http://dx.doi.org/10.3745/KIPSTA.2012.19A.2.073

복셀 맵을 이용한 단백질 표면 원자의 발견 알고리즘

김 병 주[†] · 김 구 진[††] · 성 준 경[†††]

요 약

본 논문에서는 단백질 분자로부터 표면 원자를 효율적으로 발견하는 알고리즘을 제안한다. 표면 원자란, 주어진 probe solvent P가 단백질 분자와 충돌하지 않고 접한다고 가정할 때, P와 접할 수 있는 원자의 집합을 의미한다. 단백질 분자를 구성하는 원자들은 반데르바스 반경을 갖는 구의 집합으로 표현되며, probe solvent 역시 구로 대응된다. P의 반경에 대해 분자의 오프셋 곡면을 구하여 표면 원자를 발견하는 알고리즘을 제안한다. 제안된 알고리즘은 각 구의 오프셋 곡면에 대해 복셀 맵(voxel map)을 구성하여 효율적으로 분자의 오프셋 곡면을 구하며, GPU (graphic processor unit)를 활용한 병렬처리를 수행하여 최대 6,412개의 원자를 갖는 분자에 대해 42.87 millisecond 내에 표면 원자를 발견한다.

키워드 : 단백질 분자, 표면 원자, 분자 도킹, 복셀 맵, 오프셋 곡면, GPU

An Algorithm for Finding Surface Atoms of a Protein Molecule Based on Voxel Map Representation

Byungjoo Kim[†] · Ku-Jin Kim[††] · Joon-Kyung Seong[†††]

ABSTRACT

In this paper, we propose an efficient method to extract surface atoms from a protein molecule. Surface atoms are defined as a set of atoms who can contact given probe solvent P, where P does not collide with the molecule. The atoms contained in the molecule are represented as a set of spheres with van der Waals radii. The probe solvent also is represented as a sphere. We propose a method to extract the surface atoms by computing the offset surface of the molecule with respect to the radius of P. For efficient computation of the offset surface of a molecule, a voxel map is constructed for the offset surfaces of the spheres. Based on GPU (graphic processor unit) acceleration, a data parallel algorithm is used to extract the surface atoms in 42.87 milliseconds for the molecule containing up to 6,412 atoms.

Keywords : Protein Molecule, Surface Atom, Molecule Docking, Voxel Map, Offset Surface, GPU

1. 서 론

단백질 분자 간의 도킹 (docking), 바인딩 (binding) 등 다양한 상호작용은 대부분 단백질 분자의 표면에서 발생한다. 또한 단백질 분자에 대한 분자 곡면 (molecule surface)의 계산 및 가시화, 그리고 리간드 (ligand)가 분자와 결합할 수 있는 활성 부위 (active site)를 발견하기 위해서도 단백질 분자의 내부 영역과 표면 영역을 구분하는 작업이 필요하다[1-8]. 단백질 표면 원자 (surface atom)를 효율적으로 추출하는 기법은 위와 같은 응용 분야에서 계산 효율성을 증대시키기 위한 기반 기술이라 할 수 있다. 단백질의 표면 원자가 빠른 시간 안에 계산될 수 있다면, 분자 간의 도킹이나 바인딩, 분자곡면 계산 등에 있어 처리 속도가 현격하게 개선될 수 있다. 표면 원자의 효용성에도 불구하고 현재까지 표면 원자에 대한 정의는 일관성 있게 내려지지 않았고, 응용 분야에 따라 서로 다른 표면 원자의 정의가 사용되고 있다.

Deanda and Pearman [9]은 SAS (solvent-accessible surface)를 이용하여 표면 원자를 추출하는 알고리즘을 제안하였다. 이들은 표면 원자 추출을 위해 SAS와 다른 다양한 방법을 비교하여 SAS 방법의 우수성을 보였다. SAS 방법에서는 단백질 분자를 구성하는 원자들 중에서 probe solvent 와 접할 수 있는 원자들을 표면 원자로 정의하였다.

