

JProtein : Java3D 기법을 이용한 단백질 구조 뷰어

문 남 두* · 변 상 희* · 김 진 홍* · 한 인 섭** · 이 명 준***

요 약

공공 데이터베이스의 이용 가능한 단백질 데이터의 양적 증가와 함께 포스트지놈 시대가 도래되면서, 단백질의 3차 구조에 대한 연구가 활발하게 진행되고 있다. 단백질의 구조를 효과적으로 파악하기 위해서 단백질의 3차 구조를 시각화할 필요가 있다. 최근 많은 시각화 도구들이 이미 그 구조가 알려진 단백질을 시각화하기 위해 Java 기술을 이용하여 개발되었다. 본 논문에서는 새로운 단백질 시각화 도구인 JProtein 시스템에 대하여 기술한다. JProtein 시스템은 사용하기 쉽고 플랫폼에 중립적인 시각화 도구로서 작성되었다. JProtein 시스템은 Java3D 기법을 이용하여 개발되었다. Java3D는 3D 표현을 위한 프로그래밍 인터페이스를 제공하는 API이다. JProtein 시스템은 시각화된 아미노산 내의 원자들간의 각도 및 거리 정보를 제공하며, 단백질 분자구조에 대해 여러 가지 3차원 표현 모델을 지원한다. 특히, JProtein 시스템은 비동기식 스테레오 뷰와 함께 동기식 스테레오 뷰를 지원한다.

JProtein : A Protein Structure Viewer based on Java3D Technology

Nam-Doo Moon* · Sang-Hee Byun* · Jin-Hong Kim*
In-Seob Han** · Myung-Joon Lee***

ABSTRACT

Entering the post-genome era with an increasing amount of protein data available in public databases, the study of tertiary structure of proteins has been actively in progress. To analyze the structure of a protein effectively, it is necessary to visualize the tertiary structure of a protein. Recently, many visualization tools based on Java technology have been developed to visualize a protein whose structure has been known. In this paper, we describe a new protein visualization system, named *JProtein*. It is designed to be an easy-to-use, platform neutral molecular visualization tool. The *JProtein* system is developed using Java3D technology. Java3D is an API providing a programming interface for 3D representations. The system informs us the angle and the distance of the interacting atoms in amino acids which are visualized, providing several 3D representation models of a protein molecule. In particular, the *JProtein* system presents synchronous stereo view as well as asynchronous one.

키워드 : 단백질의 3차 구조(Tertiary Structure of Proteins), 시각화(Visualization), PDB(PDB), Java3D(Java3D), JProtein(JProtein), 스테레오 뷰(Stereo View)

1. 서 론

단백질은 생명현상 유지 및 생명체 내에서 발생하는 화학반응에 있어 필수적인 역할을 담당한다. 단백질의 기능을 정확하게 파악하는 것은 생명현상을 이해하는 데 매우 중요하다. 단백질의 기능은 단백질의 3차 구조에 의해 결정되는 것으로 밝혀졌다. 즉 단백질 내부 구조의 정확한 이해를 통해서 단백질의 기능을 효과적으로 파악할 수 있다. 이를 위해 최근 단백질의 3차 구조를 파악하기 위한 연구가 활

발하게 진행되고 있다. PDB(Protein Data Bank)[1, 2]는 X-ray 결정법(X-ray crystallography), NMR(nuclear magnetic response)과 같은 실험을 통해 얻어진 단백질의 3차원 구조 정보를 제공한다.

단백질의 기능을 효과적으로 파악하기 위해서는 단백질의 3차 구조를 표현하는 도구가 필요하다. 최근에 단백질 구조를 시각화하는 많은 도구들이 개발되었다. 이러한 도구들은 웹 브라우저를 통하여 원격지에서 실행될 수 있으며, 서로 다른 플랫폼에서 동작될 수 있는 장점으로 인하여 Java2D나 Java3D[3] 기술을 사용하여 개발된다.

기존의 개발된 단백질 3차 구조 뷰어 시스템으로는 Quick PDB[4], WebMol[5], Jmol[6], JMV[7], Rasmol[8] 그리고 Qmol[9] 등이 있다. Rasmol은 애플리케이션의 형태로 제공되며 범용으로 널리 사용된다. Qmol은 OpenGL 기반의 분

* 본 연구는 한국과학재단 목적기초연구(R01-2001-00535) 사업과제의 지원으로 수행되었음.

교신저자(이명준) : mjlee@ulsan.ac.kr

† 준회원 : 울산대학교 대학원 컴퓨터정보통신공학부

** 정회원 : 울산대학교 생명과학부 교수

*** 정회원 : 울산대학교 컴퓨터정보통신공학부 교수

논문접수 : 2004년 5월 18일, 심사완료 : 2004년 9월 11일

자구조 뷰어이다. 현재, Qmol은 애플리케이션 형태로 제공되며, 사용하기 쉬운 사용자 인터페이스와 빠르고 고화질을 제공하는 장점을 갖는다. 기존의 시스템들 중에서 QuickPDB, WebMol, Jmol 그리고 JMV는 웹 브라우저에서 실행된다. QucikPDB, WebMol 그리고 Jmol은 Java2D를 이용하여 개발되었으며, JMV는 Java3D를 이용하여 개발되었다. 이들 뷰어들은 실험을 통해 밝혀진 단백질의 3차원 구조정보를 바탕으로 단백질의 3차 구조를 보여준다. 그러나 기존의 뷰어들은 PDB가 제공하는 단백질의 3차원 구조정보의 내용 중 일부만을 이용하고 있어, 단백질의 구조에 대한 정보가 상세하지 않으며, 사용자가 단백질의 구조를 효과적으로 파악할 수 있도록 다양한 표현 모델을 제공하고 있지 않다. 또한 기존의 단백질 구조 시각화 도구에서 제공하는 스테레오 뷰(Stereo view) 기능은 하나의 단백질에 대해 동일한 분자 표현 모델을 적용한 구조만을 지원하기 때문에 단백질의 구조분석에서 각 분자 표현 모델이 갖는 이점을 동시에 이용할 수 없다. 단백질 구조의 시각화에 Java3D를 이용하는 경우 3D 처리속도가 문제가 될 수 있다. Java3D를 이용한 단백질 구조의 시각화 도구들은 사용자와의 상호작용에 반응하는 속도 및 다량의 데이터를 포함하는 단백질을 로딩하고 표현하는데 성능저하의 문제점에 직면하게 된다.

