

---

# 3차원 캐릭터의 만화적 모션 제작 시스템

↓  
↓  
↓  
↓

## A Cartoon Motion Generation System for 3D Character

이지형\*, 구분기\*, 김종혁\*\*, 최정주\*\*, 황치정\*\*\*

---

↓  
↓

### 요약

최근, 3차원 그래픽스 기술은 애니메이션 필름 제작에 있어서 자주 사용된다. 그러나 많은 애니메이션 필름들은 3차원 그래픽스 기술을 사용하여 제작되더라도, 2차원 셀 애니메이션 효과를 내려고 한다. 3차원 그래픽스 기술을 이용하여 2차원 셀 애니메이션의 효과를 내기 위해서는 카툰 렌더링이 사용 된다. 그러나 이를 제외하고도 몇 가지 기술이 더 필요한데, 그중 하나가 만화 캐릭터(character)의 만화스러운 움직임이다. 기존의 연구 중에는 기존의 모션을 만화적 모션으로 변형하려는 시도가 있었으나, 셀 애니메이션의 캐릭터 움직임과 차이가 있었다. 또 생성된 모션이 만화적 모션인가 하는 의문에 대한 평가 기준이 없기 때문에, 만화적 모션의 모호성 문제가 발생하였다.

본 논문에서는 직접 애니메이션에서 모션을 얻어내는 시스템을 제안한다. 2차원 애니메이션 동영상에서 2차원 캐릭터의 자세를 보고 3차원 캐릭터의 자세로 반자동 맵핑하여, 3차원 캐릭터의 애니메이션 키 프레임을 생성하고, 이 키 프레임간의 보간을 통해 3차원 캐릭터 애니메이션을 생성한다. 생성된 3차원 캐릭터 애니메이션은 만화적 움직임을 갖게 되며, 2차원 캐릭터의 자세와 움직임을 기준으로 만들었기 때문에, 만화적 모호성을 극복할 수 있다.

↓  
↓  
↓

**핵심어:** *Computer graphics, Computer animation, Non-photorealistic Animation & Rendering, 3D character, Motion generation, Cartoon motion*

↓

## 1. 서론

최근, 3차원 그래픽스 기술은 애니메이션 필름 제작에 있어서 자주 사용된다. 그러나 많은 애니메이션 필름들은 3차원 그래픽스 기술을 사용하여 제작되더라도, 2차원 셀 애니메이션 효과를 내려고 하고, 비사실적 렌더링, 애니메이션(NPAR, non-photorealistic animation and rendering) 기술이 대두되었다[1].

NPAR 기술 중 카툰 렌더링의 경우, 3차원 그래픽스 기술을 이용하여 2차원 셀 애니메이션의 효과를 제공한다. 카툰 렌더링이 NPAR에서 널리 쓰이게 되면서, 일부 게임과 영화에서는 카툰 렌더링을 제작에 활용하기도 하였다. 그러

나 2차원 셀 애니메이션 효과를 내기 위해서는 카툰 렌더링을 제외하고도 몇 가지 기술이 더 필요하다. 그중 하나가 만화 캐릭터(character)의 만화스러운 움직임이다. 애니메이션에서 관객은 만화 주인공이 사실적으로 움직인다면 어딘가 부자연스럽게 느낄 것이다. 캐릭터가 비사실적인 움직임을 보이기 위해서는 2가지 방법이 있다. 첫 번째는 애니메이터가 애니메이션 작업시 키 프레임(key frame)을 만화스타일로 잡는 것인데, 이 경우 애니메이터의 많은 노력이 필요하고, 애니메이션의 질은 애니메이터의 능력에 좌우 된다. 두 번째 방법은 모션 캡처와 같은 기존의 사실적인 모션데이터를 변형하여 만화 같은 움직임을 만들어 내는 것이다. 이 경우, 애니메이터의 작업은 줄어들지만 만화적 모션에 대한 기

---

본 연구는 정보통신부 및 정보통신연구진흥원의 IT신성장동력핵심기술개발사업의 일환으로 수행하였음. [2005-S082-03, 비사실적 애니메이션 기술 개발]

\*주저자 : 한국전자통신연구원 디지털콘텐츠연구단, \*\*공동저자 : 아주대학교 미디어학부, \*\*\*공동저자 : 충남대학교 컴퓨터공학과

\*교신저자 : 이지형 ; e-mail: ijihyung@etri.re.kr

준이 없기 때문에 만화적 모션의 모호성 문제가 발생한다.

