

성분동작들간의 위상 대응성을 고려한 목표지향적 동작합성 기법
Goal-directed motion blending based on phase correspondence of component motions

Jung-Hee Cho, Moon-Ryul Jung, Doo-Won Paik

조정희*, 정문렬**, 백두원*

*숭실대학교 정보과학대학 컴퓨터학부

**서강대학교 영상대학원 디지털 애니메이션과

요 약

목표지향적 동작 합성기법은 원하는 동작의 타입(예, 발 차기, 손 뺏기)과 동작의 최초 및 최종상태가 주어졌을 때, 이 조건을 만족하는 새로운 동작을 유사한 동작들의 집합(동작군)에 속한 동작들을 혼합하여 자동적으로 생성하는 기법이다. 이 기법은 동작군에 속한 동작들의 커브를 지나는 연속적이고 완만한 곡면으로 생성한 후, 그 곡면 상에 있으면서 주어진 최초 및 최종상태를 만족하는 특정 동작 커브를 구하여 원하는 동작으로 삼는다. 이 기법은 성분동작을 선형보간하여 동작을 합성하는 기법과는 다른 접근방법이다. 또한 여러 동작들을 혼합하기 위해서는 동작들간에 위상(phase)이 서로 대응되어야 하므로, 본 논문에서는 성분동작의 위상대응성을 고려한 목표지향적 동작합성 기법을 제시한다.

1. 서론

3차원 애니메이션의 초기에는 애니메이터가 인체의 동작의 중요 프레임들에서의 관절각을 직접 지정하여 디스플레이 하는 키프레임 방식이 사용되었고, 근래에는 역운동학 등의 알고리즘을 이용하여 관절각을 생성하는 방법이 사용되고 있다.

그리고 최근에는 마그네틱이나 광학장비를 이용하여 인체가 움직이는 실제 동작을 캡춰한 후, 캡춰된 데이터를 실시간에 디스플레이 하는 방법이 이용되기 시작하였다. 동작캡춰 방법은 이미 캡춰된 동작과 유사한 동작이 요구될 때마다 기존의 동작 데이터를 사용하지 못하고 새로 캡춰를 해야 한다. 기존의 동작 데이터를

이용하여 새로운 동작을 생성하기 위해서 동작 변이[1,3,5,10], 동작 합성[3,4,6,9,13], 동작 전이[4,7,] 등의 방법이 제시되었다.

동작 변이는 유사한 동작 데이터를 이용하여 새로운 목표들을 만족하는 동작 데이터를 생성하는 것을 말한다[1]. 동작 합성은 두 개 이상의 서로 다른 데이터를 합성하여 새로운 데이터를 생성하는 것을 말한다[4]. 마지막으로 동작 전이는 각 동작 데이터를 서로 연결할 때 자연스럽게 연결될 수 있도록 데이터간의 사이를 보간 해주는 것을 말한다[7]. 동작합성 기법은 일반적으로 새로운 동작이 공간상의 어떤 목표를 성취해야 하는가 하는 문제는 직접적으로 고려하지 않는다. 애니메이터가 시행 착오를 통해 합성된 동작이 주어진 목표를 만족하도록 할 수 있을 뿐이다. 예외로서 [9]을 들 수 있는데, 이 기법에서는 손을 특정 목표위치에 두는 동작을 생성할 때, 손의 최종위치가 원하는 목표위치 주변에 있는 기존동작들을 선형보간하여 구한다. 선형보간에 사용하는 보간계수는 원하는 동작의 손의 최종위치와 기존동작들의 손의 최종위치간의 차이를 이용하여 구한

♣ 이 연구는 '97년도 과학재단 핵심전문연구(과제번호 971-0908-053-2)의 연구비 지원에 의한 결과임

다. 본 논문에서는 색다른 접근방법으로 이 문제를 해결한다. 즉 기존의 동작함수들의 집합을 하나의 곡면으로 표현한다. 그 다음, 원하는 동작의 최초자세와 최종자세가 주어지면 이 조건을 만족하면서 곡면 상에 존재하는 동작커브를 찾는다. 이 기법은 [9]과 비교하여 두 가지 점에서 다르다. 첫째, 원하는 동작의 최종상태뿐만 아니라 최초상태도 지정할 수 있다. 둘째, 생성할 동작을 다수의 성분 동작커브를 선형적으로 합성함으로써 구하는 것이 아니라, 다수의 성분동작커브들이 이루는 완만한 곡면 상에서 구한다. 즉 새분화된 동작군(무한히 많은 동작은 같은 동작군에 속하는 동작들)으로부터 새로운 동작을 이루는 부분동작들을 얻을 수 있다. 또한 동작의 특성을 나타내는 Behavior profile[1]을 사용하여 보간하였기 때문에 생성된 데이터는 이러한 동작의 특성을 유지하는 경우가 많다.

