거대분자 모델의 적응형 LOD 렌더링 기법

이준[0†] 박성준[††] 김지인[†††]
[†] 건국대학교 컴퓨터 정보통신공학과,
[††] 호서대학교 게임공학과,
[†††] 건국대학교 인터넷미디어공학과
[†] junlee@konkuk.ac.kr, [††] sjpark@office.hoseo.ac.kr, [†††] jnkm@konkuk.ac.kr

Adaptive LOD Rendering for Large-Scale Molecular Models

Jun Lee[0†] Sungjun Park[††] Jee-In Kim [†††]
[†] Computer Science & Engineering, Konkuk University,
[†] Game Engineering, Hoseo University,
[†] Internet & Multimedia, Konkuk University

요 약

정보생물학 분야에 있어서 분자 구조를 3차원으로 렌더링하여 보여주는 것은 매우 중요한 작업이다. 특히 분자의 표면 렌더링은 분자의 3차원 구조 분석 등에 중요하게 사용된다. 그러나 분자 표면 렌더링을 수행하기 위해서는 많은 양의 폴리곤이 필요하게 된다. 대장균 바이러스와 같은 분자량이 많은 거대 분자를 자연스럽게 렌더링 하기 위해서는 고성능이며 고가의 그래픽 전용 워크스테이션을 사용해야 한다. 본 논문에서는 PC급 시스템에서도 거대 분자를 무리 없이 렌더링 할 수 있는 효율적인 알고리즘을 제안 하였다. 제안하는 알고리즘은 사용자의 시점에서 최적의 성능 및 시각적인 기여를 할 수 있는 적응형 상세 단계 렌더링을 수행 한다. 제안된 알고리즘을 사용하여 거대 분자 모델의 렌더링시 대화식 프레임 수준이상의 성능향상을 보이며, 또한 시각적으로도 분자 모델이 가진 중요한 기하학적인 특성을 유지 할 수 있다.

1. 서론

분자 표면 렌더링(Molecular Surface Rendering)[1]은 분자의 반지름 정보를 이용하여 분자 구조의 전체적인 3차원 모습을 만들어 내는 것을 말한다. 이러한 분자 표면 렌더링은 분자 구조의 분석, 단백질 간의 상호작용 분석, 신약 물질 설계 등에 중요하게 사용되는 분자 모델링의 한 연구 분야이다.

표면 렌더링을 사용하여 분자 구조를 3차원으로 표현하여 디스플레이 할 경우, 분자량이 증가할수록 각 분자를 구성하고 있는 폴리곤(Polygon)의 개수가 급격히 늘어나기 때문에, 거대 분자 구조를 실시간으로 관측하는 것은 매우 어렵다. 이러한 문제점을 해결하기 위해, 좀더 빠른 렌더링 속도를낼 수 있는 고성능이며 고가의 그래픽 전용 워크스테이션을 구입하여 실험을 하고 있다. 또 다른 방법으로는 물체를 구성하는 폴리곤의 수를 줄이거나, 혹은 폴리곤 대신 점으로 분자 모델을 렌더링 하는 방법을 사용하기도 한다[12].

본 논문에서는 PC를 사용한다. 그리고, GPU(Graphic Processin g Unit)를 장착한 일반 그래픽 카드를 사용하며, 많은 양의 폴리곤 수를 급격히 줄일 수 있는 알고리즘을 제안하며, 거대 분자를 표면 렌더링 하는데 있어서 최적화된 렌더링 기법을 소개한다.

본 논문에서 제안하는 방법은 실시간 적응형 상세 단계 렌더링 기법(Real Time Adaptive Level of Detail Rendering)이다. 이 방법은 PC급에서 거대 분자 모델을 렌더링할 경우 대화식 프레임 속도를 보장하도록 설계 되어 있다. 렌더링을 하는 시스템의 FPS(Frame Per Second) 가 일정 기준의 대화식 프레임 속도보다 느리다면, 분자 모델의 메시를 단순화 하여 렌더링을 수행 하게 하는 Level of Detail 렌더링을 지원 한다. 이를 위하여 분자 모델을 단순화 시킨 모델들을 렌더링에 사용하게 된다. 또한 단순화 하는 분자 모델을 분자 구조의 시각적 및 기하학적인 특징에 적합하게 단순화를 수행 하게 한다. 이를 통하여 사용자는 단순화된 메시를 사용하여 렌더링을 하고 관측을 하여 대화식 이상의 프레임 속도를 보장 받을 수 있으며, 원본 모델과 큰 차이가 나지 않는 메시의 렌더링이 가능하다. 본 알고리즘에서는 대화식 프레임 이상의 속도 보장, 사용자 시점과 시각적인 정보의 유지를 위한 평가함수를 적용하여 실시간 렌더링에 적용하였다.