본 논문에서는 캐릭터의 만화적 움직임을 직접 애니메이션 영상에서 생성하는 시스템을 제안한다. 생성된 3차원 캐릭터 애니메이션은 2차원 캐릭터의 자세와 움직임을 기준으로 만들었기 때문에, 만화적 모호성을 극복할 수 있다.

## 2. 관련 연구

3차원 캐릭터 애니메이션은 고전적인 연구 주제가 되었다. 더욱이 모션 캡처 장비의 보급으로 3차원 캐릭터의 애니메이션은 보다 쉬워지는 듯하였다. 그러나 모션 캡처의 고비용, 수정의 어려움 등은 애니메이션 제작에서 키 프레임 애니메이션을 대체하지 못하였고, 보다 쉽게 3차원 캐릭터 애니메이션을 얻기 위한 방법이 지속적으로 연구되었다. 그중 하나가 2차원 비디오를 이용하여 3차원 캐릭터 애니메이션을 얻는 것으로, 컴퓨터 비전에서 지속적으로 연구되어온 인간의 움직임을 인식하여 활용하는 것이다[2]. [3]의 경우 2차원 비디오에서 캐릭터의 모션을 인식하고 이를 3차원으로 확장하는 것으로, 사실적인 캐릭터 모션으로 제한을 두어 만화적 동작을 제작하기에는 한계가 있었다.

만화적 모션을 얻기 위해, 기존 3차원 애니메이션 데이터에 필터(filter) 처리를 한 연구도 있었다[4]. 간단한 필터링에 의해 모션이 변형되어 쉽게 애니메이션을 생성할 수 있으나 생성된 모션이 만화적인가에 대한 모호성 문제가 남는다. 최근에는 본 논문과 같이 2차원 비디오에서 직접 만화적 모션을 얻어내는 다양한 연구가 시도되고 있다. 비디오에서 직접 만화적 모션을 캡처하는 연구[5]의 경우, 만화적 모션의 모호성을 해결하였으나, affine transform 만으로 만화적 모션 캡처를 시도하여 한계성을 노출하였다. 또한 3차원 메쉬(mesh) 만으로 애니메이션을 시도한 연구[6]는 애니메이션의 기준으로서 2차원 비디오를 사용하였다.

## 3. 만화적 모션 제작

본 논문에서는 만화적 애니메이션을 생성하기 위해, 직접 2차원 비디오 영상으로부터 사용자의 입력을 통해 캐릭터의 움직임을 측정된 뒤 비디오의 2차원 캐릭터의 동작을 3차원 캐릭터에 적용하여, 기존의 저작도구를 통해 생성하기 어려운 비사실적 애니메이션 동작을 생성한다. 이는 그림 1과 같이 요약될 수 있다.

예제 만화 영상은 셀 애니메이션 영상을 의미하며, 이 영상 위에서 미리 준비된 3차원 캐릭터를 오버랩하여 3차원 캐릭터의 만화적 동작을 생성 혹은 편집한다. 2차원 동작 편집이 3차원 동작 편집으로 적용되는 과정에서 자세 데이터베이스(Posture DB)가 사용되며, 이를 통해 2차원 동작이

보다 정확한 3차원 동작이 되도록 한다.

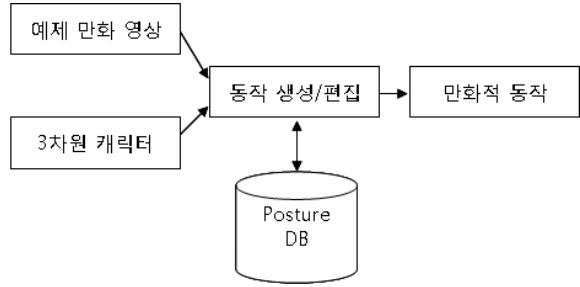


그림 1. 논문에서 제안하는 방법의 개략적인 절차

### 3.1 2차원 모션 편집

만화 동영상에 있는 2차원 캐릭터의 동작을 3차원 캐릭터의 동작으로 캡처하기 위해서는 먼저 2차원 캐릭터상의 움직임을 추출해야 한다. 이를 위해 본 논문에서는 2차원 예제 만화 영상에, 3차원 캐릭터의 본(bone)을 2차원으로 프로젝션한 구조를 이용하여 2차원 캐릭터의 자세를 만들고, 이를 3차원으로 확장하는 방법을 사용한다. 이는 2차원 영상의 캐릭터에 대한 키 프레임을 3차원 캐릭터 애니메이션의 키 프레임으로 확장하는 것으로, 2차원 영상에서 캐릭터의 자세와 3차원 캐릭터 애니메이션이 동기화되면, 모션 캡처처럼 3차원 캐릭터 관절의 변환 값(transform)을 아는 키 프레임이 생성된다.

