한국정밀공학회지 제 42 권 제 1 호 pp. 89-96 January 2025 / 89
J. Korean Soc. Precis. Eng., Vol. 42, No. 1, pp. 89-96 http://doi.org/10.7736/JKSPE.024.124
ISSN 1225-9071 (Print) / 2287-8769 (Online)

Check for updates

3D 프린팅 섬 영역 탐지 방법: CPU 기반 접근법

Detection Method for Island Regions in 3D Printing: A CPU-based Approach

강영석[1], 김윤섭[2], 나승채[2], 한상조[2,#]
Young Seok Kang[1], Yeun Seop Kim[2], Seung Chae Na[2], and Sang Jo Han[2,#]

1 서울과학기술대학교 대학원 기계공학과 (Department of Mechanical Engineering, Graduate School, Seoul National University of Science and Technology)
2 서울과학기술대학교 기계·자동차공학과 (Department of Mechanical Engineering and Automotive Engineering, Seoul National University of Science and Technology)
Corresponding Author / E-mail sjhan@seoultech.ac.kr, TEL: +82-2-970-6321
ORCID: 0009-0001-8463-1461

KEYWORDS: Additive manufacturing (적층 제조), 3D printing (3D 프린팅), Support structure (지지대)

Additive manufacturing, a key enabler of Industry 4.0, is revolutionizing the automatic landscape in manufacturing. The primary challenge in manufacturing innovation centers on the implementation of smart factories characterized by unmanned production facilities and automated management systems. To overcome this challenge, the adoption of 3D printing technologies, which offer significant advantages in standardizing production processes, is crucial. However, a major obstacle in complete automation of additive manufacturing is an inadequate placement of support structures at critical locations, which remains the leading cause of print failures. This study proposed a novel algorithm for accurate detection of island regions known to be critical areas requiring support structures. The algorithm can compare loops on two consecutive layers derived from STL files. In contrast to conventional GPU-based image comparison methods, our proposed CPU-based algorithm enables high-precision detection independent of image resolution. Experimental results demonstrated the algorithm's efficacy in enhancing the reliability of 3D printing processes and optimizing automated workflows. This research contributes to the advancement of smart manufacturing by addressing a critical challenge in the automation of additive manufacturing processes.

Manuscript received: October 10, 2024 / Revised: November 11, 2024 / Accepted: November 14, 2024

NOMENCLATURE

θ	=	overhang Angle
S_n	=	Sectional Plane at n^{th} Height
$L_{A.S1}$	=	A-exclusive Cross-sectional Shape Loop on Plane S_1
$L_{P.S2}$	=	P-exclusive Cross-sectional Shape Loop on Plane S_1
$L_{A.S2}$	=	A-exclusive Cross-sectional Shape Loop on Plane S_2
$L_{B.S2}$	=	B-exclusive Cross-sectional Shape Loop on Plane S_2
$L_{P.S2}$	=	P-exclusive Cross-sectional Shape Loop on Plane S_2
$L_{A.S3}$	=	A-exclusive Cross-sectional Shape Loop on Plane S_3
$L_{B.S3}$	=	B-exclusive Cross-sectional Shape Loop on Plane S_3
$L_{C.S3}$	=	C-exclusive Cross-sectional Shape Loop on Plane S_3
$L_{P.S3}$	=	P-exclusive Cross-sectional Shape Loop on Plane S_3
$L_{m.S4}$	=	All Merged Loops on Plane S_4
$L_{T.S1}$	=	Total Cross-sectional Shape Loop on Plane S_1
$L_{T.S2}$	=	Total Cross-sectional Shape Loop on Plane S_2
$L_{T.S3}$	=	Total Cross-sectional Shape Loop on Plane S_3
RL	=	Residual Loop
NR	=	Null Residual
SR	=	Sustained Residual

1. 서론

3D 프린팅은 4차 산업혁명의 핵심 기술로서 제조업의 혁신을 주도하며 다양한 산업 분야에서 급속히 확산되고 있다[1]. 이 기술은

Copyright © The Korean Society for Precision Engineering
This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

의료, 우주항공, 자동차, 건축, 패션 등 광범위한 분야에서 활용되며, 최근에는 4D 프린팅 기술로도 발전하고 있다[2]. 또한, 단일화된 부품 생산을 통해 경량화 및 부품 수 감소로 불량률 저감과 생산시간 단축을 실현하여 비용 절감에 기여한다.

세계 각국은 제조경쟁력 혁신을 위해 스마트 팩토리의 필요성을 강조하며 적극적으로 지원 및 추진하고 있다. 한국은 3D 프린팅, 스마트센서, CPS, 에너지 절감 기술, IoT, 클라우드, 빅데이터, 홀로그램 등을 스마트제조의 8대 핵심기술로 지정하였다[3]. 최근 스마트팩토리는 인공지능을 활용한 스마트 MES (Manufacturing Execution System) 구축을 목표로 하며, 이를 위해 ICT (Information and Communication Technology)가 적용된 대량의 표준화된 데이터가 필요하다. 3D 프린터는 기존 생산제조장비에 비해 생산공정의 표준화 및 디지털화에 유리하다[4].

그러나 3D 프린팅 공정의 완전 자동화에는 여전히 과제가 남아있다. 가장 큰 문제는 지지대(서포터)가 필요한 출력 모델에서 필수 위치를 자동으로 식별하고 생성하는 것이다. 현재까지 지지대의 필수 위치는 주로 작업자의 경험에 의존하여 결정되었으며, 이는 출력 성공률에 큰 영향을 미쳤다. 본 연구는 3D 프린팅 자동화의 핵심 요소인 지지대 필수 위치 탐색 알고리즘을 제안한다.

