한국CDE학회 논문집
Korean Journal of Computational Design and Engineering

Vol. 21, No. 3, pp. 224-233. September 2016
DOI http://dx.doi.org/10.7315/CDE.2016.224

〈학술논문〉

pISSN 2508-4003
eISSN 2508-402X

FDM 3D 프린팅을 위한 Perlin 노이즈 기반 사전 시각화 기법

임재광 · 장승호 · 홍정모†
동국대학교 컴퓨터공학과

A Pre-Visualization Method for FDM 3D Printing Based on Perlin Noise

Jae-Gwang Lim, Seung-Ho Jang, and Jeong-Mo Hong†
Department of Computer Science and Engineering, Dongguk University

Received 11 September 2015; received in revised form 11 April 2016; accepted 11 April 2016

ABSTRACT

We propose a new method to visualize 3D models for FDM (Fused Deposition. Modeling) printing that appearance of the printed results can be predicted more realistically as that the efficiency of the modeling-printing process can be improved. The layered nature of horizontal slicing and the vibratory nozzle movements of customer-level FDM 3D printers leaving the characteristic patterns of noisy stripes on the surfaces of printed objects make difficulties in prediction of printed result in company with the thermal contraction of filament material. First, our method analyses the G-codes generated by common slicers to obtain proper outlines and take advantages of a modified version of Perlin noise based texturing method for rendering efficiency and enough number of control parameters on the visual details. The results show improved rendering details of pre-visualization of FDM printing.

Key Words: 3D Printing, FDM, Layered Manufacturing, Texturing, Visualization

1. 서 론

3D 프린터는 본래 기업에서 어떤 물건을 제품화하기 전에 시제품을 제작하기 위한 용도로 개발되었다. 1980년대 초반에 미국의 3D 시스템즈사(3D Systems)에서 플라스틱 액체를 굳혀 입체 물건을 만드는 프린터를 세계 최초로 개발하였다. 그 후 플라스틱 소재에 국한되었던 초기 단계에서 발전하여 나일론과 금속 소재로 범위가 확장 되었고, 산업용 3D 프린터뿐만 아니라 개인용 3D 프린터까지 등장하게 되었다.

최근 등장한 개인용 3D 프린터는 주로 FDM 방식을 따르고 있으며 이미 여러 제품이 출시되어 3D 프린팅의 대중화를 이끌고 있다. FDM 방식의 개인용 3D 프린터는 점점 발전하고 있지만 아직 상업용 3D 프린터만큼의 안정성을 갖추지는 못했다. 출력 도중에 출력물이 베드에 고정에 되지 않고 이탈하는 경우도 발생하며 외부 충격이나 3D 프린터 자체 진동에 의해 예상과는 다른 출력물이 나오기도 한다. 따라서 대부분의 FDM 방식의 3D 프린터의 출력물들은 모터에서 발생되는 진동과 노즐(Nozzle)에서 발생하는 열 그리고 온도 차에 의한 재료 수축 등으로 인해 불규칙한 가로줄 무

†Corresponding Author, jmhong@dongguk.edu
©2016 Society for Computational Design and Engineering

늬 패턴의 표면을 갖는다.

이러한 특유의 패턴은 미학적인 효과를 주고는 있지만 사용자는 3D 프린팅 출력이 완료 될 때까지 실제 출력물의 품질을 예측하기 어렵다는 문제가 있다. 3D 프린터의 노즐이 어떻게 움직여야 하는지를 명령하는 G-Code를 시각화하는 프로그램은 존재하지만 프린터 노즐의 경로를 확인하는 것만으로는 출력물 표면의 상세한 모습을 예측하기는 어렵다. 따라서 본 논문에서는 이러한 문제를 해결하고자 G-Code를 분석하고 이를 3D 모델에 적용하여 보다 사실적으로 시각화하는 기법을 소개하고자 한다(3장).

개인용 3D 프린터는 산업용 3D 프린터에 비해 정밀하지 못하며 더 많은 진동을 발생하기 때문에 출력물 표면에 특유의 불규칙한 적층 패턴이 나타나는 것이 특징이다. 따라서 본 논문에서는 출력물 표면 품질에 영향을 주는 FDM 3D 프린터의 특성과 이를 효율적으로 가시화 하는데 필요한 텍스쳐링 기술들을 소개하고(2장) 이를 응용하여 고유의 적층 패턴을 생성해 보도록 하겠다. 그리고 실제 3D 프린터로 출력된 결과와 시각화된 결과를 비교하여 얼마나 흡사한지 알아보도록 하겠다(4장). 마지막으로 본 연구의 결론과 한계 그리고 개선사항에 대해 알아보도록 한다(5장).