본 논문에서는 기존 단백질 구조 뷰어들의 기능 및 성능상의 문제점을 보완하여 단백질의 구조를 보다 효과적으로 파악할 수 있는 JProtein[10, 11] 시스템에 관하여 기술한다. JProtein 시스템은 Java3D 기술을 이용하여 개발되었으며 동기식 스테레오 뷰와 비동기식 스테레오 뷰를 지원한다. JProtein 시스템에서 스테레오 뷰 기능은 하나의 단백질에 대해 서로 다른 분자 표현 모델을 적용한 두 가지 모양의 구조를 동시에 보여준다. 동기식은 하나의 단백질에 대해 서로 다른 표현 모델을 적용하여 그 구조를 보여주며 구조의 회전, 이동, 확대 및 축소 이벤트에 대해 동기적으로 모든 표현 모델들이 바뀌게 된다. 비동기식은 단백질 구조에 대해 서로 다른 표현 모델을 적용하여 그 구조를 보여주지만 사용자의 조작에 대해서는 이벤트가 발생한 표현 모델만이 반응한다. 주어진 단백질의 구조를 동기적 혹은 비동기적으로 조작함으로써 단백질에서의 아미노산과 원자들의 위치와 특성 및 구조를 보다 효과적으로 파악할 수 있다. 또한 JProtein은 단백질내의 2차 구조(secondary structure) 및 아미노산의 특성을 서로 다른 색으로 구분함으로써 그 구조를 보다 명확하게 파악할 수 있으며, 아미노산 내에서 원자들 간의 거리 정보 및 각도 정보를 제공한다. JProtein은 일반 사용자들이 PDB 파일을 쉽게 이해하고 많은 정보를 얻을 수 있도록 PDB 파일에 대한 한글 도움말과 단백질의 요약 정보 및 PDB 사이트 링크를 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 단백질 구조

시각화 도구들의 일반적인 기능과 관련연구에 대하여 소개한다. 3장에서는 JProtein 시스템 구조와 사용자 인터페이스에 대해 살펴본다. 4장에서는 JProtein 시스템의 제공하는 시각화 기능과 구현에 관하여 기술한다. 끝으로 5장에서는 결론 및 향후 연구에 관하여 기술한다.

2. 관련 연구

단백질의 기능 및 특성은 단백질의 3차 구조에 의하여 주로 결정되므로 단백질의 3차 구조를 시각으로 확인하고자 하는 시도 또한 활발하다. 현미경이나 기타 광학장비를 이용하여 단백질의 구조를 실험실 수준에서 관찰할 수도 있지만, 대다수의 사람들은 현미경을 통해 보이는 분자의 구조를 쉽게 이해하지 못한다. 따라서 분자 모델링 도구를 이용한 분자구조의 시각화가 필요하다.

분자 모델링 도구를 이용하면 분자의 구조와 특성에 따라 다양한 색상으로 구분함으로써 사용자가 보다 쉽게 분자의 구조 및 특성을 파악할 수 있다. 이미 오래 전부터 분자들의 구조를 시각화하기 위한 연구가 진행되어 왔으며, 최근 향상된 컴퓨터의 성능과 더불어 보다 화려하고 빠르게 분자의 구조를 볼 수 있는 도구들이 개발되고 있다. 사용자는 간단한 조작을 통해 분자구조를 회전, 이동 또는 확대/축소시킴으로써 다양한 각도에서 분자를 관찰할 수 있다. 뿐만 아니라, 시각화 도구가 제공하는 기능을 통해 분자들 간의 거리, 각도와 같은 정보도 얻을 수 있다.

본 연구에서는 웹 브라우저를 통해 원격지에서 실행될 수 있으며, 서로 다른 플랫폼에서 동작될 수 있는 장점으로 인하여 Java 기술을 이용하여 개발된 시스템들만을 비교한다.

2.1 QuickPDB

QuickPDB는 SDSC(San Diego Supercomputer Center)에서 단백질의 3차원 구조를 보여주기 위해 자바 애플릿으로 개발된 분자구조 뷰어이다. QuickPDB는 Java2D 기술을 이용하여 개발되었으며, 현재 RCSB의 PDB 웹 사이트에서 서비스를 제공하고 있다. QuickPDB는 PDB 데이터베이스로부터 단백질 정보를 구하여 아미노산의 서열과 단백질의 3차원 구조를 보여준다.

QuickPDB는 PDB에서 하나의 단백질에 대하여 제공하는 모든 정보를 읽어오지 않고, 프로그램 실행에 필요한 일부 정보만을 읽어 표현하기 때문에 실행시간이 빠르다는 장점이 있다. QuickPDB는 단백질 구조를 시각화할 때 단백질의 2차 구조(나선, 사슬, 회전 등)와 아미노산의 특성(소수성, 극성, small, 프롤린, tiny, 지방성, 방향성, 양성, 음성, 전하, 글리신 등) 별로 서로 다른 색상으로 구별하여 표현함으로써 단백질의 3차원 구조를 보다 분명하게 파악할 수 있다.

2.2 WebMol

WebMol은 UCSF(University of California, San Francisco)의 Dirk Walther(Cohen Group)에 의해 개발된 Java2D 기반의 단백질 3차 구조 뷰어이다. WebMol은 PDB 파일이 포함하는 단백질의 3차원 구조정보를 분석하고 시각화하는 도구로서 애플릿과 애플리케이션 버전으로 제공된다.

WebMol은 단백질이 가진 아미노산들의 좌표정보를 이용하여 각 아미노산내의 원자들 간의 연결 관계를 3차원 공간상에 표현하며 선-틀(wire-frame), 주-사슬(main-chain), 등뼈(back-bone) 등과 같은 구조로 표현한다. 또한 WebMol은 단백질의 구조를 원자의 종류, 아미노산의 종류, 단백질 2차 구조, B-factor, 체인 그리고 HSSP 등과 같이 다양한 기준으로 분류하고 서로 다른 색으로 표현함으로써 보다 분명하게 단백질의 구조를 파악할 수 있다. WebMol 사용자는 간단한 마우스 조작을 통해 화면에 나타난 단백질의 3차원 구조모형을 회전, 이동, 확대/축소시킬 수 있다. 사용자는 마우스 조작을 통해 단백질 내에서 아미노산 원자들 간의 거리 정보와 각도 정보를 얻을 수 있으며 단백질의 특성을 간단한 도표를 통해 볼 수 있다. WebMol은 입력 데이터로 PDB 파일을 사용하며 화면에 출력된 단백질의 3차원 구조모형을 포스트스크립트 형식의 파일로 저장하는 기능을 제공한다.

2.3 Jmol

Jmol은 Java2D를 이용하여 개발된 분자구조 뷰어이다. 현재 Jmol은 공개소스 프로젝트의 형태로 개발 중에 있으며 버전 9에서 애플리케이션과 애플릿 형태로 지원되고 있다. Jmol은 원자간의 거리 정보와 각도 정보를 제공하며, 입력데이터로서 ABINIT, ACES II, CIF/mmCIF, Dalton, Ghemical, HIN, Jaguar, PDB 그리고 VASP 등 다양한 형식을 지원한다. Jmol은 시각화된 내용을 GIF, JPG, BMP, PNG 등의 다양한 형식으로 저장이 가능하다.