※ 이 논문은 저자들이 2011년도 SIGGRAPH ASIA 학회에 포스터로 발표한 "Finding Surface Atoms of a Protein Molecule on a Graphics Processing Unit(GPU)" 논문을 확장한 것이며 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No.2011-0004094).
† 정 회 원 : 경북대학교 전자공학과 박사과정
†† 정 회 원 : 경북대학교 컴퓨터학부 부교수(교신저자)
††† 정 회 원 : 숭실대학교 컴퓨터학부 조교수
논문접수 : 2011년 8월 22일
수 정 일 : 1차 2011년 9월 26일
심사완료 : 2011년 9월 27일

원자가 probe solvent 와 교차하는 면적에 따라 실 표면 원자 (true surface atom)와 유효 표면 원자 (effective surface atom)로 구분하였다. 원자와 가장 가까운 probe solvent 를 기준으로, solvent와 접하는 경우 실 표면 원자가 되고, solvent와 원자 간의 교차 영역이 사용자가 정의한 최소 기준보다 큰 경우 유효 표면 원자라 정의하였다.

단백질 분자를 구성하는 원자들을 기하학적으로 표현할 경우, 구의 집합으로 표현하는 방법이 자주 사용된다. 구의 중심점은 원자의 중심점에 대응하고, 구의 반경은 원자의 반데르바스 (Van der Waals) 반경에 대응한다. (그림 1)에서 구의 집합으로 표현된 단백질 분자에 대해 실 표면 원자를 구한 예를 2차원으로 간략화하여 제시한다.

본 논문에서는 단백질 분자를 구의 집합 M으로 표현하고, probe solvent P를 일정한 반경을 가진 한 개의 구로 정의한 뒤, 주어진 solvent에 대해 실 표면 원자들을 발견하는 알고리즘을 제시한다. 표면 원자는 M에 대한 오프셋 곡면을 생성하는 과정을 통해 추출한다. 제안된 알고리즘은 구의 집합 M에 대해 복셀 맵(voxel map)을 구성하여 M에 대한 오프셋 곡면 계산의 효율성을 높이고, GPU를 기반으로 병렬 처리를 수행함으로써 효율적으로 표면 원자를 추출한다.

본 논문의 구성은 다음과 같다. 2절에서는 오프셋 곡면을 이용하여 표면 원자를 추출하는 방법을 제시한다. 3절에서는 실험결과를 제시하고, 4절에서 결론을 내린다.

2. 표면원자 발견 알고리즘

원자의 중심점과 반데르바스 반경 (Van der Waals radius)가 각각 $\mathbf{c}_i$, r_i라 할 때, 다음과 같이 구 $S(\mathbf{c}_i, r_i)$에 대응시킬 수 있다.

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

구 $S(\mathbf{c}_i, r_i)$와 내부의 모든 점을 포함하는 영역을 $B(\mathbf{c}_i, r_i)$로 나타낸다.

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

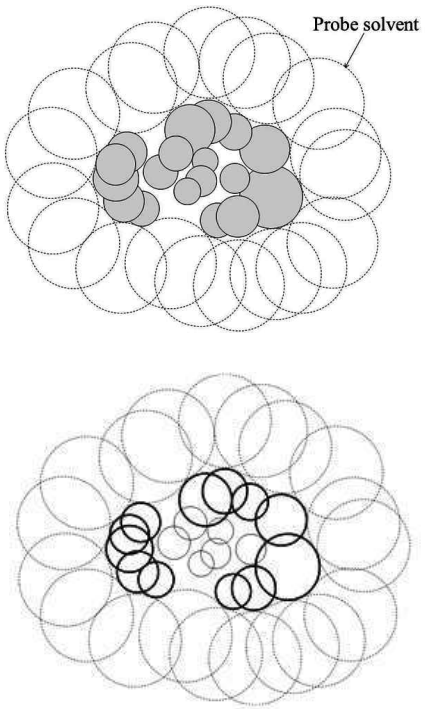
분자는 구의 집합 M으로 다음과 같이 정의된다.