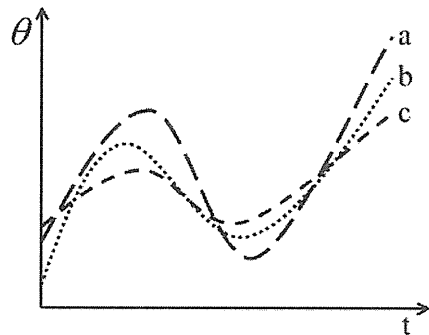
동작합성기법을 위해서는 서로 비슷한 동작들의 동작군을 형성하는 방법과 이 동작군이 이루는 곡면을 이용하여 새로운 목표를 가진 동작을 생성하는 방법이 필요하다. 또한 동작합성을 위해서는 성분동작들간의 위상대응성(phase correspondence)을 고려해야 한다. 즉 두 동작이 속도를 비롯하여 여러 가지 면에서 서로 다르더라도 위상은 서로 대응되어야 합성이 가능하다. [1]의 기법에서는 이를 고려하지 않아서 이를 적용할 수 있는 경우가 제한되어 있다. 따라서 본 논문에서는 위상대응성 문제를 고려하는 기법을 제시한다. 본 기법은 [3]에서 사용하는 동적 시간 변형 방법과는 달리 위상대응을 실시간에 처리할 수 있다. 실행 시간의 대부분은 동작군의 형성 및 부분동작으로 세분화하는 과정이 차지하고 있고, 이에 반하여 동작 곡면을 만들어서 데이터를 생성하는 과정은 상당히 적은 부분을 차지하고 있다. 동작군의 형성 및 부분동작으로 세분화하는 과정은 대부분이 처음에 한번만 수행하고 그 이후에는 이미 만들어진 동작군 데이터를 사용하기 때문에 전처리 작업을 하지 않기 때문에 실시간 처리가 가능하다.

2. 목표지향적 동작합성

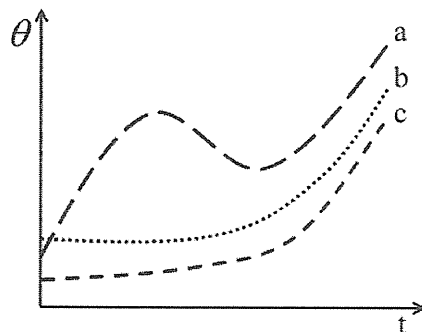
일반적인 목표지향적 기법은 동작을 취하는 캐릭터의 종단점들이 공간상에서 어떠한 위치에 도달하거나 캐릭터가 특정 위치에서 특정한 자세를 취하는 것을 목표로 삼는다. 그러나 본 기법의 목표는 목표자세를 만족시키는 동작 합성이다. 그러므로 목표지향적이라고 볼 수 없다.

2.1 동작군

인체의 동작은 관절들의 관절각이 변화함으로써 하나의 전체적인 동작을 생성한다. 이러한 동작을 동작 캡처와 같은 방법을 이용하여 손을 내미는 동작 등 종류에 따라 여러 개씩 동작들을 획득했다고 하자. 여기서 캡처된 동작 커브들이 유사하여 하나의 동작군을 형성하는지를 판단한다[그림1]. 동작군을 형성에 대한 자세한 설명은 [2]의 동작군 형성방법에 언급되어 있다.



(a) 동작 커브 a, b, c는 같은 동작군에 속한다.

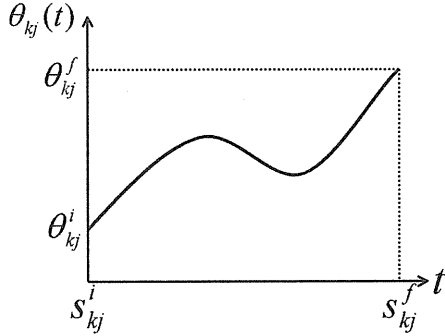


(b) 동작 커브 a는 b, c와는 다른 동작군에 속한다

[그림 1] 동작군의 분류 예

동작군을 분류하는 과정이 정확성에 따라 본 논문에서 제시된 방법의 적용 범위가 영향을 받는다. 실제로 동작 합성이 제대로 되기 위해서는 동일한 동작군에 속한 관절 커브들의 위상이 서로 대응되어야 하는데 만약 동작군 분류가 제대로 되지 않았다면 위상이 서로 대응