Jmol은 단백질 구조를 시각화할 때 그 구조 전체를 포함하는 경계 상자를 보여주고, 3차원 각 축에 대한 중심선을 보여주는 기능을 제공한다.

2.4 JMV

JMV(Java Molecular Viewer)는 NIH의 지원으로 Illinois 대학에서 BioCoRE(Biological Collaborative Research Environment)[12]라는 시스템의 일부로서 Java3D를 이용하여 개발된 분자 구조 뷰어이다. JMV는 애플릿과 애플리케이션 버전으로 제공된다. JMV는 입력데이터로 PDB 파일과 VRML(Virtual Reality Markup Language) 파일 형식을 지원한다. JMV는 단백질의 구조데이터로 원격지 파일과 로컬 파일을 모두 입력으로 사용 가능하다. JMV는 시각화된 분자 구조를 .jmv와 .jpg의 이미지 파일로 저장할 수 있다.

JMV는 효과적으로 분자의 구조를 파악하기 위해 <표 1>과 같은 표현방법을 사용한다.

<표 2> JMV의 분자 구조 표현 방법

구조	설명
선틀	각 아미노산 원자들의 인접한 원자들을 선으로 연결하는 나타낸 구조
본드	인접한 원자들을 실린더로 연결한 구조
CPK	인접한 원자들을 실린더로 연결하고 원자는 구로 표현하는 구조
VDW	원자들 각각을 본래의 반지름 크기의 구로 표현하는 구조
Licorice	원자에 상관없이 동일한 반지름 길이의 구로 표현하는 구조
튜브	각 아미노산의 C _α 를 지나는 매끄러운 곡선모양의 실린더로 표현하는 구조
Trace	각 아미노산의 C _α 들을 연결하여 직선모양의 실린더로 표현하는 구조

<표 2>에서는 JProtein 시스템과 기존 시스템과의 비교되는 점을 보여주고 있다. JProtein 시스템은 Java3D를 이용하여 단백질의 구조를 다양한 표현 모델을 적용하여 보여준다. 사용자는 JProtein 시스템의 스테레오 뷰 기능을 통해 동일한 단백질에 대해 서로 다른 표현 모델을 적용한 분자구조를 볼 수 있다. 스테레오 뷰 기능은 동기식과 비동기식으로 구분된다.

<표 2> 단백질 3차 구조 시각화 도구 비교

	JProtein	JVM	QuickPDB	WebMol	Jmol
개발언어	Java3D	Java3D	Java2D	Java2D	Java2D
이벤트처리 (회전, 이동, 확대/축소)	지원	지원	지원	지원	지원
구조모형	아미노산별	지원	지원	지원	지원
	2차구조별	지원	지원	지원	지원
	주사슬별	지원	지원	지원	지원
	등뼈	지원	지원	지원	지원
	기타 (본드, CPK)	지원	지원		지원
표현모형	스테레오 (동기식)	서로 다른 표현 모델지원	동일한 표현 모델만 지원	동일한 표현 모델만 지원	동일한 표현 모델만 지원
	스테레오 (비동기식)	지원			
거리/각도 측정	지원			지원	지원
파일로 저장		지원			지원

* 공백은 지원하지 않음을 의미

단백질 3차 구조 뷰어의 대표적인 JVM의 경우 PDB ID 가 1SYN를 시각화하기 위해 다음의 메모리량을 요구한다. 1SYN은 4300개의 원자수를 갖는다.

- 요구되는 메모리량 = 4300개의 원자 객체 + 4300개의 변형그룹(TransformGroup) 객체 + 1개의 최상위 변형 그룹 객체

JProtein의 경우 1SYN을 시각화하기 위해 다음의 메모리 양을 요구한다.

- 요구되는 메모리량 = 4300개의 원자 객체 + 4300개의 그룹(Group) 객체 + 1개의 최상위 변형그룹 객체

하나의 변형그룹 객체는 대략 1056바이트를 요구하며, 하나의 그룹 객체는 520바이트를 요구한다. 따라서 원자의 수가 많은 단백질의 구조를 표현할 때 JProtein의 경우 JVM에 비해 1/2분의 메모리량을 사용한다.

<표 3>은 동일한 조건에서 변형그룹 객체의 사용수에 따른 장면그래프에 대한 소요된 로딩시간을 보여준다. 따라서 변형그룹 객체수를 적게 사용할수록 장면그래프의 로딩 시간이 적은 것을 알 수 있다.

<표 3>

변형그룹 객체수	장면그래프의 로딩시간
1	187 밀리초
10	218 밀리초
100	234 밀리초
250	266 밀리초
500	391 밀리초
1000	593 밀리초
10000	1985 밀리초

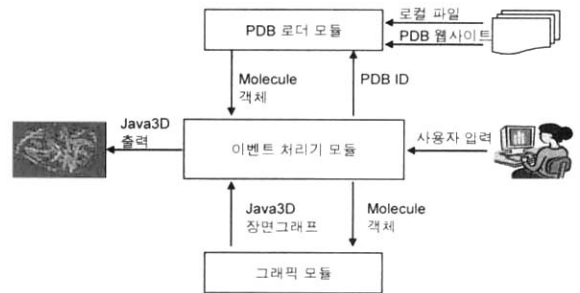
3. 시스템 구조

3.1 JProtein 시스템 구조

JProtein 시스템은 (그림 1)과 같이 크게 이벤트 처리기 모듈, PDB 로더 모듈, 그래픽 모듈 등으로 구성된다. 이벤트 처리기 모듈은 입력과 출력을 처리하며 다른 컴포넌트 모듈과의 객체 전달을 담당한다. 사용자가 보고자 하는 단백질의 PDB ID를 입력하면 PDB 로더 모듈에 의해 로컬 구조 파일을 읽거나 지정된 PDB 웹 사이트로부터 구조 파일을 다운로드하여 단백질 구조를 시각화한다.

PDB 로더 모듈은 다수의 클래스로 구성된 Molecule 객체를 생성한다. PDB 로더 모듈에 의해 읽혀진 PDB 정보는 레코드별로 ATOM, HELIX, SHEET, TURN, CHAIN 등의 클래스를 사용하여 자바 객체로 표현된다. Molecule 객체는 ATOM, HELIX, SHEET, TURN, CHAIN 등의 객체로 구성되었다. Molecule 객체는 그래픽 모듈로 전달되는데 그래픽 모듈은 사용자가 선택한 표현 모델을 적용하여 3D 모델을 표현하는 역할을 수행한다. 3D 모델은 Java3D

장면 그래프로 표현되고 Java3D 엔진에 의해 렌더링 된다. Java3D 장면 그래프는 표현하고자 하는 객체들 사이의 공간적인 관계 정보를 가지는 계층적 자료구조이다.

