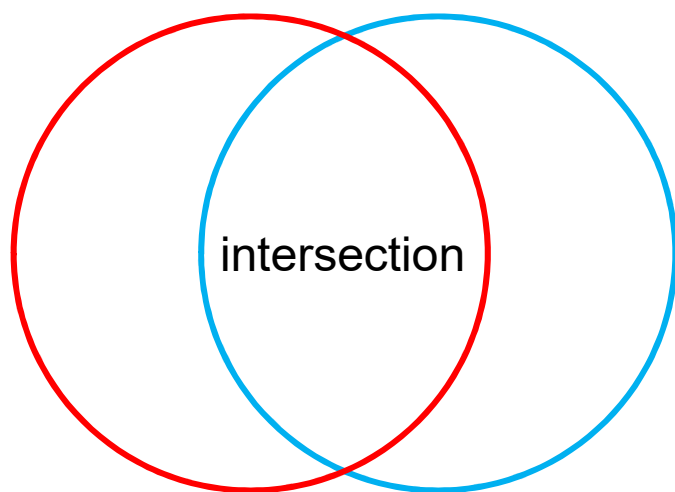


# **CONTACT CURVE BASED SIMULATION OF SIDE CHAINS FROM TWO AMINO ACIDS IN A PROTEIN MOLECULE**

**SANGHUN JEONG  
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING  
KYUNGPOOK NATIONAL UNIVERSITY, KOREA**

# 연구동기

다른 학부 연구실과 함께 단백질 분자모형 시뮬레이션을 제작하고 있었습니다. 단백질 분자모형 시뮬레이션을 만드는데 있어서 하나의 문제가 생겼습니다. 현실에 존재하는 단백질 분자모델은 물리적으로 겹칠 수 없는데 반해 컴퓨터로 계산한 시뮬레이션은 충돌이 일어나도 그냥 그리기만 할 뿐 따로 경고나 오류 메시지를 내보내지 않습니다. 그래서 우리는 그런 비현실적인 상황이 발생했을 때 감지 및 처리를 할 수 있는 것을 만들기로 하였습니다.



# 문제 정의 및 배경지식

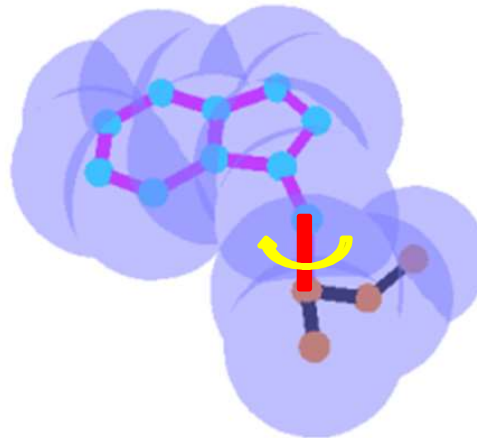
단백질 분자는 여러 개의 아미노산으로 이루어져 있습니다.

각 아미노산은 메인체인과 사이드체인으로 구성되어 있습니다.

아미노산의 메인체인은 고정되어 있지만 메인체인에 연결되어 있는 사이드체인은 각각의 축을 따라 회전할 수 있습니다.

사이드체인들이 각각의 축을 따라 회전하다 보면 다른 아미노산의 사이드체인과 충돌이 발생하는 경우가 생기게 됩니다.

그렇기 때문에 우리는 그런 상황을 처리하기 위한 알고리즘을 만들었습니다.



