스케치기반 인터페이스를 이용한 키 프레임 애니메이션 생성

양경용 이제희
육군사관학교 서울대학교
yangs@mrl.snu.ac.kr jehee@mrl.snu.ac.kr

Creating Keyframe Animation Using Sketch-based Interface

Kyungyong Yang Jehee Lee
Korea Military Academy Seoul National University

요 약

다양한 스타일의 인간 동작을 쉽고 빠르게 생성하는 일은 캐릭터 애니메이션에서 매우 중요한 문제이다. 실제로 이러한 애니메이션을 만드는데 있어서 가장 고품질의 결과를 얻을 수 있는 방법은, 전문 애니메이터가 3차원 애니메이션 제작 도구를 사용하여 직접 만드는 것이다. 하지만 일반 사용자가 이와 비슷한 애니메이션을 만드는 것은 어려운 일이다. 이 논문에서는 누구나 쉽게, 그리고 일정 품질 이상으로 3차원 캐릭터 애니메이션을 제작할 수 있는 시스템을 제안한다. 이를 위해서 스케치기반 인터페이스와 모션 데이터베이스를 이용한다. 스케치기반 인터페이스는 일반 사용자가 어렵게 느낄 수 있는, 키 프레임을 쉽고 빠르게 지정할 수 있도록 도와준다. 모션 데이터베이스는 이러한 키 프레임을 설정을 돕고, 애니메이션 제작에 필요한 키 프레임 수를 줄인다. 이 방법으로 누구나 원하는 애니메이션을 쉽고 빠르게 생성할 수 있다.

1. 서 론

좋은 품질의 3차원 캐릭터 애니메이션을 쉽고 빠르게 생성하는 일은 컴퓨터 그래픽스에서 매우 중요한 문제이다. 최근 컴퓨터 그래픽스 관련 기술이 급속도로 발달함에 따라, 많은 분야에서 3차원 캐릭터 애니메이션을 이용하고 있다. 지금까지 사용되어왔던 애니메이션이나 게임분야에서뿐 아니라, 방송, 교육, 광고 등의 분야에서도 해당 콘텐츠에 이를 접목시켜서 사용하고 있다.

3차원 캐릭터 애니메이션을 만드는 데에는 다양한 방법이 사용되고 있다. 그 중에서 전문 애니메이터가 직접 만드는 방법이 가장 좋은 품질의 애니메이션을 만드는 길이다. 하지만 이렇게 애니메이션을 제작하기 위해서는 전문 애니메이터들이 많은 시간을 소모하여야 한다. 그렇기 때문에 비용이 많이 들뿐 아니라, 많은 경우에 만들어진 3차원 캐릭터 애니메이션을 다른 콘텐츠에 다시 사용하기 어렵다. 게다가 전문적 기술이 없는 사람은 아무리 많은 시간을 투자해도 이러한 애니메이션을 제작하기 아주 어렵다.

일반적으로 애니메이션을 제작하기 위해서는 3차원 공간상에서 캐릭터의 관절들을 하나하나 조정해야한다. 하지만 모니터는 2차원의 출력장치이기 때문에, 시점을 여기저기 돌려가면서 작업을 수행하여야 하고, 이러한 과정을 빠르고 쉽게 하는 일은 오랜 기간 숙달되지 않고서는 불가능 하다. 실제로 2차원 화면을 보면서 그 안에 있는 3차원 물체를 직접 조정하는 일은 힘들기 때문이다. 게다가 이런 작업을 보통 키 프레임 단위로 수행하는데, 좋은 품질의 애니메이션을 만들기 위해서는 이러한 키 프레임을 많이(1초당 수개) 설정해줘야 한다. 이 정도로 제작하면 작업량 자체도 엄청나게 늘어나서 보통 사람이 제작하는데 매우 많은 시간이 소요된다..

이 논문에서는 3차원 캐릭터 애니메이션을 누구나 쉽게, 또 일정 품질 이상으로 만들 수 있는 방법을 제안한다. 이를 위해 우리는 모션 캡처 기법으로 얻어진 모션 데이터들을 사용한다. 캡처를 통해 얻어진 모션 데이터들은 사용하기 쉽게 모션 데이터베이스로 구축된다. 그리고 이를 이용하는, 이른바 데이터기반 기법으로 3차원 캐릭터 애니메이션을 만들 것이다. 여기에 스케치 인터페이스를 기반으로 하여 누구나 쉽게 자신이 의도하는

자세를 쉽게 키 프레임으로 지정할 수 있게 할 것이다. 이렇게 지정된 키 프레임들의 사이는 모션 데이터베이스를 기반으로 한 다이나믹 프로그래밍 기법을 이용하여 채워 넣는다. 실제 동작을 캡처한 모션 데이터를 이용해서 애니메이션을 만들 경우에 상당히 적은 키 프레임을 설정해 좋은 품질의 애니메이션을 생성 할 수 있다.

