



공학석사학위논문

분기 구조를 갖는 스윕 모델의 미세구조 모델링

Microstructure Modeling of Branch Structured Sweep Models

2022 년 2 월

서울대학교 대학원 컴퓨터공학부 함 유 경

공학석사학위논문

분기 구조를 갖는 스윕 모델의 미세구조 모델링

Microstructure Modeling of Branch Structured Sweep Models

2022 년 2 월

서울대학교 대학원 컴퓨터공학부 함 유 경

분기 구조를 갖는 스윕 모델의 미세구조 모델링

Microstructure Modeling of Branch Structured Sweep Models 지도교수 김 명 수

이 논문을 공학석사 학위논문으로 제출함 2022 년 1 월 서울대학교 대학원 컴퓨터공학부 함 유 경

함유경의 공학석사 학위논문을 인준함 2022 년 1 월

위 원 장 <u>신영길</u> (인)

부위원장 김명수 (인)

위 원 _____ 서진욱 ____ (인)

요약

본 논문에서는 두 실린더가 서로 교차하는 형태의 혈관 모델을 효과적으로 모델링하고 이를 미세구조로 채워 넣는 방법에 대해 소개한다. 먼저, 두 실린더 형태의 분기 구조 사이에 생기는 가파른 곡면을 롤링 볼 블렌딩 방법을 사용하여 부드럽게 만들 것을 제안한다. 이를 위해 두 분기 구조를 동시에 지나는 커널 곡면을 생성하는 공식을 제시한다. 이에 더해, 혈관 모델의 크기 및 용도에 따라 사용자가 원하는 정도의 부드러운 커널 곡면을 만들어낼 수 있음을 보인다.

블렌딩된 분기 구조를 미세구조로 채우기 위해, 각 혈관 모델 안의 공간을 재매변수화하는 방법을 제시한다. 모델을 더 작은 크기의 그리드들로 나눈 후, 각 그리드를 감싸는 구로 이루어진 bounding volume hierarchy를 얻을 수 있다. 이 자료 구조를 활용한 충돌 검사를 통해 두 실린더가 교차하는 부분을 굉장히 빠르 고 쉽게 구분할 수 있다. 또한, 미세구조 모델링에 필요하지 않은 나머지 조각들을 이 자료구조를 이용하여 효율적으로 찾아 내어 제거할 수 있다.

실린더 모델 외에도, 앞서 분기 구조에 생성한 커널 곡면을 여러 개의 bounding volume으로 나누어 비슷한 크기의 미세구조로 채워 넣는 방법에 대해 제시 한다. 또한, 두 실린더가 교차하는 공간 안을 다른 미세구조와 부드럽게 연결될 수 있도록 효율적으로 채워 넣는 방법에 대해 제시한다.

주요어: 미세구조, 롤링 볼 블렌딩, 분기 구조, Bounding Volume Hierarchy, 커널 곡면, 3D 모델링 학번: 2020-23988

목차

요약	i
목차	ii
그림 목차	iv
표 목차	vii
제 1 장 서론	1
1.1 연구 배경	1
1.2 연구 필요성	4
1.3 논문의 구성	6
제 2 장 예비 지식	7
2.1 스윕 곡면	7
2.2 실린더	8
제 3 장 관련 이전 연구	10
제 4 장 분기 구조 모델링 알고리즘	12
4.1 분기 구조 블렌딩	12
4.1.1 실린더 간의 교차 곡선	12
4.1.2 커널 곡면	15
4.1.3 여러 각도에서 교차하는 실린더의 분기 구조	20

	4.1.4	시행착오와 고찰	23
4.2	BVH	생성 알고리즘	24
	4.2.1	충돌 검사를 위한 BVH	24
	4.2.2	잉여 조각을 검출하기 위한 BVH	30
4.3	미세구	-조 모델링	31
	4.3.1	타일 정의	32
	4.3.2	그리드의 타일 매개변수화	34
	4.3.3	커널 곡면의 타일 매개변수화	35
	4.3.4	교차하는 그리드 미세구조 모델링	41
제 5 장	실험	결과	48
제 6 장	결론		54
참고문형	<u>H</u>		54
Abst	tract .		60

그림 목차

그림 1	1	다양한 재료와 특성을 지닌 미세구조로 모델링된 물체들[8] .	1
그림 1	2	각종 의료 기관에서 사용되는 미세구조 3D 프린팅 연구 사례	2
그림 1	3	미세구조를 통해 관절의 유연성을 조절한 손가락 모형[17] .	3
그림 1	.4	곡면의 블렌딩(blending)[11]	4
그림 2	2.1	스윕 볼륨으로 만든 매듭 모양[26]	7
그림 2	2.2	물체를 정의한 후 이를 이동시켜 얻는 스윕 볼륨[32]	7
그림 2	2.3	롤링 볼 스윕 볼륨[27]	8
그림 2	2.4	실린더	9
그림 4	l.1	90도로 만나는 두 개의 실린더	13
그림 4	1.2	교차 곡선 방정식	14
그림 4	1.3	(\delta+반지름)의 크기를 가지는 실린더와 교차곡선	15
그림 4	1.4	서로 맞닿고 있는 구와 실린더의 표현	16
그림 4	1.5	직교하는 벡터들로 새로 만들어낸 직교좌표계	16
그림 4	1.6	메인 실린더와 서브 실린더의 커널 스페이스. 메시	17
그림 4	ł.7	실린더와 커널 곡면의 평면도	18
그림 4	1.8	구가 굴러가는 궤적으로 생기는 커널 곡면과 샘플링된 점들	18
그림 4	1.9	90도로 교차하는 두 실린더 사이의 커널 곡면	19
그림 4	4.10	ϕ 의 각도로 만나는 두 실린더	20
그림 4	1.11	3차원에서 점과 직선과의 거리	21

그림 4.12	<i>t</i> 값의 샘플링 범위	22
그림 4.13	오류의 예시	23
그림 4.14	두 개의 실린더가 교차할 때 생기는 부분	25
그림 4.15	[<i>u</i> , <i>v</i> , <i>w</i>] 파라미터로 나타낸 각각의 그리드	26
그림 4.16	각 그리드를 감싸는 바운딩 스피어	26
그림 4.17	BVH와 BVTT 구조	28
그림 4.18	구의 충돌 검사	29
그림 4.19	그리드 충돌 검사의 실제 예시이다. 충돌하는 경우 빨간색,	
	충돌하지 않는 경우 초록색이다	29
그림 4.20	노란색으로 표시된 부분은 충돌이 검사되지 않았지만 사라	
	져야 하는 부분이다	30
그림 4.21	메인 실린더 바운딩 스피어 C_1 의 반지름 R 보다 그리드 조각	
	의 바운딩 스피어 C_2 와의 거리 d 가 더 작을 경우 충돌한다	31
그림 4.22	사용되는 타일의 기본 구조	32
그림 4.23	(a)각각 [4,2,4]개의 그리드로 나뉜 실린더. (b) 타일을 채워	
	넣은 것	33
그림 4.24	(a) 그리드의 매개변수 (b) 그리드 타일	35
그림 4.25	(a) 비어있는 커널 곡들 (b) 상자들로 나뉜 커널 곡면	36
그림 4.26	커널 곡면 안의 상자 메시 생성	37
그림 4.27	커널 상자 섭디비전(Subdivision)	37
그림 4.28	새로 생긴 상자에 맞게 변형된 타일	38
그림 4.29	타일이 커널 곡면 안에 채워지는 과정	39
그림 4.30	미세구조로 채워진 커널 곡면 상자들	40
그림 4.31	미세구조로 채워진 커널 곡면	40

그림 4.3	2 타일의 정의[12]	41
그림 4.3	3 타일 간의 스위핑 연결[12]	42
그림 4.34	1 커널 곡면이 존재하는 그리드의 렌더링	43
그림 4.3	5 (a) 커널 곡면의 앵커링 면 (b) 부드럽게 변형된 커널 곡면	
	타일	44
그림 4.3	5 브리지 타일과 커널 곡면 타일의 스위핑 연결	45
그림 4.3	7 두 타일 간의 Hermite 보간	46
그림 4.3	8 커널 곡면 타일의 Hermite 보간	47
그림 5.1	미세구조가 채워진 분기 구조	49
그림 5.2	여러 각도의 블렌딩 곡면 (순서대로 30, 45, 60도)	50
그림 5.3	여러 각도의 미세구조 모델링 (순서대로 30, 45, 60도)	51
그림 5.4	연결된 미세구조가 채워진 분기 구조	52
그림 5.5	내부의 연결된 미세구조	52

제 1 장 서론

1.1 연구 배경

최근 3D 프린팅 및 모델링 분야에서 수십 미크론(micron) 크기의 미세구조 (microstructure)를 활용하여 물체를 제조하는 연구에 관한 관심이 급증하고 있 다. 매우 작은 크기의 미세구조는 물체의 실제 물리적 특성을 반영할 수 있으며 적 은 비용으로 더 가볍고 유연한 물질을 만들 수가 있다는 장점이 있다. 이러한 미세 구조를 한 층씩 쌓아가며 프린팅하는 적층 제조 기법(additive manufacturing)[8] 으로 완성된 물체는 의료, 건축, 자동차, 의류, 교육 등 다양한 산업 분야에서 사 용될 수 있다.



