

데이터베이스 축소기법을 사용한 모바일 임베디드 시스템에서의 모션 캡처 기반 애니메이션

한 영 모[†]

요 약

본 논문의 목적은 기존의 모션 캡처 기반 애니메이션을 모바일 임베디드 시스템의 적은 메모리와 낮은 계산력에 적합하게 개조하는데 있다. 모바일 임베디드 시스템의 메모리를 효율적으로 사용하기 위한 방안으로서, PC 상에서 모션 캡처 데이터베이스를 압축하고, 압축된 모션캡처 데이터베이스를 모바일 임베디드 시스템에서 조금씩 압축 해제하고 사용 직후 삭제하는 패러다임을 제안한다. 이러한 목적으로 사용될 모션 캡처 데이터베이스 압축 기법으로서, 다항식 정합법을 활용하여 캡처된 운동 렌더링 파라미터의 데이터베이스를 축소하는 방법을 제안한다. 그리고 성능향상을 위해서 다항식 정합 함수를 최적화하는 방안도 함께 제안한다. 이렇게 설계된 다항식 정합법에 기반한 데이터 압축 방식을 사용하여, 상용화된 모바일 임베디드 시스템에서 효율적인 모션 캡처 기반 애니메이션 제작을 시현한다.

키워드 : 모바일 임베디드 시스템, 멀티미디어 데이터베이스, 애니메이션, 모션 캡처

Motion-Capture-Based Animation in Mobile Embedded Systems Using Motion Capture Database Reduction Technique

Youngmo Han[†]

ABSTRACT

The objective of this paper is to accommodate the existing motion-capture-based animation to small memories and low computing powers of mobile embedded systems. To use efficiently memories, we propose a paradigm in which a motion capture database is compressed on a PC and so-compressed motion capture database is decompressed little by little on mobile embedded systems and the decompressed data are eliminated right after used. As a compression method for this paradigm we propose an approach that compresses captured motion rendering parameters using a polynomial function fitting method. To enhance its performance we also propose an optimization method for the degree of the polynomial fitting function. Using so-obtained compression method we demonstrate motion-capture-based animation on commercial mobile embedded systems.

Key Words : Mobile Embedded Systems, Multimedia Database, Animation, Motion Capture

1. 서 론

모션캡처는 인체가 동작하는 움직임을 즉시 또는 약간의 시간 차이를 두고 기록하는 작업을 말한다. 기록된 정보를 바탕으로 컴퓨터 캐릭터에게 대상체의 위치 정보를 전달하여 인체의 움직임을 그대로 컴퓨터 캐릭터로 나타낼 수 있도록 한다. '키 프레임 애니메이션(Key Frame Animation)'과 '시뮬레이션'과 같은 기존 애니메이션 기술에 비해 '모션 캡처'의 가장 큰 장점은 실시간 애니메이션 영상과 자연스

럽게 보이는 높은 질의 영상을 제공할 수 있다는 점이다. 모션 캡처를 이용하면 3D로 모델링한 캐릭터를 복잡한 수학적 계산 없이 자연스럽게 움직이게 할 수 있다[1-3].

모션 캡처 기반 애니메이션 제작 기술은 센서를 사용해서 대상체의 운동을 측정하여 운동 렌더링 파라미터를 추출한 후, 이 운동 파라미터를 컴퓨터에 옮겨서 그래픽 툴을 사용하여 애니메이션화한다. 이러한 일련의 작업이 현재의 데스크 톱 컴퓨터에서는 문제없이 수행될 수 있지만, 모바일 임베디드 시스템에 그대로 적용되기에는 다소 무리가 있다. 왜냐하면, 모바일 임베디드 시스템의 경우, CPU의 계산 능력과 메모리의 크기가 매우 작기 때문에 이에 특화된 방법이 필요하게 된다.

예를 들어, 한 대상체의 움직임을 애니메이션화 한다고

※ 이 논문은 2007년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국과학기술진흥재단의 지원을 받아 연구되었음(KRF-2006-003-101610).

† 정 회 원 : 한양사이버대학교 컴퓨터공학과 교수
논문접수: 2007년 4월 23일, 심사완료: 2007년 9월 30일

생각해 보자. 이 경우, 대상체의 각속도, 선속도 성분은 모두 6개이다. 각 성분을 32bit (4 byte) 숫자로 표현한다면, 한 개의 표본 프레임에서 풍선의 운동을 나타내기 위한 데이터량은 8 byte * 6 = 48 byte 가 된다. 또한 애니메이션에서는 보통 초당 25 프레임을 사용하므로, 1초 동안의 풍선의 운동을 나타내기 위한 데이터량은 48 byte * 25 = 1.2 Kbyte 가 된다. 그리고 1분 동안의 풍선의 운동을 나타내기 위한 데이터량은 1.2 Kbyte * 60 = 72 Kbyte 가 된다. 만일 각 프레임 상의 대상체가 10개의 구성요소로 구성되어 있다면, 이 대상체의 운동을 1분 동안 표현하려면, 72 Kbyte * 10 = 720 Kbyte = 0.72 Mbyte 의 데이터량이 필요하다. 따라서 이 운동을 20분만 기술해도 14.4 Mbyte의 데이터량이 필요하다.

상용화된 모바일 폰의 RAM과 Flash Memory가 각각 30MB 정도 밖에 되지 않는다[4]는 점을 생각하면, 이 정도의 데이터량을 저장했다가 쓰기에 다소 무리가 있다. 따라서, 운동 파라미터의 데이터량을 줄여서 모바일 폰의 메모리에 저장할 수 있는 방법을 고안하는 것이 효율적이다. 이러한 취지에서 본 논문은 모바일 임베디드 시스템의 메모리 사용 효율을 높이기 위해서, 추출된 운동 렌더링 파라미터의 데이터량을 줄여서 메모리에 저장하는 방안을 모색한다.