### 3.2 3차원 모션 편집

3차원 캐릭터 애니메이션에서 캐릭터는 캐릭터 계층 구조(hierarchy)에 맞게 관절의 변환 값(transform)이 변하여 애니메이션 된다. 따라서 2차원 캐릭터의 자세를 3차원으로 확장하고, 3차원적으로 캐릭터의 관절 변환 값(transform)을 편집하는 과정에서 안정성을 보장하기 위해, 단순한 관절 변환보다는 역운동학(IK, inverse kinematics)에 근거하여 처리해야 한다. 본 논문에서는 수치적 역운동학(Numerical IK)과 예제방식의 역운동학(Example based IK)를 이용하여 모션을 생성/편집한다.

수치적 역운동학은 일반적인 3차원 애니메이션에서 캐릭터의 움직임을 제어하기 위하여 사용하는 방법으로, 3차원 캐릭터의 관절을 이동하여 직관적인 자세를 생성하도록 지원하는 보조적인 역할을 수행하며, 이를 위해 기하학적으로 움직임에 제한을 갖는 관절을 설정하고, 이를 만족시키면서 사용자의 조정에 따르는 자세를 생성하는 수치적 최적화 방법을 사용하였다. 이 최적화 문제의 에너지 함수는 다음과 같이 정의될 수 있다.

$$E(\mathbf{p}, \mathbf{q}_i) = w_p \|\mathbf{p} - \mathbf{p}_0\|^2 + \sum_i w_i \|\log(\mathbf{q}_{i,0}^{-1} \mathbf{q}_i)\|^2, \quad 0 < i < N$$

$p_0$  와  $q_0$  는 각각 사용자 입력 이전의 3차원 캐릭터의 각 관절별 위치와 회전 사원수를 나타내고,  $N$  은 캐릭터를 구성하는 전체 관절의 개수를 나타내며,  $w_p, w_i$  는 각각 관절의 위치와 회전에 대한 가중치를 의미한다. 따라서 위의 에너지 함수는 최대한 초기 입력 자세를 유지하기 위해 사용된다.

$$\arg \min_{\mathbf{p}, \mathbf{q}_i} E(\mathbf{p}, \mathbf{q}_i)$$

$$\text{subject to } f_k(\mathbf{p}, \mathbf{q}_i) = c_k$$

$f_k$  는 각 관절들에 주어진 기하학적 제한들을 나타낸다. 본 논문에서는 이러한 기하학적 제한을 제약 함수로 설정하고, 에너지 함수와 제약 함수의 합이 최소화 되도록 하는 켈레 기울기(Conjugate gradient) 최적화 방법을 사용하여 해결한다. 수치적 최적화 기법을 통해 사용자가 지정한 제한을 만족하는 형태의 수치적 역운동학 해법이 적용된 3차원 캐릭터 자세를 생성할 수 있으며, 이는 사용자와 시스템의 상호작용을 증가시키고 예제기반의 역운동학 해법을 위한 참고 자세의 생성을 위해 사용될 수 있다.

사용자가 지정한 제한이란 3차원 캐릭터의 자세를 사용자의 의도에 맞도록 수정하기 위해, 캐릭터의 각 관절에 기하학적인 제한을 의미한다. 이러한 기하학적 제한은 현재 사용자가 선택하여 이동시키고 있는 관절과, 사용자가 움직이지 못하도록 미리 설정해 놓은 관절이 해당 될 수 있다. 이러한 기하학적 제한과 함께 최적화 문제를 해결하기 위해, 기하학적 제한을 포함하여 에너지 함수를 최소화 하는 수치적 최적화 문제를 해결하는 것으로 역운동학 해법에 의한 캐릭터 자세를 생성할 수 있다.

