

# 다관절체 오브젝트 편집시스템 (Articulated Body Editing System)

최 해 옥  
Haeock Choi

시스템공학연구소 컴퓨터그래픽스연구실

## 요 약

사실감 있는 모델의 생성과 동작제어 기술은 컴퓨터 그래픽스와 가상 현실감 등의 다양한 응용 분야에서 이용되고 있다. 컴퓨터 애니메이션에서 동작 제어의 대상이 되는 물체는 다관절체(articulated body)이며, 이 다관절체를 뼈대(link)와 관절(joint) 그리고 이를 둘러싸고 있는 피부로 모델링하고, 운동학(Kinematics), 동역학(Dynamics)을 적용하여 동작을 생성한다.

본 논문은 컴퓨터 애니메이션에서 사용할 수 있는 3차원 다관절체의 생성과 애니메이션을 위한 다관절체 오브젝트 편집 시스템에 관한 것이다. 다관절체를 관리하기 위한 다관절체의 데이터 구조를 설계하고, 사용자 인터페이스를 추가하여 대화식으로 다관절체의 골격을 정의한다. 정의된 다관절체의 골격과 물체의 모양을 나타내는 기하 데이터를 집합하여 애니메이션에 적합한 물체를 모델링하기 위한 스킨-스켈레톤 랑고리즘을 제안한다. 모델링된 물체의 관절을 조작하여 키프레임 애니메이션으로 동작을 제어한다. SGI 워크스테이션에서 Open Inventor와 X/Motif를 이용하여 C++ 언어로 구현하였으며, 인체 모델을 대상으로 실험하였다.

## 1. 서 론

컴퓨터 그래픽스 및 가상 현실감 시스템의 여러 응용 분야에서 3차원 모델링과 컴퓨터 애니메이션 기술에 대한 요구가 대두되고 있다.

컴퓨터 애니메이션에서 동작 처리의 대상이 되는 모델은 사람을 비롯하여 움직이는 동물들이다. 이들 동물들은 다관절체(articulated body)로서 컴퓨터 애니메이션을 위한 다관절체의 모델링과 이의 동작제어 기술에 대해서 많은 연구가 진행되고 있다. 다관절 물체의 모델링과 이의 동작을 제어하는 기술은 먼저 관절체의 골격을 뼈대와 관절, 이를 둘러싸고 있는 피부로 모델링하고, 운동학(Kinematics)이나 동역학(Dynamics)을 기반으로 관절을 조작하여 움직임을 생성한다[1].

컴퓨터 애니메이션에서 일반적으로 사용하는 모델은 표면 모델(surface model)로서 내부의 뼈대를 둘러싸고 있는 피부를 다각형(polyhedral)의 조합이나 자유 곡면(free-form surface)으로 이루어진 표면(surface)으로 모델링한 것이다. 이런 특징

을 가지는 컴퓨터 애니메이션용 3차원 모델을 작성하기는 매우 어렵다. 다관절체의 모양을 나타내는 3차원 표면 모델을 작성하기 위해서는 3차원 모델링 도구가 필요하며, 다관절체의 관절을 표현하기 위해서는 다관절체 편집 도구도 필요하다. 또한 동작의 제어를 통한 사실적인 움직임의 표현을 위해서는 관절 데이터와 표면 데이터는 서로 연결되어 있어야 한다.

본 논문에서는 컴퓨터 애니메이션에서 사용할 수 있는 3차원 모델을 쉽게 작성하기 위한 다관절체 편집 시스템에 대하여 논한다. 이를 위해서 일반적인 다관절체의 골격 구조를 관리할 수 있는 데이터 구조를 계층적 트리 구조(hierarchical tree structure)로 설계하고, 이 골격 구조에 일반적인 모델링 시스템으로 작성된 물체의 기하 데이터를 집합(binding)하는 스킨-스켈레톤 바인딩(skin-skeleton binding) 알고리즘을 제안한다[2]. 제안하는 알고리즘은 다음의 세가지 기능 모듈로서 구성된다. 첫째, 두 종류의 데이터 즉 다관절체의 골격 데이터와 피부를 표현한 기하 데이터를 읽어서 알고리즘에서 사용하기 위한 데이터로 변환하는 전처리(preprocessing)

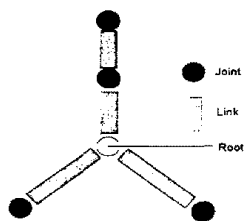
모듈, 둘째, 기하 데이터를 물체의 각 부분으로 분할하고, 분할된 피부 데이터와 골격 데이터의 바운딩 박스를 계산하는 세그멘테이션(segmentation) 모듈과, 셋째, 각 골격에 대응하는 피부 데이터를 찾아서 연결하는 바인딩(binding) 모듈로 구성된다. 바인딩 알고리즘은 골격 데이터와 피부 데이터의 연결성을 관리하는 후보 테이블(candidate table)을 사용하며, 알고리즘의 속도를 빠르게 하기 위해서 분할 해결(divide-and-conquer) 방식을 도입하였다[3]. 교차 검사(overlapping test), 그룹핑 검사(grouping test)와 연결 검사(connection test)의 3단계 검사를 통하여 실행되며, 마지막으로 하나의 피부 데이터가 두 개 이상의 골격 데이터에 연결된 경우에는 피부 데이터를 분할(facet splitting)하여 각 골격에 대응되는 피부 데이터로 변환한다. 본 알고리즘을 통하여 작성된 모델을 키프레임 애니메이션으로 동작을 제어한다.

본 논문에서 제안한 모델링 방법은 일반적인 다관절체에 적용할 수 있지만 다관절체중 복잡한 구조를 가지며 애니메이션에서 많이 이용되는 인체 모델에 대해서 실험하였다.