그럼 다음으로 모션 캡처 데이터 압축에 사용될 수 있는 기존의 압축 기술에 대해 살펴보겠다. 동영상 압축 기술은 다방면에 응용이 가능하여 지금까지 많은 연구가 수행되어 왔다. 예를 들면, 예측자/교정자 방법[10], 웨이블릿을 사용한 공간 일관성 방법[11], PCA 방법[12], 강체 운동을 하는 메쉬 부분 검출에 기반한 방법[13] 등이 있다. 이러한 기존의 동영상 데이터 압축 방법의 대부분은 전체 이미지 프레임의 복잡한 수색 과정을 포함하고 있는데, 이러한 방법은 일반 컴퓨터에서는 좋은 성능을 발휘할 수 있으나 모바일 임베디드 시스템의 적은 계산력에는 적합하다고 보기 어렵다.

최근에 Arikan [14]이 모션 캡처 데이터베이스 전용 압축 기법을 발표하였다. 이 방법은 압축률이 매우 높아서 메모리를 효율적으로 쓸 수 있다는 강점이 있다. 하지만 압축 해제 과정이 상당히 복잡하고 계산량이 많다는 단점이 있다. 따라서, 이 방법은 PC 용으로는 적합하지만, 모바일 임베디드 시스템의 저사양 CPU에는 적합하다고 보기 어렵다.

계산량이 적은 간단한 데이터 축소 방법으로는 다운 샘플링이 많이 사용되고 있다[15]. 그리고 다운 샘플링을 통해 압축된 데이터는 보간법을 사용해 압축해제가 가능하다[16]. 다운 샘플링/보간법 방법은 사용이 간편하여 실용적으로도 많이 사용되고 있는데, 그 대표적인 예가 Acrobat 3의 bicubic 스플라인 다운 샘플링이다. 다운 샘플링/보간법 방법은 계산량이 적어서 모바일 임베디드 시스템의 저 계산력에 적합한 방법이라 볼 수 있다. 하지만, 압축률과 압축 에러 면에서 약점이 있다.

본 논문에서는 모바일 임베디드 시스템에 적합하도록 계산량이 적으면서, 기존의 다운 샘플링/보간법 방식과 비교하여 압축률과 압축 에러가 우수한 모션 데이터 압축 방법을

제안하고자 한다.

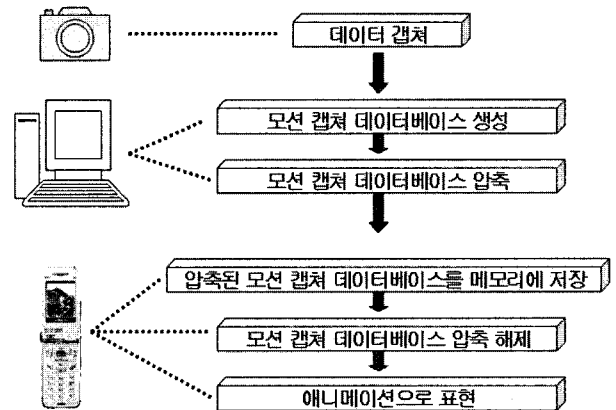
본 논문의 구성은 다음과 같다. 1장의 서론에 이어 2장에서는 모바일 임베디드 시스템 용 모션 캡처 기반 애니메이션 제작 패러다임에 대해 설명하며, 3장에서는 모바일 임베디드 시스템 용 모션캡처 데이터베이스 압축 기법을 설명하고 분석한다. 4장에서는 제안한 방식을 상용화된 모바일 임베디드 플랫폼에 적용하고 성능을 검증한다. 그리고 마지막으로 5장에서 결론을 도출한다.

2. 모션 캡처 기반 애니메이션 제작 패러다임

(그림 1)은 본 논문에서 제안하는 모바일 임베디드 시스템에서의 모션 캡처 기반 애니메이션 제작 패러다임을 보여 주고 있다.

먼저 카메라 등의 모션 캡처 장비를 사용하여 모션 캡처 데이터를 얻는다. 이렇게 얻어진 모션 캡처 데이터들을 PC로 옮겨서, 모션 캡처 데이터베이스를 생성한 후, 모션 캡처 데이터베이스 압축 기법을 사용하여 데이터를 축소한다. 압축된 모션 캡처 데이터베이스를 모바일 임베디드 시스템의 메모리에 저장한 후, 모션 캡처 데이터베이스를 필요한 부분만 조금씩 압축 해제하여 애니메이션으로 표현한다. 그리고 압축 해제된 데이터는 애니메이션으로 출력된 직후 메모리에서 삭제한다. 이와 같은 방법을 통해서 애니메이션의 전 프레임에 해당하는 많은 량의 모션 캡처 데이터를 한꺼번에 메모리에 저장하는 어려움을 피할 수 있다.

(그림 1)에서 주목할 점은 데이터 압축은 PC상에서 이루어지고, 압축해제는 모바일 임베디드 시스템 상에서 이루어진다는 것이다. PC에 비해 모바일 임베디드 시스템의 CPU가 매우 저 사양임을 고려하면, (그림 1)에서 사용할 모션 캡처 데이터 압축 방식은 압축 과정의 계산량이 높다고 해도 압축해제 과정의 계산량이 현저히 낮은 방식이어야 한다. 그리고 모션 캡처 데이터를 잘 표현할 수 있도록 압축 에러가 적은 방식이어야 한다.



