# 자연스러운 실시간 동작 전이 생성을 위한 균등 자세 지도 알고리즘

## (Uniform Posture Map Algorithm to Generate Natural Motion Transitions in Real-time)

이 범 로 † 정 진 현 ††
(Bum Ro Lee) (Chin Hyun Chung)

**요 약** 동작 포착 시스템에 의해 기록된 동작 데이타를 재사용하는 것은 비용 절감이나 작업과정의 효율성 증대를 위해 매우 중요한 기술이다. 그러나 기록된 데이타의 동작 곡선이 제어점을 가지고 있지 않기 때문에 상호작용을 통한 동작 데이타의 편집이 쉽지 않아서 데이타의 재사용에 어려움이 있다. 기존 동작 데이타를 재사용하는 기술로서 많은 학자들은 조각 동작(Clip motion)들을 부드럽게 연결하여 새로운 동작을 만들어 내는 동작 전이 기술을 제안하고 있다. 본 논문에서는 이러한 동작 전이 기술의 구현 방법으로 균등 자세 지도(Uniform Posture Map:UPM)알고리즘을 제안한다. 학습 단계에서 UPM은 다관절체의 다양한 자세들을 비감독 경쟁 학습을 통해 양자화한다. 이 단계에서 서로 유사한 자세를 나타내는 출력 뉴런을 기하학적으로 근접한 위치에 배치해서 다관절체의 전체 동작 지도를 생성한다. 생성된 UPM의 이러한 특징을 이용해서 적용된 두 동작의 중간 자세를 만들어 내고, 이 자세를 전체 중간 동작을 만들어 내는 키 프레임으로 사용한다. 많은 계산량이 요구되며, 결과 동작을 제어하기가 어려운 다른 동작 전이 알고리즘들과 비교하여 UPM 알고리즘은 중간 동작 생성에 상대적으로 적은 계산량을 요구하며, 하나의 변수를 이용하여 생성된 동작을 제어할 수 있어서 편리한 상호작용 작업 환경을 제공한다. 특히 자기 조직 지도(Self-Organizing Map:SOM) 알고리즘을 이용해 자세 지도를 생성할 때, 실제로 존재하지 않은 자세가 포함될 수 있는 가능성을 학습 단계에서 제거함으로써 자세 생성에 있어서 안정성을 확보할 수 있다. 이로 인해 선형 보간법에 비해서 실제 동작에 가까운 동작 곡선을 생성함으로써 보다 자연스러운 동작을 만들어 낼 수 있다. 본 논문에서 제안된 동작 전이 기법은 삼차원 애니메이션 제작, 삼차원 게임, 가상 현실 등의 다양한 분야에 유용하게 적용될 수 있다.

***Abstract*** It is important to reuse existing motion capture data for reduction of the animation producing cost as well as efficiency of producing process. Because its motion curve has no control point, however, it is difficult to modify the captured data interactively. The motion transition is a useful method to reuse the existing motion data. It generates a seamless intermediate motion with two short motion sequences. In this paper, Uniform Posture Map (UPM) algorithm is proposed to perform the motion transition. Since the UPM is organized through quantization of various postures with an unsupervised learning algorithm, it places the output neurons with similar posture in adjacent position. Using this property, an intermediate posture of two active postures is generated; the generating posture is used as a key-frame to make an interpolating motion. The UPM algorithm needs much less computational cost, in comparison with other motion transition algorithms. It provides a control parameter; an animator could control the motion simply by adjusting the parameter. These merits of the UPM make an animator to produce the animation interactively. The UPM algorithm prevents from generating an unreal posture in learning phase. It not only makes more realistic motion curves, but also contributes to making more natural motions. The motion transition algorithm proposed in this paper could be applied to the various fields such as real time 3D games, virtual reality applications, web 3D applications, and etc.

· 이 논문은 2000년도 광운대학교 교내학술연구비 지원에 의해 연구되었음.
† 비 회 원 · 광운대학교 제어계측공학과
redcom@crcval.net
†† 정 회 원 광운대학교 정보제어공학과 교수
chung@daisy.kwangwoon.ac.kr
논문접수 · 2001년 7월 25일
심사완료 · 2001년 9월 27일