본 논문에서는 제안하는 알고리즘의 효율성을 검증하기 위해 본 연구팀에서 개발한 가상현실 기반의 분자 모델링 시스템인 VRMMS (Virtual Reality Molec

ular Management System) [2,3]와 기존 렌더링 방식을 사용한 시스템과 렌더링 속도를 비교 분석 하였다. 실험 데이터는 대장균 바이러스, ATPase 등 실제 시뮬레이션 실험에 사용 되는 거대 단백질들을 사용하였다.

본 논문은 다음과 같이 구성된다. 먼저 2장 관련연구에서는 거대 분자 모델의 실시간 렌더링을 위한 상세 단계 렌더링에 대한 연구를 다룬다. 3장에서는 제안한 알고리즘에 대한 전반적인 설명을 다룬다. 4장에서는 제안한 알고리즘의 성능 평가를 하였으며, 5장에서는 결론 및 향후 연구를 다룬다.

2. 관련 연구

분자 표면 렌더링을 생성하는 알고리즘들은[4, 5] 분자량이증가 할수록 분자를 구성하고 있는 폴리곤 수가 급격히 증가 하기 때문에 거대 분자의 실시간 렌더링은 어렵다. 분자 모델링 도구 중에서 널리 사용되는 VMD[6]는 그래픽 전용 워크스테이션에서 G PU가속을 이용한 렌더링 방식을 제안 하였다. 일반적으로 사용자가 디스플레이 되는 화면에 대한 상호작용을 느끼기 위해서는 6FPS이상이 되어야 하며, 약 15FPS 이상이 될 경우에 이를 실시간 렌더링이라고 할 수 있다[7]. 하지만 일반 사양의 그래픽 카드에서는 이러한 성능이 나오지 않는다.

거대 분자 모델의 렌더링시 폴리곤의 개체를 줄이기 위한 방법으로는 분자 모델의 메시를 단순화 하여 렌더링을 적용 하는 상세 단계 렌더링(LOD) 방법이 있다[8, 9]. 이러한 상세 렌더링 알고리즘들은 동적으로 분자 모델의 상세화를 지원한다. 하지만 시스템의 사양에 맞는 대화식 프레임 이상의 성능 보장을 하고 있지 못하며, 또한 단순화 과정 중에 분자 모델 구조의 기하학적인 정보를 유지가 어렵다.

본 논문에서는 이러한 문제를 해결하기 위해 거대 분자 모델의 실시간 적응형 상세 단계 렌더링 기법을 제안 하였다. 제안한 알고리즘은 분자 모델링의 기하학적인 특징을 유지하며 메시 단순화를 수행한다. 이러한 과정을 통하여 분자와 분자간에 결합이 일어나게 되는 위치인 Active Site와 같은 중요한 부분의 기하학적인 특징을 유지하게 해준다. 또한 단순화된 분자 모델들은 제안한 평가 함수를 통하여 시스템 환경에 따라 적합하게 선택이 됨으로써 대화식 프레임 이상의 속도를 보장 하게 된다.

3. Adaptive LOD Rendering of Molecular Models

3.1. Process Overview

우리는 거대 분자 모델에 대한 적응형 상세화 렌더링을 수행 하기 위해 다음의 그림1과 같이 2개의 과정을 지원한다. 첫 번째는 전처리 과정이다. 전처리 과정에서는 분자 구조의 3차원 정보를 담고 있는 분자 구조 파일(PDB, Protein Data Base)로부터 3차원 정보를 읽어 들인다. 이후에 이에 분자 표면 생성 알고리즘을 사용하여[4] 분자의 메시를 생성한다. 생성된 메시 데이터는 메시 단순화 알고리즘에 의해 적절하게 단순화가 이루어진다. 단순화 과정은 분자 모델의 시각적인 특징에 적합하게 100%, 80%, 60%, 40%, 20%로 구성을 하였다. 이제 단순화된 메시 정보들은 렌더링에 최적화된 파일구조(VLF, Classified LOD files)로 저장이 된다. 이에 대한 전체적인 과정은 그림 1.(a)에 나와있다.