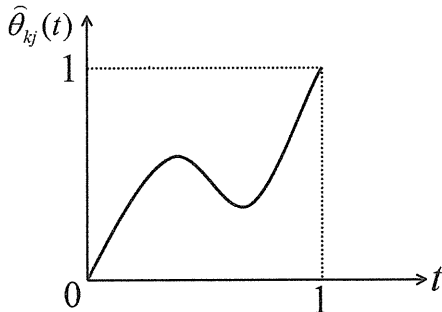
되지 않은 곡선들도 같은 동작군에 속할 수 있다. 경우에 따라 사람의 관여가 필요하다. 본 논문은 동작군 분류 과정이 정확하게 됐다는 가정 하에서 동작군에 속하는 동작 커브들의 극점들이 서로 유사한 분포를 가진다고 생각하여 동작을 합성한다.



[그림 2] 특정 관절의 커브(θ_{kj}^i 은 초기값, θ_{kj}^f 은 최종값, s_{kj}^i 은 시작 프레임, s_{kj}^f 은 최종 프레임, t 은 시간축, $\theta_{kj}(t)$ 은 관절축, k 는 캡춰된 동작의 수, j 는 각 관절을 나타낸다.)

2.2 동작 곡면 생성

임의의 관절 j 에 대한 동작군의 k 번째 관절 함수를 $\theta_{kj}(t)$ ($1 \leq k \leq m$, m 은 캡춰 동작의 수, $1 \leq j \leq n$, n 은 관절의 수, t 는 시간)라 하자 [그림2].



[그림 3] 정규화 관절 커브($\widehat{\theta}_{kj}(t)$ 축은 초기값과 최종값을 이용하여 θ 축을 정규화한 축, t 축은 시간을 $[0,1]$ 사이로 정규화한 축)

[그림2]와 같은 관절 함수들을 비교하기 위하여 정의 구역(시간)과 치역(관절각)을 정규화

(normalize) 한다[그림3].

정규화 관절각 함수 $\widehat{\theta}_{kj}(t)$ 는 다음과 같다.

$$\widehat{\theta}_{kj}(t) = \frac{\theta_{kj}(s(t)) - \theta_{kj}^i}{\theta_{kj}^f - \theta_{kj}^i} \quad (1)$$

$$s(t) = (s_{kj}^f - s_{kj}^i)t + s_{kj}^i$$

s_{kj}^i : 시작 프레임의 시간

s_{kj}^f : 최종 프레임의 시간

θ_{kj}^i : 초기 관절값

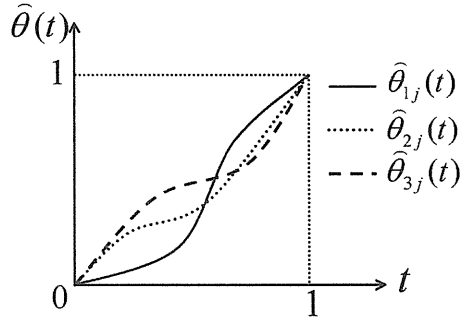
θ_{kj}^f : 최종 관절값

t : ($0 \leq t \leq 1$)

k : $1 \leq k \leq m$, m 은 캡춰 동작의 수

j : $1 \leq j \leq n$, n 은 전체 관절의 수

위 (1)식에 따라 생성된 $\widehat{\theta}_{kj}(t)$ 는 동작의 behavior profile을 나타낸다[1]. 동일한 동작

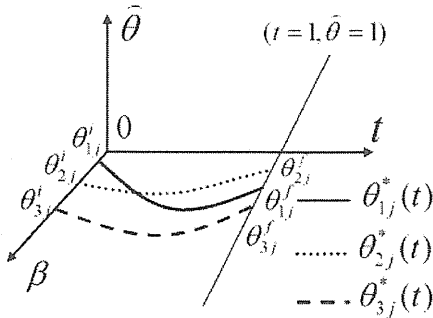


[그림 4] 정규화 관절각 함수 $\widehat{\theta}_{1j}, \widehat{\theta}_{2j}, \widehat{\theta}_{3j}$

