KIPS Tr. Comp. and Comm. Sys.
Vol.3, No.1 pp.1~6 pISSN: 2287-5891

http://dx.doi.org/10.3745/KTCCS.2014.3.1.1

Approximating the Convex Hull for a Set of Spheres

Byungjoo Kim† · Ku-Jin Kim†† · Young J. Kim†††

ABSTRACT

Most of the previous algorithms focus on computing the convex hull for a set of points. In this paper, we present a method for approximating the convex hull for a set of spheres with various radii in discrete space. Computing the convex hull for a set of spheres is a base technology for many applications that study structural properties of molecules. We present a voxel map data structures, where the molecule is represented as a set of spheres, and corresponding algorithms. Based on CUDA programming for using the parallel architecture of GPU, our algorithm takes less than 40ms for computing the convex hull of 6,400 spheres in average.

Keywords : Convex Hull, Spheres, Protein Molecule, Molecular Interface, GPU

구 집합에 대한 컨벡스헐 근사

김 병 주† · 김 구 진†† · 김 영 준†††

요 약

현재까지 컨벡스헐 (convex hull) 의 계산 알고리즘들은 주로 점 집합 (point set) 에 대해 연구가 수행되어 왔다. 본 논문에서는 이산 공간에서 다양한 반경을 갖는 구 집합에 대한 컨벡스헐을 근사하는 방법을 제시한다. 구 집합에 대한 컨벡스헐 계산은, 특히 단백질 분자의 구조적인 특성을 연구하는 여러 응용분야에서 계산 효율성을 증대시키기 위한 기반 기술이라 할 수 있다. 분자에 대응하는 구의 집합에 대해 복셀 맵 (voxel map) 자료구조를 적용하고 이를 이용하여 컨벡스헐을 계산하는 알고리즘을 제시한다. 제안된 방법은 GPU를 활용한 병렬처리를 수행하여 평균적으로 6,400개 이하의 구가 포함된 집합에 대해 40ms 이내에 컨벡스헐을 계산하는 성능을 보인다.

키워드 : 컨벡스헐, 구, 단백질 분자, 분자 인터페이스, GPU

1. 서 론

리간드 (ligand) 가 분자와 결합할 수 있는 활성 부위 (active site) 를 발견하고, 외부로부터 활성 부위까지 도달할 수 있는 채널 (channel) 및 터널 (tunnel) 을 탐색하기 위해 단백질 분자의 내부 영역과 외부 영역이 구분되어야 하며, 이를 위해 컨벡스헐 계산 알고리즘이 사용되어 왔다 [1-11].

단백질 분자의 채널을 계산하는 기존 연구들은 분자의 컨벡스헐을 계산하기 위해 주로 Quickhull 알고리즘 [12] 을 사용하였으며, 이 알고리즘은 점 집합 (point set) 에 대해서 매우 빠른 속도로 컨벡스헐을 계산한다. Quickhull 알고리즘은 점 집합에 대해 극단점들 (extreme point) 을 찾고, 극단점들에 의해 정의된 영역 내부에 속하는 점들을 제거하는 과정을 재귀적으로 수행하면서 컨벡스헐을 계산한다. 최근에는 Quickhull 알고리즘을 바탕으로 GPU 병렬 처리 가속을 CUDA 라이브러리로 구현한 Cudahull 알고리즘 [13] 이 발표되었다. Cudahull 알고리즘은 Quickhull 알고리즘에서 재귀적으로 반복 수행되는 루틴을 병렬적으로 처리하여 수십배의 성능 향상을 보인다.

기하학적인 측면에서 단백질 분자는 구의 집합 (sphere set) 으로 표현될 수 있으며, 분자의 컨벡스헐 계산은 구의 집합에 대한 컨벡스헐 계산에 대응된다. 현재까지 분자에 대한 컨벡스헐 계산 알고리즘들은 주로 단백질 분자의 중심점 집합에 대해 적용되었고, 서로 다른 반경을 가진 구로 이루어진 집합에 대해서는 구의 표면에서 점들을 추출하여 근사적으로 계산할 수밖에 없었다.