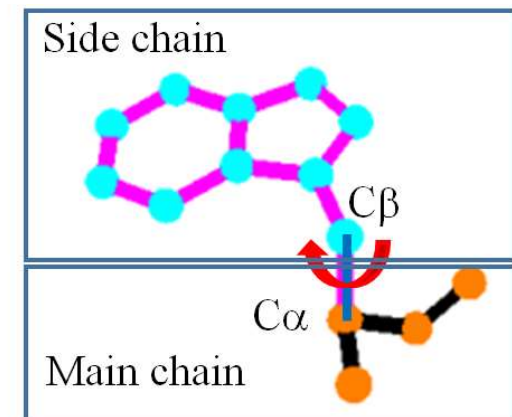
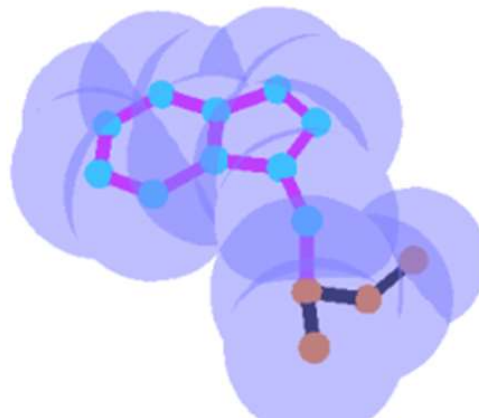
# 문제 정의 및 배경지식

단백질 분자는 연속적인 아미노산들로 이루어져 있으며 이 아미노산들은 아톰으로 구성되어 있습니다.

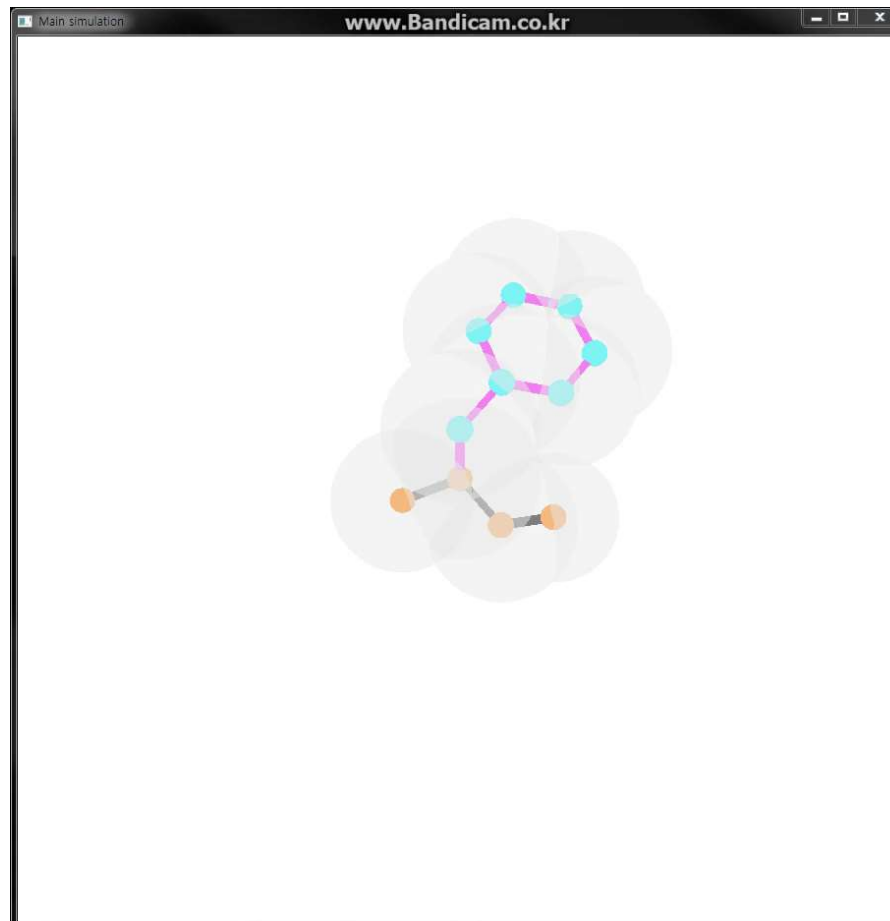
단백질 분자의 기하학적 표현은 **van der waals** (반 데르 발스) 반지름을 갖는 구로 구성되어 있습니다.

**2**개의 원자가 서로 공유결합을 갖는 경우 서로 겹쳐있으며 이를 이용하여 표시해 두었습니다.

단백질 분자의 아미노산은 **main chain**(주황) 과 **side chain**(청록) 으로 나누어진 **2**개의 원자그룹이 있습니다.