2. 관련 연구

모션 데이터의 활용을 위한 모션 데이터베이스에 관한 연구는 이전에 몇몇 사람들에 의해서 이루어졌다. KOVAR [1] 는 2002년에 모션그래프를 이용하여 다른 스타일의 걷는 동작들로 바닥에 그어놓은 임의의 선들을 따라가는 일이 가능하게 하였다. ARIKAN [2] 은 2002년에 여러 가지 모션 캡처 데이터로 계층구조의 그래프를 형성하여 사용자가 입력하는 제약조건을 만족하는 품질 좋은 일련의 동작들을 검색하여 얻어내었다. [3] 그리고 모션 데이터에 주석을 달아 해당하는 모션들로 이루어진 애니메이션을 다이나믹 프로그래밍을 이용하여 생성하였다. LEE [4] 는 2002년에 모션 그래프를 생성하고 거기에 빠른 검색을 위해 비슷하게 연결된 데이터들로 클러스터 그룹을 형성하여 검색에서의 효율성을 매우 늘렸다. 이는 사용자가 실시간으로 생성모션의 제약조건을 바꾸어도 실시간으로 충분히 돌아갈 수 있게 하였다.

스케치 인터페이스를 3차원 캐릭터에 적용한 몇몇 연구를 살펴보면 다음과 같다. LEE [4] 는 2002년에 사용자가 캐릭터의 이동경로를 스케치로 지정해주면 실시간으로 모션 데이터베이스를 검색하여 자연스럽게 경로를 따라갈 수 있게 하였다. DAVIS [5] 는 2003년에 사람의 관절 연결부분을 골라 자세히 스케치하여 이를 키 프레임으로 하는 3차원 애니메이션을 생성할 수 있게 하였다. 이때 2차원에서 3차원 데이터로 전환에서 부족한 정보가 생기게 되는데, 이때 사용자가 관절들의 제약조건을 기반으로 생성된 3차원에서의 자세들을 보고 원하는 자세를 골라냄으로써 해결했다. 하지만 스케치가 매우 정교하게 이루어져야 한다는 점에서 전문적인 스케치가 필요하게 된다. THORNE [6] 는 2003년에 가능한 모션들의 패턴을 단순화 하여 이들을 스케치하면 가장 비슷한 패턴을 골라서 캐릭터가 그대로 행동하는 연구를 하였다. 매우 직관적으로 다양한 동작을 만들 수 있지만 패턴들이 모두 지정되어져야 결과를 얻을 수 있다.

3. 모션 데이터베이스

이 논문에서는 적은 수의 키 프레임으로도 높은 품질의 애니메이션을 만들기 위해 모션 데이터베이스를 기반으로 애니메이션을 생성한다. 모션 캡처를 이용하여 3차원 캐릭터 애니메이션을 제작하기 위해서는 만들고자 하는 콘텐츠에 대응하는 데이터가 필요하다. 이때 비슷한 동작이 반복되어 사용되어도 따로따로 전부 캡처 하여 사용하는 경우가 많다. 모션 캡처 데이터의 재사용이 거의 이루어 지지 않는 것이다. 여기서는 효과적으로 모션 데이터를 사용하기 위해 널리 알려져 있는 모션 그래프 기법을 이용하여 모션 데이터베이스를 구성한다. 이는 뒤에서 다룰 키 프레임을 기반으로 하여 애니메이션을 생성하는데 많은 도움이 된다.

4. 스케치기반 인터페이스

스케치 인터페이스의 가장 큰 특징은 매우 직관적이라는 것이다. 일상에서 누군가에게 무엇인가를 설명할 때 우리는 펜과 종이, 분필과 칠판을 이용해서 쓰고 그려가면서 설명한다. 이는 우리가 익숙하고 쉽게 사용할 수 있다는 것을 의미한다. 실제로 3차원 도형을 표현할 때도 보통 종이에 선으로 입체감을 살려 2차원으로 표현한다. 컴퓨터 모니터에서도 결국에는 2차원의 모니터에 3차원 정보를 보여주고 있다. 즉, 우리는 주로 2차원 입출력 장치를 사용하여 생각이나 의도를 표현한다.