2. 관련연구

실제 물체를 제작하지 않고 사전에 그 결과를 예측할 수 있는 기술은 많은 연구가 진행되고 있다. Mizuno *et al.*[1]은 압력감지 펜을 이용하여 가상 공간에서 모델을 조각할 수 있는 기법을 구현하여 실시간으로 조각품을 확인할 수 있도록 하였으며, Mizuno *et al.*[2]에서는 나무재질의 물체를 조각하였을 때 결과를 예측할 수 있는 기술을 제안하였다. 또한 Mello *et al.*[3]은 입력 받은 이미지의 나무 조각품을 예측하여 시각화하는 기술을 제안하였다.

FDM 3D 프린터로 제작된 결과물의 표면에는 자연적으로 불규칙성이 나타나는 특징이 있다. 이러한 FDM의 특징에 대해서는 이미 많은 연구가 진행되어 왔다. Jun *et al.*[4] 그리고 Ha *et al.*[5]는 재료의 사출간격과 노즐의 경사가 출력물의 표면 거칠기에 미치는 영향에 대해 연구하였으며 그에 대한 향상 방법을 모색하였다. 또한 Jun *et al.*[6]에서는 FDM방식으로 생산된 제품의 표면에서 발생하는 계단 형상을 예측하는 방법을 연구한 바 있다. 자연적인 형상표현을 위하여 불규칙한 패턴이 불가피하게 필요한 경우도 있다. Perlin 노이즈[7,8]는 Gradient 노이즈 생성 기법의 일종으로 Gradient 벡터로 구성된 격자로부터 불규칙한 노이즈 값을 계산해 낸다. Perlin 노이즈에서는 일반적으로 사용자의 입력이나 또는 무작위로 구성된 값을 통해 구성된 Gradient 벡터장을 사용한다. 구름이나 안개 그리고 지형 등의 변화를 표현하기 위해 Gradient 벡터를 계산할 수 있으며 이를 통해 자연에서 발견되는 불규칙한 패턴을 생성할 수 있다[9-11]. Perlin 노이즈는 주로 2차원이나 3차원 형태로 사용되지만 1차원에서도 적용되어 횡스크롤(Side-Scrolling) 지형이나 필기체 모양의 라인(Handwritten Line)을 생성하는 등 다양하게 활용되고 있다. 본 연구에서도 3D 프린팅의 불규칙한 패턴을 표현하고자 Perlin 노이즈 기법을 사용하였고 Gradient 벡터장을 구성하기 위해 G-Code의 경로 벡터를 활용하였다.

G-Code는 가장 폭넓게 사용되고 있는 수치 제어 프로그래밍 언어로 G 프로그래밍 언어라고도 불리고 있다[12]. 주로 컴퓨터 기반의 공작기계(Machine tool)에 사용되고 있으며, 어느 위치로 움직이는지 또는 얼마나 빠르게 움직이는지 등의 제조 과정을 정의하는 언어라고 할 수 있다. 대부분의 개인용FDM 3D 프린터도 이 G-Code를 기반으로 기계를 제어하고 있으며 정의된 경로와 속도에 따라 재료를 적층하여 출력물을 완성한다.

G-Code는 여러 개의 명령어를 포함하고 있지만 3D 프린팅에서는 주로 G00와 G01명령 코드로 생성 경로를 제어하며 보통 명령코드 뒤에는 목표 좌표(위치)와 속도 값이 기술된다. G00명령의 경우는 필라멘트를 사출하지 않고 노즐를 해당 위치로만 이동시키고 G01명령은 목표 위치까지 필라멘트를 사출하며 이동시킨다. 일반적인 3D 프린터 사용자는 주로 슬라이싱(Slicing) 알고리즘을 통해 G-Code를 계산해 낸다. 슬라이싱 알고리즘은 어떠한 3D 모델을 여러 개의 레이어(Layer)로 잘라내고 각 레이어에서 적층경로를 계산한다. 대표적인 슬라이서(Slicer)로는 Cura[13], Kisslicer[14] 그리고 Sli3er[15]가 있으며, 3D 프린팅과 관련된 G-Code명령어들은 3D 프린터 오픈 소스 프로젝트인 RepRap(Replication Rapid Prototype)에서 상세히 설명되어 있다[16].