# 문제 정의 및 배경지식

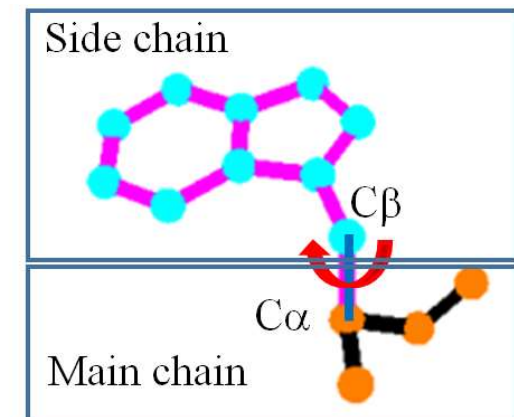
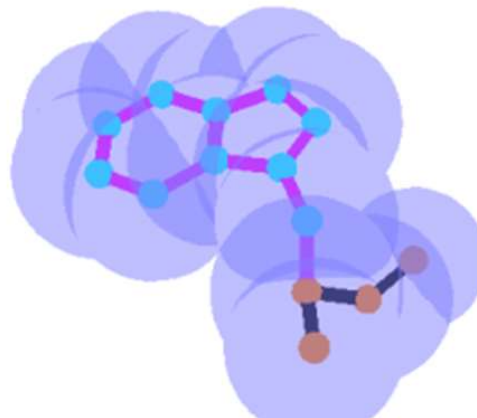
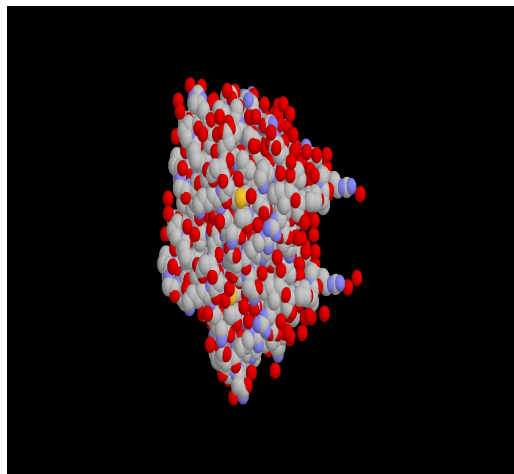


# 문제 정의 및 배경지식

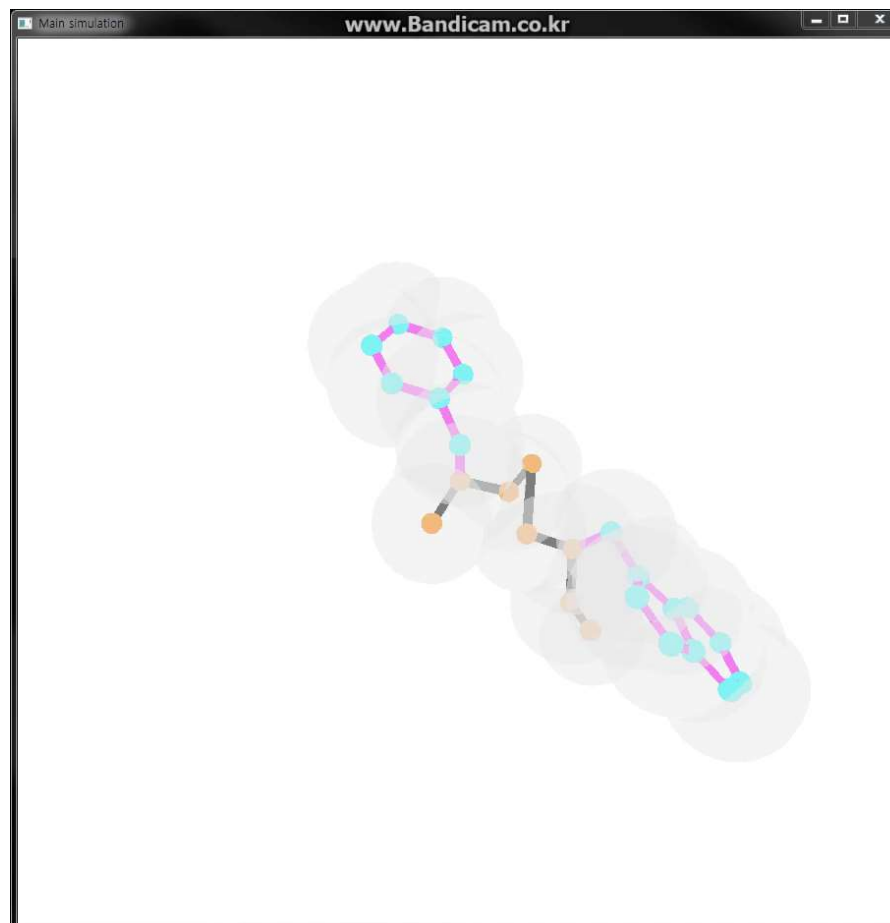
아미노산의 **side chain**은 **main chain**의 **carbon alpha**와 **side chain**의 **carbon beta**를 축으로 회전하게 됩니다.