※ 본 논문은 저자들이 2012년 12월 VRCAI 학회에서 포스터로 발표한 논문[16]을 개선하여 확장한 것임. 본 논문은 2012학년도 경북대학교 학술연구비에 의하여 연구되었으며, 저자들 중 김구진 교수는 2013년도 정부(교육부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 연구를 수행하였음(NRF-2013R1A1A2A10004391). 저자들 중 김영준 교수는 한국연구재단 중견연구자지원사업(No. 2012R1A2A2A01046246, No. 2012R1A2A2A06047007)의 지원을 받았음.
† 정 회 원 : 경북대학교 컴퓨터학부 박사후연구원
†† 정 회 원 : 경북대학교 컴퓨터학부 정교수
††† 비 회 원 : 이화여자대학교 컴퓨터공학과 교수
논문접수 : 2013년 7월 16일
심사완료 : 2013년 10월 17일
* Corresponding Author : Ku-Jin Kim(kujinkim@gmail.com)

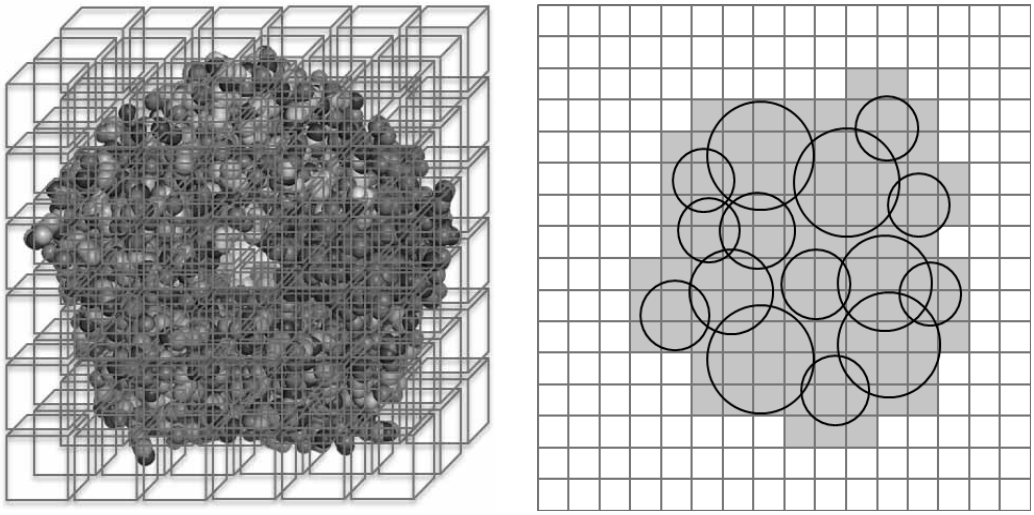


Fig. 1. Voxel map based molecule representation

Boissonnat 등 [14] 은 구의 집합에 대한 컨벡스헐 알고리즘을 제안했으나, 이들이 제안한 내용은 3차원 이상의 공간에서 구의 집합에 대해 이론적으로 컨벡스헐을 계산하는데 초점을 둔 연구이기 때문에, 실제 응용분야에서 빠른 시간내에 컨벡스헐을 계산하기에는 어려움이 있다.

본 논문에서는 이산 공간에서 구의 집합에 대한 컨벡스헐을 매우 빠른 속도로 계산하여 구의 집합에 대해 내부와 외부 영역을 구분하는 알고리즘을 제시한다. 기존의 Cudahull 과 같은 알고리즘을 분자의 컨벡스헐 계산에 적용할 경우에는 각 원자 표면에서 표본 점들을 추출하여 사용하여야 하며, 결과로 구해지는 컨벡스헐은 다각형 메쉬 (polygonal mesh) 형태이므로 본 논문에서 고려하는 이산 공간에서 컨벡스헐을 계산하기에는 비효율적이다. 본 논문에서 제안하는 알고리즘은 구의 집합을 포함하는 경계 상자 (bounding box) 를 계산하고, 경계 상자에 대한 복셀 맵 (voxel map) 을 구성한다. 경계 상자의 최외곽에 있는 복셀들로부터 단백질 분자까지의 최소 거리 계산을 통해 컨벡스헐을 계산하고, GPU를 기반으로 병렬 처리를 수행함으로써 효율적으로 컨벡스헐을 추출한다.