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

본 논문에서 제시하는 표면 원자 발견 알고리즘은 다음의 관측을 기반으로 구성된다.

1. 분자 M과 접할 수 있는 probe solvent의 중심점의 궤적은 probe solvent의 반경 r 만큼 분자에 대하여 오프셋을 수행하여 얻은 곡면 M_{offset}=boundary surface of $\cup_{0 \le i < n} B(\mathbf{c}_i, r_i+r)$와 같다.
2. 분자 M에 속한 구 $S(\mathbf{c}_i, r_i)$를 거리 r로 오프셋한 결과가 M_{offset}에 포함될 경우, 해당 구는 표면 원자에 속한다. 즉, 구 $S(\mathbf{c}_i, r_i)$가 표면 원자에 속할 필요충분조건은 $S(\mathbf{c}_i, r_i+r) \cap M_{\text{offset}} \neq \varnothing$이다.

알고리즘은 M에 속한 각 원자의 오프셋 $B(\mathbf{c}_i, r_i+r)$에 대해 복셀 맵 V를 구성하는 단계, V를 이용하여 $S(\mathbf{c}_i, r_i+r)$ 중에서 M_{offset}과 교집합이 존재하는 것들을 표면 원자로 추출하는 두 단계로 구성된다.



(그림 1) 표면 원자의 예
(위: 단백질 분자(회색)와 이를 둘러싼 probe solvent (점선), 아래: 단백질 분자 중 표면 원자(굵은 선))

2.1 복셀 맵 구성

오프셋된 구의 집합과 이에 대한 바운딩 박스 (bounding box)가 주어질 때, 바운딩 박스를 $n_x \times n_y \times n_z$ 개의 복셀로 분할하여 복셀의 집합 V를 구성한다.

$$V = \{\ V^{\alpha\beta\gamma} \mid 0 \le \alpha < n_x,\ 0 \le \beta < n_y,\ 0 \le \gamma < n_z\}.$$

분자에 속한 각 구를 오프셋 거리 r로 오프셋팅하여구의 집합 M^r을 구한다. M^r에 속한 각 구를 B_i^r 로 표시할 때, $B_i^r = B(\mathbf{c}_i, r_i+r) = \{\mathbf{p} \mid \|\mathbf{c} - \mathbf{p}\| \le r+r_i\}$ 로 정의할 수 있다. $V^{\alpha\beta\gamma}.box$ 가 복셀 $V^{\alpha\beta\gamma}$의 영역을 나타낼 때, $V^{\alpha\beta\gamma}$.balls는 다음과 같이 복셀에 포함된 B_i^r 의 정보를 가진다.

$$V^{\alpha\beta\gamma}.balls = \{B_i^r \mid B_i^r \cap V^{\alpha\beta\gamma}.box \neq \varnothing\}.$$

알고리즘 1에서는 GPU를 이용하여 복셀 맵을 구성하는 과정을 보인다. 각 구 별로 쓰레드 (thread)가 할당되며, 쓰레드들은 병렬로 커널 (kernel) 프로그램을 수행한다. 각각

의 쓰레드는 해당 구가 교차하는 복셀들을 발견한 뒤, 각 복셀에 구의 정보를 저장하는 과정을 수행한다.

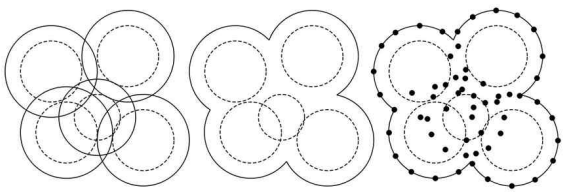
Algorithm 1: Voxel map construction
function VoxelMap (M^r, B^r)
Begin
For each $B_i^r \in M^r$, $0 \le i < n$, **do in parallel**/*GPU code */
  For each $V^{\alpha\beta\gamma}$.*box* that satisfies $B_i^r \cap V^{\alpha\beta\gamma}.box \neq \varnothing$ **do**
    Attach B_i^r to $V^{\alpha\beta\gamma}.balls$;
End.