그림 1.1: 다양한 재료와 특성을 지닌 미세구조로 모델링된 물체들[8]



(b) Wake Forest Institute for Regenerative Medicine 2015

그림 1.2: 각종 의료 기관에서 사용되는 미세구조 3D 프린팅 연구 사례

미세구조 3D 프린팅 기술은 이 중에서도 특히 의공학 분야에서 많이 활용되고 있다. 의족, 의수, 보철 등과 같은 의료 보형물 및 인공 장기 제작에 사용되는 것은 물론, 수술 시뮬레이션, 의료진 교육 등 다양한 의료 산업 분야에 폭넓게 기여하고 있다. 실제로, 현재 많은 의공학 연구소 및 의료 기기 회사에서 3D 미세구조를 조직 공학에 적용하여 해부학에 기반한 실제 인공 장기를 만들거나 손상된 장 기를 복구시키는 등 다양하게 응용한 사례가 있다(그림 1.2). 가장 유명한 연구 예시로는, 뼈와 연골 등의 근골격계 조직(tissue)을 미세한 3차원 스캐폴드 단백 질 구조로 만들어낸 Hurmacher[14]의 연구가 있다. 또한, Wake Forest Institute 에서 이러한 미세구조 3D 프린팅을 통해 재생의학 분야에 실제로 사용되는 여 러 인공 장기들을 만드는 연구를 활발하게 진행 중이다. 그 외에도, 3D 프린팅 기술을 사용하여 개인 맞춤형 의족 및 의수를 만드는 스타트업 회사의 예시도 있다.

컴퓨터 그래픽스 및 모델링 분야에서는 이러한 의료 제조 산업에 사용되는 미세구조를 얼마나 더 복잡하고 정교한 형태로 만들어낼 수 있는지에 관한 연 구가 주로 진행되고 있다. 일반적으로 사용되는 적층 제조 기술에서, 미세구조 자체의 구조적 복잡성과 실제 생산 비용 및 시간은 크게 연관이 없기 때문에 오히 려 적은 재료를 사용하여 복잡한 형태를 만들어내는 것이 가격에서 더 유리하게 여겨진다. 또한, 물체를 이루는 미세구조가 더 복잡할수록 물체 전체의 탄성, 투 과성 등의 다양한 특성들을 조절하는 데에 유리하기 때문에 이를 중점으로 두고 다양한 접근을 하고 있다[21]. 미세구조의 밀도와 재질을 다르게 하여 그림 1.4와

같은 실제 구부러지는 손가락과 비슷한 형태를 만들어낸 사례도 있다[17].



그림 1.3: 미세구조를 통해 관절의 유연성을 조절한 손가락 모형[17]

사람의 몸 구조는 개인마다 각기 다른 특성이 있고 매우 섬세하므로 인공 장기 들을 이에 맞추어 더 정교하고 자세하게 제작(fabrication)하는 것은 생체 의공학 분야에서 매우 중요한 문제이다. 또한, 의사 및 의료 종사자들이 수술 시뮬레이 션을 할 때 매번 실제 장기를 사용할 수 없어 보통 인공 장기를 사용하는데, 이 장기들이 실제 장기와 최대한 흡사해야 수술의 성공률을 높이는 데 도움이 된다. 하지만, 매번 인공 장기를 프린팅한 후 이를 환자에 맞추어 변형시키는 데에는 한계가 있으므로 이를 프린팅 전에 최대한 정확하게 모델링하는 것이 요구된다.

본 논문은 미세구조를 사용하여 여러 인체 장기 중에서도 가장 가장 복잡하고 섬세한 편에 속하는 혈관을 모델링 하고자 한다. 혈관은 외막, 중막, 내막의 총 3개의 막으로 구성되어 있고, 각 막은 서로 다른 특성이 있기 때문에 더 세밀한 미세구조 모델링이 필요하다. 또한, 한 줄기로 이루어져 있는 것이 아니라 매우 얇고 복잡한 형태로 뻗어나가는 형태를 하고 있으므로 여러 혈관이 서로 만났을 때 교차하는 부분을 잘 처리해주어야 한다. 복잡하고 정교한 혈관 모델링은 특히 뇌졸중 혹은 뇌동맥류의 치료에서 굉장히 중요한 역할을 할 것으로 보인다.

1.2 연구 필요성

본 논문은 서로 교차하는 형태의 혈관을 효과적으로 모델링하고 이를 미세 구조로 채워 넣는 방법을 중점으로 논의한다. 혈관을 이루는 것은 기본적으로 실린더 형태의 스윕 볼륨 모델인데, 이러한 스윕 모델이 서로 만나면 가지와 같은 형태의 분기 구조가 생기게 된다. 단순히 실린더 형태가 합쳐져 생기는 분기 구 조는 매우 가파른 곡면을 가지게 되기 때문에 이를 실제 혈관과 비슷한 부드러운 곡면으로 만들 필요가 있다.



그림 1.4: 곡면의 블렌딩(blending)[11]

또한, 이렇게 블렌딩이 된 분기 구조를 미세구조로 채우기 위해서는 해당 곡면 안의 공간을 새롭게 정의하여 매개변수화(parameterize)시켜야 한다. 정리하자 면, 본 논문의 연구는 두 실린더 형태가 교차하여 생기는 분기 구조에 롤링 볼 블렌딩 방법(rolling-ball blending)[11, 27]을 적용하여 이를 부드럽게 만들고 이 안의 공간을 미세구조로 채우는 방법에 대해 제시한다.

본 논문에서 제시하는 방법과 기여한 점을 요약하면 다음과 같다.

- 두 개의 실린더가 서로 교차할 때 생기는 분기 구조에 롤링 볼 블렌딩 방법을 사용하여 두 실린더 사이를 부드럽게 연결하는 커널(canal) 곡면을 생성한
 다. 기하학 수식을 사용하여 블렌딩을 구현하기 때문에 사용자가 원하는
 크기의 부드러운 곡면을 만들어낼 수 있다.
- 각 실린더를 [u, v, w]의 파라미터를 가지는 그리드(grid)로 분할한 후 각각 의 그리드를 바운딩 스피어(bounding sphere)로 감싼다. 이 구들을 BVH (Bounding Volume Hierarchy) 자료구조를 통해 두 개의 실린더가 교차 하는 부분을 구분할 수 있다. 이를 통해, 두 실린더의 그리드를 하나씩 비 교하는 방법보다 더 빠른 시간 안에 교차하는 부분을 알아내어 처리할 수 있다.
- 본 논문에서는 생성한 커널 곡면을 여러 개의 바운딩 볼륨(Bounding Volume)으로 나누어 안에 비슷한 크기의 미세구조를 채워 넣는 방법에 대해 논한다. 또한, BVTT를 통해 제거된 공간 안의 미세구조를 다른 실린더 그리드의 미세구조와 부드럽게 연결될 수 있도록 채워넣는 방법에 대해 설명한다.

1.3 논문의 구성

본 논문의 구성은 다음과 같다. 먼저, 제2장에서는 스윕 곡면과 실린더의 수학 적 정의에 대한 간단한 기초 지식에 대해 다룬다. 그 후, 제3장에서는 BVH, 곡면 블렌딩, 미세구조 모델링에 대한 이전 연구들에 대해 살펴본다. 제4장에서는 본 논문의 알고리즘에 대한 자세한 설명을 포함한다. 그다음 5장에서는 결과물의 예 시를 보이고, 마지막으로 6장에서 본 논문에 대한 결론을 맺고 향후 연구 방향을 제시한다.

제 2 장 예비 지식

2.1 스윕 곡면

혈관은 분기된 튜브 형태의 스윕 곡면(sweep surface)으로 구성된다. 3차원 물체를 3차원 공간상에 주어진 경로에 따라 이동시키면 스윕 볼륨을 얻게 되는데, 이 경계를 정의하는 것을 스윕 곡면이라고 한다. 스윕 곡면을 정의하기 위해서는 경로와 그 경로를 따라 움직이는 물체가 필요하다. 본 논문에서는 3차원 공간상에 서 이동할 경로를 실린더의 중심축으로 설정하였으며 해당 축은 사용자가 정의할 수 있다. 또한, 이 중심축을 중심으로 생성되는 실린더를 주어진 경로를 따라 이 동시켜 이를 통해 얻는 스윕 볼륨으로 혈관의 분기 구조 형상을 모델링하였다.



그림 2.1: 스윕 볼륨으로 만든 매듭 모양[26]



그림 2.2: 물체를 정의한 후 이를 이 동시켜 얻는 스윕 볼륨[32]

분기 구조 사이를 블렌딩하는 부분 또한 특정 크기의 반지름 δ를 가지는 구가 두 실린더의 교차 곡선을 따라 지나갈 때 생성되는 스윕 볼륨을 사용하여 생성 하였다[26]. 본 연구에서 사용하는 스윕 볼륨은 대부분 그림 2.3과 같은 형태를 지니고 있다.



그림 2.3: 롤링 볼 스윕 볼륨[27]

2.2 실린더

일반적으로 실린더는 두 밑면이 합동 관계의 원(circle)이고, 한 밑면의 원주 에서 다른 밑면까지의 가장 가까운 점까지의 선분을 포함한 3차원 입체도형을 일컫는다. 이때, 해당 선분들의 집합은 원기둥의 옆면을 생성한다. 공간 좌표상 에서 원기둥은 두 밑면은 표현하지 못하고, 높이가 무한한 옆면만을 표현할 수 있다. 실린더는 고정된 축과 항상 평행인 직선의 회전으로 생기는 입체 도형의 일종이므로, 2차원 좌표 위의 원이 축을 따라 이동하는 형태이며 무수히 많은 원 위의 점들로 이루어진 직선(line)들이 축에 평행하게 존재하는 것이라고 이해할



그림 2.4: 실린더

수 있다[30]. 축을 *z*축으로 둔 데카르트 좌표계(Cartesian coordinate system) 에서, *r*의 반지름을 가지는 원기둥의 방정식은 다음과 같다.

$$x^2 + y^2 = r^2 \tag{2.1}$$

이러한 방정식은 매개변수 방정식(parametric equation)으로도 표현 가능하다.

$$x = r \cos \theta$$
$$y = r \sin \theta$$
$$z = \varphi$$

이 때, $\theta \in [0, 2\pi], \varphi \in \mathbb{R}$ 과 같은 조건을 만족한다.