수치적 역운동학 해법에 의해 생성된 캐릭터 자세는 캐릭터가 취할 수 있는 자세인지의 여부와 관계없이 사용자의 입력과 기하학적 제한에 따라 캐릭터의 자세를 설정하기 때문에, 기계적인 목적의 다관절체가 아닌, 그 형상이 중요한 의미를 갖는 캐릭터 애니메이션을 위한 다관절체의 자세 생성에 바로 적용되기 어려운 부자연스러운 자세를 생성해 낼 가능성이 있다. 따라서 수치적 역운동학 해법에 따라 캐릭터의 자세를 생성하는 것과 달리, 이미 존재하는 예제 자세들을 기반으로, 캐릭터가 취할 수 있는 자세를 유추하여 3차원 애니메이션의 캐릭터 표현에 적합한 자연스러운 자세를 만들어 내는 예제 기반 역운동학 기법에 대한 연구가 활발히 이루어지고 있다[7]. 예제 기반 역운동학 기법의 연구는 대부분 적은 수의 연속적인 자세에 나타나는 예제 자세를 기반으로 기계 학습 단계를 거쳐 저차원의 공간에서 빠르게 예제 자세를 찾는 방향으로 이루어지고 있다.

본 논문에서는, 숙련된 애니메이션 제작자에 의해 만들어진 불연속적인 자세 데이터베이스를 구성하고, 사용자가 원

하는 자세를 얻기 위해 수치적 역운동학 해법의 도움을 받아 대략적인 자세를 구성한 뒤, 자세 데이터베이스에서 유사한 자세를 찾아 적용하는 방식의 접근을 시도하였다. 이를 위해 자세 데이터 베이스상의 각 자세들은 최상위 골격이 동일한 위치와 방위를 갖도록 구성하였다.

사용자가 수치적 역운동학 해법과 상호작용에 의해 생성한 참고 자세와 자세 데이터베이스 내의 자세간 유사성은 다음 식과 같이, 참고 자세와 자세 데이터베이스 내의 각 관절의 기하학적 위치의 차이를 통해 에너지 함수를 구성하고, 이 에너지 함수의 결과가 최소인 자세 데이터베이스 내의 자세를 선택하는 것으로 해결할 수 있다.

$$E(\mathbf{p}_j) = \sum_i w_i \|\mathbf{p}_{j,i} - \mathbf{p}_{0,i}\|^2, \quad 0 \leq i < N, \quad 1 \leq j < M$$

$p_0$  은 사용자에 의해 생성된 참고 자세를 나타내며,  $p_j$  는 자세 데이터베이스 내의 각 자세들을 나타낸다.  $N$  은 캐릭터를 구성하는 관절의 개수,  $M$  은 자세 데이터베이스에 포함된 자세의 개수를 나타낸다. 현재 사용자가 구성한 참고 자세와 가장 비슷한 데이터베이스 내의 자세는  $E(p_j)$ 가 가장 작은 자세를 선택하는 것으로 해결된다.

### 3.3 키 프레임 애니메이션

본 논문에서는 2차원 동영상에서 결정된 키 프레임에서, 설정된 캐릭터의 자세의 키(key)를 3차원 캐릭터 애니메이션 키로 확장하여 키 프레임을 설정한다. 따라서 3차원 캐릭터의 각 키 프레임 사이의 애니메이션을 처리해야 한다. 이는 애니메이션에서 전통적으로 사용되는 키 사이의 인비트윈(in-between)을 사용하여 전체 애니메이션을 생성한다. 두 개의 키 사이의 인비트윈은 3차원 캐릭터의 각 관절의 값들을 사원수(quaternion)로 보간(interpolation)하였다.

$$\mathbf{k}_j = \{t, \mathbf{p}_0, \mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_N\}, \quad 1 \leq j < M$$

키 프레임은 캐릭터 계층 구조상 최상위의 중심 관절의 위치와 방위  $p_0, q_0$ 와 다른 모든 관절의 회전을 나타내는 회전 사원수, 키 프레임의 시간적 위치를 나타내는  $t$ 로 표현될 수 있다. 여기서  $M$  은 전체 키 프레임의 개수를 의미한다. 위 식과 같이 키 프레임이 구성되면, 키 프레임 사이의 자세는 각 관절의 회전 사원수와 중심 관절의 위치벡터, 방위를 나타내는 회전사원수를 보간하여 그 자세와 위치를 결정할 수 있다.