3D 프린팅 결과의 특징을 시각화하기 위해 우리는 텍스쳐를 사용하였다. 일반적으로 G-Code는 많은 경로 데이터로 구성이 되어 있기 때문에 그 경로들을 기하학적으로 표현하기에는 다소 무리가 따르며 다른 프로그램에 응용하기에도 효율적이지 못하다. 따라서 본 논문에서는 실시간 응용을 위하여 G-Code의 경로들을 텍스쳐로 반영하여 시각화하였다. 이미 많은 연구에서 텍스쳐를 활용해 다양한 시각적 효과를 보여 주고 있다. 특히 범프 맵핑(Bump Mapping)은 매우 보편적으로 사용되고 있는 기술 중에 하나이다. 텍스쳐를 사용해 물체 표면을 세밀하게 표현하는 기술로써 Blinn[17]에 의해 소개되어 현재까지도 게임이나 영화 분야 등에서 폭넓게 사용되고 있다. 노말 맵(Normal Map)기술은 범프 맵핑과 비슷하나 높이 값 대신 노말 값을 직접 텍스쳐에 저장한다는 차이점이 있다[18]. 우리는 이러한 텍스쳐 기술들을 이용하여 효율적으로 3D 프린팅 결과를 시각화 할 수 있었고 웹 환경에서도 실시간 렌더링이 가능하도록 하였다.

FDM 3D 프린터가 대중화됨에 따라 그에 맞춰 프린팅의 안정성에 관한 연구도 활발히 진행되고 있다. Jun *et al.*[19]은 FDM 3D 프린터에서 재료가 오류 없이 적층될 수 있는 조건과 방법에 대해 연구를 하였고 Jun *et al.*[20]에서는 프린팅 경로가 중첩될 때 발생하는 오류를 최소하는 방법을 제시하였다. 또 Jun *et al.*[21]는 3D 프린터의 해상도와 같은 수치가 결과물 표면 걸치기에 미치는 영향을 연구한 바 있다. 우리는 3D 프린팅과 관련된 연구가 앞으로도 활발히 진행될 것이라 예상한다. 따라서 3D 프린팅 시각화 기법이 단순히 응용 프로그램에서만 사용되는 것이 아니라 관련 연구 분야에서도 유용하게 사용될 것이라 생각된다.

3. 3D 프린팅 시각화 기법

본 논문에서 제안하는 시각화 기법은 3D 모델 데이터와 G-Code를 분석하여 텍스쳐를 생성하고 3D 프린팅 결과를 렌더링한다. Fig. 1은 3D 모델과 그와 관련된 G-Code를 입력 받아 최종적으로 3D 프린팅 결과를 가시화하는 시스템을 보여 주고 있다. 그림에서와 같이 크게 4단계로 나누어 작업이 진행된다. 1단계에서는 G-Code를 분석하여 모델의 외곽선에 해당하는 경로 세그먼트(Path Segment)들을 추출하게 되며, 2단계에서는 추출된 경로 세그먼트들에 Perlin 노이즈 값을 계산하여 부여한다. 3단계에서는 각 경로 세그먼트들과 인접한 메쉬들로부터 보간을 통해 경로 세그먼트들의 텍스쳐 좌표를 계산한다. 이렇게 텍스쳐 좌표 계산이 완료되면 4단계에서는 경로 세그먼트들이 위치한 텍스쳐 좌표로 노이즈 값을 반영하여 Bump Map 텍스쳐를 생성하고 최종적으로 렌더링을 처리하게 된다.

3D 프린팅 결과물의 시각적인 특징은 모델의 외곽선 근처를 지나가는 G-Code 경로에 의해 결정된다. 이러한 사실을 고려하여 본 연구에서는 메쉬 표면 근처에 존재하는 G-Code 경로들을 모델의 텍스쳐에 반영하여 3D 프린팅 결과를 시각화하였다. 3D 프린터는 필라멘트를 녹여 한층 한층 적층하는 방식으로 물체를 완성해 간다. 그렇기 때문에 Fig. 2처럼 출력물의 표면은 완벽하게 매끄럽지 못하며 각층마다 미세한 틈이 발생하게 된다. 우리는 이러한 형태를 Bump Map으로 표현하여 3D모델에 적용하였다.

