

# 비정형 물체의 실시간 애니메이션을 위한 안정적 질량-스프링 모델 Stable Mass-Spring Model for Real-time Animation of Flexible Objects

강영민, 조환규  
부산대학교 전자계산학과  
그래픽스 응용 연구실  
{ymkang,hgcho}@pearl.cs.pusan.ac.kr

박찬종  
한국전자통신연구원  
가상현실 연구센터  
cjpark@etri.re.kr

## 초록

본 논문은 유연한 비정형 물체의 애니메이션을 위한 효율적인 기법을 제안한다. 비정형 물체를 표현하기 위해 질량-스프링 모델이 사용되었다. 지금까지 많은 기법들이 부드러운 객체의 사실적인 애니메이션을 생성하기 위해 질량-스프링 모델을 사용하였다. 질량-스프링 모델의 애니메이션을 수행하기 위한 가장 손쉬운 접근법은 명시적 오일러 방법(explicit Euler method)인데, 이 방법은 '불안정성 문제'라는 잘 알려진 문제가 발생한다는 단점을 가지고 있다. 이 불안정성 문제를 해결하기 위한 해법으로 암시적 적분법이 사용될 수 있다. 그러나, 이 암시적 방법의 가장 결정적인 약점은 대규모의 선형 시스템을 풀어야 한다는 것이다. 본 논문은 암시적 방법의 근사(approximation)를 이용하여 질량-스프링 모델을 빠른 시간에 애니메이션 할 수 있는 기법을 제시한다. 제안된 기법은  $n$  개의 질점이  $O(n)$  개의 스프링으로 연결되어 있을 때, 각 질점의 상태를  $O(n)$  시간에 안정적으로 갱신할 수 있다. 본 논문은 사실적인 결과를 위해 비정형 물체와 공기와의 상호 작용도 고려하였다.

## Abstract

In this paper, we propose an efficient technique for the animation of flexible thin objects. Mass-spring model was employed to represent the flexible objects. Till now, many techniques have used the mass-spring model to generate plausible animation of soft objects. A straight-forward approach to the animation with mass-spring model is explicit Euler method, but the explicit Euler method has serious disadvantage that it suffers from 'instability problem'. The implicit integration method is a possible solution to overcome the instability problem. However, the most critical flaw of the implicit method is that it involves a large linear system. This paper presents a fast animation technique for mass-spring model with approximated implicit method. The proposed technique stably updates the state of  $n$  mass-points in  $O(n)$  time when the number of total springs are  $O(n)$ . We also consider the interaction of the flexible object and air in order to generate plausible results.

## 1 서론

Terzopolous가 비정형 물체의 움직임을 변형 가능한 곡면 문제로 형식화한 이후[12], 비정형 물체를 표현하는 많은 기법들이 제안되었다 [2, 3, 7, 10, 13]. 이들 기법들은 유한 요소법, 입자 시스템, 변형 가능 표면 모델과 같은 다양한 모델을 사용하였는데[4, 15], 이들 모델 가운데 질량-스프링 모델은 가장 쉬운 면에서도 직관적인 모델이다.

질량-스프링 모델을 이용하여 그림 1과 같은 비정형 물체 동작을 생성하기 위한 많은 기법들이 제안되었다 [5, 6, 10]. 질량-스프링 모델에 기반한 애니메이션 기법들은 간단한 미분 방정식으로 표현될 수 있다. 각 질점에 가해지는 힘은 스프링의 힘으로 간단히 구할 수가 있으며 이 힘을 수치적으로 적분함으로써 질량-스프링 모델의 움직임을 생성할 수 있다.

명시적(explicit) 적분 방법(explicit integration method)은 이러한 문제를 해결하는 가장 간단한 접근 방법이다. 이 방법은 현재 상태의 값들을 간단한 오일러 방법과 같은 식에 적용하여 바로 다음 상태의 값들을 계산하는 것이다. 이러한 접근법은 불안정성이라는 잘 알려진 문제를 가지고 있다 [8, 9, 16]. 따라서 이러한 접근법이 실제로 적용되기 위해서는 아주 짧은 시간 간격을 시뮬레이션에 적용해야만 한다.

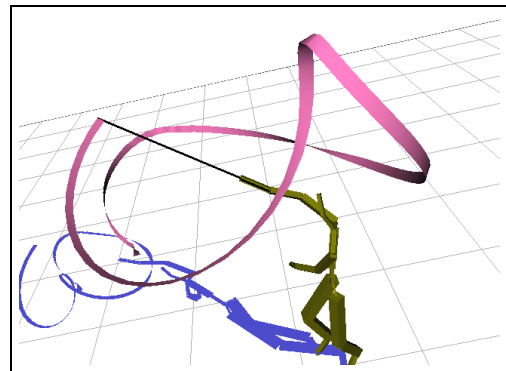


Figure 1. 비정형 물체 움직임의 예

수치 해석 분야에서는 이 불안정성 문제를 해결하기 위해 암시적 적분이 잘 알려져 있는데 [8, 9], 최근에는 Baraff가 암시적(implicit) 방법을 사용하여 시뮬레이션에서 큰 시간 간격을 사용할 수 있는 기법을 제안하였다 [1]. 이 방법은 큰 시간 간격을

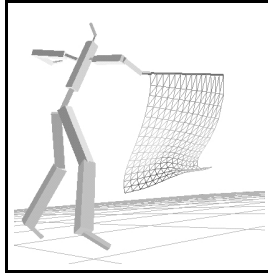
사용하여 시물레이션이나 애니메이션 생성 시간을 크게 줄일 수 있기 때문에, 질량-스프링 모델의 애니메이션에 가장 적당한 방법이라 할 수 있다. 그러나 이 암시적 방법은 대규모의 선형 시스템을 풀어야만 한다는 단점을 가지고 있다. 암시적 방법이 애니메이션 생성 과정을 안정적인 수치 적분으로 만들기는 하지만, 이 대규모 선형 시스템이 실시간 애니메이션의 큰 장애가 된다.