두 번째 과정은 Adaptive LOD를 적용하여 실시간으로 렌더링 하는 과정이다. 적응형 LOD 렌더링이 필요한 이유는 시스템의 사양에 따라 대화식 프레임 이상의 성능을 낼 수 있는 분자 모델의 메시의 복잡도가 결정이 된다. 그러나 분자 모델이 단순화 될수록 분자 모델이 가지는 기하학적인 정보의 손실이 발생하게 된다. 따라서 시스템의 성능을 유지하며 분자 모델의 기하학적인 정보를 사용자에게 최대한 자세히 보여 줄 수 있는 적합한 상세 단계 렌더링을 수행해야 한다. 이러한 작업을 위하여 전처리 과정에서 생성된 VLF 파일을 읽어 들여 계층적인 LOD 모델을 구성하게 된다. 이후 렌더링 과정에서는 현재 시스템의 성능에서 대화식 이상의 프레임을 낼 수 있으며 또한 가장 자세한 정보를 보여 줄 수 있는 LOD 모델을 선택을 하게 된다. 선택된 LOD 모델은 실시간으로 렌더링을 수행하게 된다.

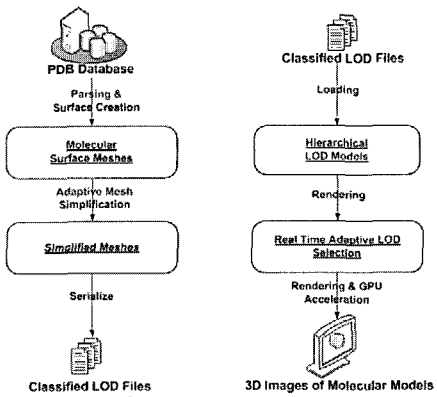

그림 1. System Overview
(a) Pre-Processing (b) Real-Time Rendering

3.2. Preprocessing

3.2.1. Mesh Simplification

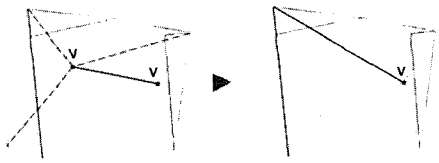

그림 2 Edge Construction 의 예

Mesh Simplification은 일반적으로 LOD에서 적용 되는 메시의 단순화 알고리즘을 개선하여 분자 모델에 Adaptive한 단순화를 한다. 본 연구에서는 메시 단순화 알고리즘 중에서 Edge Contraction방법을 사용한다. Edge Contraction은 메시 단순화를 수행할 때 삼각형을 이루는 점들 중에서 2개의 점을 하나의 점으로 줄이는 방법이다. 이러한 Edge Contraction 알고리즘은 다른 메시 단순화 알고리즘에 비해 빠른 처리 속도를 가지며 또한 높은 화질을 유지 하는 장점을 가진다. 다음의 그림2는 Edge Contraction을 적용하는 예이다. 정점 v1 과 v2를 있는 선분이 제거 되면서 삼각형이 제거 되며 이 삼각형의 주위에 있는 3개의 삼각형도 정보가 변하는 것을 알 수 있다. 본 논문에서는 Garland 와 Heckbert의 연구에 기반한 알고리즘을 사용한다[10].

알고리즘 동작 과정은 다음과 같이 구성된다. 생성된 메시 정보를 축소하기 위해서 먼저, 삼각형들을 이루는 선분을 구성하는 정점들의 리스트를 생성한다. 이후 이들 정점들 중 Edge Contraction을 위한 후보들을 선출하기 위해서 정점들과 이 정점들이 속한 평면들의 거리들의 합을 계산하여 평가를 하게 된다. 이를 위해서 각 정점 v = [x, y, z, 1]^T 와 이 정점을 포함하고 있는 각 평면 as n_i = [a, b, c, d]^t, $n_i^t x$ = 0 들의 거리의 합을 사용하는 2차 함수 Q를 수식(1)과 같이 정의한다.

$$Q(v) = \sum_{t=1}^{k} (n_i^t v)^2 \quad (1)$$

이식을 통하여 각 정점에 대한 평가 값을 이끌어 낼 수 있다. 즉, Edge Contraction을 수행할 정점들을 각각 v1, v2라고 할 때, 이 정점들은 하나의 정점 v` 로 합쳐지게 될 때 이차 함수의 값은 다음의 수식(2)와 같이 정의 된다. 따라서 생성된 정점들의 쌍의 리스트에 대해서 각 2차 함수의 합을 비교 하여 가장 낮은 값을 가지는 정점들의 쌍을 후보로 결정하게 된다.