## 1. 서 론

산업 전반에 걸친 컴퓨터 시스템의 도입은 단순 반복 작업의 자동화를 통해서 작업의 효율을 극대화하고 작

업자가 작업 자체에 집중할 수 있도록 하는데 기여하였다. 디지털 애니메이션을 제작하는 과정에서도 과거 제작자들이 각 프레임을 손으로 일일이 그려서 작업 하던 것을 컴퓨터 시스템을 이용한 키 프레임 기법을 도입함으로써 작업자들이 보다 창조적인 작업에 시간을 투입할 수 있도록 해 주었다. 그러나 이러한 방식에서는 애니메이션 내부의 모든 동적 특성들을 제작자들의 경험과 직관에 의존하여 사실적인 동적 특성을 표현하는데 한계를 갖기 때문에, 캐릭터의 과장된 표현이 요구되는 애니메이션 등의 분야에 적합할 수 있지만 현실감이 요구되는 VR(virtual reality) 등의 분야에는 적합하지 못한 측면이 있다. 이러한 문제에 대한 대안으로 동작 포착 시스템(motion capture system)이 도입되었다. 동작 포착 시스템은 인간을 비롯한 다관절체에 부착된 위치 센서의 출력을 기록하는 시스템으로서 실시간에 다관절체의 동적특성을 정확히 포착할 수 있어서 키 프레임 방식에 비해 현실감 있는 애니메이션을 보다 저렴한 비용으로 제작할 수 있는 장점을 가진다. 그러나 동작 포착 데이타의 경우에 키 프레임 방식의 경우처럼 동작 곡선에 제어점을 포함하고 있지 않아서 상호작용을 통한 동작을 편집할 수 없다. 따라서 미세한 차이를 가지는 유사 동작들을 편집하여 재사용하지 못하고 필요한 모든 동작을 새로 포착하여야 한다는 비효율성을 가진다. 다양한 관점에서 포착된 동작 데이타의 편집에 대한 연구가 진행되고 있는 이유가 여기에 있다[1][2][3][4]. 이미 포착된 동작들을 편집해서 재사용할 수 있다면, 엄청난 제작 비용절감 효과를 가져올 수 있을 뿐만 아니라, 제작 효율성 향상에도 많은 기여를 할 수 있다. 다양한 동작 편집 기법 중에서 본 논문에서는 동작 전이 방식을 이용한다. 동작 전이란 짧은 단위 동작들로 구성된 동작 데이타베이스를 구성하고, 동작 데이타베이스에서 추출된 두개의 동작과 새롭게 생성된 중간 동작을 자연스럽게 연결하여 새로운 동작을 만들어 내는 동작 편집 기법이다. Rose[5]는 중간 동작을 생성하기 위해서 비선형 프로그래밍 기법을 도입하였다. 비선형 프로그래밍은 제한 조건을 가지는 비선형 목적 함수가 최소값을 가지게 하는 해를 구하는 지법이다. 일반적으로 동작 전이의 경우에는 목적함수로서 동작의 에너지 함수를 사용하여 동작의 에너지를 최소화 시키는 해를 구한다. 그러나 다관절체의 실제 행동 방식이 반드시 에너지를 최소화하는 방향으로 이루어지는 것은 아니기 때문에 비선형 프로그래밍 방식은 자연스럽지 못한 동작을 생성해 낼 가능성이 있다. 또한 많은 계산량이 요구되고 알고리즘 자체에 작업자가 직관적으로 결과 동작을 제어할 수 있는 매개변수에 대한 고려가 없어서 실시간 상호작용을 통한 동작 전이에는 적합하지 않다. 본 논문에서는 다관절체의 동작의 경향을 나타내는 균등 자세 지도 (Uniform Posture Map: UPM)을 이용한 동작 전이 기법을 제안한다. 제안된 기법에서는 동작 전이를 실제 동작에 의해 학습된 UPM에 근거해서 합성하기 때문에 보다 사실적인 중간 동작을 만들어 낼 수 있다. 또한 제안된 방법이 학습 단계에서는 다소 많은 계산량을 요구하지만 일단 UPM이 작성된 이후에는 동작 생성 단계에서 더 이상의 학습과정이 필요 없기 때문에 빠른 처리가 가능하다. 또한 생성되는 동작을 하나의 매개변수를 이용해서 직관적으로 조절할 수 있기 때문에 사용자 상호작용을 통한 실시간 동작 제어에 매우 적합하다. 적용되는 다관절체의 자세는 각 관절의 DOF(Degree of Freedom) 값으로 이루어진 벡터로 나타내며, 학습과 중간 자세 생성 과정의 계산 부하를 줄이기 위해 각 관절의 특성에 따라 자세 벡터를 4개의 벡터로 분할하여 각각 별도의 학습, 생성과정에 적용하였다. 생성된 중간 자세는 중간 동작 생성 과정에서 키 프레임으로 사용된다. 본 논문에서는 먼저 사용된 골격 모델과 특성에 따라 분류된 관절 클래스를 통한 동작 전이 수행의 효율성에 대해서 설명한다. 다음으로 동작 전이에 이용되는 UPM과 유사 알고리즘과의 장단점을 비교하고, 실제로 학습된 UPM 알고리즘을 이용한 동작 전이 실험과 결과에 대한 분석 등을 기술하였다.

## 2. 균등 자세 지도

### 2.1 골격 모델

동작 데이타를 시스템에 적용하기 전에 미리 설정된 모델에 적용할 수 있도록 동작 데이타에서 필요 이상의 DOF를 제거하는 선처리 작업이 필요하다. 이는 학습이나 합성과정의 계산량을 줄여 전체 과정의 효율성을 높이는 긍정적인 역할을 한다. 본 논문에서는 전체 동작에 주된 영향을 미치는 21개의 관절로 이루어진 모델을 사용한다. 대부분의 상용 동작 포착 시스템에 의해 생성된 동작 포착 데이타는 DOF의 생략을 통해 설정된 모델에 적용시킬 수 있다. 표 1은 모델을 구성하고 있는 각 관절의 명칭과 그것이 전체 자세에 미치는 영향을 분석하였고, 그림 1에서는 본 논문에서 사용된 골격 모델을 표시하였다. 루트는 관절로 분류되지 않는 단지 3개의 회전(rotation) DOF와 3개의 병진(translation) DOF를 가지는 하나의 점이며, 또한 설정된 모델에서 유일한 병진 DOF를 가지는 요소이기도 하다. 또한 전체 자세의 결정에 가장 큰 영향을 미치며, 관절과 상이한 특성을 가

지고 있으므로 UPM 알고리즘을 이용하지 않고 별도의 알고리즘을 통해 처리한다.

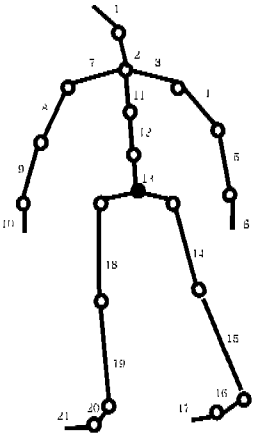


그림 1 골격 모델

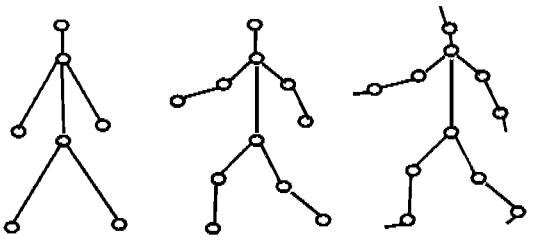
표 1 각 관절과 전체 자세에 미치는 영향

| No | Name | Effects on posture | No | Name | Effects on posture |
|---|---|---|---|---|---|
| 0 | root | Very Large | 11 | thorax | Large |
| 1 | head | Medium | 12 | upperback | Large |
| 2 | lowerneck | Large | 13 | lowerback | Large |
| 3 | lshoulderjoint | Large | 14 | lfemur | Large |
| 4 | lhumerus | Large | 15 | ltibia | Medium |
| 5 | lradius | Medium | 16 | lfoot | Small |
| 6 | lhand | Small | 17 | ltoes | Small |
| 7 | rshoulderjoint | Large | 18 | rfemur | Large |
| 8 | rhumerus | Large | 19 | rtibia | Medium |
| 9 | rradius | Medium | 20 | rfoot | Small |
| 10 | rhand | Small | 21 | rtoes | Small |

* The name of symmetric bones has a prefix to specify their location, 'r-' or 'l-'