본 논문의 구성은 다음과 같다. 2절에서는 복셀 맵을 구성하는 알고리즘을 제시하고, 3절에서는 복셀 맵을 이용하여 컨벡스헐을 계산하는 알고리즘을 제시한다. 4절에서는 실험 결과를 제시하고, 5절에서 결론을 내린다.

2. 복셀 맵 구조

기하학적인 측면에서 단백질 분자를 표현할 경우, 분자를 구성하는 각 원자를 반데르바스 반경 (van der Waals) 을 갖는 구로 표현하는 방법이 범용적으로 사용된다. 분자가 n 개의 원자로 구성되고, 원자들이 각각 중심점 $\mathbf{c}_i$ ($0 \le i < n$)와 반데르발스 반경 (van der Waals) r_i 를 가질 때, 각 원자는 다음과 같이 한 개의 구로 둘러싸인 영역 $B(\mathbf{c}_i, r_i)$로 표현할 수 있다.

$$B(\mathbf{c}_i, r_i) = \{\mathbf{p} \in \mathrm{R}^3 \mid \|\mathbf{p}-\mathbf{c}_i\| \le r_i\}$$

또한, 이러한 원자들로 구성된 한 개의 분자는 다음과 같은 집합 M으로 정의된다.

$$M = \{B(\mathbf{c}_i, r_i) \mid 0 \le i < n\}.$$

분자를 표현하는 집합 M을 포함하는 3차원 공간이 경계 상자의 형태로 주어졌다고 가정한다. 본 논문에서는 단백질 분자를 다양한 반경을 가진 구의 집합 M으로 표현하고, M의 컨벡스헐을 근사화한다. 이산 공간에서 컨벡스헐을 근사화하기 위해 M에 포함된 모든 구를 포함하는 최소영역인 경계 박스를 정의하고, 경계 박스를 복셀(voxel)들의 집합으로 분할한다.

Kim 등 [15] 은 단백질 분자를 복셀 맵 구조 및 스피어 트리 구조로 표현하여, 각 구조에 대해 proximity query를 수행하기 위한 알고리즘을 제시하고 계산 시간을 비교하였다.

복셀 맵 구조는 M을 포함한 공간에 대한 경계 상자 B가 x, y, z축의 방향을 따라 일정한 크기로 분할되어 복셀의 집합을 구성하고, 각 복셀마다 복셀과 교차하는 원자의 정보를 저장한 것이다. 한 개의 복셀 $V^{\alpha\beta\gamma}$ 이 포함한 구의 정보는 $B(\mathbf{c}_i, r_i) \cap V^{\alpha\beta\gamma} \neq \emptyset$ 을 만족하는 구의 집합으로 표시된다. (Fig. 1)에서는 단백질 분자에 대해 복셀 맵을 구성한 예를 보인다.

본 논문에서 제안하는 알고리즘은 구의 집합에 대한 컨벡

스헐 계산을 위하여 Kim 등 [15] 이 제시한 방법을 적용하여 복셀 맵 구조 상에서 한 점으로부터 구의 집합에 대해 최소 거리를 계산한다.

3. 외부 복셀 제거를 이용한 컨벡스헐 계산

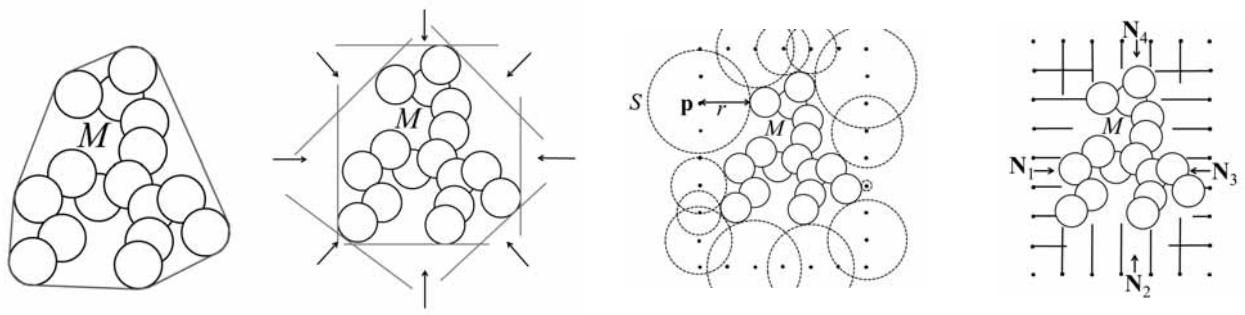
본 논문에서 제시하는 컨벡스헐 알고리즘은 복셀 맵으로 표현된 이산 공간에서 효율적으로 구의 집합에 대한 컨벡스헐의 내부 및 외부 복셀을 판별하는 것을 목표로 한다.