스케치는 사용자의 마음대로 자유롭게 그릴 수 있는 입력이다. 버튼을 누른다든지, 여러 메뉴에서 하나를 고르는 식의 선택형 입력방식에 비해서 입력의 형태가 훨씬 다양하다. 한 사람이 같은 의도로 스케치를 해도 완전히 똑같은 입력을 다시 만들기는 쉽지 않다. 이러한 특징은 인터페이스라는 입장에서 장점이 될 수 있지만 단점이 될 수도 있다. 따라서 이러한 특성을 어떻게 사용하느냐를 고민할 필요가 있다.

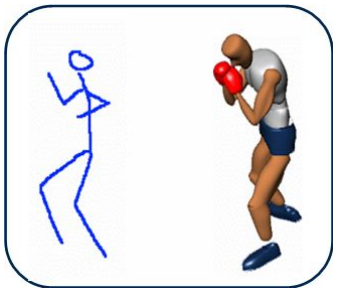
우리가 사용할 스케치는 2차원 인터페이스이지만 이를 가지고 해야 할 일은 3차원 공간 안에 있는 캐릭터의 제어이다. 하지만 3차원 정보와 2차원 정보 사이의 매핑에는 일대일 대응이 존재하지 않는다. 그나마 3차원에서 2차원으로의 매핑은 특정 평면을 기준으로 프로젝션 하면 유일한 값을 얻을 수 있지만, 부족한 정보로 인하여 2차원에서 3차원으로 매핑은 일은 매우 어려운 문제이다. 따라서 이러한 문제를 해결하기 위한 여러 방법들이 연구되고 있다.

모션 데이터베이스를 이용하는 애니메이션 기법에서 스케치 인터페이스를 사용한 가장 큰 이유는 데이터베이스 검색을 매우 쉽게 할 수 있기 때문이다. 모션 데이터베이스는 그 특성상 여러 가지 수치적인 값들의 연속으로 이루어져 있다. 따라서 이러한 모션 데이터들을 분류하는 일은 상당히 노력이 많이 들고, 자동화가 힘들다. 실제로 많은 경우에 모션 데이터베이스 하나하나에 주석을 달아서 사용하는 경우가 많다. 하지만 스케치 입력을 이용하여 캐릭터의 자세와 연관을 시켜서 생각해보면 사람이 보고 판별하는 것과 같은 시각적 측면에서의 인식효과를 얻을 수 있다. 즉, 눈으로 보기에 비슷한 동작이나 자세를 찾기에 충분히 효율적이고 의도적으로 맞는 일을 수행 할 수 있다는 것이다. 따라서 모션 데이터베이스의 종류를 몰라도 스케치 입력을 통해 여러 가지 원하는 자세 데이터를 쉽게 검색하여 사용할 수 있다.

5. 스케치를 이용한 자세 찾기

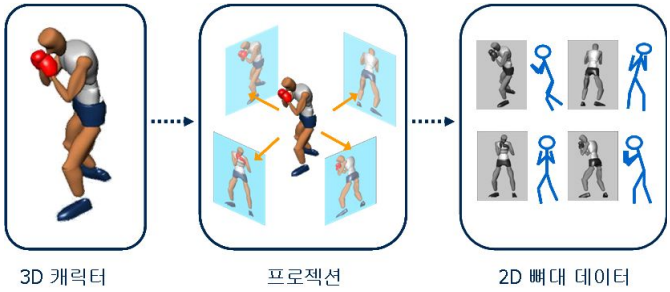
일반적으로 사람의 자세를 그리는 데에는 수많은 방법이 존재한다. 아주 단순하게 표현할 수도 있는 반면, 거의 실제의 모습과 비슷할 정도로 세밀하게 표현할 수도 있다. 하지만 후자의 경우 많은 연습을 통해 숙달된 사람만이 할 수 있다. 여기서는 이러한 방법들 중에서 가장 단순한 캐릭터 구조를 사용하였다. 이는 동작 데이터에서 찾는 자세를 처음부터 세밀하게 검색하기보다는, 의미가 통하는 자세를 큰 범주에서 찾고 나중에 사용자의 의도에 맞게 세밀한 수정을 가하고자 하는 의도이다. 또한 사용자의 스케치 표현능력이 숙달되지 않아도 충분히 의도하는 수준의 좋은 결과를 얻을 수 있을 것으로 기대된다. 실제로 3차원 애니메이션 툴의 경우, 캐릭터의 정확한 자세를 얻기 위해서는 3차원 공간상에서 캐릭터의 각 관절 부분을 이리저리 한참 움직인 후에야 원하는 자세를 얻을 수 있다. 하지만 이러한 방법은 전혀 직관적이지 않다.