G-Code의 경로를 분석하여 3D 프린터 노즐의

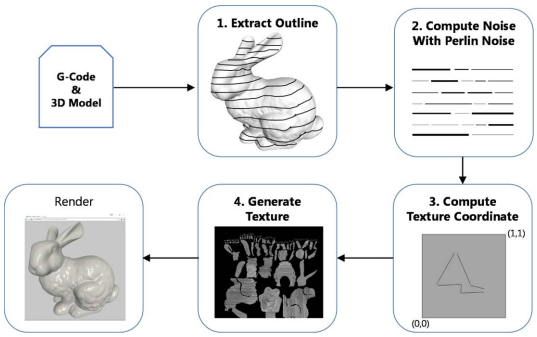


Fig. 1 The overview of our visualization system for FDM 3D printing

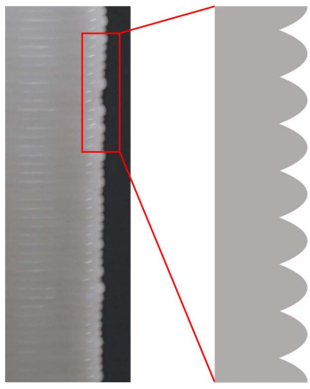


Fig. 2 The close-up picture of FDM printed object

움직임을 시각화하는 것은 복잡한 작업이 아니며 대부분의 슬라이서 프로그램이 지원하고 있다. 하지만 많은 수의 G-Code 경로를 이용하여 3D 모델의 출력 결과를 시각화하는 방법은 많은 컴퓨터 자원과 소요시간을 요구할 수 있다. 따라서 본 논문에서는 텍스쳐 기술을 적용하여 실시간 응용 프로그램에서 사용가능한 시각화 기술을 소개한다. 이 시각화 기법은 본래 3D 모델과 G-Code 분석으로 만들어진 텍스쳐를 이용하여 시각화가 이루어지기 때문에 실시간으로 결과를 확인할 수 있다. 따라서 본 논문에서는 기술의 활용도를 증명하고 개선하기 위해 WebGL을 기반으로 웹 환경에서 모든 예제들을 렌더링하였다.

3.1 G-Code 분석

앞서 언급한 바와 같이 3D 프린팅의 시각 결과물은 모델의 외곽선을 지나는 G-Code 경로에 의해 결정된다. Fig. 3은 3D모델의 슬라이싱된 G-Code 경로를 보여주고 있다. 그림에서 보이는 바와 같이 모델의 외곽 근처 외에 존재하는 경로가 다수 존재하기 때문에 우리는 G-Code 분석을 통해 외곽에 해당하는 경로를 분류해야 한다. 모델의 외곽 근처에 인접하는 경로를 찾기 위하여 G-Code에서 G00와 G01명령을 파싱(Parsing)하였고 각각 명령에서 위치 값을 추출하여 경로 세그먼트들을 계산하였다. 그리고 각각의 경로 세그먼트들이 외곽선에 해당하는지를 확인하기 위하여 모델을 구성하고 있는 메쉬와 교차 판별을 하였다. 메쉬의 양이 크면 교차 판별에 많은 부담이 되므로, 본 논문에서는 메쉬 정보를 사전에 몇 개의 레이어로 나눠서 관리하였다. 따라서 각 레이어에 해당하는 경로 세그먼트와 메쉬만 교차 판별을 함으로써 효율성을 높였다.

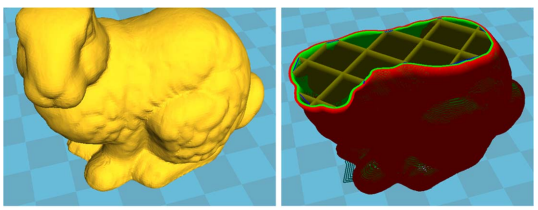


Fig. 3 The Standard Bunny 3D model and the G-Codes paths sliced and visualized with Cura slicer v14.03

3.2 Perlin 노이즈 생성

개인용 3D 프린터에서 발생한 진동은 출력 결과에 직접적으로 영향을 미치기 때문에 결과물 표면에는 진동에 의한 패턴이 발견된다. 이러한 요인에서 3D 프린팅 결과를 보다 정확하게 예측하기 위하여 표면의 패턴을 미리 시각화할 필요가 있다. 진동에 의해 생성된 패턴은 예측 불가한 노이즈라고 할 수 있으므로 우리는 많이 사용되고 있는 Perlin 노이즈 기법을 적용하여 모델의 텍스쳐를 생성하였다.