(Fig. 2A)에서는 구의 집합으로 표현된 단백질 분자에 대해 실 컨벡스헐을 구한 예를 2차원으로 간략화하여 제시한다. 컨벡스헐은 무한대의 반경을 갖는 구 (즉, 지역적인 표면이 평면과 같아 보이는 구) 를 M의 외부에서 M과 접하도록 굴린다고 가정할 때, 구의 envelope 곡면을 추출한 것이라고 가정할 수 있다 (Fig. 2B). 만약 외부에서 굴리는 구의 반경이 무한대보다 작아지면, 이러한 구의 envelope 곡면은 형태가 컨벡스헐과 유사하지만 concave한 부분이 있는 곡면이 구해질 것이다. 최외곽 복셀들 (즉, 복셀 맵의 경계 상자 표면에 속한 복셀들) 의 중심점의 집합을 P라 할 때, P에 속한 각 점 **p**에서 M에 대한 최소 거리를 계산하여 M의 컨벡스헐을 계산하는 예를 (Fig. 2C)에서 제시한다. 점 **p** $\in P$ 로부터 M에 대한 최소 거리가 r이라 할 때, 중심점이 **p**이고 반경이 r인 구 S를 계산하여 S에 포함되는 복셀들을 제거하면, 남은 복셀들은 M에 대한 컨벡스헐 내부 영역을 구성하게 된다. P에 속한 점들을 중심점으로 하는 모든 구에 대해 이 과정을 적용하여 구에 포함된 복셀들을 제거하는 작업을 수행할 경우, 여러 개의 구에 중복 포함된 복셀에 대해서는 같은 작업이 반복적으로 수행된다. 이러한 비효율적인 면을 개선하기 위해 다음의 방법을 적용할 수 있다. (Fig. 2C)에서는 몇 개의 표본 점들로부터 M에 대한 최소 거리가 반경이 되는 구들을 예로 보인다.

점 **p**가 속한 경계 상자의 면이 법선 벡터 (normal vector) **N**을 가질 경우, 경계 상자의 내부를 향하는 방향으로 **N**을 따라 가며 만나는 모든 복셀들을 컨벡스헐의 외부 복셀로 판단하는 방법이다. 점 **p**와 M 간의 거리가 r이라고 가정하자. 점 **p**가 속한 면에 따라 **N** 방향으로 거리 r만큼 진행하며 만나는 복셀들을 차례대로 외부 복셀로 판별한다. **N** 방향의 진행 거리가 r이 되면 이 작업을 중단한다. (Fig. 2D)에서는 각 면 별로 법선 벡터 방향을 $\mathbf{N}_i$ $(1 \le i \le 4)$로 표시하고, 해당 $\mathbf{N}_i$ 방향으로 진행하며 만나는 외부 복셀들을 선분으로 표시하였다.

컨벡스헐을 계산하는 알고리즘은 최외곽 복셀의 각 중심점 **p** 마다 thread를 할당하고, CUDA 언어를 이용하여 병렬로 처리된다. 구의 집합 M에 대해 구성한 복셀 맵 구조가 V라 할 때, 한 점 **p**로부터 구의 집합 M에 대한 최소 거리를 MinDist(M, V, **p**) 라 할 때, 알고리즘은 다음과 같다.

```
Function ConvexHull(M, V, P)
/* M: a set of spheres */
/* V: a voxel map structure for M */
/* P: a set of points embedded on the bounding box of M */
Begin
  For each p ∈ P do in parallel begin/* GPU code */
      minDist := MinDist(M, V, p);
      VL := the list of voxels in V within the
          distance of minDist from p along Ni;
      For each v∈VL do
      Determine voxel v as an exterior voxel;
  End
  return modified V;
End
```



(A) (B) (C) (D)