2. 지지대 생성 위치 탐색

2.1 지지대의 필요성

3D 프린팅 공정 자동화의 핵심 요소는 반복적으로 사용자의 개입 없이 출력 성공이 가능해야 한다는 점이다. 출력실패의 원인을 정확히 파악하여 비슷한 형상의 문제를 사전에 해결해야 한다. 다양한 출력 방식이 존재함에 따라 3D 프린팅 실패의 원인 또한 다양하다. 부적절한 출력 방향, 출력 불가능한 형상, 외부 환경 문제(온도, 진동 등), 공정속도 등 다양한 원인이 있다. 출력 실패의 원인을 해결하기 위해 출력방향 전환, 모델 형상 변경, 모델 분할 출력, 구조물 보강, 온도조절, 진동 조절, 지지대 생성 등 다양한 요소를 적용 또는 통제하여 해결한다. 다양한 출력 방식 중 실패의 원인이 비교적 적어 자동화에 유리한 출력 방식들이 있지만 이는 결국 높은 비용과 직결된다. 결국 낮은 비용의 출력 장비들은 숙련된 작업자가 개입하여 출력의 성공률을 올리는 작업을 하게 된다.

3D 프린터는 사용되는 재료 또는 적층 원리에 따라 다양한 장비가 존재한다. ISO 17296-1에 따라 3D 프린터는 7가지 카테고리로 분류할 수 있다[5,6].

액조 광경화(Vat Polymerization) - DLP, SLA 등
재료 분사(Material Jetting)
접착제 분사(Binder Jetting)
재료 압출(Material Extrusion) - FDM, DMD 등
분말 베드 융해(Powder Bed Fusion) - SLS 등
판재 적층(Sheet Lamination)
에너지 제어 용착(Directed Energy Deposition) - SLM 등

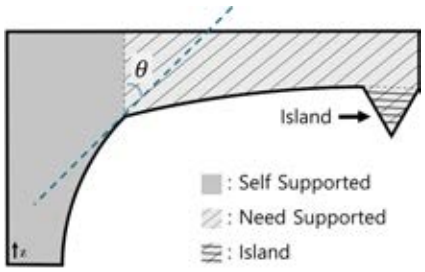


Fig. 1 The angle θ shown in the figure is 45°. Overhang part is where the angle θ exceeds a threshold angle. Self -supported area where the angle is under 45° usually can stand without supporters

다양한 방식에도 불구하고 대부분의 출력 실패 원인은 반드시 지지대가 필요한 위치에 적절하게 생성되지 못한 상황에서 발생된다. 지지대가 필요한 이유 또한 다양하다. 적층 중인 재료의 중력에 의한 변형, 온도 또는 잔류응력에 의한 변형, 재료의 도포 및 이동에 의한 변형, 부족 또는 과다한 지지대 등이 있다[7]. 우리가 주로 다루게 될 FDM과 DLP 방식에서는 출력 모델의 오버행과 섬에 적절한 지지대 생성이 이루어져야 한다. Fig. 1에서 그려진 모습과 같이 섬(Island) 영역은 지지대가 없다면 재료와 각도 무관하게 형상이 적층 될 수 없다. 연구에서는 이 섬의 위치를 탐색하는 알고리즘에 대해 설명하고자 한다.

2.2 STL 파일 형식과 지지대 생성 선행 연구

출력하고자 하는 모델은 STL (Stereolithography) 형식의 파일로 저장된다. STL 파일은 3D 모델을 삼각형을 최소 단위로 표현하며 “Standard Triangle Language”나 “Standard Tessellation Language”로 표현하는 경우도 있다.

STL 파일에서 물체의 표면을 구성하는 삼각형은 12개의 수로 이루어져 있으며 한 개의 법선 벡터와 세 개의 꼭짓점을 3차원 좌표 표현한다. 오른손 법칙에 따라 법선 벡터N 방향에 맞추어 V_1, V_2, V_3 좌표를 정리하고 안과 밖을 구분한다[8].

$$N = (n_x, n_y, n_z)$$

$$V_1 = (x_1, y_1, z_1)$$

$$V_2 = (x_2, y_2, z_2)$$

$$V_3 = (x_3, y_3, z_3)$$

지지대 생성이 필요한 위치를 탐색하는 방법에는 많은 연구가 이루어져왔고 크게 두 가지로 나뉜다. 평면의 법선 벡터와 Z축 관계를 활용하여 주어진 조건에 만족하는 지지대 생성 위치 여부를 판별하는 방법과 연속된 단면을 비교 분석하여 탐색하는 방법이다. 이 두 기본적인 방법을 응용하여 다양한 연구가 진행되었다.

Vanek, J. 등은 주변 지점들보다 낮은 위치에 있는 지점을 섬으로 정의했다[9]. 그들은 평면의 법선 벡터가 기준 각도를 초과하는 위치에 원뿔을 생성하고, 이 원뿔들이 교차하는 영역을

활용하여 효율적으로 필수 지지대를 생성하는 방식을 제안하였다. 하지만 지지 구조 생성의 효율성을 높였으나, 섬 탐지 과정에서 한계점을 보였다. 주변 지점들과 비교는 임의로 지정된 간격을 기준으로 이루어졌기 때문에, 이 간격 설정에 따라 탐지 정확도가 달라질 수 있었다. Chen., Yong. 등은 LDNI (Layered Depth-Normal Image) 기법 활용하여 분석하고 지지대 위치 지정했다[10]. LDNI 기법을 활용하여 얻어진 1차원 정보들을 분류하여 각 레이어의 모형 외곽선(Contour)을 추출한다. 이후 전후 레이어의 외곽선을 비교하여 교차되지 않는 영역을 식별하고, 두 외곽선 간의 이격 거리를 기반으로 직전 레이어가 자체적으로 지지 가능한지 분석하여 지지대를 생성한다. 또한, 미리 준비된 XY 격자 간격에 따라 배치된 지지대와 외곽선 간의 거리를 분석하여 지지가 필요한 영역을 결정한다고 하였다. 이 방법은 섬 구조 탐지에 앞서 미리 설정된 격자를 기반으로 생성된 지지대와의 관계를 분석하여 지지 가능성을 분석하여 필요에 따라 수정하는 방식이었다. 따라서 직접적인 섬 탐지 알고리즘은 아니었다. 또한 분석 간격, 스스로 지지 가능 거리, 지지대의 지지 가능 거리 등의 변수를 경험을 바탕으로 설정하여 진행하였다. Jang, S. 등은 'Directed z-LDI'을 활용하여 섬과 효율적인 지지대 생성 위치를 찾는 방법을 제안했다[11]. 그들은 컴퓨터 그래픽스에서 널리 사용되는 시야 절두체(View Frustum) 기법 중 직교 시야 절두체(Orthogonal View Frustum)를 활용하였다. XY 격자를 Z축 방향으로 출력 모델에 투사하여 표면과 만나는 진입점(Entry Point)과 반대편으로 뚫고 나오는 출구점(Exit Point)을 얻어 섬을 식별하고, 각 점들의 관계를 분석하여 오버행 각도와 가중치를 적용하여 최적의 지지대 위치를 찾는 방법을 제안했다. Wang, Z. 등은 오버행 영역을 점, 모서리, 면으로 분류하여 법선 벡터의 각도에 따라 섬에 대한 정의를 내렸다[12]. 인접한 모든 두 면의 법선 벡터의 방향 중 하나라도 90°를 초과하면 오버행 점 또는 오버행 모서리로 간주된다. 그러나 이 방법은 XY 평면과 수직인 평면에 의해 생성된 단면을 통해 분석해야 하는 한계가 있다.