제 3 장 관련 이전 연구

최근 Computer-Aided Design(CAD)외에도 조직 공학, 항공 설계 공학, 재료 공학 등 다양한 학문 및 산업 분야에서 다공성의(porous) 반복적인 미세구조를 통해 3차원 부피를 가지는 물체를 제조하고 설계하는 연구에 대해 급격한 관심을 가지고 있다. 또한, 적층 제조의 발전[8]이 이루어짐에 따라 의학이나 생물학[2, 21] 등 다양한 분야에서 실제로 활용되고 있다. 적층 제조란 레이어를 쌓아가면서 물 체를 만들어가는 기법을 말하는데, 주어진 목적에 따라 미세 구조에 사용되는 재료, 물체에 배치하는 방법, 사용하는 프린팅 기술 등이 다양하게 연구되고 있 다[15, 19, 28, 31].

Burczyński et al.[4]은 미세 구조를 대량으로 최적화하는 방법에 대해 제안하 였다. 물체의 모양, 변형력, 팽창력 등 여러 특성을 정의하는 여러 변수 집합들이 마이크로 레벨에서 정의되고 유한요소법 (Finite element method, FEM)이 분석 단계에서 사용되었다. 또한, 유전 알고리즘에서 사용되는 진화 스킴(evolutionary scheme)을 최적의 변수 값들을 찾는 데에 활용하였다. 이 외에도, 미세 구조를 대량으로 마이크로 및 나노 레벨에서 모델링하는 기법들이 연구되었다[29, 36].

Antolin et al.[1]은 다공성의 미세 구조를 기하학적 특성을 고려하여 더 정확 하게 디자인하는 방법을 제안하였다. 기존과 같이 트러스(truss-like) 구조를 사용 하는 것이 아닌, 삼변량 볼륨을 가진 V-model(trivariate volumetric model)[18] 을 사용하여 미세구조를 채워넣었다. V-model의 사용을 통해 미세구조를 채울 때 연결되는 부분의 곡면의 개수가 서로 다른 경우에 대해 대응할 수 있다. Antolin et al.은 특히 열 교환기(heat exchanger)와 같은 여러 물리적 응용 분야의 효율 을 극대화할 수 있는 복잡한 미세구조 기하학에 대해 논하였다. Hong et al.[12] 은 잘라진(trimmed) V-model에 미세구조를 넣는 방법을 제안하였다. 본 연구 에서는 Antolin et al.의 연구를 기반으로 실린더와 곡면 안을 매개변수화하여 미세구조로 채워 넣고, Hong et al.의 방법을 통해 서로 교차하는 공간을 제거한 후 스토캐스틱(stochastic)하게 연결한다.

곡면을 블렌딩하는 방법은 Rossignac et al.[26], Vida et al.[33], Woodwark et al.[35]의 연구를 기반으로 확립되었다. 또한, 2차 곡면을 블렌딩하는 방법 및[6, 9, 22] 곡률 연속적인 일반적인 알고리즘들[11, 24, 25]이 연구되었다. 본 논문에 서는, 이러한 연구들을 기반으로 정리된 롤링-볼 블렌딩 알고리즘을 사용하여 실린더가 교차하는 부분을 매끄럽게 처리한다.

BVH(Bounding Volume Hierarchy) 자료 구조는 기하학 알고리즘에 사용되 기 위해 여러 제약 및 조건들이 정의되었다[3, 13]. BVH에서 부모 노드가 가지고 있는 물체의 부분은 해당 노드의 자식 노드들에 의해 전부 덮어져야 한다. 또한, 모든 노드들은 부모가 가진 물체보다 더 타이트(tight)한 핏(fit)을 가져야 한다. 마지막으로, 계층 구조는 전부 사용자와의 상호 작용없이 자동적으로 만들어져야 하며, 구조 안의 바운딩 볼륨은 최대한 기존 물체에 꼭 맞게 구현되어야 정확성을 높일 수 있다.

마지막으로, 본 논문에서 BVH에 사용하는 바운딩 볼륨인 구(sphere)에 관한 연구는 Bradshaw et al.[3], Hubbard et al.[13], Quinlan et al.[23]에 의해 제안 되었다. 연구된 BVH 자료 구조를 통해, 본 알고리즘은 실린더의 조각들의 교차 및 충돌을 더 빠르게 검출할 수 있으며 이들을 더 정확하게 렌더링할 수 있다.

11

제 4 장 분기 구조 모델링 알고리즘

4.1 분기 구조 블렌딩

4.1.1 실린더 간의 교차 곡선

본 알고리즘은 두 개의 실린더를 사용하여 분기 구조를 나타낸다. 분기 구조 는 그림 4.1과 같이, 두 개의 실린더 중 하나는 축이 *x*축인 메인 실린더이고 이를 기준으로 다른 하나의 서브 실린더가 *x*축을 중심으로 회전하는 형태를 지닌다. 메인 실린더의 반지름의 크기가 *R*일 때 실린더의 방정식은 다음과 같다.

$$x^2 + y^2 = R^2 \tag{4.1}$$

또한, 마찬가지로 서브 실린더의 식도 이와 같이 표현할 수 있다. 두 실린더가 *x*축 을 기준으로 수직으로 만난다고 가정했을 때 반지름의 크기가 *r*인 서브 실린더의 방정식은 다음과 같다.

$$x^2 + z^2 = r^2 \tag{4.2}$$

실린더 두 개가 교차할 때, 위에서 구한 두 개의 식의 연립방정식을 통해서 교차 곡선을 구할 수 있다[10]. 해당 식은 두 실린더 사이의 공통된 값을 찾는 식이므로 메인 실린더의 원 파라미터 θ가 주어졌을 때 교차 곡선 위의 임의의 점(*x*, *y*, *z*)은 다음 조건을 만족해야 한다.

$$x = R \cos \theta$$

$$y = R \sin \theta$$

$$z = t$$
(4.3)



그림 4.1: 90도로 만나는 두 개의 실린더

수식 4.1을 수식 4.2에 대입하면 다음과 같은 식이 도출된다.

$$z^{2} = r^{2} - x^{2} = r^{2} - (R^{2}\cos^{2}\theta)$$
(4.4)

수식 4.4의 해가 1개일 때는 z = 0이므로 다음과 같다.

$$r^{2} - (R^{2}cos^{2}\theta) = 0$$

$$\Leftrightarrow \cos\theta = \frac{r}{R}$$

$$\therefore \theta = \arccos(\frac{r}{R})$$
(4.5)

수식 4.4의 해가 2개일 때는 $z = \pm \sqrt{r^2 - R^2 \cos^2 \theta}$ 인 경우인데, 4.3의 값들을 $z = s, \cos \theta = t$ 로 각각 대입하여 정리하면 다음과 같다.

$$t = \pm \sqrt{r^2 - R^2 s^2} \tag{4.6}$$



그림 4.2: 교차 곡선 방정식

수식의 s, t 값들은 그림 4.2처럼 타원의 방정식의 형태를 띄며, $-\theta_0 \le s \le \theta_0$ 일 때 다음과 같은 방정식으로 나타낼 수 있다.

$$\frac{s^2}{\left(\frac{r}{R}\right)^2} + \frac{t^2}{r^2} = 1$$

$$\left(-\left(\frac{r}{R}\right) \le s \le \left(\frac{r}{R}\right)\right)$$

$$(4.7)$$

이와 같은 분기 구조를 지닌 실린더 두 개가 교차할 때, 본 알고리즘은 해당 교차하는 부분을 부드럽게 만들기 위해 임의의 디자인 파라미터 δ 크기의 반지름 을 가진 구를 굴려 블렌딩하는 롤링 볼 알고리즘[11, 26]을 제시한다. 그림 4.1의 형태를 확장시켜 각각의 실린더를 감싸고 있는 (*radius* + δ) 크기의 실린더를 가정하여 교차 곡선을 구한다. 이를 통해, 그림 4.3에서 보여지는 것과 같이 구가 두 실린더 사이의 교차하는 부분을 굴러가는 궤적을 구할 수 있다. 교차 곡선의 식은 다음과 같다.

$$\frac{s^2}{(\frac{r+\delta}{R+\delta})^2} + \frac{t^2}{(r+\delta)^2} = 1$$
(4.8)



그림 4.3: (δ+반지름)의 크기를 가지는 실린더와 교차곡선

4.1.2 커널 곡면

교차 곡선의 점(*s*,*t*)들을 본래의 3차원 좌표 (*x*,*y*,*z*)로 샘플링한 후, 이 점들 을 중심으로 가지는 구와 각 실린더가 맞닿는 점 *P*₁, *P*₂를 구한다. 그림 4.4과 같이, 두 실린더와 맞닿는 점을 각각 *P*₁, *P*₂, 원의 중심을 *C*, 중심으로부터의 벡터를 **v**₁, **v**₂로 표현하였고, 빨간색으로 표시된 곡선은 *P*1, *P*2 사이의 원호를 의미한다. 본 알고리즘은 구가 굴러가는 궤적에서 맞닿는 점들과의 원호를 전부 연결하여 커널 곡면을 구하기 때문에 이를 수식으로 나타내는 과정이 필요하다. 점 *P*₁과 *P*₂사이의 각도를 θ라고 했을 때, 이 사이의 원호에 여러 점을 샘플링하여 이를 나타낼 수가 있다.