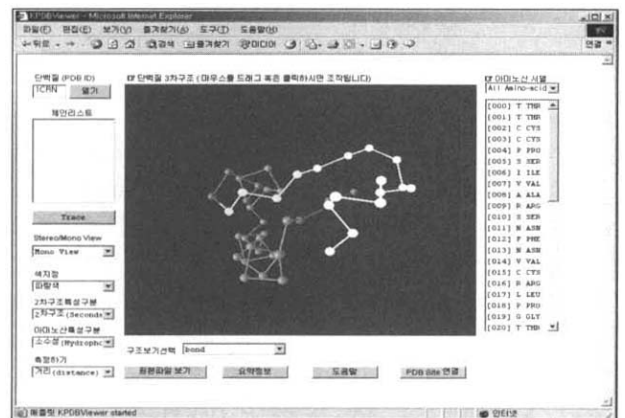


(그림 1) JProtein 시스템 구조

JProtein은 입력데이터로 원격의 URL 혹은 로컬 파일 시스템에 위치한 PDB 파일을 사용하며, 읽어들인 자료는 PDB에서 구분한 각 레코드별로 자바 객체에 저장된다. JProtein은 다양한 표현 모델을 적용하여 단백질 3차 구조를 보여주며, 효과적으로 단백질 구조를 파악하기 위해 단백질의 특성을 아미노산, 원자, 2차 구조 등으로 분류하여 서로 다른 색으로 구별하여 보여준다. 또한 원자 간의 거리, 각도 등을 측정할 수 있는 기능도 함께 제공한다. JProtein의 도움말 기능은 PDB 파일의 레코드의 의미와 형식을 설명함으로써 사용자가 PDB 파일의 정보를 보다 쉽게 알 수 있다.

3.2 사용자 인터페이스

JProtein 시스템의 사용자 인터페이스는 (그림 2)와 같다. 단백질 구조를 시각화하는 도구는 분자구조 파악이 용이하고 사용이 편리해야 한다. JProtein은 한글화된 인터페이스를 제공함으로써 전문적인 지식을 가지지 않은 사용자도 쉽게 단백질 구조를 파악하고 시스템을 사용할 수 있다.



(그림 2) JProtein 시스템의 사용자 인터페이스

4. 시각화

JProtein 시스템은 단백질의 구조를 보다 효과적으로 분석하고 파악하기 위해 다양한 분자구조 표현 모델을 사용하여 그 구조를 보여준다. 분자의 구조를 시각화하기 위해 선(line), 구(sphere), 실린더(cylinder) 등이 이용된다. JProtein 시스템은 이러한 시각적 도구를 이용하여 3차원 공간상에 단백질 구조를 표현한다.

JProtein 시스템은 단백질의 3차원 구조설명을 위해 등뼈 모델, 선-틀 모델, 콩과 막대 모델 그리고 튜브 모델 등 다양한 표현 모델들을 지원한다. 각 표현 모델은 단백질 구조를 시각화하는데 있어서 다른 표현 모델과 비교되는 장단점을 갖는다.

단백질 구조의 시각화에 Java3D를 이용하는 경우 3D 처리속도가 문제가 될 수 있다. Java3D를 이용한 단백질 구조의 시각화 도구들은 사용자와의 상호작용에 반응하는 속도 및 다량의 단백질 데이터를 로딩하고 표현하는데 성능저하의 문제점에 직면하게 된다. 따라서 JProtein 시스템은 이러한 상황을 개선하기 위해 두 가지 방법을 적용하였다. 첫 번째 방법은 사용자와의 상호작용 반응속도를 향상시키기 위해 변형그룹(TransformGroup) 객체의 생성개수를 줄이는 것이다. 단백질 구조의 회전, 이동, 확대 및 축소를 위해서 Java3D에서 제공하는 TransformGroup 클래스를 이용하게 된다.

장면 그래프는 표현하고자 하는 객체들 사이의 공간적인 관계 정보를 가지는 계층적 자료구조이다. 단백질 구조를 표현하기 위한 장면 그래프 구성시 다수의 변형그룹 객체의 생성은 사용자와의 반응속도를 느리게 한다. JProtein 시스템은 아미노산의 원자 레벨에서 변형그룹 객체를 생성하지 않고 아미노산 레벨에서 변형객체를 생성함으로써 변형객체의 생성을 줄였다. 두 번째 방법은 각 아미노산의 $C\alpha$ 를 표현하기 위해 다수의 동일한 모양을 갖는 Shape3D 객체를 생성하지 않고 하나의 도형객체만 생성하도록 하였다. 다수의 동일한 모양을 갖는 Shaped3D 객체의 생성을 제거하여 불필요한 메모리 사용을 줄임으로써 다량의 아미노산을 포함하는 단백질의 구조를 표현할 수 있다.

4.1 분자구조 표현 모델

4.1.1 등뼈 모델

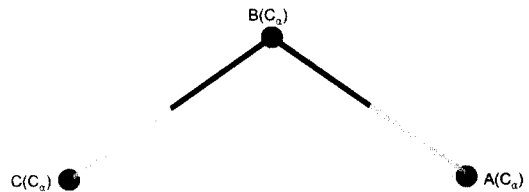
등뼈 모델은 분자내의 각 아미노산의 $C\alpha$ (alpha-carbon) 원자를 직선으로 연결한 구조를 시각화한다. 등뼈 모델은 복잡한 단백질의 3차원 구조를 간략화시킴으로써 한 눈에 단백질의 구조를 파악하기에 효과적이다.

JProtein 시스템의 등뼈 모델은 다음의 과정을 통해 단백질의 구조를 시각화한다.

- ① PDB 파일로부터 3차원 구조정보를 읽어 ATOM 객체

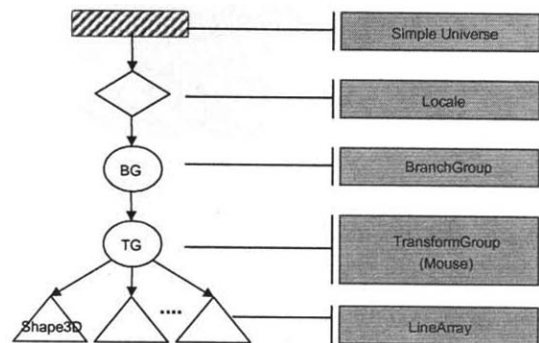
에 저장한다. 저장된 각 아미노산 내의 원자의 3차원 좌표 가운데 $C\alpha$ 원자의 좌표를 얻어온다.

