스 소거법을 적용시키는 과정에서 대각선상의 원소 a_{ii} 는 0이 아니므로, 항상 a_{ii} 를 피벗 원소(pivot element)로 사용할 수 있고, 또 가상 모빌은 $O(n)$ 개의 제약조건만을 가지므로, 행렬 $\mathbf{A} = \mathbf{J}\mathbf{W}\mathbf{J}^T$ 는 단지 $O(n)$ 개의 0이 아닌(non-zero) 원소만을 가진다. 다음, 가우스 소거를 수행하는 과정에서, $j > i$ 일 때, a_{ij} 가 0이 아니라면 a_{ij} 를 제거해야 한다. 이때 j 번째 행에서 i 번째 행을 뺄 필요가 있고, 그럴 때 행렬 \mathbf{A} 의 특수한 형태 때문에 j 번째 행의 0인 값들이 0이 아닌 값들로 바뀌는 경우가 전혀 없다. 따라서 값이 0인 원소가 0이 아닌 원소로 바뀔 수가 없다. 그리고 제거되는 원소의 개수 또한 $O(n)$ 개이므로, 행렬식의 최종 해를 구하기까지는 총 $O(n)$ 번의 연산만 하면 충분하다. 결과적으로 트리 형태를 가지는 가상 모빌에서는 선형 시간에 행렬식을 풀 수 있다.

6. 충돌 처리

제약조건을 가지는 동역학을 이용하여 모빌의 움직임을 시뮬레이션할 때 물체들간의 충돌이 일어날 수 있다. 일반적인 경우의 충돌 감지(collision detection) 방법들에는 여러 가지가 있지만[14, 15], 가상 모빌 시스템에서는 효율적인 처리를 위하여 가상 모빌의 특징을 이용한다.

우선, 일부 물체들 간에서 전혀 충돌이 일어나지 않는 것들이 있다. 예를 들어 물체 O_3 과 O_5 는 모빌의 구조상 충돌이 불가능하다. 이러한 물체들은 전처리를 통하여 충돌 감지의 대상에서 제외된다. 또, 일부 물체들은 비교적 간단한 외형을 가지므로, 실린더나 구로 근사시킬 수 있다. 이 경우에는 일반적인 다면체들 간의 충돌 감시 방법 대신 실린더와 실린더, 구와 실린더, 구와 구

사이의 충돌 감시 방법을 사용할 수 있으므로 빠른 처리가 가능하다.

모빌의 경우에는 충돌 반응의 처리 시에 조인트에서의 제약조건들도 고려하여야 한다. 즉, 각 조인트에서의 제약조건을 고려하여 충돌 반응을 처리하여야 하고, 충돌 지점에서의 마찰력도 고려하여야 한다. Moore가 제시한 다관절체에 대한 충격 방정식(impulse equation) [17]은 조인트에서의 제약조건들도 적용 가능한 특성이 있으므로, 가상 모빌 시스템에서는 이 방법을 사용한다. 물체 O_i 의 질량과 관성 모멘트를 각각 m_i 와 \mathbf{I}_i 라 하고, 충돌 이전에 O_i 는 선속도 \mathbf{v}_i 와 각속도 $\boldsymbol{\omega}_i$ 를 가진다고 하자. 충돌에 의하여 O_i 는 새로운 선속도 $\bar{\mathbf{v}}_i$ 와 새로운 각속도 $\bar{\boldsymbol{\omega}}_i$ 를 가지게 된다. 충격력(impulse)을 이용한 충돌 반응 방법은 운동량의 보존에 기초하여 $\bar{\mathbf{v}}_i$ 와 $\bar{\boldsymbol{\omega}}_i$ 를 계산한다. 즉, 충돌 전후의 운동량 변화는 충돌에 의한 충격력의 총합과 같으므로, 다음의 충격 방정식이 만족되어야 한다[17].

$$m_i(\bar{\mathbf{v}}_i - \mathbf{v}_i) = \mathbf{P} + \sum_{j=1}^n \mathbf{P}_{ij} \quad (7)$$

식 (7)에서 \mathbf{P} 는 O_i 에 가해진 충격력이고, \mathbf{P}_{ij} 는 조인트 제약조건에 의하여 O_i 에 연결된 O_j 를 통하여 O_i 에 가해지는 부가 충격력(attachment impulse)이다.