2.2 표면 원자의 추출

분자 M의 오프셋 곡면 M_{offset}은 각 원자에 대한 오프셋 곡면의 합집합을 구한 뒤 경계 곡면 (boundary surface)을 추출한 것과 같다. 표면원자를 추출하는 단계는 다음과 같다. 각 원자의 오프셋 곡면 $S(\mathbf{c}_i, r_i+r)$를 균일하게 분포된 표본점들의 집합 P_i^r 으로 근사한다. P_i^r에 속한 점들 중에서 다른 원자의 오프셋 곡면 ($S(\mathbf{c}_j, r_j+r)$, $i \neq j$)에 속하지 않는 것이 존재할 경우, $S(\mathbf{c}_i, r_i)$는 표면 원자에 속한다((그림 2) 참조).

알고리즘 2는 구의 오프셋 곡면에 대한 표본점들을 이용하여 표면 원자를 추출하는 과정을 보인다. 알고리즘에서 FP는 표면원자의 집합을 나타내고, $B_i = B(\mathbf{c}_i, r_i)$, $B_i^r = B(\mathbf{c}_i, r_i+r)$이다.

알고리즘 2에서는 두 단계로 커널 프로그램이 사용된다. 첫번째 단계에서는 각각의 원자마다 쓰레드를 할당하여, 각 쓰레드가 구의 오프셋 곡면에 대한 표본점을 구하여 집합 P_i^r 를 구성한다. 두번째 단계에서는 각각의 표본점마다 쓰레드를 할당한다. 각 쓰레드는 해당 표본점이 다른 원자의 오프셋 곡면 내부에 포함되면 P_i^r 에서 삭제하고, 최종적으로 P_i^r 에 남아있는 표본점에 대응되는 원자들을 표면 원자의 집합인 FP에 저장한다.



(a) (b) (c)

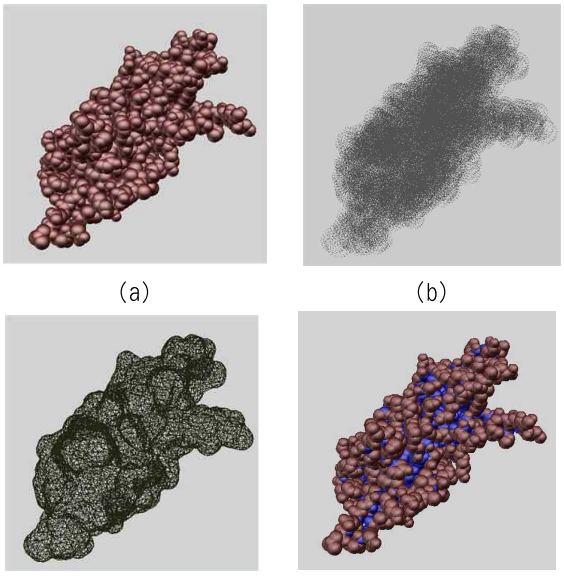
(그림 2) 표본 점을 이용한 표면 원자의 결정
(a) 원자 (점선)과 원자의 오프셋 곡면 (실선),
(b) 분자 (점선)와 분자에 대한 오프셋 곡면 (실선),
(c) 원자의 오프셋 곡면에 대한 표본 점

Algorithm 2: Sampling the offset surface
function SamplingOffsetSurface
Begin
  $FP = \varnothing$; /* Initialize the set of surface atoms */
  For each B_i^r, $0 \le i < n$, **do in parallel** /* GPU code */
    Generate $P_i^r = \{\text{Sampling points on } S(\mathbf{c}_i, r_i+r)\}$;
  For each $\mathbf{q} \in P_i^r$, $0 \le i < n$, **do in parallel** /* GPU code */
  begin
    For each $V^{\alpha\beta\gamma}$ such that $V^{\alpha\beta\gamma}.balls$ contains B_i^r **do**
      if $\mathbf{q} \cap B_j^r \neq \varnothing$, where $i \neq j$, **then**
        Remove $\mathbf{q}$ from P_i^r;
    if $P_j^r \neq \varnothing$ **then** add B_i to FP;
  end
  return FP;
End.

