DOI: 10.3745/KIPSTB.2008.15-B.2.103

사실적인 컴퓨터 애니메이션 구현을 위한 증분형 영상 기반 운동 렌더링 기법

한 영 모†

요 약

사실적인 컴퓨터 애니메이션 제작 시 종종 모션 캡쳐 기술이 사용된다. 모션 캡쳐 기술은 대상체의 운동을 측정해서 모델링한 운동 렌더링 결과를 그래픽스로 표현한다. 본 논문에서는 카메라를 사용해서 얻어진 2차원 영상 정보로부터 대상체의 3차원 운동을 측정하여 모델링하는 영상 기반 운동 렌더링 문제를 다룬다. 기존의 영상 기반 운동 렌더링 알고리즘은 계산량이 너무 많거나 정확도가 떨어지는 등의 단점이 있었다. 첫 번 째 단점은 장편 애니메이션 제작시 제작 시간이 너무 길어서 문제가 되고, 두 번 째 단점은 사실적인 애니메이션 구현시 사실감이 저하되는 문제를 야기 시킨다. 이와 같은 기존 방식의 단점을 보완하기 위하여 본 논문에서는 계산량이 적고 정확도가 높은 영상 기반 운동 렌더링 알고리즘을 제안한다. 본 논문에서 제안한 방식에서는 계산량이 적은 증분형 운동렌더링 알고리즘을 최적제어 이론의 시각에서 분석하여 정확도를 향상시키도록 개조한다. 본 방식을 광학식 모션 캡쳐 기술에 적용할 경우 표시자(marker)의 부착 없이도 모션 캡쳐가 가능하다는 부가적인 장점 또한 얻을 수 있다.

키워드 : 영상 기반 운동 렌더링, 모션 캡쳐, 사실적인 컴퓨터애니메이션, 최적제어

Incremental Image-Based Motion Rendering Technique for Implementation of Realistic Computer Animation

Youngmo Han†

ABSTRACT

Image-based motion capture technology is often used in making realistic computer animation. In this paper we try to implement image-based motion rendering by fixing a camera to a PC. Existing image-based rendering algorithms have disadvantages of high computational burden or low accuracy. The former disadvantage causes too long making-time of an animation. The latter disadvantage degrades reality in making realistic animation. To compensate for those disadvantages of the existing approaches, this paper presents an image-based motion rendering algorithm with low computational load and high estimation accuracy. In the proposed approach, an incremental motion rendering algorithm with low computational load is analyzed in the respect of optimal control theory and revised so that its estimation accuracy is enhanced. If we apply this proposed approach to optic motion capture systems, we can obtain additional advantages that motion capture can be performed without any markers, and with low cost in the respect of equipments and spaces.

Key Words : Image-Based Motion Rendering, Motion Capture, Realistic Computer Animation, Optimal Control Theory

1. 서 론

컴퓨터 애니메이션을 제작할 때, 제작 효율을 높이기 위해서 또는 정확한 모델링의 어려움 때문에, 일반적으로 대상체의 운동을 단순화하여 모델링하는 경우가 많다. 예를 들면, 동역학(dynamics)을 생략하고 기구학(kinematics)만으로 운동을 묘사하는 경우가 많다. 하지만 이러한 운동 모델의 단순화 때문에, 애니메이션의 운동이 자연스럽지 않게 된다. 이를 보완하여 좀더 사실적인 컴퓨터 애니메이션을 제작하기 위해서, 운동 모델링에 기구학 뿐 아니라 동역학까지 고려하려는 연구가 진행되었다 [4][5]. 일반적으로 3차원 운동의 동역학이 매우 복잡한 관계로, 이러한 접근법은 많은 계산량을 필요로하게 된다. 기구학 또는 동역학을 고려한 애니메이션 제작시, 또 다른 어려움은 애니메이터가 그 운동을 일일이 기술해야 한다는 것인데 그 작업량이 일반적으로 상당히 크다. 이것은 애니메이션 제작자에게 큰 부담을 주어서, 제작 시간과 제작비를 상승시키는 요인이 된다.

사실적인 컴퓨터 애니메이션 제작시 작업량을 줄이기 위

※ 이 논문은 2007년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음(KRF-2006-003-I01610).
† 정 회 원 : 한양사이버대학교 컴퓨터공학과 조교수
논문접수 : 2007년 6월 29일, 심사완료 : 2007년 9월 20일

해서, 최근들어 '영상기반 운동 렌더링(image-based motion rendering) 기법'이 연구되고 있다. 이는 애니메이션으로 제작하고 싶은 대상체를 카메라로 모니터링 하거나 녹화한 뒤, 그 영상으로부터 대상체의 3차원 운동 정보를 복원하고 모델링 하는 기법이다. 이렇게 얻어진 대상체의 3차원 운동 정보를 배경 영상과 적절히 혼합하고 컴퓨터 그래픽스 도구를 사용하여 표현하면 원하는 애니메이션을 얻게 된다. '영상기반 운동 렌더링(image-based motion rendering) 기법'의 핵심은 영상 데이터로부터 3차원 정보를 복원하는 '3차원 정보 복원(reconstruction of 3D-information from images) 기법'이라 할 수 있다.

영상 데이터로부터 3차원 정보를 복원하는 문제의 시초는 Longuet-Higgins의 연구[3]와 Ullman의 연구[6]라 할 수 있는데, 이 후 본 주제에 대해 많은 연구가 수행되었다(예를 들어, 참고문헌 [7], [8], [9] 참조). 본 주제의 대표적인 접근법으로는 운동 파라미터와 거리(depth) 파라미터를 분리하여 복원하는 알고리즘 (이후 '분리형 알고리즘' 으로 부르기로 한다) 과 이 두 종류의 파라미터를 결합하여 복원하는 알고리즘 (이후 '결합형 알고리즘'으로 부르기로 한다) 으로 나누어 진다.

분리형 알고리즘은 대부분 Epipolar 제약조건에 기반하여 거리(depth) 파라미터와 운동(motion) 파라미터 를 분리한다. 이러한 접근법의 가장 큰 장점은 3차원 정보 복원 문제가 선형 형태로 표현되므로 적은 계산량으로도 3차원 운동 정보를 복원할 수 있다는 점이다. 하지만, 분리형 알고리즘은 전이 운동(translation motion)의 정보를 복원하는데 바이어스(bias) 현상이 생긴다거나 [10] [11], 잡음과 양자화(quantization)에 대해 상당히 민감하다는 단점이 있다[9]. 뿐만 아니라 Epipolar 제약조건에 기반한 접근법은 전이 속도(translation velocity)가 0인 경우 3차원 복원에 실패하게 된다. 그 밖의 거리 파라미터와 운동 파라미터의 분리 과정에 의한 성능저하 문제가 참고문헌 [1]에 잘 연구되어 있다.