Desbrun은 선형 시스템을 미리 계산된 행렬(미리 계산된 필터)을 이용하여 푸는 효율적인 방법을 이용하여 옷감과 같은 비정형 물체를 빠른 시간에 애니메이션 할 수 있는 기법을 제안하였다. 그러나, 이 방법 역시 계산된 행렬이 희소 행렬(sparse matrix)이 아닐 수도 있기 때문에 많은 계산량을 요구하며, 미리 계산된 필터가 질량이나, 시간 간격, 그리고 물체의 경직도에 좌우되기 때문에 애니메이션 도중에 이들 파라미터를 수정하는 것이 불가능하다는 단점을 가지고 있다.

본 논문은  $n$  개의 질점 상태를  $O(n)$  시간에 안정적으로 갱신하는 빠르고 안정적인 애니메이션 기법을 제안한다. 이 방법은 Desbrun이 제안한 미리 계산된 필터를 사용하는 방법에서는 사용할 수 없었던, 적응적 시간 간격(adaptive time step) 방법이나, 동적 파라미터 변경을 제한 없이 사용할 수도 있다.

## 2 질량-스프링 모델과 안정적 적분

질량-스프링 모델은 비정형 물체를 표현하는 간단한 기법이다. 이 모델은 물체를 질량을 가진 질점과 이들을 연결하는 스프링으로 표현한다. 이 스프링들에 의해 생성된 힘이 질점들을 움직이게 하는 것이다. 이때 스프링은 임의적으로 구성될 수 있다. 그림 2는 이 질량-스프링 모델을 이용하여 비정형 물체를 생성한 예를 보이고 있다.



**Figure 2.** 질량-스프링 모델을 이용하여 유연한 물체를 생성한 예

명시적 오일러 적분법을 이용하여 이 질량-스프링 모델을 쉽게 애니메이션 할 수 있다. 그러나, 불안정성 문제 때문에 단순한 오일러 적분 방법은 아주 짧은 시간 간격을 요구한다. 즉, 각 질점의 상태를 매우 짧은 시간 간격마다 생성해야 하므로, 전체 애니메이션을 생성하는데 걸리는 시간이 지나치게 많이 걸린다는 단점을 가지고 있다. 따라서, 이 명시적 방법은 상호작용적(interactive)인 응용 시스템에는 적용이 불가능하다.

### 2.1 안정적 애니메이션을 위한 암시적 방법

암시적 방법은 불안정성 문제를 해결하는 해법이다. 암시적 방법을 이용함으로써 우리는 각 질점의 상태를 안정적으로 갱

신할 수 있는데, 암시적 오일러 적분 방법은 다음과 같이 질점의 상태를 갱신한다:

$$\begin{aligned} \mathbf{v}_i^{t+h} &= \mathbf{v}_i^t + \mathbf{F}_i^{t+h} \frac{h}{m_i} \\ \mathbf{x}_i^{t+h} &= \mathbf{x}_i^t + \mathbf{v}_i^{t+h} h \end{aligned} \quad (1)$$

이때,  $\mathbf{v}_i^{t+h}$ 는  $i$  번째 질점의 시간  $t$ 에서의 속도를 나타내며,  $\mathbf{F}_i^{t+h}$ 는 이 질점에 시간  $t$ 일 때 작용하는 힘을 의미한다. 또한  $\mathbf{x}_i^{t+h}$ 는 이  $i$  번째 질점의 시간  $t$ 에서의 위치를 의미한다.  $m_i$ 는 이 질점의 질량이며,  $h$ 는 애니메이션 프레임 사이의 시간 간격이다.

이 암시적 적분 방법은 질량-스프링 모델의 상태를 안정적으로 갱신하기 때문에, 시간 간격  $h$ 를 크게 잡을 수 있다. 따라서, 빠른 애니메이션 생성이 필요한 상호작용적 시스템에서는 명시적 방법이 아니라 이 암시적 적분 방법을 사용해야 한다. 그러나, 암시적 오일러 방법은 현재 상태에서 알 수 없는  $\mathbf{F}_i^{t+h}$  값을 포함하고 있다. 이 값은 1차 도함수를 이용하여 다음과 같이 근사할 수 있다:

$$\mathbf{F}^{t+h} = \mathbf{F}^t + \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \Delta \mathbf{x}^{t+h}$$

이때,  $\mathbf{F}^{t+h}$ 는 시간  $t$ 에 모든 질점에 작용하는 스프링 힘들로 이루어진 벡터를 의미한다. 즉,  $\mathbf{F}^t$ 는  $[\mathbf{F}_1^t, \mathbf{F}_2^t, \dots, \mathbf{F}_n^t]^T$ 이다. 그리고,  $\partial \mathbf{x}$  역시  $[\Delta \mathbf{x}_1^t, \Delta \mathbf{x}_2^t, \dots, \Delta \mathbf{x}_n^t]^T$ 를 의미한다. 이때,  $\partial \mathbf{F} / \partial \mathbf{x}$ 는 시스템의 헤시안 행렬이므로[6], 앞으로  $\mathbf{H}$ 로 나타내기로 한다. 그런데,  $\Delta \mathbf{x}^{t+h} = \mathbf{x}^{t+h} - \mathbf{x}^t = (\mathbf{v}^t + \Delta \mathbf{v}^{t+h})h$ 이므로, 식 1에 표현된 암시적 갱신식의 첫 번째 수식은 다음과 같이 다시 쓸 수 있다:

$$(\mathbf{I} - \frac{h^2}{m} \mathbf{H}) \Delta \mathbf{v}^{t+h} = (\mathbf{F}^t + h \mathbf{H} \mathbf{v}^t) \frac{h}{m} \quad (2)$$

이때,  $\Delta \mathbf{v}^{t+h}$ 는  $\mathbf{v}^{t+h} - \mathbf{v}^t$ 로 각 질점의 속도 변화로 구성된 벡터를 의미한다. 만약 이 속도 변화 벡터  $\Delta \mathbf{v}^{t+h}$ 를 계산할 수 있다면, 우리는 식 1을 이용하여 각 질점의 속도를 계산할 수 있으므로 쉽게 다음 상태의 위치를 계산할 수 있다. 따라서 비정형 물체의 애니메이션은 바로 이 속도 변화  $\Delta \mathbf{v}^{t+h}$ 를 계산하는 문제라고 할 수 있다. 식 2에 보이는  $h \mathbf{H} \mathbf{v}^t$ 는 Desbrun이 언급한 것처럼 힘에 더해지는 추가적인 점성력으로 다음과 같은 식으로 쉽게 계산할 수 있다 [6].

$$(h \mathbf{H} \mathbf{v}^t)_i = h \sum_{(i,j) \in E} k_{ij} (\mathbf{v}_j - \mathbf{v}_i)$$

이때  $E$ 는 질점들 사이의 스프링의 집합이며,  $k_{ij}$ 는  $i$  질점과  $j$  질점을 연결하는 스프링의 스프링 상수이다.