그림 4.4: 서로 맞닿고 있는 구와 실린더의 표현



그림 4.5: 직교하는 벡터들로 새로 만들어낸 직교좌표계

점 P_1 과 θ_i 만큼 떨어진 점 P_i 를 구해야 하므로 우선 $\mathbf{v_1}$, $\mathbf{v_2}$ 이 포함된 평면에 수직인, 즉 두 벡터와 동시에 수직인 벡터 $\mathbf{v_3}$ 을 구한다. 또한, $\mathbf{v_1}$, $\mathbf{v_3}$ 에 동시에 수직인 벡터 $\mathbf{v_2}'$ 를 구한다. 이를 통해, $\mathbf{v_3} = \mathbf{v_1} \times \mathbf{v_2}$, $\mathbf{v_2}' = \mathbf{v_1} \times \mathbf{v_3}$ 이므로 $\mathbf{v_1}$, $\mathbf{v_2}'$, $\mathbf{v_3}$ 은 그림 4.5과 같은 새로운 3차원 직교좌표계를 이루게 된다. 이 때, 새로운 직교좌표계에 대한 회전행렬 R을 다음과 같이 정의한다.

$$R = \begin{bmatrix} \mathbf{v_1} & \mathbf{v_2}' & \mathbf{v_3} \end{bmatrix}$$
(4.9)

기존 좌표계에 속해있던 점 X는 이 좌표계에서 다음과 같이 표현할 수 있다.

$$Y = R^{-1}(X - C) (4.10)$$

반대로, 새 좌표계의 점 Y는 원래 좌표계에서 다음과 같이 표현할 수 있다.

$$X = RY + C \tag{4.11}$$

새 좌표계에서 기존의 축 (1,0,0)을 *θ*_i만큼 회전시킨 후, 이 점을 수식 4.10, 수식 4.11을 사용하여 원래 좌표계로 3차원 변환시켜 샘플링하면 그림 4.4의 원호의 점들을 구할 수 있다.



그림 4.6: 메인 실린더와 서브 실린더의 커널 스페이스. 메시



그림 4.7: 실린더와 커널 곡면의 평면도



그림 4.8: 구가 굴러가는 궤적으로 생기는 커널 곡면과 샘플링된 점들

메인 실린더와 서브 실린더의 외부(outer) 실린더와 내부(inner) 실린더마다 δ 크기만큼 반지름을 더하여 서로 교차하는 곡선을 구하고, 이를 통해 얻어내는 구와 맞닿는 원호의 점들을 쿼드 메시로 그리드로 연결하면 그림 4.6와 같은 커널 곡면을 만들어낼 수 있다. 그림에서 안쪽의 커널 곡면은 메인 실린더와 서브 실린 더의 내부 실린더에 블렌딩을 하여 만든 것이며, 오른쪽의 곡면은 외부 실린더에 블렌딩을 하여 만든 것이다.



그림 4.9: 90도로 교차하는 두 실린더 사이의 커널 곡면

4.1.3 여러 각도에서 교차하는 실린더의 분기 구조



그림 4.10: ↓의 각도로 만나는 두 실린더

다음은 두 실린더가 수직으로 만나는 경우 외에 x축을 기준으로 모든 각도로 만나는 경우에도 수식을 적용하는 방법에 대해 논하도록 하겠다. 메인 실린더와 x축을 기준으로 각도 ϕ 로 교차한다고 했을 때 서브 실린더의 식은 수식 4.2와 달리 아래와 같이 구하여 설정한다. 우선, z축인 $\hat{\mathbf{z}} = (0,0,1)$ 을 x축에 대해서 ϕ 만큼 회전하여 다음과 같이 서브 실린더의 축을 구한다.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -\sin \phi \\ \cos \phi \end{bmatrix}$$
(4.12)



그림 4.11: 3차원에서 점과 직선과의 거리

그림 4.11과 같이, 실린더 축의 임의의 점 x_1 , x_2 와 실린더 위의 점 x_0 에 3 차원에서의 점과 직선과의 관계를 적용하여 반지름의 값을 구할 수 있다. 그림의 (a)는 다음과 같은 수식으로 정리된다.

$$r = \left|\frac{(x_2 - x_1) \times (x_1 - x_0)}{x_2 - x_1}\right| \tag{4.13}$$

그림 (b)와 같은 실린더 위의 점 $x_0 = (x, y, z)$, 실린더 축 위의 두 점을 $x_1 = (0, -\sin\phi, \cos\phi), x_2 = (0, -2\sin\phi, 2\cos\phi)$ 라고 가정했을 때, 수식 4.13에 이를 대입하여 구해지는 실린더의 방정식은 다음과 같다.

$$r^{2} = \frac{|(0, -\sin\phi, \cos\phi) \times (-x, (-\sin\phi - y), (\cos\phi - z))|^{2}}{\cos\phi^{2} + \sin\phi^{2}}$$

= $(\cos\phi y + \sin\phi z)^{2} + (-\cos\phi x)^{2} + (-\sin\phi x)^{2}$ (4.14)
= $x^{2} + (\cos\phi y + \sin\phi z)^{2}$

구해진 실린더의 방정식에 수식 4.3의 변수 값들을 대입하여 교차식을 구한 다면, 다음과 같다.

$$x^{2} + (\cos \phi y + \sin \phi z)^{2} = r^{2}$$

$$R^{2} \cos^{2} \theta + (R \cos \phi \sin \phi) + \sin \phi t)^{2} - r^{2} = 0$$

$$R^{2} \cos^{2} \theta + R^{2} \cos^{2} \phi \sin^{2} \theta + 2R \cos \phi \sin \phi t \sin \theta + \sin^{2} \phi t^{2} - r^{2} = 0$$

$$let \sin \theta = s,$$

$$R^{2}(1 - s^{2}) + R^{2} \cos^{2} \phi s^{2} + 2R \cos \phi \sin \phi t s + \sin^{2} \phi t^{2} - r^{2} = 0$$

$$(R^{2} \cos^{2} \phi - R^{2})s^{2} + (2R \cos \phi \sin \phi t)s + (R^{2} + \sin^{2} \phi t^{2} - r^{2}) = 0$$

$$\therefore s = \frac{-R \cos \phi \sin \phi t \pm \sqrt{(R \cos \phi \sin \phi t)^{2} - (R^{2} \cos \phi^{2} - R^{2})(R^{2} + \sin \phi^{2} t^{2} - r^{2})}{R^{2} \cos \phi^{2} - R^{2}}$$

$$(4.16)$$



그림 4.12: *t*값의 샘플링 범위

이차방정식을 풀어 $s(\sin \theta)$ 값을 구한 후, (s,t)좌표를 (x, y, z)로 변환하여 샘 플링하고 교차 곡선을 구한다. 두 실린더가 서로 수직이 아닌 ϕ 의 각도로 만나기 때문에 서브 실린더가 기울어져 t의 범위가 바뀌게 된다. 그림 4.12과 같이, t의 범위를 계산할 수 있다. t의 길이 L과 중점 T는 삼각함수의 법칙으로 구할 수 있으며 t의 범위는 T를 기준으로 $t \in [T - \frac{L}{2}, T + \frac{L}{2}]$ 를 가지게 된다.

$$L = \frac{2r}{\sin\phi}$$

$$\frac{T}{R} = \tan(\frac{\pi}{2} - \phi) = \cot(\phi) \qquad (4.17)$$

$$\therefore R \cot(\phi) - \frac{r}{\sin\phi} \le t \le R \cot(\phi) + \frac{r}{\sin\phi}$$

4.1.4 시행착오와 고찰



(a) Distortion

(b) Infinite problem

그림 4.13: 오류의 예시

본 연구에서 블렌딩 곡면을 구현할 때 겪은 여러 시행착오(trial and error)를 통해 깨닫게 된 점에 대해 설명하고자 한다. 그림 4.13의 (a)는 90도가 아닌 다른 각도 *φ*로 만나는 실린더 사이의 공간보다 롤링 볼의 크기가 더 클 경우 생기는 일그러짐(distortion) 현상을 나타낸다. 본 연구에서는 이를 방지하기 위해, 디자 인 변수(design parameter)인 구의 반지름을 충분히 작게 설정하여 구현하였다. 각도가 더 가파르게 올라갈 수록 메인 실린더와 서브 실린더 사이의 공간이 더 좁아지고, 구의 반지름이 작아질수록 커널 곡면의 크기 또한 줄어들기 때문에 이를 고려하여 구의 반지름의 길이를 적절하게 조절해주어야 한다.

그림 4.13의 (b)는 2.2절에서 설명한 것과 같이, 실린더는 2차원의 원이 무 한한 형태인 도형이기 때문에 이로 인해 생기는 렌더링 오류의 결과이다. 90°로 서로 만나는 분기 구조에서 메인 실린더가 무한한 서브 실린더와 교차하게 되면 서브 실린더가 있는 곳 외에 반대편의 다른 위치에도 점들이 샘플링되기 때문 이다. 이 때, 양쪽 위치의 점들 모두에 대해 본 알고리즘에서 커널 곡면 메시를 구현하기 위해 사용하는 알고리즘을 수행하였더니 저런 가시가 돋힌 형태로 잘못 렌더링되었다. 이를 방지하기 위해 본 알고리즘은 해당 서브 실린더가 있지 않는 곳은 점을 샘플링하지 않도록 제한하였다.

