

3D 프린터를 위하여 3D 모델 데이터의 분산 변환 기법 개발¹⁾

모준서*, 주우성*, 이규영*, 김성석*, 양순옥**
*서경대학교 컴퓨터학과
**가천대학교 컴퓨터공학과

Developing a distributed conversion algorithm of 3D model data for 3D printers

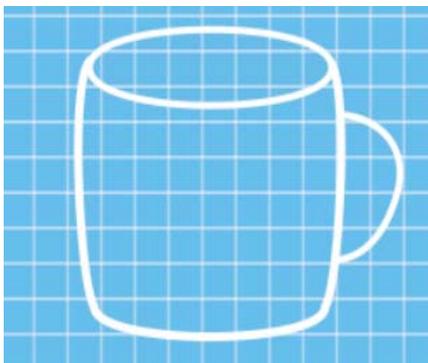
Junseo Mo*, Woosung Joo*, Kyooyoung Lee*, Sungsook Kim*,
Sun Ok Yang**

*Dept of Computer Science, Seokyeong University

**Dept of Computer Engineering, Gachon University

요 약

3D 프린터는 연속적인 계층에 특수한 재료를 출력시켜 3차원 물체를 만들어 내는 장치이다. 3D 프린팅을 위해서는 3D 모델을 생성한 후, 이를 3D 프린터에 출력할 수 있도록 G-code로 변환해야 한다. 본 논문에서는 이 변환 작업을 완전 분산 방식으로 처리할 수 있는 알고리즘을 제안한다. 이를 위해 하나의 메인 노드와 N개의 작업 노드로 구성된 시스템에서 2단계에 걸쳐 분할 정복(divide-and-conquer) 방식으로 변환하도록 하였다. 실제 구현한 시스템을 이용하여, 성능에 미치는 요소(모델의 크기 및 정밀도)에 따른 변환 시간의 단축 효과를 보였다.



(a) 3D 모델 데이터 생성



(b) 3D프린터에 의해 계층적 출력
(그림 1) 3D 프린팅 과정



(c) 완성된 3D 출력물

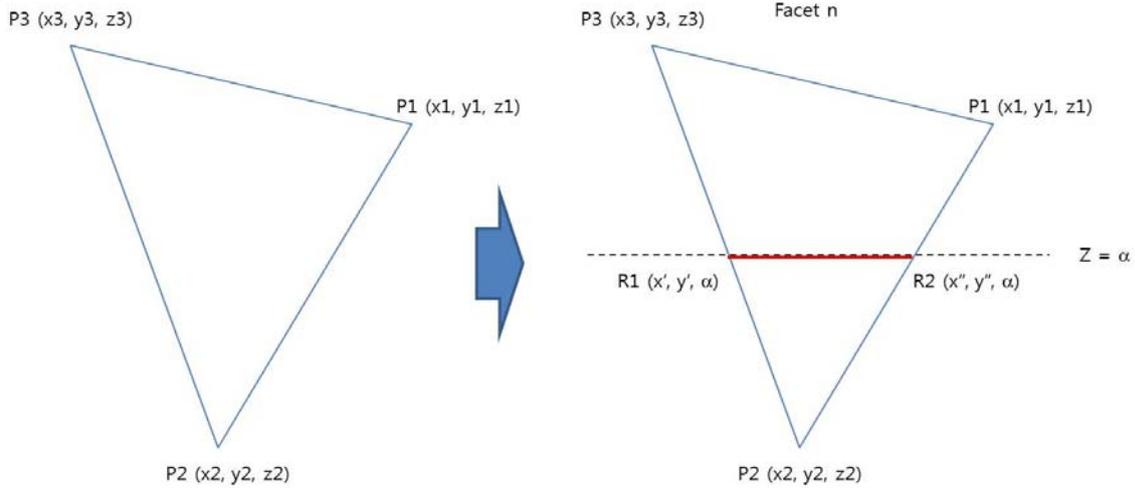
1. 서론

3D 프린터는 연속적으로 잉크를 뿌려 3차원 물체를 만들어내는 제조 기술로, 1984년 처음 개발되었다[1, 2]. 이후 여러 요소 기술들이 함께 발전되면서 최근 제조 혁명을 이룰 수 있을 것으로 평가받고 있으며, 최근 몇 년간 가트너에서 발표한 10대 전략 기술에 3D 프린팅과 관련된 내용이 포함되어 있다. 3D 프린팅을 하기 위해서 먼저 3D 모델을 생성해야 하는데, 이는 3D CAD와 같은 일반적인 소프트웨어나 3D 스캐너와 같은 장비를 사용하여 얻