### 2.2 다중 클래스 자세 공간

인체와 같은 다관절체의 자세는 각 관절의 DOF 값에 의해 표현된다. 각 관절은 두개의 뼈대 사이에 위치하고, 두 뼈대는 전체 계층 내에서 부모-자식 관계를 가진다. 일반적으로 관절과 루트 사이의 거리가 전체 자세에 관절의 변화가 미치는 영향의 정도를 결정한다. 다시 말하면 루트와 관절 사이의 거리가 가까우면 가까울수록 전체 자세에 미치는 영향은 커지고, 단말쪽의 뼈대일수록 전체 자세에 미치는 영향력은 미약하다. 그림 2는 전체 골격에 단말측의 뼈대가 추가되면서 전체 동작의 묘사가 세밀하게 변화됨을 보인다. 그러나 대략적인 자세는 이미 그림 2-(a)에서 결정되고, 그림 2-(b)와 그림 2-(c)는 더욱 세밀한 묘사가 이루질 뿐 전체적인 자세의 윤곽은 변하지 않는다. 따라서 루트와 가까운 관절들의 DOF 값에 의해 전체 자세의 윤곽이 정해짐을 알 수 있으며, 이러한 특성들에 따라서 각 관절들을 몇 개의 클래스로 분류하고, 학습과 합성 과정에서 분류된 클래스에 속한 관절의 DOF 값들을 클래스별로 처리하게 되면 데이타 처리에 매우 효과적이다. 이 방법은 다해상도 화상 압축과 유사한 기법으로, 다해상도 화상 압축에서는 화상 신호가 가지고 있는 주파수의 범위에 따라 신호를 분류하고 분류된 신호를 대상으로 독립적인 압축을 수행하여 효율적인 화상 압축을 가능하게 한다. 이러한 접근 방식은 신호간의 상호 관계(correlation)를 이용하여 신호를 효과적으로 처리할 수 있다는 이론에 바탕을 두고 있다[6].



(a) 대략적인 묘사 (b) 중간 단계 묘사 (c) 세부적인 묘사

그림 2 세부 관절 추가에 따른 다관절체의 자세 묘사 단계

본 논문에서는 각 관절의 변화 범위와 전체 자세에 미치는 영향의 정도에 따라 모든 관절을 4개의 클래스로 분류하고, 각 클래스에 속한 관절의 DOF 값들로 이루어진 부분 자세 벡터를 정의 한다. 표 2에서 각 클래스에 속한 관절들을 나타내고 있다. Body 클래스는 전체 골격의 척추에 해당되는 클래스로서 5개의 관절을 포함하고 있다. 상대적으로 작은 변화 범위를 가지고 있으나, DOF의 변화량에 따르는 전체 자세에 대한 영향력은 가장 큰 클래스이다. Global 클래스는 5개의 관절을 포함하고 있으며, 큰 변화 범위를 가지고 전체 자세에 비교적 큰 영향을 끼친다. Medium 클래스도 5개의 관절을 포함하고 있으며, 가장 큰 변화 범위를 가지고 전체 자세에는 중간 정도의 영향을 끼친다. Detail 클래스는 작은 변화 범위의 DOF로 구성되며, 전체 자세에 미치는 영향력도 크지 않다.

표 2 관절 클래스

| Class name | Body | Global | Medium | Detail |
|---|---|---|---|---|
| Bone name | lshoulderjoint | lowerneck | head | lfoot |
| | rshoulderjoint | lhumerus | ltibia | rfoot |
| | thorax | rhumerus | rtibia | ltoes |
| | upperback | lfemur | lradius | rtoes |
| | lowerback | rfemur | rradius | lhand |
| | | | | rhand |

### 2.3 균등 자세 지도

동작 합성이나 동작 전이와 같은 동작 신호 처리 분야에 있어서 가장 많이 사용되는 알고리즘은 Sequential Quadratic programming이나 BFGS와 같은 비선형 프로그래밍(Non-linear Programming: NLP)이며, 많은 학자들은 NLP를 이용한 시공 제약 조건(spacetime constraints)[7] 기법을 사용자 상호작용에 의한 동작 편집 방법으로 제안하여 왔다. 대부분 기존의 동작 데이타를 변형하기 위한 연구들이었고, NLP에 적용되는 변수는 동작 데이타의 동작 곡선의 제어점이 대부분이었는데, 1996년 Rose[5][8]가 시공 제약 조건을 이용한 NLP 알고리즘으로 포착된 동작 데이타들을 연결하여 새로운 동작을 생성하는 동작 전이 기법을 처음으로 제안 하였다. 그러나 이러한 접근 방법은 몇 가지 문제점을 수반하는데, 첫번째가 계산량에 대한 문제[9][10] 이다. 본 논문에서는 21개의 관절로 정의되는 모델을 사용하는데, 한 관절이 3 DOF를 가지므로 하나의 자세를 표현하는데 63개의 인자를 가지는 DOF 벡터가 사용된다. 따라서 적절한 해를 찾기 위해서 63차원의 에러 공간에 대한 검색과 63차원의 목적 함수를 각 반복 단계마다 매번 계산을 해야 한다. 실제로 알고리즘에 따라 약간의 차이는 있지만 일반적으로 NLP는 많게는 $O(n^3)$에서 적게는 $O(n^2)$에 해당되는 계산 로드를 가지는 것이 일반적이다[8][10]. 따라서 대상체가 63차의 고차원의 벡터일 경우 엄청난 계산량을 필요하고, 이것은 실시간 처리되는 사용자 상호작용 시스템에 있어서 심각한 문제점이라 할 수 있다. 또 다른 문제점은 NLP 기법에서는 항상 목적 함수로 설정된 각 DOF의 에너지 함수를 최소화하는 해를 구한다는 것이다[11]. 그러나 자신의 내부적인 근육의 힘에 의해 자발적으로 움직이는 다관절체가 항상 자신의 동작 에너지 값을 최소화하는 방향으로 움직이는 것은 아니다. 따라서 구해진 해에 대응되는 결과 동작이 자칫 실세계에 존재하지 않는 부자연스러운 동작이 될 가능성이 있다. 특히 발레와 같이 의도적으로 안무된 동작을 대상으로 할 경우 이러한 문제점은 더욱 두드러질 수 있다. 세번째로 동작 제어의 편의성에 관한 문제로서 작업자가 상호작용을 통해 동작을 제어하기 위해서는 직관적이고도 예측 가능한 제어 변수가 필요하다. NLP의 경우에는 이러한 제어 변수를 갖지 않아 직관적인 제어에 어려움이 있다. Kohonen의 자기 조직 지도 (self-organizing map)[12][13]는 이러한 문제점들에 대한 해결에 적합한 비감독 학습 네트워크이다. 입력 뉴런과 출력 뉴런 그리고 양측의 뉴런을 연결하는 연결 강도로 구성되며, 학습과정 중에 입력된 정보들의 경향을 반영하여 특성 지도(feature map)를 조직한다. 자기 조직 지도의 학습 과정에 다관절체의 동작 데이타를 입력 패턴으로 사용한다면 자기 조직 지도의 특성 지도는 다관절체의 동작 패턴을 반영할 것이고, 이는 자연스러운 동작 생성에 기여할 수 있다. 그림 3은 기본적인 자기 조직 지도의 기본 구조를 나타낸다.