## 2. 다관절체 모델

### 2.1 다관절체 구조와 동작제어 기술

다관절 물체는 [그림1]과 같이 강체 링크(rigid link)가 조인트(joint)에 연결되어 있는 개루프(open chain) 형태의 구조로 되어 있다. 이러한 다관절 구조의 물체에 대한 제어는 로봇이나 기계 분야에서 많이 연구되어 졌다[4]. 다관절 물체의 조인트는 움직임의 형태에 따라서 회전(revolute) 조인트, 미끄럼(translational or prismatic) 조인트, 원통(cylindrical) 조인트, 유니버설(universal) 조인트와 자유(free or ball) 조인트 등으로 나눈다.



[그림1] 다관절체의 구조

컴퓨터 애니메이션에서 동작 제어의 대상이 되는 동물은 대부분 다관절 구조이며, 동물을 다관절 구조로 표현할 때, 관절을 조인트로, 관절 사이를 잇는 뼈대를 링크로 표현한다. 컴퓨터 애니메이션을 위한 조인트는 일반적으로 회전 조인트와 자유 조인트가 많이 사용된다. 인체 모델을 예로 들면, 무릎이나 팔꿈치 관절과 같이 한 축을 기준으로만 회전하는 관절을 회전 조인트로, 어깨 관절과 같이 세축으로 자유롭게 회전하는 관절은 자유 조인트로 모델링한다.

관절체 모델로 컴퓨터 애니메이션에서 동작을 제어하기 위한 연구는 Zeltzer에 의해서 많이 시도되었다[5]. 관절체 모델의 동작 제어는 먼저 관절의 구조를 DH(Denavit-Hartenberg)방식[6]이나 AP(Axis Position) joint 방식[7]등으로 정의한 후 운동학(kinematics)을 기반으로 동작을 제어한다. 운동학은 링크와 조인트로 연결되는 관절 구조의 위치, 속도를 제어할 수 있다. 인체 모델에 운동학을 적용하는 방법은 전운동학(forward kinematics)/역운동학(inverse kinematics)의 두가지 방법으로 적용된다. 전운동학을 이용하는 방법은 예를 들어, 인체 관절중 움직임을 생성하기 위한 관절의 각도 즉 팔을 움직이고 싶을 경우 어깨, 팔꿈치, 손목의 세관절의 각을 지정하여 움직임을 생성한다. 즉 <식1>과 같이 각 관절의 회전각도( $\theta$ )를 입력하여 움직여진 위치(X)를 계산하는 방법이다.

$$X = f(\theta) \text{ ----- } \langle \text{식1} \rangle$$

where X : position in the 3D space  
 $\theta$  : rotation angle of the joint

역운동학은 <식2>와 같이 한 체인에 연결되어 있는 관절의 마지막 말단효과(end effector) 관절의 위치(X)만을 지정하여 모든 중간 관절들의 각도( $\theta$ )를 계산한다. 즉 팔을 움직이는 애니메이션을 위해서 손끝이 놓일 위치만을 지정하여 어깨, 팔꿈치, 손목의 중간 관절의 각도를 계산한다.

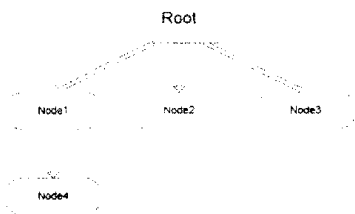
$$\theta = f^{-1}(X) \text{ ----- } \langle \text{식2} \rangle$$

where X : position in the 3D space  
 $\theta$  : rotation angle of the joint

일반적으로 다관절체의 애니메이션을 위해서 모든 관절의 각도를 지정하는 것은 불가능하다. 따라서 동작을 이루는 관절의 주요 위치만을 지정하고 나머지 관절의 각도를 계산하여 동작을 제어하는 역운동학 기술은 관절체 모델의 동작제어에서 필수적으로 구현되어야 하는 기술이다. 그러나 역운동학으로 모든 중간 관절의 각을 계산해야 하기 때문에 계산 시간이 오래 걸리는 단점이 있으나 실시간으로 역운동학을 계산하기 위한 알고리즘의 개발과[8] 최근의 하드웨어 시스템의 발전, 그래픽 전용 계산 시스템 등을 사용하면 실시간으로 역운동학 기반의 애니메이션도 가능하다.

## 2.2 다관절체 데이터 구조

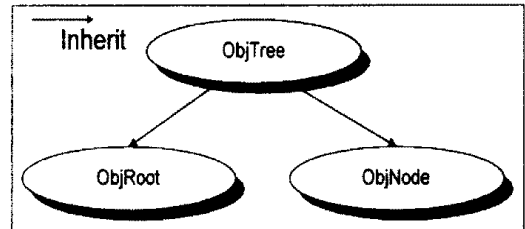
다관절체는 [그림1]과 같이 링크와 조인트로 구성되는 세그먼트(segment)들이 연결되어 있는 형태의 물체이다. 한 조인트의 축은 상위 조인트의 회전과 링크의 길이에 의존적으로서, 한 조인트의 상태가 변하면 그 조인트에 연결되어 있는 모든 조인트와 다시 그 조인트의 하위 조인트로 영향이 파급된다. 다관절체의 중심 관절을 제외한 각 관절은 한 개의 상위관절에 연결되어 있으며 N개의 다른 관절이 그 관절에 연결되어 있는 것으로 볼 수 있다. 즉, 한 개의 부모(parent)관절과 N개의 자식(child)관절로 구성된다. 이러한 다관절체의 계층적 특성을 가장 잘 표현할 수 있는 자료 구조는 트리(tree) 구조이다. [그림1]의 다관절체는 [그림2]와 같이 중심이 되는 관절 즉 루트(root)를 기준으로 계층적(hierarchical) 트리 구조를 가지게 된다. 일반적인 다관절체의 구조를 관리하기 위하여 객체 지향 개념으로 다음과 같이 클래스(class)를 정의하였다.