본 논문에서 키 프레임 애니메이션은 기본적으로 자세에서 자세 (pose to pose)로 이루어지는 키 프레임을 사용한다. 이는 2차원 애니메이션 영성으로부터 이와 비슷한 3차원 캐릭터의 자세를 생성하고, 이를 통해 3차원 애니메이션을 생성하고자 하는 것이 목적이기 때문이다. 그러나 자세에서

자세로 이루어지는 키 프레임 애니메이션은 제작과정이 직관적이기 때문에 참고한 자세로부터 쉽게 애니메이션을 생성할 수 있다는 장점이 있지만, 최종적으로 생성되는 애니메이션의 결과가 부자연스러운 형태로 나타날 가능성이 있다. 이는 캐릭터의 움직임이 모든 관절에서 정확하게 같은 시간에 키 값을 갖는 형태로 나타나지 않기 때문이다. 따라서 자세에서 자세로 이루어지는 키 프레임 애니메이션 뿐만 아니라, 각 관절에 대한 회전이나 위치에 대하여 별도로 키 프레임 지정할 수 있어야 한다. 따라서 자세에서 자세로 이루어지는 키 프레임 애니메이션을 사용하고, 이와 별도로 캐릭터를 구성하는 모든 관절에 대해 키 프레임을 지정할 수 있도록 키 프레임의 범위를 확장하였다. 키 프레임은 모든 관절에 각각 지정될 수 있으며, 이때 지정되는 키는 관절의 특성에 따라 중심관절의 경우 위치벡터와 방위를 표현하는 회전 사원수, 키의 시간적 위치가 포함되고, 중심관절이 아닌 다른 관절의 경우 회전 사원수와 키의 시간적 위치만을 포함하는 형태로 다음과 같이 정의될 수 있다.

$$\mathbf{k}_{j,0} = \{t, \mathbf{p}, \mathbf{q}\}, \mathbf{k}_{j,i} = \{t, \mathbf{q}\}, \quad 1 \leq i < N, 0 \leq j < M_i$$

설정된 키 프레임은 사용자의 조작 또는 애니메이션 진행 상황에 따라 키 프레임이 지정되지 않은 프레임을 표시하게 될 경우, 해당 프레임의 주변 키 프레임을 보간하여 자세를 생성한다. 이때 회전 사원수는 구면선형보간(SLERP) 기법을 통해 보간하고, 위치 벡터는 선형보간을 통해 보간된 값을 얻는다.

#### 4. 만화적 모션 제작 시스템

본 논문에서 제안하는 시스템은 그림 2에서 나타나 있듯이, 2차원 뷰(view), 3차원 뷰(view), 그리고 자세 데이터베이스 그리고, 키를 표시하는 타임라인 등 UI로 구성된다.

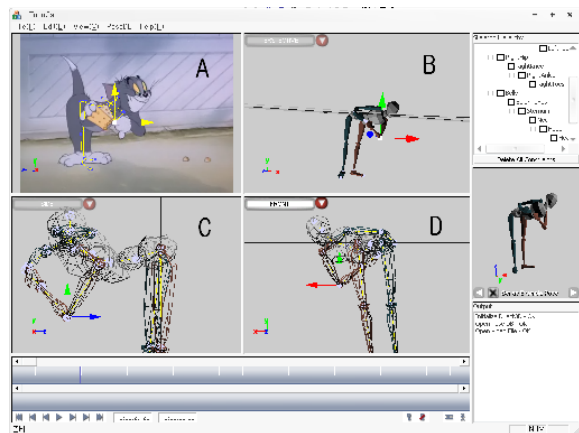


그림 2. 시스템 스크린샷

2차원 뷰는 그림 2에서 A영역이며, 여기에는 동영상 표시되고, 3차원 캐릭터의 본(bone)이 2차원으로 프로젝션

되어, 동영상에 맞추어 2차원적으로 캐릭터의 자세를 얻을 수 있게 되어있다. 3차원 뷰는 그림 2에서 B, C, D의 영역이며, 여기서는 perspective, top/botton, left/right, front/back 뷰를 선택 가능 하도록 구성되어, 3차원 뷰 상에서 2차원 캐릭터의 자세와 편집결과가 3차원에서 어떻게 반영되는지를 확인하고, 수정 가능하도록 되어 있다.