4.2 BVH 생성 알고리즘

4.2.1 충돌 검사를 위한 BVH

두 실린더 객체가 서로 교차할 때, 그림 4.14처럼 서로 교차하여 겹치는 부분이 생기게 된다. 이러한 부분을 제대로 처리해주지 않는다면 미세구조를 집어넣거 나 모델링할 때 같은 위치에 중복된 조각이 생기거나 서로 충돌하여 잘못된 형상 변이가 나타나게 될 수 있다. 만약 모든 실린더의 조각을 각각 일일이 비교하여 찾아낸다면 매우 오랜 시간이 걸릴 것이다. 이러한 처리를 위해서, 실린더의 어떤 부분이 충돌했는지를 판단할 필요가 있다. 이번 챕터에서는 본 연구에서 두 개의 실린더 간의 충돌 여부를 바운딩 볼륨 계층구조(Bounding Volume Hierarchy, BVH)[3, 13]를 통해 알아내고 이를 처리한 방법에 대해 설명하도록 하겠다.



그림 4.14: 두 개의 실린더가 교차할 때 생기는 부분

메인 실린더와 서브 실린더는 충돌 검사 및 미세구조 모델링을 용이하게 하기 위해 [*u*, *v*, *w*] 파라미터로 나타낸 각각의 그리드(grid)들로 이루어져 있다. 그림 4.15와 같이, *u*, *v*, *w*는 각각 순서대로 실린더의 두께, *θ* 크기, 높이를 나타낸다. 본 알고리즘에서는 이러한 각각의 그리드마다 바운딩 스피어(Bounding Sphere) 로 감싸고 이를 바운딩 볼륨(Bounding Volume, BV)으로 구성하여 서로 충돌 하는지에 대한 여부를 판단한다. 기하학적 물체에 대한 연산을 가속화하기 위한 자료구조 중 하나인 BVH는 복잡한 형태의 구조를 간단한 도형의 집합들로 감싸 계층적인 트리 구조를 만드는 것이다. 구(sphere)를 통해 BV를 만들면 생성 시간 이 빠르고 물체 간의 충돌 여부를 쉽게 계산할 수 있다는 이점이 있다[3, 13, 23]. 실 린더 하나마다, Bottom-up 방식으로 전체 실린더부터 각 그리드 조각까지 BVH 를 구성하게 되며 그림 4.16처럼 모든 그리드는 이를 완전하게 감싸고 있는 구를 각각 가진다.


그림 4.15: [u, v, w] 파라미터로 나타낸 각각의 그리드



그림 4.16: 각 그리드를 감싸는 바운딩 스피어

각 그리드 노드들에 대해 BVH를 구성하기 위해서 우선 top-down 형식으로 모든 노드들에 대한 바운딩 스피어를 정의한다. 알고리즘은 다음과 같다.

Algorithm 1: BVH construction
Data: G =Grid node, P =Parameter of $[u, v, w]$
Result: BVH structure for each grid nodes
1 Approximate $G(P)$ with bounding sphere
2 if $G(P)$ is leaf then
3 Set leaf flag and insert $G(P)$ to tree
4 end
5 else
6 foreach Each parameter $p \in P$ do
7 Subdivide p to p_l and p_r /* divide into half */
8 Call this routine recursively for $G(p_l)$ and $G(p_l)$
9 end
10 Union bounding spheres of left and right Child
11 end

실린더를 구성하는 모든 그리드 노드들부터 시작하여 리프 노드(leaf node)에 도달할 때까지 반으로 나누어져(subdivide) 재귀적으로 내려가고, 전체적으로는 이진 트리의 형태를 갖추게 된다. 알고리즘의 10번째 줄과 같이, 노드들이 쪼개지 는 과정에서 각 리프 노드의 바운딩 스피어를 합쳐(union) 새로운 노드의 바운딩 스피어를 각각 저장하게 된다. 본 논문에서는 이런식으로 생성된 각 실린더의 BVH을 통해 바운딩 볼륨 테스트 트리(Bounding Volume Test Tree, BVTT)를 구성하여 충돌을 검사하였다.



그림 4.17: BVH와 BVTT 구조

각 실린더의 BVH의 루트(root) 노드부터 시작하여 해당 노드를 감싸고 있는 바운딩 스피어가 충돌할 경우 자식 노드끼리 충돌 검사를 한다. 그 후, 이 과정을 리프 노드에 도달할 때까지 재귀적으로 반복하여 충돌된 각각의 그리드 노드를 검사한다.

Algorithm 2: Collision Detection between two BVHs
Data: N_1 : node of BVH_1 , N_2 : node of BVH_2
1 if $isSphereCollision(N_1,N_2)$ then
2 if N_1 and N_2 are both leaf nodes then
3 $Collision[N_1] \leftarrow True$
4 $Collision[N_2] \leftarrow True$
5 end
6 else
7 Call this routine recursively for left, right child of N_1, N_2
8 end
9 end

다음 알고리즘에서 사용되는 구의 충돌 검사의 경우(1번째 줄), 그림 4.18과 같이 두 구의 각 중심간의 거리 *d*를 구한 후 각 반지름 R과 r을 더한 값과 비교하여 수행한다.

if
$$d > (R+r)$$
, $collision = False$
if $d < (R+r)$, $collision = True$ (4.18)



그림 4.18: 구의 충돌 검사



그림 4.19: 그리드 충돌 검사의 실제 예시이다. 충돌하는 경우 빨간색, 충돌하지 않는 경우 초록색이다.

4.2.2 잉여 조각을 검출하기 위한 BVH



그림 4.20: 노란색으로 표시된 부분은 충돌이 검사되지 않았지만 사라져야 하는 부분이다.

이와 같은 BVH 알고리즘은 그림 4.14과 같이 서로 충돌하는 그리드 조각들을 확인하는 용도 외에, 렌더링할 필요가 없는 잉여 그리드 조각들을 알아내는 데에 도 유용하게 사용되었다. 그림 4.20과 같이 두 실린더가 교차할 때 서브 실린더가 메인 실린더 안으로 깊이 파고드는 경우, 충돌 검사를 통해서는 검출되지 않지만 제거해야 하는 필요 없는 조각들이 생성된다. 이 조각들을 제대로 처리해주지 않는다면 그림에서 노란색으로 칠해진 부분과 같은 잉여 조각들이 가운데에 떠 있게 되어 올바르지 않은 렌더링 결과를 도출해낸다. 이 부분을 알아내기 위해 다음과 같은 간단한 조건식을 사용하였다.

if
$$collision = True and d \le R, included = True$$

if $collision = True and d > R, included = False$ (4.19)



그림 4.21: 메인 실린더 바운딩 스피어 C₁의 반지름 *R*보다 그리드 조각의 바운딩 스피어 C₂와의 거리 *d*가 더 작을 경우 충돌한다.

만약 충돌 검사에서 통과한 서브 실린더 그리드 조각의 바운딩 스피어가 그림 4.21과 같이 메인 실린더 바운딩 스피어 안에 포함될 경우 이 안에 포함되었다고 가정한다. 추후에 실린더의 분기구조를 모델링하기 위해 두 실린더 그리드 조각 들을 불리안 인터섹션(Boolean Intersection)할 때, 포함되어있는 그리드 조각에 대해서만 수행하면 되기 때문에 시간을 훨씬 단축시킬 수 있다.

4.3 미세구조 모델링

본 논문에서 소개하는 알고리즘을 전체적으로 요약하자면, 4.1절에서 설명한 것과 같이 교차하는 실린더를 블렌딩시키고, 4.2절의 방법으로 서로 충돌하는 실린더 그리드 조각들을 알아낸 다음, 마지막으로 생성된 모형 안에 미세구조를 채워넣어 모델링하는 것이다. 이번 절에서는 이 중 마지막 단계에 해당하는 미세 구조 모델링 부분에 대해 설명하도록 하겠다. 이를 위해서, 먼저 파라미터 공간을 정의한 후 이 안을 채우는 미세 구조를 생성하여야 한다.

4.3.1 타일 정의

본 알고리즘에서는 [0,1]³을 도메인으로 가지는 단위정육면체(unit cube)안에 특정한 타일(tile)을 넣는 방식[1]을 사용한다. 일반적으로, 타일에는 다양한 종류 가 있는데 보통 3D 프린팅 분야에서는 가볍고 유연한 특성때문에 격자(lattice) 구조 형태의 타일을 많이 채워넣는다[16, 28]. 매우 다양한 형태의 타일 및 미세 구조 형상들이 존재하지만 본 알고리즘에서는 그림 4.22과 같은 구현하기 쉽고 속도가 빠른 크로스-큐브(cross-cube)[20] 모양의 타일을 사용하였다.



그림 4.22: 사용되는 타일의 기본 구조

본 알고리즘은 단위정육면체의 중심으로부터 각 면의 중심까지 뻗어나가는 실린더를 각각 생성하고, 이를 합친 형태를 타일로 사용하였다. 그림 4.22에서 노란색으로 표시된 것과 같이 하나의 큰 실린더는 여러 개의 작은 실린더로 나 누어져있다. 해당 예시에서는 위를 향하고 있는 실린더와 아래를 향하고 있는 실린더가 각각 4개의 더 작은 실린더로 구성되어 있음을 확인할 수 있다. 타일 들을 후에 다른 유클리디안(Euclidean) 공간 안에 매개변수화할 때, 더 자유롭고 유연하게 형상이 변형될 수 있게 하기 위해 큰 구조를 더 작은 구조들로 나누어 타일을 구성하였다.



그림 4.23: (a)각각 [4,2,4]개의 그리드로 나뉜 실린더. (b) 타일을 채워넣은 것.