- ② $C\alpha$ 는 아미노산내의 여러 원자들 가운데 하나로서 아미노산의 대표성을 갖는다. 따라서 $C\alpha$ 를 한 아미노산의 중심좌표로 정한다.
- ③ 두 아미노산의 경계지점을 명확히 하기 위해 두 아미노산($C\alpha$)을 연결한 직선의 가운데 지점을 경계지점으로 정한다.
- ④ 결과적으로 하나의 아미노산을 표현하기 위해서는 두 개의 선이 필요하다. 즉, 한 아미노산의 영역은 아미노산($C\alpha$)을 기준으로 이전 아미노산과의 거리의 중간 지점에서, 다음 아미노산까지의 중간지점까지이다. 다음의 (그림 3)은 각 아미노산($C\alpha$)을 연결하여 등뼈 구조로 표현한 예제이다.
- ⑤ 각 아미노산($C\alpha$)의 영역은 두 개의 선으로 표현되며, 아미노산 좌표와 더불어 계산된 아미노산간의 중간지점 좌표를 저장하고 이후에 각 아미노산 별로 색을 구분하여 표현할 때 사용된다.



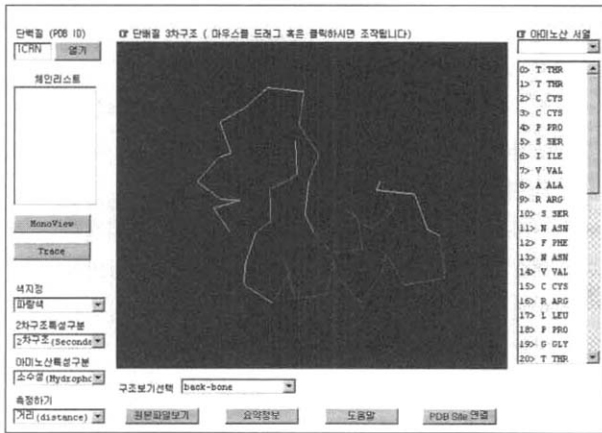
(그림 3) 등뼈 모델

다음의 (그림 4)는 등뼈 모델을 적용하여 단백질 구조를 Java3D로 표현하기 위한 장면그래프이다. 선으로 표현된 등뼈 구조를 마우스를 이용하여 렌더링하기 위해서 모든 선들을 하나의 변형그룹(TransformGroup)의 자식노드로 추가하였다. 또한 변형그룹 객체에 마우스 이벤트를 등록하여 간단한 마우스 조작을 통해 등뼈 구조를 회전, 이동, 확대 및 축소가 가능하도록 하였다.



(그림 4) 등뼈 모델 구조의 장면 그래프

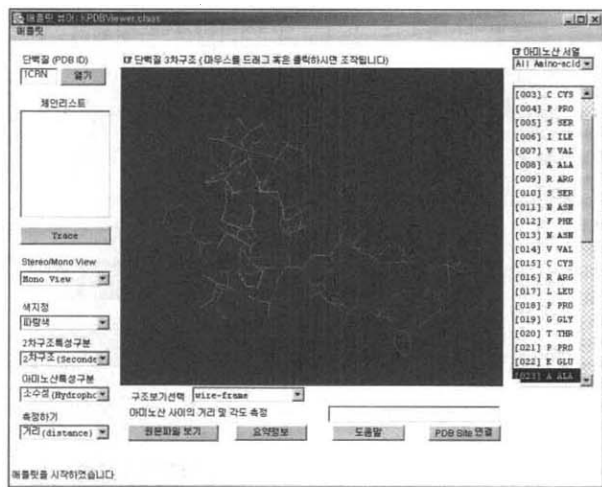
(그림 5)는 단백질의 2차 구조 별로 색을 지정하고 등뼈 모델을 적용하여 표현된 단백질 구조이다.



(그림 5) 2차 구조별 등뼈 구조

4.1.2 선-틀 모델

등뼈 모델은 아미노산(C_α)을 연결하여 단백질을 단순하게 보여주는 반면, 선-틀 표현 모델은 (그림 6)과 같이 단백질의 구조를 보다 자세하게 보여주기 위해 아미노산의 내부의 원자들의 관계를 선으로 연결하여 표현하는 구조이다. 선-틀 모델은 아미노산 내의 각 원자들의 연결 상태를 선으로 표현한다. JProtein 시스템은 PDB 파일에 포함된 원자들 간의 연결정보를 이용한다. JProtein 시스템의 선-틀 모델은 등뼈 모델을 표현하는 방법과 동일하게 하나의 원자에 소속된 두 직선 중 첫 번째는 이전 원자와의 중간지점까지를, 두 번째 직선은 다음 원자와의 중간지점까지를 직선으로 나타낸다.



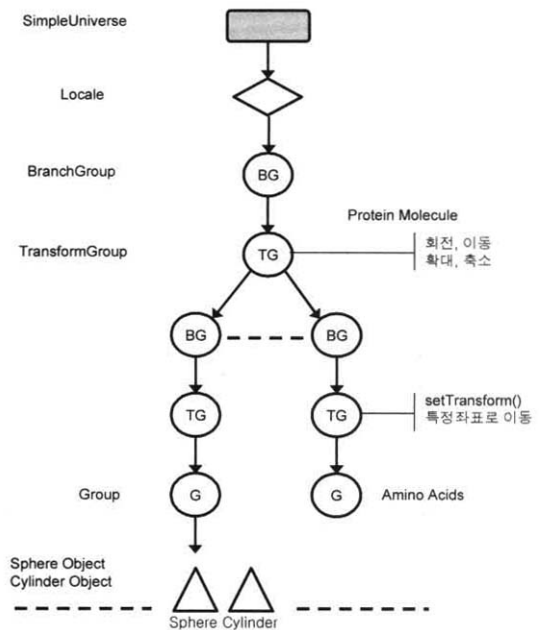
(그림 6) 선-틀 구조

4.1.3 공과 막대 모델

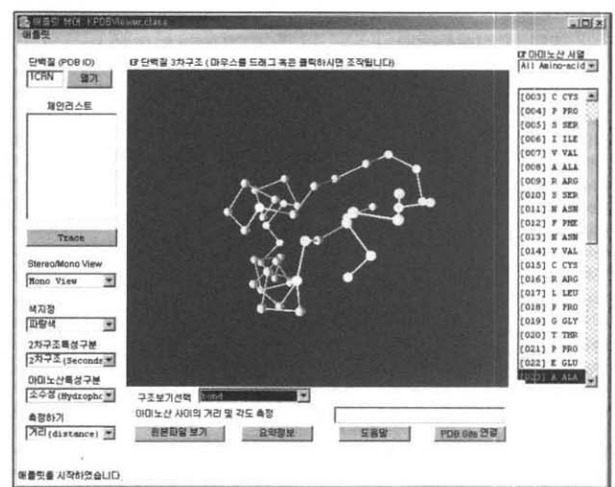
그래픽 기반 기술과 하드웨어의 성능향상으로 단백질의

구조를 보다 향상된 모양으로 보여줄 수 있게 되었다. 선으로 단백질의 구조를 시각화하는 것에 비해 체적을 가진 모양의 구나 실린더를 이용하면 분자의 부피를 시각화하는데 유용하다.