암시적 방법의 문제점은 식 2가  $O(n^2)$  크기의 행렬인  $\mathbf{I} - (h^2/m) \mathbf{H}$ 를 포함하고 있다는 것이다. 이 행렬 때문에, 암시적 방법은 질량-스프링 모델의 상태를 갱신하기 위해 대규모 선형 시스템을 풀어야 한다. 이런 계산상의 부담을 줄이기 위해 Desbrun은 스프링 힘의 선형 요소만을 고려한 헤시안 행렬  $\mathbf{H}$ 를 근사를 이용하여 효율적이며 안정적으로 각 질점의 상태를 갱신하는 방법을 제안하였다. Desbrun이 제안한 방법은 헤시안 행렬의  $i$  행  $j$  열 요소  $\mathbf{H}_{ij}$ 를  $k_{ij}$ 로 근사하고,  $\mathbf{H}_{ii}$ 를  $-\sum_{j \neq i} k_{ij}$ 로 근사하는 것이다. 그러면,  $(\mathbf{I} - (h^2/m) \mathbf{H})$  행렬은 애니메이션 도중에 변하지 않는 상수 행렬이 된다. 따라서 이 행렬의 역함수  $(\mathbf{I} - (h^2/m) \mathbf{H})^{-1}$  역시 상수 행렬이다. Desbrun은 이  $(\mathbf{I} - (h^2/m) \mathbf{H})^{-1}$ 를 미리 계산해 두었다가 각 애니메이션 단

계마다 힘을 필터링하는데 사용한다. 즉, 이 방법은 다음과 같은 식으로 나타낼 수 있다.

$$\Delta \mathbf{v}^{t+h} = (\mathbf{I} - \frac{h^2}{m} \mathbf{H})^{-1} \frac{\tilde{\mathbf{F}}^t h}{m}$$

이때  $\tilde{\mathbf{F}}$ 는 스프링 힘과 점성력을 합한 힘이다 (즉,  $\tilde{\mathbf{F}}^t = \mathbf{F}^t + h\mathbf{H}\mathbf{v}^t$ ).

Desbrun이 제안한 방법은 암시적 방법을 사용하는 다른 일반적 기법들에 비해 훨씬 빠르다. 그러나,  $\mathbf{I} - (h^2/m)\mathbf{H}$ 이 희소 행렬 (sparse matrix)라 할지라도, 그 역행렬  $\mathbf{I} - (h^2/m)\mathbf{H}$ 은 희소 행렬이 아닐 수도 있으며, 이 경우  $O(n^2)$ 의 수행 시간을 보인다. 또한,  $\mathbf{I} - (h^2/m)\mathbf{H}$  행렬은 시간 간격, 질량, 그리고 스프링의 경직도(stiffness)에 의존적인 값이므로, 이 방법은 애니메이션 수행 도중에 이런 파라미터들을 조정하려고 할 때 새로 그 역행렬을 계산해야 한다는 단점이 있다. 따라서 이 방법은 적응적 시간 간격(adaptive time step) 기법을 사용할 수가 없으며, 쉽게 질량, 경직도 등을 조정하면서 그 동작을 확인해 보는 것도 힘들다. 이러한 이유로, 우리는 이 미리 계산된 필터를 이용한 기법을 사용하지 않았다.

## 2.2 상호작용적 애니메이션을 위한 근사 방법

헤시안 행렬의  $\mathbf{H}_{ij}$ 는  $i$  번째 질점과  $j$  번째 질점이 서로 연결되어 있지 않을 때 0의 값을 가진다. 따라서 우리는 각 질점의 속도 변화를 계산하는 식 2를 연결된 질점만을 고려하는 다음과 같은 식으로 표현할 수 있다:

$$(1 - \frac{h^2 \mathbf{H}_{ii}}{m_i}) \Delta \mathbf{v}_i - \frac{h^2}{m_i} \sum_{(i,j) \in E} (\mathbf{H}_{ij} \Delta \mathbf{v}_j) = \frac{\tilde{\mathbf{F}}_i^t h}{m_i}$$

우리는 계산의 단순화를 위해 Desbrun이 사용한 헤시안 행렬 근사를 받아 들였다. 모든 스프링에 대해 동일한 스프링 상수  $k$ 를 가정하고,  $n_i$ 가  $i$  번째 질점에 스프링으로 연결된 모든 질점의 수라고 할 때, 헤시안 행렬의 각 원소는  $\mathbf{H}_{ij} = k$  ( $i$ 와  $j$ 는 스프링으로 연결된 질점들)와  $\mathbf{H}_{ii} = -kn_i$ 로 나타낼 수 있다. 따라서 속도 변화를 계산하는 식은 다음과 같이 나타낼 수 있다.

$$\frac{m_i + h^2 kn_i}{m_i} \Delta \mathbf{v}_i^{t+h} = \frac{\tilde{\mathbf{F}}_i^t h}{m_i} + \frac{h^2 k \sum_{(i,j) \in E} \Delta \mathbf{v}_j^{t+h}}{m_i}$$

따라서,  $\Delta \mathbf{v}_i^{t+h}$ 는 다음과 같이 나타낼 수 있다:

$$\Delta \mathbf{v}_i^{t+h} = \frac{\tilde{\mathbf{F}}_i^t h + kh^2 \sum_{(i,j) \in E} \Delta \mathbf{v}_j^{t+h}}{m_i + kh^2 n_i} \quad (3)$$

그러나, 우리는 이  $\Delta \mathbf{v}_i^{t+h}$ 를 바로 계산할 수가 없는데, 그 이유는 이 식 3이 아직 알 수 없는 값인  $\Delta \mathbf{v}_j^{t+h}$ 를 포함하고 있기 때문이다. 이 값은  $i$  번째 질점에 스프링으로 연결되어 있는 질점  $j$ 의 시간  $t+h$ 에서의 속도 변화이다.