(그림 1) 제안하는 모션 캡처 기반 모바일 임베디드 시스템에서의 애니메이션 제작 패러다임

3. 모션 캡처 데이터베이스 압축 기법

그럼 지금부터는 (그림 1)에 표시된 ‘모션 캡처 데이터베이스 압축’ 기법에 대해 살펴보도록 하자. (그림 1)의 모션 캡처 데이터베이스 압축 기법의 필요조건은, 첫째 압축해제 과정이 매우 단순해야 하며, 둘째 모션 캡처 데이터를 잘 표현할 수 있도록 압축에러가 적어야 한다는 것이다.

예를 들어, 최근에 Arikian [14] 이 발표한 모션 캡처 데이터베이스 축소 기법의 경우 압축률이 매우 높아서 메모리를 효율적으로 쓸 수 있다는 강점이 있다. 그러나 압축 해제 과정이 상당히 복잡하고 계산량이 많아서 PC 용으로는 적합하지만 모바일 임베디드 시스템의 저 사양 CPU에는 적합하지 않다. 모바일 임베디드 시스템은 메모리가 적을 뿐 아니라 CPU도 매우 저 사양이므로, 압축률 뿐 아니라 압축 해제 과정의 계산량도 중요한 요인이 된다. 이러한 취지에서 본 논문에서는 압축 해제 과정의 계산량이 많지 않은 한도 내에서 압축률을 높이는 방안을 모색하고자 한다.

계산량이 적은 기존의 데이터 압축법으로는 다운 샘플링과 양자화 기법(quantization) 등을 들 수 있다. MPEG 압축에서와 같이 일반적으로 사용되는 양자화 기법(quantization)에 의한 데이터 축소는 각 프레임 간의 운동의 불연속을 초래해서 모션 캡처 기반 애니메이션의 장점인 운동의 자연스러움을 깨뜨릴 수 있다. 따라서 본 논문에서는 모션 캡처 기반 애니메이션의 장점인 운동의 자연스러움을 보존할 수 있도록, 각 프레임 간 운동 파라미터의 자연스러운 연속성을 강조한 모션 캡처 데이터 축소 기법을 개발하고자 한다. 이와 같은 목적으로 본 논문에서는 추출된 모션 캡처 데이터(운동 렌더링 파라미터)의 프레임 별 변화를 연속적(continuous)이고 매끈한(smooth) 시간 정합 함수(fitting function)로 압축 근사화하는 기법을 도입하고자 한다. 이 경우, 정합된 함수의 파라미터가 바로 압축된 데이터가 된다. 이렇게 얻어진 정합 함수의 파라미터를 메모리에 저장하고, 정합 함수의 함수값 계산을 통해 모션캡처 데이터를 복원해 낼 수 있다. 결국 (그림 1)의 ‘모션 캡처 데이터베이스 압축 해제’ 과정은 단순히 정합 함수 값 계산으로 귀결되므로, 본 방법은 매우 쉽고 효율적인 압축 해제 방법을 제공하게 된다.

이러한 정합 함수 기반 모션 캡처 데이터 압축 기법의 정합 함수로서 다양한 함수가 사용될 수 있는데, 모바일 임베디드 시스템의 CPU 계산력이 낮다는 점을 감안해서 가능한 쉬운 정합 함수를 사용하는 것이 압축, 해제 과정을 단순화할 수 있는 요령이 된다. 본 논문에서는 가장 단순한 함수 중 하나인 다항식 곡선을 사용하여 제안하는 함수 정합법 기반 모션 캡처 데이터베이스 압축 방식을 설명하도록 하겠다.

다항식 정합법은 기존의 수치해석 관련 서적들에서 기본 알고리즘이 소개되어 있다[17]. 그 사용 예를 잠시 살펴보도록 하자. 편의상 기존에 알려진 다항식 정합법을 ‘polyfit’이라 부르도록 한다. 모션 캡처 기술을 사용하여, 대상체의 운동 파라미터 벡터 P의 시간별 변화 값을 데이터로 얻었다고

하자. 그리고 P의 각 스칼라 성분을 polyfit를 사용해서 다항식 곡선으로 근사화하는 경우를 생각해 보자.

본 논문에서 제안하는 아이디어는 P의 각 성분의 프레임 별 데이터를 모바일 임베디드 시스템의 메모리에 저장하는 것보다, 다항식 곡선 정합 함수의 계수를 메모리에 저장하고, 필요시 함수 값을 계산하여 쓰자는 것이다. 일반적으로 다항식 정합 곡선의 계수는 데이터 계수보다 작거나 같으므로 이 같은 방법을 통해 데이터 압축의 효과를 기대할 수 있다. 주어진 모션 캡처 데이터베이스를 세부 구간(편의상 ‘윈도우’라 부르기로 한다)으로 나눈 후, 각 세부 구간별로 위와 같은 다항식 정합법을 적용하여 전체 모션 캡처 데이터베이스를 압축할 수 있다.

다항식 정합법에 기반한 압축방식을 사용할 경우, 압축 해제 과정은 이 정합 함수 계수로부터 함수값을 계산해 내는 단순 과정으로 귀결된다. 따라서 압축 해제 과정에 거의 추가 계산량을 필요로 하지 않는다. 이는 (그림 1)의 모션 캡처 데이터베이스 압축 과정의 필요조건 중 나머지 하나인 계산량이 매우 적은 압축해제 과정에 부합하는 것이다. 따라서 본 다항식 정합법에 기반한 모션 캡처 데이터베이스 압축기법은 (그림 1)의 모션 캡처 데이터베이스 압축 기법의 두 가지 필요조건, 즉 압축해제 과정이 매우 단순해야 하며, 모션 캡처의 자연스러운 모션을 유지할 수 있어야 한다는 조건을 만족한다고 할 수 있다.