공과 막대 모델은 등뼈 모델로 표현된 구조보다 실제 단백질의 구조에 가까운 모양으로서, 각 아미노산(C_α)을 체적이 있는 구로 표현하고 구와 구를 체적이 있는 실린더로 연결한 구조이다. 따라서 공과 막대 모델을 적용하여 단백질 구조를 표현하기 위해서는 구와 실린더를 표현하는 메소드 구현이 필요하다.



(그림 7) 공과 막대 구조의 장면 그래프



(그림 8) 공과 막대 구조

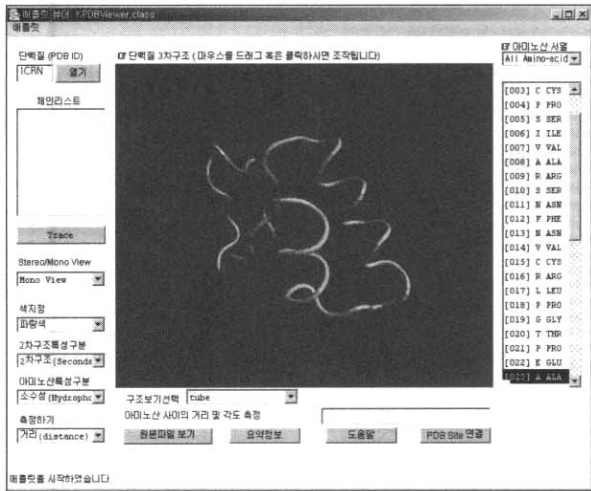
공 모양의 구조는 Java3D의 Sphere 클래스를 이용하여 구현할 수 있으며, 막대 모양의 구조는 Cylinder 클래스를

이용하여 구현할 수 있다. 하지만 Sphere 클래스는 구의 크기를 그리는 API만 제공할 뿐 구를 3차원 공간상의 특정 좌표에 그리는 방법은 제공하지 않는다. 마찬가지로 Cylinder 클래스는 정해진 반지름과 높이를 가진 실린더를 그리는 API만을 제공한다. 따라서 JProtein 시스템은 변환 그룹(TransformGroup)에 구와 실린더를 추가하고, 변환 그룹에 지정된 Transform3D 객체의 setTransform() 메소드를 이용하여 구와 실린더를 3차원 공간상의 특정 좌표로 이동시킴으로써 단백질 구조를 표현하였다. (그림 7)은 공과 막대 모델을 표현하기 위한 장면그래프를 보여준다.

(그림 8)은 PDB ID가 1crn인 단백질에 대해 공과 막대 표현 모델을 적용한 모습을 보여준다.

4.1.4 튜브 모델

(그림 9)와 같이 튜브 구조는 아미노산의 C α 를 연결하는 단순한 직선 모양의 등뼈 구조와 달리, 각 C α 를 지나는 부드러운 곡선모양의 채적이 있는 원통으로 단백질의 구조를 나타낸다. 튜브 모양의 구조는 Java3D에서 제공하는 SplineInterpolation 클래스와 Cylinder 클래스를 이용하여 구현하였다.



(그림 9) 튜브 구조

4.2 스테레오 뷰

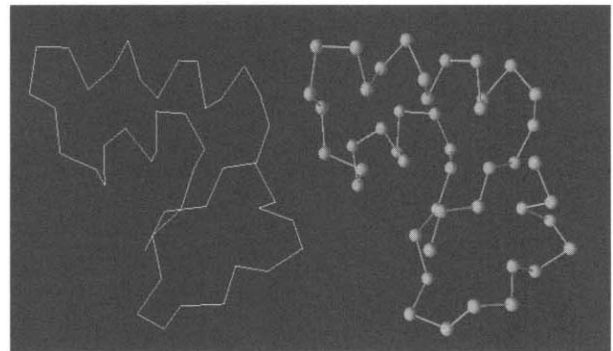
JProtein 시스템은 단백질 3차 구조의 효과적인 분석을 위해 스테레오 뷰 기능을 제공한다. 스테레오 뷰의 기본적인 기능은 하나의 단백질 구조에 대해 두 개의 시각화된 장면을 보여주는 것이다. 기존 시스템들이 제공하는 스테레오 뷰 기능은 하나의 단백질에 대해 동일한 분자구조 표현 모델을 적용한 형태로만 시각화하며 사용자의 조작에 대해 동기적으로 동작함으로써 두 장면 모두 동일한 모습만을 보여주게 된다.

JProtein 시스템은 기존의 단백질 구조 뷰어들이 제공하는 스테레오 뷰 기능을 확장하여 동기식과 비동기식을 지

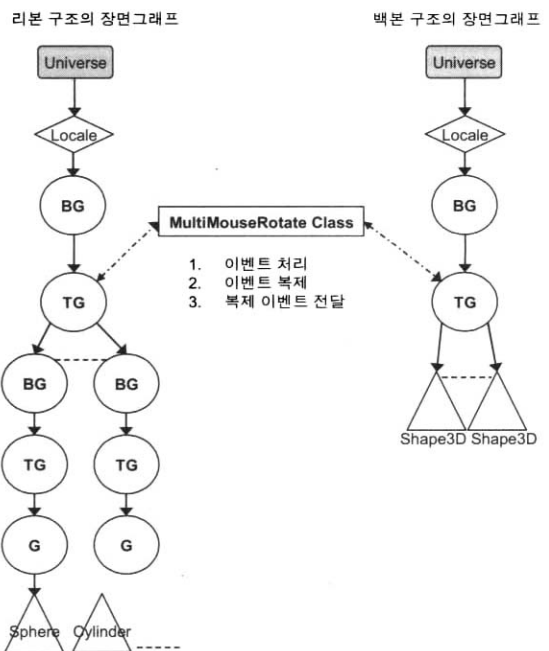
원한다. JProtein 시스템의 스테레오 뷰 기능은 하나의 단백질에 대해 동시에 두 가지 구조로 보여준다. 예를 들어, 한쪽에는 등뼈 모델을 적용한 구조로 다른 면에는 선-틀 모델을 적용한 구조로 동시에 보여줄 수 있다. 비동기식은 회전, 이동, 확대 및 축소 등의 이벤트에 대해 이벤트가 발생한 구조만이 반응하며 동기식은 한 구조의 이벤트가 다른 구조에도 동일하게 적용된다.

4.2.1 동기식

동기식 스테레오 뷰 기능은 동일한 단백질에 대해 서로 다른 표현 모델을 적용한 구조를 동시에 보여줌으로써 보다 효과적으로 그 구조를 파악할 수 있다. JProtein 시스템은 동기식 스테레오 뷰 상태에서 (그림 10)과 같이 사용자가 입력한 마우스 이벤트에 대해 두 표현 모델이 동일하게 반응하도록 구현되었다.