군에 속한 여러 개의 $\widehat{\theta}_{kj}(t)$ ($k=1, \dots, m$)는 [그림 4]와 같이 나타낼 수 있다. $\theta_{kj}(t)$ 를 완전히 나타내기 위해서는 이것의 behavior profile $\widehat{\theta}_{kj}(t)$ 와 $\theta_{kj}(t)$ 의 초기값과 최종값이 필요하다. 따라서 $\theta_{kj}(t)$ 는 다음과 같이 $\theta_{kj}^*(t)$ 로 나타낼 수 있다.

$$\theta_{kj}^*(t) = [\widehat{\theta}_{kj}(t) + \{ \text{초기값, 최종값} \}] \quad (2)$$

$\theta_{kj}^*(t)$ 가 나타내는 정보는 [그림5]와 같이 나타낼 수 있다.



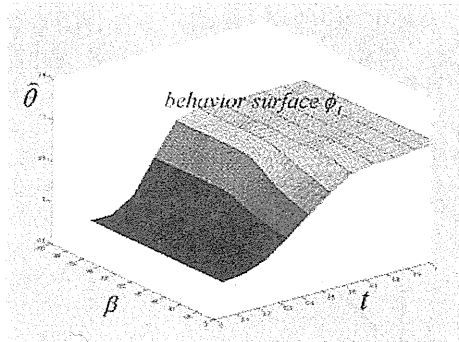
[그림5] θ_{kj} 의 behavior profile $\widehat{\theta}_{kj}(t)$ 와 초기값과 최종값을 3차원 공간상에 표현하고 있는 $\theta_{kj}^*(t)$ (β 축은 초기값과 최종값을 나타내는 축, $\widehat{\theta}$ 는 정규화된 관절, t 축은 시간을 나타낸다)

$\theta_{kj}^*(t)$ 의 의미를 알아보기 위해 $\widehat{\theta}_{1j}(t)$ 를 예로 들어보자. [그림4]에서 굵은 실선 $\widehat{\theta}_{1j}(t)$ 는 [그림5]에서 굵은 실선 $\theta_{1j}^*(t)$ 로 표현된다. $\theta_{1j}^*(t)$ 을 $(t-\widehat{\theta})$ 평면으로 투영시키면, [그림4]에서 보는 것과 같이 θ_{1j} 의 behavior profile인 $\widehat{\theta}_{1j}(t)$ 가 된다. 그리고 $\theta_{1j}^*(t)$ 는 $\theta_{1j}(t)$ 의 초기값에 해당하는 점 $(0,0,\theta_{1j}^i)$ 와 최종값에 해당하는 점 $(1,1,\theta_{1j}^f)$ 을 지나간다. 따라서 $\theta_{kj}^*(t)$ 는 $\theta_{kj}(t)$ 의 behavior profile과 초기값 및 최종값을 나타내므로 원래 커브 $\theta_{kj}(t)$ 을 잘 대표한다.

[그림5]에 표현되지 않은 다른 동작들은 $\theta_{1j}^*, \theta_{2j}^*, \theta_{3j}^*$ 과 인접해 있다고 볼 수 있다. 다시 말해서 같은 동작군에 속한 다른 커브들은 $\theta_{kj}^*(t)$ 들을 지나가는 곡면 상에 존재한다고 볼 수 있다. 두 곡선들이 꼬이는 경우, [그림6]과 같은 곡면이 존재할 수 없다고 생각할 수 있다. 그러나 곡선 $\theta_{1j}^*(t), \theta_{2j}^*(t), \theta_{3j}^*(t)$ 상의 점들을 각각 샘플링 한 후 이점을 지나가는 B-spline Surface로 곡면을 표현한다. 따라서 두 곡선이 꼬이더라도 곡면생성에 지장을 주지는 않는다. 이 곡면은 least square method를 사용하여 구한다[11]. B-spline이 동작데이터의 특성을 반영하지는 못하지만, 부드러운 곡면을 얻을 수 있다.

관절 j 에 대한 곡면 ϕ_j 는 관절 j 의 behavior profile 함수들로부터 만들어졌기 때문에 한 동작군의 behavior를 나타낸다. 이러

한 곡면 ϕ_j 를 behavior surface 라고 부른다 [그림6].