$$(v_1, v_2) \rightarrow v_3, \quad (2)$$

$$Q(v_3) = Q(v_1) + Q(v_2)$$

이러한 이유는 새로 생성되는 정점 v`의 비용은 정점 v1과 v2 을 포함하고 있는 평면에 영향을 받는다. 새로 생성되는 정점 v` 에서 원래의 평면 까지의 거리가 기존의 정점들 에서의 거리 보다 크면 (v1, v2) → v`의 비용은 더 비싸지게 된다. 최종적인 비용 값인 Q(v`)는 정점들의 쌍의 순위를 정하는데 사용 된다[10].

3.2.2. Boundary Protection for Molecular Modeling

이 알고리즘은 일반적인 삼각형 메시에 적용이 이루어진다. 그러나 분자 모델은 isosurface로 구성되어 있다. 이러한 분자 모델의 특징상 메시 단순화 과정이 이루어 질 때, 외곽선들이 잦은 붕괴로 가까워 지게 되면 hole 및 t-junction 등의 현상이 발생하게 되어 사용자에게 올바른 정보를 제공하기 어렵게 된다. 이러한 이유는 특히 분자 모델링 실험 중 닥킹 시뮬레이션과 같은 에너지 안정화 작업에서는 Active Site와 같은 분자 모델의 기하학적인 정보에 후보 물질들이 바인딩이 이루어지기 때문에 단순화 과정에서 윤곽선의 모양을 최대한 유지하는 게 매우 중요하다. 특히 분자의 외곽을 구성하는 경계선들은 위에서 정의한 방법으로 Edge Contraction을 수행할 경우 합쳐지는 평면의 면적이 상대적으로 적기 때문에 가장 먼저 Edge Contraction이 일어나게 된다. 따라서 이러한 경계선들을 보호해 주어야 한다. 본 알고리즘에서는 이를 위해서 모서리들을 구성하는 정점을 따로 관리를 하며 가중치 계산에서 페널티를 적용한다.

적용하는 페널티는 다음과 같은 과정을 통하여 적용된다. 먼저 경계선의 선분(v1, v2)이 한 삼각형에 속해 있을 때, 이 선분을 구성하는 삼각형에 대한 법선 벡터 n 를 구한다. 이후 이 선분에 대한 벡터의 차(v2 - v1)와 법선 벡터 n 대한 외적에 대한 단위 벡터로 표현할 수 있다 (3). 이러한 값에 일정 가중치를 곱하여 삼각형들의 평가 비용을 계산할 때 적용을 한다. 그러므로 외곽선들의 평가 비용이 높아지게 되어 외곽선들이 쉽게 Edge Contraction이 일어 나지 않도록 해준다[8]. 다음의 그림 3은 이러한 Adaptive Mesh Simplification을 적용하여 HIV 바이러스를 렌더링한 모습이다.

$$n^* = \frac{n \times (v_2 - v_1)}{\|n \times (v_2 - v_1)\|} \quad (3)$$

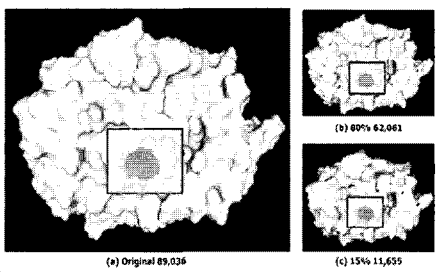

그림 3. Mesh simplification with maintaining geometrical features of critical parts of different LOD models: (a) The original model consists of 89,036 polygons, (b) The 80% simplified version has 62,061 polygons, (c) The 15% simplified version is composed of 11,655 polygons.

3.3 Real-Time Rendering

3.3.1 Cost Function

Cost Function은 렌더링 과정에서 2가지의 기능을 수행한다. 첫 번째는 현재 렌더링을 수행하는 분자 구조 모델에 대한 FPS의 속도를 일정값 이상으로 유지를 한다. 사용자가 3차원 렌더링에 대해 상호작용을 느낄수 있는 속도는 6FPS이며 사용자가 행동과 반응에 몰입할 수 있는 속도는 15FPS 이상이다[7]. 15FPS 이상 일때를 실시간 렌더링이라고 하며, Cost Function은 렌더링 속도가 15FPS 이상이 나오도록 분자 구조 모델을 단순화 하여 성능을 향상한다. 또한 렌더링 속도가 15FPS보다 훨씬 빠른 경우에는 분자 구조 모델을 상세화 하여 보여줌으로써 사용자에게 기하학적인 정보를 최대한 보여주게 된다.