을 수 있다. 일반적으로 3D 프린터는 제일 아래 면부터 계층적으로 재료를 뿌려 최종적으로 3D 출력물을 얻게 된다(그림 1 참고). 이러한 3D 프린팅에는 다수의 학문 분야가 연관되어 있다. 3D 프린터를 만들기 위한 장비 제조 기술, 각 응용에 맞는 재료를 연구하는 재료나 화학 분야, 그리고 3D 모델을 생성하여 3D 프린터용 데이터로 변환하는 기술이 필요하다. 본 논문에서는 3D 모델을 이용하여 3D 프린터가 인식할 수 있는 명령어인 G-code를 생성하는 분야에 관심이 있다.

일반적으로 3D 모델은 STL 파일 포맷으로 저장되어 있는데, 이는 3D 모델을 구성하는 면을 작은 삼각형 모양의 페이스(Facet)으로 표현한다[3, 4]. 이에 반해 3D 프린

* “이 논문은 2015년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2015R1D1A1A01061173)”



(그림 2) 현재 플레이트 위치($Z=\alpha$)와 만나는 페이스트에 대한 출력위치 결정 방법

터는 대개 출력물이 생성될 플레이트를 일정 위치(Z 축 기준)에 고정시킨 후 헤더가 X - Y 축으로 이동하면서 적절한 위치에 재료를 뿌려 프린팅이 진행된다. 이렇게 3D 프린터의 헤드 및 플레이트를 이동시키는 명령어는 대개 G-code로 표현된다[5]. 하지만 하나의 3D 모델에는 엄청나게 많은 수의 페이스트가 있다. 일반적인 변환 작업은, 현재 플레이트의 위치와 겹치는 페이스트를 모두 찾은 후, 이들로부터 헤더의 X - Y 축으로의 이동 및 출력을 담당하는 G-code 명령어로 변환해야 한다. 이 변환 작업은 많은 시간이 소요되므로, 이 시간을 단축할 수 있는 분산 알고리즘을 개발하는 것이 본 논문의 주요한 내용이다.

2. 페이스트로부터 G-code 분산 변환 알고리즘

G-code는 플레이트를 위·아래로 일정간격만큼 이동하고 멈추는 명령, 헤드를 X - Y 좌표로 이동하는 명령, 이동하면서 재료를 분출하는 명령으로 구분할 수 있다. 일단 플레이트가 $Z = Z_c$ 위치에 멈추면, 페이스트 중 (가장 작은 Z 축의 값 $\leq \alpha \leq$ 가장 큰 Z 축의 값)을 만족시키는 페이스트를 찾아야 한다. 해당 페이스트가 $Z = Z_c$ 위치의 플레이트와 만나는 선분을 계산하면, 이 선분이 결국 헤더가 이동하면서 재료를 뿌리는 위치가 된다(그림 2 참고). 하지만 이 작업은 3D 모델의 크기, 정밀도 등에 따라 매우 시간이 많이 소요되는 작업이다. 따라서 이를 분산방식으로 처리할 수 있는 알고리즘을 제안한다.

STL 파일로부터 G-code를 생성하는 것은 두 단계에 걸쳐 진행된다. 첫번째 단계로서, 모든 페이스트로부터 가장 큰 Z 값과 가장 작은 Z 값의 쌍을 찾은 후 이를 가장 작은 Z 값을 기준으로 정렬한다. 두 번째 단계는 플레이트가 $Z = 0$ 부터 단위 크기(α)만큼 위로 이동할 때 만나는 페이스트들을 찾아서 그림 2처럼 교차하는 선분(실제 잉크가 뿌려져야 하는 부분)을 계산하는 작업이다. 이러한 각 단계들은 완전 분산방식으로 동작할 수 있으므로, 이를 작업 노드의 개수(N)만큼 배분하여 나누어 준 후 결과를 취