[그림2] 다관절체의 계층적 트리노드 구조

### 2.2.1 다관절체의 트리구조 클래스

다관절체의 데이터는 한 세그먼트를 하나의 노드로 하는 트리구조로 구성되며 한 트리 구조는 하나의 루트 노드와 그 이외의 모든 노드로서 구성된다. 따라서 트리내의 모든 노드는 하나의 루트 노드(ObjRoot)를 제외하고 모두 오브젝트 노드(ObjNode)로 정의된다. [그림3]과 같이 트리 구조를 관리하기 위한 트리노드(ObjTree)가 슈퍼클래스로 모든 하위 노드로 상속(inherit)된다. 다관절체를 관리하기 위한 클래스는 다음과 같다.



[그림3] 트리노드의 상속

#### 2.2.1.1 ObjTree 클래스

트리구조를 생성하고 관리한다. 생성된 노드의 Identifier, Name을 관리하며 1개의 부모노드와 N개의 자식노드에 대한 pointer를 가진다. 이때 N개의 자식노드는 ObjList클래스에 의해 관리된다.

#### 2.2.1.2 ObjRoot 클래스

트리의 루트노드로 전체 트리내에서 단 한개만 존재한다. 트리노드는 다관절체 전체의 위치, 크기, 회전을 관리하는 클래스로서 루트노드의 위치, 크기, 회전이 변화하면 전체 다관절체가 변화한다.

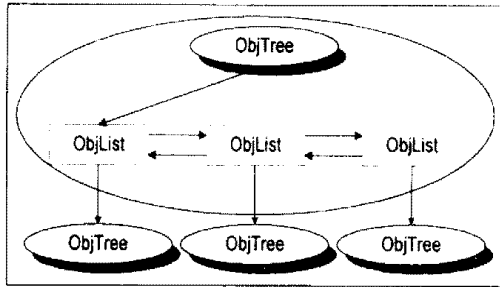
#### 2.2.1.3 ObjNode 클래스

링크와 조인트로 구성되는 하나의 세그먼트를 표현하는 클래스이다. 다관절체의 하나의 관절에 대한 데이터를 의미하며 각 관절의 종류, 움직임의 범위를 나타내는 최대/최소 회전각, 현재의 회전상태정보와 링크의 길이 정보를 가지고 있다.

#### 2.2.1.4 ObjList 클래스

트리에서 한 노드로부터 파생되는 자식 노드의

수는 제한이 없다. 이를 처리하기 위해서 양방향 링크드 리스트로 자식 노드를 관리한다. 트리내의 각 노드는 실제로는 [그림4]와 같이 표현된다. 그러나, 이들의 관리는 ObjList를 통해 관리되고 있으므로 사용자 레벨에서의 추상화가 제공된다.



[그림4] ObjList 클래스에 의한 추상화

## 2.2.2 Tree의 탐색

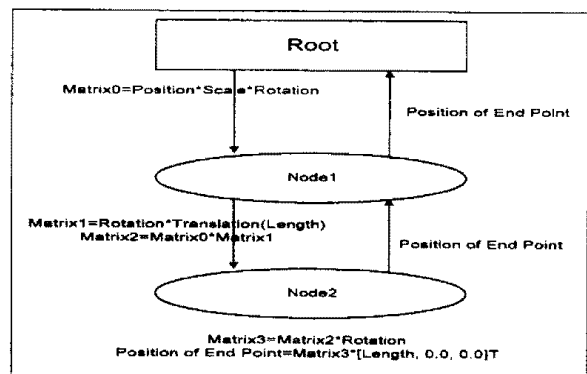
트리에 존재하는 노드로의 접근은 특별한 경우를 제외하고 일반적으로 상위 노드를 통해서만 이루어진다. 이런 방법으로만 노드를 접근할 수 있도록 하면 어떤 노드에 삭제, 생성등의 연산을 원할 경우 해당 노드까지 접근하기 위한 방법이 필요하다. 이를 위해서 루트로부터 해당 노드까지의 경로(path)를 담고있는 ObjPath 클래스 또는 ObjIdPath 클래스를 이용하여 한 노드의 연산을 루트에 요구할 수 있도록 하였다. 예를 들어 [그림5]의 matrix 연산과정에서 Node2의 끝점의 좌표를 얻고 싶을 경우 루트에서의 위치, 크기, 회전각에 관한 정보를 기본으로 Matrix가 생성되고 이것이 Node1에 전달된다. Node1에서는 현재 조인트의 회전각과 세그먼트의 길이를 기준으로 새로운 Matrix를 생성하여 루트에서 전달된 Matrix에 곱한다. 이렇게 생성된 Matrix를 Node2에 전달하면 Node2에서는 조인트의 회전각을 기준으로 Matrix를 만들어 곱하고 이렇게 생성된 Matrix와 (세그먼트의 길이, 0.0, 0.0)을 곱해 끝점을 얻게 된다.

### 2.2.2.1 ObjPath 클래스

트리내에서 루트로부터 원하는 노드까지의 경로(path)를 pointer의 배열로 관리한다.

### 2.2.2.2 ObjIdPath 클래스

Pointer를 이용한 접근은 단단계 Undo기능이 구현되면 위험하다. 예를들어 어떤 노드의 값을 바꾼 후(1), 그 노드를 삭제하고 나서(2), 취소(Undo)하게 되면 다시 노드를 생성하고 저장되었던 값들을 복구해야 하는데 이때 객체가 생성되는 address가 바뀌게 된다. 따라서, 두 번의 연속된 취소(Undo)후 원래 노드를 정확히 찾아 갈 수 없게 된다. 따라서 각 노드를 pointer로 관리하지 않고 고유한 Id값으로 관리하도록 한다.