이전 절(4.2)에서 설명한 것과 같이, 미세구조를 채워넣어야 하는 메인 실 린더와 서브 실린더는 그림 4.15과 같은 변수들로 매개변수화된 그리드 조각들 로 이루어져있다. 그리드의 각 [*u*,*v*,*w*] 파라미터는 특정 범위를 가지고 있는데, 숫자 1을 그리드의 각 개수에 따라 나눈 값을 오프셋(offset)으로 두어 각 그 리드를 구분할 수 있다. 그림 4.23의 예시와 같이, 모든 그리드들은 각각의 매 개변수의 범위로 나타낼 수 있다. 그림에서 파란색으로 표시된 그리드의 경우 [*u*,*v*,*w*] = [*u* ∈ [0.75, 1.0], *v* ∈ [0.5, 1.0], *w* ∈ [0.0, 0.25]]와 같은 매개변수를 가지고 있다.

생성된 기본 타일은 [0,1]³ 단위정육면체를 기반으로 만들어져있으므로, 즉 매개 변수는 다음과 같이 정의된다.

$$u \in [0, 1], v \in [0, 1], w \in [0, 1]$$

4.3.2 그리드의 타일 매개변수화

이제 만들어진 타일을 이 각각의 그리드에 어떤 식으로 적용하였는지에 대 해 설명하도록 하겠다. 본 알고리즘에서는 정의된 매개변수를 그리드의 [u,v,w] 에 맞게 변환시킨 후[1], 기존의 타일들의 메시(mesh)를 옮겨서 다시 렌더링하는 방식을 사용하였다.

Algorithm 3: Tile Parameterization from Cube to Grid
Data: S : grid's space parameter, R : range of parameter,
cyH : cylinder's height, $radL$: radius's length between outer
circle and inner circle, $density:$ each numbers of $\left[u,v,w\right]$ of
cylinder
$1 \ grid_u \leftarrow [\mathrm{radL} \ ^* \mathbf{R}(\mathbf{S}_v)]$
2 $grid_v \leftarrow [2\pi * R(S_u)]$
$3 \ grid_w \leftarrow [\text{cyH* } \mathbf{R}(\mathbf{S}_w)]$
4 for $i,j,k \in density$ do
5 $[u_i, v_i, w_i] \leftarrow [grid_{ui}, grid_{vi}, grid_{wi}]$ for each grid.
6 end

7 Move mesh data to new parameter $[grid_{ui}, grid_{vi}, grid_{wi}]$

알고리즘 3의 1-3줄과 같이, 그림 4.24의 (a)처럼 순서대로 실린더 그리드의 반지름의 길이, θ, 높이에 따라 매개변수의 값을 저장한다. 이러한 값들을 삼중 for 문(4-6)을 통해 각 그리드마다 매개변수를 바꿔서 저장한다. 컴퓨터 그래픽스에 서 어떠한 물체를 변형하여 렌더링할 때, 해당 물체의 indices 정보만 알고 있다면 해당 vertices들이 다른 값들로 변형되어도 그대로 연결 정보를 유지하면서 다시 그려낼 수 있다. 이처럼, 단위정육면체의 기본 타일의 모든 indices의 정보를 미리



그림 4.24: (a) 그리드의 매개변수 (b) 그리드 타일

저장하고 vertices 값들을 알고리즘 3에서 정의한 것과 같이 변환(transform)하여 렌더링하면 그림 4.24의 (b)와 같은 형태의 메시를 그려낼 수 있다. 이론적으로, 이와 같은 방법을 사용하면 [u, v, w]의 파라미터로 나타낼 수 있는 모든 물체을 단위정육면체를 기반으로 제작된 타일로 채워넣는 것이 가능하다.

4.3.3 커널 곡면의 타일 매개변수화

실린더 그리드 조각들 외에 교차하는 두 실린더 사이에 블렌딩 기법으로 생 성한 커널 곡면(그림 4.9)안을 미세구조로 채우는 방법에 대해 논하도록 하겠다. 우선 기본적으로 생성된 커널 곡면은 두 실린더와 맞닿은 구의 원호를 따라 만든 메시이기 때문에 안의 공간은 나누어지지 않은 하나의 공간이다. 이 안을 다른 실린더 그리드 조각들과 같이 타일들로 채워넣기 위해서는, 앞서 설명한 것과 마찬가지로 매개변수 [*u*,*v*,*w*]로 나타낼 수 있는 공간들로 정의하여야 한다. 이를 위해, 본 논문에서는 커널 곡면을 여러 개의 상자(box)들로 나누어 표현하였다. 커널 곡면은 외부 커널 곡면과 내부 커널 곡면 두 가지가 있으므로(그림 4.6), 외부 커널 곡면 위의 원호 A_i 위에 여러 개의 점을 샘플링하고 이 점들 중 순서대 로 두 점(A_i , i, A_i , i+1)을 바로 옆의 외부 원호 A_{i+1} 위의 점들 $A_{i+1,i}$, $A_{i+1,i+1}$ 과 이은 후, 이들을 내부 커널 곡면의 같은 점들과 연결하여 총 8개의 정점들을 연결하여 상자 메시를 생성한다. 이러한 작업을 모든 샘플링된 점들에 수행하면 곡면을 여러 개의 상자들로 만들어낼 수 있다. 그림 4.27과 같이, 생성된 상자를 또 여러 개로 나누어 원하는 크기의 공간을 만들어낼 수 있다. 해당 예시의 경우 n = 2로 나누어 총 두 개의 상자 공간을 갖도록 하였다.



(a) Canal Space (Mesh)



그림 4.25: (a) 비어있는 커널 곡들 (b) 상자들로 나뉜 커널 곡면



그림 4.26: 커널 곡면 안의 상자 메시 생성



그림 4.27: 커널 상자 섭디비전(Subdivision)

만들어진 상자들은 정육면체가 아닌 기울어진(tilted) 형태를 하고 있다. 따라 서, 단위정육면체를 기반으로 만들어진 타일을 새로운 상자들에 맞게 변형시키기 위해서는, 상자의 중심과 실린더의 회전축을 수정해줄 필요가 있다. 본 알고리즘 에서는 *f_i*를 새로운 상자의 각 면의 중심, *C_i*를 상자의 중심이라고 할 때, 다음과 같이 이를 정의하였다.

$$C_i = \frac{sum(x, y, z)}{8} \tag{4.20}$$

$$\widehat{\mathbf{v}}_i = C_i - f_i \tag{4.21}$$

정의된 $\hat{\mathbf{v}}_i$ 는 각 실린더의 회전축 방향이 된다. 이에 맞게 회전 변환을 해주면 그 림 4.28과 같은 변형된 형태의 타일이 생성된다. 그리드에 사용된 타일과는 달리, 단위정육면체를 기반으로 만든 기존의 타일과 비슷하게 생겼지만 조금 비틀어진 형태를 하고 있음을 확인할 수 있다.



그림 4.28: 새로 생긴 상자에 맞게 변형된 타일

그림 4.29은 본 논문에서 앞서 설명한 커널 곡면 안을 상자로 나누고 기존의 타일을 매개변수화하여 넣는 과정을 요약한 것이다. 다시 정리하자면, 본 연구에 서는 구를 굴려 맞닿은 부분의 원호로 메시를 만든 후(a), 상자로 나누어 공간을 매개변수화하고(b), 이 안에 주어진 타일을 변형시켜 채워넣는다(c). 본 연구는 이러한 일련의 과정을 공이 굴러가면서 생성되는 모든 원호에 대해 수행한다.



(c) Tiles in Boxes



그림 4.29: 타일이 커널 곡면 안에 채워지는 과정.



(a) Empty Canal Space

(b) Filled Canal Space

그림 4.30: 미세구조로 채워진 커널 곡면 상자들



그림 4.31: 미세구조로 채워진 커널 곡면

4.3.4 교차하는 그리드 미세구조 모델링

마지막으로, 서로 교차하여 충돌이 검출된 그리드 조각들을 처리하고 그 안을 미세구조로 채우는 방법에 대해 논하도록 하겠다. 본 연구에서는 Hong et al.[12] 에 의해 제안된 방법을 적용하여 미세 구조를 연결하였다.



그림 4.32: 타일의 정의[12]

Hong et al.[12]의 논문에서는 그림 4.32과 같이, 마크로(macro) V모델(Volumetric Model) 안에 채워져있는 타일을 내부 타일(inside tile), 교차 타일 (intersecting tile), 내부 타일 중 교차 타일과 가까운 것을 브리지 타일(to-be-bridged) 이라고 정의하였다. 그림의 (a)에서 내부 타일은 초록색으로, 외부 타일은 노란색 으로 칠해져있다. 두 V모델을 연결(union)시키기 위해서는 이 안을 채우고 있는 타일들도 연결시켜야 하는데, 이를 위해서는 먼저 교차 타일을 먼저 다 제거해야 한다. 그 후, 타일을 제거한 곳에 빈 공간이 생기기 때문에 이 사이를 적절하게 연 결하는 작업이 필요하다. Hong et al.[12]은 그림 (b)의 노란색 브리지 타일의 각 끝 면을 앵커링 면(anchoring face)라 정의한 후, 이 앵커링 면에서 다른 V모델의 면 중 가장 가까운 곳에 스위핑(sweeping)시켜 빨간색의 면으로 연결시킨다.



그림 4.33: 타일 간의 스위핑 연결[12]

본 연구는, Hong et al.[12]의 브리지 타일의 앵커링 면끼리 서로 스위핑하여 연결하는 알고리즘을 사용하여 다음과 같이 진행하였다.