위 방법들은 픽셀 크기 또는 XY 평면과 수직인 평면들의 간격보다 작은 섬이 누락될 수 있는 단점이 있었다. 또한 출력 방향과 수직인 평면 외에도 다수의 평면 단면이 분석에 필요했다.

이 외에도 기존 탐색 방법을 응용하여 경제적이고 효과적인 지지대 생성에 최적화하는 연구가 많이 이루어져 왔지만, 정확하게 섬(Island)을 찾는 알고리즘을 제안한 연구는 찾기 힘들었다. 또한 많은 경우 여전히 작업자의 경험이 필요한 상수를 입력해줘야 하는 단점이 발생한다.

본 연구는 단면의 루프를 활용하여 해상도나 출력물의 방향이나 크기 상관없이 정밀한 섬을 탐색하는 알고리즘을 설명하고자 한다.

2.3 CPU와 GPU 탐색의 차이점

STL파일의 단면 슬라이싱 방법에는 크게 두 가지 방법으로 나눌 수 있다. CPU를 이용한 벡터 기반 슬라이싱과 GPU를 활용한 픽셀 기반 슬라이싱으로 구분할 수 있다[13,14]. 각 방법을 적용한 슬라이싱 결과와 연속된 두 레이어 간 비교 시 나타나는 특징을 정리하였다. Fig. 2를 통하여 안티엘리어싱이 적용되어 경계가 불명확한 이미지와 루프로 표현되어 경계가 명확한 이미지를 비교할 수 있다.

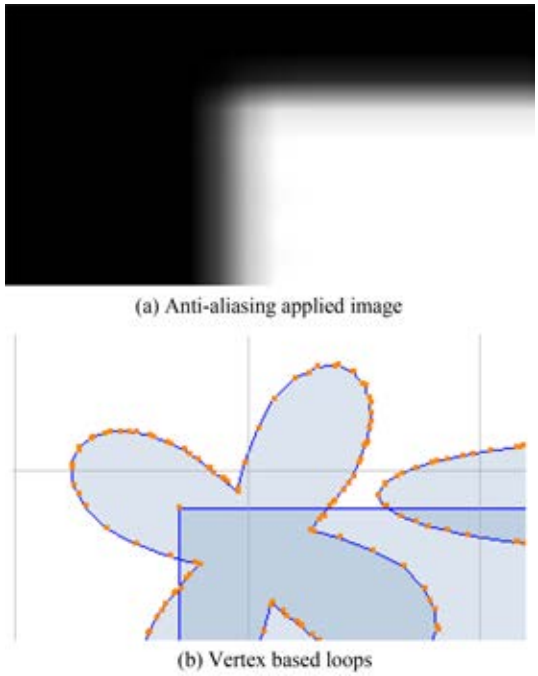


Fig. 2 Sliced layer images. (a)GPU-based method. The boundary is ambiguous due to gray pixels and (b) CPU-based method. The boundary is well-defined

CPU 기반 슬라이싱 결과로 얻어진 레이어의 비교 방법에 대한 주요 특징은 다음과 같다.

1) 벡터 연산을 기반으로 하여 이미지 처리가 불필요하다.
2) 픽셀 기반이 아니므로 해상도에 독립적이다.
3) 해상도의 영향을 받지 않아 XY 간격에 관계없이 정확한 섬 탐지가 가능하다.
4) 안티앨리어싱(Anti-aliasing) 적용 이전 단계에서 단면의 루프 분석이 가능하다.
5) 슬라이스 후 얻어지는 루프로부터 얻어진 정보는 단면 형상 편집이 용이하다.
6) 구조물의 영향 범위를 정밀하게 계산할 수 있다.
7) 지지대 생성 후 재검사가 용이하며, 지지대의 문제점도 예측 가능하다.
8) 출력물 이형 과정에서 가해지는 응력 분석에 유리하다.
9) 모든 삼각형을 모든 단면과 계산을 직접 해야 하는 단점이 있다.

반면, GPU 기반 슬라이싱 결과로 얻어진 레이어의 비교 방법에 대한 특징은 다음과 같다:

1) 벡터 데이터의 픽셀화 후 슬라이싱을 수행한다.
2) GPU 슬라이싱 후 단면 비교 분석은 이미지의 Vector화

작업이 필요하여 추가적인 연산 시간이 소요된다.

3) 안티앨리어싱 적용된 이미지 비교 분석의 기준이 모호해질 수 있다.

4) 지지대 생성 후 재검사 과정에서 슬라이싱이 요구된다.

5) 픽셀 간격보다 작은 섬은 탐지가 불가능할 수 있다.