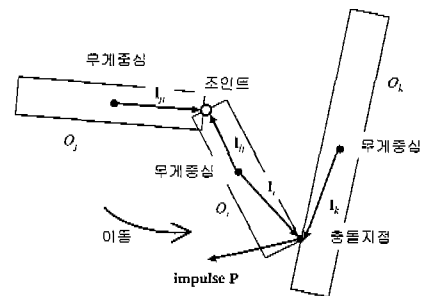


그림 6 충격력기반 충돌반응

조인트에서의 제약조건을 이용하여 추가적인 관계식을 세울 수 있다. 우선, 스피리컬 제약조건(spherical constraint)로 연결된 O_i 와 O_j 에서는 연결 지점에서의 상대 속도가 일치하여야 한다. 따라서, 그림 6에서와 같이, O_i , O_j 의 무게 중심으로부터 스피리컬 조인트(spherical joint)까지의 거리 벡터를 각각 \mathbf{l}_i , \mathbf{l}_j 라 하면,

$$\bar{\mathbf{v}}_i + \bar{\boldsymbol{\omega}}_i \times \mathbf{l}_i = \bar{\mathbf{v}}_j + \bar{\boldsymbol{\omega}}_j \times \mathbf{l}_j \quad (8)$$

의 관계를 만족하여야 한다.

물체 O_k 가 한 점에 고정된 경우는 고정점 제약조건에 해당하는데, 고정점에서의 선속도가 0이 되어야 한다. 따라서, $\mathbf{1}_{kk}$ 를 O_k 의 무게 중심으로부터 고정점까지의 거리라 하면,

$$\bar{\mathbf{v}}_k + \bar{\boldsymbol{\omega}}_k \times \mathbf{1}_{kk} = \mathbf{0} \quad (9)$$

라는 관계가 성립한다[21].

Moore는 위의 관계식들을 마찰력이 있는 경우로 확장하였으나, Moore의 방법에서는 연립 방정식을 한꺼번에 계산해야만 각 충돌 지점에서의 마찰력을 알아낼 수 있는 단점이 있다.

가상 모빌 시스템에서는 Moore의 관계식들에 Mirtich의 편별식[15]을 결합시켜서 계산 속도를 향상시켰다.

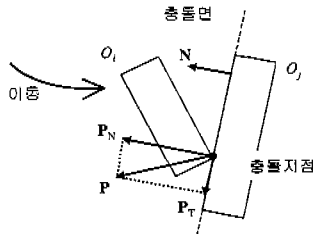


그림 7 충돌 반응

그림 7에서와 같이, O_i 의 표면상의 한 지점에서 O_j 의 한 점이 충돌한다고 하자. O_i 에 가해지는 충격력 \mathbf{P} 는 O_i 의 표면의 법선 벡터 \mathbf{N} 을 기준으로, 수직 성분 \mathbf{P}_N 과 수평 성분 \mathbf{P}_T 로 분리할 수 있다. 이 때의 마찰 상태는 정지 마찰력이 작용하는 경우와 운동 마찰력이 작용하는 경우로 분류할 수 있다.

충격력의 관점에서는 정지 마찰력이 작용하는 경우는 충돌 지점에서의 미끄러짐이 없음을 의미하고, 마찰 계수 μ 에 대하여 $|\mathbf{P}_T| \leq \mu |\mathbf{P}_N|$ 의 관계를 만족한다. 역으로, $|\mathbf{P}_T| > \mu |\mathbf{P}_N|$ 인 경우에 충돌 지점은 O_i 의 표면을 따라 미끄러지고, 이 때는 운동 마찰력이 작용한다.