기본적으로 다항식 정합법의 경우 정합 함수의 차수선택이 데이터 압축 성능에 큰 영향을 미친다고 볼 수 있다. 정합 곡선의 차수를 높일수록 정합된 데이터의 근사화 에러가 더욱 작게 될 것이라는 사실을 추측할 수 있다. 하지만, 정합 곡선의 차수를 높일수록 메모리에 저장해야 하는 정합 곡선의 계수가 많아져서 데이터량이 증가되는 단점이 생기게 된다. 따라서 정합 곡선의 차수와 데이터량 사이에 trade-off 문제가 발생하게 된다. 이점에 대해 좀 더 자세히 살펴보도록 하자.

곡선의 차수를 점차 높여 가면, 곡선의 계수를 많이 저장해야 하므로 데이터량이 늘게 되고, 반면 근사화 에러는 줄게 된다. 예를 들어, 압축할 데이터의 개수가 n개라면, n-1차 정합 곡선을 사용하는 경우에 정합 에러가 최소가 된다. n-1차 정합 곡선의 경우, 정합 함수 계수는 모두 n개가 되므로, 정합 함수의 계수 값을 정하는 문제가 exact-determined problem이 된다. 따라서 이 경우에 가장 근사화 에러 값이 작아지게 된다. 반면 데이터로서 n-1차 정합 곡선의 계수 n개를 사용해야 하므로, 데이터량의 축소 효과가 없어지게 된다.

그리고 또 한 가지 고려해야 할 사항은 같은 크기로 분할된 세부 구간에 같은 차수의 정합 곡선을 적용하는 확실적인 다항식 정합 방법은 모션 파라미터의 변화량이 많은 세부 구간에서 상대적으로 오차가 크게 된다. 즉, 모션 파라미터의 변화량이 큰 구간에서는 정합 곡선의 차수를 높이거나, 다항식 정합을 적용하는 윈도우의 크기를 줄이는 것이 압축 에러를 줄이는 방법이 된다. 따라서 알고리즘의 효율을 높

이고, 압축 에러를 줄이기 위해서는 '세부 구간의 크기/정합 곡선의 차수' 비(ratio)를 최적화 할 필요가 있다.

이상의 내용을 바탕으로 앞에서 제안한 '다항식 정합법에 기반한 모션 캡처 데이터베이스 압축 기법'에 '윈도우의 크기/정합 곡선의 차수' 비를 최적화 하는 모듈을 추가하여 다음과 같이 확장 제안한다.

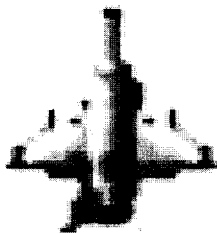
[알고리즘1] 정합 함수 차수의 최적 선택에 기반한 데이터의 다항식 정합

- (a) 정합을 수행할 세부 구간의 크기, 즉 윈도우의 크기를 정한다. 예를 들어, n-프레임 데이터로 가정하자.
- (b) 정합 곡선의 차수를 n-1로 정하고, 윈도우 안의 데이터에 대해 n-1 차의 다항식 정합을 수행한다.
- (c) 근사화 에러값이 임계값 보다 작으면, 정합 곡선의 차수를 하나 줄이고, 주어진 윈도우 안의 데이터에 대해 다항식 정합을 수행한다.
- (d) 근사화 에러 값이 임계값을 넘을 때까지 (c)를 반복한다.
- (e) 근사화 에러 값이 임계 값을 넘기 직전의 값을 최종 결과값으로 채택한다. 즉, 이 때의 정합 함수의 계수를 최종 정합 데이터로 사용한다. 그리고 그 다음 윈도우에 대해서, (b)~(e)를 반복한다.

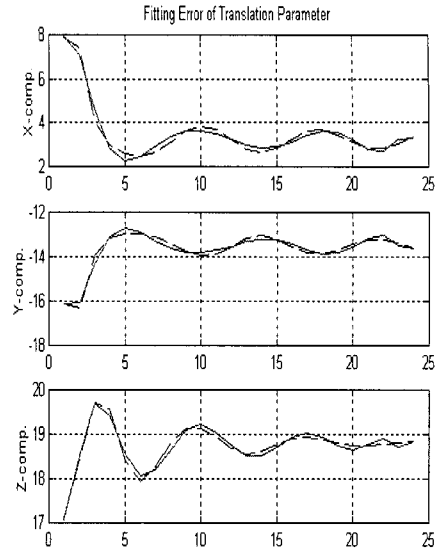
본 논문에서 제안한 다항식 정합법 기반 데이터 축소 기법은 데이터량을 줄여주는 효과 외에도 몇 가지 부가적인 장점을 갖게 된다. 예를 들면, 각 프레임 간의 보간 효과를 가져와서 필요시 좀 더 많은 프레임 데이터를 산출해 낼 수 있다는 이점이 있다.

4. 구현 및 실험 결과

이 절에서는 본 논문에서 제안한 정합법 기반 모션 캡처 데이터베이스 압축 기술의 성능을 평가하고, 이를 이용하여 상용화된 모바일 임베디드 시스템에서 애니메이션을 구현하는 예를 보인다.



(그림 2) 모션 캡처 기반 애니메이션의 대상체



(그림 3) 대상체의 운동 파라미터와 제안한 방법을 사용한 운동 파라미터 비교.

먼저 본 논문에서 제안한 알고리즘을 적용할 시뮬레이션 시나리오를 살펴보도록 하자. 본 시뮬레이션 시나리오에서는 (그림 2)에 주어진 F-15 전투기를 모션 캡처의 대상체로 선정하고, 기존에 알려진 모션 캡처 기술을 활용해서 (그림 3)의 점선과 같이 24 프레임의 운동 파라미터가 구해졌다고 가정한다. (그림 3)은 대상체의 운동 파라미터와 본 논문에서 제안한 방법을 사용해서 근사화된 운동 파라미터를 비교한 것이다. 이 그림에서 점선은 대상체의 운동 파라미터이고, 실선은 본 논문에서 제안한 방법을 사용해서 압축한 후 다시 압축 해제를 통해 복원한 운동 파라미터이다.