6) 모델 방향 변경 시마다 GPU 슬라이싱을 재수행해야 한다.

CPU를 이용한 벡터 기반 슬라이싱은 출력 영역의 루프 정보만을 처리하므로 연산량이 상대적으로 적다. 반면, 이미지 기반 슬라이싱은 출력 면적 전체의 픽셀을 처리해야 하므로 불필요한 연산이 포함될 수 있다.

기존에는 CPU 기반 슬라이싱이 병렬 연산 적용이 어려워 처리 시간이 길다는 인식이 지배적이었다. 그러나 최신 컴퓨터의 연산 능력 향상으로 이러한 제약이 크게 완화되었다. 특히 다중 모델 출력을 준비하는 경우에도, 섬 구조 탐색 시 개별적으로 지지대 생성 단계에서 분석하므로 처리해야 할 데이터양은 크지 않다. 따라서 전체 처리 시간에 미치는 영향은 미미하다.

최종 출력용 안티앨리어싱 이미지 생성 단계에서만 GPU를 활용함으로써, 전체 출력 준비 과정의 효율성을 크게 높일 수 있다. 이는 CPU와 GPU의 장점을 적절히 조합하여 전체 프로세스의 성능을 최적화하는 접근 방식이다.

본 연구에서는 모든 단면에서 출력물의 형상이 하나로 합쳐지기 전과 후의 모든 루프를 활용하여 분석해야 하므로, CPU 기반 슬라이싱이 더 적합하다고 판단하였다.

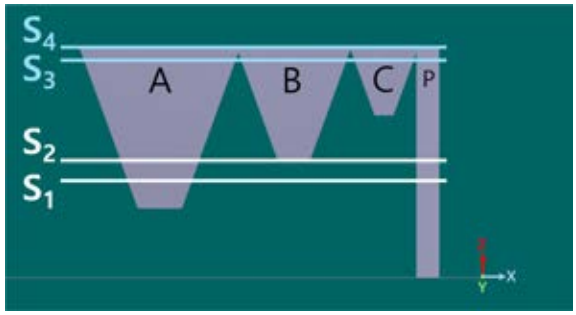


Fig. 3 A simplified model for island formation. A, B, C (square frustrum) and P(rectangular prism) represent part of a continuous landmass with three emergent island peaks

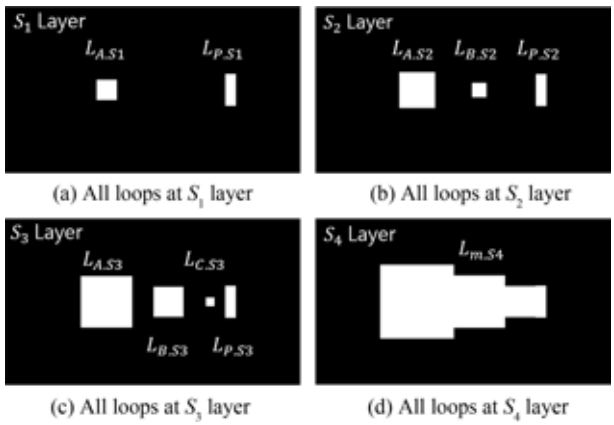


Fig. 4 S_2 layer exhibits the emergence of island loop($L_{B.S2}$).S_4 layer features merged shape($L_{m.S4}$), incorporating shape P

3. 필수 지지대 위치 탐색 알고리즘

3.1 탐색 알고리즘 준비

알고리즘 작동 원리를 설명하기 위해 Fig. 3과 같이 단순한 모형을 준비하였다. 상단 부분이 이어져 단일 부피 요소이지만 편의상 A, B, C, P로 부분을 나누어 설명하고자 한다. A, B, C는 단면이 정사각형인 사각뿔대이고 P는 직육면체 형상 기둥이고 출력이 시작되는 바닥에 위치하고 있다.

S_{n-1} 단면에는 없던 형상이 어디에도 연결되지 않은 상태로 S_n 단면에 나타나는 영역을 섬, 영어로는 Island 또는 Free-floating Area라고 불린다.

B의 단면 형상이 처음 나타나기 직전 높이의 평면을 S_1, B의 단면 형상이 처음 나타나는 높이의 평면 S_2로 한다. 모든 형상 A, B, C가 바닥부터 올라온 P 형상과 한 몸이 되는 높이의 평면을 S_4, 그 직전 평면을 S_3로 한다.

Fig. 4는 각 평면의 단면이며 각 형상의 루프와 관련 루프의 기호를 그림과 같이 정의한다. 오른손 법칙에 따라 법선 방향 기준으로 루프의 안과 밖이 구분되지만 본문에서 흰색을 루프 안쪽으로 정의한다.

3.2 탐색 알고리즘

특정 평면에서 형상이 나타났을 경우 이 부분이 섬인지 아닌지 판별하는 알고리즘은 Fig. 5와 같다. 각 항은 Clipper 라이브러리를 통해 루프 간의 불리언(Boolean) 연산을 통해 얻어진 결과이다.

탐색하고자 하는 형상이 준비되면 판별하고자 하는 형상에 알고리즘을 적용한다.

먼저 평면 S_2에서 형상 A의 단면 형상의 섬 여부를 판별하는 방법이다. 평면 S_2에서 형상 A에 해당하는 루프(Fig. 6(a))에서 평면 S_1에 있는 모든 형상의 루프(Fig. 6(b))와 겹치는 부분을 제거한 영역을 구하면 $RL_{A.S2}$(Residual Loop, Fig. 6(c)) 영역이 남게 된다. 다시 평면 S_2에서 형상 A 루프(Fig. 6(a))에서 $RL_{A.S2}$(Fig. 6(c))와 겹치는 영역을 제거한다. 이때 남아 있는 영역(Fig. 6(d), SR : Sustained Residual)이 존재한다면 이 영역과 연속적으로 연결되어 있는 상태를 의미한다. 출력 중인 몸체의 지지를 받고 있으므로 섬이 아닌 영역이다.