시뮬레이션 상에서 각 아미노산의 **side chain**이 회전을 하다 보면 다른 아미노산의 **side chain**과 충돌을 일으킬 수 있습니다.

하지만 충돌이 일어나도 한 단백질 분자 안에 아미노산들이 매우 많기 때문에 사람의 눈으로 확인할 방법이 거의 없습니다. 그렇기 때문에 우리는 충돌이 일어나는 상황을 미리 체크하여 벗어나기 위해 이 논문에서 소개하는 알고리즘을 만들었습니다.



# 문제 정의 및 배경지식

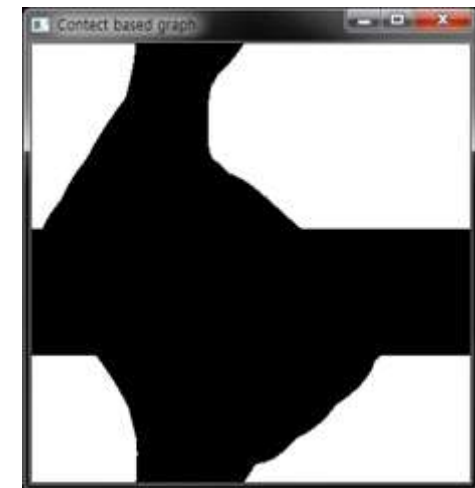
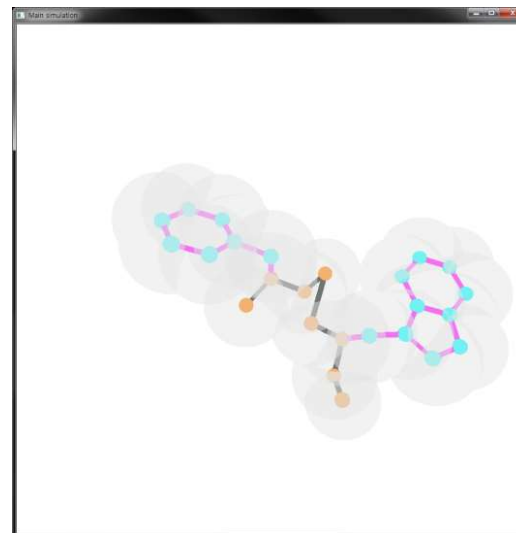


# 입출력

우리는 **pdb**파일에 미리 정의되어 있는 복합아미노산 구조 중 **2개의 아미노산 3차원 좌표**만을 이용하여 **3차원 아미노산 운동 시뮬레이션**과 **2차원 충돌범위 그래프**를 제작하였습니다.

Table 1. Input PHE and TRP information.