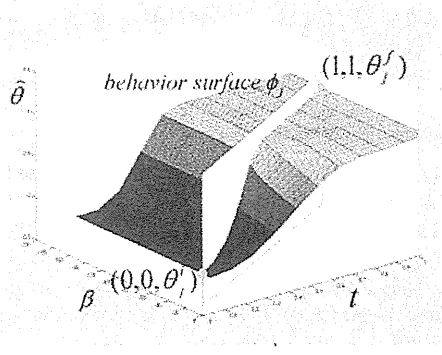
[그림 6] 동작군에 속한 관절커브를 지나가는 behavior surface ϕ_j

2.3 동작생성

사용자는 자신이 원하는 동작의 타입, 초기 자세 ($\theta_j^i, j=1, n$), 최종 자세 ($\theta_j^f, j=1, n$)을 입력한다. 시스템에서 애니메이션가 3차원 공간에 캐릭터의 자세를 인터랙티브한 방식으로 초기자세와 최종자세를 위치시키고, 시스템이 inverse kinematics를 이용하여 각 관절값들을 정하여 준다[12]. 시스템은 관절 j 의 동작이 behavior surface ϕ_j 상에 존재한다고 가정한다. 그리고 초기값 $(0,0,\theta_j^i)$ 와 최종값 $(1,1,\theta_j^f)$ 을 지나가며 behavior surface ϕ_j 상에 존재하는 커브 $\theta_j^*(t)$ 을 구한다. 다른 정보가 없기 때문에 이러한 커브 중에서 가장 간단한 커브 $\theta_j^*(t)$ 을 구한다. 커브 $\theta_j^*(t)$ 은 $(0,0,\theta_j^i)$ 와 $(1,1,\theta_j^f)$ 을 지나면서 $\widehat{\theta}$ 축과 평행한 평면과 behavior surface ϕ_j 가 교차하는 모든 점들로 이루어지는 커브이다. 구체적인 알고리즘은 [1]을 참조하라[그림7]. 만약 사용자가 원하는 초기 및 최종 관절각을 가진 동작이 이미 기존의 동작군에 존재하는 경우에는 알고리즘에 의해서 생성되어 나온 동작과 이미 존재하는 동작은 서로 일치한다.

여기서 각 관절들을 독립적으로 처리하여 동작을 생성할 경우 end-effector의 움직임이 어떻게 될 것인가와 생성한 커브가 단순히 θ_j^i 와 θ_j^f 을 만족하는 임의의 커브가 되지 않을까 하는 의구심이 생길 수 있다. 그러나 본 기법은 최초 자세에서 목표 자세로 향하여 가는 동작을 합성하는 알고리즘이다. 중간에

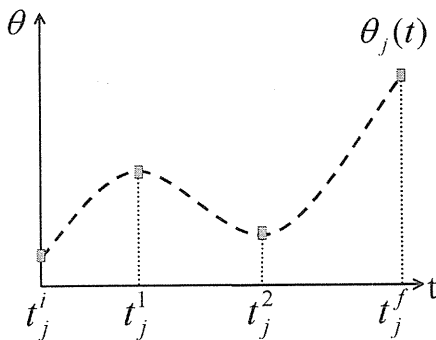
end-effector 가 어떻게 움직이던 간에 합성된 동작이 기존의 성분동작과 유사하므로 크게 문제되지 않는다.



[그림 7] 동작 생성 (초기값과 최종값 θ_j^i 와 θ_j^f 이 주어질 때, 점 $(0,0,\theta_j^i)$ 와 점 $(1,1,\theta_j^f)$ 을 지나고 $\hat{\theta}$ 축에 평행인 평면과 ϕ_j 의 교차커브를 새로운 동작으로 삼는다.)

2.4 위상 대응성

동작 합성이 제대로 되기 위해서는 동일한 동작군에 속한 관절 커브들의 위상이 서로 대응되어야 한다. 예를 들어 두 개의 손 뻗기 동작을 합성할 때 한 동작은 완전히 뻗은 상태이고, 다른 동작은 뻗었다가 굽힌 상태라면 두 동작은 위상(phase)이 서로 다르다고 본다. 위상이 서로 다른 두 관절 커브들을 합성한다는 것은 별 의미가 없다. 그러함에도 [1]에서는 이 문제를 고려하지 않고 있다.