기본적으로 [그림 1]과 같은 방법으로 자세를 간단히 스케치 하였다. 이는 아주 간단한 형태로 누구나가 쉽게 그릴 수 있는 방법이다.



[그림 1] 자세 스케치 입력

이렇게 찾고자 하는 자세를 스케치 했으면 이제는 스케치한 자세 입력에 해당하는 실제 3차원 캐릭터의 자세를 찾아야한다. 하지만 이 때, 한 가지 문제점이 존재한다. 우리가 다루려 하는 캐릭터는 3차원 공간상에서 존재한다. 하지만 입력으로 사용된 자세의 스케치는 2차원 평면데이터로 나타난다. 따라서 실제 캐릭터의 자세와 스케치로 입력된 자세와의 정확한 1:1매칭은 존재 할 수 없다. 그리하여 우리는 3차원 공간상의 캐릭터의 자세를 특정한 몇몇 시점의 평면으로 프로젝션 하여 얻어낸 데이터를 실제 스케치한 2차원 데이터와 비교 하였다. 이는 우리가 보통 사람의 자세를 그릴 때 임의의 각도에서 바라본 모습을 그리는 것이 아니라 매우 한정된 각도에서만 그린다는 점에서 기인하였다. 실제로 상하좌우가 아닌 다른 각도에서 사람의 모습을 스케치하는 일은 생각보다 매우 어려운 작업이 될 수 있다. 따라서 [그림 2]에서와 같이 정면, 측면, 후면에서의 프로젝션을 이용하여 스케치된 자세와 비슷한 3차원 캐릭터의 자세를 데이터베이스에서 찾도록 하였다.

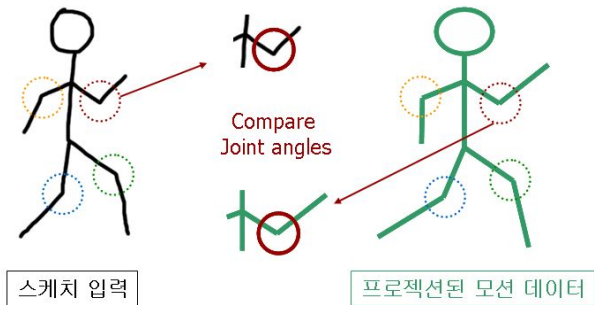


[그림 2] 3차원 모션 데이터의 프로젝션

스케치로 입력된 자세를 모션 데이터베이스에서 찾는 데에는 간단한 기하학적인 비교가 사용되었다. 스케치 입력을 벡터들의 집합으로 보고, 프로젝션 시킨 모션 데이터 역시 벡터들의 집합으로 생각하여, 내적을 이용 이들 각 관절의 각을 비교하여 그 차이들의 합을 얻어낼 수 있다.

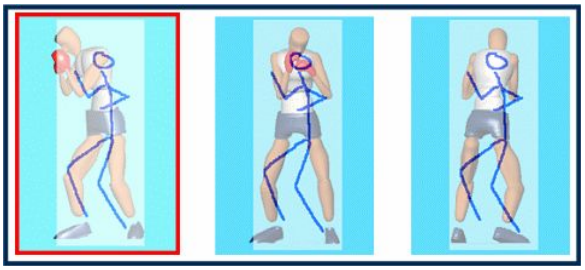
$$D = \sum_k w_k (v_k \circ u_k)$$

여기서 v_k는 스케치입력값 벡터이고, u_k는 실제 모션데이터값 벡터, w_k는 각 관절부분의 가중치이다. 이렇게 얻어진 D값을 이용하여 이 값이 일정 수치 이하인 작은 자세들을 얻는다. 이렇게 하면 반드시 우리가 스케치 입력에서 의도한 자세만을 찾을 수는 없겠지만, 우리가 스케치로 의도한 가능성이 있는 자세는 모두 찾을 수 있다.

