

신체 가중치를 이용한 동일 감정 표현의 몸동작 변형*

이창숙*, 김대성**, 엄기현***, 조경은***

동국대학교 대학원 컴퓨터공학과*, 동국대학교 영상대학원 멀티미디어학과**

동국대학교 영상미디어대학 멀티미디어공학과***

shireea@dongguk.edu, irrev1234@nate.com, {khum, cke}@dongguk.edu

A Study of Use of Body Motions and Body-weighted Values for Motion Display in Virtual Characters

Changsook Lee^{○*}, Daxing Jin^{**}, Kyhyun Um^{***}, Kyungeun Cho^{***}

Dept. of Computer Science and Engineering, Graduate School^{*}

Dept. of Multimedia, Graduate School of Digital Image and Contents^{**}

Dept. of Multimedia Engineering^{***}, Dongguk University

요 약

게임에서 흔히 사용하는 진신행 가상 캐릭터를 이용한 감정 표현은 주로 몸동작을 이용한다. 이때 가상 캐릭터가 표현하는 모든 감정에 대하여 각기 다른 애니메이션을 만들어 주어야 한다. 이런 이유로 인간과 같이 감정의 강도에 따라 다른 크기의 동작을 표현하려면, 만들어야 할 애니메이션의 수가 기하급수적으로 늘어나게 된다. 이러한 문제를 해결하기 위해 본 연구에서는 가상 캐릭터가 감정 강도에 따라 다른 크기의 동작을 행할 수 있게 하는 방법을 제안한다. 본 연구에서는 감정 강도별로 캐릭터가 동작을 형성하기 위해 회전시키는 뼈대에 가중치를 주어 소극적인 표현 또는 과장된 표현이 가능하도록 하였다. 제안하는 방법을 검증하기 위해 실제 가상 캐릭터에 신체 가중치를 적용할 수 있는 EATool(Emotional Animation Tool)을 구현하였다. 구현된 환경을 통해 걷기 동작에 서로 다른 감정을 부여한 후, 각 감정의 강도에 따라 신체 가중치를 적용하였다. 실험 비교 결과 감정 강도에 따라 동일한 걷기 동작의 형태가 달라지는 것을 확인할 수 있었다.

ABSTRACT

Body motions are commonly used to express emotions in virtual characters based on body parts, which are frequently employed in games. For this purpose, it is necessary to create different types of animations corresponding to the emotions shown by virtual characters. Therefore, a large of number of animations should be created for different gestures depending on the level of human emotion. In this paper, we propose a method for displaying gestures with various degrees of complexity on the basis of the level of emotion in virtual characters. In particular, this method can be used to display passive and exaggerated expressions by adding weighted values to the frames that rotate the characters to make them show different gestures depending on the level of emotion. To verify the effectiveness of the proposed method, we use the Emotional Animation Tool (EATool), with which body-weighted values can be applied to the actual or virtual characters. After assigning different emotions to walking motions in the newly developed environment, we apply different body-weighted values depending on the level of each emotion. The results of a comparative test reveal that a given type of walking motion differs with the level of emotion.

Keywords : Artificial Emotion, Emotional Character, Body Movement, Emotion Expression

접수일자 : 2010년 11월 03일, 심사완료 : 2010년 11월 30일. 교신저자(Corresponding Author) : 조경은

※ 본 논문은 한국연구재단 기초연구사업의 일반연구자 지원사업 일환으로 수행하였음(2009-0077594, 서사구조의 자동 분석 기법을 통한 캐릭터 감성표현 모델 연구)

※ 이 논문은 2010년 한국게임학회 추계학술발표대회에서 우수논문으로 선정된 논문입니다.

1. 서론

인간은 여러 가지 방법으로 타인에게 자신의 내적 상태와 성격을 나타내는 감정을 전달한다. 인간이 타인에게 감정을 전달할 때는 주로 음성과 표정을 통해 전달한다. 그러나 음성을 이용하지 않고 표정만으로도 자신의 상태를 어느 정도 전달할 수 있다. 또한 부가적으로 몸 전체를 이용하여 감정을 전달할 수 있다. 예를 들어 피곤한 상태이거나 화가 난 상태일 경우 몸의 경직도나 어깨나 척추의 상태, 자세 등으로 자신의 상태를 알릴 수 있다.

게임내 가상 캐릭터의 감정 전달은 주로 몸동작을 통해 표현한다. 게임 속 가상캐릭터는 전신형 캐릭터의 경우가 많기 때문에, 표정 표현은 몸동작에 비해 상대적으로 적다. 과거에 비해 많은 수의 몸동작으로 의사 전달을 하며, 그 중 감정을 표현하는 몸동작도 대다수 포함되어 있다. 그러나 캐릭터의 성향이나 특징에 따른 동작이 아닌, 일률적으로 정해진 동작만을 행한다.

예를 들어 ‘울기’의 동작을 행할 때, 게임 속 캐릭터는 미리 정해진 ‘울기’의 동작을 일률적으로 행한다. 그러나 인간의 경우 자신의 성향과 감정의 강도에 따라 조금씩 다른 형태의 ‘울기’ 동작이 가능하다. 또한 가장 기본적인 동작인 ‘걷기’의 경우 현재 감정의 정도, 성격, 환경 등 복합적인 요인을 이용하여 다양한 형태의 ‘걷기’ 동작이 가능하다.