두 번째는 렌더링 하게 되는 분자의 시각적인 특징을 유지해야 한다는 점이다. 분자 구조 모델의 각 폴리곤들은 이를 구성하는 원자에 대응이 된다. 이러한 원자는 종류에 따라서 색 정보를 갖게 되는데 메시의 단순화가 수행 되면 이러한 원자들의 색 정보들이 섞이거나 원자의 정보가 없어지는 현상이 발생하게 된다. 따라서 단순화 정도를 적절하게 수행 함으로써 원자의 정보가 최대한 유지 될 수 있도록 해야 한다. 본 논문 에서는 한 원자당 렌더링 되는 메시의 폴리곤의 수의 비율을 적절하게 유지되게 함으로써 시각적으로 원자의 정보를 최대한 유지하게 한다. 다음의 수식 (4)는 Cost Function에서 계산하는 수식이다.

즉 현재의 프레임 속도가 C_{fps} 가 목표 프레임 속도인 T_{fps} (T_{fps} = 15fps)보다 작게 되면 속도 향상을 위하여 분자 모델을 단순화 시키게 된다. 총 렌더링 하게 되는 분자 모델의 폴리곤의 수를 P_{cnt} 라고 하며 이를 단순화 할 때, 사용자 시점과 해당 모델의 거리인 Z_{eye} 와 전체 원자 개수당 A_{cnt} 분자 모델의 정점의 수의 비율을 조정하여 최적의 단순화 정도를 결정 하게 된다. 반대로 현재 프레임 속도가 목표 프레임 속도 보다 훨씬 크다면 보다 상세화를 시켜 렌더링을 수행 하게 된다. 이러한 시각화 정보를 최대한 유지할 수 있는 단순화 정도를 결정 함으로써 시각적으로 최대의 효과를 낼 수 있다.

$$P_{cnt} = \begin{cases} P_{cnt} - Z_{eye} \bullet \dfrac{P_{cnt}}{A_{cnt}}, \cdots\cdots\cdots if\ C_{fps} \le T_{fps} \\ P_{cnt} + Z_{eye} \bullet \dfrac{P_{cnt}}{A_{cnt}}, \cdots\cdots\cdots if\ C_{fps} > T_{fps} \end{cases} \quad (4)$$

4. 실험 및 결과

우리는 알고리즘의 성능 측정을 위하여 3.2Ghz의 CPU, 2GB 의 메모리, nVidia Quadro 3400 그래픽 카드를 장착한 시스템을 사용하였으며, VRMMS 시스템은 Windows XP환경에서 VC++과 OpenGL기반으로 작성되었다.
실험에 사용된 데이터는 PDB Bank[11]에서 다운로드 한 거대 단백질 분자를 사용하였다. 실험을 4개의 단백질로 한정한 이유는 이보다 작은 수의 분자량을 가진 단백질을 테스트 할 경우에는 기존의 방식으로도 높은 대화식 프레임 속도를 가질 수 있기 때문이다. 다음의 표 1은 실험에 사용된 4개의 분자 모델에 대한 원자량 및 폴리곤의 개수이다.

표 1. Selected molecular models.

PDB CODE	Name	Atom Num	Polygons
1AOD	Ballicus	14,165	533,774
1CX2	Musculus	22,228	801,601
1YCE	Na+ ATPase	27,147	970,187
1MT5	Hydrolyase	64,155	2,656,246

알고리즘의 성능 평가를 위한 실험은 3가지 방법으로 측정 되었다. 첫 번째 실험은 실제 분자 시뮬레이션 구조에서 사용되는 거대 바이러스 물질들의 렌더링 성능을 측정 하였다. 제안한 방법의 상세 렌더링에 따른 렌더링 성능을 측정하기 위해 다음과 같은 실험을 하였다. 실험 과정은 100초 동안 분자 모델에 대한 FPS를 측정 하였으며, 특히 VRMMS 시스템은 20초 단위로 LOD 모델을 점진적으로 변화 시키면서 측정을 수행하였다. 즉, 0-20초 에는 100% 모델, 20-40초에는 80% 모델, 40-60초에는 60% 모델, 60-80 초에는 80% 모델 그리고 80-100초에는 20% 모델을 사용하여 렌더링을 수행하고 FPS를 측정 하였다. 다음의 그림4는 측정한 실험 결과에 대한 그래프이다. 그림4 를 보면 각 분자 모델을 단순화 할 때마다 FPS 값이 점진적으로 개선 되어 향상됨을 알 수 있다.