합하는 방식으로 동작한다(분할정복 기법). 이를 위해 시스템에는 총 $N+1$ 개의 노드가 있다고 가정한다. 0번 노드는 메인 노드 역할을 하며, 1~ N 번 작업 노드는 메인 노드로부터 작업을 받아 처리하는 작업을 한다. 즉, 매 단계마다 메인 노드는 작업을 분배하여 각 작업 노드에게 나눠준다. 모든 작업노드로부터 결과를 받으면(2.1 절), 이를 취합한 후 다음 단계에도 다시 작업 노드에게 배분한 후 취합한다(2.2 절).

2.1 단계 1 : 페이스트 정렬하기

메인 노드는 3D 모델 데이터를 N 등분하여 각 작업 노드들에게 전달한다. 작업 노드들은 각각 받은 페이스트로부터 (id, Z_{max}, Z_{min}) 쌍의 데이터를 추출한다. 이 추출된 데이터 집합을 F_{tag} 라고 한다. 이후 F_{tag} 에서 각 데이터를 Z_{min} 에 따라 오름차순으로 정렬한 후 이를 메인 노드에게 전달한다.

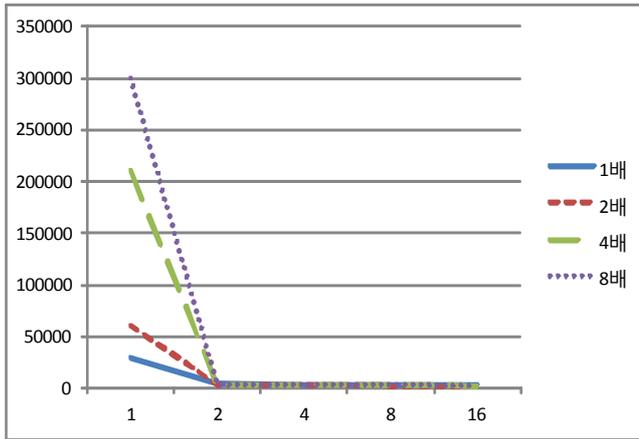
2.2 단계 2 : 정렬된 페이스트를 이용하여 G-code 생성하기

대상 3D 모델의 크기를 H 이라고 한다면, 메인 노드는 단계 1에서 정렬된 데이터들을 높이 값에 따라 다시 N 등분한다. 즉 정렬된 페이스트 데이터들을 높이 값(즉, Z 값의 최소값)에 따라 $[0, \lceil H/N \rceil], [\lceil H/N \rceil, \lceil H/N \rceil \times 2], \dots, [\lceil H/N \rceil \times (N-1), \lceil H/N \rceil \times N]$ 로 구분하여 각각의 작업 노드들에게 전달한다. 각 작업노드는 플레이트가 자신의 담당하는 높이일 때 만나는 페이스트를 찾아 그림 2처럼 만나는 선분을 계산하고, 이를 G-code로 변환한다.