가상 캐릭터를 좀 더 인간답게 표현하기 위해서는 동일한 감정 정도일지라도 캐릭터의 성향에 따라 다른 동작 표현이 가능해야 한다. 그러나 감정의 강도에 따른 동작을 일일이 애니메이션으로 정의하는 것은 사실상 불가능에 가깝다. ‘슬픔’을 표현하는 대표적인 감정 동작인 ‘울기’ 동작 하나만으로도 애니메이터에 따라 매우 다양한 형태의 동작을 만들어 낼 수 있기 때문이다.

반면 하나의 감정 동작을 토대로 과장된 표현이나 소극적 표현을 하는 것은 가능하다. 가상 캐릭터의 동작은 뼈대의 회전으로 만들어진다. 움직이는 뼈대에 가중치를 더하여 좀 더 큰 회전이거나 작

은 회전을 만들어 낼 수 있다. 회전이 큰 경우 상대적으로 과장된 몸동작이 생성되며, 작은 경우 소극적인 몸동작 표현이 가능하다. 이러한 점을 활용하여 본 논문에서는 감정 강도에 따른 다양한 동작 생성을 위해 신체 가중치를 이용, 감정 강도에 따른 동작을 생성하는 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2장 관련 연구에서는 가상 캐릭터의 동작 생성과 감정적 동작을 생성할 때 나타나는 신체의 특징을 알아본다. 3장에서는 캐릭터의 동작 변형을 위한 신체 가중치 산출 방법과 실제 캐릭터에 가중치를 적용하여 동작을 생성하는 방법을 기술한다. 4장 실험 및 결과 부분에서는 3장의 내용을 실제 시뮬레이션하는 감정 애니메이션 툴(EATool, Emotional Animation Tool)을 통해 생성한 다양한 애니메이션 동작 결과를 기술한다. 5장 결론 및 향후 연구에서는 이 논문의 결론과 연구의 한계점, 향후 해결점에 대해 기술한다.

2. 관련연구

이 장에서는 행동을 통해 가상 캐릭터의 감정을 표현한 연구에 대해 알아보고, 본 연구와의 차이점을 기술한다. 또한 감정을 표현하는 행동에 따라 어떤 신체적 특징이 있는지 알아본다.

2.1 행동을 통한 가상 캐릭터의 감정 표현

행동을 통한 가상 캐릭터의 감정 표현의 연구로 첫 번째 J.Cassell의 BEAT프로젝트를 들 수 있다 [1]. 텍스트 입력만을 이용하여 인간과 비슷한 캐릭터를 움직이는 이 연구는, 대화 파라미터와 애니메이터가 기술한 제스처의 변동이 가능하다. 이 시스템에서 캐릭터는 몸동작과 표정, TTS엔진을 활용한 음성까지 조합한 형태로 감정을 표현한다. 애니메이터가 기술한 제스처의 변동 사항에 따라 감정의 강도별로 다른 애니메이션의 표현이 가능하다.

두 번째 W.Su의 Affective Story Character에 관한 연구이다[2]. 이 연구에서는 주어진 다이얼로그를 바디랭귀지를 이용하여 가상 캐릭터가 직접 연기하게 하는 것을 초점으로 두고 있다. 개개의 텍스트가 아닌 문맥 해독에 주요 초점을 두어, 100가지의 제스처를 조합하여 하나의 짧은 이야기를 연기하는 것이 가능하다.

위의 두 연구는 몸동작을 통하여 감정을 표현하는 것에 있어 본 연구와 가장 근접한 연구들이다. 그러나 J.Cassell의 연구에서 감정 강도에 따른 애니메이션은 일일이 애니메이터가 기술을 해 주어야 한다. 이 때문에 많은 수의 애니메이션을 기술하기는 어려우며, 기술되지 않은 애니메이션은 표현할 수 없다. W.Su의 연구는 제스처의 조합이라고는 하나 100가지의 한정된 제스처 내에서 조합을 하므로, 표현할 수 있는 행동 자체가 제한되어 있는 단점이 있다.

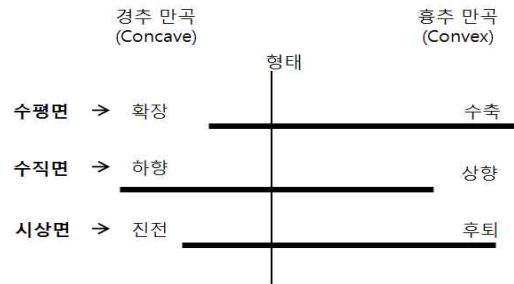
이 외에도 PEAL라이브러리를 사용하여 캐릭터의 내면 상태를 몸짓 언어로 자동화 한 Nayak의 연구[3], Laban의 LMA를 기반으로 한 Chi의 몸동작을 통한 감정 표현[4], 확률론적 노이즈 함수를 이용하여 캐릭터의 감정 표현을 한 Perlin의 연구[5] 등이 있다.

위의 기술한 연구들은 캐릭터의 성향을 반영하여 감정 강도에 영향을 주고 있지 않으며, 한정된 애니메이션 동작을 조합하여 감정을 표현하고 있다. 따라서 감정 강도에 따른 애니메이션이 없을 경우 표현 자체가 불가능하다. 또한 각각의 동작을 합성하여 표현할 경우 다수의 애니메이션 파일이 존재해야 한다.