ATOM	1	N	PHE	A	1	21.320	22.197	64.569	1.00	10.94
ATOM	2	CA	PHE	A	1	19.900	22.372	64.205	1.00	11.49
ATOM	3	C	PHE	A	1	19.058	22.732	65.374	1.00	12.09
ATOM	4	O	PHE	A	1	17.939	23.205	65.206	1.00	12.13
ATOM	5	CB	PHE	A	1	19.317	21.042	63.629	1.00	12.11
ATOM	6	CG	PHE	A	1	20.330	20.438	62.718	1.00	8.33
ATOM	7	CD1	PHE	A	1	20.663	21.045	61.540	1.00	8.88
ATOM	8	CD2	PHE	A	1	21.011	19.291	63.124	1.00	7.80
ATOM	9	CE1	PHE	A	1	21.691	20.538	60.771	1.00	10.46
ATOM	10	CE2	PHE	A	1	22.046	18.812	62.377	1.00	9.63
ATOM	11	CZ	PHE	A	1	22.389	19.443	61.204	1.00	8.54
ATOM	12	N	TRP	A	2	19.533	22.544	66.617	1.00	10.96
ATOM	13	CA	TRP	A	2	18.829	22.870	67.838	1.00	10.24
ATOM	14	C	TRP	A	2	19.950	23.033	68.928	1.00	11.45
ATOM	15	O	TRP	A	2	20.112	22.121	69.741	1.00	11.40
ATOM	16	CB	TRP	A	2	17.870	21.716	68.159	1.00	9.36
ATOM	17	CG	TRP	A	2	16.866	22.129	69.200	1.00	8.64
ATOM	18	CD1	TRP	A	2	16.781	21.770	70.478	1.00	9.52
ATOM	19	CD2	TRP	A	2	15.749	23.012	68.983	1.00	13.34
ATOM	20	NE1	TRP	A	2	15.725	22.368	71.102	1.00	11.57
ATOM	21	CE2	TRP	A	2	15.072	23.163	70.204	1.00	14.95
ATOM	22	CE3	TRP	A	2	15.299	23.744	67.862	1.00	19.41
ATOM	23	CZ2	TRP	A	2	13.925	23.970	70.310	1.00	16.89
ATOM	24	CZ3	TRP	A	2	14.170	24.536	67.988	1.00	18.89
ATOM	25	CH2	TRP	A	2	13.524	24.639	69.209	1.00	15.29



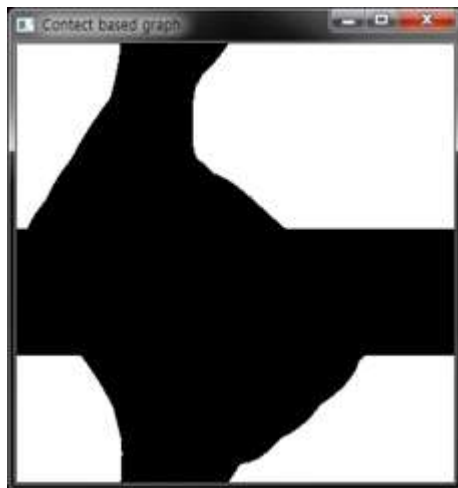


# 그래프 알고리즘 설명

우측에 보이는 알고리즘을 이용하여 **A, B** 아미노산들의 사이드체인이 각각 **360도** 회전 할 동안 충돌이 있는지를 체크해 냅니다.

충돌이 있으면 충돌이 있는 부분을 검게 만들고 충돌이 없는 부분은 흰색으로 놉니다.

검사가 끝나게 되면 아래와 같은 **360X360** 그래프가 생성됩니다.