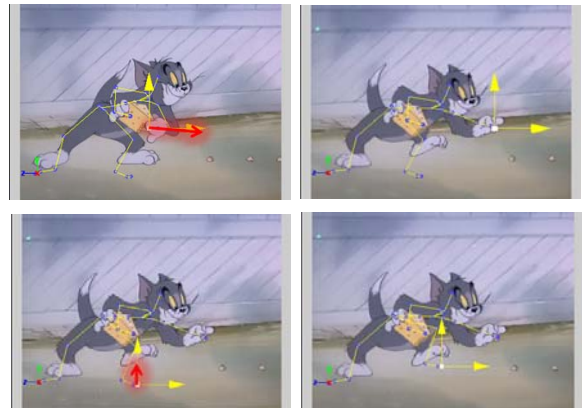


그림 3. 2차원 뷰에서 캐릭터 관절 제어로 자세 형성

그림 3은 2차원 뷰에서 캐릭터 관절 편집으로 2차원 자세를 만드는 과정을 보여주며, 만들어진 2차원 자세는 3차원 뷰의 캐릭터의 자세와 동기화 되어 있다.

3차원 캐릭터의 자세는 2차원에서 편집된 자세를 기반으로 하므로 3차원에서 수치적 역운동학으로 편집할 수도 있지만, 사용자의 오류를 최소화하기 위해 예제 기반 역운동학이 사용된다. 이때 자세 데이터베이스에 보관된 여러 자세 중 가장 근접한 자세를 검색하여 사용한다. 그림 4는 그림 2의 2차원 뷰에 있는 자세를 자세 데이터베이스에서 검색한 결과이다.

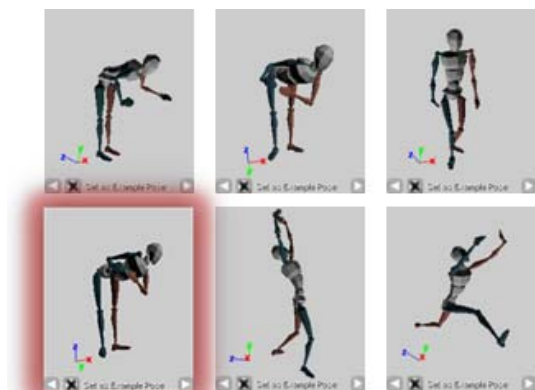


그림 4. 자세 데이터베이스의 자세 및 선택된 자세

이외에 예제 만화 동영상상의 프레임별 재생이 가능하고, 3차원으로 캡처된 모션에서 키 프레임별 처리 및 재생이 가능한 타임라인이 있고, 관절 선택을 도와주는 캐릭터 계층구조 트리(tree) 뷰 등의 UI가 있어 보다 작업의 효율을 높게 한다.

## 5. 실험 및 결과

본 논문에서 제안한 시스템을 이용하여 그림 5와 같은 만화적 영상을 3차원 모션으로 생성하였다.



그림 5. 원영상의 스틸 컷

그림 6은 생성된 3차원 캐릭터 모션을 Microsoft DirectX viewer에서 본 모습이며, 그림 7은 생성된 캐릭터 모션을 Autodesk Maya에서 캐릭터에 적용하여, 렌더링한 영상이다.

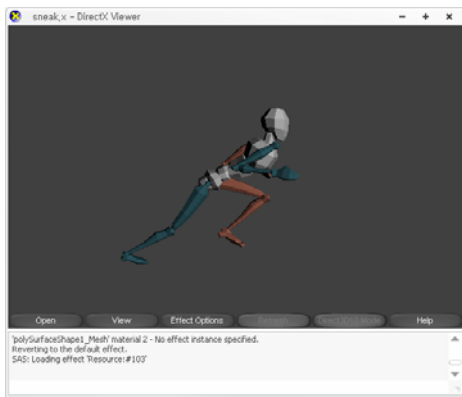


그림 6. Microsoft DirectX viewer에서 본 생성된 모션

생성된 모션은 직접 만화를 기준으로 작성되어, 종래의 모션 변형을 통해 얻은 만화적 모션에 비해 월등히 만화적 이므로, 양질의 만화적 모션 생성이 가능하다.