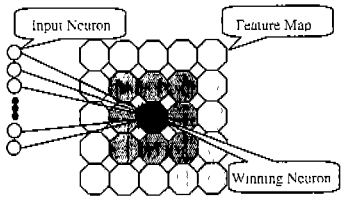
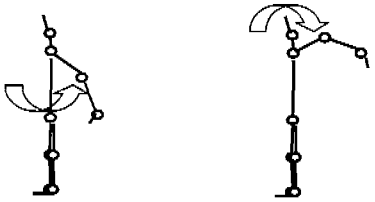



그림 3 자기 조직 지도



(a) 반시계 방향 회전 (b) 시계 방향 회전

그림 4 인체의 기계적 제약 조건의 예

일반적으로 자기 조직 지도는 샘플 벡터들로 구성된 샘플 공간을 정확하게 반영하며, 상대적으로 빠른 학습 시간을 나타내지만 샘플 공간이 매우 복잡한 모양을 가질 경우 샘플 공간을 적절히 반영하지 못하는 단점을 가진다. 특히 샘플 공간이 오목한(non-convex) 부분을 가질 경우에는 샘플 공간 밖의 유효하지 않은 공간에 출력 뉴런이 위치하는 경우가 간혹 발생하게 된다. 이것은 자기 조직 지도가 출력 뉴런간에 근접 관계를 유지하고자 하는 특성에 기인한다[12][13]. 대부분의 응용

분야에서 이런 출력 뉴런은 발생 빈도가 적고, 전체 결과에 크게 영향을 미치지 않기 때문에 무시될 수 있다. 그러나 인간과 같이 실존하는 다관절체는 각 관절에 물리적인 제한이 존재하기 때문에 DOF 값으로 이루어진 벡터 공간상에 샘플 영역은 오목한 부분이 있을 수 있고 이것은 실존하는 두 자세를 토대로 산술적으로 계산된 중간 자세가 존재하지 않을 수 있다는 의미이다. 그림 4는 인체의 물리적 제약의 예를 보인다. 그림 4의 (a)는 반시계 방향으로 회전하는 팔을 나타내고, 그림 4의 (b)는 시계방향으로 회전하는 팔을 나타낸다. (a)와 (b)의 중간 자세는 (a)보다 반시계 방향으로 더 회전한 형태가 되어야 하지만 어깨 관절이 더 이상 회전하지 않기 때문에 실제 두 모션의 중간값은 정상적인 상황에서 존재하지 않는다. 이러한 예는 인간의 자세를 표현해 주는 DOF 값으로 구성된 벡터들의 집합이 오목하다는 것을 증명한다. 따라서 인체 골격을 모델로 하는 자세 교정 시스템이나 의료 시스템 또는 캐릭터 애니메이션 등의 분야에서 이러한 뉴런이 결과에 상당한 악영향을 미칠 수 있다. 또한 자기 조직 지도의 학습 과정에서는 많은 매개 변수들의 값을 결정해야 하지만 이러한 변수들을 적절하게 결정하기 위해서는 학습과 생성된 자기 조직 지도를 조사해야 하는 몇 번의 시행착오 과정이 요구된다. 이는 요구되는 많은 변수들을 결정하는 유효한 규칙들이 없기 때문이다. 이러한 자기 조직 지도의 문제들을 해결하기 위해서 본 논문에서는 균등 자세 지도를 제안한다. 초기 연결 강도를 난수로 초기화 하고, 그것을 바탕으로 지도를 구성하는 자기 조직 지도와는 달리 UPM에서는 주어진 입력 샘플 벡터의 정보를 바탕으로 지도 구조를 결정한다 이 과정에서 미리 결정되어야 하는 매개변수는 근접도의 반지름 값과 출력 뉴런의 최대 허용 반지름 값 뿐이다. 다음은 UPM 알고리즘에 대한 설명이다.

1. 초기화
   - 출력 뉴런의 최대 허용 반지름 R을 초기화 한다.
   - 초기 근접도(neighborhood)를 초기화 한다.
   - 처음 입력된 샘플 벡터를 첫번째 출력 뉴런의 연결 강도로 세팅한다.
2. 입력된 샘플 벡터와 모든 출력 뉴런과의 거리가 R을 초과하는지 시험한다.
   - 모든 출력 뉴런과의 거리가 R을 초과하는 샘플에 대해서는 새로운 뉴런을 추가하고, 그에 대한 연결 강도로 세팅한다.
   - 하나 이상의 출력 뉴런이 거리 R이내에 존재하면 3번 단계로 진행한다.
3. 활성 출력 뉴런 계산
   - 활성 출력 뉴런 $O_j$는 다음 식에 의해 계산된다.

$$O_j - F_{\min}(d_j) = F_{\min}(\sum_i (X_i - W_{ji})^2)$$

여기서 $F_{mn}$는 단위 함수로서 j 번째 출력 뉴런이 최소의 활성도 함수 $d_j$ 값을 가지는 경우 그 출력 뉴런과 인접 출력 뉴런에 대해서는 1을 출력 시키고, 나머지 출력 뉴런에 대해서는 0을 출력한다.

4. 연결강도 학습
   - 활성화된 출력 뉴런에 대한 연결 강도를 다음 식에 의거하여 갱신 시킨다.

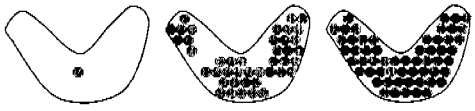
$$\Delta W_{ji} = O_j \eta (x_i - W_{ji})$$

여기서 $\eta$는 학습이 반복됨에 따라 감소하는 학습율이다. $(0 < \eta < 1)$

   - 전체 입력 샘플에 대한 학습 과정을 마친 후 학습률 $\eta$과 근접도를 감소시킨다.
   - $\eta \rightarrow 0$이 될 때까지 위 과정을 반복 한다.

그림 5는 자기 조직 지도와 UPM이 가지는 지도 학습 방식의 차이를 나타낸다.