운동 마찰력과 정지 마찰력의 계산은 각기 다른 방법으로 이루어져야 하기 때문에, 충돌시에 어떤 상황이 벌어졌는지를 검사하는 것이 필수적이다. Mirtich는 운동 마찰력이 작용하는 지를 예측할 수 있는 관계식을 다음과 같이 도입하였다.

$$\left(\frac{1}{K_{13}}\right)^2 + \left(\frac{1}{K_{23}}\right)^2 \geq \mu^2 \left(\frac{1}{K_{33}}\right)^2 \quad (10)$$

식 (10)에서 3×3 정방행렬 $\mathbf{K} = [\mathbf{k}_{ij}]$ 는 3×3 단위행렬 \mathbf{E} 를 이용하여 $\left[\left(\frac{1}{m_i} + \frac{1}{m_j}\right)\mathbf{E} - (\mathbf{1}_i \times \mathbf{I}_i^{-1} \times \mathbf{1}_i + \mathbf{1}_j \times \mathbf{I}_j^{-1} \times \mathbf{1}_j)\right]$ 로 계산된다. 충돌지점에서 위의 관계식이 성립하는 경우에

는 운동마찰력이 작용한다는 것이 알려져 있으므로, 가상 모빌 시스템에서는 이 관계식을 이용하여 충돌 반응을 좀 더 빠르게 처리한다.

7. 구현 예제

본 논문에서 예제로 구현한 Steel Fish는 8개의 조인트로 연결된 9개의 물체로 구성된다. 각 조인트들은 모두 자유도 3을 가지므로, 제약조건을 가지는 동역학 계산기는 총 24개의 제약조건을 만족시켜야 한다. 따라서, 제약조건힘들은 구하기 위해서는 24×24 행렬로 구성된 선형방정식을 계산하여야 한다.

실제 움직임을 시뮬레이션하기 위해서는 고정되어 있는 물체 O_0 를 제외한 8개 물체들의 선속도와 각속도를 각각 구하여야 한다. 또 충격력을 이용한 충돌 처리를 위해서는 추가로 충격력 \mathbf{P} 와 이에 의한 최대 8개의 부가 충격력 \mathbf{P}_{ij} 를 계산하여야 한다. 따라서 구해야 하는 미지수의 총합은 75개이고, 충격역학 계산기는 최대 75×75 행렬로 구성된 선형 방정식의 해를 구한다.

그림 8은 가상 모빌시스템이 생성해 낸, 가상 모빌의 움직임을 보여주고 있다. 이 예제의 생성에 사용된 컴퓨터는 128MByte의 RAM을 가진 Pentium II 400MHz 급 PC이고, 소프트웨어로 구현된 OpenGL 라이브러리를 사용하여 렌더링하였다. 가상 모빌은 총 1788개의 다각형으로 구성되어 있다. 모빌에 움직임이 일어나도록 하기 위해, 사용자가 화면의 정면에서 모빌을 향해 바람을 생성한 경우이고, 가상 모빌 시스템은 평균 초당 9 프레임의 속도로 모빌의 움직임을 시뮬레이션하였다.

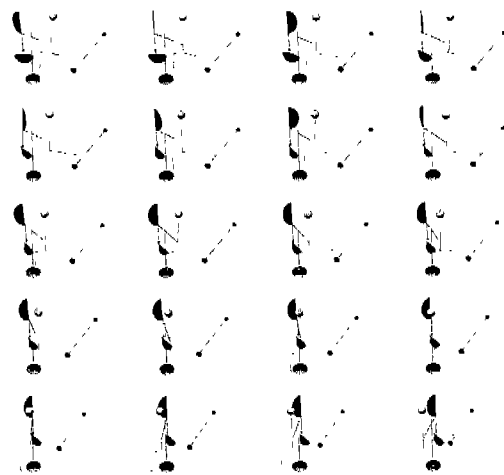


그림 8 가상 모빌에서의 적용

8. 결론

본 논문의 목적은 컴퓨터 시스템에서 실세계의 모빌을 재현하는데 있고, 이를 위해 물리법칙에 따라 움직이는 가상 모빌 시스템을 개발하였다. 대화형으로 움직이는 가상 모빌을 보여주기 위해서, 본 논문은 가상 바람 모델의 도입, 제약조건을 가지는 동역학의 빠른 해결, 충격역학의 빠른 적용 방법 등의 3가지 요소에 중점을 주었다.