3. 실험 결과

실험은 AMD Athlon2 x4 640 CPU (3.0GHz), 8.0GB의 DRAM 및 nVidia GTX280 GPU가 장착된 컴퓨터에서 실행하였다. <표 1>과 <표 2>에서는 실험에서 입력으로 사용한 단백질 분자의 pdb id [10] 와 주어진 probe solvent의 반경 r = 1.4에 대해 계산하여 추출한 표면 원자의 개수를 제시하였다. 또한, 본 논문에서 제안된 알고리즘을 단일 CPU (single-core CPU)를 이용하여 계산한 결과 및 GPU 기반의 CUDA 프로그램을 이용하여 계산한 결과를 제시하였다. M_{offset}을 포함하는 bounding box는 각 변의 길이가 각각 2Å인 복셀로 분할된 경우와 각 변의 길이가 1Å인 복셀로 분할된 경우에 대해 각각 실험되었다. 실험 결과, GPU 기반의 프로그램은 CPU로 구현한 프로그램에 비해 8116,412개의 원자에 대해 약 1.93배에서 4.58배 계산 속도가 향상되었다.

(그림 3)에서는 한 개의 단백질 분자에 대해 추출한 오프셋 곡면과 표면원자를 보인다.



(a) (b) (c) (d)

(그림 3) 단백질 분자 1R95에 대한 계산 예: (a) 분자, (b) 각 원자의 오프셋에 대한 표본 점의 집합, (c) 분자에 대한 오프셋 곡면, (d) 표면 원자(붉은 색)와 비표면 원자(푸른 색)

〈표 1〉 각 변의 길이가 2Å인 복셀로 구성된 복셀 맵을 이용한 표면 원자 계산 결과

Pdb id	전체 원자의 개수	표면 원자의 개수	CPU 기반		GPU 기반		Speed up*
			수행시간 (ms)	Frame rate(FPS)	수행시간 (ms)	Frame rate(FPS)	
1VQI	811	473	16.49	60.64	8.56	116.82	1.93
1R95	1,468	805	30.38	32.92	9.14	109.41	3.32
1TNR	2,208	1,168	45.08	22.18	11.37	87.95	3.96
1B9T	3,040	1,226	62.47	16.01	15.59	64.14	4.01
1B44	4,186	1,747	87.19	11.47	22.60	44.25	3.86
1ACC	5,282	2,447	110.48	9.05	26.06	38.37	4.24
1AYM	6,412	2,747	137.33	7.28	38.68	25.85	3.55

*Speedup = (CPU기반 수행시간) / (GPU기반 수행시간)

〈표 2〉 각 변의 길이가 1Å인 복셀로 구성된 복셀 맵을 이용한 표면 원자 계산 결과

Pdb id	전체 원자의 개수	표면 원자의 개수	CPU 기반		GPU 기반		Speed up*
			수행시간 (ms)	Frame rate(FPS)	수행시간 (ms)	Frame rate (FPS)	
1VQI	811	473	20.92	47.80	9.21	108.58	2.27
1R95	1,468	805	38.53	25.95	9.81	101.94	3.93
1TNR	2,208	1,168	58.65	17.05	12.81	78.06	4.58
1B9T	3,040	1,226	82.57	12.11	18.36	54.47	4.50
1B44	4,186	1,747	114.49	8.73	25.85	38.68	4.43
1ACC	5,282	2,447	143.75	6.96	35.05	28.53	4.10
1AYM	6,412	2,747	171.81	5.82	42.87	23.33	4.01

*Speedup = (CPU기반 수행시간) / (GPU기반 수행시간)

4. 결 론

본 논문에서는 GPU 기반으로 단백질 분자에서 표면 원자를 효율적으로 추출하는 방법을 제안하였다. 단백질 분자를 구성하는 원자들을 구의 집합으로 표현하고, probe solvent를 고정된 반경의 구로 표현한다. Probe solvent의 반경을 오프셋 거리로 사용하여 각 구의 오프셋 곡면을 구한 뒤, 이를 표본 점의 집합으로 근사한다. 표본 점들이 분자의 오프셋 곡면에 포함되는 구를 표면 원자로 추출한다. 제안된 방법은 GPU를 활용한 병렬처리를 수행하여 최대 6,412개 가량의 원자를 갖는 분자에 대해 42.87millisecond 내에 표면 원자를 발견한다.