우리는  $\Delta \mathbf{v}_i^{t+h}$ 를 계산하기 위해 이  $\Delta \mathbf{v}_j^{t+h}$ 의 값을 근사하는 방법을 사용하였다.  $\Delta \mathbf{v}_j^{t+h}$ 는  $\Delta \mathbf{v}_i^{t+h}$ 와 마찬가지로 다음과 같이 표현될 수 있다:

$$\Delta \mathbf{v}_j^{t+h} = \frac{\tilde{\mathbf{F}}_j^t h + h^2 \sum_{(j,l) \in E} k_{jl} \Delta \mathbf{v}_l^{t+h}}{m_j + h^2 \sum_{(j,l) \in E} k_{jl}}$$

여기서  $h^2 \sum_{(j,l) \in E} k_{jl} \Delta \mathbf{v}_l^{t+h}$ 를 생략함으로써 우리는 다음과 같은 근사값을 얻을 수 있다:

$$\Delta \mathbf{v}_j^{t+h} \simeq \frac{\tilde{\mathbf{F}}_j^t h}{m_j + h^2 \sum_{(j,l) \in E} k_{jl}}$$

이 근사값을 이용하여 우리는  $\Delta \mathbf{v}_i^{t+h}$ 를 계산하는 갱신식을 다음과 같이 얻을 수 있다:

$$\Delta \mathbf{v}_i^{t+h} = \frac{\tilde{\mathbf{F}}_i^t h + h^2 k \sum_{(i,j) \in E} \tilde{\mathbf{F}}_j^t h / (m_j + h^2 k n_j)}{m_i + h^2 k n_i} \quad (4)$$

$\tilde{\mathbf{F}}_i^t$ 와  $\tilde{\mathbf{F}}_j^t$ 는 이미 아는 값이므로, 우리는 식 4를 이용하여 쉽게 각 질점의 속도 변화를 계산할 수 있다. 이것은 상호작용적 애니메이션의 가장 큰 장애물이었던 대규모 선형 시스템의 풀이 없이 비정형 물체의 동작을 안정적으로 생성할 수 있다는 것을 의미한다.

이 식을 이용하면, 어떤 질점의 속도 변화를 그 질점에 연결된 질점들만 고려해서 계산할 수 있으므로 모든 질점의 다음 상태를 계산하는데  $O(n)$ 의 시간만 필요하다는 것은 명백하며, 이는 미리 계산된 역행렬(미리 계산된 필터)을 이용하거나 일반적인 암시적 방법을 사용하는 기법들보다 훨씬 효율적으로 애니메이션을 생성할 수 있다는 것을 의미한다. 더구나 이 방법은 미리 계산된 역행렬 방법과 달리, 질량, 스프링 상수, 시간 간격 등의 파라미터가 변경되었을 경우에 어떤 추가적 계산도 필요 없다는 장점을 가지고 있다. 우리는 이 식을 이용하여 비정형 물체의 동작을 생성해 보았으며, 매우 안정적인 결과를 얻을 수 있었다.

## 2.3 제안된 기법의 안정성

본 논문이 제안한 기법의 안정성을 알아보기 위해, 간단한 예를 살펴보자. 단 두 개의 질점  $i$ 와  $j$ 가 존재하고, 이 두 질점이 휴지 길이가 0인 스프링으로 연결되어 있다고 하자. 이 스프링의 스프링 상수는  $k$ 이다.

두 질점의 현재 속도가 모두  $[0,0,0]^T$ 라고 하고, 점성력을 고려하지 않을 경우 두 질점에 작용하는 스프링 힘은 다음과 같다.

$$\begin{aligned} \tilde{\mathbf{F}}_i^t &= -k(\mathbf{x}_i^t - \mathbf{x}_j^t)^t \\ \tilde{\mathbf{F}}_j^t &= -k(\mathbf{x}_j^t - \mathbf{x}_i^t)^t = -\tilde{\mathbf{F}}_i^t \end{aligned}$$

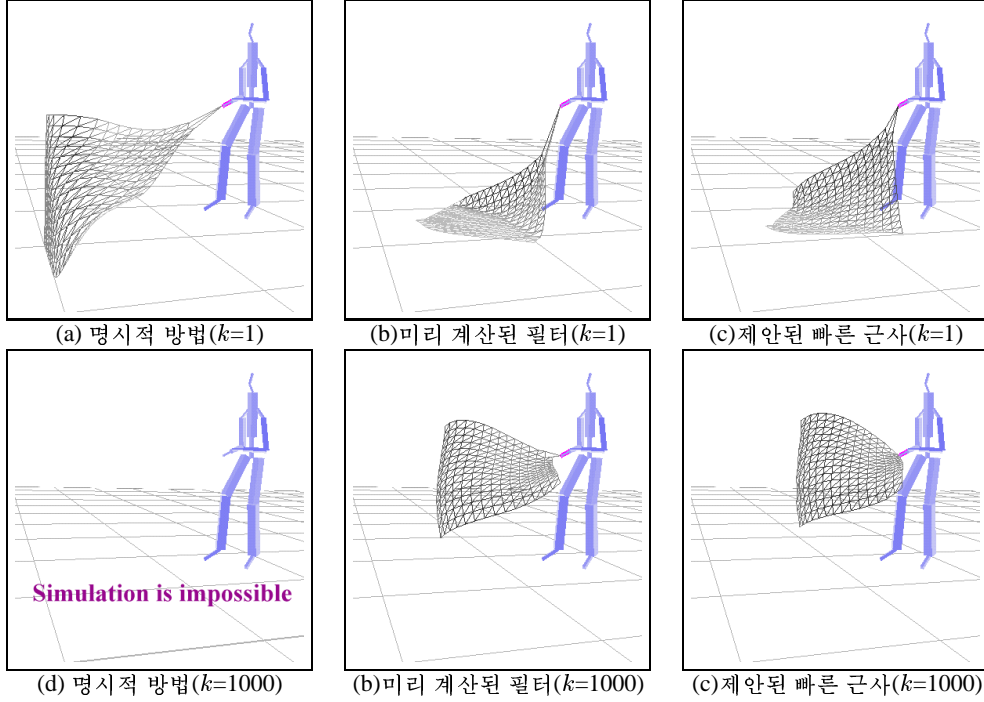
만일  $|\mathbf{x}_i^{t+h} - \mathbf{x}_j^{t+h}| \leq |\mathbf{x}_i^t - \mathbf{x}_j^t|$  라면, 두 질점의 위치가 발산하지 않는다는 것은 명백하다.  $n_i = n_j = 1$ ,  $\tilde{\mathbf{F}}_j^t = -\tilde{\mathbf{F}}_i^t$ , 그리고  $m_i = m_j = m$ 이므로, 식 4를 이용하여 각 질점의 다음 위치를 계산하면 다음과 같다.