$$L_{A.S2} - L_{T.S1} = \mathrm{RL}_{A.S2},\ L_{A.S2} - RL_{A.S2} = SR$$

다음은 평면 S_2에서 B의 단면 형상의 섬 여부를 판별하는 방법이다.

$$L_{B.S2} - L_{T.S1} = \mathrm{RL}_{B.S2},\ L_{B.S2} - RL_{B.S2} = NR$$

위와 동일한 방법으로 마지막에 $RL_{B.S2}$(Fig. 7(c))과 겹치는

Algorithm : Island Detection Algorithm in Layered Planes

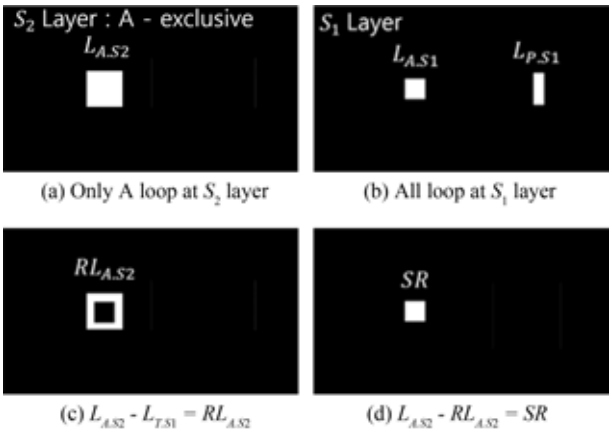
Input: $L_{(l,S_n)}$: Loop of shape l in plane S_n, where:
- l is one of the shapes: A, B, C, or P.
- S_n represents the n^{th} plane, where $n = 1, 2, \ldots,$ n.

$L_{(T,S_n)}$: Loops of all shapes in plane S_n.
$RL_{(l,S_n)}$: Residual Loop of shape l in plane S_n.

Output: Boolean value indicating whether each shape l is an island in plane S_n

foreach *Shape* l *in plane* S_n do
  Step 1: Calculate Residual Loop;
  $RL_{(l,S_n)} \leftarrow L_{(l,S_n)} - L_{(T,S_{n-1})}$ // Remove intersections with shapes in plane S_(n-1)
  Step 2: Calculate Remaining Area;
  Result $\leftarrow L_{(l,S_n)} - RL_{(l,S_n)}$;
  Step 3: Determine Island Status;
  if *Result has remaining parts* **then**
    Shape l is **not** an island;
  else
    Shape l is an island;
    while *Shape* l *remains an island in previous layers* **do**
      Move to next layer S_{n+1};
      Repeat Steps 1 and 2 for *Shape* l;
      if *Result has remaining parts* **then**
        Shape l is **not** an island anymore;
        break loop;

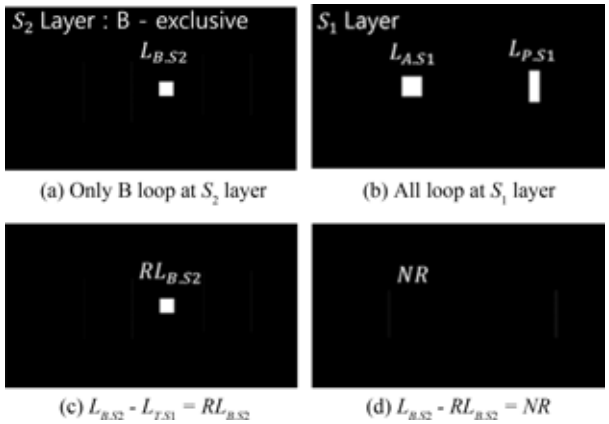
Fig. 5 Algorithm for detection of island



(a) Only A loop at S_2 layer (b) All loop at S_1 layer

(c) $L_{A,S2} - L_{T,S1} = RL_{A,S2}$ (d) $L_{A,S2} - RL_{A,S2} = SR$

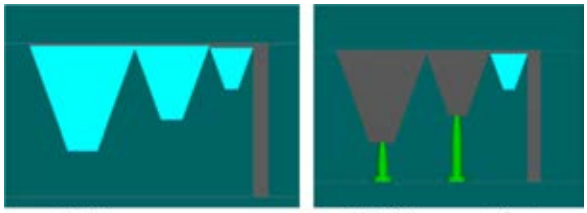
Fig. 6 Example to explain that cross section of the shape A is not an island at plane S_2



(a) Only B loop at S_2 layer (b) All loop at S_1 layer

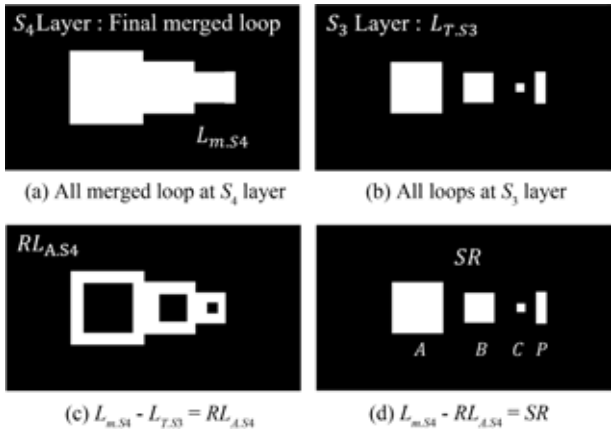
(c) $L_{B,S2} - L_{T,S1} = RL_{B,S2}$ (d) $L_{B,S2} - RL_{B,S2} = NR$

Fig. 7 Example to explain that cross section of the shape B is an island at plane S_2



(a) Without any support (b) With supports for two island areas.