Fig. 2. Convex hull construction for a set of spheres

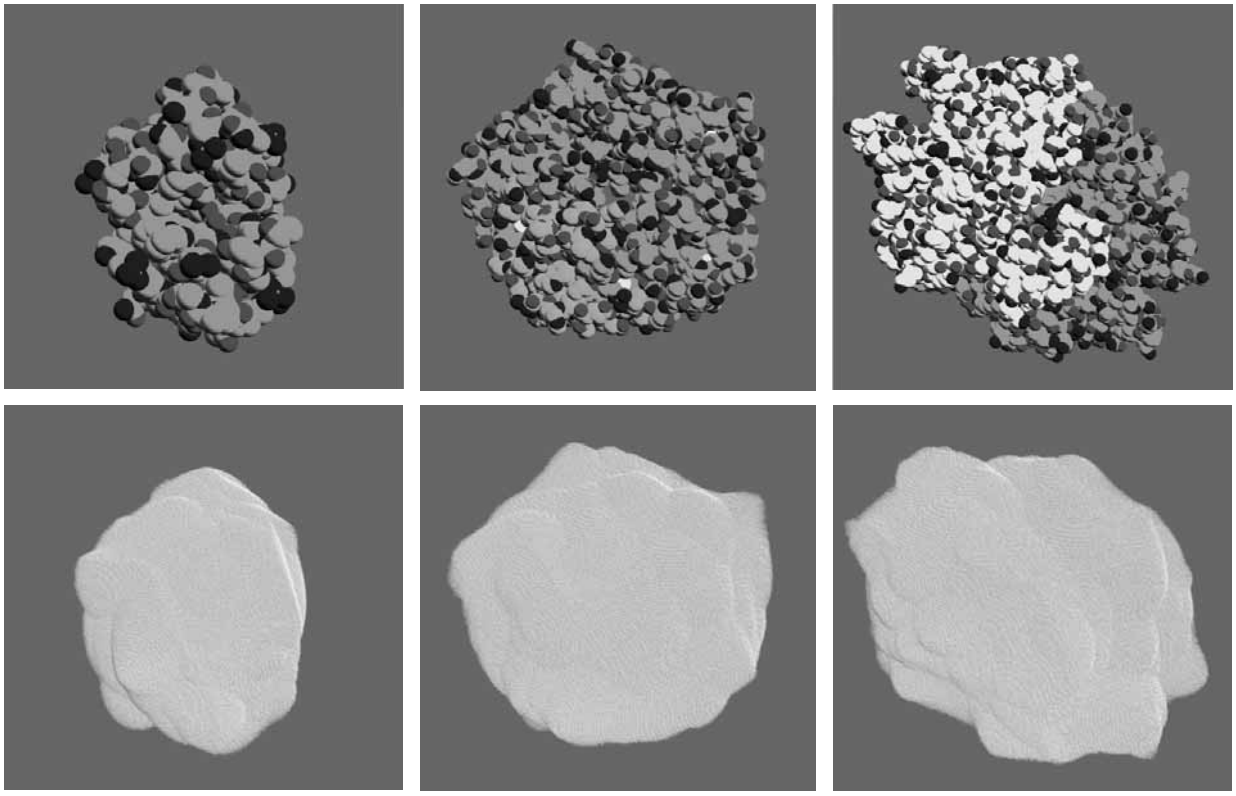


Fig. 3. Convex hull approximation for protein molecules (from left to right pdb id : 2LYZ, 1B44, 1DDZ)

Table 1. Comparison for the computation time of CPU and GPU based algorithms

Protein molecule		GPU based computation time (ms)			CPU based computation time (ms)			*Speedup
PDB id.	Number of atoms	Voxel map construction	Convex hull computation	Total	Voxel map construction	Convex hull computation	Total	
2LYZ	1,001	0.05	15.62	15.67	0.66	1286.48	1287.14	82.14
1CA2	2,039	0.05	24.13	24.18	1.35	1381.29	1382.64	57.18
1EA1	3,509	0.06	32.88	32.94	2.29	1558.50	1560.79	47.38
1B44	4,190	0.07	38.48	38.55	2.63	1630.31	1632.94	42.36
1DDZ	7,485	0.10	57.90	58.00	4.75	2248.03	2252.78	38.84
1Q0D	11,183	0.15	58.40	58.55	7.66	2194.58	2202.24	37.61
average	6,354	0.08	37.90	37.98	3.22	1716.59	1719.76	45.28