$$\begin{aligned} \mathbf{x}_i^{t+h} &= \mathbf{x}_i^t + \frac{\tilde{\mathbf{F}}_i^t h - kh^2 \tilde{\mathbf{F}}_i^t h / (m + kh^2)}{m + kh^2} h \\ \mathbf{x}_j^{t+h} &= \mathbf{x}_j^t - \frac{\tilde{\mathbf{F}}_i^t h - kh^2 \tilde{\mathbf{F}}_i^t h / (m + kh^2)}{m + kh^2} h \end{aligned}$$

이제  $\mathbf{u}$ 가  $\mathbf{x}_i^t - \mathbf{x}_j^t$ 를 나타낸다고 할 때,  $\tilde{\mathbf{F}}_i^t$ 는  $-ku$ 로 표현할 수 있으므로, 두 질점의 위치는 다음과 같이 표현할 수 있다.

$$\mathbf{x}_i^{t+h} - \mathbf{x}_j^{t+h} = \mathbf{u}(1 - 2 \cdot \frac{m kh^2}{(m + kh^2)^2})$$

$0 \leq m kh^2 / (m + kh^2)^2 \leq 1$ 인 경우에  $|\mathbf{x}_i^{t+h} - \mathbf{x}_j^{t+h}|$ 가  $|\mathbf{x}_i^t - \mathbf{x}_j^t|$ 보다 작거나 같다는 것은 명백하며,  $m, k, h$  모두 0보다 큰 값을 가지므로  $m + kh^2 / (m + kh^2)^2$ 이 0보다 크다는 것도 명백하다. 따라서 이제  $m kh^2 / (m + kh^2)^2$ 가 1보다 크다는 것, 즉,  $m kh^2$ 이  $(m + kh^2)^2$ 보다 크다는 것만 보이면 된



**Figure 3.** 명시적 방법과, 미리 계산된 필터를 이용하는 방법, 그리고 제안된 기법의 비교: (a), (d) - 명시적 방법. (b), (e) - Desbrun이 제안한 미리 계산된 필터 이용. (c), (f) - 논문에서 제안한 기법 이용. 경직도가 1000일 때, 명시적 방법은 애니메이션 생성에 실패. 미리 계산된 필터를 이용하는 방법과 본 논문에서 제안하는 방법은 안정적으로 애니메이션을 생성. 시간 간격은 1/30초, 각 질점의 질량은 0.1Kg, 사용된 질점의 수는 225 개. (a),(b),(c)는 스프링 상수를 1로 하였고, (d),(e),(f)는 경직도를 높여 스프링 상수를 1000으로 하였다.

다.  $(m + kh^2)^2$ 는  $m^2 + k^2h^4 + 2mkh^2$ 이므로, 우리는 쉽게  $(m + kh^2)^2 - mkh^2$ 이  $m^2 + k^2h^4 + mkh^2$ 임을 알 수 있고, 이 값은 0보다 크다. 따라서  $mkh^2/(m + kh^2)^2$ 가 1보다 작다는 것을 알 수 있다.

그림 3은 명시적 방법과, 미리 계산된 필터를 이용하는 방법, 그리고 제안된 기법을 비교한 것이다. 그림 3의 (a)와 (d)는 명시적 방법을 사용했을 때의 결과이다. (b)와 (e)는 Desbrun이 제안한 미리 계산된 필터를 이용해 생성한 결과이다. 그리고, (c)와 (f)는 본 논문에서 제안한 기법을 이용하여 생성한 결과이다. 경직도가 1000일 때 명시적 방법은 애니메이션 생성에 실패하였다. 반면, 미리 계산된 필터를 이용하는 방법과 본 논문에서 제안하는 방법은 안정적으로 애니메이션을 생성할 수 있었다. 사용된 파라미터로, 시간 간격은 1/30초, 각 질점의 질량은 0.1kg, 그리고 사용된 질점의 수는 225 개씩이었다. (a),(b),(c)는 스프링 상수를 1로 하였고, (d),(e),(f)는 경직도를 높여 스프링 상수를 1000으로 하였다.

### 3 공기와의 상호작용

사실적인 비정형 물체 애니메이션을 생성하기 위해서는 공기와 비정형 물체와의 상호작용이 고려되어야 한다. 일반적으로 어떤 물체가 유체 속에서 운동할 때, 두 가지 종류의 힘을 받게 되는데, 그 중 하나가 항력이고, 다른 하나가 양력이다. 항력

의 크기는 다음과 같은 식으로 알려져 있다.

$$|F_D| = \frac{1}{2} C_D \rho |V|^2 S \sin \theta$$

이때,  $|F_D|$ 는 항력의 크기를 나타내며,  $C_D$ 는 항력 계수, 그리고  $\rho$ 는 유체의 밀도,  $V$ 는 유체 속에서 운동하는 객체의 유체와의 상대 속도를 나타낸다. 그리고,  $S$ 는 객체의 면적이며,  $\theta$ 는 면과 속도가 이루는 방향이다. 항력의 방향은 속도의 반대 방향이 된다.