본 논문에서 사용한 바람 모델은 사용자가 설정한 방향으로 인위적인 바람을 효과적으로 생성할 수 있다. 이 모델은 인위적으로 생성되는 방향성을 가지는 바람의 생성에 유용하다. 또, 바람의 생성과 제어 시에 마이크를 입력 도구로 사용할 수 있음을 보였다. 특히 이러한 마이크 인터페이스는 시간의 흐름에 따라 풍속을 제어하기에 효과적이다.

가상 모빌 시스템은 물리 기반 모델링에 기초하고 있기 때문에, 모빌의 움직임은 제약조건을 가지는 동역학 계산기와 충격역학 계산기의 두 단계를 거쳐 생성된다. 대화형으로 모빌을 움직여 볼 수 있도록 하기 위해서, 이들은 각각 모빌의 특성을 최대한 이용하여 효과적으로 해를 구하도록 개선되었다. 개선된 방법들은 모빌 이외에도 트리 형태를 가지는 다관절체들에 적용 가능하다.

본 논문에서 구현한 가상 모빌 시스템은 모빌을 컴퓨터 상에서 재현함으로써, 사용자가 실세계의 모빌을 가상 환경에서 감상할 수 있는 방법을 제시하였다. 이 방법은 이제까지 주로 회화나 조각을 대상으로 했던 예술 작품들의 디지털 형태 재생을 모빌이라는 동적인 분야로까지 확대시켰다는 의미를 가진다. 또한 이 과정에서 개발된 개선 방안들은 물리 기반 모델링의 계산 속도를 향상시키는 데에 적용 가능하다. 향후에는 다양한 형태의 모빌들을 모델링하여 하나의 시스템에서 여러 종류의 모빌들을 감상할 수 있도록 개선할 필요가 있다. 모빌의 충돌에 따른 음향 효과 역시 앞으로 다루어야 할 과제이다.

참고 문헌

- [1] H. McWhinnie, The Electronic Museum, *Computers and Graphics*, 12(2):269, 1988.
- [2] F. McGuire, The origins of sculpture: Evolutionary 3D design, *IEEE CG&A*, 13(1):9-11, 1993.
- [3] P. Rademacher and G. Bishop, Multiple-Center-of-Projection Images, *SIGGRAPH '98*, pp.199-206, 1998.
- [4] B. Mirtich, V-Clip: Fast and Robust Polyhedral Collision Detection, *ACM Trans. on Graphics*, 17(3):177-208, 1998.
- [5] W. Reeves, Particle Systems - A Technique for Modeling a Class of Fuzzy Objects, *ACM Trans. on Graphics*, 2(2):91-108, 1983.
- [6] J. Wejchert and D. Haumann, Animation Aerodynamics, *SIGGRAPH '91*, pp.19-22, 1991.
- [7] M. Shinya and A. Fournier, Stochastic Motion - Motion Under the Influence of Wind, *EUROGRAPHICS '92*, 11(3):119-128, 1992.
- [8] A. Patterson, *A First Course in Fluid Dynamics*, Cambridge University Press, 1989.
- [9] R. Feynmann, R. Leighton and M. Sands, *The Feynman Lectures on Physics*, Addison-Wesley, 1965.
- [10] P. Schröder and D. Zeltzer, The Virtual Erector Set: Dynamic Simulation with Linear Recursive Constraint Propagation, *Computer Graphics (1990 Symposium on Interactive 3D Graphics)*, 24(2): 23-31, 1990.
- [11] M. Surles, An Algorithm with Linear Complexity for Interactive, Physically-based Modeling of Large Proteins, *SIGGRAPH '92*, pp.221-230, 1992.
- [12] D. Baraff, Linear-Time Dynamics using Lagrange Multipliers, *SIGGRAPH '96*, pp.137-146, 1996.
- [13] M. Gleicher and A. Witkin, Through-the-lens Camera Control, *SIGGRAPH '92*, pp.331-340, 1992.
- [14] P. Hubbard, Approximating Polyhedra with Spheres for Time-Critical Collision Detection, *ACM Trans. on Graphics*, 15(3):179-210, 1996.
- [15] B. Mirtich, Impulse-based Dynamic Simulations of Rigid Body Systems, *Ph.D. thesis*, University of California, Berkeley, 1996.
- [16] J. Hahn, Realistic Animation of Rigid Bodies, *SIGGRAPH '88*, pp.299-308, 1988.
- [17] M. Moore and J. Wilhelms, Collision Detection and Response for Computer Animation, *SIGGRAPH '88*, pp.289-298, 1988.
- [18] J. Neider, T. Davis and M. Woo, *OpenGL Programming Guide*, Addison-Wesley Publishing Company, 1993.
- [19] 이남경, 실시간 재생을 위한 인위적인 바람 모델, 경북대학교 석사학위논문, 1999.
- [20] 이남경, 백낙훈, 이종원, 류관우 "실시간 응용을 위한 인위적인 바람의 생성", 한국정보과학회 논문지, Vol. 27, No. 8, pp.701-709, 2000.
- [21] 정대현, 물리기반 모델링에서 선형시간 충돌반응, 경북대학교 석사학위논문, 1999.