[그림5] 각 노드의 matrix 연산과정

## 3. 스킨-스켈레톤 바인딩 (Skin-Skeleton Binding)

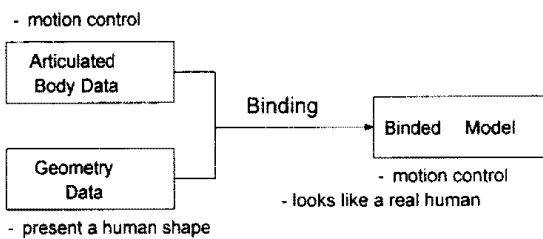
### 3.1. 개요

컴퓨터 애니메이션에서 사용하는 3차원 모델은 먼저 다관절 물체의 골격을 뼈대(skeleton)와 관절(joint)로 정의하고, 이를 둘러싸고 있는 피부(skin)를 간단한 표면 모델(surface model)이나 보다 생물학에 기초를 둔(anatomically-based) 방법으로 모델링한다 [9]. 즉 애니메이션을 위한 모델은 동작 제어의 대상인 관절 데이터와 사실적으로 보이도록 하는 표면 기하 데이터가 함께 있어야 한다.

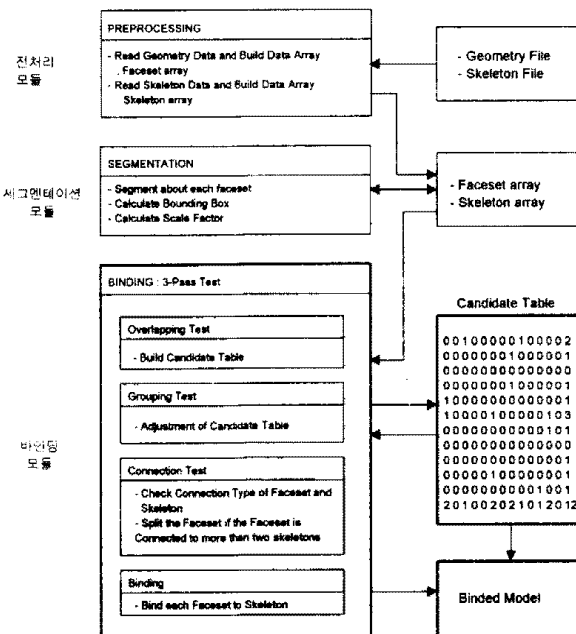
본 논문에서는 [그림6]과 같이 컴퓨터 애니메이션에서 사용할 수 있는 3차원 모델을 쉽게 작성하기 위해서 다관절체의 골격 구조 데이터와 일반적인 모델링 시스템으로 작성된 물체의 표면을 나타내는 기하(geometry) 데이터를 접합하는 스킨-스켈레톤 바

인딩 알고리즘을 제안한다. 제안하는 알고리즘은 [그림7]과 같이 세가지 기능 모듈로서 구성된다.

제안한 알고리즘은 다음의 두가지 기법을 도입하여 알고리즘의 효율을 높이도록 하고 있다. 후보 테이블(candidate table)을 이용하여 각 faceset과 skeleton의 연결성을 관리하는 기법과 분할 해결(divide-and-conquer) 기법을 도입하여 많은 수의 후보 데이터들에 대해서 알고리즘을 모두 적용하지 않고 먼저 연결의 가능성이 있는 후보들을 찾은 후, 이 후보 데이터들에 대해서만 알고리즘을 적용하여 실행 시간을 줄이도록 하였다. 또한 본 알고리즘은 데이터의 위치 좌표를 이용하여 계산하는 수학적(mathematical) 방법과 수학적 방법으로 해결되지 않는 부분에 대한 경험적(heuristic) 방법을 함께 사용한다.



[그림6] 스킴-스켈레톤 바인딩



[그림7] 스킴-스켈레톤 바인딩 알고리즘 구성도

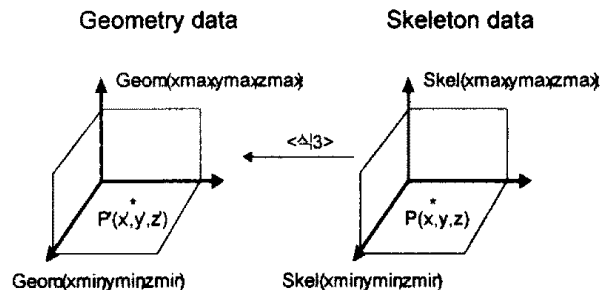
일반적으로 3차원 물체의 외모를 표현하는 데이터는 매우 복잡하고 내부에 포함되는 뼈대 데이터와 수학적으로 일치하지 않는 경우가 많기 때문이다. 경험적 방법은 알고리즘의 바인딩 모듈에서 그룹핑 검사에 적용된다.

### 3.2 전처리(Preprocessing)

전처리 모듈에서는 두종류의 데이터 즉 관절구조 데이터와 기하 데이터를 읽어서 스킴-스켈레톤 바인딩 알고리즘에서 필요한 데이터를 추출하여 데이터 배열을 구성한다. 관절 구조 데이터로부터는 관절의 전체 크기 정보와 각 관절의 위치 및 크기 정보를 관리하는 skeleton 배열을 구성하고, 기하 데이터로부터는 전체 기하 데이터의 크기 정보와 각 faceset의 위치 및 크기를 관리하는 faceset 배열을 구성한다.

### 3.3 세그멘테이션(Segmentation)

전처리 모듈에서 구성된 데이터 array의 각 faceset과 skeleton의 바운딩 박스(bounding box)를 계산한다.바운딩 박스는 각 세그먼트를 감싸는 직육면체의 최대, 최소값과 바운딩 박스의 크기, 중심의 위치정보를 가지고 있으며 그룹핑과 바인딩 모듈에서 검사를 위해서 사용된다. 또한 입력된 두 파일 즉 피부 데이터와 골격 데이터는 크기가 같지 않으므로 두 데이터의 크기를 보정하기 위한 두 좌표계간의 좌표 보정이 필요하다. 본 논문에서는 [그림8]과 같이 골격 데이터를 피부 데이터로 보정하며, 좌표 보정을 위한 식은 <식3>과 같다.