객체에 작용하는 양력도 항력과 비슷하게 표현된다:

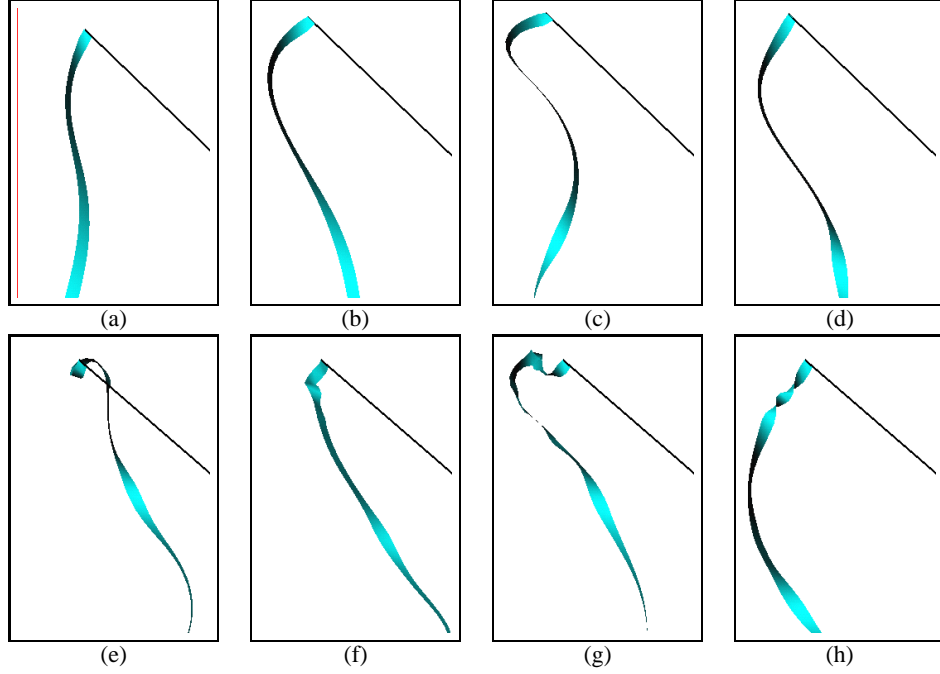
$$|F_L| = \frac{1}{2} C_L \rho |V|^2 S \cos \theta$$

이때,  $|F_L|$ 는 양력의 크기를 나타내며,  $C_L$ 은 양력 계수를 나타낸다. 양력의 방향은 속도 방향에 수직이다.

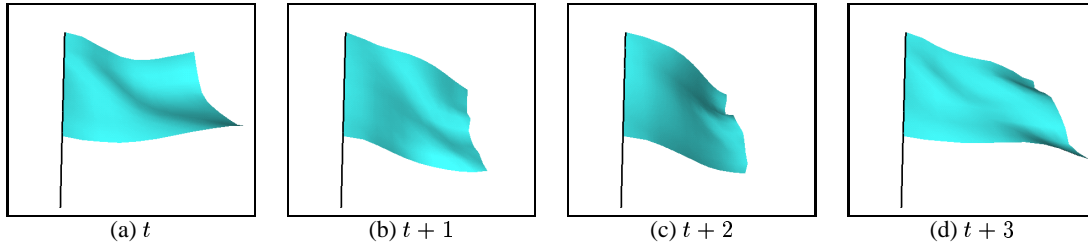
우리는 비정형 물체의 더욱 사실적인 동작을 생성하기 위해 이 두 가지 힘을 고려하였다. 항력은 다음의 식 5와 같이 구현하였다.

$$\mathbf{F}_{D_i} = -K_D |\hat{\mathbf{N}}_i \cdot \hat{\mathbf{v}}_i| |\mathbf{v}_i|^2 \hat{\mathbf{v}}_i \quad (5)$$

이때,  $\mathbf{F}_{D_i}$ 는  $i$  번째 질점에 작용하는 항력,  $K_D$ 는 항력 제어 파라미터,  $\hat{\mathbf{N}}_i$ 는  $i$  번째 질점에서의 단위 법선 벡터,  $\hat{\mathbf{v}}_i$ 는  $i$  번째 질점의 유체에 대한 상대속도를 크기로 나눈 것이다. 법선벡터와 속도가 이루는 각은  $(\pi/2 - \theta)$ 이므로,  $\hat{\mathbf{N}}_i \cdot \hat{\mathbf{v}}_i$ 는  $\sin \theta$ 가 된다.



**Figure 4.** 공기 중의 리본 움직임 생성 : 그림 (a)에서 (d)까지는 공기와의 상호작용을 고려하지 않고 애니메이션을 생성한 결과이며, (e)에서 (h)까지는 항력과 양력을 고려한 결과이다



**Figure 5.** 바람 속의 깃발 : 막대에 매달린 깃발에 왼쪽에서 오른쪽으로 바람이 부는 모습

양력을 구현하기 위해서는 양력이 어느 방향으로 작용할 것 인지를 결정해야 한다. 우리는 다음과 같은 방법으로 양력의 방향  $U_i$ 를 결정하였다.

$$\begin{aligned}\tilde{\mathbf{N}}_i &= \hat{\mathbf{N}}_i, \text{ if } \hat{\mathbf{N}}_i \cdot \hat{\mathbf{v}}_i > 0 \\ &-\hat{\mathbf{N}}_i, \text{ otherwise} \\ \mathbf{U}_i &= (\tilde{\mathbf{N}}_i \times \hat{\mathbf{v}}_i) \times \hat{\mathbf{v}}_i\end{aligned}$$

이 양력 방향  $U_i$ 를 이용하여 질점  $i$ 에 작용하는 양력을 다음의 식 6과 같이 구현하였다.