이 동 준

1999년 경북대학교 컴퓨터공학과 졸.
2001년 경북대학교 컴퓨터공학과 석.
2001년 ~ 현재 한국전자통신연구원 증강현실연구팀. 관심분야는 컴퓨터 그래픽스, 증강현실



배 회 정

1999년 경북대학교 컴퓨터공학과 박사수료. 1997년 ~ 현재 경북대학교 컴퓨터공학과 디지털 미디어연구실 연구원. 2001년 ~ 현재 전자통신연구소 VR센터 위촉연구원. 관심분야는 Real-Time animation, Physics based Modelling, 계산기하학



이 남 경

1998년 경북대학교 공과대학 컴퓨터공학과 졸업. 2000년 경북대학교 컴퓨터공학과 공학석사. 2000년 ~ 현재 경북대학교 컴퓨터공학과 박사과정. 관심분야는 컴퓨터 그래픽스, 알고리즘



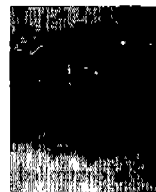
백 낙 훈

1990년 한국과학기술원 과학기술대학 전산학과 졸업. 1992년 한국과학기술원 전산학과 공학석사. 1997년 한국과학기술원 전산학과 공학박사. 1997년 ~ 1998년 미국 조오지워싱턴 대학 초청 연구원. 1998년 ~ 2001년 경북대학교 전자전기공학부 초빙교수. 2001년 ~ 현재 동국대학교 컴퓨터공학과. 관심분야는 컴퓨터 그래픽스, 계산기하학



정 대 현

1998년 경북대학교 공과대학 컴퓨터공학과 졸업. 2000년 경북대학교 컴퓨터공학과 공학석사. 2000년 ~ 현재 경북대학교 컴퓨터공학과 박사과정. 관심분야는 컴퓨터 그래픽스, 알고리즘



이 중 원

1982년 ~ 1886년 경북대학교 수학과 학사졸업. 1989년 ~ 1992년 미국 일리노이 공과대학 Computer Science 석사 졸업. 1992년 ~ 현재 미국조지워싱턴대학교 Computer Science, Computer Graphics 전공, 박사과정. 관심분야는 컴퓨터 그래픽스, 계산기하학



김 창 태

2000년 ~ 현재 (주)KOG 기술연구소 재직. 관심분야는 컴퓨터 그래픽스



유 관 우

1980년 경북대학교 전자공학과 졸업(공학사). 1982년 한국과학기술원 전산학과 졸업(이학석사). 1990년 미국 메릴랜드대학(University of Maryland) 졸업. 1982년 ~ 현재 경북대학교 컴퓨터공학과 부교수.



이 동 규

1996년 경북대학교 컴퓨터공학과 졸업. 1998년 경북대학교 컴퓨터공학과 석사. 2000년 경북대학교 컴퓨터공학과 박사 수료. 2001년 8월 ~ 현재 (주)케이오지. 관심분야는 컴퓨터 그래픽스