이렇게 얻어진 F-15 전투기의 운동 파라미터의 데이터를 압축하기 위해서, 앞 절에서 제안한 다항식 정합법에 기반한 데이터 압축 기술을 적용하였다.

<표 1>은 본 논문에서 제안한 데이터 압축 기술을 통해 얻은 압축된 데이터, 즉 정합 함수의 계수를 보여 주고 있다. 이 표에서 P(1), P(2), P(3)는 각각 대상체의 x, y, z-축 방향 궤적을 표시한다. 그리고 a0, ..., a10은 정합 곡선의 차수를 표시하는데, a0가 상수항, a10이 최고차 항이다.

<표 1> 제안한 방법을 사용해서 압축된 운동 파라미터

	P(1)	P(2)	P(3)
a0	-0.0000	0.0000	0.0000
a1	0.0000	-0.0000	-0.0000
a2	-0.0000	0.0000	0.0000
a3	0.0003	-0.0003	-0.0008
a4	-0.0061	0.0065	0.0162
a5	0.0803	-0.0835	-0.1957
a6	-0.6966	0.7087	1.4858
a7	3.8428	-3.8228	-6.7758
a8	-12.1202	11.9365	16.8902
a9	16.9155	-17.4036	-18.9525
a10	-0.0569	-7.4951	24.6070

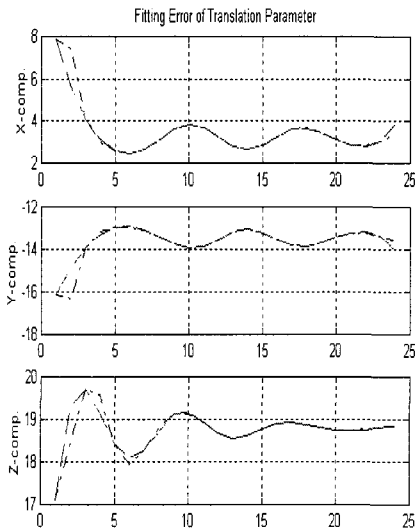
<표 1>에서 보듯이 정합 함수 계수는 모두 11개이므로, 결과적으로 24프레임의 데이터가 10차수의 정합 함수로 근사화 되었음을 보여주고 있다.

데이터 압축률을 따져 보면, 24개의 운동 파라미터가 11개의 정합 함수 데이터로 축소되었으므로, 압축률은 $13/24=54.2\%$ 가 된다.

표1에 주어진 압축된 데이터(다항식 정합 함수의 계수)를 압축 해제(정합 함수값 계산)하여 복원한 운동 파라미터가 (그림 3)의 실선으로 주어져 있다. 이 그림을 통해서 압축 에러를 계산할 수 있는데, 그 척도로서 압축 에러율을 $\sum_i (\bar{x}_i - x_i)/x_i$ 로 정의하자. 여기서 x_i 는 주어진 대상체의 운동 파라미터((그림 3)에서 점선으로 표시)이고, \bar{x}_i 는 압축과 압축 해제 과정을 거친 운동 파라미터((그림 3)에서 실선으로 표시)이다. (그림 3)의 압축 에러율을 계산해 보면, 첫 번째, 두 번째, 세 번째 그래프의 압축 에러율이 각각 5.05(%), 1.12(%), 0.46(%) 이 된다.

본 논문에서 제안한 압축 방식의 성능을 기존의 방식과 비교해 보기 위해서, 같은 압축률을 갖는 스플라인 방식(다운 샘플링을 통해 압축하고 스플라인 보간법을 사용해서 복원하는 방식)의 경우도 고려하였다. (그림 3)의 점선으로 주어진 대상체의 운동 파라미터에 50%의 압축률을 갖도록 2:1 다운 샘플링을 한 후, 스플라인 보간법을 통해 복원한 결과가 (그림 4)에 주어져 있다. (그림 4)는 대상체의 운동 파라미터와 스플라인 방식을 통해 근사화된 운동 파라미터를 비교한 것이다. 이 그림에서 점선은 대상체의 운동 파라미터이고, 실선은 운동 파라미터를 2:1 다운 샘플링을 통해 압축한 후 스플라인 보간법을 통해 복원한 운동 파라미터이다.

(그림 3)과 (그림 4)를 비교해 보면, 신호가 급변하는 구간인 0-5 초 사이와 끝부분인 24초 근처에서 스플라인 방식이 본 논문에서 제안한 방식보다 압축 에러가 큰 것을 볼 수 있다.