결합형 알고리즘의 대표적인 접근법으로는 Szeliski 와 Kang에 의해 연구[2]된 특징점의 좌표(feature coordinates)의 이론값과 측정값 사이의 차이를 최소화하는 기법이 있다. 이 연구에서는 Levenberg-Marquardt 알고리즘을 사용해서 특징점 좌표의 이론값과 측정값 사이의 차이를 가중치를 곱해서 최소화 하였다. 이러한 접근법은 위에서 열거한 epipolar 제약조건의 접근법에서 발생하는 문제점들에 대해 심각하게 영향 받지 않는다는 장점이 있다. 하지만 본 접근법의 가장 큰 어려움은 아주 큰 차원의 파라미터 탐색 공간(특징점을 n 개 사용했을 경우 파라미터 탐색 공간의 차원은 n+6)이다. 아주 큰 차원의 파라미터 탐색 공간은 계산량과 계산시간을 급증시킴으로써 실시간 연산을 어렵게 한다. 이 문제를 완화시키기 위해서 McLauchlan [1]은 근사화 방법을 사용하는 순차적 알고리즘(sequential algorithm)을 개발하여 계산 효율을 증대시켰다. 하지만 이 방법은 순차 연산 중에 오차가 누적되는 단점이 있다. 이러한 오차 누적은 추정치의 정확도를 떨어뜨리고, 나아가 오차 누적 량이 증가함에 따라 결국 3차원 정보 복원에 실패하게 된다. 따라서 이 접근법은 큰 정확도가 필요하지는 않고 실시간으로 3차원 정보를 복원하고자 할 때 적합하다.

위에서 살펴본 바와 같이 3차원 정보 복원 문제(reconstruction of 3D-information)는 많은 학자들에 의해 광범위하게 연구되어 오고 있다. 그러나 현재 알려진 접근법들은 각각 장단점을 갖고 있는데, 그 중 특히 계산량과 결과의 신뢰도(결과의 정확성에 비례, 오차 누적 현상에 반비례) 사이의 트레이드오프(trade-off) 문제가 심각하다. 본 논문에서는 컴퓨터 애니메이션 제작에 사용될 영상 기반 운동 렌더링 알고리즘을 제안한다. 본 논문에서는 주어진 시간 동안 영상 정보를 사용해서 지속적으로 3차원 정보 복원을 수행하는 것을 영상 기반 운동 렌더링으로 정의한다. 특히 실용성을 위해서 계산량이 적고 신뢰도가 높은 알고리즘을 제안하는데 주안점을 둔다. 주어진 시간 동안 3차원 정보복원을 지속적으로 수행하는 가장 직접적인 방법은 Szeliski와 Kang 의 비선형 알고리즘[2]을 연속적으로 수행하는 것이다. 하지만 이 경우 과도한 계산량이 필요하게 된다. 그래서 본 연구에서는 계산량을 줄이기 위한 방법으로 McLauchlan의 순차적 알고리즘[1]을 발전시킨 증분형 알고리즘(incremental algorithm)를 도입하고자 한다.

증분형 알고리즘(Incremental algorithm)이란 영상 기반 운동 렌더링의 초기에만 비선형 알고리즘을 사용한 정확한 3차원 정보 복원을 계산하고, 그 다음부터는 그 증분(increment) 만을 계산하여 보충해 주는 방식이다. 그러나 증분형 알고리즘의 단점은 증분을 보충해 주는 과정에서 오차가 발생했을 경우, 정정이 되지 않고 그 오차 값이 시간이 지남에 따라 점점 더 누적될 수 있다는 것이다. 이를 오차 누적 현상이라 한다. 이렇게 축적된 오차 때문에 결과값의 신뢰도가 떨어지고 나아가 영상 기반 운동 렌더링 자체가 실패하게 된다.

본 논문에서는 증분형 알고리즘의 오차 누적 현상을 완화하고, 결과값의 신뢰도를 높이는 문제에 집중한다. 그 기본 아이디어는 다음과 같다. 증분형 알고리즘은 기본적으로 시간적 진화(time-evolution)[1] 문제를 포함하고 있다. 그럼에도 불구하고 기존의 증분형 알고리즘에서는 time-evolution을 배제하고, 각 시간에서 파라미터 만을 최적화하였다. 하지만 본 연구에서는 time-evolution 과정에서의 오차 전파를 방지하고 더욱 정확한 결과값을 얻기 위해서, time-evolution을 포함하여 주어진 시간 전체에 대해서 파라미터를 최적화할 것을 제안한다.

2. 모션 캡쳐를 사용한 제안하는 사실적인 컴퓨터 애니메이션 제작 패러다임

모션 캡쳐 기술은 주로 인간의 캐릭터 애니메이션을 제작

1) 상태 방정식에 의해 시간 경과에 따라 상태 변수가 갱신(update)되는 것을 지칭한다. 본 논문에서는 시간에 따른 상태 변수의 변화가 상태 방정식에 의해 제약되고 있음을 강조하여 단순한 상태 변수의 시간적 변화라는 표현 대신 상태 변화의 시간적 진화란 표현을 사용하였다.

하는 데 많이 사용되어 왔다. 본 연구에서는 이에 한 발 더 나아가 모션 캡쳐 기술이 복잡한 역학 계산이 필요한 사실적인 컴퓨터 애니메이션 제작에도 유용하게 사용될 수 있음을 보이고자 한다. 겉보기에는 단순해 보여도 정확하게 운동을 묘사하려면 매우 복잡한 역학 계산을 해야 하는 경우가 종종 있다. 예를 들면, 자동차가 출발하거나 정차할 때는 기구학 뿐 아니라 자동차 차체의 동역학 및 관성력, 타이어와 지면 사이의 마찰력, 타이어와 기어 스프링의 탄성력 등이 복잡하게 얽히게 된다. 따라서 단순 기구학 계산만으로는 사실적인 운동 묘사가 어렵고, 그렇다고 이 많은 역학 관계를 고려하여 계산하기는 너무 어렵다. 또 한 가지 간단한 예를 들면, 공이나 풍선을 던지거나 낙하하는 경우를 생각할 수 있다. 이 단순해 보이는 운동에도 기구학 뿐 아니라, 중력에 의한 등가속도 운동, 공기에 의한 마찰력, 지면에 부딪칠 때의 탄성력 및 지면에 대한 마찰력 등의 복잡한 역학 관계가 얽혀 있다. 이와 같은 복잡한 역학 관계가 얽혀 있는 경우의 사실적인 컴퓨터 애니메이션 제작시 종종 물리 기반 컴퓨터 그래픽스 (physics-based computer graphics) 기술이 사용된다.

이는 역학 관계를 최대한 반영하여 운동을 계산하여 에니메이션을 제작하는 것인데, 모델링이 너무 어렵고 계산량이 지나치게 많은 단점이 있다. 이런 경우 보다 편리한 방법은 모션 캡쳐 기술을 사용하여 운동을 모델링 함으로써, 컴퓨터 애니메이션을 제작할 수 있다.

이러한 취지에서 본 연구는 모션 캡쳐 기술을 복잡한 역학 관계가 얽혀있는 사실적인 컴퓨터 애니메이션 구현에 적용하는 경우를 집중적으로 다뤄 보고자 한다.

사용의 편리성을 강조하여 본 논문에서는 광학식 모션 캡쳐 패러다임을 사용하는 경우를 다루고자 한다. 물론 광학식 모션 캡쳐 패러다임에도 단점이 있다. 앞에서 살펴 본 바와 같이 표시자(marker) 부착에 따른 제약, 많은 수의 장비 및 전용 스튜디오의 필요 등이 그것이다. 표시자(marker)의 부착을 위해서는 대상체 뿐 아니라 배경의 색상까지 통제해야 하는 불편이 뒤따른다. 이와 같은 표시자(marker) 부착의 불편은 결국 광학식 모션 캡쳐가 실내에서 제한된 형태로만 사용되도록 그 용도를 축소하게 된다.