3D 프린팅과정은 3차원 공간에서 선을 그려 물체를 출력하는 과정이라 할 수 있다. 또한 Perlin 노이즈는 1차원 환경에서 사용되어 필기체 선을 표현하는데 사용되기도 한다. 따라서 3D 프린팅에서 만들어진 선을 Perlin 노이즈로 표현한다면 불규칙한 패턴을 가진 결과물을 가시화 할 수 있다. 단 3D 프린팅 과정에서는 여러 개의 선을 적층하여 물체를 완성하기 때문에 적층 패턴을 보이게 되는 특징이 있다. 이러한 특징을 반영하기 위해 우리는 2차원 형태의 격자로부터 Perlin 노이즈를 계산하기로 했다. 2차원의 벡터장을 구성하고, 3D 프린터가 경로 세그먼트를 적층하듯이 격자의 각 행에 경로 세그먼트를 배치하였다. 그러면 노이즈를 계산하는 과정에서 보간을 통하여 각 층의 경로가 간접적으로 영향을 받게 되어 적층 패턴을 나타내게 된다. Perlin 노이즈를 생성하기 위해선 우선 2차원의 Gradient 벡터장을 구성해야 한다. 앞서 설명한 바와 같이 Perlin 노

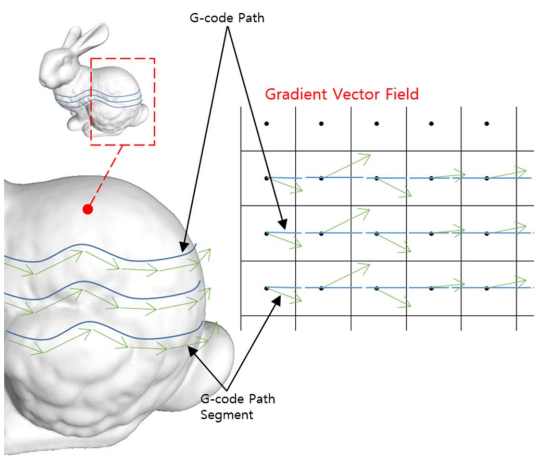


Fig. 4 Gradient vector field definition with G-code Path segments for Perlin Noise

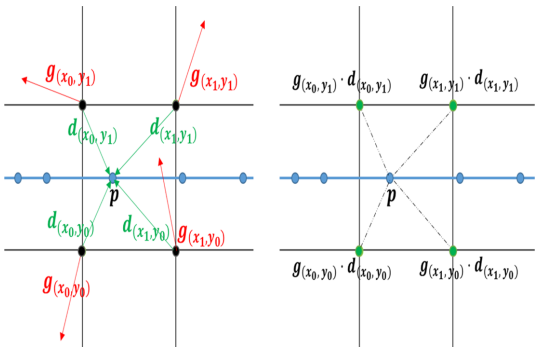


Fig. 5 Dot products of g and d vectors which calculate influence values at the position p from each node

이즈는 일반적으로 무작위로 생성된 값으로 Gradient 벡터장을 생성한다. 하지만 본 논문에서는 G-Code 데이터를 노이즈 생성에 반영하기 위하여 추출된 세그먼트들을 사용하여 2차원 Gradient 벡터장을 구성하였고 G-Code의 레이어 수에 맞춰 벡터장의 가로와 세로 해상도를 설정하였다. 앞서 모델의 외곽선에 인접하는 경로 세그먼트들이 추출되었다면 층별로 모델을 두르고 있는 선의 형태를 생각해 볼 수 있다. Fig. 4와 같이 그 선을 일직선을 펼쳐서 격자 각각의 행에 매칭하여 배치하였다고 가정해 보자. 이때 각각의 레이어에 존재하는 총 경로의 길이는 모두 다르기 때문에 가장 긴 총 경로의 길이를 기준으로 벡터장의 가로 길이에 맞게 각 경로의 길이를 조절하여 배치하였다. 그런 다음 경로 세그먼트 별로 경로의 벡터를 격자에 투영하여 Gradient 벡터장을 얻어낸다.