(a) 자기 조직 지도의 학습 방식



(b) UPM의 학습 방식

그림 5 학습 방식의 차이

그림 5에 나타난 바와 같이, UPM의 경우에는 정해진 출력 뉴런의 수가 없다. 만약 입력 샘플에 대해 특정 거리 내에 어떤 출력 뉴런도 존재하지 않을 경우, 기존 출력 뉴런의 연결강도를 변화시키지 않고 새로운 출력 뉴런을 UMP에 추가한 후 그때 적용된 입력 샘플로 연결강도를 설정한다. 그러므로 오목한 구역에서 무리하게 기존 출력 뉴런의 구조를 변형시켜 유효하지 않는 범위에 출력 뉴런이 위치하게 되어 야기되는 자기 조직 지도의 문제점을 개선할 수 있다. 그리고 자기 조직 지도에서 미리 정해야 하는 많은 매개변수를 자동으로 결정하여 학습 과정의 시행 착오를 대폭 감소 시킨다. 또한 일단 생성된 UPM을 이용해 중간 동작을 계산하는 과정 중에도 입력된 두 벡터와의 거리, 그리고 특정 지점

에서 대한 각 출력 뉴런까지의 거리 비교 연산만을 필요로 하기 때문에 단지 $O(n)$에 해당하는 계산량만이 요구된다. 따라서 계산량을 상당히 감소시킬 수 있는 장점을 가진다. 그러나 UPM은 학습의 결과가 학습 과정에서 입력되는 샘플에 종속된다는 문제점을 가지고 있다. 따라서 입력 샘플의 적절한 선별이 우수한 결과를 만들어 내는 중요한 요소로 고려 되어야 한다.

## 3. UPM을 이용한 동작 전이 시스템

자기 조직 지도를 기반으로 설계된 UPM의 두 출력 뉴런의 기하학적 거리는 출력 뉴런간의 유사성 정도와 매우 깊은 관련이 있으며, 이러한 특성을 이용해서 UPM을 동작 전이에 응용할 수 있다. 서로 다른 두 동작의 전이를 위해서는 적절한 중간 동작을 생성하고 이를 이용하여 두 동작을 자연스럽게 연결시키는 과정이 필요하다. 본 논문에서는 이 과정을 통해 전후 동작의 특성을 반영하는 중간 자세를 생성하고 전후 동작의 정보를 함께 적용하여 B-스플라인 기법을 이용해서 두 동작을 연결하는 방식으로 처리하였다. 이는 애니메이션 제작자가 키 프레임 방식의 애니메이션을 제작하는 과정과 매우 유사하므로, 본 논문에서 제안한 알고리즘은 수작업으로 키 프레임을 잡는 애니메이션 제작자의 역할을 대신하는 것이라 할 수 있다.

### 3.1 중간 자세 생성

그림 6과 같이 일반적인 응용 분야에서 비감독 학습 기반 신경망의 출력값을 찾고자 할 때, 입력에 대한 모든 출력 뉴런과의 기하학적 거리를 계산하고 그 중 가장 작은 거리를 갖는 출력 뉴런의 연결 강도를 알아내는 것이 일반적이다. UPM의 경우에는 두개의 입력과 하나의 출력을 가지며 사용 목적이 다르기 때문에 새로운 출력 방식이 필요하다.

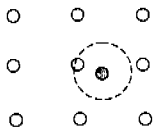


그림 6 일반적 출력 방식

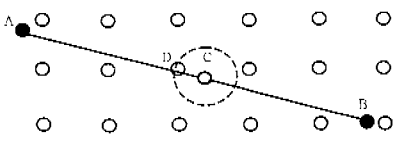
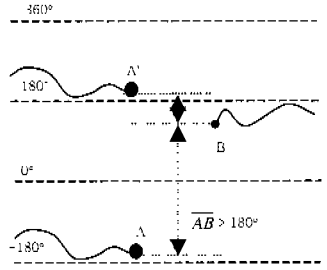
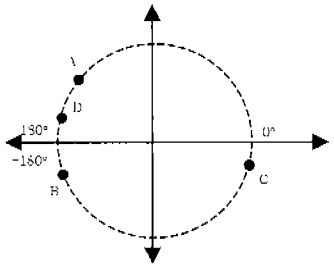



그림 7 중간 자세 출력 방식

그림 7은 UPM으로부터 입력된 두 자세에 대한 중간 자세를 찾아내는 과정을 설명하고 있다. 먼저 두 입력 벡터 A와 B를 UPM의 해당 위치에 대응 시키고, 두 입력 벡터의 중간 위치에 있는 벡터 C를 계산한다. 다음으로 기하학적으로 벡터 C와 가장 가까이 위치하고 있는 뉴런 D를 구할 수 있다. 이 단계에서 최종적인 결과값을 사용자의 의도에 따라 제어하고자 한다면 와 의 길이 비율을 조정하기만 하면 된다. 그에 따라 두 벡터에 대해서 서로 다른 합성비를 가지는 새로운 출력 뉴런이 결정 되며, 따라서 원하는 결과들을 얻을 수 있게 된다. 그러므로 상호작용을 통해 작업자가 원하는 자세를 선택할 수 있다.




(a) 동작 곡선의 표현




(b) 중복된 중간점

그림 8 중간점 선택의 문제

일반적으로 DOF 값은 180°에서 180°사이의 범위를 가진다. DOF 값이 180°를 초과하게 되면 360° 감소시켜 정규화를 하고, 같은 방법으로 -180° 이하의 DOF 값을 가진다면 360°를 더해서 정규화를 한다. 이런 DOF의 표현 방식은 동작 곡선의 불연속성을 초래하게 되므로 동작 전이 처리를 수행할 때에는 이 문제에 대한 고려가 필요하다. 만약 두 동작 곡선이 그림 8의 (a)와 같고 동작 전이를 하고자 하는 두 단말점 A와 B의 차이가 180°를 초과한다면 그림 8의 (b)에서 보인 것과

같이 중간점으로 점 C 대신 실질적인 중간점인 점 D를 선택해야 한다. 그렇지 않으면 결과 동작이 부자연스러운 특성을 보이게 된다. 이러한 목적으로 사용된 알고리즘은 표 3과 같다.

표 3 중간점 선택 알고리즘

```
double AngularMedium(double dRatio, double dA,
double dB) {
    double dMedium;
    if (fabs(dA-dB) > 180) {
        if (dA < dB)
            dA += 360;
        else
            dB += 360;
    }
    dMedium = dA * dRatio + dB * (1-dRatio);
    if (dMedium > 180)
            dMedium -= 360;
    return dMedium;
}
```

전체적인 중간 자세 생성 알고리즘은 다음과 같이 정리할 수 있다.