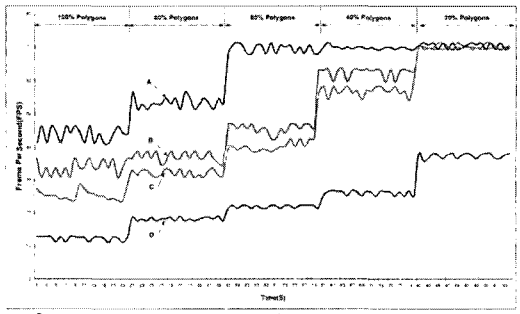

그림 4. Rendering speeds (in FPS values) of three dimensional interactions using molecular models, when the proposed algorithm is applied. A=Ballicus, B=Musclus, C=Na+ ATPase, D=Hydrolyase.

두 번째 실험은 기존의 일반적인 렌더링 방식을 지원 하는 VMD 시스템과 렌더링 성능을 비교 하였다.

이 경우 비교 측정은 시각적으로 단순화된 정보가 크게 차이 나지 않는 60% 단순화 과정 까지를 평균하여 측정을 하였다. 각 분자 모델에 대한 VRMMS 시스템과 VMD의 측정 결과는 다음의 표2와 같다. 표2 에서 보는 것과 같이 VMD 시스템에 비해 월등한 성능 향상 결과를 가져 오는 것을 알 수 있다. 이러한 이유에는 상세화 렌더링을 통한 성능 향상을 하였으며 GPU를 사용하여 렌더링 함으로써 CPU의 부하를 줄일 수 있기 때문에 가능하다.

표2. **Comparing the average FPS values of real-time rendering the three dimensional molecular models using VMD and VRMMS.**

PDB CODE	VMD	VRMMS
1AOD	12.8	55.64
1CX2	7.5	38.05
1YCE	6.4	31.94
1MT5	2.6	17.85

(FPS)

세 번째 실험 방식은 제안 하는 렌더링 알고리즘에서 해당 모델의 를 Z_{eye} 값(사용자와의 거리) 을 변경하여 LOD 모델을 변경 시 표현되는 FPS및 원자당 폴리곤의 개수의 변화를 측정하였다. 이러한 이유는 실제 시스템의 환경에 따라 대화식 프레임 이상의 성능을 낼 수 있는 분자 구조 모델의 메시의 단순화 정도와, 시각적으로 가장 많은 원자의 정보를 보여 줄 수 있는 단순화 정도의 교점을 찾기 위해서다. 측정 결과는 다음 그림6과 같다. 그림 6.(a)에서는 사용자의 시점에 따른 상세화를 증가할 때 FPS 의 그래프를 나타낸 그래프 이다. 즉 사용자 시점에서 가까워 지면서 상세화 될수록 FPS는 감소를 한다. 6.(b)는 사용자의 시점에 따른 상세화가 진행 될수록 원자당 폴리곤의 비율을 나타낸 그래프 이다. 여기서는 상세화가 진행할수록 원자당 폴리곤 비율은 증가한다. 이 2개의 그래프를 통하여 상세화에 따른 FPS의 성능과 시각적으로 좋은 화질을 보여줄 수 있는 경우는 60% 상세화 라는 것을 알 수 있다. 그림 7, 8, 9, 10은 분자 모델의 상세화를 적용한 모습이다.

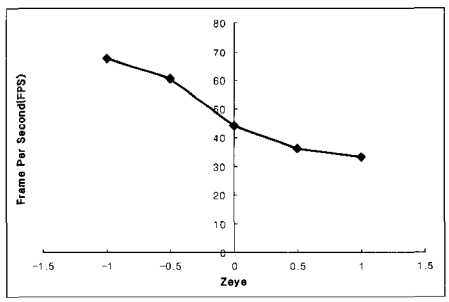

그림 6. (a) The FPS values of the Musculus model depending on the LOD levels.

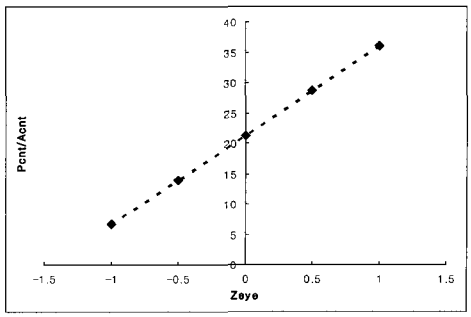

그림 6. (b) The Pcnt/Acnt ratios of the Musculus model depending on the LOD levels.