이와 같은 단점이 발생한 원인을 분석해 보면, 광학식 모션 캡쳐 패러다임의 기본 전략이 멀티비전(multi-vision)을 사용하여 카메라의 2차원 정보로부터 3차원 정보를 복원해 내는 것이기 때문이다. 즉, 멀티비전 시스템을 구축하기 위해서 다수의 카메라 장비를 필요로 하고, 카메라들의 최적 각도를 확보하기 위해서 넓은 전용 스튜디오를 필요로 하게 되며, 카메라 영상 사이의 대응점(corresponding point)을 쉽게 찾기 위해서 표시자(marker)를 부착해야 한다. 표시자를 사용해야 하는 또 하나의 이유는 3차원 정보 복원을 위해 사용하는 영상 기반 운동 렌더링 알고리즘으로서 이산 접근법(discrete approach)를 사용하기 때문이다. 이산 접근법에서는 특이점(feature points)을 설정하고 이 특이점의 위치를 수시로 인식하고 모니터링해야 하기 때문에, 특이점 위치에 표시자를 부착하여 인식이 쉽게 해야 한다.

이와같은 기존의 광학식 모션 캡쳐의 단점을 보완하여 영상 기반 모션 캡쳐 기술이 좀 더 편리하게 사용될 수 있도록 하기 위해서 본 논문에서는 몇 가지 전략을 도입하고자 한다. 먼저 다수의 카메라 장비와 넓은 전용 스튜디오의 사용을 피하기 위해서 모노 비전(mono-vision) 방식을 채택한다. 또한 표시자 사용을 피하고 계산량을 줄이기 위해서 증분형 알고리즘을 채택한다. 이와 같은 연구 전략을 통해서 본 논문에서는 별도의 스튜디오나 장비를 사용하지 않고, 카메라 1대를 사용해서 모션 캡쳐하고자 하는 대상을 녹화하고, PC에서 녹화된 동영상을 분석해서 모션 캡쳐를 수행하는 편리한 광학식 모션 캡쳐 방법을 연구한다. 이와 같은 광학식 모션 캡쳐 기술의 개발에서는 영상 기반 운동 렌더링 알고리즘의 설계가 핵심을 이루게 된다. 본 논문에서는 표시자 없이 3차원 운동을 복원하고, 계산량이 적으면서도 결과의 정확도가 높은 증분형 영상 기반 운동 렌더링 알고리즘을 개발하는데 주력한다.

3. 표시자를 사용하지 않는 영상 기반 운동 렌더링

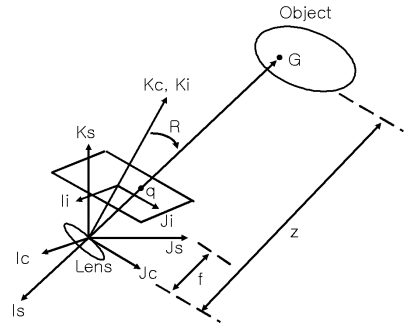
이 절에서는 영상 기반 모션 캡쳐의 핵심이 되는 영상기반 운동 렌더링 알고리즘을 제안하고자 한다. 본 논문에서는 표시자를 사용하지 않는 모노 비전 용 영상 기반 운동 렌더링 알고리즘을 제안하기 위해서, 증분형 알고리즘(incremental algorithm)을 채택하고자 한다. 그리고 이러한 증분형 영상 기반 운동 렌더링 알고리즘의 단점인 정확도를 향상 시키는 연구를 수행하고자 한다. 이를 위해 먼저 기존의 증분형 영상 기반 운동 렌더링의 문제점을 분석하고, 이를 개량한 증분형 영상 기반 운동 렌더링 기법을 제안하고자 한다.

3.1 기존의 증분형 영상 기반 운동 렌더링 알고리즘

영상 기반 운동 렌더링을 위한 증분형 알고리즘의 연구는 몇몇 학자들에 의해 행해져 왔다. 증분형 알고리즘에서는 특별히 특이점을 구별할 필요가 없으며, 작업 초기에 무작위로 몇 개의 렌더링 점 (rendering point)을 선정하면 된다. 그리고 렌더링 점에서의 영상 속도를 측정해서, 렌더링 점의 다음 위치를 추정해 낸다. 따라서 다음 영상 프레임에서 렌더링 점의 위치를 찾아 내기 위해서, 렌더링 점 위치에 표시자를 부착하는 일은 하지 않아도 된다.

그럼 영상 기반 운동 렌더링에 사용되는 기존의 증분형 알고리즘을 구체적으로 설명하기 위해서, (그림 1)에 주어진 예제 문제를 생각해 보자.

공간좌표계 {SF}는 공간상에 고정된 좌표계로서, 원점이 카메라의 렌즈 중심에 있고, Ks-축은 이점에서 렌즈에 수직하게 놓여 있다. 카메라는 공간좌표계 {SF}의 원점을 중심으로 자유롭게 회전할 수 있으며 카메라 좌표계 {CF}가 부착되어 있다. 편의상 카메라 좌표계 {CF}의 원점과 공간좌표계 {SF}의 원점이 일치하도록 놓여있다. 또한 영상(image)



(그림 1) 카메라에 의한 대상체의 관찰
(Fig. 1) Inspection of an object by a camera

평면은 렌즈 중심으로부터 카메라 좌표계 {CF}의 Kc-축을 따라 초점거리(focal length) f의 거리에 있다. 이 영상 평면에는 이미지 좌표계 {IF}가 부착되어 있는데, 세 축 Ii, Ki, Ji은 카메라 좌표계 {CF}의 각 축과 같은 방향으로 놓여 있다.

주어진 대상체(object) 상의 한 렌더링 점 $G \in R^3$이 경사투영(perspective projection)에 의해 이미지 상에 q 점으로 투영되었다고 하자. 이 경우 다음의 관계식이 성립한다.

$$G = (x, y, z)^T$$
$$q = f(G/z) = f(x/z, y/z, 1) \qquad \text{식(1)}$$

여기서 x, y, z 는 각각 카메라 좌표계 {CF}에서 바라 본 G점의 Ic, Jc, Kc 축 방향 성분이다.

주어진 강체가 각속도 $w = [w_1, w_2, w_3]^T \in R^3$와 선속도 $v = [v_1, v_2, v_3]^T \in R^3$로 움직인다고 생각해 보자. 이 경우, 강체상의 점 $G \in R^3$은 아래 식으로 표현되는 강체 운동을 하게 된다.

$$\dot{G} = [w]G + v \qquad \text{식(2)}$$

여기서,

$$[w] = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix}$$

식(1)과 식(2)를 결합하고, $[w]q = -[q]w$의 관계식을 사용하면 아래 관계식을 얻게 된다.

$$\dot{q} - A(-[q]w + av) = 0 \qquad \text{식(3)}$$

$$\dot{a} = e_3^T(-[q]w + av) \qquad \text{식(4)}$$

여기서, $a = f/z$,는 역 거리(inverse depth)이고, $A = I - qe_3^T$, e_3는 카메라의 광축 방향의 단위벡터이다.

식(3)과 식(4)는 주어진 강체 상의 모든 점에도 성립한다. 즉, 강체상의 임의의 점들 $G_i, i = 1, \cdots, n$ 을 선택했을 때, 이 점들에 대해서도 다음과 같은 식이 성립된다.

$$\dot{q_i} - A_i(-[q_i]w + a_i v) = 0,\ i = 1, \cdots, n \qquad \text{식(5)}$$

$$\dot{a_i} = e_3^T(-[q_i]w + a_i v),\ i = 1, \cdots, n \qquad \text{식(6)}$$

여기서, $a_i = f/z_i$, $A_i = I - q_i e_3^T$, e_3는 카메라의 광축 방향의 단위벡터이다.