- 두 실린더가 분기할때 충돌하는 실린더 그리드 조각들(4.2절), 생성된 커 널 곡면 상자와 충돌하는 실린더 그리드 조각들 안의 타일을 교차 타일로 정의하고, 해당 타일을 모두 제거한다.
- 각 실린더의 내부 타일 중 삭제되어 사라진 교차 타일과 가장 가까운 타일 들을 브리지 타일이라고 정의한다. 이 브리지 타일은 교차 타일을 삭제하여 생긴 공간 S에서 가장 가까운 거리에 위치하고 있는 그리드의 타일과 연결 한다.
- 각 타일은 6개의 실린더들로 이루어져 있으므로(4.3.1), 중심으로부터 가장 멀리 떨어진 실린더 단면을 각각 앵커링 면이라고 정의한다. 각 브리지 타일 의 앵커링 면은 S의 타일들까지의 거리 중 최소 거리를 가진 것을 선택하여 스위핑하여 연결된다.



그림 4.34: 커널 곡면이 존재하는 그리드의 렌더링

앞서 설명한 과정에 대해 예시와 함께 더 자세히 설명하도록 하겠다. 본 알고 리즘에서 연결되어야 하는 두 객체는 커널 곡면 안의 타일과 이 타일에서 가장 가까운 실린더 그리드 타일이다. 이를 위해, 그림 4.34와 같이 실린더 그리드 중 커널 곡면이 존재하는 부분은 안의 타일을 렌더링하지 않고 충돌 검사를 통해 알 아낸 주변 그리드의 타일들을 브리지 타일로 설정하였다. 본 연구에서는 시각적 편의를 위해 브리지 타일을 하늘색으로 렌더링하였다. 각 브리지 타일의 끝 면은 앵커링 면으로 정의되고 다른 타일의 앵커링 면과 연결될 수 있도록 모든 정점의 위치와 좌표가 저장된다.

또한, 커널 곡면 안의 타일들 중 가장 바깥에 위치한 타일들의 면을 커널 곡 면의 앵커링 면으로 정의하여 그림 4.35의 (a)와 같이 하늘색으로 렌더링하였다. 실린더 그리드의 브리지 타일과 마찬가지로 커널 곡면의 앵커링 면들 또한 모 든 정점의 위치 및 좌표를 저장하여 후에 다른 앵커링 면들과 연결할 수 있도록



그림 4.35: (a) 커널 곡면의 앵커링 면 (b) 부드럽게 변형된 커널 곡면 타일

하였다. 이 때, 그림 4.35의 (a)의 꺾인 타일들을 더 부드럽게 렌더링하기 위해 Hermite 보간법(interpolation)을 사용하여 (b)와 같이 매끄럽게 만드는 과정을 수행하였다.

본 알고리즘은 커널 타일의 각 앵커링 면에서 브리지 타일 중 가장 가까운 실린더 그리드를 거리 검사를 통해 구한 후, 그 안에서 가장 가까운 앵커링 면 을 선택하여 서로 연결한다. 앵커링 면을 이루는 정점의 개수를 n이라고 가정할 때, *i*번째 커널 곡면 타일의 앵커링 면의 정점들을 *P*_{ij}라 하고(*j* = 0...*n*) 점 *P*_i 에서 가장 가까운 실린더 그리드 *Q*_m의 선택된 정점들을 *Q*_{mk}(*k* = 0...*n*)라고 한다. 이 때, *P*_{ij}와 *Q*_{mk}를 각각 일대일 대응시켜 Hermite 보간법을 사용하여 부드럽게 연결시켜 메시(mesh)를 생성한다. 해당 알고리즘에서는 Cao et al.[5]



(a) Not connected

(b) Connected

그림 4.36: 브리지 타일과 커널 곡면 타일의 스위핑 연결

에 의해 제안된 방법을 통해 최소한의 계산으로 회전 변환을 구하여 알고리즘을 가속화하였다. Cao et al.은 double reflection[34]을 활용하여 불연속(discrete) 한 위치마다 rotation minimizing coordinate frame을 할당하는 방안을 제시하였 으며, 본 논문은 해당 기법을 통해 메시의 순서가 효율적으로 연결될 수 있도록 하였다.

다음으로 본 논문에서 두 종류의 타일을 서로 보간한 방법에 대해 상세히 설명 하도록 하겠다. 앞서 기술한 것과 같이, 두 타일은 Hermite 보간법으로 부드럽게 연결되었으며 각 앵커링 면의 모든 정점에 대해 수행하여 메시를 생성하였다. Hermite 보간법이란 두 끝점과 각 점에서의 접선 벡터(tangent vector)를 통해



그림 4.37: 두 타일 간의 Hermite 보간

 C^2 연속을 만족하는 3차 Hermite 곡선으로 표현하는 것을 말한다. 그림 4.37과 같이, 각 타일의 앵커링 면의 점을 제어점(control point) C_0 , C_3 으로 두고 각 점에서 접선 벡터 $\mathbf{t}_0, \mathbf{t}_3$ 를 더한 점을 나머지 제어점 C_1 , C_2 으로 설정하여 3차 Hermite 곡선을 3차 베지어(Bézier) 곡선 P(t)로 변환한다.

$$\begin{cases} C_0 = C_0 \\ C_1 = C_0 + \mathbf{t_0} \\ C_2 = C_3 + \mathbf{t_3} \\ C_3 = C_3 \end{cases}$$
(4.22)
$$P(t) = (1-t)^3 C_0 + (1-t)^2 t C_1 + (1-t) t^2 C_2 + t^3 C_3$$
(4.23)

생성된 베지어 곡선 위에 일정한 간격으로 점을 샘플링하고 이를 연결하면 그림 4.37과 같은 매끄러운 형태의 메시를 생성해낼 수 있다. 이 때, 제어점을 생성할 때 사용하는 t₀,t₃의 값을 수동으로 조정하여 곡선의 기울기를 원하는 만큼 조절 할 수 있다. 그림 4.35(b)의 커널 곡면 타일 또한 마찬가지로 보간하여 생성할 수 있다(그림 4.38).



그림 4.38: 커널 곡면 타일의 Hermite 보간

제 5 장 실험 결과

본 연구는 3D 모델링 소프트웨어 툴 중 하나인 블렌더(Blender 2.93)[7]에 서 Python Script를 사용하여 모든 물체를 렌더링하였다. 해당 프로그램은 Intel Core i7-6700 3.40GHz CPU, 16.0 GB의 램(RAM), 그리고 NVIDIA GeForce GTX 1070의 환경에서 실행되었다.

본 논문에서는 혈관의 형태를 표현하기 위해 메인 실린더의 크기에 비해 서브 실린더의 크기를 더 작게 설정하였으며, [u, v, w] 변수들 중 실린더의 두께(v)의 그리드 개수를 현저히 더 적게 설정하여 얇게 나타내었다. 메인 실린더의 내부 실린더의 반지름의 길이는 3, 외부 실린더의 반지름의 길이는 3.5, 높이는 20으로 설정하였고, 서브 실린더의 내부 실린더와 외부 실린더의 반지름의 길이 각각 1, 1.5로 메인 실린더보다 작게 설정하였다. 서브 실린더의 높이는 메인 실린더와 똑같이 20으로 설정하였다. 총 6개의 큰 실린더와 각각을 나누는 작은 실린더로 구성되는 기본 타일은, vertex를 16으로 설정하여 렌더링의 품질을 높였으며, 프 로그램의 속도에 영향을 미치기 때문에 지나치게 큰 숫자로는 설정하지 않았다. 롤링 볼 알고리즘에 사용되는 디자인 변수 반지름 δ는 90도로 교차하는 경우에는 2로 설정하였지만, 그 외의 각도에서는 비틀어짐 및 오류가 생길 수 있기 때문에 1로 설정하였다.

본 알고리즘은 BVH 이진 트리를 사용하기 때문에, 모든 그리드의 개수가 2 의 배수여야 한다. 그림 5.1은, 총 [2⁶,2²,2⁷]개의 그리드를 렌더링한 결과이다. 총 65536개의 실린더 그리드들 안에 타일 미세구조가 채워져 있고, 이들이 서로 잘 연결되어있음을 확인할 수 있다.

48



그림 5.1: 미세구조가 채워진 분기 구조

그림 5.1의 (b)는 커널 곡면도 같이 렌더링한 결과이다. 실린더 그리드 안의 타 일들과, 커널 곡면 안의 타일들의 크기가 비슷하여 서로 연결되기 쉽게 되어있다. 그림의 (d)는 그림 실린더의 내부의 모습이다. 서로 충돌하는 그리드의 타일은 초록색으로, 그렇지 않은 그리드의 타일은 빨간색으로 설정하였다. 초록색 타일 들은 이후에 전부 제거되어 다른 타일들과 연결된다.



그림 5.2: 여러 각도의 블렌딩 곡면 (순서대로 30, 45, 60도)



그림 5.3: 여러 각도의 미세구조 모델링 (순서대로 30, 45, 60도)



그림 5.4: 연결된 미세구조가 채워진 분기 구조



그림 5.5: 내부의 연결된 미세구조

그림 5.4는 총 $[2^4, 2^1, 2^4]$ 개의 그리드를 가지는 두 실린더를 렌더링하여 서로 교차하는 부분의 미세구조를 전부 제거한 후 이 안을 연결한 결과이다. 그림 5.5과 같이 내부가 제대로 연결됨을 확인할 수 있다.