이렇게 경로 세그먼트들로부터 투영된 Gradient 벡터장은 Perlin 노이즈 기법의 보간법을 통해 노이즈 값을 계산하는데 다시 사용된다. Fig. 5은 Fig. 4과 마찬가지로 Gradient 벡터장안에 배치된 경로 세그먼트를 나타내고 있다. g는 각 노드에 저장된 Gradient 값을 의미하며, 파란 점은 경로 세그먼트들의 시작과 끝점을 의미한다. 벡터 d는 격자점에서 p까지의 거리를 나타낸다. Perlin 노이즈 기법에서는 g와 d 벡터의 내적 연산을 통해 격자점(녹색점)들의 영향력 값(스칼라 값)을 계산하고 보간법을 통해 최종적으로 경로 세그먼트의 시작과 끝 위치(p 위치)의 노이즈 값을 계산해 낸다.

3.3 FDM 적층 패턴

FDM 3D 프린터는 재료를 한층 씩 쌓아가는 방식으로 물체를 생산한다. 때문에 생산 과정에서 자연스럽게 이미 적층되어 있는 재료의 영향을 받는다. 고로 다른 레이어 영향을 고려해야 할 필요가 있지만 앞서 설명한 것처럼 경로 세그먼트 정보를 격자로 옮겨서 노이즈 값을 보간하는 방법을 이용한다면 간접적으로 다른 레이어의 영향을 받게 된다.

각각의 경로 세그먼트가 고유의 노이즈 값을 갖게 되었지만 편차가 큰 구간이 존재하기도 하기 때문에 자연스러운 패턴을 구성하기는 힘들다. 앞서 설명한 바와 같이 FDM 방식의 3D 프린터는 필라멘트(재료)를 분출하며 적층하는 방식으로 출력물을 생성한다. 대게 필라멘트를 연속적으로 사출하여 선을 그리듯이 적층하기 때문에 프린팅된 결과물의 표면에는 가로줄 무늬 패턴이 나타난다.

실제 출력물과 비슷한 가로줄 무늬의 패턴을 구현하기 위해 우리는 경로 세그먼트들에 저장된 노이즈 값을 한쪽 방향으로 전파하였다. 식 (1)와 같이 각 레이어 l 별로 $n-1$번째 경로 세그먼트의 노이즈 $v_0^{l,n-1}$ 을 가중치 α에 따라 $v_0^{l,n}$ 으로 전파하였다. 식 (1)에서 $v_0^{l,n}$ 는 경로 세그먼트의 시작점의 노이즈 값이며, $v_1^{l,n}$ 은 세그먼트의 끝점의 노이즈 값이다.

$$v_0^{l,n} = v_0^{l,n} \times (1-\alpha) + v_0^{l,n-1} \times \alpha$$
$$v_1^{l,n} = v_1^{l,n} \times (1-\alpha) + v_1^{l,n-1} \times \alpha \quad (1)$$

FDM 3D 프린터의 또 다른 특징은 일반적으로

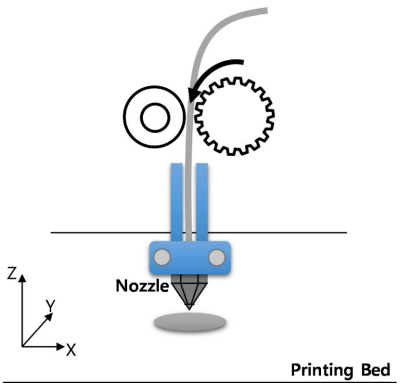


Fig. 6 The structure of printing nozzle which extrudes melted filament

X, Y보다 높이에 해당하는 Z 해상도가 높다는 것이다. Fig. 6과 같이 노즐의 열기에 녹아 사출되는 필라멘트는 아래로 흘러내리면서 자연스럽게 X, Y 평면으로 펴진 형태가 되기 때문에 Z의 정밀도가 더 높은 현상을 보인다. 그러므로 다음 장에 Bump Map 텍스쳐를 생성 할 때 이러한 특징을 고려해야 한다.