그러나 본 논문에서는 제안하는 방법은 캐릭터 성향을 반영하여, 감정 강도에 따라 신체 가중치를 적용하여 동작을 변형한다. 동작을 변형할 때 다수의 애니메이션 파일을 사용하지 않고, 원본 애니메이션 파일 하나에서 가중치만 주어 동작의 크기를 변형한다. 이를 통해 기존 연구보다 상대적으로 적은 수의 애니메이션 파일로도 감정의 강도에 따라 다수의 동작 생성이 가능하다.

2.2 감정 표현 동작 시 신체적 특징

인간의 행동에 가깝게 가상 캐릭터의 행동을 만들려면, 인간의 신체적 특징을 알아야 한다. Lamb와 Watson의 이론에 의하면, 인간은 자세와 동작을 표현할 때 [그림 1]과 같이 운동학적 구역(Kinematics zone)이 존재한다[6].



[그림 1] Lamb와 Watson의 운동학적 구역

[그림 1]에서와 같이 자세의 모양은 경추만곡(Concave)과 흉추만곡(Convex)으로 나눌 수 있으며, 이를 구성하는 방법은 수직면, 수평면, 시상면으로 나눌 수 있다. 폐쇄적이고 우울한 상태일 때 경추만곡 모양을 취하며, 개방적이고 흥분된 상태일 때 흉추만곡의 상태를 취한다.

감정의 강도나 종류에 따라 동일한 동작의 크기가 달라진다. K.Amaya는 그의 연구에서 공을 차는 발차기 동작이 감정의 종류에 따라 달라짐을 보였다[7]. 같은 공 차기 동작이라도 우울할 때의 발 동작과 화가 났을 때의 동작, 중립적 감정일 때의 동작을 수행하였다. 비교 결과 우울할 때 동작이 상대적으로 축소된 발차기를 보였으며, 화가 났을 때 상대적으로 확장된 발차기를 보였다.

위의 내용에서 동일한 동작이라 할지라도, 감정의 상태에 따라 다른 크기의 동작이 가능함을 알 수 있다. 또한 우울한 감정일 때는 상대적으로 수축된 동작을 취하는 것을 확인할 수 있다.

본 논문에서는 Lamb와 Watson의 이론과 K.Amaya의 연구 이론을 활용하여, 감정 강도에 따라 각기 다른 형태의 동작 생성을 위한 신체 가

중치 적용 방법을 제안한다.

3. 신체 가중치를 통한 캐릭터 동작의 변형

가상 캐릭터를 이용하여 감정을 표현하려면 최소한 다음과 같은 조건이 만족되어야 한다. 첫째, 표현하고자 하는 감정이 무엇인지를 정의해야 한다. 인간이 표현할 수 있는 감정은 무수히 많다. 하나의 감정뿐만 아니라 다수의 감정을 혼합하여 표현할 수도 있다. 수많은 감정을 모두 표현하기란 사실상 불가능하다. 따라서 몇 개의 감정을 캐릭터로 표현을 할지 정의해야 한다. 둘째, 표현하고자 하는 감정의 특징적 동작을 정의해야 한다. 특징적 동작은 각 감정을 대표하는 동작으로서, 동작을 보면 어떤 감정을 표현하고 있는지 알 수 있는 동작이다. 동작은 얼굴 근육의 움직임에 이용한 표정, 신체 각 부위를 이용한 몸동작, 그리고 두 가지 모두 이용하는 경우가 있다.

보다 인간다운 감정 표현을 하려면 감정의 강도에 따라 다양한 표현이 가능해야 한다. 감정의 강도는 외부 환경과 성격 등 많은 변수가 추가되어 형성된다. 동일한 감정이라도 추가되는 변수에 따라 표현할 수 있는 감정의 세기가 달라진다.

감정의 세기를 조절함에 있어 가장 큰 역할을 하는 것은 성격과 외부 환경이다. 사람의 성향을 나타내는 성격의 경우 감정의 억제와 증폭에 관여를 하여, 동일한 상황에서의 대처 능력을 달리 보여준다. 외부 환경은 성격과 연계하거나 무관하게 감정의 세기를 조절하는 역할을 한다. 그러나 외부 환경에 따른 감정 세기는 환경에 영향을 받을 수 있는 변수가 무수히 많기 때문에 이를 다 고려할 수 없다.

본 논문에서는 정의된 감정의 대표 동작이 감정 강도에 따라 몸동작을 변화시키기 위해 다음과 같은 제약을 둔다.

첫째, 감정의 강도는 성격을 이용하여 조정한다. 성격은 다수의 특성이 모여 하나의 커다란 성격을

형성한다. 이 중에는 특정 감정의 강도에 영향을 주는 특성들도 상당수 존재한다. 그러나 외부 환경 변수보다 적은 수를 이용하여 감정의 강도를 보다 신뢰성 있게 조정할 수 있다. 따라서 감정 강도를 조정하는 변수는 성격으로 한정한다.

둘째, 감정을 표현하는 대표 동작은 해당 감정의 가장 큰 강도 값을 표현하는 동작이어야 한다. 인간은 각 신체 부위별로 절대로 움직일 수 없는 회전 각도가 존재한다. 예를 들어 팔이나 다리를 반대로 꺾는 경우, 뼈는 정상 회전 값을 벗어났으므로 부러진다. 그러나 가상 캐릭터는 이러한 한계치가 없으므로, 부러지지 않고 부자연스러운 동작으로 움직이는 것이 가능하다. 감정 강도에 따른 신체 가중치에 의해 부자연스러운 동작을 방지하기 위해 이러한 제약을 정의한다.