실험에 사용된 동영상은 약 7초로서, 200 프레임 내외의 편집 분량이 된다. 키프레임 애니메이션 제작 경험이 거의 없는 초보자가 제안하는 시스템을 이용하여 3차원 모션을 약 1시간 내외의 작업으로 생성 가능한 것으로 보아, 본 논문에서 제안하는 방법은 만화적 모션 생성에 있어 양질의 모션을 빠른 시간에 제작할 수 있는 방법으로 판단된다.

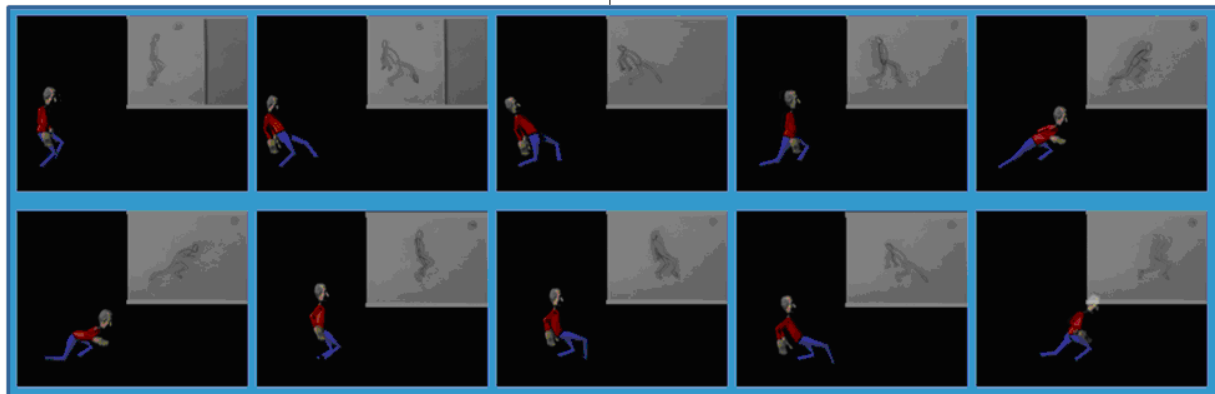


그림 7. Autodesk Maya에서 캐릭터에 적용한 만화적 모션

## 6. 결론

본 논문에서는 캐릭터의 만화적 움직임을 직접 애니메이션에서 생성하는 시스템을 제안하였다. 본 시스템을 이용하여 사용자는 빠르게 만화적 움직임을 생성할 수 있었으며, 생성된 3차원 캐릭터의 움직임은 2차원 캐릭터의 자세와 움직임을 기준으로 생성하므로 만화적 모호성을 극복하였다.

본 논문에서의 시스템은 동영상에서 키 프레임을 사용자가 선택하고 그 키 프레임에서 캐릭터의 자세를 보고 3차원 캐릭터의 자세를 설정하여 3차원 캐릭터 애니메이션의 키 프레임으로 설정하는 구조로 되어있다. 따라서 향후 동영상에서 키 프레임을 자동으로 추출하고, 동영상에서의 캐릭터 움직임을 영상 추적하면 보다 빠르게 캐릭터의 만화적 동작을 생성할 수 있을 것이다.

## 참고문헌

- [1] 김성예 외 4인, "비사실적 렌더링 기술 동향", 전자통신동향분석 제20권 제4호, 2005년 8월, ETRI, pp. 43~57, 2005.
- [2] C. Bregler, "Learning and Recognizing Human Dynamics in Video Sequences", proc. CVPR97, pp. 568, 1997.
- [3] M. J. Park et al., "Video-Guide Motion Synthesis Using Example Motions", ACM Transactions on Graphics, 25(4) pp. 1327~1359, 2006.
- [4] Jue Wang et al., "The Cartoon Animation Filter", proc. SIGGRAPH06, pp. 1169~1173, 2006.
- [5] C. Bregler et al., "Turning to the Masters: Motion Capturing Cartoons", proc. SIGGRAPH02, pp. 399~407, 2002.
- [6] Kun Zhou, et al., "Large Mesh Deformation Using the Volumetric Graph Laplacian", Proc. Siggraph05, pp. 496~503, 2005.
- [7] K. Grochow et al., "Style-Based Inverse Kinematics", proc. Siggraph04, pp. 522~531, 2004