대상체상의 영상 위치와 영상 속도 쌍 $(q_i, \dot{q_i}),\ i = 1, \cdots, n$ 이 주어졌을 경우, 식(5)와 식(6)을 활용하여 강체의 3차원 운동 (w, v)를 복원해 낼 수 있다. 이 3차원 정보 복원 문제는 흔히 최적화 문제로 정식화 된다. 예를 들면, 최적화를 위한 목적함수로서 다음 식을 선정할 수 있다.

$$L(w, v) = \sum_{i=1}^{n} \|\dot{\tilde{q_i}} - A_i(-[q_i]w + a_i v)\|^2 \qquad \text{식(7)}$$

여기서, $\dot{\tilde{q}}_i$ 는 $\dot{q_i}$ 의 측정값을 의미한다.

식(7)은 $\dot{\tilde{q}}_i$, q_i, a_i가 주어지는 경우에, w와 v에 대해 선형 형태가 된다.

초기의 a_i 값이 주어졌다고 하면, 식(7)을 최소화 하는 (w, v)를 구하고, 다시 식(6)을 사용하여 $\dot{a_i}$ 를 계산한다. 그리고 $\dot{a_i}$를 사용하여, 다음 영상 프레임에서의 a_i의 위치를 계산한다. 이렇게 얻어진 a_i 를 식(7)에 대입한 후, 식(7)를 최소화 하는 (w, v)를 찾아낸다. 이와 같이 식(6)을 사용한 a_i 의 예측 과정과 식(7)를 사용한 (w, v)의 최적값 계산을 매 영상 프레임마다 반복하여 강체의 3차원 운동을 복원해 낸다.

이와 같이 증분형 알고리즘은 작업 초기에 무작위로 선정된 렌더링 점(rendering point)의 영상 속도를 측정해서, 렌더링 점의 다음 위치를 추정해 내기 때문에, 렌더링 점 위치에 굳이 표시자를 부착하지 않아도 된다는 장점이 있다. 그리고 증분형 알고리즘의 또 하나의 장점은 렌더링 점의 영상 좌표와 거리파라미터를 증분하여 얻어냄으로써, 최적화 부분이 선형 형태로 단순화 될 수 있다는 점이다. 그러나 증분형 알고리즘의 단점은 렌더링 점의 위치를 계속적으로 합산해 나가기 때문에, 시간이 지날수록 누적 오차가 커져서 결국 영상 기반 운동 렌더링에 실패하게 된다는 것이다.

본 연구에서는 증분형 알고리즘에서의 오차 누적 문제를 축소하여, 증분형 알고리즘의 안정성을 높이고자 한다. 이를 위해서 먼저 영상 기반 운동 렌더링에 사용되는 기존의 증분형 알고리즘을 분석한 후, 좀 더 안정적인 증분형 알고리즘을 제안하고자 한다.

3.2 기존의 증분형 영상 기반 운동 렌더링 알고리즘을 최적 제어의 견지에서 재해석

시스템의 해석 시 종종 상태 변수와 상태 방정식의 개념

이 도입된다. 상태 변수란 어떤 시스템의 상태를 표현할 수 있는 내부 파라미터 집합을 말하며, 상태 방정식이란 상태 변수가 주어진 시스템의 상태를 올바로 표현하기 위해서 만족해야 하는 관계식이다. 만일 $X \in R^n$이 상태 변수라면, 상태 방정식은 일반적으로 다음의 형태를 갖게 된다.

$$\dot{X} = f(X, U) \qquad \text{식(8)}$$

여기서, $U \in R^m$은 시스템의 변화를 가져오는 요인, 즉 시스템 입력이 된다.

이러한 견지에서 식(6)을 살펴 보도록 하자. 만일 우리가 a_i 를 상태 변수로 선택한다면, 식(6)는 이 상태 변수에 대한 상태 방정식의 형태, 즉 식(8)이 된다. 이 경우, (w, v)가 시스템 입력에 해당하게 된다. 식(6)과 식(8)의 외관을 비교해 볼 때, 이와 같은 해석이 가능하다. 마찬가지로 식(5)와 식(8)의 외관을 비교해 보면 q_i 를 상태 변수로 선택할 경우, 식(5)는 이 상태 변수에 대한 상태 방정식의 형태, 즉 식(8)이 된다.

그러면 이와 같은 수식 상의 해석이 물리적으로도 의미가 있는지 살펴 보도록 하자. 만일 대상체가 하나의 동역학 시스템이라 가정해 보자. 이 경우, 이 대상체에 (w, v)로 표현되는 운동을 일으킬 경우, 그에 대한 응답으로 대상체의 위치가 바뀌게 된다. 대상체의 위치는 카메라 광축에 평행한 2차원 영상 좌표(q_i)와 광축에 수직한 역 거리(a_i)가 있다. 이 중 영상 좌표는 측정에 의해 얻어질 수 있으므로, 결국 대상체의 상태인 위치는 역 거리, 즉 a_i 에 의해 결정된다. 따라서 대상체의 위치 상태는 a_i 라는 상태 변수에 의해 표현되고, 위치 상태를 변화 시키는 운동 입력은 (w, v)가 된다. 그리고 주어진 운동 입력에 대해서 a_i는 식(4)가 표현하는 관계를 만족하며, 그 값이 바뀌게 된다. 결국, 앞에서 살펴 본 식(4)에 대한 해석이 물리적으로도 의미가 있음을 알 수 있다.

그럼 지금부터 $X = (a_1, \cdots, a_n)$ 를 상태 변수, $U = (w, v)$를 시스템 입력, 식(4)를 상태 방정식($\dot{X} = f(X, U)$)으로 간주하고 앞 절에서 살펴 본 기존의 증분형 영상 기반 운동 렌더링 알고리즘을 해석해 보자. 그러면 주어진 초기값 $X(t_0) = (a_1(t_0,), \cdots, a_n(t_0))$에 대해, 앞 절의 알고리즘은 다음 형태로 정리될 수 있다.

단계1	주어진 $X(t)$에 대해, 다음 목적함수를 최소화하는 시스템 입력 $U(t)$를 구한다. $L(X) = L(w, v)$ 식(9) 여기서, $L(w, v)$는 식(7)에 주어져 있다.
단계2	주어진 시스템 입력 $U(t)$에 대해서, 상태 방정식을 적분하여 상태 변수를 계산한다. 이 계산 과정은 다음의 적분식으로 표현된다. $X(t) = \int_{t_0}^{t} \dot{X} dt + X(t_0)$ 식(10) 여기서, $\dot{X} = f(X, U)$이고, $X(t_0)$는 상수이므로 최적화 과정 중에 무시될 수 있다.
단계3	단계1. 단계2의 과정을 반복하여 모든 t에 대한 $X(t)$, $U(t)$를 구한다.

이 과정을 살펴 보면, 최적 제어 패러다임과 상당히 흡사함을 알 수 있다. 비교를 위해서 잠시 최적 제어 패러다임을 살펴보도록 하자.

단계1	주어진 $X(t)$에 대해, 다음 목적함수를 최소화하는 시스템 입력 $U(t)$를 구한다. $J(t) = \phi(t_f) + \int_{t_0}^{t_f} L(\tau) d\tau$ 식(11) 여기서, $\phi(t_f)$는 최종 시간의 목표치를 표현한다.
단계2	주어진 시스템 입력 $U(t)$에 대해서, 상태 방정식을 적분하여 상태 변수를 계산한다. 이 계산 과정은 다음의 적분식으로 표현된다. $X(t) = \int_{t_0}^{t} \dot{X} dt + X(t_0)$ 식(12) 여기서, $\dot{X} = f(X, U)$
단계3	단계1. 단계2의 과정을 반복하여 모든 t에 대한 $X(t)$, $U(t)$를 구한다.