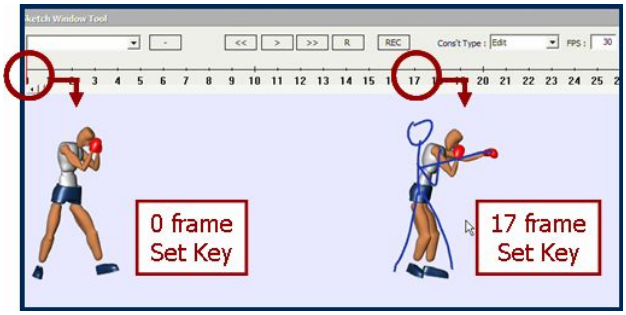


[그림 3] 각 관절 부분의 비교

따라서 이러한 방법으로 동작 데이터베이스에 있는 자세들 중에 입력한 스케치의 의도에 맞을 가능성이 있는 자세들을 모두 찾아준다. 그러면 그중에서 실제 사용자가 원하는 동작을 찾아서 이를 키 프레임으로 지정할 수 있다.



[그림 4] D값을 기준으로 찾아진 자세와 그 중에서 선택된 자세



[그림 5] 선택한 자세를 키 프레임으로 지정

6. 애니메이션 생성

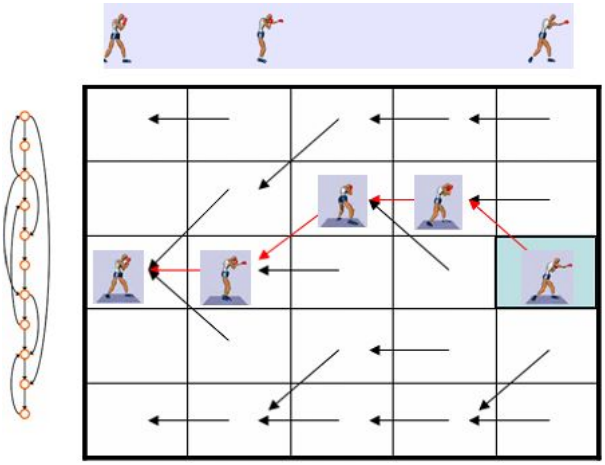
이렇게 스케치된 자세들을 키 프레임으로 설정하면, 완전한 애니메이션을 만들기 위해서 이들 사이의 프레임들을 채워야한다. 일반적인 애니메이션 제작에서는 키 프레임 부분을 변형시키거나, 비-스플라인 같은 방법을 이용하여 키 프레임들 사이를 보간하여 만든다. 그리고나서 눈에 거슬리는 부분들은 수작업으로 처리하면 애니메이션이 완성된다.

여기서는 이러한 방법을 사용하지 않고, 모션 데이터베이스를 이용하여 애니메이션을 생성한다. 앞에서 우리는 캡처된 모션 데이터로 모션 그래프를 이용하여 모션 데이터베이스를 구축하였다. 이렇게 구축된 모션 데이터베이스를 이용하면 높은 품질의 애니메이션을 빠르고 쉽게 만들 수 있다. 모션 그래프에서 프레임단위로 동작정보를 분석하여 전이 가능한 프레임들 사이를 에지로 연결해 놓았다. 그리고 그 에지에는 전이비용이 함께 저장되어 있기 때문에, 이 에지정보를 이용해서 일련의 애니메이션을 만들 수 있다. 물론 이때 이미 선택해 놓은 키 프레임 정보를 기반으로 사용할 것이다. 하지만 우리가 캡처해서 얻어놓은 데이터에 이러한 키 프레임들 사이를 원하는 타이밍에 메워 줄 수 있는 데이터가 없을 수 있다. 따라서 우리는 모션 데이터베이스를 최대한 이용하여 최선의 솔루션을 찾아야하고, 이를 위해 다이나믹 프로그래밍 기법을 사용하였다.