[그림8] 두 좌표계간의 좌표 보정

$$scl_x = \frac{Geom.xmax - Geom.xmin}{Skel.xmax - Skel.xmin}$$

$$scl_y = \frac{Geom.ymax - Geom.ymin}{Skel.ymax - Skel.ymin} \quad \langle \text{식3} \rangle$$

$$scl_z = \frac{Geom.zmax - Geom.zmin}{Skel.zmax - Skel.zmin}$$

$$x' = (x - skel.xmin) * scl_x + Geom.xmin$$

$$y' = (y - skel.ymin) * scl_y + Geom.ymin$$

$$z' = (z - skel.zmin) * scl_z + Geom.zmin$$

<표1> 후보 테이블(Candidate Table)

	Skeletons	Total
Faceset[0]	: 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1	
Faceset[1]	: 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3	
Faceset[2]	: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0	
Faceset[3]	: 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 2	
Faceset[4]	: 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 2	
Faceset[5]	: 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1	
Faceset[6]	: 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1	
Faceset[7]	: 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1	
	⋮	
Faceset[26]	: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1	
Faceset[27]	: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 2	
계	: 2, 2, 2, 6, 3, 2, 2, 4, 2, 3, 3, 3, 2, 3, 3, 1, 0	

### 3.3 바인딩(Binding)

피부의 각 부분을 나타내는 faceset 배열과 인체의 골격 데이터인 skeleton 배열을 검사하여 각 뼈대에 해당하는 피부 데이터를 찾는다. 바인딩 알고리즘은 <표1>의 후보 테이블(Candidate Table)을 이용한다. 후보 테이블은 faceset과 skeleton과의 접합 관계를 나타내는 테이블로서 가로 방향의 skeleton과 세로 방향의 faceset이 교차하는 항목이 '0'이면 접합 관계가 없는 것을, '1'이면 접합 관계가 있는 것을 의미한다. 또한 분할 해결 (divide-and-conquer) 방식을 적용하여 그룹핑을 3단계의 테스트(3-pass test)를 거쳐서 수행한다. 알고리즘의 설명을 위해서 본 논문의 실험 데이터인 인체 모델을 이용한다.

#### 3.3.1 교차 검사(overlapping test)

각 faceset array와 skeleton array의 바운딩 박스를 검사하여 후보 테이블을 작성한다. 교차 검사를 위한 식은 <식4>와 같다.

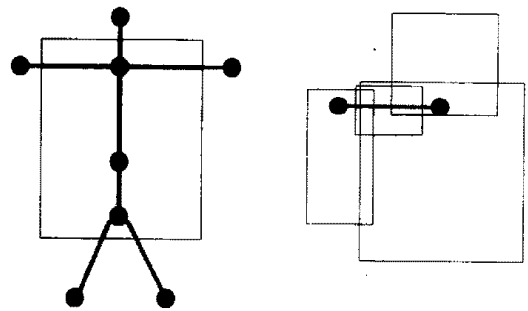
$$T(i, j) = '0' : \text{if } F(i)_{xmax} < S(j)_{xmin} \text{ or } F(i)_{ymax} < S(j)_{ymin} \text{ or } F(i)_{zmax} < S(j)_{zmin} \text{ or } F(i)_{xmin} > S(j)_{xmax} \text{ or } F(i)_{ymin} > S(j)_{ymax} \text{ or } F(i)_{zmin} > S(j)_{zmax}$$

$$'1' : \text{elsewhere} \quad \langle \text{식4} \rangle$$

where  $T(i, j)$  : (i, j)th item in Candidate Table  
 $F(i)_{xmin}, F(i)_{ymin}, F(i)_{zmin}$  : bounding box  
 $F(i)_{xmax}, F(i)_{ymax}, F(i)_{zmax}$  : of ith faceset  
 $S(j)_{xmin}, S(j)_{ymin}, S(j)_{zmin}$  : bounding box  
 $S(j)_{xmax}, S(j)_{ymax}, S(j)_{zmax}$  : of jth skeleton

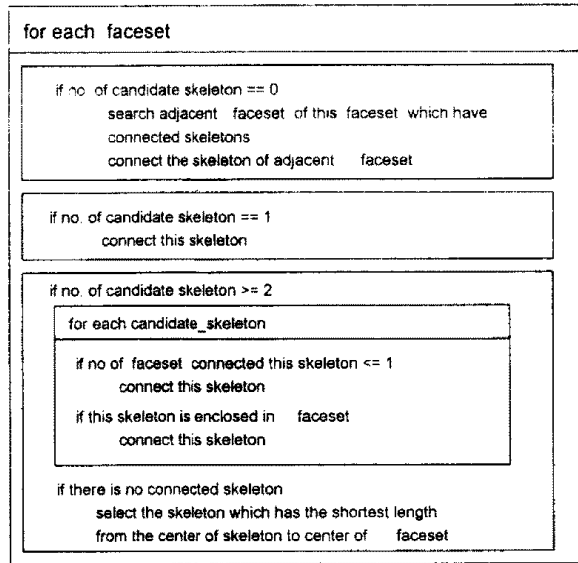
#### 3.3.2 그룹핑 검사(grouping test)

교차 테스트를 통하여 작성된 후보 테이블은 [그림9]의 (1)과 같이 상반신 faceset의 경우 7개의 skeleton과 교차한다. 또한 상반신 faceset과 교차하고 있는 skeleton중 오른쪽 어깨 골격은 [그림9]의 (2)와 같이 4개의 faceset과 교차하고 있다. 실제로 상반신 faceset에 연결되어야 할 skeleton은 척추의 두 skeleton뿐이며 또한 오른쪽 어깨 관절도 이 관절에 해당되는 faceset으로 연결되도록 하는 처리가 필요하다.



(1)상반신 faceset에 연결 (2)오른쪽 어깨 skeleton에 연결된 faceset들  
 [그림9] 교차 검사로 연결된 skeleton과 faceset

이를 위해서는 데이터의 특성을 고려한 경험적인 방법의 처리가 필요하며 후보 테이블의 각 faceset에 대해서 [그림10]의 알고리즘을 적용하여 그룹핑 검사를 행한다.