Fig. 8 Visualization of unsupported regions



(a) All merged loop at S_4 layer (b) All loops at S_3 layer

(c) $L_{m,S4} - L_{T,S3} = RL_{A,S4}$ (d) $L_{m,S4} - RL_{A,S4} = SR$

Fig. 9 Example to explain that cross section of the merged shape ($L_{m,S4}$) is not an island anymore since plane S_4

영역을 제거하였을 때 아무것도 남지 않으면(Fig. 7(d), NR : Null Residual) 평면 S_2에서 형상 B는 어느 몸체에도 연결되지 않고 처음 나타난 영역을 의미한다. 이는 섬을 의미하며 필수적으로 지지대가 생성되어야 할 위치이다. 처음으로 섬으로 판별이 되면 파란색으로 표시하게 되고, 다른 섬이 아닌 영역과 만나 섬이 아니라는 결과가 나올 때까지 반복하게 된다.

실제 출력의 경우A, B, C 모형 위 알고리즘으로 연산하게 되면 모두 Island 형상으로 인식되며 최종적으로 바닥과 연결된 P 형상과 만나는 높이까지 Island로 Fig. 8과 같이 하늘색으로 표시하게 된다.

모든 형상이 만나 섬이었던 영역이 더 이상 섬 영역이 아니게 되는 높이의 평면을 S_4로 지정하고 섬영역이 끝나게 되는 상황에 대해 설명하고자 한다.

$L_{m,S4}$는 모든 형상이 합쳐진 루프를 의미하며 편의상 A, B, C, P로 구분 지었던 루프의 경계가 하나의 루프(Fig. 9(a))로 된다. S_3까지 섬이었던 A, B, C 형상이 평면 S_4에도 동일 위치에 있는 루프 $L_{m,S4}$와 단면 형상이 이어지게 된다. 이어진 $L_{m,S4}$의 섬 판별 연산 결과 SR (Fig. 9(d))의 잔여 형상이 존재하게 되며 섬이 아닌 영역으로 바뀌게 된다. 만약 이 잔여 영역이 계속 섬이었던 형상과 이어진 영역이었다면 섬 판별 알고리즘은 반복되며 여전히 하늘색 영역이 된다. 하지만 이 잔여 형상 SR (Fig. 9(d))은 바닥으로부터 이어져 올라온, 섬이 아니었던 형상 P와 연결되었기 때문에 판별 알고리즘은 완료된다.

본 알고리즘은 이미지 비교가 아닌 루프의 X, Y 값 연산을

통해 모든 섬 구조의 위치를 누락 없이 식별할 수 있다. 다만, 출력 장비의 적층 레이어 두께와 사용 재료의 특성을 고려하여, 실제 출력 결과에 유의미한 영향을 미치지 않는 미세 영역은 연산에서 제외할 수 있다. 단면 루프 이미지들은 개념의 시각적 이해를 돕기 위해 이미지로 표현되었으나, 본 알고리즘은 Clipper 라이브러리를 통해 슬라이스 된 루프의 X, Y 위치를 이용한 연산이다.

4. 실험

4.1 실험 장비

본 실험에 사용한 슬라이서 프로그램은 논문 저자가 직접 해당 알고리즘을 적용해서 개발한 'SMART SLICER'를 사용하였다. 작동 PC는 Intel Core i7-9700K CPU, RAM 32 GB를 사용하였다. 분석 단면 간격은 장비 출력 높이인 0.025 mm로 하였다. 출력은 광경화수지를 재료로 사용하는 에이치텍(HTEC)의 LCD 기반 MSLA 장비를 사용하였다. 출력 재료는 캐스터블 레진을 사용하였다. 3D 프린터 관련하여 널리 사용되는 스탠포드 모델들의 섬 영역 탐색을 진행하였다.

4.2 섬 영역 찾기

네 가지 스탠포드 모델의 높이, 삼각형 개수를 다양하게 하여 섬 탐색을 진행해 보았다. 높이와 삼각형 개수에 비례하여 탐색 시간이 증가하는 것을 확인하였다. 탐색 시간은 PC RAM 메모리 상태와 기타 프로그램 실행 상황의 영향으로 일정하지 않아 5번의 평균 시간을 기록하였다.

Fig. 10과 같이 섬 영역은 최초로 나타나는 단면에서 진한 파란색으로 표현되고 섬이 끝나는 높이까지 하늘색으로 표현되도록 하였다. 단면 간 넓이 변화를 시각적으로 파악하여 오버행의 정도를 판별할 수 있는 장점이 있다.

또한 모형의 크기가 증가함 따라 탐색 단면의 두께 보다 높아진 섬 위치를 정확하게 찾아낸 것을 확인할 수 있었다. 작은 모형 Fig. 10(e)에서는 무시 가능할 영역이었지만 부피가 커짐에 따라 Fig. 10(f)는 지지대가 필요한 형상으로 판단하였다.

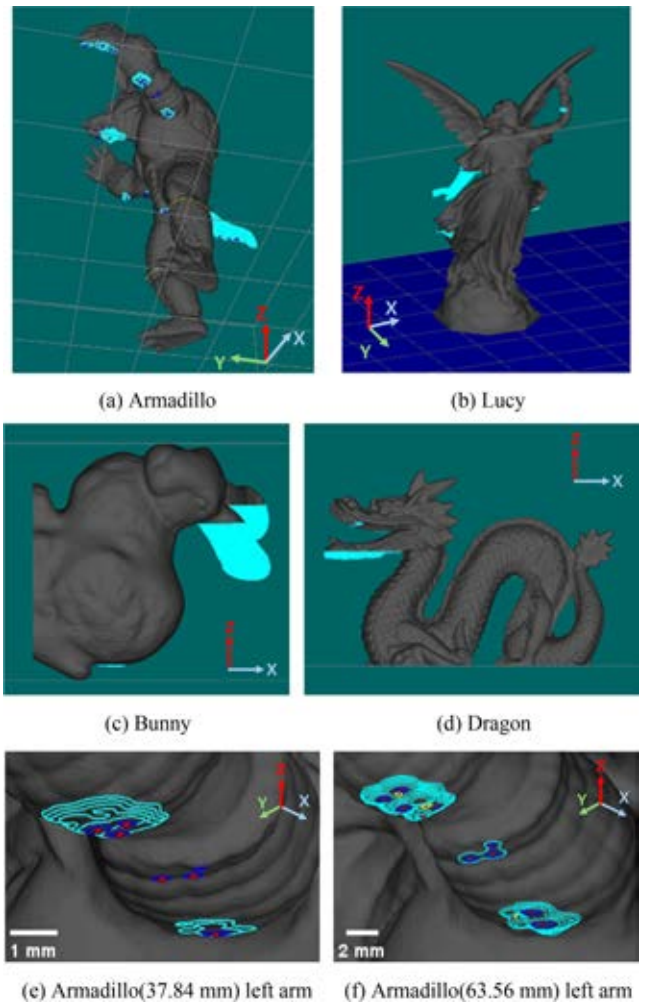
'Chitu Box'라는 타사 프로그램(GPU Base)과 섬 탐지 성능을 비교해보았다. Fig .11에서 'Chitu Box"는 섬영역을 진한 분홍색으로 표시하였으며, 법선 벡터와 Z축과의 각도가 45°가 넘는 표면 삼각형들은 빨간색으로 표시되었다. GPU Base 방식은 정확하게 필수 서포터가 들어가야 하는 부분을 보여주는 것이 아니고 대략적인 Overhang과 각도 조건에 근거해서 결과를 보여주는 것을 확인할 수 있다.