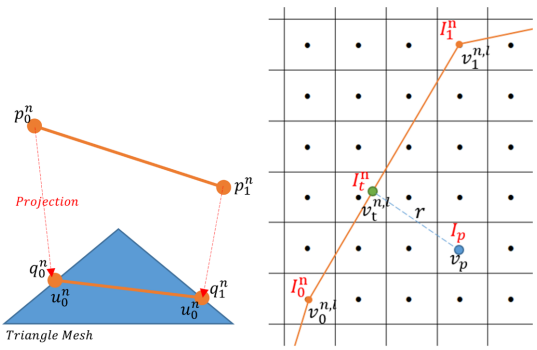
3.4 Bump Map텍스쳐 생성

앞서 모델의 외곽선에 해당하는 경로 세그먼트들의 Perlin 노이즈 값을 계산하였다면, 이제는 경로 세그먼트를 텍스쳐에 반영할 수 있다. 우선 경로 세그먼트가 위치한 텍스쳐 좌표를 계산하기 위해선 인접한 메쉬 표면으로 투영하여야 한다. 원래 경로 세그먼트의 위치가 p_0^n, p_1^n 이라 한다면, 투영된 위치는 q_0^n, q_1^n 라 가정하겠다. 그리고 투영된 위치에서의 UV 텍스쳐 좌표는 u_0^n, u_1^n 이라 하겠다.

Fig. 7과 같이 텍스쳐 이미지 상에 존재하는 경로 세그먼트는 주변 픽셀에 자신이 가지고 있는 노이즈 값을 반영한다. 식 (2)을 통해 우리는 경로 세그먼트가 텍스쳐 상에서 어느 위치에 있는지 알 수 있다. 텍스쳐의 넓이와 높이가 각각 *width*와 *height*라고 한다면 *n*번째 경로 세그먼트의 UV 좌표(u_0^n, u_1^n)를 텍스쳐 상에서의 위치(I_0^n, I_1^n)로 변환할 수 있다.

$$I_0^n = (u_{0x}^n \times width,\ u_{0y}^n \times height)$$
$$I_1^n = (u_{1x}^n \times width,\ u_{1y}^n \times height) \quad (2)$$

식 (3-4)에서는 값을 결정할 한 점 I_p과 가장 가까운 경로 세그먼트상에서의 위치 I_t^n 를 구하고 그 위치에서의 값 v_t^n 을 보간을 통해 계산한다.



(a) World Coordinates (b) Image Coordinates

Fig. 7 G-Code paths on world and image coordinates

$$w_d = (I_p - I_0^n)\cdot(I_1^n - I_0^n)/(I_1^n - I_0^n)\cdot(I_1^n - I_0^n)$$
$$w_d = clamp(0,1,w_d)$$
$$clamp(a,b,c) = \begin{cases} a, & if\ c<a \\ b, & if\ c>b \\ e, & else \end{cases} \quad (3)$$

$$I_t^n = I_0^n \times (1-w_d) + I_1^n \times w_d$$
$$v_t^n = v_0^n \times (1-w_d) + v_1^n \times w_d \quad (4)$$

3D 프린팅의 시각적 특징을 결정하는 것은 모델의 외곽선 부근을 지나치는 경로 세그먼트이다. 그러므로 메쉬 표면과 평행에 가까운 경로가 시각적으로 더 많은 영향을 미치게 된다. 따라서 식 (5)과 같이 원래 경로 세그먼트 방향과 투영된 방향의 내적 연산을 통해 영향력 w_q를 계산하였다. 식 (6)에서는 실제 텍스쳐에 반영되는 값을 결정하게 된다. *r*은 I_p과 경로 세그먼트의 유효거리를 나타내는 변수로서 적층되는 필라멘트의 두께를 결정한다. 앞서 사출된 필라멘트는 X, Y 평면으로 펴지는 성질이 있다고 언급하였다. 따라서 X, Y 평면과 수직인 방향에서는 적층된 필라멘트의 두께가 두껍게 보이게 된다. 이러한 이유에서 *r*을 결정하기 위해 식 (7)을 사용한다. Nor_{tri}은 투영되는 메쉬의 법선 벡터를 뜻하며 *r*은 0에서 4의 값을 갖도록 설정하였다. 고로 I_p를 기준으로 최대 4픽셀 반경에 있는 경로 세그먼트만 v_p에 영향을 미치게 된다.

$$w_q = (|p_1^n - p_0^n| \cdot |q_1^n - q_0^n|)^3$$
$$w_q = clamp(0,1,w_q) \quad (5)$$

$$w = clamp(0,1,1-h)$$
$$v_p = \frac{\Sigma_n^N (w \times v_t^n)(w \times w_q)}{\Sigma_n^N (w \times w_q)} \quad (6)$$

$$r = clamp(0,1,Nor_{tri}\cdot(0,0,1)) \times 4$$
$$h = \frac{\|I_p - I_t^n\|}{r} \quad (7)$$