[그림10] 그룹핑 검사(Grouping Test)알고리즘

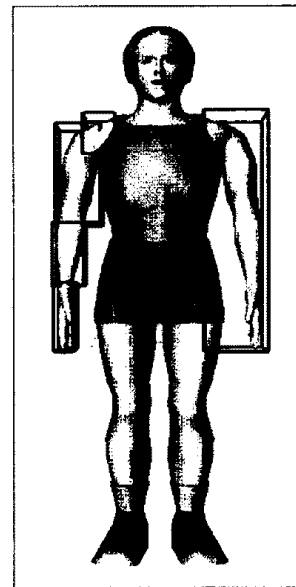
이상의 알고리즘으로 그룹핑 테스트를 수행하면 상반신 faceset에는 두 개의 척추 골격이 연결되고, 오른쪽 어깨 골격에는 오른쪽 어깨의 모양을 표현한 faceset이 연결된다.

### 3.3.3 연결 검사(connection test)

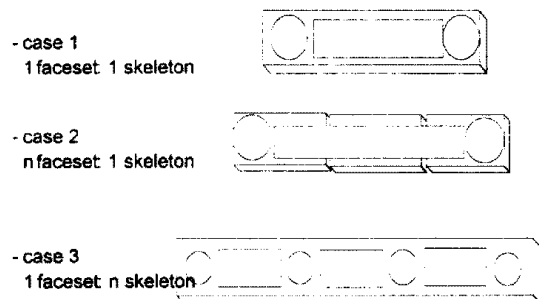
접합된 모델에서 하나의 뼈대에는 하나의 피부 데이터가 연결되어 있는 것이 이상적이다. 그러나 그룹핑을 거쳐서 작성된 후보 테이블을 살펴보면 하나의 faceset에 2개 이상의 skeleton이 연결되어 있거나 하나의 skeleton에 2개 이상의 faceset이 연결되어 있는 것이 있다. 이것은 [그림11]과 같이 인체의 피부 데이터를 작성하는 사람에 따라서 여러 가지 방법으로 다르게 기하 데이터를 모델링하거나 같은 형태의 데이터도 다르게 표현될 수 있기 때문이다.

여기서 하나의 뼈대에 여러개의 faceset이 연결되어 있는 경우는 문제로 삼지 않는다. 왜냐하면 하나의 skeleton이 움직일 때 여러개의 faceset이 같이 붙어서 움직이도록 하면 움직임에 이상이 없기 때문이며 때에 따라서는 하나의 skeleton에 여러개의 faceset이 붙어야 할 때도 있기 때문이다. 위의 예에서 얼굴 부분의 skeleton은 얼굴뿐만이 아닌 머리, 눈, 입술 등의 faceset을 포함하는 것을 예로 들 수 있다.

그러나 [그림11]의 왼쪽 팔과 같이 하나의 faceset에 여러개의 skeleton이 연결되어 있는 경우는 독립적으로 움직이는 각 skeleton의 움직임을 처리할 수 없으므로 하나의 faceset을 연결되어 있는 skeleton들에 맞도록 잘라서 각 skeleton에 연결해야 한다. 이를 위해서 각 faceset과 skeleton의 연결 형태를 조사하는 연결 검사(connection test)와 연결 검사에서 하나의 faceset에 두 개 이상의 skeleton이 연결되어 있는 경우의 처리를 수행한다. 연결 검사에 의한 Faceset과 skeleton의 연결 상태는 [그림12]와 같이 3경우로 구분된다.



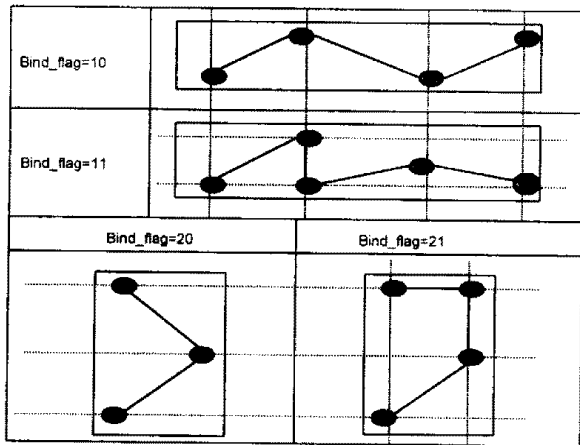
[그림11] 다양한 형태의 faceset의 예



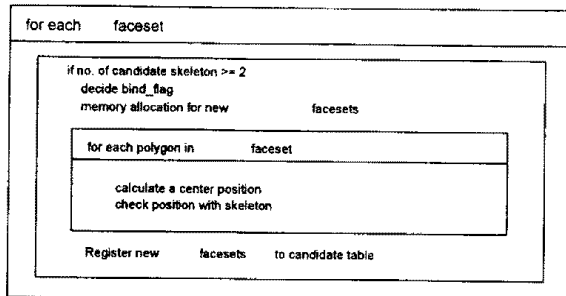
[그림12] 연결 검사에 의한 연결 상태

### 3.3.4 Faceset의 분할(split)

연결 검사에서 “case 3”에 해당하는 faceset은 연결되어 있는 skeleton의 각 관절의 위치를 기준으로 분할되어 각 skeleton에 연결되어야 한다. 이를 위해서 faceset과 skeleton의 형태를 검사하여 bind\_flag를 정하고 faceset의 각 polygon의 위치를 검사하여 polygon이 위치하는 skeleton에 polygon이 연결될 수 있도록 faceset을 분할한다. faceset과 skeleton의 모양에 따른 bind\_flag는 [그림13]과 같다. 각 polygon의 위치에 대한 검사는 polygon의 중심의 위치를 구한 후, [그림13]에서 bind\_flag가 10 또는 20인 경우는 각각 X축 또는 Y축 방향으로만 polygon의 중심의 위치로 검사하여 해당하는 skeleton을 찾는다. 그러나 bind\_flag가 11 또는 21인 경우는 각 검사축에 일치하는 skeleton에서는 반대방향으로의 검사를 다시 실행해야 한다. Faceset의 분할을 위한 알고리즘은 [그림14]와 같다.