결국 식(9)을 식(11)의 형태로 확장하면, 영상 기반 운동 렌더링 알고리즘이 최적 제어 패러다임과 일치하게 된다. 그러면 식(9)와 식(11)의 차이점을 잠시 살펴 보도록 하자. 적분 구간을 세분화하면 식(11)의 적분항은 다음 형태로 근사화 될 수 있습니다.

$$J(X) \approx \phi(t_f) + \sum_{i=1}^{N} L(X_i) \Delta t \qquad \text{식(13)}$$

여기서, $X_i = X(t_0) + N\Delta t$ 이다.

Δt의 값은 상수이기 때문에, 결과적으로 식(13)의 의미는 주어진 현재 시간 t에 대해 $L(X_i)$를 최적화 한후, 그 결과를 더한 것이라 할 수 있다. 따라서 식(9), 식(10)에 의해 기술된 알고리즘은 최종 시간의 목표치를 생략하고, 최적 제어 패러다임을 작은 구간으로 쪼개어 계산한 후, 각 구간의 결과를 누적하는 방식으로 최적제어 패러다임을 근사화하는 방법임을 알 수 있다. 이와 같은 재해석에 기반 하여 기존의 증분형 영상기반 운동 렌더링 알고리즘을 온전한 최적제어 형태로 확장할 경우, 좀 더 정확한 해를 구할 수 있을 것이라는 착안을 하게 된다.

3.3 제안하는 증분형 영상 기반 운동 렌더링 알고리즘

앞 절에서 살펴 본 기존의 증분형 영상 기반 운동 렌더링 알고리즘의 재해석에 기반해서, 본 절에서는 영상기반 운동 렌더링 알고리즘을 온전한 최적 제어 형태로 확장하고자 한다. 즉, 영상기반 운동 렌더링의 문제에 최종 시간의 목표치를 추가하고, 각 세부 시간 별 국부 최적화 대신 전 운동 렌더링 전 시간에 걸친 전역 최적화 문제로 확장할 것을 제안한다.

제안하는 최적화를 위한 목적함수는 아래와 같다.

$$J(t) = \phi(t_f) + \alpha \int_{-\infty}^{t} L(\tau) d\tau \qquad \text{식(14)}$$

여기서, $\alpha \in R$는 첫 번째 항과 두 번째 항 사이의 가중치이고. $L(w, v)$는 식(9)에 주어져 있으며, $\phi(t)$는 최종 목

표치로서 본 연구에서는 다음과 같이 정의 하였다.

$$\phi(t) = \sum_{i=1}^{n} \|\widetilde{q_i}(t) - q_i(t)\|^2 \qquad \text{식(15)}$$

여기서 $\widetilde{q_i}(t)$는 $q_i(t)$의 측정값을 의미한다.

$q_i(t)$는 식(5)을 적분하여 얻게 된다. 즉, 다음 식에 의해 구해 진다.

$$q_i(t) = \int_{t_0}^{t} \dot{q_i} dt + q_i(t_0)$$

여기서 $\dot{q_i} = A(-[q_i]w + a_i v)$ 식(16)

식(14)의 두 번째 항은 다음과 같이 변형 될 수 있다.

$$\int_{-\infty}^{t_f} L(U(\tau), X(\tau))d\tau = L(U(t_0), X(t_0)) + \int_{t_0}^{t} L(U(\tau), X(\tau))d\tau \qquad \text{식(17)}$$

여기서 $L(U(t_0), X(t_0))$는 상수이므로 최적화 과정에서 무시 될 수 있다. 그러면 식(14)는 다음과 같이 바꿔 쓸 수 있다.

$$J(t) = \phi(t_f) + \int_{t_0}^{t} L(U(\tau), X(\tau)d\tau \qquad \text{식(18)}$$

이상의 내용들을 종합하여 본 연구에서 제안하는 최적 제어 패러다임에 입각한 영상 기반 운동 렌더링 알고리즘은 다음과 같이 요약 될 수 있다.

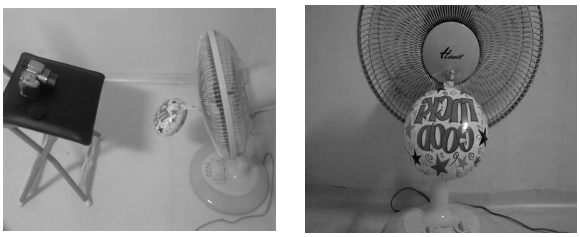
단계1	주어진 상태변수 $X(t) = (a_i(t), q_i(t))$에 대해, 다음 목적함수를 최소화하는 시스템 입력 $U(t) = (w(t), v(t))$를 구한다. $J(t) = \phi(t_f) + \int_{t_0}^{t_f} L(\tau)d\tau$ 식(19) 여기서, $L(w, v)$는 식(7)에 주어져 있고, $\phi(t)$는 식(15)에 주어져 있다.
단계2	주어진 시스템 입력 $U(t)$에 대해서, 상태 방정식을 적분하여 상태 변수를 계산한다. 이 계산 과정은 다음의 적분식으로 표현된다. $a_i(t) = \int_{t_0}^{t} \dot{a_i} dt + a_i(t_0)$ 식(20) $q_i(t) = \int_{t_0}^{t} \dot{q_i} dt + q_i(t_0)$ 여기서, $\dot{a_i}$ 와 $\dot{q_i}$ 는 각각 식(6)와 식(5)에 의해 구해진다.
단계3	단계1. 단계2의 과정을 반복하여 모든 t에 대한 $(a_i(t), q_i(t))$, $(w(t), v(t))$을 구한다.

4. 결 과

모션 캡쳐 기술은 주로 인간의 운동을 애니메이션화 하는데 많이 사용되어 왔지만, 복잡한 역학 계산이 필요한 사실적인 컴퓨터 애니메이션 제작에도 유용하게 사용될 수 있다. 예를 들어, (그림 2)에 주어진 풍선이 바람에 날리는 불규칙한 운동은 예측하기 어려운 변수들이 많아서 수학적으로 모델링하기 매우 어렵다. 그리고 섣불리 단순화한 운동 모델을 사용할 경우, 풍선 운동의 자연스러움이 깨지게 된다. 이처럼 정확한 수학적인 모델 설정이 어려운 경우, 무리한 단순화를 통한 수학적 모델의 도출 보다는 모션 캡쳐 기술을 사용하는 것이 운동의 자연스러움을 보존할 수 있는 더 효과적인 방법이다.

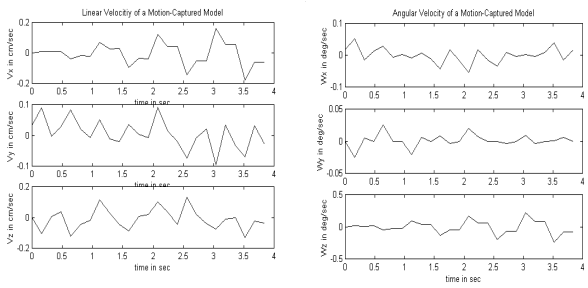
이러한 취지에서 본 절에서는 (그림 2)에 주어진 풍원 근처의 풍선 운동을 모션 캡쳐를 통해 모델링 하는 시나리오를 살펴본다. 본 시나리오에서는 그림의 왼쪽에 보이는 카메라를 사용해서 동영상으로 녹화하고, 저장된 동영상에 본 논문에서 제안한 증분형 영상기반 운동 렌더링 알고리즘을 적용하여 풍선의 운동을 복원해 낸다.