4.3 출력 결과

실험 결과 출력 실패 없이 성공적으로 출력물을 얻을 수 있었다. 후처리 공정인 출력 플레이트에서 분리할 때와 잔여 레진 세척 과정에서 출력된 서포터가 변형이 생길 수 있다.

Table 1 Model height, triangles and island detection time

Model	Height (*mm*)	Triangles	Time (*sec*)
Armadillo	37.84	345,944	14.8
		172,972	11.1
	63.56	345,944	22.2
		172,972	22.1
Dragon	36.87	318,460	14.9
		159,230	13.3
	67.92	318,460	26.5
		159,230	23.8
Lucy	36.73	701,939	15.5
		350,697	12.7
	67.08	701,939	26.6
		350,697	25.2
Bunny	36.68	112,402	10.5
		56,201	10.4
	67.97	112,402	20.3
		56,201	20.1

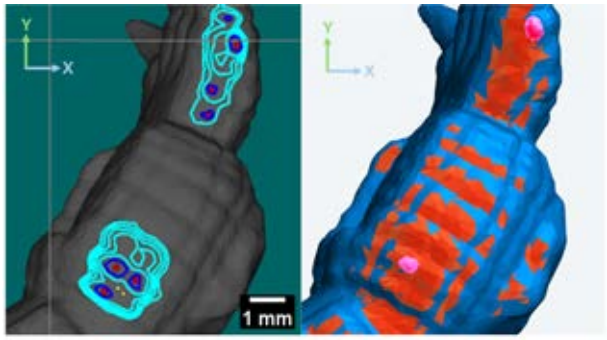


(a) Armadillo (b) Lucy

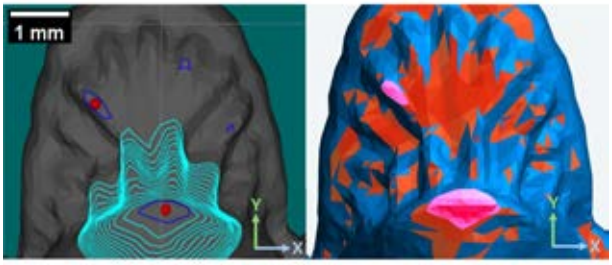
(c) Bunny (d) Dragon

(e) Armadillo(37.84 mm) left arm (f) Armadillo(63.56 mm) left arm

Fig. 10 Four stanford models and their differences beyond size



(a) Armadillo(37.84mm) left arm – left(SMART) / right(Chitu)



(b) Armadillo(37.84mm) chin – left(SMART) / right(Chitu)

Fig. 11 Comparison between CPU and GPU methods in island detection

표시된 필수 위치에 지지대 생성 후 재탐색 과정을 통해 누락된 섬이 없는지 확인하였다. 이 과정에서 잘못 생성된 지지대가 섬이 될 경우도 파악이 가능하다. Figs. 12(a) 및 12(c)와 같이 놓친 섬은 없는 것으로 확인되었다.

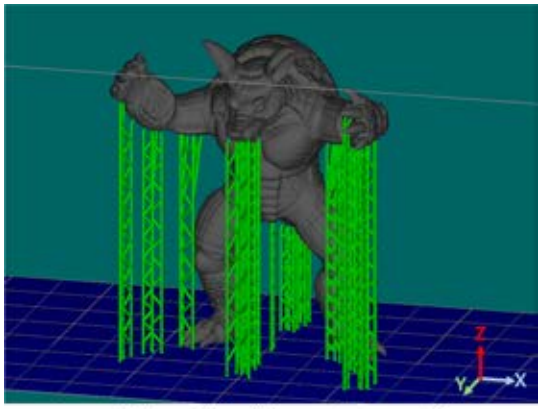
5. 결론

본 연구에서는 3D 프린터 출력 준비 과정의 핵심인 지지대 생성 위치 선정에 있어, 가장 중요한 요소인 섬 영역을 효과적으로 탐지하기 위한 CPU 기반 루프 탐색 알고리즘을 제안하였다. 제안된 알고리즘의 효용성은 실제 출력을 통해 검증되었다.

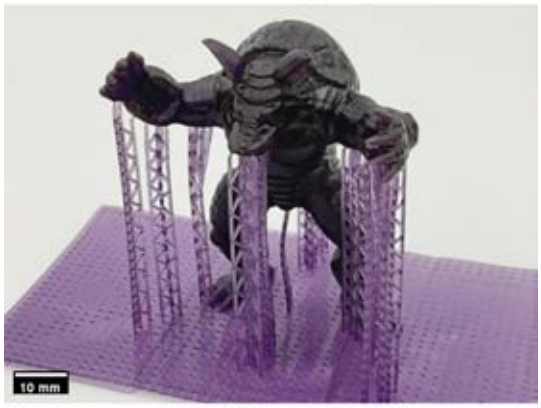
기존에 시간 소모적이라고 여겨졌던 연속된 두 단면 비교 방식을 개선하여, 이미지가 아닌 슬라이스 된 루프의 X, Y 좌표 비교 연산을 통해 신속하고 효율적인 결과를 얻을 수 있었다. 이는 지지대 생성 후 검토를 위한 반복적인 탐색에도 효과적인 것으로 나타났다.