(그림 4) 대상체의 운동 파라미터와 스플라인 방법을 사용한 운동 파라미터 비교

<표 2> 운동 파라미터의 압축 에러율 비교

압축에러율	제안하는방법	스플라인 방법
P(1)	0.0505	0.1005
P(2)	0.0112	0.0236
P(3)	0.0046	0.0103
합계	0.0107	0.0225

<표 3> 압축과 압축해제 과정의 계산량 비교

계산량(sec)	제안하는방법	스플라인 방법
압축	2.5340	0.0300
압축해제	0.0000	0.0600

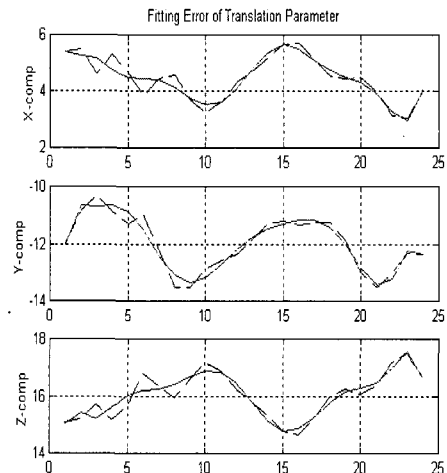
에러율을 좀 더 정확히 살펴보기 위해서 앞에서 정의한 에러율 계산법에 따라 두 방식의 에러율을 계산한 결과가 표 2에 주어져 있다. 이 표에서 압축 에러율은 $\sum_i (\bar{x}_i - x_i)/x_i$ 로 계산된다. 여기서, x_i 와 \bar{x}_i 는 각각 대상체의 운동 파라미터와 압축 후 복원된 운동 파라미터이다.

(그림 3)과 (그림 4)에서 확인한 것과 같이 제안하는 방법이 기존의 스플라인 방법에 비해 같은 압축률일 경우 좀 더 낮은 압축 에러율을 보임을 알 수 있다.

표3은 본 논문에서 제안한 방식과 스플라인 방식의 계산량을 보여주고 있다. 여기서 계산 시간을 계산량의 척도로 삼았다. 표3의 결과를 보면 제안하는 방법이 스플라인 방식에 비해 압축 계산량은 많지만, 압축해제 계산량은 훨씬 적은 것을 볼 수 있다.

이러한 특성은 본 논문에서 제안한 압축 및 압축해제 방식이 (그림 1)에서 제안한 모션캡처 패러다임, 즉 압축은 PC 상에서, 그리고 압축 해제는 모바일 임베디드 시스템 상에서 행한다는 패러다임에 적합하다고 볼 수 있다.

왜냐하면, PC의 계산력은 충분한데 비해 임베디드 시스템의 계산력이 매우 열악하므로, 압축과정의 계산량이 증가하더라도 압축해제과정의 계산량을 줄이는 것이 필요하기 때문이다.



(그림 5) 대상체의 운동 파라미터와 제안한 방법을 사용한 운동 파라미터 비교

<표 4> 제안한 방법을 사용해서 압축된 운동 파라미터

	P(1)	P(2)	P(3)
a0	-0.0000	0.0000	0.0000
a1	0.0000	-0.0000	-0.0000
a2	-0.0000	0.0000	0.0000
a3	0.0001	-0.0001	-0.0001
a4	-0.0023	0.0017	0.0022
a5	0.0327	-0.0259	-0.0324
a6	-0.3017	0.2661	0.3177
a7	1.7984	-1.7770	-2.0262
a8	-6.6711	7.4431	8.1074
a9	14.3592	-18.5150	-18.9603
a10	-15.9540	24.6540	22.8380
a11	12.1512	-24.0734	4.7927

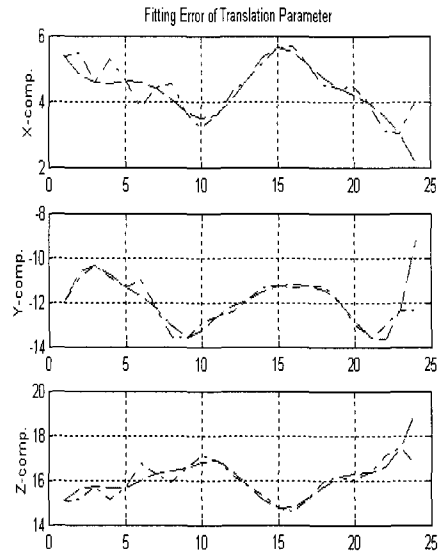
지금까지 얻은 결과를 확실하게 하기 위하여 예를 하나 더 살펴보도록 하자. (그림 2)에 주어진 F-15 전투기를 모션 캡처의 대상체로 선정하고, 기존에 알려진 모션 캡처 기술을 활용해서 (그림 5)의 점선과 같이 24 프레임의 운동 파라미터가 구해졌다고 가정하자.

앞 예제에서와 같이 (그림 5)에 주어진 대상체의 운동 파라미터에 본 논문에서 제안한 방식을 적용하였다. 그 결과 표4와 같은 압축 데이터를 얻었다. 24개의 운동 파라미터가 12개의 정합곡선의 계수로 압축되었으므로, 압축률은 50%가 된다.

비교를 위해서 50%의 압축률을 갖는 스플라인 방식을 함께 고려하였다. 본 논문이 모바일 임베디드 시스템 용 모션 캡처 데이터베이스 압축 기법을 제시한 최초의 논문인 관계로, 비교용으로 사용할 모바일 임베디드 시스템용 압축기법을 찾기 어렵다. 그래서 일반 데이터 압축 기법 중에서, 모바일 임베디드 시스템의 성능 한계에 적응 가능한, 즉 계산량이 적은 방법을 선정하였다. 이러한 취지에서 간단하면서도 실용화되어 널리 사용되고(예를 들면, PDF 구현[19] 등) 있는 다운샘플링에 의한 압축 기법과 스플라인 보간법을 사용한 압축 복원 기법을 비교 대상으로 선정하였다. 다운샘플링(또는 서브 샘플링)은 신호의 샘플링 비를 줄이는 작업을 말하는데, 데이터의 크기를 줄이는 데 많이 사용된다 [15]. 그리고 샘플링된 신호의 복원은 일반적으로 보간법을 통해 이루어진다 [16]. 많은 종류의 보간 함수가 알려져 있는데, 그 중 스플라인 보간법은 임의의 개수의 데이터에 적용 가능하므로 실용적으로 많이 사용되고 있다[20]. 그 대표적인 예가 Acrobat 3의 bicubic 스플라인 다운 샘플링이다 [19]. 스플라인 보간법에 대한 자세한 설명은 참고문헌 [20]에서 찾아 볼 수 있다.