(그림 2)의 오른쪽 사진은 모션 캡쳐의 대상체가 되는 풍선의 확대 사진을 보여 주고 있다. 풍선 위에 편의상 선정한 렌더링 점(rendering point)을 숫자로 표시하였다. 이 렌더링 점의 선정은 특별한 제약조건은 없고 영상 기반 운동 렌더링 초기에 사용자가 임의로 선정할 수 있다. 이와 같은



(그림 2) 바람에 날리는 풍선의 모션캡쳐를 위한 실험 장치. 렌더링 점은 아라비아 숫자로 표시 되어 있다

(Fig. 2) Experiment set for motion capture of a balloon blown off by wind. Rendering points are denoted by arabic numbers



(그림 3) 모션 캡쳐되는 모델의 선속도 (왼쪽 그림)와 각속도 (오른쪽 그림) - 각속도와 선속도가 작은 경우.

(Fig. 3) Linear velocity (left figure) and angular velocity (right figure) of a motion-captured model - case of small linear and angular velocity

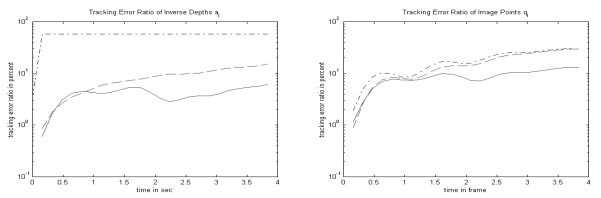
실험 세트를 통해 영상의 초기 화면에서의 실험 조건을 얻고, 초기 화면 이후의 풍선의 운동은 사인, 코사인 함수와 랜덤 함수를 섞어서 만든 임의의 풍선 운동을 선정하여, 그 운동에 대한 시뮬레이션 결과를 분석하였다.

(그림 3)의 왼쪽 그림과 오른쪽 그림은 본 시뮬레이션에 사용된 풍선의 운동 매개변수, 즉 참값으로 사용된 각속도와 선속도의 시간 변화를 보여 주고 있다. 가능한 풍선의 실제 운동을 모사할 수 있도록 회전 운동과 선형 운동을 불규칙하게 배치하였다.

이렇게 인위적으로 생성된 시뮬레이션 시나리오를 사용해 분석 한 이유는 제안한 알고리즘의 정확한 성능 분석을 위해서 풍선 운동의 참값을 정확히 알아야 하기 때문이다. 하지만 비록 설정된 풍선 운동에 대해 분석을 한다고 해도 가능한 실제 상황에 밀접한 시뮬레이션 조건을 얻을 수 있도록 하기 위해서 실험의 초기 설정은 실험 데이터를 따랐다. 이렇게 설정된 시뮬레이션 시나리오에 본 논문에서 제안한 시간적 진화(time-evolution)를 포함해 최적화 된 증분형 알고리즘을 적용하였다. 그리고 성능 비교를 위해서, 기존의 유사 유형의 알고리즘, 즉 시간적 진화(time-evolution)를 포함하지 않고 최적화된 기존의 증분형 알고리즘을 함께 적용하였다. 또한 Bregler가 제안한 알고리즘[12]도 함께 적용하였다. 본 시뮬레이션은 504MB 램을 장착한 1.20GHz의 인텔 펜티움 PC에서 MATLAB을 사용해 수행되었다.

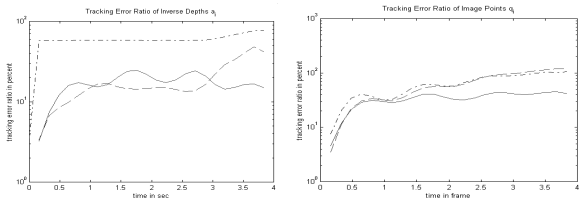
4.1 영상 잡음의 크기 변경 실험

거리(depth)의 초기 측정값과, 렌더링 점의 영상 위치값에 각각 0.5 화소 잡음(0.5-pixel noise)을 더했을 경우의 영상 기반 운동 렌더링 결과가 (그림 4)~(그림 9)에 주어져 있다. 여기서 성능 평가 지수로는 렌더링 오차율(rendering error ratio), 즉 참 값에 대한 오차의 비율을 선택하였다.
(그림 4)의 왼쪽 그림과 오른쪽 그림은 (그림 2)에 표시된 렌더링 점의 위치에 대한 렌더링 오차율을 보여주고 있다. 이 두 그림에서 실선은 본 논문에서 제안한 방식의 결과를 나타내고, 점선은 기존의 증분형 알고리즘의 결과를 나타내며, '-.' 선은 Bregler 알고리즘을 나타낸다. (그림 4)의 왼쪽 그림은 영상 좌표의 렌더링 오차율을 보여주고 있고, (그림 4)의 오른쪽 그림은 역 거리(inverse depth)의 렌더링 오차율을 보여주고 있다.



(그림 4) 영상 잡음이 0.5-화소인 경우에 거리의 역수 a_i (왼쪽 그림) 와 영상 점 q_i (오른쪽 그림)의 렌더링 오차율 (세로축은 로그 스케일로 표시)

(Fig. 4) Rendering error ratio of inverse depths a_i (left figure) and image points q_i when image noise is 0.5-pixels (vertical axis is in log scale)



(그림 5) 영상 잡음이 2-화소인 경우에 거리의 역수 a_i (왼쪽 그림)와 영상 점 q_i (오른쪼 그림)의 렌더링 오차율 (세로축은 로그 스케일로 표시)

(Fig. 5) Rendering error ratio of inverse depths a_i (left figure) and image points q_i (right figure) when image noise is 2-pixels (vertical axis is in log scale)

이 그림들을 보면 본 논문에서 제안한 증분형 알고리즘이 기존의 증분형 알고리즘과 Bregler 알고리즘에 비해, 영상 위치와 역 거리 두 면에서 모두 렌더링 오차율이 적음을 알 수 있다. 그리고 렌더링 오차율의 향상도는 영상 기반 운동 렌더링이 진행될수록 더욱 뚜렷해지는 것을 볼 수 있다. 이는 렌더링 오차의 향상분이 시간에 따라 누적되기 때문이다.

(그림 5)의 왼쪽 그림과 오른쪽 그림은 렌더링 점의 위치값에 2 화소 잡음(2-pixel noise)을 더했을 경우의 영상 기반 운동 렌더링 결과를 보여주고 있다. 이 그림들을 살펴보면 영상 잡음이 증가하였어도, 여전히 본 논문에서 제안한 증분형 알고리즘이 기존의 증분형 알고리즘과 Bregler 알고리즘에 비해, 영상 위치와 역 거리 두 면에서 모두 렌더링 오차율이 적은 것을 볼 수 있다.