## Algorithm 1: Collision Area Computation

Input:  $\delta$  // the step size

$A_i(s)$ ,  $0 \leq i < N_A$  // Atoms in side chain A

$B_j(t)$ ,  $0 \leq j < N_B$  // Atoms in side chain B

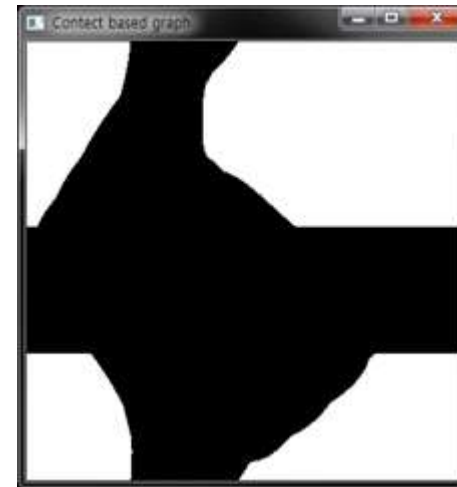
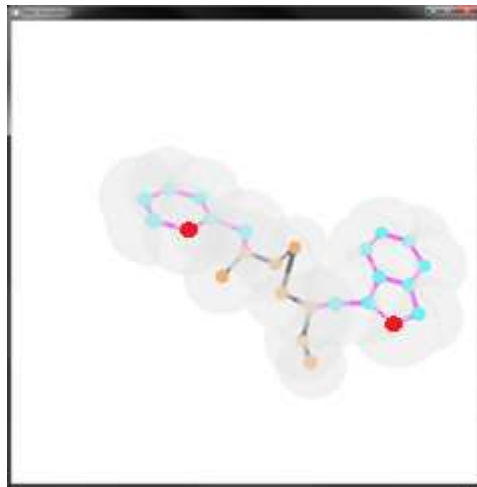
```
1 BEGIN
2   for (  $s = 0$  ;  $s < 2\pi$  ;  $s += \delta$  ) BEGIN
3     for (  $t = 0$  ;  $t < 2\pi$  ;  $t += \delta$  ) BEGIN
4        $M[s][t] = \text{FALSE}$ ;
5       for (  $i = 0$  ;  $i < N_A$  ;  $i++$  ) BEGIN
6         for (  $j = 0$  ;  $j < N_B$  ;  $j++$  )
7           if (  $A_i(s) \cap B_j(t) \neq \emptyset$  ) BEGIN
8             Set TRUE to  $M[s][t]$ ;
9             Break;
10          END
11        END
12      END
13    END
14  END
```

# 그래프 설명

위 알고리즘을 이용하여 생성된 그래프 중 입니다.

좌측 그래프는 각 아미노산에서 특정한 원자 한 개 씩만 회전에 따라 충돌하는지 추출한 것이고 그렇게 생성된 그래프들을 모두 합치면 가운데 그래프와 같은 모양이 됩니다.

가운데와 같은 그래프들을 모두 모으면 우측 그래프 모양이 됩니다.



# 그래프 설명

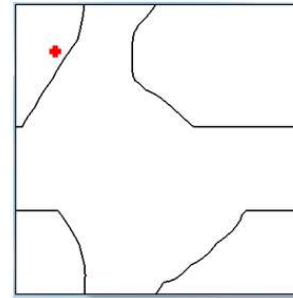
다음 그림들은 왼쪽 **amino**의 회전 각에 따른 그래프와 시뮬레이션의 변화입니다.

그림 a에서는 겹치는 구간이 없는 그림입니다

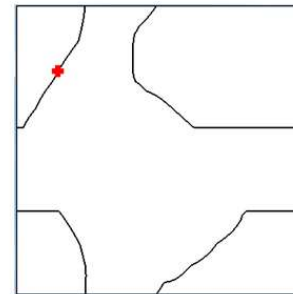
그림 b는 최초의 **contact** 상태이며

그림 c에서는 완전히 **intersection**된 상태입니다.

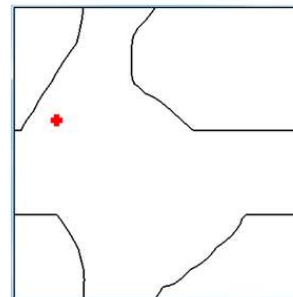
옆의 그림을 보면 우측 **side chain**은 고정되어 있고 좌측 **side chain**만 움직이므로 그래프에서도 붉은 점이 세로방향으로만 움직이는 것을 알 수 있습니다.