[그림13] bind\_flag



[그림14] Faceset 분할 알고리즘

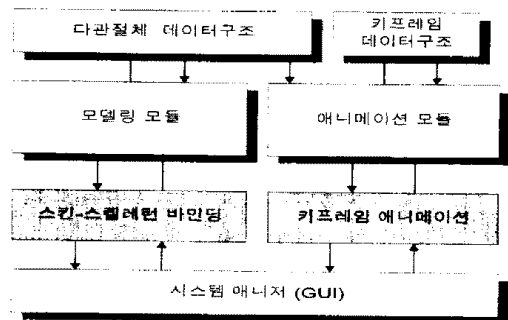
이러한 과정을 거쳐서 하나의 faceset이 연결되어 있는 skeleton의 수로 분할되면, 후보 테이블에서 원래의 faceset 열은 '0'으로 reset되고 분할된 faceset들의 열만큼 후보 테이블에 등록되어야 한다. 따라서 이과정에서 후보 테이블의 faceset 수가 늘어나게 된다.

### 3.3.5 바인딩(binding)

지금까지의 알고리즘을 수행하여 형성된 후보 테이블을 기준으로 각 skeleton에 해당하는 faceset을 연결한다. 이러한 스킴-스켈레톤 집합으로 구성된 모델로 컴퓨터 애니메이션을 수행할 수 있다.

## 4. 다관절체 편집 시스템

다관절체의 모델링을 위해서는 다관절체 구조를 생성하고 관리하는 시스템이 필요하다. 본 논문에서는 일반적인 다관절체 물체를 생성하기 위한 다관절체 편집 시스템을 구현하고 생성된 다관절체 물체의 키프레임 애니메이션을 구현하였다. 이를 위해서 앞에서 설명된 바와 같이 다관절체 오브젝트의 데이터 구조를 설계하고, 사용자가 쉽게 링크와 조인트를 생성할 수 있도록 사용자 인터페이스를 구현하였다. 구현된 다관절체 편집 시스템의 구성은 [그림15]와 같다.



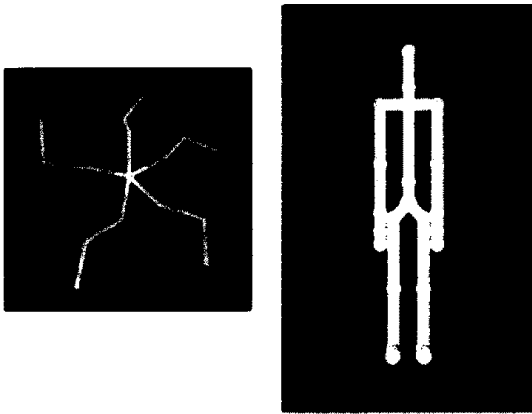
[그림15] 다관절체 편집 시스템 구성도

시스템은 다관절체를 생성, 편집하는 다관절체 모델링 모듈과 생성된 다관절체 오브젝트의 동작을 제어하기 위한 애니메이션 모듈로 구성된다. 동작의 처리는 키프레임 애니메이션으로 처리한다. 각 모듈은



다관절체 데이터 구조와 키프레임 데이터 구조를 관리하며, 사용자 인터페이스를 처리하는 시스템 매니저 모듈이 전체 시스템을 관리한다. 구현된 시스템은 일반적인 다관절체를 생성, 처리할 수 있으며, 본 시스템으로 생성된 다관절체의 예는 [그림16]과 같다.

시스템 개발에 사용된 플랫폼은 Silicon Graphics Workstation이며 Open Inventor를 사용하였다 [10][11]. 사용자 인터페이스를 위한 메뉴나 다이얼로그 박스는 X/Motif를 사용하여 구현하였다.



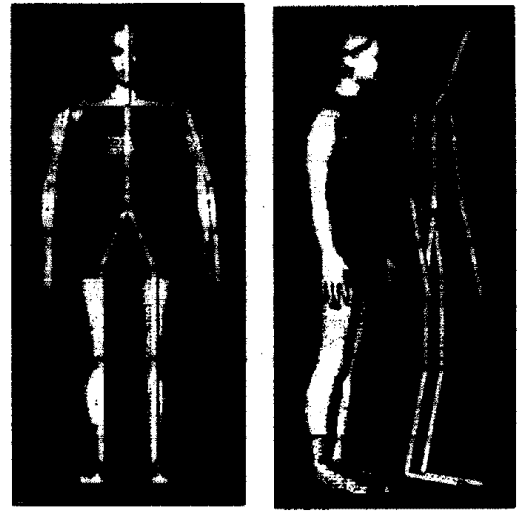
[그림16] 생성된 다관절체 모델의 예

## 5. 실험

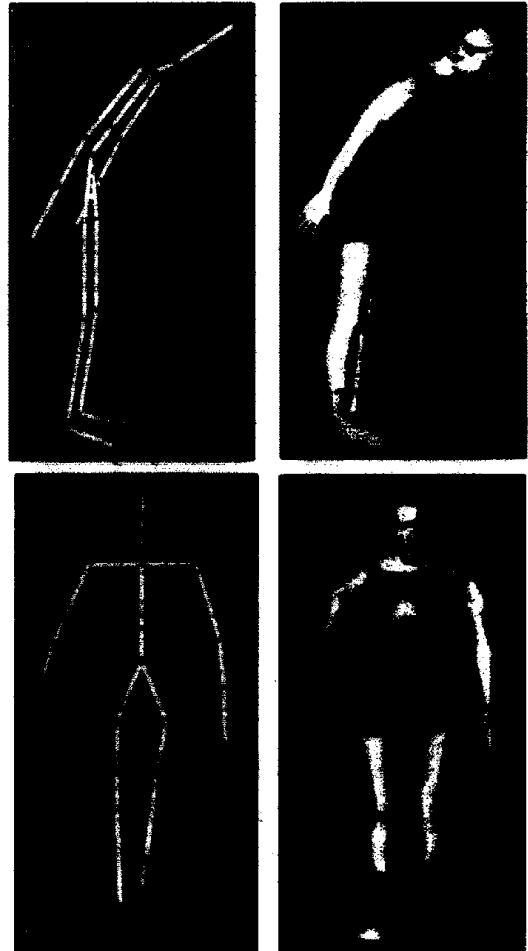
구현된 다관절체 편집 시스템을 인체 모델링에 대하여 실험하였다. 실험을 위한 작업 순서는 다음과 같다.