(그림 4)~(그림 5)를 살펴보면, 본 논문에서 제안한 증분형 알고리즘이 잡음의 크기에 상관없이 기존의 증분형 알고리즘과 Bregler 알고리즘 보다 영상 위치와 역 거리에 대해 적은 렌더링 오차율을 보임을 알 수 있다. 반면, Bregler 알고리즘은 기존의 증분형 알고리즘에 비해 영상 위치 렌더링 오차율은 비슷하지만 역 거리 렌더링 오차율이 많이 떨어지는 것을 볼 수 있다. 그 이유는 Bregler 알고리즘이 선형화된 알고리즘을 얻기 위해 경사 투영(perspective projection)을 스케일이 조정된 직교 투영(scaled orthographic projection)으로 근사화한데 원인이 있다.

반면 Bregler 알고리즘의 역거리 오차율은 기존의 증분형 알고리즘에 비해 시간 경과에 따른 증가폭이 크지 않다. 그 이유는 Bregler 알고리즘은 선형 근사를 통해 얻어진 간단한 계산 공식을 사용하여 시간 적분을 하지 않아도 거리 매개 변수를 구할 수 있기 때문이다.

<표 1>~<표 2>는 (그림 4)~(그림 5)의 결과를 영상 잡음에 대해 정리한 통계치를 보여주고 있다. <표 1>은 영상 위치에 대한 평균 렌더링 오차율과 최종 렌더링 오차율을 보여 주고 있다. 그리고 <표 2>는 역 거리에 대한 평균 렌더링 오차율과 최종 렌더링 오차율을 보여주고 있다. 이 표들을 살펴보면 본 논문에서 제안한 증분형 알고리즘이 기존의 증분형 알고리즘과 Bregler 알고리즘에 비해 렌더링의 평균 오차율과 최종 오차율 모두에서 우수성을 보이고 있음을 확인할 수 있다. 그 이유는 본 논문에서 제안한 증분형

〈표 1〉 영상 잡음에 대한 영상 점 q_i의 평균 운동 렌더링 오차율 / 최종 운동 렌더링 오차율

〈Table 1〉 Average motion rendering error ratios / final motion rendering error ratios of image points q_i in the respect of image noises

	0.5-화소 영상잡음	2-화소 영상잡음
Bregler	17.1673 / 29.7686	61.2268 / 104.5930
기존의 증분형	15.3228 / 29.8039	61.3926 / 119.2335
제안하는 증분형	8.3778 / 12.8336	32.7231 / 41.2343

〈표 2〉 영상 잡음에 대한 거리의 역수 a_i의 평균 운동 렌더링 오차율 / 최종 운동 렌더링 오차율

〈Table 2〉 Average motion rendering error ratios / final motion rendering error ratios of inverse depths a_i in the respect of inverse depths

	0.5-화소 영상잡음	2-화소 영상잡음
Bregler	55.6571 / 58.0418	58.5510 / 75.9484
기존의 증분형	8.0124 / 15.0881	17.9768 / 41.8490
제안하는 증분형	3.9058 / 6.0474	16.3652 / 14.9057

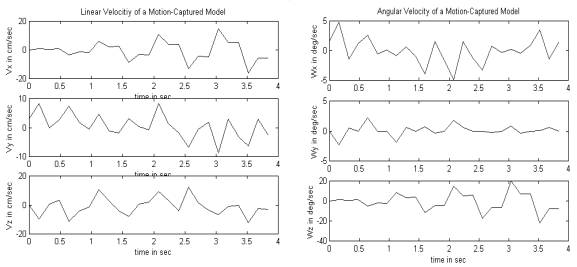
알고리즘은 렌더링 오차율의 누적 값뿐만 아니라 최종값도 최적화하도록 설계 되었기 때문이다.

<표 1>~<표 2>를 살펴보면, Bregler 알고리즘과 본 논문에서 제안한 증분형 알고리즘이 상대적으로 영상 잡음에 덜 영향 받고, 기존의 증분형 알고리즘이 영상 잡음에 민감하게 영향 받음을 볼 수 있다. 그 결과로서 2-화소의 고 영상 잡음의 경우에는 Bregler 알고리즘이 기존의 증분형 알고리즘보다 우수한 영상 렌더링 오차율을 보임을 표1에서 확인할 수 있다.

4.2 대상체의 운동 속도 변경 실험

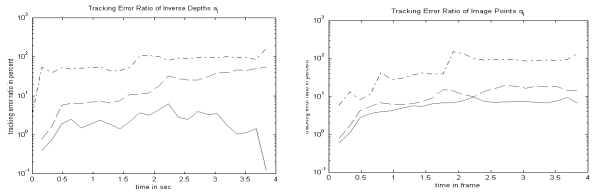
지금까지는 영상 잡음의 크기를 바꿔가면서 Brelger알고리즘, 기존의 증분형 알고리즘, 본 논문에서 제안한 증분형 알고리즘을 비교 분석하였다. 이번에는 대상체의 속도를 바꿔서 이 세 가지 알고리즘을 비교 분석해 보도록 하자.

(그림 6)의 왼쪽 그림과 오른쪽 그림은 본 시뮬레이션에 사용할 풍선의 운동 매개변수, 즉 참값으로 사용된 각속도



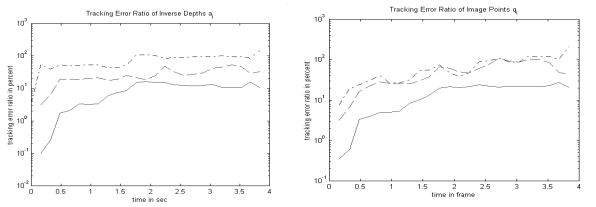
(그림 6) 모션 캡쳐되는 모델의 선속도 (왼쪽 그림)와 각속도 (오른쪽 그림) - 각속도와 선속도가 큰 경우

(Fig. 6) Linear velocity (left figure) and angular velocity (right figure) of a motion-captured model - case of large linear and angular velocity



(그림 7) 영상 잡음이 0.5-화소인 경우에 거리의 역수 a_i (왼쪽 그림)와 영상 점 q_i (오른쪽 그림)의 렌더링 오차율 (세로축은 로그 스케일로 표시)

(Fig. 7) Rendering error ratio of inverse depths a_i (left figure) and image points q_i (right figure) when image noise is 0.5-pixels (vertical axis is in log scale)



(그림 8) 영상 잡음이 2-화소인 경우에 거리의 역수 a_i (왼쪽 그림)와 영상 점 q_i (오른쪽 그림)의 렌더링 오차율 (세로축은 로그 스케일로 표시)

(Fig. 8) Rendering error ratio of inverse depths a_i (left figure) and image points q_i (right figure) when image noise is 2-pixels (vertical axis is in log scale)

와 선속도의 시간 변화를 보여 주고 있다. (그림 6)의 왼쪽 그림과 오른쪽 그림은 (그림 3)에 비해 각속도와 선속도의 크기가 큰 경우에 해당함을 확인할 수 있다.

거리(depth)의 초기 측정값과, 렌더링 점의 영상 위치값에 각각 0.5 화소 잡음(0.5-pixel noise)을 더했을 경우의 영상 기반 운동 렌더링 결과가 (그림 7)의 왼쪽 그림과 오른쪽 그림에 주어져 있고, 2 화소 잡음의 경우가 (그림 8)의 왼쪽 그림과 오른쪽 그림에 주어져 있다.

<표 3>~<표 4>을 <표 1>~<표 2>와 비교해 보면, Bregler 알고리즘의 렌더링 오차율이 큰 폭으로 상승해 있는 것을 볼 수 있다. 그 이유는 Bregler 알고리즘은 선형화를 위해서 작은 각속도를 가정하기 때문에 각속도가 큰 경우 오차가 커지기 때문이다. 결과로서 (그림 4)에서와는 달리 (그림 7)에서는 Bregler 알고리즘의 영상 위치 렌더링 오차율이 기존의 증분형 알고리즘과 비슷하지 않고 상당히 큰 것을 볼 수 있다.