셋째, 감정을 표현하는데 주로 이용하는 신체 부위를 선택하여 가중치를 적용한다. 의사 전달을 위해 사용하는 제스처의 경우, 몸의 자세(앉기, 눕기, 걷기 등)와 합성되어 표현된다. 감정 전달을 위한 제스처 역시 팔, 손, 발 등의 특정 부위를 움직여 표현한다. 따라서 감정의 강도에 따라 현재 감정 표현 제스처를 형성하는 신체 부위에만 가중치를 적용한다.

감정 강도에 따른 신체 가중치의 산출 방법과 몸동작 생성을 위한 가중치 적용 방법은 각 절에서 상세히 기술한다.

3.1 감정 정도에 따른 신체 가중치 산출

감정 강도의 산출은 현재의 감정 강도에서 입력되는 감정 강도의 합으로 이루어진다. 이 과정을 식으로 나타내면 다음과 같다.

$$E_{t+1} = E_{t-1} + E_t \quad (\text{식 1})$$

(식 1)에서 E_t 는 외부로부터 입력되는 감정 강도이다. 통상 긍정적인 감정일 경우 양의 강도 값을 가지며, 부정적인 감정일 경우 음의 강도 값을 가진다. E_{t-1} 는 이미 생성된 감정 강도이며, 현재

표현되고 있는 감정의 강도를 의미한다. E_{t+1} 은 현재 표현되는 감정이 끝난 후 표현될 감정의 강도를 의미한다.

(식 1)은 현재의 감정에 다른 감정이 입력될 경우 이들 감정의 차이에 따라 다른 감정으로 전이를 하거나 증폭되는 결과를 산출한다. 그러나 성격, 환경 등과 같은 감정을 증폭 시키는 변수에 의한 부분이 없기 때문에 모든 캐릭터가 동일한 값에 의해 감정을 표현하게 된다.

이를 방지하기 위해 성격 특성을 적용하여 입력되는 감정을 조정하면, 동일한 감정이라도 성격에 따라 변화를 줄 수 있다. 입력된 감정 강도를 조정하는 방법은 (식 2)와 같다.

$$mE_{t+1} = E_t + E_t(\alpha + \beta + \gamma \dots) \quad (\text{식 2})$$

mE_{t+1} 은 현재 입력된 감정인 E_t 를 성격 특성을 적용하여 조정한 값이다. α, β, γ 는 해당 감정의 증폭에 영향을 주는 성격 특성으로서, 특성 개수에 따라 다르다[8].

(식 2)의 값을 적용하여 감정을 산출할 경우 (식 3)과 같이 정리할 수 있다.

$$E_{t+1} = E_{t-1} + mE_{t+1} \quad (\text{식 3})$$

감정에 영향을 주는 성격 특성과 해당 특성 값은 사람에 따라 모두 다르다. 그렇기에 입력되는 감정에 성격 특성을 적용하여 조정한 후 이 값을 (식 1)에 적용을 시킨다. 이렇게 하면 성격 특성 별로 동일한 감정이라 하더라도 시간 t 별로 다른 감정 강도 패턴을 보여준다[8].

산출되는 감정값을 이용하여 신체 가중치를 구한다. (식 4)는 (식 3)에서 산출되는 감정값을 이용하여 신체 가중치를 구하는 식이다.

$$T = \frac{E_{t+1}}{100} \quad (\text{식 4})$$

(식 3)에서 산출되는 감정값은 1~100 사이의 값을 가지게 된다. 이 값을 그대로 신체 가중치에 적용할 경우 값의 크기로 인하여 제대로 된 애니메이션을 만들 수 없다. 따라서 (식 4)에서와 같이 산출된 감정을 백분율로 변환하여 적용한다.

3.2 신체 가중치를 적용한 몸동작 생성

이 절에서는 가상 캐릭터에게 3.1절에서 산출한 신체 가중치를 적용하여 몸동작을 생성하는 방법에 대해 기술한다.

3.2.1 캐릭터 애니메이션 데이터의 구조

게임이나 가상현실에서 사용하는 애니메이션이 포함된 가상캐릭터는 두 가지 데이터를 포함하고 있다. 하나는 가상 캐릭터의 외형을 나타내는 모델링 데이터이고, 다른 하나는 애니메이션 데이터이다. 모델링 데이터에는 외형을 형성하는 폴리곤과 뼈대로 이루어져 있다. 이 뼈대를 이용하여 캐릭터를 움직이게 된다.

캐릭터의 움직임을 저장하고 있는 애니메이션 데이터는 계층 구조로 되어 있다. 게임에서 주로 사용하는 DirectX의 경우, 3D 데이터를 위해 X-file이라는 포맷을 제공한다. 애니메이션 데이터는 AnimationSet, Animation, AnimationKey의 세 가지 구조로 이루어져 있다[10].

AnimationSet은 걷기, 뛰기, 앉기 등과 같은 하나의 동작을 표시한다. Animation은 AnimationSet의 동작을 위한 하나의 뼈 동작을 표시한다. 예를 들어 걷기의 경우 팔과 다리 동작에 관한 애니메이션 두 가지가 속성으로 정의된다.