제 6 장 결론

본 논문에서는 두 개의 실린더가 교차하여 만날 때 생기는 분기 구조를 미 세구조로 모델링하는 방법에 대해 제시하였다. 실린더 사이의 분기 구조를 롤링 볼 블렌딩 알고리즘으로 매끄럽게 블렌딩하여 교차하는 각도에 따라 다른 커널 곡면을 만들었다. 또한, BVH 생성 알고리즘을 통해 서로 충돌하는 실린더 그리 드들을 알아내어 이를 제거할 수 있게끔 하였다. 마지막으로, 두 실린더와 커널 곡면, 그리고 충돌이 확인되어 제거된 공간 안을 적절히 매개변수화하여 미세구 조로 모델링하는 방법에 대해 논하였다.

이를 통해 만들어진 분기 구조 스윕 모델은 실제 혈관의 모양과 특성을 적용하 여 더 정교하게 만들 수 있으며, 미세구조의 무게나 밀도(density) 등을 조절하여 혈관의 내벽과 외벽을 구현할 수 있을 것으로 보인다. 또한, 렌더링 소프트웨어가 아닌 실제 3D 프린터를 사용하면 훨씬 더 미세하게 나노 및 마이크로 레벨로 혈관 모델을 제작하는 것 또한 가능할 것이다.

본 연구에서는 두 개의 실린더가 만나 분기 구조를 가지는 경우에 대해서만 논하였지만, 같은 원리를 사용하여 세 개 이상의 여러 개의 실린더가 만나는 분기 구조에 대해서도 블렌딩이 가능하다. 마찬가지로, 서브 실린더가 두 개 이상인 경 우에도 각각에 대해 본 논문의 알고리즘을 수행하면 문제없이 여러 개의 블렌딩 구조가 생기게 될 것으로 보인다. 실제 혈관 구조는 두 개의 실린더 형태가 아닌, 수많은 실린더가 서로 얽혀있는 복잡한 형태를 가지고 있다. 향후 연구로는 더 현실적인 혈관 구조를 나타내기 위한 다수의 실린더를 처리하는 방법이 필요할 것으로 보인다.

54

참고문헌

- P. Antolin, A. Buffa, E. Cohen, J. F. Dannenhoffer, G. Elber, S. Elgeti, R. Haimes, and R. Riesenfeld. Optimizing micro-tiles in micro-structures as a design paradigm. *Computer-Aided Design*, 115:23–33, 2019.
- [2] A. Armillotta and R. Pelzer. Modeling of porous structures for rapid prototyping of tissue engineering scaffolds. *The International Journal of Advanced Manufacturing Technology*, 39(5):501–511, 2008.
- [3] G. Bradshaw and C. O'Sullivan. Sphere-tree construction using dynamic medial axis approximation. In Proceedings of the 2002 ACM SIG-GRAPH/Eurographics symposium on Computer animation, pages 33–40, 2002.
- [4] T. Burczyński and W. Kuś. Microstructure optimization and identification in multi-scale modelling. In *Eccomas multidisciplinary jubilee symposium*, pages 169–181. Springer, 2009.
- [5] L. Cao, Z. Yuan, B. Chan, and W. Wang. Reprojection of textured depth map for network rendering. In *Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pages 1–7. IEEE, 2014.
- [6] V. Chandru, D. Dutta, and C. M. Hoffmann. On variable radius blending using dupin cyclides. 1989.

- [7] B. O. Community. Blender a 3D modelling and rendering package.Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.
- [8] W. Gao, Y. Zhang, D. Ramanujan, K. Ramani, Y. Chen, C. B. Williams, C. C. Wang, Y. C. Shin, S. Zhang, and P. D. Zavattieri. The status, challenges, and future of additive manufacturing in engineering. *Computer-Aided Design*, 69:65–89, 2015.
- C. Hoffmann and J. Hopcroft. Quadratic blending surfaces. Computer-Aided Design, 18(6):301–306, 1986.
- [10] C. M. Hoffmann. Algebraic and numerical techniques for offsets and blends. In *Computation of curves and surfaces*, pages 499–528. Springer, 1990.
- [11] C. M. Hoffmann and J. E. Hopcroft. The potential method for blending surfaces and corners. Technical report, Cornell University, 1985.
- [12] Q. Y. Hong and G. Elber. Conformal microstructure synthesis in trimmed trivariate based v-reps. *Computer-Aided Design*, 140:103085, 2021.
- [13] P. M. Hubbard. Approximating polyhedra with spheres for time-critical collision detection. ACM Transactions on Graphics, 15(3):179–210, 1996.
- [14] D. W. Hutmacher. Scaffolds in tissue engineering bone and cartilage. Biomaterials, 21(24):2529–2543, 2000.
- [15] Y. Liu, G. Zheng, N. Letov, and Y. F. Zhao. A survey of modeling and optimization methods for multi-scale heterogeneous lattice structures. *Journal of Mechanical Design*, 143(4):040803, 2021.

- [16] D. Mahmoud and M. A. Elbestawi. Lattice structures and functionally graded materials applications in additive manufacturing of orthopedic implants: a review. *Journal of Manufacturing and Materials Processing*, 1(2):13, 2017.
- [17] J. Martínez, J. Dumas, and S. Lefebvre. Procedural voronoi foams for additive manufacturing. ACM Transactions on Graphics, 35(4):1–12, 2016.
- [18] F. Massarwi, J. Machchhar, P. Antolin, and G. Elber. Hierarchical, random and bifurcation tiling with heterogeneity in micro-structures construction via functional composition. *Computer-Aided Design*, 102:148– 159, 2018.
- [19] A. Nazir, K. M. Abate, A. Kumar, and J.-Y. Jeng. A state-of-the-art review on types, design, optimization, and additive manufacturing of cellular structures. *The International Journal of Advanced Manufacturing Technology*, 104(9):3489–3510, 2019.
- [20] D. S. Nguyen and F. Vignat. A method to generate lattice structure for additive manufacturing. In 2016 IEEE international conference on industrial engineering and engineering management (IEEM), pages 966– 970. IEEE, 2016.
- [21] J. Panetta, Q. Zhou, L. Malomo, N. Pietroni, P. Cignoni, and D. Zorin. Elastic textures for additive fabrication. ACM Transactions on Graphics, 34(4):1–12, 2015.

- [22] M. J. Pratt. Cyclides in computer aided geometric design. Computer Aided Geometric Design, 7(1-4):221-242, 1990.
- [23] S. Quinlan. Efficient distance computation between non-convex objects. In Proceedings of the 1994 IEEE International Conference on Robotics and Automation, pages 3324–3329. IEEE, 1994.
- [24] A. P. Rockwood. The displacement method for implicit blending surfaces in solid models. ACM Transactions on Graphics, 8(4):279–297, 1989.
- [25] A. P. Rockwood and J. C. Owen. Blending surfaces in solid modeling. Geometric Modeling: Algorithms and New Trends, pages 367–383, 1987.
- [26] J. Rossignac and A. Requicha. Constant-radius blending in solid modelling. 1984.
- [27] M. A. Sanglikar, P. Koparkar, and V. Joshi. Modelling rolling ball blends for computer aided geometric design. *Computer Aided Geometric Design*, 7(5):399–414, 1990.
- [28] T. A. Schaedler and W. B. Carter. Architected cellular materials. Annual Review of Materials Research, 46:187–210, 2016.
- [29] M. Sitharam, J. Youngquist, M. Nolan, and J. Peters. Cornersharing tetrahedra for modeling micro-structure. *Computer-Aided Design*, 114:164–178, 2019.
- [30] G. Strang and E. P. Herman. Calculus. Volume 1. OpenStax College, Rice University, 2016.

- [31] F. Tamburrino, S. Graziosi, and M. Bordegoni. The design process of additively manufactured mesoscale lattice structures: a review. *Journal* of Computing and Information Science in Engineering, 18(4), 2018.
- [32] J. J. Van Wijk. Ray tracing objects defined by sweeping a sphere. Computers & Graphics, 9(3):283–290, 1985.
- [33] J. Vida, R. R. Martin, and T. Varady. A survey of blending methods that use parametric surfaces. *Computer-Aided Design*, 26(5):341–365, 1994.
- [34] W. Wang, B. Jüttler, D. Zheng, and Y. Liu. Computation of rotation minimizing frames. ACM Transactions on Graphics, 27(1):1–18, 2008.
- [35] J. R. Woodwark. Blends in geometric modelling. In Proceedings on Mathematics of surfaces II, pages 255–297, 1987.
- [36] J. Youngquist, M. Sitharam, and J. Peters. A slice-traversal algorithm for very large mapped volumetric models. *Computer-Aided Design*, 141:103102, 2021.

Abstract

We introduce an effective method of modeling a blood vessel model in which two cylinders cross each other and filling it with microstructures. First, we propose to soften the steep-curved surface generated between the two cylindrical branch structures, by using a rolling-ball blending method. To this end, we present a mathematical formula for generating a kernel surface passing through the two branch structures at the same time. In addition, we show that it is possible to create a smooth kernel curved surface to the extent desired by the user according to the size and use of the blood vessel model.

In order to fill the blended branch structure with microstructures, we present a method of reparameterization of the space in each cylinder-shaped model. After dividing each cylinder models into smaller-sized grids, a bounding volume hierarchy consisting of a sphere surrounding each grid can be obtained. Through collision detection using this data structure, it is possible to quickly and easily distinguish the cylinder grids intersecting each other. Furthermore, the surplus grids that are not required for modeling can be efficiently found and removed using this data structure.

In addition to the cylinder model, we present a method of dividing the kernel surface created in the branch structure into several bonding volumes and filling it with microstructures of similar size. Also, an efficient method of filling the space in which the two cylinders intersect so that they can be smoothly connected to other microstructures is presented.

Keywords: Microstructure, Rolling-ball blending, Branching structure, Bounding Volume Hierarchy, Canal surface, 3D modelingStudent Number: 2020-23988