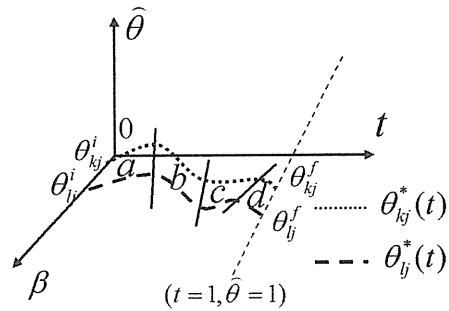


[그림 8] 사용자가 원하는 관절 함수 $\theta_j(t)$

θ_{kj} 와 θ_{lj} 가 동일한 동작군에 속할 경우, 두 커브는 동일한 수의 극점을 가진다. 여기서 극

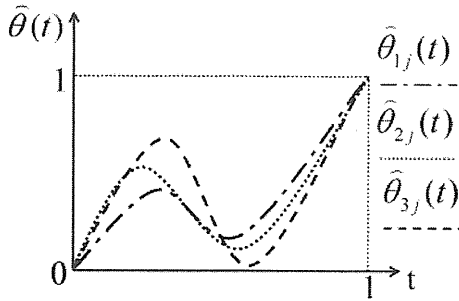
점이라는 것은 속도의 방향이 달라지는 커브들의 분기점이다. 또한 전체 동작을 부분동작으로 나누어줄 수 있는 Key Pose는 극점 상에 위치한다. 일반적으로 같은 동작군에 속한 동작 커브들은 이러한 극점이 서로 유사한 형태로 존재한다. 극점은 동작 데이터를 B-spline curve 를 fitting[11]한 후에 찾게 된다. [그림8]에서 보는 것과 같이 사용자는 생성하기 원하는 관절 커브 $\theta_j(t)$ ($j=1, n$) 가 지나가야 할 각 극점들의 시간과 값을 정해주어야 한다.

관절 커브 $\hat{\theta}_{kj}(t)$ 가 극점을 가지고 있는 경우 극점 사이의 커브를 커브 세그먼트(curve segment)라 한다. 동일 동작군에 속하는 두 관절 커브 $\theta_{kj}^*(t)$ 와 $\theta_{lj}^*(t)$ 의 커브 세그먼트들은 [그림 9]에서 보는 것과 같이 서로 대응되는 위상을 가지고 있다. $\theta_{kj}^*(t)$ 와 $\theta_{lj}^*(t)$ 를 합성할 때 대응되는 커브 세그먼트끼리 그 위상을 대응시켜 합성한다. 본 논문에서는 두 커브 세그먼트의 위상을 대응시키기 위해 커브 세그먼트의 시간을 선형 변환하여 정규화 한다. 다시 말하면 한 커브 세그먼트의 시간(time)을 선형 변환하여 대응되는 커브 세그먼트의 시간(time)과 일치시킨다. 이러한 선형변환으로도 충분한 이유는 극점과 극점사이의 커브는 패턴이 크게 다르지 않기 때문이다.

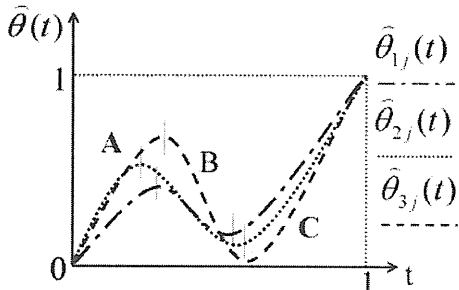


[그림 9] 두 개의 관절 커브 $\theta_{kj}^*(t)$ 와 $\theta_{lj}^*(t)$ 의 위상 대응(a,b,c,d는 대응되는 커브 세그먼트를 나타낸다. β 축은 초기값과 최종값을 나타내는 축, $\hat{\theta}$ 는 정규화된 관절, t 축은 시간을 나타낸다)

위상대응에 대해서 좀더 예를 들어보자. [그림10]은 극점을 가진 정규화 관절각 함수 $\hat{\theta}_{1j}, \hat{\theta}_{2j}, \hat{\theta}_{3j}$ 을 보여준다. 각각의 $\hat{\theta}_{1j}, \hat{\theta}_{2j}, \hat{\theta}_{3j}$ 은 3 개씩의 극점을 가지고 있다. [그림11]은 [그림10] 의 정규화 관절각 함수 $\hat{\theta}_{1j}, \hat{\theta}_{2j}, \hat{\theta}_{3j}$ 가 위상대응을 위하여 각각 A, B, C로 표시된 커브 세그먼트로 분할된 것을 보여준다.