AnimationKey는 애니메이션 데이터의 가장 상위 구조이다. Animation을 구성하는 뼈 동작에 대한 상세 속성값을 가지는 부분이다. 움직이는 각 뼈에 대해 한 프레임당 움직이는 시간, 움직인 각도, 움직임 값에 대응되는 애니메이션 계층 구조의 상대적 지역 변환 데이터를 속성으로 한다.

움직이는 뼈대를 추출하여, 해당 뼈대에 가중치를 주어 좀 더 크거나 작은 회전을 하기 위해 애니메이션을 구성하는 하위 단위인 AnimationKey를 분석하여, 이 부분에 신체 가중치를 적용한다.

3.2.2 AnimationKey를 이용한 신체 가중치 적용 방법

3.2.1절에서 기술하였듯이, 움직이는 뼈대에 가중치를 적용하려면 AnimationKey를 분석해야 한다. 각 동작이 어떻게 저장되는 지를 분석하여, 이 부분에 가중치를 적용하면 된다. AnimationKey는 각 뼈의 동작을 두 가지 방식으로 저장할 수 있다.

첫 번째 변환 매트릭스를 사용하여 저장이 가능하다. 16개의 실수로 된 하나의 변환 매트릭스로서 각 뼈의 회전값과 움직임값을 저장한다.

두 번째 STR(Scaling, Translate, Rotation)을 이용한 저장이다. 총 10개의 실수로 Scaling(3개), Translate(3개), Rotation(4개)의 변환을 표시한다. 본 연구는 움직이는 각 뼈대에 가중치를 주어, 좀 더 큰 동작이나 작은 동작을 만드는 것이다. 따라서 STR을 이용하여 가중치를 적용한다.

STR에서 동작을 변환하기 위해 필요한 값은 Rotation값이다. Scaling과 Translate값을 변경하게 되면, 캐릭터의 계층 구조가 변화하게 되므로, 애니메이션 구조 전체가 바뀌게 된다. 또한 뼈대의 회전에 관여하는 것은 Rotation값이므로, Scaling과 Translate값은 굳이 사용하지 않아도 된다.

Rotation값은 4개의 값으로 이루어진 쿼터니온(Quaternion)으로 이루어져 있다.

쿼터니온은 캐릭터가 취하고 있는 뼈대 움직임의 상대적 회전값이다. 쿼터니온 값이 상대적이라는 이유는 쿼터니온 값에 0을 대입해 보면 알 수 있다.



[그림 2] 쿼터니온 값을 0으로 대입했을 경우

[그림 2]는 쿼터니온 값에 0을 대입한 그림이다. 그림에서와 같이 캐릭터가 취하는 자세와 다른 형태임을 알 수 있다. 따라서 캐릭터의 기본 자세를 형성할 때의 회전값이 상대적인 기본 회전 값으로 쿼터니온에 적용되어 있음을 알 수 있다. 즉, 쿼터니온 값에 가중치를 곱하여 새로운 쿼터니온을 산출하여 동작을 변화할 수 있다. 그러나 직접적으로 쿼터니온 값에 가중치를 곱하면 원하는 동작의 값이 나오지 않는다. 그 이유는 [그림 3]에서와 같이 쿼터니온의 구성과 계산 방식 때문이다[11].

$$\begin{aligned} \text{angle} &= 2 * \text{acos}(qw) \\ x &= qx / \text{sqrt}(1-qw*qw) \\ y &= qy / \text{sqrt}(1-qw*qw) \\ z &= qz / \text{sqrt}(1-qw*qw) \end{aligned}$$

[그림 3] 쿼터니온 계산식

[그림 3]에서와 같이 쿼터니온은 (x,y,z)으로 이루어진 회전축과 각도로 표시되는 회전각으로 이루어져 있다. 3D공간에서의 회전은 회전각 뿐만 아니라 x,y,z의 방향값을 가지고 회전한다. 따라서 쿼터니온 값에 무조건 가중치만 곱하게 되면, 회전각만 변화하는 것이 아니라 회전 방향도 변화하게 된다. 이 때문에 한 동작에서 다른 동작으로 변환할 때 잘못된 값으로 변환될 가능성이 있다

이것을 해결하기 위해 본 연구에서는 다음과 같은 과정을 추가한다. 첫 번째 쿼터니온을 3개의 오일러 각으로 변환한 다음 두 번째, 각각의 오일러

각에 가중치를 곱한다. 세 번째 가중치를 곱한 오일러 각을 다시 쿼터니온으로 변환하여 캐릭터에 적용한다.

그러나 이 과정을 적용하여도 위에서 언급하였듯이, 잘못된 값을 산출할 가능성이 있다. 회전 행렬의 경우, 행렬을 곱하는 축의 순서가 달라지면 산출되는 값이 달라진다. 때문에 반드시 정의한 순서대로 행렬을 곱해야 한다. 본 연구에서는 DirectX를 이용 하여, Yaw, Pitch, Roll을 기준으로 행렬 곱의 순서를 정의 한다. Yaw, Pitch, Roll은 각각 y, x, z로 대응되지만, 실제 DirectX 내에서 행렬 곱하기는 [그림 4]와 같이 z, x, y의 순서이다.

$$Z: \begin{pmatrix} CZ & SZ & 0 \\ -SZ & CZ & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad X: \begin{pmatrix} 1 & 0 & 0 \\ 0 & CX & SX \\ 0 & -SX & CX \end{pmatrix} \quad Y: \begin{pmatrix} CY & 0 & -SY \\ 0 & 1 & 0 \\ SY & 0 & CY \end{pmatrix}$$

[그림 4] DirectX에서의 행렬 곱하기 순서

[그림 4]에서 CZ, CX, CY의 C는 코사인(Cosine)을 의미하며, SX, SY, SZ의 S는 사인(sines)을 의미한다.