*Speedup = (CPU based total computation time) / (GPU based total computation time)

4. 실험 결과

실험은 Intel core(TM) i5 CPU (2.8GHz)와 4.0GB의 DRAM 및 nVidia GTX590 GPU가 장착된 컴퓨터에서 실행하였다. Window 환경에서 Visual C++, CUDA를 이용하여 실험 결과를 얻었고, OpenGL을 이용하여 시각화하였다. 표 1에서는 실험에서 입력으로 사용한 단백질 분자의 pdb id (http://www.pdb.org) 와 분자를 복셀 맵으로 구성하는 데 걸린 수행 시간 및 컨벡스헐을 계산하는 데 걸린 시간을 제시한다. 또한, 본 논문에서 제안한 알고리즘을 단일 CPU

(single-core CPU) 를 이용하여 계산한 결과 및 GPU 기반의 CUDA 프로그램을 이용하여 계산한 결과를 제시하였다. 자료구조를 구성하는 데 걸린 수행 시간과 컨벡스헐을 계산하는 데 걸린 시간은 100번 반복 수행한 평균값이다.

사용한 pdb 파일은 1,001개의 원자를 포함하는 2LYZ부터 11,183개의 원자로 구성된 1Q0D까지 총 6개의 단백질 분자를 대상으로 하였으며, 평균적으로 6,354개의 원자에 대해 37.94ms의 수행시간이 소요되었다. GPU 기반의 프로그램은 CPU로 구현한 프로그램에 비해 평균적으로 약 45.28배 계산 속도가 향상되었다.

(Fig. 3)은 제안된 알고리즘을 사용하여 단백질 분자 2LYZ, 1B44, 1DDZ 에 대한 실험 결과를 보여준다. 위쪽 열(row)은 단백질 분자를 구 집합으로 표현한 예이고, 아래쪽 열은 컨벡스헐의 경계 부분에 해당하는 복셀에 대해서만 시각화한 예이다.

5. 결 론

본 논문에서는 단백질 분자에 대해 복셀 맵을 이용하여 컨벡스헐을 계산하는 알고리즘을 제안하였다. 분자를 구성하는 원자들이 각각 구로 표현될 때, 각 복셀과 교차하는 원자들의 집합을 구하여 복셀 맵을 구성한다. 경계 복셀에서 복셀 맵과의 최소 거리 계산을 수행하여 컨벡스헐을 계산한다. 제안된 방법은 GPU를 활용한 병렬처리를 수행하여 6,400개 이하의 구의 집합에 대해 평균적으로 40ms 이내에 컨벡스헐을 계산하는 성능을 보여준다. 향후, 제시된 알고리즘들을 좀 더 병렬화하여 성능을 개선할 계획이다.

참 고 문 헌

[1] T.J.A. Ewing, S. Makino, A.G. Skillman, and I.D. Kuntz, “DOCK4.0: Search Strategies for Automated Molecular Docking of Flexible Molecule Databases,” Journal of Computer-Aided Molecular Design, Vol.15, No.5, pp.411-428, 2001.

[2] S.K. Lai-Yuen and Y.S. Lee, “Interactive Computer-Aided Design for Molecular Docking and Assembly,” Computer-Aided Design and Applications, Vol.3 No.6, pp.701-709, 2006.

[3] D. Levine, M. Facello, P. Hallstrom, G. Reeder, B. Walenz, and F. Stevens, “Stalk: an Interactive System for Virtual Molecular Docking,” IEEE Computational Science and Engineering, Vol.4, No.2, pp.55-65, 1997.

[4] H. Nagata, H. Mizushima, and H. Tanaka, “Concept and Prototype of Protein-Ligand Docking Simulator with Force Feedback Technology,” Bioinformatics, Vol.18, No.1, pp.140-146, 2002.

[5] R.D. Taylor, P.J. Jewsbury, and J.W. Essex, “A Review of Protein-Small Molecule Docking Methods,” Journal of Computer-Aided Molecular Design, 16(3), 151-166(2002).

[6] O. Trott and A.J. Olson, “AutoDock Vina: Improving the Speed and Accuracy of Docking with a New Scoring Function, Efficient Optimization, and Multithreading,” Journal of Computational Chemistry, Vol.31, No.2, pp.455-461, 2010.