[그림 10] 극점을 가진 3개의 정규화 관절각 함수 $\hat{\theta}_{1j}, \hat{\theta}_{2j}, \hat{\theta}_{3j}$



[그림 11] 세 개의 구간 A, B, C로 나누어진 정규화 관절각 함수 $\hat{\theta}_{1j}, \hat{\theta}_{2j}, \hat{\theta}_{3j}$

이와 같이 정규화 관절 커브 $\hat{\theta}_{kj}(k=1, m)$ 들의 커브 세그먼트들이 추출되면 이에 해당되는 관절곡면 ϕ_j^s (s 는 각 구간)을 2.1절과 2.2절에서 기술한 방법을 적용하여 형성한다. 원하는 관절 커브 $\theta_j(t)$ 는 커브 세그먼트 $\theta_j^s(t)$ 을 연결하여 만든다. $\theta_j(t)$ 는 초기값 θ_j^i 와 최종값 θ_j^f , 그리고 각 극점 θ_j^i ($i < t < f$)을 지나면서 사용자가 입력한 동작 타입의 behavior profile을 가진다.

2.5 동적 시간 변형 방법과의 비교

동적 시간 변형방법[3]와 본 논문은 위상을 대응시킬 두 커브가 전체적으로, 즉 시작과 끝이 서로 대응된다는 전제를 가지고 있다. 그러나 동작 커브들의 위상 대응 문제를 처리하는

방식이 다르다.

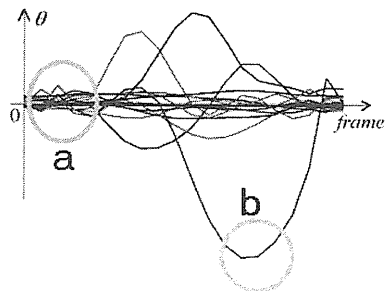
먼저 동적 시간 변형 방법[3]은 예를 들면 2개의 동작 커브 상의 점들을 프레임 수만큼 샘플링하여 두 커브의 “차이”가 최소화하는 방향으로 대응되는 점들의 위치를 바꾼 후에 동작 합성을 한다.

본 논문에서는 이와는 달리 2개의 동작 커브들을 극점에 따라서 여러 개의 커브 세그먼트로 나눈 뒤 각 커브 세그먼트끼리 위상을 대응시킨다. 이 때 한 커브 세그먼트의 시간(time)을 선형 변환하여 대응되는 커브 세그먼트의 시간(time)과 일치시킨다. 이러한 선형변환으로도 충분한 이유는 극점과 극점사이의 커브는 패턴이 크게 다르지 않기 때문이다. 동적 시간 변형 방법[3]은 최적화 알고리즘을 사용하므로 많은 시간이 걸리지만 본 방법은 실시간에 처리 가능하다.

3. 실험

본 논문에서는 2절의 이론에 따라서 인체의 걷는 동작에 실험하였다. 위상을 고려하지 않은 경우와 위상을 고려한 동작 합성을 실험하였다.

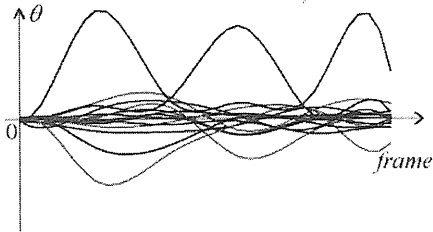
실험 조건은 두 경우 같은 동작군(걷는 동작)의 샘플데이터를 사용하였으며, 사용자가 원하는 초기값과 최종값 또한 동일한 값을 입력하였다. 먼저 [그림12]는 위상 대응성을 고려하지 않고 동작 합성을 한 결과이다. [그림 12]를 자세히 보면 a, b 부분이 심하게 왜곡된 것을 알 수 있다. [그림12]의 동작 커브를 애니메이션하면 걷는 동작과는 전혀 다른 결과를 얻는다.



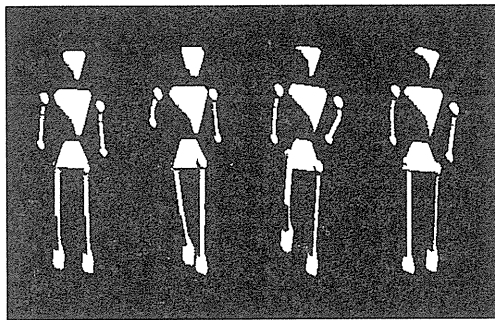
[그림 12] 위상대응을 적용하지 않고 합성한 걷는 동작 (각 곡선 커브는 인체의 각 관절의 변화를 나타낸다)

다음으로 위상 대응성을 고려하였을 때를 실험한 결과는 [그림 13]과 같다. [그림 13]에서 보는 것과 같이 동작 곡선의 모든 구간이 부드럽

러운 커브를 보여준다. [그림13]의 동작 곡선을 이용하여 애니메이션 하면 [그림14]와 같은 결과를 볼 수 있다.