오일러 각을 구하기 위하여 각각의 행렬을 곱한 후, 오일러 각을 구한다. [그림 5]는 각각의 행렬을 곱한 결과를 나타내며, [그림 6]은 x, y, z축의 오일러 각을 구하는 방법을 보여준다.

$$\begin{pmatrix} CZCY+SZSXSY & SZCX & -CZSY+SZSXCY \\ -SZCY+CZSXSY & CZCX & SZSY+CZSXCY \\ CXSY & -SX & CXCX \end{pmatrix}$$

[그림 5] 행렬 곱 결과

$\frac{R12}{R22} = \tan Z$	→	$Z = \arctan2(R12, R22)$
$\frac{R23}{R33} = \tan Y$	→	$Y = \arctan2(R23, R33)$
$R32 = \sin X$	→	$X = \arcsin(R32)$

[그림 6] x, y, z축의 오일러 각 산출

[그림 6]에서 R12와 R22등은 [그림 5]의 행렬 곱 결과를 의미한다. 예를들어 R12는 [그림 5]의 행렬 1행 2열에 해당하는 값과 같다.

기준 행렬을 정의한 후 [그림 6]에서와 같이 오일러 각을 산출한 뒤 신체 가중치를 곱하여도 원하는 애니메이션을 얻을 수 없다. 예를 들어 현재 구한 오일러 각이 한 개의 축에만 존재할 경우 나머지 값은 이전 쿼터니온 값을 가지게 된다. 이때 가중치 값을 곱하게 되면, 그대로 유지해야 하는 쿼터니온의 값도 변화가 된다. 즉, 엉뚱한 방향으로 뼈대가 회전하는 결과를 가져 올 수 있다.

이를 방지하기 위하여 본 논문에서는 가중치를 곱하기 전 “보증값”을 추가하여 곱한다. 보증값이란 가중치를 곱하였을 때, 엉뚱한 방향으로 틀어진 동작을 원래 방향으로 되돌리기 위한 값이다. (식 5)는 보증값을 추가하여 신체 가중치를 적용한 것이다.

$$NewR = (1 + T)OriR + ComR \quad (\text{식 } 5)$$

(식 5)에서 *NewR*은 새로 만들어질 동작의 회전 값이다. *OriR*은 캐릭터가 원래 가지고 있는 애니메이션의 회전 값이다. *ComR*은 보증값이며, *T*는 신체 가중치이다. *ComR*값과 *T*값을 지정하면, 원래 만들어진 애니메이션에서 회전값을 변화하여, 동작의 크기를 변화할 수 있다.

4. 실험 및 결과

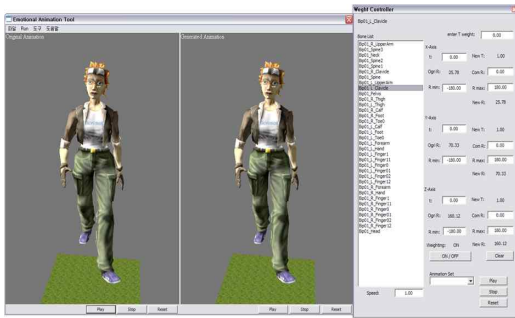
이 장에서는 본 논문에서 제안한 방법을 건기 애니메이션을 수행하는 캐릭터에 적용하여 동작을 변환하기 위한 실험을 기술한다.

4.1 실험 환경 및 실험 방법

실험을 위하여 간단한 애니메이션 툴(EATool, Emotional Animation Tool)을 구축하였다. [그림 7]은 구축된 애니메이션 툴의 전체 스크린샷이다.

실험을 위해 DirectX의 튜토리얼에서 제공하는 Tiny 캐릭터를 이용하였다[12].

실험 방법은 성격별로 동일한 감정일 경우, 똑같은 동작이 어떻게 달라지는 지를 비교한다. 실험에서 사용한 동작은 '걷기'이며, 사용한 감정은 기쁨과 슬픔이다.



[그림 7] EATool의 전체 모습

총 6명의 사람으로부터 성격 데이터를 얻었으며, 감정 입력은 기쁨 10, 슬픔 5로 정의하였다[8]. 입력된 각 감정을 성격에 따라 감정값을 산출하였으며, 이를 이용하여 신체 가중치를 적용하였다. 산출된 신체 가중치를 실제 EATool에 적용할 때는 소숫점 1자리까지 적용하였다.

4.2 실험 결과

실제 산출된 신체 가중치를 EATool에 적용하여 기쁨과 슬픔 감정에 관한 동작을 비교하였다. 비교를 위하여 [표 1]과 같이 총 6명의 실험자로 부터 기쁨과 슬픔의 감정값에 대한 신체 가중치 값을 산출하였다.