(그림 10) 동기식 스테레오 뷰



(그림 11) 동기식 스테레오 뷰를 위한 장면 그래프

동기식 스테레오 뷰를 보여주기 위한 장면 그래프는 (그림 11)와 같다. 스테레오 뷰 기능을 제공하기 위한 장면 그래프는 (그림 11)에서 보는 바와 같이 동일한 단백질에 대해 서로 다른 표현 모델을 적용한 두 개의 장면 그래프를 포함한다. 동기식 스테레오 뷰 기능을 제공하기 위해 한 쪽 구조에서 발생한 이벤트를 처리하고, 이를 복사하여 다른 쪽의 구조에 전달한 후 복사된 이벤트가 처리되도록 MultiMouseRotate 클래스를 구현하였다.

다음은 동기식과 비동기식 스테레오 뷰의 이벤트 처리를 위한 알고리즘을 보여준다.

• 동기식 스테레오 뷰의 이벤트 처리 :

- ① 각 표현모델마다 마우스 클릭과 같은 이벤트를 처리할 이벤트 핸들러 객체를 생성한다.
- ② 각 이벤트 핸들러 객체는 동기식으로 동작할 표현모델 객체들에 대한 참조를 저장할 ArrayList 유형의 객체를 생성한다.
- ③ 각 이벤트 핸들러는 반복문을 사용하여 ArrayList 객체에 표현모델 객체들에 참조를 저장한다.
- ④ 이벤트 핸들러의 add 메소드와 remove 메소드를 이용하여 표현모델 객체들에 대한 참조를 동적으로 추가하거나 삭제할 수 있다. ArrayList 객체에 저장되는 표현모델 객체는 Component 클래스의 하위 클래스로부터 생성된 객체이다.
- ⑤ processMouseEvent 메소드는 이벤트 발생시 이벤트 처리를 위한 메소드이며, 이벤트 핸들러 클래스에서 재정의(override)하여 사용된다. processMouseEvent 메소드는 내부적으로 다음의 작업을 수행한다.
 - 이벤트가 발생하면 부모클래스(MouseRotate)의 processMouseEvent 메소드를 호출하여 자신의 이벤트를 처리한다.
 - if문을 이용하여 발생한 이벤트의 유형이 복제된 이벤트 유형인지 파악하고 복제된 이벤트가 아닌 경우 아래의 '다' 과정을 계속해서 수행하고 복제된 이벤트인 경우에는 이벤트처리를 완료한다.
 - 이벤트 핸들러는 자신의 ArrayList 객체로부터 동기식으로 동작할 각 표현모델 객체들에 대한 참조를 구한다. 이후, 이벤트 핸들러는 자신에게 발생한 이벤트와 동일한 이벤트를 복사한다. 복사된 이벤트는 원본 이벤트와 동일한 발생시간, 발생 좌표, 클릭수, 이벤트ID 정보를 갖으며 이벤트가 발생한 근원지에 대한 정보만이 서로 다르다. 이벤트 핸들러는 복사된 이벤트를 동기식으로 동작할 이벤트 핸들러에게 dispatchEvent 메소드를 이용하여 전달한다. 복제된 이벤트를 전달받은 이벤트 핸들러는

위에서 언급한 처리과정의 '가'와 '나'를 수행한다.

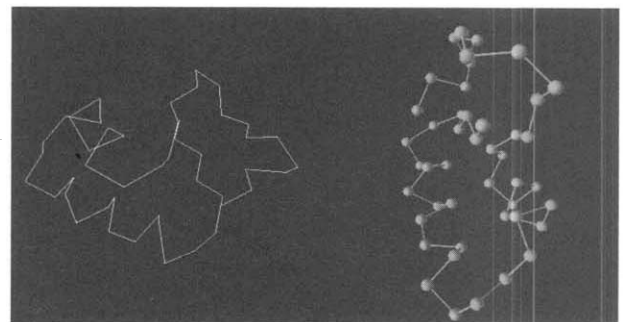
• 비동기식 스테레오 뷰의 이벤트 처리 :

비동기식 스테레오 뷰의 이벤트 처리는 동기식 스테레오 뷰의 이벤트 처리보다 간단하다.

- ① 각 표현모델마다 이벤트를 처리할 이벤트 핸들러 객체를 생성한다.
- ② 각 이벤트 핸들러 객체는 동기식으로 동작할 표현모델 객체들에 대한 참조를 저장할 ArrayList 객체를 생성한다.
- ③ 이벤트 핸들러의 add 메소드와 remove 메소드를 이용하여 동적으로 표현모델 객체들에 대한 참조를 추가하거나 삭제할 수 있다. ArrayList 객체에 저장되는 표현모델 객체는 Component 클래스의 하위 클래스로부터 생성된 객체이다.
- ④ ArrayList 객체는 동기식으로 동작할 표현모델 객체들에 대한 참조를 저장하기 위한 기억장소이다. 따라서, ArrayList 객체에서 비동기식으로 동작하고자 하는 표현모델 객체에 대한 참조를 삭제함으로써 참조가 삭제된 표현모델 객체는 이벤트에 대해 비동기적으로 동작하게 된다.
- ⑤ 이벤트 발생후의 처리는 동기식 스테레오 뷰의 이벤트 처리과정의 ⑤와 같다.

• 비동기식

비동기식 스테레오 뷰 기능은 동일한 단백질에 대해 서로 다른 표현 모델을 적용한 구조를 동시에 보여줌으로써 보다 효과적으로 그 구조를 파악할 수 있다. JProtein 시스템은 비동기식 스테레오 뷰 상태에서 (그림 12)와 같이 사용자가 입력한 마우스 이벤트에 대해 이벤트가 발생한 구조만이 이벤트에 반응하도록 구현되었다.



(그림 12) 비동기식 스테레오 뷰

5. 결 론

단백질의 기능이 단백질의 3차 구조에 의해 결정된다는

것이 밝혀짐으로 인해 단백질의 3차 구조에 대한 연구가 활발하게 진행되고 있다. 단백질의 구조를 효과적으로 파악하기 위해서는 단백질의 구조를 시각화하는 도구가 필요하다. 다수의 단백질 시각화 도구들이 웹 브라우저를 통해 원격지에서 실행될 수 있으며, 서로 다른 플랫폼에서 동작될 수 있는 장점으로 인하여 Java 기술을 이용하여 새롭게 개발되고 있는 추세이다.