키 프레임 정보를 가지고 애니메이션을 만들 때 고려해야할 사항은 2가지이다. 첫째는 원하는 키 프레임에 해당하는 자세가 원하는 타이밍에 나타나야하고, 두 번째는 이들 사이가 매끄럽게 연결되어야 한다는 점이다. 이를 위한 다이나믹 프로그래밍의 과정은 다음과 같이 생각되어질 수 있다.

$$P_{i,j} = P_{i,j-1} + \min\left(T_{j-1} + KD_{j-1}\right)$$

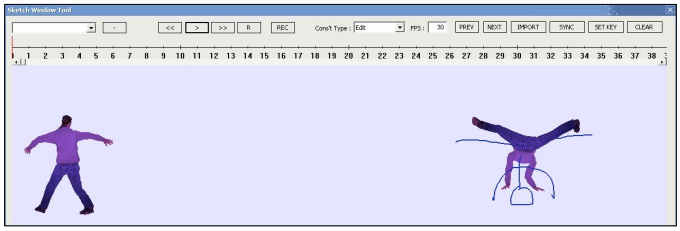
$P_{i,j}$는 i번째 프레임에서 j번째 프레임까지 생성되어지는 최적의 애니메이션 순서가 되고, T_ii 는 i번째 프레임에서 다음 프레임으로 넘어가는데 드는 전이비용, KD_i는 i번째가 키 프레임일 때 키 프레임으로 선택된 자세와 해당 프레임의 자세와의 차이이다. 이를 위해 프레임간의 차이 값 정보와 순서 정보를 테이블로 만들면 $O(n*m)$ 크기의 테이블이 2개 완성되게 된다. 이때 n은 만들고자 하는 애니메이션의 전체길이가 되고, m은 모션 데이터베이스의 크기가 된다. 최종적으로 우리가 얻으려는 프레임의 순서는 순서정보 테이블에서 $\min T(n)$에 해당하는 프레임부터 시작하여, 뒤에서부터 역 순차적으로 추적하면서 얻어낼 수 있다.



[그림 6] 다이나믹 프로그래밍 결과 생성된 테이블. 가로축은 생성하고자 하는 애니메이션의 총 프레임수, 세로축은 모션 데이터베이스의 총 프레임수이다. 검은색화살표는 각 프레임에서의 전이비용이 가장 낮은 전 프레임을 나타낸다.

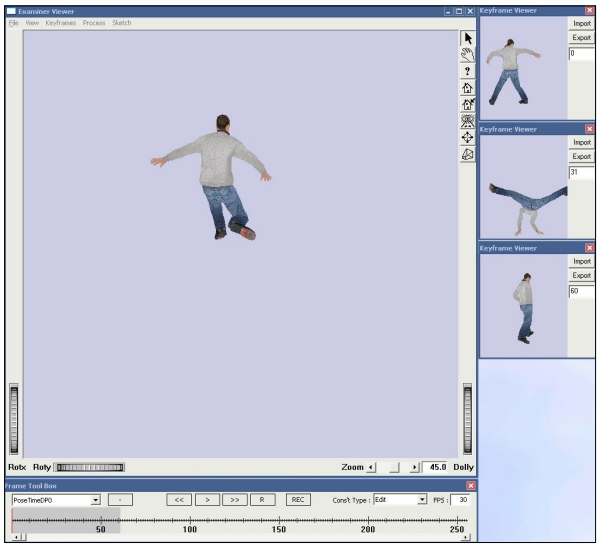
7. 실험 결과

스케치 기반 인터페이스를 이용하여 원하는 프레임에 해당 자세를 스케치하게 되면 그와 비슷한 자세를 모션 데이터베이스에서 찾아서 계산상 가장 비슷한 자세부터 보여준다. 이러한 방식으로 만들고자 하는 애니메이션의 키 프레임을 지정할 수 있다.



[그림 7] 스케치 입력 인터페이스를 통한 자세 검색

키 프레임을 전부 지정하였다면 이를 가지고 다이나믹 프로그래밍 기법을 이용하여 해당 키 프레임을 고려한 최적의 애니메이션을 생성한다.