[표 1] 실험자별 입력 감정값 및 신체 가중치

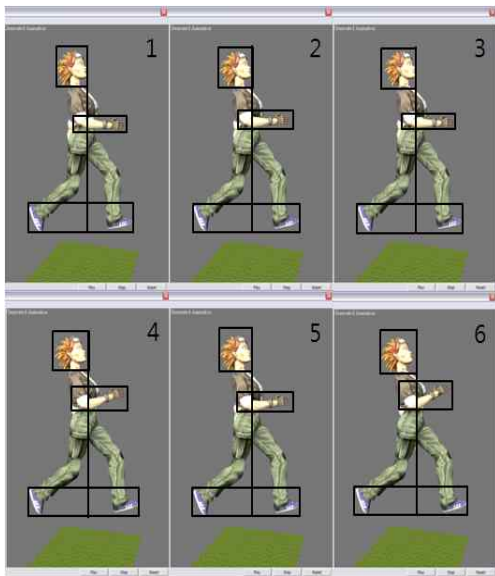
입력 감정	감정 값	실험자					
		1	2	3	4	5	6
기쁨	10	0.2	0.4	0.3	0.8	0.5	1
슬픔	5	0.1	1	0.7	0.3	0.4	0.2

[표 1]의 가중치값으로 실험을 시작하기 전에 원본 애니메이션과 비교를 위해 [그림 8]과 같이 감정을 최대(신체 가중치 : 1)로 적용하였을 때를 비교하였다.



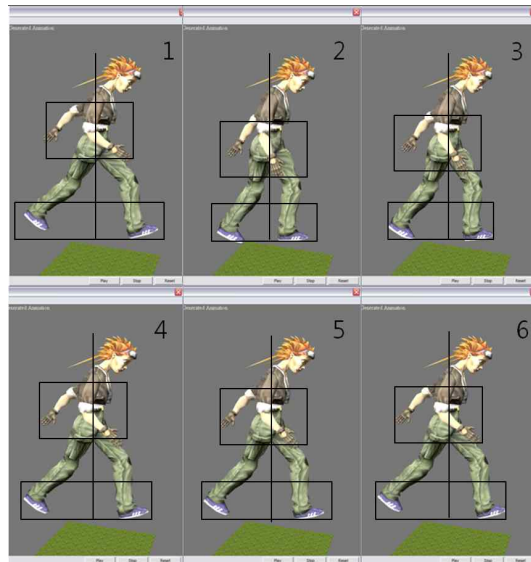
[그림 8] 원본 애니메이션과 감정 동작 비교

[그림 8] A는 원본 애니메이션의 걷기 동작을 캡처한 것이다. [그림 8] B는 기쁨 감정을 최대로 하였을 때 취하는 동작이다. [그림 8] C는 슬픔 감정이 최대일 경우 취하는 걷기 동작이다. Lamb의 이론에 따라 기쁨일 경우 가슴이 뒤로 젖혀지는 흉추 만곡 형태로 애니메이션을 생성하였다. 슬픔의 경우 목 부분을 비롯한 몸통이 앞으로 굽어지는 경추 만곡 형태로 생성하였다. 보폭은 기쁨일 경우 원본 애니메이션과 동일하게 설정하였고, 슬픔일 경우 보폭을 작게 하였다. 팔 동작의 경우, 기쁨 감정은 씩씩함을 표현하기 위해 손을 위로 향하도록 하였다. 슬픔 감정에서는 축 처진 애니메이션을 위해 손을 아래로 향하도록 하였다.



[그림 9] 기쁨 감정의 실험자별 걷기 결과

[그림 9]의 스크린 샷에서 각 그림의 번호는 실험자 번호와 동일하다. 가중치로 인해 변화된 부분의 구별을 쉽게 하기 위해 중심축을 표시한 후, 변화된 부분을 박스로 표시하였다. 표시를 해 본 결과, 감정의 강도가 클수록 손의 위치가 위로 향하는 것을 볼 수 있다. 또한 흉추 만곡형의 동작을 만들기 위해 상체에 가중치를 준 결과, 강도가 클수록 중심축에 비해 고개가 뒤로 가는 것을 볼 수 있다. 보폭 역시 미세하기는 하나 감정의 강도가 클수록 보폭 크기가 커짐을 알 수 있다.



[그림 10] 슬픔 감정의 실험자별 걷기 결과

[그림 10]은 슬픔 감정을 적용하였을 때 실험자별 결과이다. 스크린샷에서 변화된 부분의 구별을 위해 몸의 중심축과 박스로 구분하였다. 슬픔 감정의 보폭은 원본 애니메이션보다 훨씬 작게끔 변화를 주었다. 그 결과 2번 실험자와 3번 실험자와 같이 감정 차이가 별로 나지 않아도 보폭으로서 애니메이션이 달라짐을 쉽게 확인할 수 있었다.

전체적으로 분석한 결과 슬픔 감정이 기쁨 감정에 비해 육안으로 쉽게 감정의 강도를 구분할 수 있었다. 기쁨 감정의 경우 몸통과 손의 위치에만 변화를 주어 감정 표현을 하였다. 그 결과 감정 강도가 미세한 경우 차이가 확연하지 않아 육안으로 구분이 힘들다. 그러나 슬픔 감정처럼 몸통, 보폭 등 신체 전반부를 이용한 경우, 감정 강도가 미세해도 그 차이를 육안으로 구분할 수 있었다.