- ① 인체의 외양을 모델링한 3차원 기하 데이터를 입력하여 display 한다.
- ② 기하 데이터의 위에서 관절을 정의한다.
- ③ 스킨-스켈레톤 알고리즘을 수행시킨다.
- ④ 관절을 조작하여 동작을 생성한다.

물체의 외양을 표현하는 기하 데이터는 Open Inventor의 iv 포맷의 데이터를 이용하였다. [그림17]과 같이 인체의 기하 데이터위에 관절 데이터를 입력하고 두 데이터를 접합하여 애니메이션에 사용할 수 있는 관절과 피부가 접합된 모델을 생성한다. [그림18]은 접합된 인체 모델에 동작을 적용한 예이다.



[그림17] 인체의 기하 데이터와 관절 데이터의 정의



[그림18] 접합된 인체 모델의 동작 적용 예

## 6. 결론 및 향후 연구과제

본 논문에서는 컴퓨터 애니메이션에서 사용하는 다관절체의 모델링을 위한 방법을 제안하였다. 다관절체 물체를 생성하고 관리하기 위한 다관절체의 데이터 구조를 설계하였으며 키프레임 애니메이션으로 동작을 처리하였다. 이와 함께 물체의 사실적인 표현을 위해서 다관절체 골격 구조에 피부 표면의 기하 데이터를 접합하는 스킨-스켈레톤 바인딩 (skin-skeleton binding) 알고리즘을 제안하였다. 제안된 알고리즘은 20개의 관절로 구성된 인체의 골격 모델과 Open Inventor 포맷의 기하 데이터와의 접합에 대하여 실험하였다.

본 논문에서 제안한 스킨-스켈레톤 바인딩 기법은 3차원 애니메이션에서 가장 복잡하고 어려운 문제로 대두되고 있는 인체의 모델링을 쉽게 할 수 있도록 한 것에 의미가 있다. 현재 구현된 시스템은 기하 데이터로서 Inventor 포맷의 데이터를 사용하고 있으며 관절 데이터의 크기와 기하 데이터의 크기가 달라도 접합이 가능하다. 그러나 3차원 기하 데이터가 관절체 구조에 맞지 않게 작성된 경우에는 접합이 되지 않으며, 이를 위해서 3차원 기하 데이터를 관절 구조에 맞도록 편집하는 기능이 추가되면 일반적으로 작성된 3차원 모델 데이터를 쉽게 애니메이션에 사용할 수 있는 데이터로 변환 할 수 있다. 또한 설계한 다관절체 데이터의 구조는 인체 모델만이 아닌 일반적인 다관절체 구조를 관리할 수 있도록 설계되었으므로 네다리를 가진 동물이나 기계와 같은 형태의 모든 다관절체 모델링에 적용할 수 있다.

본 논문에서는 피부의 변형에 대해서는 다루지 않았다. 현재 골격 구조와 연결된 피부 데이터 구조를 피부 변형이 가능한 형태의 데이터로 바꾸고 두 데이터가 결합된 인체 모델에 동작이 적용될 때의 피부 변형을 위한 연구가 진행중이다.

이러한 기술은 컴퓨터 그래픽스와 가상 현실감의 다양한 응용 시스템에 활용할 수 있다. 예를 들어 과학 기술용 시뮬레이션 시스템을 비롯하여, 3차원 대화식 게임이나 최근에 부상하고 있는 네트워크를 이용한 공유 가상 환경에 등장하는 에이전트 (agent) 등의 처리를 위한 기반 기술로 활용 가능하다.

## 참고문헌

- [1] Watt A, Watt M, "Advanced Animation and Rendering Techniques", p384-394, Addison Wesley, 1995.
- [2] 최혜옥, 전병민, 김문현, "인체 모델링을 위한 스킨-스켈레톤 바인딩 기법", 한국컴퓨터그래픽스학회 논문지, Dec. 1997. (예정)
- [3] Sara Baase, "Computer Algorithms, Introduction to Design and Analysis, 2ed", p53-65, Addison-Wesley Publishing Company, 1988.
- [4] Farid M.L. Amirouch, "Computational Methods in Multibody Dynamics", 1992, Prentice-Hall International, p61-73.
- [5] Zeltzer D., "Motor Control Techniques for Figure Animation", IEEE Computer Graphics and Applications, 2(9), p53-59, 1982.
- [6] Denavit J, Hartenberg RS, "A Kinematic notation for lower-pair Mechanisms Based on Matrices", J. Applied Mechanics, 215(21), June 1955.
- [7] Sims K, Zeltzer D., "A Figure Editor and Gait Controller for Task Level Animation", SIGGRAPH Course Notes, 4, p164-181, 1988. Wesley, 1994.
- [8] Phillips CB, Zhao J., Badler NI, "Interactive Real-time Articulated Figure Manipulation Using Multiple Kinematic Constraints", p245-250, 1990.
- [9] F.Schfepers, R.R.Parent, W.E.Carson, S.F.May, "Anatomically-Based Modeling of the Human Musculature", Computer Graphics, SIGGRAPH '97 Conference Proceedings, p163-172, Aug. 1997.
- [10] Josie Wernecke, "The Inventor Mentor", Addison Wesley, 1994.
- [11] Open Inventor Architecture Group, "The Open Inventor C++ Reference Manual", Addison