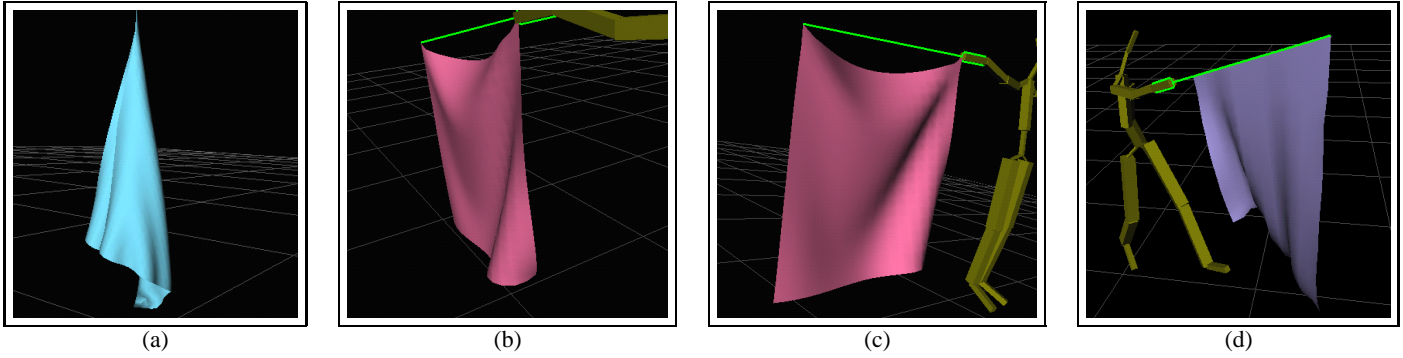
$$\mathbf{F}_{Li} = (K_L \cos \theta |\mathbf{v}_i|^2) \mathbf{U}_i \quad (6)$$

그림 4와 5는 이 양력과 항력을 이용하여 생성한 애니메이션의 결과이다. 그림 4의 (a), (b), (c), (d)는 공기와의 상호작용을 고려하지 않았을 때의 결과이며, 그림 4의 (e), (f), (g), (h)는 공기와의 상호작용을 고려했을 때의 그림이다. 그림에서 볼 수 있듯이 공기와의 상호작용을 고려했을 때 더욱 사실적인 움직임을 얻을 수 있었다. 그림 5는 공기와의 상호작용을 고려하여, 바람에 날리는 깃발의 동작을 생성한 결과이다.

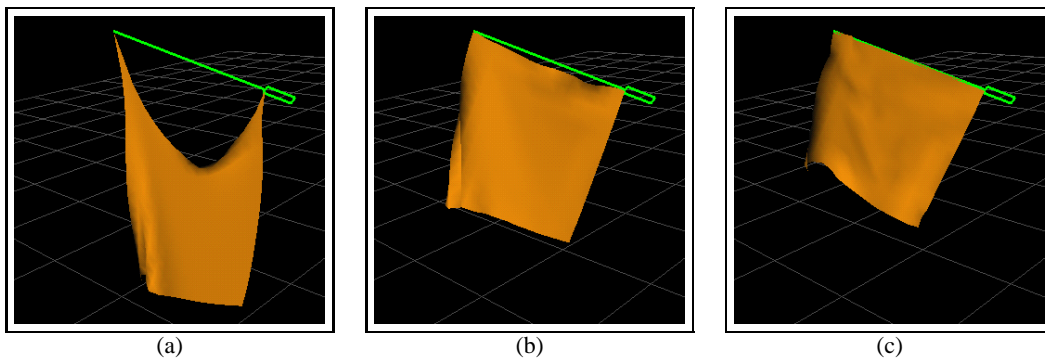
## 4 실험결과

우리는 본 논문에서 제안한 기법을 바탕으로 SGI Indigo<sup>2</sup> R10000 워크스테이션에서 C++, *OpenInventor* 라이브러리를 사용하여 수 백 개의 질점을 가진 천을 초당 60 프레임의 시간 간격으로 실시간 생성할 수 있는 비정형 물체 생성 시스템을 구현했다. 그러나, 질량-스프링 모델은 옷감과 같은 비정형 물체를 정확히 표현하는데 한계를 가지고 있다. 질량-스프링 모델은 실제 옷감과 달리 지나치게 많이 늘어난다. 이를 해결하기 위해서는 과도하게 늘어난 스프링을 줄여주는 인버스 다이내믹스 (inverse dynamics) 과정이 필요한데, 우리는 Provot이 [10]에서 제안한 방법을 사용하였다.

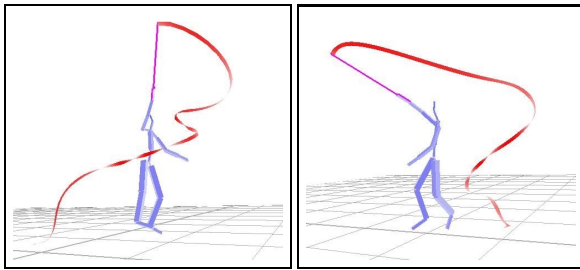
또한 비정형 물체는 자체 충돌이 일어난다는 특성을 가지는데, 자체 충돌이란 물체 자체의 일부분이 다른 부분에 부딪히는 것을 말한다. 자체 충돌을 감지하는 방법으로는 Volino와 Provot의 방법이 일반적이다 [11, 14]. 그러나, 우리는 더욱 빠른 충돌 해결을 위해 공간 분할과 해싱을 이용한 실시간 충돌 회피 기법을 구현하여 실험하였다. 아직은 빠른 속도로 움직일 때의



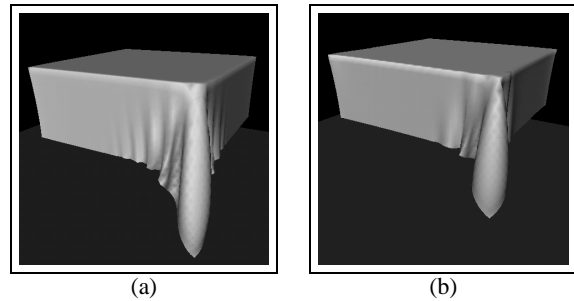
**Figure 6.** 제안한 기법을 이용한 비정형 물체 동작 생성 예: (a) 한 질점의 움직임이 제한되었을 때, (b),(c) 두 질점의 움직임이 입력에 의해 제한되었을 때 (d) 다수의 질점이 제한되었을 때



**Figure 7.** 서로 다른 경직도를 주었을 때, 비정형 물체의 동작 생성 (a)  $k=1$  (b)  $k=65$  (c)  $k=500$



**Figure 8.** 리본 체조 동작의 생성 예



**Figure 9.** 테이블 위에 놓여진 천의 모습 (a) 경직도 100 (b) 경직도 225

충돌을 완벽히 해결하지 못해 보완 중이다.