5. 결론 및 향후 과제

인간의 감정을 가상캐릭터로 표현할 경우, 표현되는 감정만큼 다양한 애니메이션을 만들어야 한다. 인간의 경우 같은 감정 표현의 동작이라도 강

도에 따라 다르게 표현할 수 있다. 본 논문에서는 신체 가중치를 이용하여 감정 강도에 따라 동일한 감정 표현 동작의 크기를 다르게 표현하는 방법을 제안하였다. 제안한 방법을 실제 캐릭터에게 적용하여, 각 성격별 감정 강도에 따른 걷기 동작을 비교하였으며, 그 결과 감정 강도에 따라 성격별로 다른 크기의 걷기 형태를 볼 수 있었다. 또한 특정 신체의 한 가지 부분 보다는 여러 부위를 이용하여 감정 표현을 한 동작이 미세한 감정 강도를 표현하는데 좀 더 효과적임을 알 수 있었다.

향후 보다 신뢰성 있는 결과를 위하여 실험에서 사용한 감정보다 더 많은 수의 감정 산출값에 따른 신체 가중치를 이용해야 할 것이다.

- 한국 게임학회논문지, Vol.10, No.2, 2010
- [9] <http://local.wasp.uwa.edu.au/~pbourke/dataformats/directx/>
 - [10] 김용준, 3D 게임 프로그래밍, 한빛 미디어, 2010
 - [11] L.M.Bishop, J.M.Verth, 김규열 역, 게임 & 인터랙티브 애플리케이션을 위한 수학, 지앤선, 2008
 - [12] Microsoft, DirectX SDK June 2010 Tutorial

참고문헌

- [1] J.Cassell, H.Vihjalmsson, T.Bickmore, "BEAT : The Behavior Expression Animation Toolkit", Proc.ACMSIGGRAPH '01., pp.477-486, 2001
- [2] W. Su, B. Pham, A. Wardhani, "Personality and Emotion -Based High-Level Control of Affective Story Characters", IEEE Transaction on Visualization and Computer Graphics, pp.281-293, 2007
- [3] V.Nayak and M.Turk, "Emotioanl expression in virtual agents through body language", Int'l Symposium on Visual Computing, Springer, pp.313-320, 2005
- [4] D. Chi, M.Costa, L.Zhao, N.I.Badler, "The EMOTE Model for Effort and Shape", Proc. SIGGRAPH, pp.173-182, 2001
- [5] K.Perlin, "Real Time Responsive Animation with Personality", IEEE Trans. Visualization and Computer Graphics, vol. 1, pp.5-15, 1995
- [6] W. Lamb and E. Watson, "Body Code-The Meaning Movement", Princeton Book Company, 1979
- [7] K.Amaya, "Emotion from Motion", Proceedings of the conference on Graphics interface pp. 222 - 229, 1996
- [8] 이창숙, 엄기현, 조경은, "캐릭터 성격에 따른 동일 감정 표현의 다양화를 위한 감정 조정 방안",



이 창 숙 (Changsook Lee)

2002.2 경인여자대학 멀티미디어과 전문학사
2004.2 동국대학교 전자계산학 전자계산학과 이학사
2006.3 동국대학교 영상대학원 멀티미디어학과 공학석사
2008.2 동국대학교 대학원 컴퓨터공학과 컴퓨터공학 전공
박사과정 수료
2007.9-2008.2 동국대학교 광고홍보연구소 전임연구원
2008.3-2008.5 동국대학교 게임연구소 전임연구원

관심분야 : 게임 인공지능, 감성 지능형 캐릭터



김 대 성 (Daxing Jin)

2009.9 장춘대학교 소프트웨어엔지니어학과 공학사
2010.3-현재 동국대학교 영상대학원 멀티미디어학과 석사
과정

관심분야 : 게임 프로그래밍, 게임 알고리즘



엄 기 현 (Kyhyun Um)

1975.2 서울대학교 공과대학 응용수학과
1977.2 한국과학기술원 전산학과
1994.2 서울대학교 대학원 컴퓨터공학과
1978.3-2007.6 동국대학교 컴퓨터멀티미디어 공학과 정
교수
2007.7-현재 동국대학교 영상미디어대학 게임멀티미디어
공학과 교수
1995.3-1999.2 동국대학교 정보관리처장 역임
2001.3-2003.2 동국대학교 정보산업대학 학장 역임
2009.9-현재 동국대학교 영상미디어대학 학장 겸 영상
대학원 원장
2005.3-현재 한국게임학회 자문위원
1998.12-2001.12 한국멀티미디어학회 부회장, 자문위원,
수석부회장 역임
2007.1-2007.12 한국멀티미디어학회 회장

관심분야 : 게임시스템 및 구조 설계, 멀티미디어응용
시스템, 멀티미디어데이터베이스



조 경 은 (Kyungeun Cho)

1993.2 동국대학교, 전자계산학과(공학사)
1995.2 동국대학교, 컴퓨터공학과 대학원(공학석사)
2001.8 동국대학교, 컴퓨터공학과 대학원(공학박사)
2003.9-2005.8 동국대학교 정보산업대학 컴퓨터멀티미디어
공학과 전임강사
2005.9-2009.8 동국대학교 영상미디어대학 게임멀티미디어
공학과 조교수
2009.9-현재 동국대학교 영상미디어대학 멀티미디어
공학과 부교수

관심분야 : 컴퓨터 게임 알고리즘, 게임 인공지능, 멀티
미디어 정보처리