참 고 문 헌

[1] Can, T., Chen, C.I., Wang, Y.F.: Efficient molecular surface generation using level-set methods. J. of Molecular Graphics and Modeling 25(4), 442-454, 2006.

[2] Connolly, M.L.: Analytical molecular surface calculation. J. Appl. Crystallogr. pp.548-558, 1983.

[3] Connolly, M.L.: Solvent-accessible surfaces of proteins and nucleic acids. Science 221, 709-713, 1983.

[4] Nicholls, A., Sharp, K., Honig, B.: Protein folding and association: insights from the interfacial and thermodynamic properties of hydrocarbons. Proteins 11(4), 281-296, 1991.

[5] Pettersen, E.F., Goddards, T.D., Huang, C.C., Couch, G.S., Greenblatt, D.M., Meng, E.C., Ferrin, T.E.: Ucsf chimera: a visualization system for exploratory research and analysis. J. Comput. Chem. 25(13), 1605-1612, 2004

[6] Ryu, J., Park, R., Kim, D.S.: Molecular surfaces on proteins via beta shapes. Computer-Aided Design 39, 1042-1057, 2007.

[7] Sanner, M.F., Spehner, J.C., Olson, A.J.: Reduced surface: an efficient way to compute molecular surfaces. Biopolymers 38(3), 305-320, 1996.

[8] Zauhar, R.J.: Smart: a solvent-accessible triangulated surface generator for molecular graphics and boundary element applications. J Comput Aided Mol Des 9(2), 149-159, 1995.

[9] Deanda, F., Pearlman, R.S.: A novel approach for identifying the surface atoms of macro-molecules. Journal of Molecular Graphics and Modelling 20, 415-425, 2002.

[10] http://www.pdb.org

김 병 주

e-mail : kbj113@hotmail.com
2002년 경북대학교 전자전기공학부(학사)
2004년 경북대학교 전자공학과(공학석사)
2006년 경북대학교 전자공학과(박사수료)
2006년～2010년 (주)휴원 과장
2010년～2011년 (주)LG전자MC연구소 선임연구원 과장
2011년～현 재 경북대학교 전자공학과 박사과정
관심분야 : 컴퓨터그래픽스, 컴퓨터비전, 기하 모델링, 병렬처리 등

김 구 진

e-mail : kujinkim@yahoo.com
1990년 이화여자대학교 전자계산학과(학사)
1992년 한국과학기술원 전자계산학과(석사)
1998년 포항공과대학교 컴퓨터공학과(박사)
1998년～2000년 Dept. of Computer Sciences, Purdue University, PostDoc.
2000년～2002년 아주대학교 정보통신전문대학원 BK21조교수
2002년～2003년 Dept. of Mathematics and Computer Science, University of Missouri-St. Louis, Visiting Assistant Professor
2004년～2007년 경북대학교 컴퓨터학부 조교수
2008년～현 재 경북대학교 컴퓨터학부 부교수
관심분야 : 컴퓨터 그래픽스, 컴퓨터 애니메이션, 곡면 및 기하 모델링 등

성 준 경

e-mail : seong@ssu.ac.kr
2000년 서울대학교 전산학과(학사)
2005년 서울대학교 컴퓨터공학부(공학박사)
2005년～2008년 University of Utah School of Computing 박사후연수
2008년～2010년 카이스트 전산학과 연구조교수
2010년～현 재 숭실대학교 컴퓨터학부 조교수
관심분야 : 컴퓨터 그래픽스, 기하 알고리즘, 계산 해부학 등