〈표 3〉 영상 잡음에 대한 거리의 역수 q_i의 평균 운동 렌더링 오차율 / 최종 운동 렌더링 오차율

〈Table 3〉 Average motion rendering error ratios / final motion rendering ratios of inverse depths q_i in the respect of image noises

	0.5-화소 영상잡음	2-화소 영상잡음
Bregler	65.1460 / 136.4894	67.0024 / 212.4387
기존의 증분형	10.8604 / 13.9843	50.4236 / 43.9826
제안하는 증분형	5.7017 / 6.5154	14.5478 / 20.5747

〈표 4〉 영상 잡음에 대한 거리의 역수 a_i의 평균 운동 렌더링 오차율 / 최종 운동 렌더링 오차율
〈Table 4〉 Average motion rendering error ratios / final motion rendering error ratios of inverse depths a_i in the respect of image noises

	0.5-화소 영상잡음	2-화소 영상잡음
Bregler	75.8179 / 152.4187	76.2477 / 164.9961
기존의 증분형	20.5845 / 53.6126	25.9809 / 33.0618
제안하는 증분형	2.2289 / 0.1193	8.7763 / 10.5704

하지만, (그림 8)과 <표 3>을 살펴보면, Bregler 알고리즘의 영상 위치 렌더링 오차율이 기존의 증분형 알고리즘과 비슷해 진 것을 볼 수 있다. 그 이유는 <표 3>에서 볼 수 있듯이 기존의 증분형 알고리즘이 영상의 잡음에 매우 민감해서 (그림 8)과 같은 2-화소 고 영상 잡음의 경우에는 렌더링 오차율이 다른 두 방법에 비해 상대적으로 큰 폭으로 상승하기 때문이다.

4.3 실험결과 정리

본 절에서는 모션 캡쳐 대상체의 운동 속도와 영상 잡음을 바꿔 가면서, 본 논문에서 제안한 증분형 알고리즘을 기존의 증분형 알고리즘과 Bregler 알고리즘에 비교하여 성능을 평가하였다. 그 결과를 정리하면 다음과 같다. Bregler 알고리즘은 역 거리에 대해 렌더링 오차율이 큰 편이고, 또한 렌더링 오차율이 각속도의 크기에 민감하다. 기존의 증분형 알고리즘은 영상 위치 렌더링 오차율은 Bregler 알고리즘과 비슷한 수준이고, 역 거리 렌더링 오차율은 더 좋은 결과를 보인다. 하지만 영상 잡음에 민감하다. 반면 본 논문에서 제안한 알고리즘은 대상체의 운동 속도와 영상 잡음에 상대적으로 덜 민감하며, 영상 위치 렌더링 오차율과 역 거리 렌더링 오차율 모두에서 Bregler 알고리즘과 기존의 증분형 알고리즘에 비해 안정적으로 우수한 렌더링 오차율을 보인다.

5. 결 론

본 논문에서는 광학식 모션캡쳐에 사용되는 영상 기반 운동 렌더링 알고리즘을 연구하였다. 제안하는 영상 기반 운동 렌더링 알고리즘은 시간적 진화(time-evolution)을 고려하여 최적화된 증분형 알고리즘이다. 이렇게 설계된 영상 기반 모션 캡쳐 기술이 캐릭터 애니메이션 제작 뿐 아니라 사실적인 컴퓨터에니메이션 제작에 사용될 수 있음을 주장하였다. 그 한 예로서, 풍원 근처에서 불규칙하게 움직이는 풍선의 운동을 모델링 하는 데 본 논문의 알고리즘을 적용하였다. 결과로서 풍선상의 선택점의 위치 복원에 있어 기존의 알고리즘들에 비해 본 논문의 알고리즘이 우수한 정확도를 보였다.

참 고 문 헌

[1] P. F. McLauchlan and D. W. Murray, “A unifying framework for structure and motion recovery from image sequences,” Proc. IEEE Int. Conf. Computer Vision, Cambridge, MA, USA, pp. 314-320, 1995.

[2] R. Szeliski and S. B. Kang, “Recovering 3D shape and motion from image streams using non-linear least squares,” J. Visual Communication and Image Representation, vol. 5, no. 1, pp. 10-28, 1994.

[3] 최은옥, “스킨, 스켈레턴 바이딩 알고리즘을 이용한 다 관절체 모델의 구현”, 충북대학교 대학원, 박사학위논문, p.9, 1998.

[4] K. Choi and H. Ko, “On-line motion retargetting”, J. Visualiza. Comput. Anim., vol. 11, no. 5, pp. 223-235.

[5] H. J. Shin, J. Lee, and M. Gleicher, “Cpmuter puppetry: An importance-based approach”, ACM Trans. Graph, vol. 20, no. 2, pp. 67-94.

[6] S. Ullman, “The interpretation of structure from motion,” Proc. Royal Soc. London, vol. B203, pp. 405-426, 1979.

[7] S. Maybank, Theory of reconstruction from image motion. New York: Spring-Verlag, 1993.

[8] J. K. Aggarwal and N. Nandhakumar, “On the computation of motion from sequences of images-a review,” Proc. IEEE, vol. 7B, no. 8, pp. 917-935, 1988.

[9] J. Weng, N. Ahuja, and T. S. Huang, “Optimal motion and structure estimation,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 15, no. 9, pp. 864-884, 1993.

[10] K. Kanatani, “3d interpretation of optical flow by renormalization,” Int. J. of Computer Vision, vol. 11, no. 3, pp. 267-282, 1993.

[11] A. D. Jepson and D. J. Heeger, “Linear subspace methods for recovering translation direction,” in Spatial Vision in Humans and Robotics, (L. Harris and M. Jenkins, Eds.), Cambridge: Cambridge Univ. Press, pp. 39-62, 1993.

[12] C. Bregler, J. Malik and K.Pullen, “Twist based acquisition and tracking of animal and human kinematics”, International Journal of Computer Vision, vol. 56, no. 3, pp. 179-194, 2004.

한 영 모

e-mail : ymhan123@hanmail.net

1992년 서울대학교 물리교육학과 (이학사, 준우등 졸업)

1995년 서울대학교 제어계측공학과 (공학사, 차석 졸업)

1998년 서울대학교 대학원 전기공학부 (공학석사, 신호처리 전공)

2002년 서울대학교 대학원 기계항공공학부(공학박사, 컴퓨터비전 전공)

2002년~2003년 세종-록히드마틴 우주항공연구소 전임연구원, 전임연구교수

2004년~2005년 이화여자대학교 정보통신공학과 연구전임강사, 연구교수

2006년~2008년 한양사이버대학교 컴퓨터공학과 전임강사

2008년~현재 한양사이버대학교 컴퓨터공학과 조교수

2007년~현재 International Biographical Centre (England), Deputy Director General, Honorany Director General

2007년~현재 World Congress of Arts, Science and Communications, Vice President

2006년~현재 Marquis Who's Who(USA) 5회 등재

2007년~현재 International Biographical Centre (England) 17회 등재 및 수상

관심분야 : 컴퓨터비전 응용 멀티미디어 및 생체 영상 인식, 모바일 및 로봇 임베디드 시스템 소프트웨어, 인간과 컴퓨터의 시각적 상호작용, IT 분야의 통합 과학적 접근법 등