[그림 8] 다이나믹 프로그래밍을 이용한 키 프레임 애니메이션 결과

8. 결론 및 향후 계획

3차원 캐릭터 애니메이션을 만드는데 있어서 스케치기반 인터페이스와 모션 데이터베이스를 이용하는 방법은 매우 효과적인 방법이었다. 스케치 인터페이스를 이용하여 키 프레임을 설정하는 방법은 단지 수초만의 시간으로 원하는 결과를 얻을 수 있었다. 이는 스케치에 전문적이지 않은 사람들이라도 가능한 일이다. 또 이러한 키 프레임을 설정하는데 있어서 보통 특별한 자세를 설정하는 경우가 대부분이기 때문에, 보통의 경우 이러한 자세가 등장하는 빈도 수 만큼만 키 프레임을 설정하면 실제 동작과 비슷해 보이는 품질의 애니메이션을 얻을 수 있었다. 이는 모션 데이터를 이용하지 않는 키 프레임 애니메이션이 초당 몇 개씩의 키 프레임을 필요로 하는 것과 비교해 매우 효율적이다. 결국 데이터베이스가 잘 확보되어있으면 스케치 입력을 가지고 원하는 애니메이션을 쉽게 만들 수 있다.

하지만 데이터베이스에 의존적이기 때문에, 데이터에 존재하지 않는 동작은 만들어내기 힘들다는 문제점이 있다. 키 프레임 자세를 설정할 때에도 데이터베이스에서 유사한 자세를 검색하기 때문에 원하는 동작이 존재하지 않을 수도 있다. 그리고 데이터베이스가 매우 작을 경우에는 설사 키 프레임을 찾았다 하더라도 애니메이션을 생성할 때 키 프레임을 포함하는 일련의 애니메이션이 찾아 지지 않을 수도 있다. 하지만 보통 한 번에 캡처하는 모션 데이터 정도의 양이면 등장하는 주요 자세들로 이루어진 애니메이션은 생성 가능하다.

스케치 입력을 사용하여 데이터베이스를 검색하기 때문에 원하는 자세와 비슷한 자세는 금방 찾을 수 있지만 정확히 일치하는 자세를 찾기는 쉽지 않다. 검색된 자세의 한 부분만이 마음에 들지 않는다면, 이를 수정하여 사용하면 편하겠지만, 이는 모션데이터베이스의 수정을 의미하기 때문에 현재의 시스템으로는 할 수가 없다. 비전문적인 사용자는 이러한 결과에 만족하겠지만, 좀 더 세부적인 묘사를 원하는 사용자를 위해 스케치를 이용한 데이터베이스의 수정을 가능하게 하면 좋을 것 같다. 이러한 요소가 추가되면 비전문적인 사용자를 위한 스케치입력을 사용함으로써 잃어버릴 수 있는 전문성을 보충할 수 있을 것이다.

또한 만들어진 애니메이션을 다양한 스타일로 자유롭게 변경할 수 있으면 좋을 것이다. 모션 데이터베이스를 이용하여 각 데이터에 있는 스타일들을 뽑아내서 이를 자유롭게 적용시킬 수 있는 시스템을 만든다면 보다 다양한 애니메이션을 쉽게 생성할 수 있을 것이다.

참고 문헌

[1] KOVAR, L., GLEICHER, M., AND PIGHIN, F. 2002. Motion graphs. ACM Transactions on Graphics (SIGGRAPH 2002) 21, 3, 473 – -482.

[2] ARIKAN, O., AND FORSYTH, D. A. 2002. Interactive motion generation from examples. ACM Transactions on Graphics (SIGGRAPH 2002) 21, 3, 483 – -490.

[3] ARIKAN, O., FORSYTH, D. A., AND O'BRIEN, J. F. 2003. Motion synthesis from annotations. ACM Transactions on Graphics (SIGGRAPH 2003) 22, 3, 402 – -408.

[4] LEE, J., CHAI, J., REITSMA, P. S. A., HODGINS, J. K., AND POLLARD, N. S. 2002. Interactive control of avatars animated with human motion data. ACM Transactions on Graphics (SIGGRAPH 2002) 21, 3, 491 – -500.

[5] DAVIS, J. AGRAWALA, M., CHUANG, E., POPOVIĆĆ, Z., SALESIN, D. A sketching interface for articulated figure animation. ACM SIGGRAPH 2003.

[6] THORNE, M., BURKE, D., AND VAN DE PANNE, M.

2004. Motion doodles: An interface for sketching character motion. ACM Transactions on Graphics (SIGGRAPH 2004) 23, 3, 424 – -431.

[7] IGARASHI, T., MATSUOKA, S. TANAKA, H., Teddy: A Sketching Interface for 3D Freefrom Design, ACM SIGGRAPH 1999.

[8] LEE, J., AND LEE, K. H. 2004. Precomputing avatar behavior from human motion data. In SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 79 – -87.