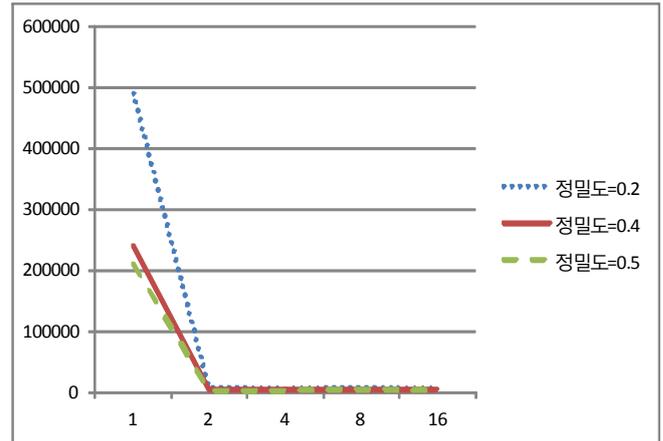
3. 실험

제안한 분산 알고리즘의 성능을 평가하기 위한 시스템

- 2) 현재 페이스트에서 가장 큰 Z 값
- 3) 현재 페이스트에서 가장 작은 Z 값



(a) 3D 모델의 크기에 따른 변환시간



(b) 3D모델의 정밀도에 따른 변환시간

(그림 3) 분산 알고리즘의 성능 측정

을 JAVA언어로 구현하였다. 이 실험을 위해 우선 CAD를 통해 3D 모델 데이터를 생성한 후, 이 데이터의 크기/정밀도와 참여하는 작업 노드의 수를 변환하면서 변환에 걸리는 시간을 측정하였다. 측정된 시간을 비교하기 위하여 이 변환작업에 널리 사용되고 있는 CURA 소프트웨어와 비교 작업을 수행하였다[6]. 즉 CURA는 STL 파일을 G-code로 변환하는데 널리 사용된다. 다만 이것은 단일 시스템에서만 동작하므로, 본 논문의 분산 알고리즘의 성능 비교를 위한 기본 값으로 적절하다.

3.1 3D 모델의 크기에 따른 변환 시간

먼저 3D 모델의 크기를 변화시키면서 변환 시간에 미치는 영향을 측정하였다(정밀도 = 0.5). 정밀도가 고정되어 있으므로, 대상의 크기가 커질 때 대개는 그 크기에 비례하여 하나의 페이지의 크기도 커질 것이다. 하지만, 실제 3D 모델에서는 더 많은 수의 페이지가 생성되며, 따라서 그만큼의 처리 시간이 늘어나게 된다. (그림 3)-(a)에서 X-축은 작업노드의 개수를 의미하며, 따라서 작업노드가 1인 경우는 CURA 소프트웨어에서 얻어진 기본값을 의미한다(최악의 경우). 모든 경우에서 노드의 개수가 1일 때 변환 시간이 제일 오래 걸리며, 분산 알고리즘을 사용하면 훨씬 적은 시간 내에 처리됨을 알 수 있다. 실제 1배에 대하여 노드가 한 개일 때 걸리는 시간은 대략 30,000초가 걸리지만, 8배가 되면 300,000초 이상이 걸렸다. 하지만 제안한 분산 알고리즘을 사용하면, 1배-8배까지 크기를 변화시키더라도 걸리는 시간은 2,100~2,600초 이내에 변환이 가능했다.

3.2 3D 모델의 정밀도에 따른 변환 시간

3D 모델의 정밀도가 정밀해지면, 훨씬 많은 수의 페이지로 모델이 구성되게 된다. 따라서 당연히 처리 시간이 많이 소요되게 된다. (그림 3)-(b)는 모델의 크기를 1배로 가정하고 정밀도를 0.5~0.2까지 변화할 때 걸리는 시간을 보여주고 있다(정밀도 값이 작을수록 훨씬 정밀함을 의미

한다). 물론 작업노드 1개에서 처리하는 CURA에 비해 1/100의 시간내에 처리됨을 알 수 있다.

4. 결론

본 논문에서는 3D 모델 데이터를 3D 프린팅을 위해 G-code로 변환하기 위한 분산 알고리즘을 제안하였다. 이 기법은 기본적으로 분할 정복 알고리즘 방식으로 처리되었으며, 그 구현 과정이 단순하여 기존 소프트웨어에 비해 획기적으로 변환시간을 단축하였다.

앞으로 제안한 알고리즘을 클라우드 시스템에서 구현하고자 한다. 클라우드 시스템은 다양한 목적으로 사용될 수 있으며, 시간이 많이 소요되는 G-code 변환 작업을 클라우드에서 처리될 수 있다면, 3D 프린터 대중화 및 클라우드 시스템 활용 다양화로 이어질 것으로 판단된다.

참고문헌

- [1] https://en.wikipedia.org/wiki/3D_printing
- [2] Jacobs, Paul Francis, "Rapid Prototyping & Manufacturing: Fundamentals of Stereolithography". Society of Manufacturing Engineers. 1992
- [3] "StereoLithography Interface Specification", 3D Systems, Inc., 1988
- [4] Lynch, Mike, "When programmers should know G code", Modern Machine Shop (online ed.), 2010
- [5] 3dprintingindustry.com, "THE FREE BEGINNER'S GUIDE TO 3D PRINTINGS"
- [6] <https://ultimaker.com/en/products/cura-software>