1. 입력 DOF 벡터를 미리 정해놓은 클래스 분류에 따라 각 클래스에 속하는 4개의 부분 벡터로 나눈다.
2. 각 부분 벡터들을 같은 클래스에 속한 부분 UPM에 적용하여 중간 자세 생성과정을 시작한다.
3. 합성비 $\alpha$를 정의한다. 이것은 두 입력 벡터 중, 합성된 최종 중간 자세에 어느 자세가 더 영향을 주는지를 결정하는 매개 변수이다.
4. $\overline{AB}$ 위에서 합성비 $\alpha$를 만족하는 점 $C$를 찾는다.
5. 점 $C$에서 최소 거리를 가지는 출력 뉴런을 UPM에서 찾는다.
6. 4개의 결과값을 조합하여 전체 중간 자세를 만든다.

### 3.2 동작 합성

중간 자세의 생성이 끝나면 생성된 중간자세를 이용해서 동작을 합성하는 과정이 수행된다. 우선 루트에 대한 고려를 먼저 해야 하는데, UPM에는 루트 동작에 대한 어떠한 정보도 가지고 있지 않기 때문에 별도의 알고리즘을 이용해서 루트의 동작을 결정해야 한다. 루트는 6개의 DOF값을 가지며, 이는 3개의 병진운동 DOF와 3개의 회전운동 DOF로 이루어진다. 회전운동에 대한 처리에는 선형 보간법과 B-스플라인을 이용한다. 병진운동의 Y축 성분은 일반적으로 나타나는 동작 요소 중에서 적은 변화량을 가지는 것이기 때문에 회전운동과 동일한 방법으로 처리한다. X와 Z축의 변화량은 ROSE[5]가 제안한 알고리즘을 변경하여 사용한다. X와 Z축의 병진운동 변화량을 계산하는 식은 다음과 같이 정의된다.

$$f(t)=\begin{cases} t<t_1 & : p(t)=l(t) \\ t\le t_2, t\ge t_1 & : p(t)=p(t_1)+m(t) \\ t>t_2 & : p(t)=p(t_2)+n(t) \end{cases}$$

$$m(t)=\sum_{\alpha=t_1}^{t}\left\{v_1\left(1-\frac{\alpha-t_1}{t_2-t_1}\right)+v_2\left(\frac{\alpha-t_1}{t_2-t_1}\right)\right\},$$

$$n(t)=\sum_{\alpha=t_2}^{t}v_2(\alpha-t_2)$$

여기서 $l(t)$는 선행 동작의 루트 궤적이고, $t_1$은 선행 동작의 마지막 프레임에 대한 시간이며, $t_2$는 후속 동작의 첫번째 프레임에 대한 시간이다. $p(t_1), p(t_2)$는 $t_1, t_2$일 때의 좌표를 나타낸다. 루트 이외에 다른 관절에 대한 중간 동작은 UPM에 의해 생성된 중간 자세를 키 프레임으로 하여 B 스플라인 기법을 통해 생성된다.

## 4. 실험 및 결과

### 4.1 실험

그림 9과 그림 10에 나타낸 바와 같이 동작 전이 과정은 모두 3단계로 이루어 지는데 각 과정은 데이타 전처리 단계, UPM 학습 단계 그리고 동작 전이 단계로 구성된다. 데이타 전처리 단계는 광학 동작 포착 시스템을 이용해서 얻어진 동작 데이타를 설정된 모델에 적용할 수 있도록 수정, 처리하는 단계이다. 광학 동작 포착 시스템에서 얻어진 데이타는 각 광학 표시기(Optical Marker)들의 3차원 위치값을 의미하며, 이 값을 각 관절의 DOF값으로 변환시키고, 변환된 값들 중에서 불필요한 값들을 삭제한다. UPM 학습 단계에서는 각 DOF

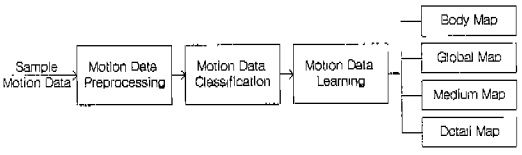



그림 9 전처리 과정과 학습 과정

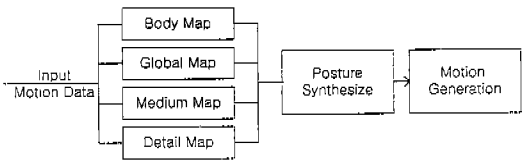


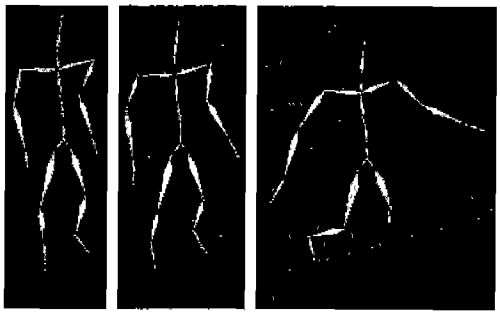

그림 10 동작 전이 과정

벡터를 각 관절이 속한 클래스에 따라 body, global, medium 그리고 detail 클래스에 속한 4개의 부분 벡터로 분해하고 각 클래스에 대한 4개의 부분 UPM을 별도로 생성한다. 일단 한번 생성된 UPM은 재학습 과정 없이 동작 전이 처리에 재사용할 수 있다. 동작 합성 단계에서는 입력된 두 자세의 특성을 동시에 가지는 중간 자세를 생성한다. 이 과정에서도 입력 벡터를 4개 클래스의 부분 벡터로 분해하고 각각을 부분 UPM에 적용하여 결과값을 얻은 후에 이를 결합하여 전체 자세를 만든다. 생성된 자세는 B-스플라인을 이용한 동작 생성 과정을 거쳐 두 동작의 전이 동작을 얻을 수 있다.

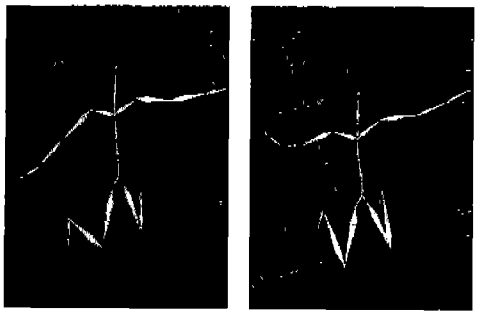
본 실험에 사용될 동작 데이타를 얻기 위해 VICON 광학 동작 포착 시스템을 사용하였다. 또한 여분의 DOF 삭제, 걸과 동작의 랜더링 및 필요한 결과 출력을 위해 FilmBox 를 사용하였다. 실험에는 초당 120 프레임, 전체 1800 프레임으로 구성된 9개의 동작 클립을 200프레임씩 나누어 사용하였으며, UPM의 학습, 합성 알고리즘을 구현하기 위해서 MS Visual C++를 사용하였다.