[그림 13] 위상대응을 적용한 후에 합성한 걷는 동작(각 곡선 커브는 인체의 각 관절의 변화를 나타낸다)



[그림 14] 최종 생성된 동작의 연속 화면(좌측에서부터 애니메이션 시작)

결과적으로 위상 대응성을 고려하지 않은 동작 합성의 결과[그림12]와 위상 대응성을 고려한 동작 합성의 결과[그림13]을 비교할 때, 위상대응성을 고려한 방법이 위상을 고려하지 않은 방법보다 자연스러운 동작을 생성함을 알 수 있다.

4. 결론

본 논문에서는 동작 곡선간의 위상 대응성을 고려하여 인체의 걷는 동작에 대한 동작 합성을 하였다[그림13]. 성분동작들을 선형보간하는 방식과는 달리 성분동작들이 이루는 곡면을 생성한 후 이 곡면 상에 존재하면서 주어진 초기값과 최종값을 만족하는 커브를 찾아내는 방식을 제안하였다. 본 논문에서는 곡선 커브들의 극점들을 대응시키는 방법을 사용하여 위상을 대응시킨다. 이 과정이 실시간에 가능하고, 또

한 곡면 생성도 실시간에 가능하므로 실시간 동작 합성이 가능하다.

본 논문에서는 다른 기법에서와 마찬가지로 두 동작의 위상 대응을 관절각 공간(joint space)에서 고려했는데, 이를 3차원 공간에서 고려하는 방법에 대하여 연구할 필요가 있다. 두 동작의 위상이 대응되느냐 어긋나느냐 하는 것은 궁극적으로 3차원 공간에서 판단할 현상이기 때문이다. 초기값과 최종값은 유사하나 세부적으로 다른 동작 곡선들의 동작 합성도 연구해야 될 과제이다.

5. 참고 논문

- [1] 안 효정, 정 문렬, 백 두원. 모션 캡춰 데이터를 이용한 목표지향적 동작 혼합 기법. "컴퓨터 그래픽스 학회 춘계학술대회 논문지", 1997, pp. 1-5.
- [2] 안 효정, 정 문렬, 백 두원. 목표지향적 동작 혼합 기법. "컴퓨터 그래픽스 학회 논문지", Vol. 3, No. 2, December 1997.
- [3] A. Bruderlin and L. Williams. Motion Signal Processing. "Proceedings of SIGGRAPH '95", pp. 97-103.
- [4] D. Dow and S. Semwal. Fast Techniques for Mixing and Control of Motion Units for Human Animation. "Proceedings of Pacific Graphics", 1994, pp. 229-242.
- [5] M. Gleicher. Motion Editing with Space time Constraints. "Proceedings of the Symposium on Interactive 3D Graphics", 1997.
- [6] S. Guo, J. Roberge, and T. Grace. Controlling Movement Using Parametric Frame Space Interpolation. "Computer Animation '93 Proceedings", pp. 216-227.
- [7] C. Rose, B. Guenter, B. Bodenheimer, and M. Cohen. Efficient Generation of Motion Transitions using Spacetime Constraints. "Proceedings of SIGGRAPH '96", pp. 147-154.
- [8] F. Silva, L. Velho, P. Cavalcanti, and J. Gomes. An Architecture for Motion Capture Based Animation. "<http://www.visgraf.impa.br/Projects/mcapture>"
- [9] D. Wiley and J. Hahn. Interpolation Synthesis of Articulated Figure Motion. "IEEE Computer Graphics and Applications", 1997, pp. 39-45.
- [10] A. Witkin and Z. Popovic. Motion

Warping. "*Proceedings of SIGGRAPH '95*", pp. 105-108.

[11] Les Piegl and Wayne Tiller, "*The NURBS Book*", Springer-Verlag, 1995.

[12] Sung-Kwang Moon, Jung-Hee Cho, Moon-Ryul Jung, and Doo-Won Paik, Script-based goal-directed animation system, "*컴퓨터 그래픽스 학회 논문지*", Vol. 2, No. 2, December 1996.

[13] Charles Rose, Michael F. Cohen, and Bobby Bodenheimer. Verbs and Adverbs: Multidimensional Motion Interpolation. "*IEEE Computer Graphics and Applications*", September/October, 1998, pp.32-40.