제안하는 방법과 스플라인 방법을 사용하여 압축과 압축 해제 과정을 거친 운동 파라미터가 각각 (그림 5)와 (그림 6)에 실선으로 표시되어 있다.

그리고 압축 에러와 계산량이 표 5와 표 6에 주어졌다. 표5를 살펴보면, 같은 압축률을 갖는 경우 본 논문에서 제안한 방법이 기존의 스플라인 방법에 비해 에러율이 낮음을 알 수 있다.



(그림 6) 대상체의 운동 파라미터와 스플라인 방법을 사용한 운동 파라미터 비교.

<표 6>의 결과를 살펴보면, 본 논문에서 제안한 방법이 스플라인 방법에 비해 압축 계산량은 많지만 압축 해제 계산량이 훨씬 적은 것을 알 수 있다. 이 결과는 <표 3>과 <표 4>의 결과와 일치하는 것으로서, 본 논문의 방식이 (그림 1)에서 제안한 임베디드 시스템용 모션 캡처 패러다임에 적합한 방식임을 보여주고 있다.

마지막으로 표1에 주어진 압축된 데이터, 즉 정합 함수 계수를 사용하여, F-15(그림 2)의 운동 파라미터를 복원하고 이를 사용해서 모바일 임베디드 시스템에서 F-15의 애니메이션을 제작하는 예를 보이도록 하자.

먼저 본 구현에 사용할 상용화된 모바일 임베디드 플랫폼으로서, 신지 소프트에서 개발한 GNEX를 사용하였다. GNEX는 SK-Telecom에서 채택한 GVM 플랫폼을 그래픽 기능을 강화해 확장한 버전으로서, 모바일 C를 사용하여 모바일 폰의 프로그램을 제작하고 시뮬레이션 할 수 있는 통합 개발 환경을 제공한다[18].

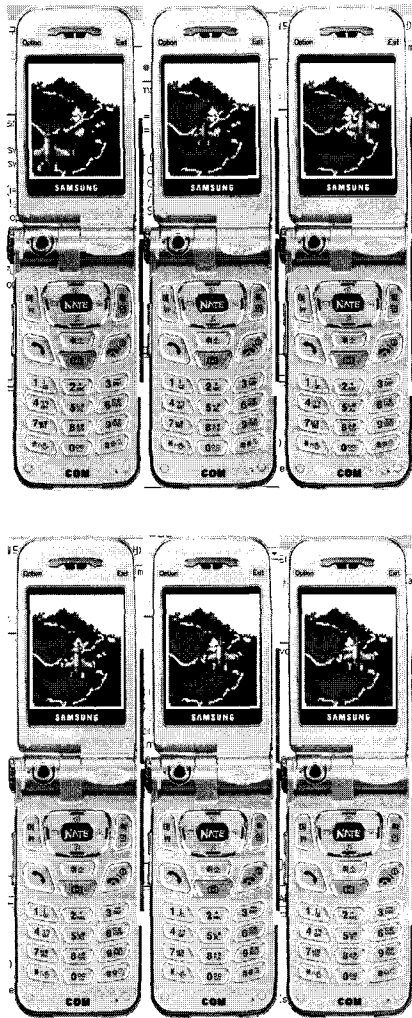
(그림 7)은 GNEX의 통합 개발 환경에서, 모바일 C를 사용하여 표 1의 축소된 모션 캡처 데이터베이스(다항식 정합 함수의 계수)를 압축해제(정합함수 값 계산)하여 F-15의 운동 파라미터를 복원하고, 애니메이션으로 출력한 결과 화면을 보여주고 있다. 차례대로 0, 2, 4, 8, 12, 20 번째 프레임을 보여주고 있다.

<표 5> 운동 파라미터의 압축 에러율 비교

압축에러율	제안하는방법	스플라인 방법
P(1)	0.0573	0.1100
P(2)	0.0195	0.0577
P(3)	0.0157	0.0370
합계	0.0205	0.0501

〈표 6〉 압축과 압축해제 과정의 계산량 비교

계산량(sec)	제안하는방법	스플라인 방법
압축	0.8210	0.0200
압축해제	0.0000	0.0500



(그림 7) 축소된 데이터를 사용한 모바일 임베디드 시스템의 애니메이션 제작 결과

(그림 7)의 모바일 폰 영상은 GNEX에서 제공하는 에뮬레이터 기능을 활용하여 제작한 것으로서, 삼성의 SCH-E2000 모바일 폰에서 구현한 경우를 보여주고 있다. 모바일 C의 경우, ANSI C에 비해 그 기능이 미약하지만 본 논문에서 제안한 방식은 압축 해제 과정이 매우 간단해서, 제한된 기능의 모바일 C와 모바일 임베디드 시스템 플랫폼 상에서도 손쉽게 구현이 가능하다.

5. 결 론

일반적으로 모바일 임베디드 시스템은 CPU가 저사양이

고, 메모리가 매우 작은 한계를 가지고 있다. 따라서 기존의 모션 캡처 기반 애니메이션 제작과정을 그대로 적용하기에는 운동 파라미터의 데이터 양이 많아서 무리가 있다.

본 논문에서는 모션 캡처 기반 애니메이션 제작을 모바일 임베디드 시스템에서도 효율적으로 제작할 수 있도록 하기 위하여, 모바일 임베디드 시스템 용 모션 캡처 기반 컴퓨터 애니메이션 제작 패러다임을 제안하였다. 이 패러다임에서는 PC 상에서 모션 캡처 데이터베이스를 압축한 후, 압축된 모션캡처 데이터베이스를 모바일 임베디드 시스템에서 조금씩 압축 해제하고 삭제하는 일련의 과정을 통하여 모바일 임베디드 시스템의 메모리 사용 효율을 높인다.