### 4.2 결과 분석

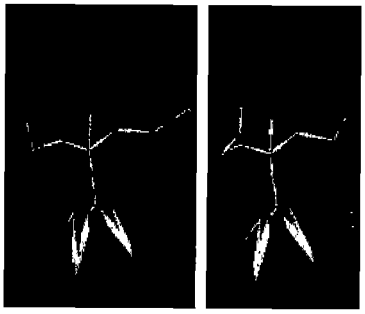
제안된 알고리즘을 다양한 조각 동작에 적용한 결과 중에서 두 가지 걸과는 아래 그림 11, 그림 12과 같다. 적용된 조각 동작은 200 프레임의 길이를 가지며, 20



(a) 195 프레임 (b) 200 프레임 (c) 205프레임

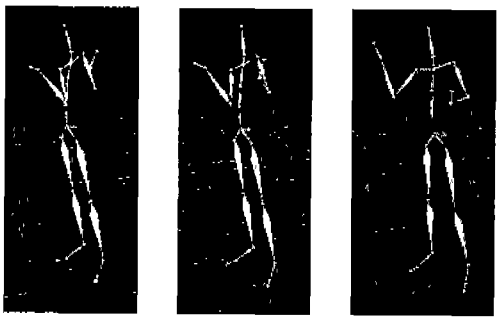


(d) 210 프레임 (e) 215 프레임

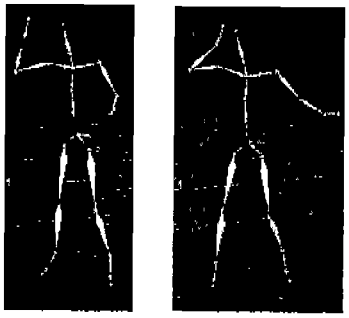


(f) 220 프레임 (g) 225 프레임

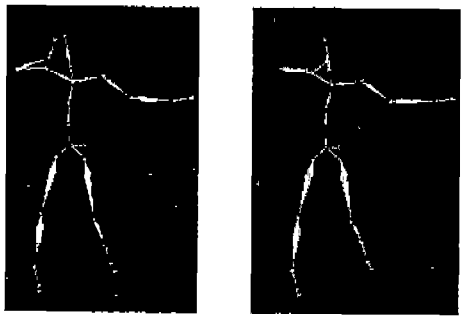
그림 11 동작 전의 춤



(a) 195 프레임 (b) 200 프레임 (c) 205 프레임



(d) 210 프레임 (e) 215 프레임

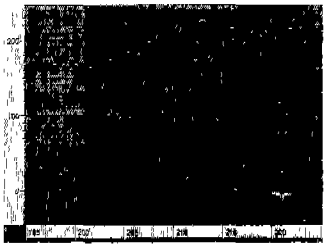


(f) 220 프레임 (g) 225 프레임

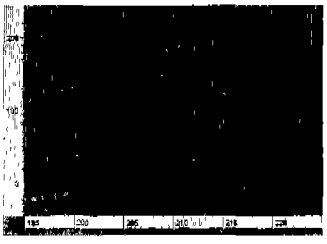
그림 12 모션 전이 - 칵테일

프레임의 전이 구간을 설정하였다. 따라서 다음 그림에서와 같이 200번째 프레임부터 220번째 프레임까지가 전이 구간이다. 그림 11에서는 두개의 춤 동작을 합성한 전이 동작을 다섯 프레임마다 한 프레임씩 나타내었다. 동작 전이 과정에서 UPM의 입력값으로 직접 사용한 프레임은 (b)와 (f) 프레임이고, (d)가 출력된 결과 프레임이다. 그림 12에서도 동일한 결과를 나타낸다. 동작 전이 과정에서 실제 자세 데이타에 의해 학습된 정보를 바탕으로 중간 동작들을 합성하게 되므로, 각 자세는 대상 다관절체의 고유한 동작의 특성을 유지하면서 자연스러운 동작을 연출하게 된다.

예를 들어 그림 12의 (d)를 보면, 오른쪽 어깨의 DOF 값이 그림 (b)와 (f)간의 산술적 평균 DOF값이 아님을 알 수 있는데, 이는 그림 13의 (a)에 나타난 오른쪽 어깨의 DOF 값의 그래프를 보면 더욱 확실히 알 수 있다. 그림 13의 (a)는 그림 12의 동작 전이 과정 중에서 오른쪽 어깨 관절의 DOF 값의 변화를 그래프로 나타낸 그림이다. 선형 보간법으로 동작 전이를 실행한 경우를 나타낸 그림 13의 (b)와 비교하였을 때, 자연스러운 곡선을 그리는 그림 13의 (a)와 다르게 그림 13의 (b)는 전이 구간에서 굴곡 없이 직선으로 떨어지는 모습을 보이며, 이러한 전이 과정은 부자연스러운 동작을 생성할 가능성이 있다. 그림 14은 UPM기법과 선형 보간법의 사용시, 전체 구간의 동작 곡선을 비교하여 이러한 동작 곡선의 차이를 더욱 명확하게 나타내었다.

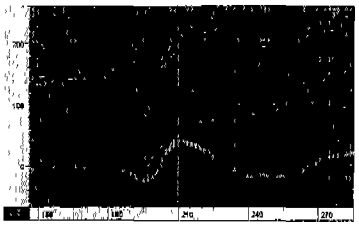


(a) UPM을 이용한 보간법

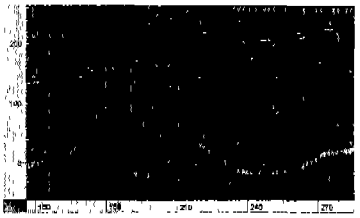


(b) 선형보간법

그림 13 전이 구간에서의 동작 곡선 비교



(a) UPM을 이용한 보간법



(b) 선형보간법

그림 14 전체 범위에서의 동작 곡선 비교

일반적으로 동작 전이 과정의 계산량은 사용되는 모델의 DOF수에 비례한다. spacctime 전이 방법을 이용하는 Rose의 실험[8]에서는 Pentium 100MHz 시스템상에서 44 DOF의 모델을 사용한 동작 전이를 수행하는데 72초가 소요되었다. 본 논문에서 제안한 UPM을 이용하여 계산속도가 유사한 Pentium 133MHz 시스템에서 63 DOF의 모델을 가지고 동작 전이를 수행한 결과 2초 이하의 시간이 소요되었다. 이것은 UPM 알고리즘이 계산량의 측면에서 장점을 가지고 있다라는 사실을 증명한다. UPM 알고리즘은 많은 동작 데이타들에 적용,실험한 결과로서 계산 속도나 결과 동작 측면에서 만족스러운 결과를 얻을 수 있었고 대부분의 결과에서 실세계에 존재하는 중간 자세들로 이루어진 부드러운 동작 전이를 얻을 수 있었다. 이와 같은 결과를 표 4에 정리하였다.