그림 6은 본 논문에서 제안한 기법을 구현한 시스템을 이용하여 생성한 결과들이다. 그림 6의 (a)는 하나의 질점이 고정되었을 때의 모습이고, 그림 6의 (b)와 (c)는 두 개의 질점이 입력에 의해 움직일 때, 생성된 결과이다.

그림 7은 서로 다른 스프링 상수를 적용하여 비정형 물체의 경직도를 조절하였을 때의 결과이다. 그림 7의 (a), (b), (c) 모두 경직도를 제외하고는 동일한 파라미터를 사용하였다. 그림 7의 (a)는 경직도  $k = 1$ , (b)는  $k = 65$ , 그리고 (c)는  $k = 500$ 이 사용되었다. 경직도 값을 올려도 제안된 기법은 안정적으로 동작을 생성할 수 있었다. 그림 8은 이 제안된 기법을 이용하여 체조 동작에 따라 리본의 움직임을 생성한 것이다. 그림에 보이는 캐

릭터는 모션 캡처 시스템을 통해 얻은 것이며, 이 동작에 가상의 리본을 부착해 그림 8과 같은 결과를 얻었다.

제안된 기법이 얼마나 사실적인 결과를 생성하는지 확인하기 위해 그림 9과 같이 테이블 위에 놓여진 천의 모습을 생성하여 보았다. 그림 9의 (a)와 (b)는 각각 100과 225의 경직도로 생성된 결과이다. 그림에서 확인할 수 있듯이 제안된 기법은 빠른 계산을 통해서 매우 사실적인 결과를 생성할 수 있었다.

본 논문에서 제안된 기법을 이용하여 생성한 결과 이미지와 동영상은 <http://pearl.cs.pusan.ac.kr/cloth>에서 확인할 수 있다.

## 5 결론

우리는 질량-스프링 모델의 빠르고 안정적인 애니메이션을 위해 암시적 수치 적분을 근사한 기법을 제안하였으며, 제안된 기법을 이용하여 구현된 애니메이션 시스템을 통해 생성한 결과들을 보였다. 본 논문의 기법은 물리적 정확성보다는 비정형 물체의 그럴듯한 동작을 빠르게 생성하는 것이 1차적 목적이다. 실험 결과는 제안한 기법이 큰 시간 간격을 사용한 실험에서도 매우 그럴듯 하면서 안정적인 움직임을 생성하고 있음을 보여 주고 있으며, 상호작용적 응용시스템이나 실시간 시스템에 적용될 수 있음을 보이고 있다.

제안한 기법은 암시적 방법의 필터링 특성을 가지고 있기 때문에 매우 안정적이다. 또한, 암시적 방법을 사용할 때 계산상의 중요한 병목이 되는 대규모 선형 시스템의 풀이 과정이 필요하지 않기 때문에, 명시적 방법만큼이나 빠른 속도를 보인다. 제안한 시스템의 또 다른 장점은 적응적 시간 간격 기법이나, 동적인 파라미터 변경에도 추가적 계산 없이 안정적으로 동작한다는 것이다.

본 논문의 애니메이션 기법은 비정형 물체의 움직임을 빠른 시간 안에 생성해야 하는 다양한 응용 시스템에 적용이 가능하다.

## 감사의 글

본 연구는 한국 전자 통신 연구원과의 위탁과제 연구 계약을 통한 지원으로 이루어졌습니다.

## References

- [1] D. Baraff and A. Witkin. Large steps in cloth simulation. *Computer Graphics (Proc. of SIGGRAPH '98)*, pages 43–52, 1998.
- [2] D. Breen, D. House, and M. Wozny. Predicting the drape of woven cloth using interacting particles. *Computer Graphics (Proc. of SIGGRAPH '94)*, pages 365–372, 1994.
- [3] M. Carignan, Y. Yang, N. Thalmann, and D. Thalmann. Dressing animated synthetic actors with complex deformable clothes. *Computer Graphics (Proc. of SIGGRAPH '92)*, pages 99–104, 1992.
- [4] G. Celniker and D. Gossard. Deformable curve and surface finite-elements for free-form shape design. *Computer Graphics (Proc. of SIGGRAPH '91)*, pages 257–266, 1991.
- [5] Y. Chen, Q. Zhu, and A. Kaufman. Physically-based animation of volumetric objects. *Proc. of Computer Animation '98*, pages 154–160, 1998.
- [6] M. Desbrun, P. Schröder, and A. Barr. Interactive animation of structured deformable objects. *Proc. of Graphics Interface '99*, 1999.
- [7] B. Eberhardt, A. Weber, and W. Strasser. A fast, flexible, particle-system model for cloth draping. *IEEE Computer Graphics and Applications*, 16:52–59, 1996.
- [8] M. Kass. An introduction to continuum dynamics for computer graphics. In *SIGGRAPH Course Note*. ACM SIGGRAPH, 1994.
- [9] S. Nakamura. Initial value problems of ordinary differential equations. In *Applied Numerical Methods with Software*, pages 289–350. Prentice-Hall, 1991.
- [10] X. Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. *Proc. of Graphics Interface '95*, pages 147–154, 1995.
- [11] X. Provot. Collision and self-collision handling in cloth model dedicated to design garments. *Proc. of Graphics Interface '97*, pages 1177–1189, 1997.
- [12] D. Terzopoulos, J. Platt, and A. Barr. Elastically deformable models. *Computer Graphics (Proc. of SIGGRAPH '87)*, pages 205–214, 1987.
- [13] P. Volino, M. Courchesne, and N. Magnenat-Thalmann. Versatile and efficient techniques for simulating cloth and other deformable objects. *Computer Graphics (Proc. of SIGGRAPH '95)*, pages 137–144, 1995.
- [14] P. Volino and N. Magnenat-Thalmann. Efficient self-collision detection on smoothly discretized surface animations using geometrical shape regularity. *Proc. of Euro-Graphics '94*, pages 155–166, 1994.
- [15] B. Wang, Z. Wu, Q. Sun, and M. Yuen. A deformable model of thin flexible surfaces. *Proc. of WSCG'98*, pages 440–446, 1998.
- [16] A. Witkin and D. Baraff. Differential equation basics. In *SIGGRAPH Course Note*. ACM SIGGRAPH, 1995.