(a) Free



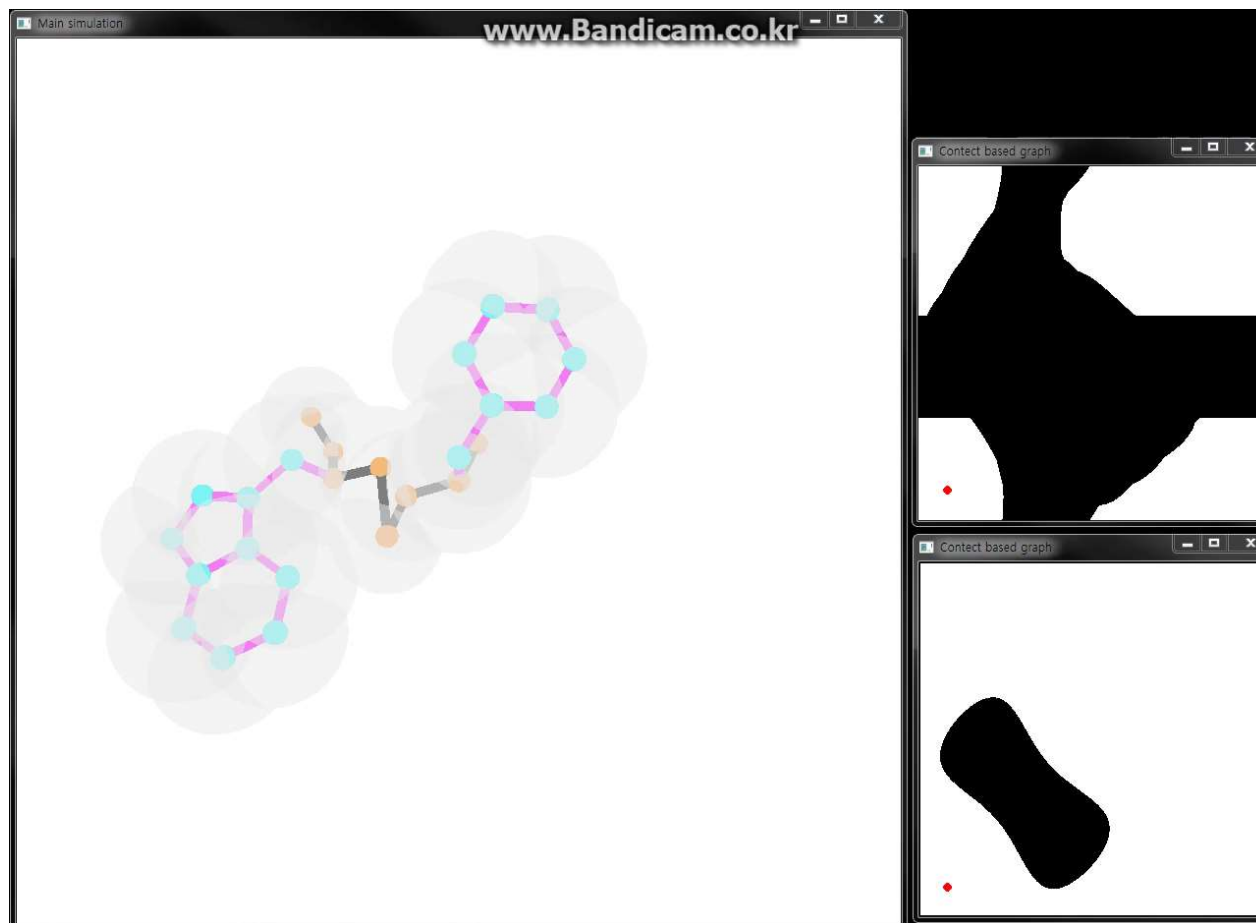
(b) Point of contact



(c) Intersection



# 그래프 설명

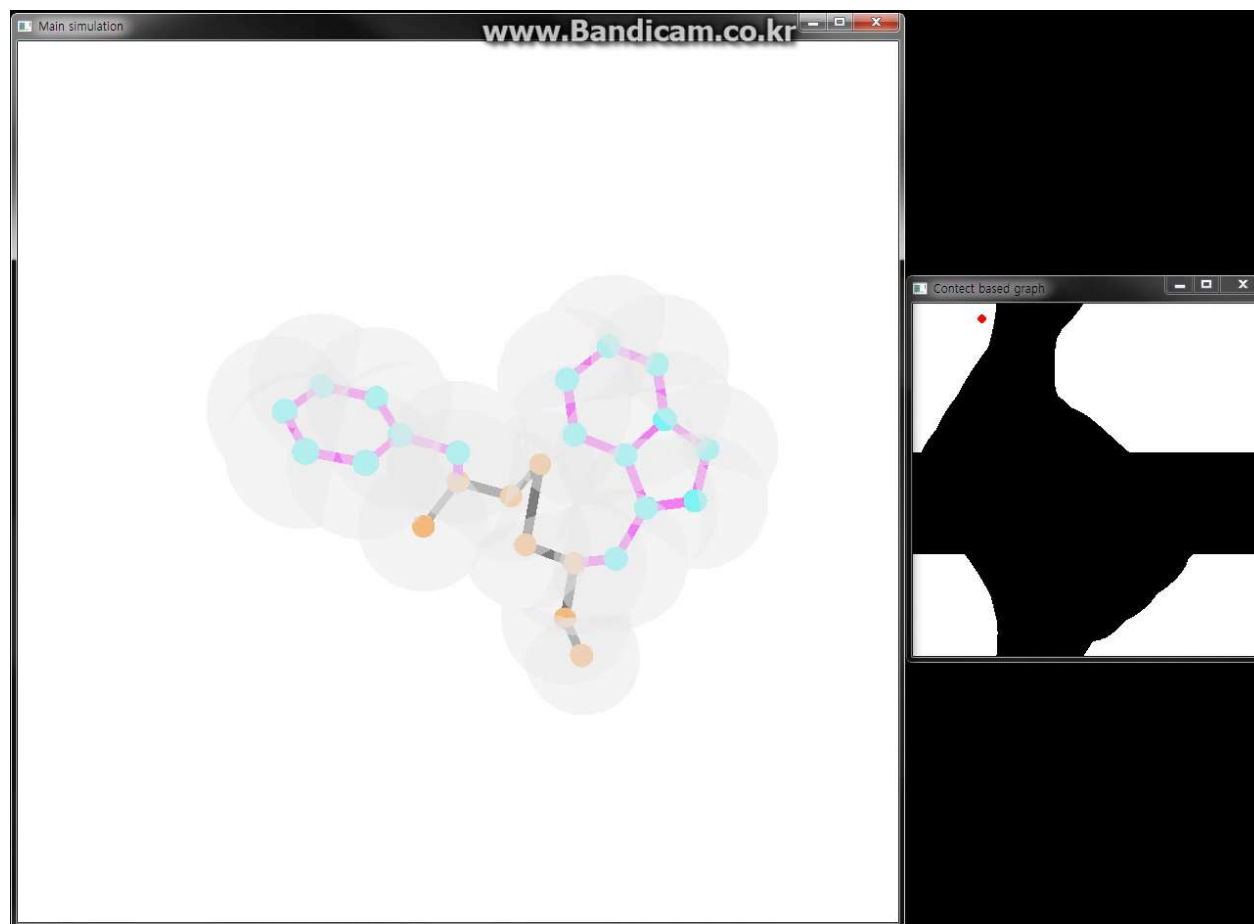


## 결과

위에서 생성된 그래프를 가지고 시뮬레이션 다시 하였습니다.

이번에는 사이드체인이 회전을 하다가 충돌하는 영역에 들어가면 회전하는 아미노산과 부딪히는 아미노산의 사이드체인을 충돌 하지 않는 영역이 나올 때 까지 밀도록 하였으며 만약 모든 구간이 충돌인 경우에 기존 아미노산의 회전을 멈추도록 하였습니다.

# 결과



# 향후계획

앞으로 더 많은 아미노산들을 그릴 때 아미노산들의 충돌 여부를 더 빠르게 계산하고 더 적은 수의 그래프에 더 많은 아미노산 충돌정보를 쉽게 볼 수 있도록 만들 계획입니다.

# THANK YOU

[tkdgns3042@naver.com](mailto:tkdgns3042@naver.com)