본 논문에서는 단백질의 구조정보를 포함하고 있는 PDB 파일을 분석하여 단백질의 3차 구조를 시각화하는 JProtein 시스템에 관하여 기술하였다. JProtein 시스템은 Java3D 기술을 이용하여 개발되었으며 애플릿으로 동작한다. JProtein 시스템은 원격의 URL 또는 로컬 시스템의 PDB 파일을 입력데이터로 처리하며 PDB 데이터의 효율적인 사용을 위해 읽어진 구조 정보를 자바객체에 저장하여 관리한다. JProtein 시스템은 단백질의 구조를 단백질 내의 2차구조 및 아미노산의 특성별로 서로 다른 색을 지정하여 표현함으로써 그 구조를 보다 명확하게 파악할 수 있다. 또한 JProtein 시스템은 시각화된 단백질 구조에서 사용자의 간단한 마우스 조작만으로 아미노산 내의 원자들 간의 거리 정보 및 각도 정보를 제공한다.

JProtein 시스템은 단백질의 3차 구조를 시각화하기 위해 등뼈 구조, 선-틀 구조, 공과 막대 구조, 튜브 구조 등 다양한 분자구조 표현 모델을 적용하여 표현한다. JProtein은 동일한 단백질에 대해 서로 다른 표현 모델을 적용하여 동시에 보여주는 스테레오 뷰 기능을 제공함으로써 단백질의 3차 구조를 보다 효과적으로 파악할 수 있다. JProtein은 동기식과 비동기식의 스테레오 뷰 기능을 제공한다. JProtein은 Java3D 장면 그래프 구성시 3D 객체 생성을 최소화하고 중복정보를 제거함으로써 Java3D가 초래할 수 있는 불필요한 메모리 사용과 사용자의 상호작용 속도저하를 개선하였다. PDB 파일은 다량의 데이터를 포함하고 있지만, 사용자가 이해하기에 그 구조가 비교적 복잡하다. 따라서, JProtein 시스템은 도움말 기능을 통해 PDB 파일의 내용을 쉽게 파악할 수 있도록 레코드별 의미정보와 각 단백질에 대한 요약정보를 제공한다.

현재 JProtein 시스템은 입력데이터로 PDB 파일을 지원하고 있으며 향후 다양한 형식의 3차원 단백질 구조정보 파일을 지원할 계획이다.

참 고 문 헌

[1] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T.N.Bhat, Helge Weissig, Ilya N. Shindyalov and Philip E. Bourne, "Protein Data Bank," *Nucleic Acids Research*, Vol.28, No.1, pp.235-242, 2000.

[2] John Westbrook, Zukang Feng, Shri Jain, T. N. Bhat, Narmada Thanki, Veerasamy Ravichandran, Gary L. Gilliland, Wolfgang Bluhm, Helge Weissig, Douglas S. Greer, Philip E. Bourne and Helen M. Berman, "The Protein Data Bank: unifying the archive," *Nucleic Acids Research*, Vol.30, No.1, pp.245-248, 2002.

[3] Java 3D API Specification, Version 1.3, June 2002, http://java.sun.com/products/java-media/3D/forevelopers/J3D_1_3_API/j3dguide/index.html.

[4] QuickPDB : <http://cl.sdsc.edu/QuickPDB.html>.

[5] D. Walther, "WebMol - a Java based PDB viewer," *Trends Biochem Sci*, 22, pp.274-275, 1997, www.cmpchem.ucsf.edu/webmol.pl.

[6] Jmol : <http://jmol.sourceforge.net/>.

[7] M. Bach, R. Brunner, J. Stone, K. Vandivort, "JMV User's Guide," Verson 0.85, June 25, 2002, <http://www.ks.uiuc.edu/Development/jmv/>.

[8] Sayle, R., A. Bissell, "RasMol: A Program for Fast Realistic Rendering of Molecular Structures With Shadows," in *Proceedings of the 10th Eurographics UK '92 Conference*, University of Edinburgh, Scotland, April, 1992.

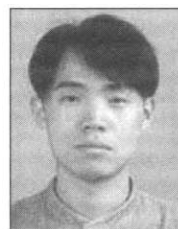
[9] Gans J, Shalloway D, "Qmol : A program for molecular visualization on Windows based PCs," *Journal of Molecular Graphics and Modelling*, 19, pp.557-559, 2001.

[10] 변상희, 김진홍, 문남두, 이명준, "KPDBViewer : Java3D를 이용한 PDB 뷰어 개발", *한국정보과학회 '2003 가을 학술논문집*, Vol.30, No.2, pp.832-834, 2003.

[11] 구형서, 주영준, 박양수, 이명준, "PDB 한글 뷰어", *한국정보과학회 '2002 가을 학술발표논문집*, Vol.29, No.2, pp.373-375, 2002.

[12] Milind Bhandarkar, Gila Budescu, William Humphrey, Jesus A. Izaguirre, Sergei Izrailev, Laxmikant V. Kale, Dorina Kosztin, Ferenc Molnar, James C. Phillips, and Klaus Schulten. *BioCoRE: A collaboratory for structural biology*. In Agostino G. Bruzzone, Adelinde Uchrmacher, and Ernest H. Page, editors, *Proceedings of the SCS International Conference on Web-Based Modeling and Simulation*, pp.242-251, San Francisco, California, 1999.

문 남 두



e-mail : ndmoon@dreamwiz.com

1997년 울산대학교 전자계산학과 공학사

1999년 울산대학교 컴퓨터정보통신공학부 공학석사

2003년 울산대학교 컴퓨터정보통신공학부 공학박사

관심분야 : 웹기반정보시스템, 협업지원시스템, 분산프로그래밍 등



변 상 희

e-mail : heeya@ulsan.ac.kr
2003년 울산대학교 컴퓨터정보통신공학과
공학사
2003년~현재 울산대학교 대학원 컴퓨터
정보통신공학부 석사과정
관심분야 : 생물정보학, 협업지원 시스템,
웹 프로그래밍 등



김 진 홍

e-mail : avenue@ulsan.ac.kr
1999년 울산대학교 전자계산학과 공학사
2001년 울산대학교 대학원 컴퓨터정보통신
공학부 공학석사
2001년~현재 울산대학교 대학원 컴퓨터
정보통신공학부 박사과정

관심분야 : 협업지원시스템, 생물정보학, 웹 프로그래밍, 프로그
래밍 언어 등



한 인 섭

e-mail : hanis@ulsan.ac.kr
1983년 서울대학교 이학사
1988년 조오지아대학교 이학석사
1988년 조오지아대학교 이학박사
1988년~1991년 Univ. of California, Irvine
(post-doc)

1991년~현재 울산대학교 교수
관심분야 : 면역학, 유전체학, 생물정보학 등



이 명 준

e-mail : mjlee@ulsan.ac.kr
1980년 서울대학교 수학과 학사
1982년 한국과학기술원 전산학과 석사
1991년 한국과학기술원 전산학과 박사
1982년~현재 울산대학교 컴퓨터정보통신
공학부 교수

1993년~1994년 미국 버지니아대학 교환교수
관심분야 : 웹기반 정보시스템, 프로그래밍언어, 분산 프로그래밍,
생물정보학