5. 결론 및 향후 연구

본 논문에서는 거대 분자 모델의 3차원 렌더링에 가지는 성능 문제, 즉 PC 급에서 사용자가 상호작용을 느끼지 못할 정도로 낮은 FPS를 가지는 문제를 적응형 상세 단계 렌더링 기법을 사용하여 해결 하였다. 전처리 과정에서는 분자 모델의 메시는 기하학적인 특징에 적합하게 단순화가 이루어지고 실시간 렌더링을 위한 파일로 관리된다. 렌더링 과정에서는 이 파일을 로딩 하여 렌더링을 수행 하게 되며 렌더링 시에는 시스템이 대화식 이상의 프레임을 이상을 보장하도록 렌더링 되며 또한 시각적으로 사용자에게 가장 많은 정보를 보여 줄 수 있도록 분자 구조의 메시가 결정을 한다. 이후 결정된 모델은 최적화 되어 렌더링을 수행 하게 된다. 제안한 알고리즘의 성능 측정 결과 기존의 일반적인 렌더링을 사용하는 거에 비해 큰 성능향상 결과를 보여 주었다. 특히 PC급에서도 대화식 이상의 프레임을 보장 하였으며 상세 단계 렌더링시 시각적으로 원본의 데이터 구조와 최대한 비슷한 단순화된 모델을 렌더링을 하였다. 이러한 본 논문에서 제안한 렌더링 알고리즘을 사용한다면 대규모 병렬 처리 시스템을 통하여 수행하는 거대 바이러스 물질의 닥킹 시뮬레이션 등에 적용이 가능하다. 특히 3차원 가상현실 시스템 기반에서 거대 바이러스 물질의 신약 설계 작업등의 중간 과정의 관측 및 제어을 통한 작업 능률의 향상에 사용 될 수 있다.

향후 연구에서는 먼저 Mesh Simplification 과정 중에서 Active Site 등의 분자 구조에 의미가 있는 부분은 단순화가 아닌 보다 상세 하게 표현할 수 있는 Context Plus 알고리즘을 적용 하여 개선할 예정이다. 또한 실제 신약 설계 시뮬레이션 과정에 도입을 하여 오랜 시간이 걸리는 신약 설계 과정을 보다 효율적이고 체계적으로 관리 할 수 있도록 하겠다.

참고문헌

[1] F.M Richards, "Areas, volumes, packing and protein structures," in Annual Review of Biophysics and Bioengineering, 6, pp. 151-176, 1977.

[2] Jee-In Kim, Sungjun Park, Jun Lee, Youngjin Choi and Seunho Jung, "Development of a Gesture-Based Molecular Visualization Tool Based on Virtual Reality for Molecular Docking", Bulletin of the Korean Chemical Society, Vol.25, No.10, pp 1571-1574, 2004.

[3] Sungjun Park, Jun Lee and Jee-In Kim, "A Molecular Modeling System Based on Dynamic Gestures", LNCS 3480, pp.886–895, 2005.

[4] Amitabh Varshney, Frederick P. Brooks and Jr. William V. Wright, "Linearly Scable Computation of Smooth Molecular Surfaces", IEEE Computer Graphics and Applications, Vol. 14, No.5, pp 19 – 25 Sept 1994.

[5] C.H. Lee and A. Varshney, "Representing Thermal Vibrations and Uncertainty in Molecular Surfaces", SPIE Conference on Visualization and Data Analysis 2002, San Jose, CA, Jan, 2002.

[6] VMD, http://www.ks.uiuc.edu/Research/vmd/

[7] Tomas Akenine-Moller and Eric Haines, *REAL-TIME RENDERING, The 2nd Edition*, AK Perters.

[8] Stefan Birmanns, Maik Bolyes, Hwrwig Zilken and Willy Wriggers, "Adaptive Visuo-Haptic Rendering for Hybrid Modeling of Macromolecular Assemblies", In: Proceedings Mechatronics and Robotics, Vol.4, pp. 1351-1356, 2004.

[9] Chandrajit Bajaj, Peter Djeu, Vinay Siddavanahalli and Anthony Thane, "Interactive visual exploration of large flexible multi-component molecular complex", Proc. of the Annual IEEE Visualization Conference, pp. 243-250, 2004.