현재 제안한 방법은 섬 영역만을 식별하며 오버행에 관한 정보는 제공하지 못하는 한계가 있다. 3D 프린팅에서 장비 특성과 재료 속성을 고려한 최적화된 자동 지지대 생성을 위해서는 오버행 영역의 정확한 탐지가 포함되어야 한다. 이를 위해 연속된 단면 간 루프 변화를 분석하는 후속 연구가 요구된다.

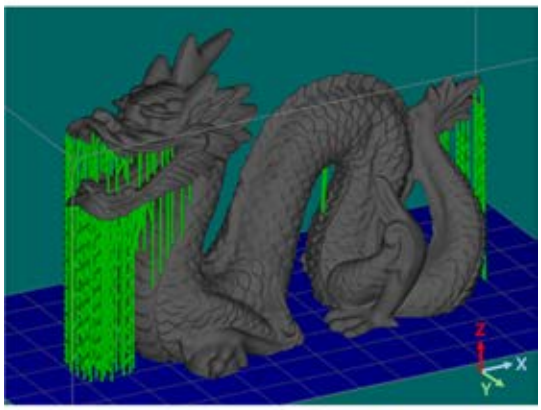
연속된 단면 간의 루프 형상과 면적의 변화를 정량적으로 분석하여 급격한 변화에 발생하는 문제점과 이에 대한 보완 방안을 마련하고자 한다. 또한 분석하는 레이어의 두께와 사용되는



(a) Armadillo with supporter generated



(b) Armadilo with supporters



(c) Dragon with supporter generated

(d) Dragon with supporters

Fig. 12 Results of armadillo and dragon 3D printing

재료의 특성을 손쉽게 반영하여 할 수 있도록 추가적인 변수들 간의 연관성에 대해 연구하고자 한다.

결론적으로, 본 연구에서 제안한 알고리즘은 3D 프린팅의 성공률을 높이는 데 기여할 것으로 기대되며, 향후 오버행 탐지 기능을 추가함으로써 더욱 완성도 높은 자동 지지대 생성 시스템으로 발전할 수 있을 것이다.

ACKNOWLEDGEMENT

이 연구는 서울과학기술대학교 교내연구비의 지원으로 수행되었습니다.

REFERENCES

1. Kawalkar, R., Dubey, H. K., Lokhande, S. P., (2022), A review for advancements in standardization for additive manufacturing, Materials Today: Proceedings, 50, 1983-1990.
2. Momeni, F., Mehdi Hassani. N, S., Liu, X., Ni, J., (2017), A review of 4D printing, Materials & Design, 122, 42-79.
3. Ministries, J., (2015), Policy to implement manufacturing innovation 3.0 strategy, 7th meeting to promote trade and investment in Korea.
4. Park, J.-M., (2015), Technology and issue on embodiment of smart factory in small-medium manufacturing business, The Journal of Korean Institute of Communications and Information Sciences, 40(12), 2491-2502.
5. Lee, I. H., Kim, H. C., Ahn, D. K., (2020), Korean terminologies for additive manufacturing according to the ISO/ASTM 52900 standard, Journal of the Korean Society for Precision Engineering, 37(12), 929-936.
6. Hur, J., Lee, K., (1996), Efficient algorithm for automatic support structure generation in layered manufacturing, International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 97621, V006T06A058.
7. Jiang, J., Xu, X., Stringer, J., (2018), Support structures for additive manufacturing: A review, Journal of Manufacturing and Materials Processing, 2(4), 64.
8. Hu, J., (2017), Study on STL-based slicing process for 3D printing, Proceedings of the 28th Annual International Solid Freeform Fabrication Symposium.
9. Vanek, J., Galicia, J. A. G., Benes, B., (2014), Clever support: Efficient support structure generation for digital fabrication, Computer graphics forum, 33(5), 117-125.
10. Chen, Y., Li, K., Qian, X., (2013), Direct geometry processing for telefabrication, Journal of computing and information science in engineering, 13(4), 041002.
11. Jang, S., Moon, B., Lee, K., (2020), Free-floating support structure generation, Computer-Aided Design, 128, 102908.
12. Wang, Z., Zhang, Y., Tan, S., Ding, L., Bernard, A., (2021), Support point determination for support structure design in additive manufacturing, Additive Manufacturing, 47, 102341.
13. Zhang, X., Xiong, G., Shen, Z., Zhao, Y., Guo, C., Dong, X., (2017), A GPU-based parallel slicer for 3D printing, 13th IEEE Conference on Automation Science and Engineering (CASE), 55-60.
14. Huang, P., Wang, C. C., Chen, Y., (2013), Algorithms for layered manufacturing in image space, Advances in Computers and Information in Engineering Research, 1.

Young Seok Kang
Ph.D. candidate in the Department of Mechanical Engineering, Seoul National University of Science and Technology. His research interest is additive manufacturing and device design.
E-mail: ehfy2k@gmail.com

Yeun Seop Kim
B.Sc. candidate in the Department of Mechanical Engineering, Seoul National University of Science and Technology. His research interest is Turbine engineering.
E-mail: dbstjq0408@gmail.com

Seung Chae Na
B.Sc. candidate in the Department of Mechanical Engineering, Seoul National University of Science and Technology. His research interest is biomimetics.
E-mail: theo2002@naver.com

Sang Jo Han
Professor in the Department of Mechanical Engineering, Seoul National University of Science and Technology. His research interest includes heat and mass transfer, thermal process design, gas turbines, and organic Rankine cycle systems.
E-mail: sjhan@seoultech.ac.kr