이 패러다임에 사용될 압축 방식은 모바일 임베디드 시스템의 저사양 CPU에 적합하도록 압축 해제 과정이 최대한 단순해야 하며, 모션 캡처 데이터를 잘 표현할 수 있도록 압축 에러가 적어야 한다. 이러한 조건을 만족하는 데이터 압축 기술로서 다항식 정합법에 기반한 모션 캡처 데이터 압축 기법을 제안하였다. 그리고 본 방식의 성능 향상을 위하여, 다항식 정합 함수의 차수를 최적화하는 방법 또한 제안하였다. 본 기법을 F-15 전투기의 모션 캡처 기반 애니메이션 제작 과정에 적용한 결과 기존의 스플라인 방식(다운 샘플링을 통해 압축하고, 스플라인 보간법을 사용해서 복원)에 비해 압축에러가 적고 압축 해제 과정의 계산량이 적음을 확인하였다. 그리고 본 기법의 구현 예로서, 상용화된 모바일 임베디드 시스템 플랫폼인 GNEX 상에서 모바일 C를 사용하여 축소된 데이터베이스의 압축과정과 애니메이션 출력 과정을 시연하였다.

본 논문의 방법이 갖는 한계점으로는 다음 두 가지를 지적할 수 있다. 첫째, 윈도우 전체에 대해 하나의 정합 함수를 사용하기 때문에, 윈도우 내의 국부적으로 급변하는 불규칙 신호가 전체 윈도우의 압축률을 떨어뜨릴 수 있다. 둘째, 정합 함수의 차수를 최적화하기 때문에 압축된 데이터의 크기가 가변적이어서, 일정 크기의 변수에 저장하기 불편하다.

이와 같은 한계점을 개선하기 위해서 추후 연구에서는 다음 사항에 집중할 것이다. 첫째, 국부적인 불규칙 신호에도 안정된 압축률을 갖도록 보완 설계한다. 둘째, 압축된 데이터의 크기가 안정적일 수 있도록 정합 함수의 차수 대신 윈도우의 크기를 최적화하는 방안을 모색한다.

참 고 문 헌

- [1] 최인규, 박현주, "가상 현실에서 모션 캡처의 활용가능성에 관한 연구", 한국멀티미디어학회 추계학술발표논문집, pp.1-14, 2002.
- [2] 이경수, "Motion Control 기술이 영상제작에 미치는 영향에 관한 연구", 명지대 산업 대학원 석사학위논문, pp.87-88, 1998.
- [3] 최은옥, "스킨, 스켈레톤 바인딩 알고리즘을 이용한 다 관절체 모델의 구현", 충북대학교 대학원, 박사학위논문, p.9.

1998.

- [4] “임베디드 S/W 프로그래밍 교육을 통한 PDA 관련 교육 키트 개발에 관한 연구”, 한국전자통신연구원, 한국정보통신대학교 수탁과제 최종 연구 보고서.
- [5] C. M. Ginsberg, Human body motion as input to an animated graphical display, Master Thesis, Massachusettes institute of Technology, May 1993.
- [6] D. R. Maxwell, Graphical Marionette: A Mordern day Pinocchio, Master Thesis, Massachusettes Institute of Technology, June 1983.
- [7] S. Dyer, J. Martin, J. Zulauf, Motion Capture White Paper. Technical Report. Silicon Graphics, December 12, 1995.
- [8] F. W. Silva, L. Velho, P. R. Cavalcanti, and J. Gomes, “An architecture for motion capture based animation”, <http://w3impa.br/~nando/publ/sib97.pdf>.
- [9] G. B. Guerra-Filhol, “Optical Motion Capture: Theory and Implementation”, RITTA, Volume 7, Numero 2, 2005.
- [10] L. Ibarria and J. Rossignac, “Dynapack: space-time compression of the 3d animations of triangle meshes with fixed connectivity”, Proc. 2003 ACM SIGGRAPH/Eurographics symposium on computer animation, pp. 126-135, 2003.
- [11] I. Guskov and A Khodakovsky, “Wavelet compression of parametrically coherent mesh sequences”, Proc. 2004 ACM SIGGRAPH/Eurographics symposium on computer animation, pp. 183-192, 2004.
- [12] M. ALEXA and W. Muller, “Representing animations by principal components”, Eurographics computer animation and simulation, vol. 19, pp. 411-418, 2000.
- [13] D. L. James and C. D. Twigg, “Skinning mesh aimations”, Proc. SIGGRAPH, vol. 24, no. 3, pp. 299-407, 2005.
- [14] Okan Arikan, “Compression of Motion Capture Databases”, SIGGRAPH2006, pp. 1-8, 2006.
- [15] <http://en.wikipedia.org/wiki/Downsampling>.
- [16] “Advanced Topics in Signal Processing”, Jae S. Lim and Alan V. Oppenheim, Prentice-Hall, 1988.
- [17] 용호택, “MATLAB을 이용한 수치해석”, 다성출판사, 2000.
- [18] “WIPI GNEX를 이용한 모바일 프로그래밍”, 생능 출판사, 2006.
- [19] “HR Reporter”, <http://www.hrreporter.com/advertise/default.asp?load=MaterialSpeces06>.
- [20] “Applied Numerical Methods in C”, S. Nakamura, Prentice-Hall, 1993.



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~현재 한양사이버대학교 컴퓨터공학과 전임강사

2007년~현재 International Biographical Centre (England), Deputy Director General (Vice President 급)

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

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