[10] Michel Garland and Paul S.Heckbert, "Surface Simplification using quadric error metrices", In Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pp 209-216, ACM Press/Addision-Wesly Publishing Co. 1997.

[11] PDB Bank, http://www.rcsb.org/

[12] 이준, 박성준, 김지인, "GPU 기반의 거대 분자의 효율적인 실시간 렌더링 기법", 한국 컴퓨터 그래픽스학회 추계학술대회, pp 25~28, 2005

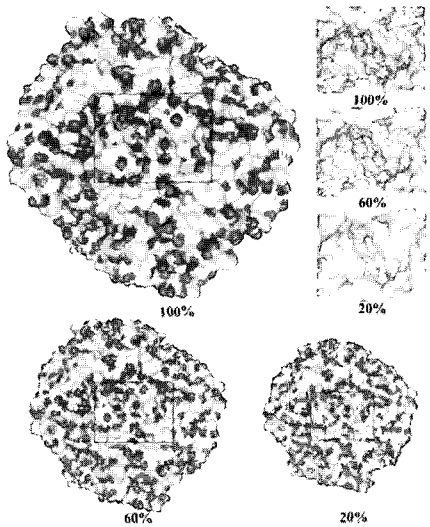

그림 7. Ballicus : The 100% version with 533,774 polygons, the 60% simplified version with 314,696 polygons, the 20% version with 100,104 polygons.

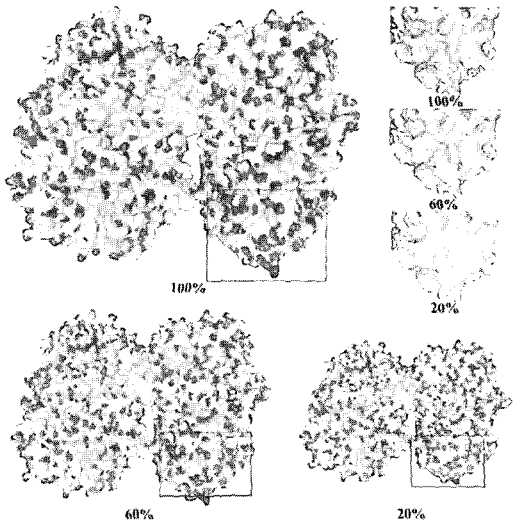

그림 9. Na+ATPase : The 100% version with 970,187 polygons, the 60% simplified version with 571,680 polygons, the 20% version with 181,978 polygons.

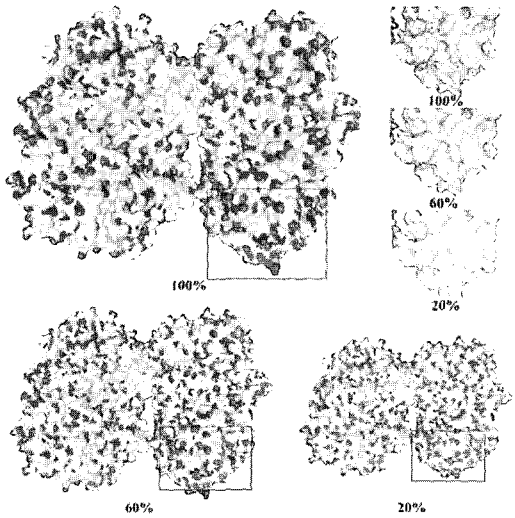

그림 8. Musculus : The 100% version with 801,601 polygons, the 60% simplified version with 473,407 polygons, the 20% version with 149,711 polygons.

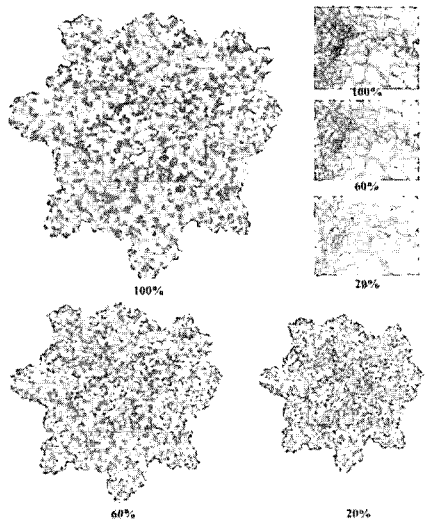

그림 10. Hydrolyase : The 100% version with 2,656,246 polygons, the 60% simplified version with 1,565,592 polygons, the 20% version with 496,340 polygons.