[7] C.M. Venkatachalam, X. Jiang, T. Oldfield, and M. Waldman, “LigandFit : a Novel Method for the Shape-Directed Rapid Docking of Ligands to Protein Active Sites,” Journal of Molecular Graphics and Modelling, Vol.21, No.4, pp.289-307, 2003.

[8] Y. Zhao and M.F. Sanner, “Protein-Ligand Docking with Multiple Flexible Side Chains,” Journal of Computer Aided Molecular Design, Vol.22 No.9, pp.673-679, 2008.

[9] M. Petrek, M. Otyepka, P. Banas, P. Kosinova, J. Koca, and J. Damborsky, “CAVER: a new tool to explore routes from protein clefts, pockets and cavities,” BMC Bioinformatics, Vol.7, pp.316-324, 2006.

[10] M.Petrek, P. Kosinova, J. Koca, and M. Otyepka, “MOLE: a Voronoi diagram based explorer of molecular channels, pores, and tunnels,” Structure 2007, Vol.15, No.11, pp.1357-1363, 2007.

[11] R. G. Coleman, and K. A. Sharp, “Finding and Characterizing Tunnels in Macromolecules with Application to Ion Channels and Pores,” Biophysical Journal, Vol.96, No.2, pp.632-645, 2009.

[12] C. B.Barber, D. P. Dobkin, and H. Huhdanpaa, “The quickhull algorithm for convex hulls,” ACM Transactions on Mathematical Software, Vol.22, No.4, pp.469-483, 1996.

[13] A. Stein, E. Geva, and J. El-Sana, “CudaHull: Fast parallel 3D convex hull on the GPU,” Computers & Graphics, Vol.36, No.4, pp.265-271, 2012.

[14] J.-D. Boissonnat, A. Cerezo, O. Devillers, J. Duquesne, M. Yvinec, “An algorithm for constructing the convex hull of a set of spheres in dimension d. Computational Geometry, Vol.6, No.2, pp.123-130, 1996.

[15] B. Kim, J. E. Lee, Y. J. Kim, K.-J. Kim, “Comparison of voxel map and sphere tree structures for proximity computation of protein molecules,” Journal of Korea Multimedia Society, Vol.15, No.6, pp.794-804, 2012.

[16] B. Kim, and K.-J. Kim, “Computing the convex hull for a set of spheres on a GPU,” Procd. of VRCAI 2012 (poster abstract), Dec.2-4, Singapore, pp.345, 2012.

김 병 주

e-mail : kbj113@hotmail.com

2002년 경북대학교 전자전기공학부(학사)

2004년 경북대학교 전자공학과(공학석사)

2006년 경북대학교 전자공학과(공학박사 수료)

2006년~2010년 (주)휴원 과장

2010년~2011년 (주)LG전자MC연구소 선임연구원(과장)

2013년 경북대학교 전자공학과(공학박사)

현 재 경북대학교 컴퓨터학부 박사후연구원

관심분야 : 컴퓨터그래픽스, 컴퓨터비전, 기하 모델링, 병렬처리 등

김 구 진

e-mail : kujinkim@gmail.com

1990년 이화여자대학교 전자계산학과(학사)

1992년 한국과학기술원 전자계산학과(석사)

1998년 포항공과대학교 컴퓨터공학과(박사)

1998년~2000년 미국 Purdue University (PostDoc.)

2000년~2002년 아주대학교 정보통신전문대학원 BK21조교수

2002년~2003년 Dept. of Mathematics and Computer Science, University of Missouri-St. Louis, Visiting Assistant Professor

현 재 경북대학교 컴퓨터학부 정교수

관심분야 : 계산생물학, 컴퓨터 그래픽스, 곡면 및 기하모델링, 병렬처리 등

김 영 준

e-mail : kimy@ewha.ac.kr

1993년 서울대학교 계산통계학과(학사)

1996년 서울대학교 계산통계학과(석사)

2000년 미국 Purdue University(박사)

현 재 이화여자대학교 컴퓨터공학과 교수

관심분야 : 실시간 컴퓨터 그래픽스, 컴퓨터 게임, 로보틱스, 햅틱스, 기하모델링