표 4 계산 시간의 비교

| Method | CPU | DOF | Processing Time |
|---|---|---|---|
| UPM | Pentium 133MHz | 63 | 1~2 Sec |
| Spacetime | Pentium 100MHz | 44 | 72 Sec |

## 5. 결 론

다관절체의 동적 특성을 분석적 방법으로 기술하기 위해 많은 연구가 진행되고 있으며 훌륭한 결과가 발표되고 있다[14]. 그러나 자발적인 힘으로 움직이는 다관절

체의 동작 양식이 매우 주관적이고, 예측이 어렵기 때문에 그 동작을 분석적인 방식으로 기술하는 것은 매우 난해한 일이다. 학습 네트웍은 다관절체의 동적 특성을 양자화하여 근사적으로 행동 경향을 표현하는 기법으로서 분석적인 동작 기술 방법 대신에 동작 특성의 귀납적인 기술에 이용될 수 있다. UPM은 기존의 조각 동작들을 이용해서 새로운 동작을 합성해 낼 수 있는 기법이다. UPM은 중간 동작의 생성에, 정도의 계산 부하만을 요구하며 실시간 동작 전이가 가능하다. 또한 결과 동작을 하나의 직관적인 제어 변수로 변형할 수 있으며, 사용자 상호 작용을 통한 동작 편집을 지원할 수 있다. 인체 동작 패턴의 특성에 의해 발생하는 비현실적인 자세의 생성을 학습 단계에서 차단함으로써 생성된 자세들을 실세계에 존재하는 자연스러운 자세들로 국한시킬 수 있다. 따라서 UPM은 작업자들이 자연스럽고, 사실적인 동작을 실시간 상호작용을 통하여 직관적으로 생성할 수 있는 도구로서 애니메이션 제작 과정에서 많은 기여를 할 수 있다. 또한 실시간 3D 게임이나 가상현실 그리고 Web3D 등의 다양한 응용 분야에도 이용될 수 있다.

## 참 고 문 헌


[1] Soon Ki Jung, Motion Analysis of Articulated Objects For Optical Motion Capture, Ph.D. thesis, Dept. of CS, KAIST, 1997.

[2] Douglas J. Wiley and James K. Hahn, Interpolation Synthesis of Articulated Figure Motion, IEEE Computer Graphics and Applications, Vol.17, No.6, pp.39-45, November, 1997.

[3] Armin Brudelin and Lance Williams, Motion Signal Processing, Proc. of SIGGRAPH 95, pp.97-104, ACM Press, 1995.

[4] Andrew Witkin and Zoran Popovic, Motion Warping, Proc. of SIGGRAPH 95, pp.105-108, ACM Press, 1995.

[5] Charles Rose, Bobby Bodenheimer and Michael F. Cohen, Verbs and Adverbs: Multidimensional Motion Interpolation, IEEE Computer Graphics and Applications, v. 18, no 5, pp. 32-40, September. 1998.

[6] Jehee Lee and Sung Yong Shin, A Hierarchical Approach to Interactive Motion Editing for Human-like Figures, Proc. of SIGGRAPH 99, pp. 39-48, ACM Press, 1999.

[7] Michael Gleicher, Motion Editing with Spacetime Constraints, Proc. of Symposium on Interactive 3D Graphics, pp.149-148, 1997.

[8] Charles Rose, Brian Guenter, Bobby Bodenheimer and Michael F. Cohen, Efficient generation of motion transitions using spacetime constraints, Proc. of SIGGRAPH 96, pp.147-154, ACM Press, 1996.

[9] Matthew Brand and Aaron Hertzmann, Style Machine, Proc of SIGGRAPH 2000, pp.183-192, ACM Press, 2000.

[10] Radek Grzeszczuk, Demetri Terzopoulos and Geoffrey Hinton, NeuroAnimator: Fast Neural Emulation and Control of Physics-Based Models, Proc. of SIGGRAPH 98, pp.9-20, ACM Press, 1998.

[11] Michael F. Cohen, Interactive Spacetime Control for Animation, Proc. of SIGGRAPH 92, pp. 293-302, ACM Press, 1992.

[12] Ritter, Martinetz and Schukten. Neural Computation and Self Organizing Maps. Addison Wesley. New York. 1992.

[13] Stephen I. Gallant, Neural Network Learning and Expert System. MIT Press, 1993.

[14] Jianmin Zhao and Norman I.Badler, Inverse Kinematics Positioning Using Nonlinear Programming for Highly Articulated Figures, ACM Transactions on Graphics, Vol.13, No.4, pp.313-336, October, 1994.



이 범 로

1993년 광운대학교 제어계측공학과(공학사). 1995년 광운대학교 제어계측공학과(공학석사). 1998년 광운대학교 제어계측공학과(박사과정수료). 1995년 1월 ~ 1996년 8월 현대전자 S/W 연구소 연구원 1996년 9월 ~ 1998년 4월 현대정보기술 인터넷 사업팀 선임. 1998년 4월 ~ 2000년 4월 LG 인터넷 시스템 개발팀 대리. 2000년 4월 ~ 2001년 3월 (주)OC7N 기술연구소 시스템 개발팀장. 2001년 4월 ~ 현재 (주)크리벨시스템즈 개발1실 실장. 관심분야는 Animation, VR, DSP, VHDL, Recognition, Network.



정 진 현

1981년 연세대학교 전기공학과(공학사). 1983년 연세대학교 전기공학과(공학석사). 1990년 Rensselaer Polytechnic Institute(Ph.D). 1991년 ~ 1995년 광운대학교 제어계측공학과 조교수. 1995년 ~ 2000년 광운대학교 제어계측공학과 부교수. 2000년 ~ 현재 광운대학교 정보제어공학과 교수. 관심분야는 DSP, VHDL, Automation, Intelligent Control, Recognition, Biometrics, Embedded System, Network.