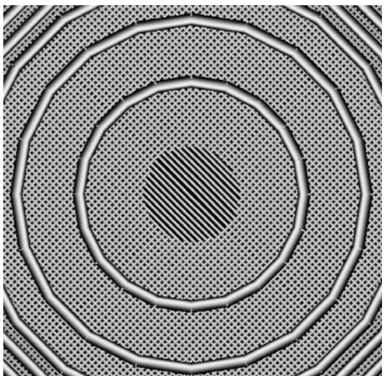


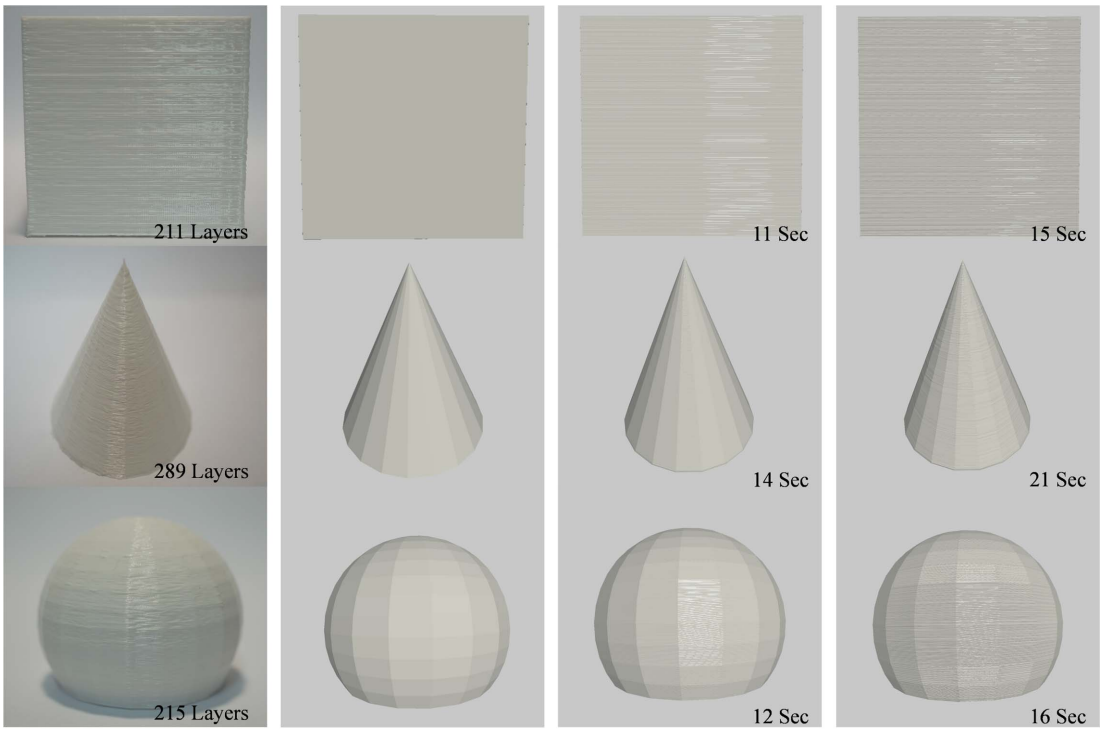
Fig. 8 Bump map texture generated by our method

4. 결 과

본 논문에서는 모델파일과 G-Code를 분석하여 해당 모델의 3D 프린팅된 결과를 시각화하는 시스템을 구현하였다. 모델의 G-Code 생성을 하기 위해서 Cura 슬라이서를 사용하였으며, 모든 예제의 출력물은 ALMOND 3D 프린터[22] 제품을 사용하였다. 또한 시각화된 결과물들은 응용프로그램으로 쉽게 확장 할 수 있도록 하기 위해서 WebGL 기반의 Three.js[23]를 통해 렌더링 되었다.

소개된 시각화 기법은 Fig. 8과 같은 텍스쳐 이미지를 구성하게 된다. Fig. 8은 G-Code의 경로를 반영하여 생성된 텍스쳐를 확대한 모습이다. 일반적으로 Cura 슬라이서는 각 층마다 외곽선 경로를 먼저 구성하고 안쪽을 채워 나가는 방식이기 때문에 그림에서도 등고선처럼 그려진 외곽선과 그 안쪽에 채워진 경로들을 살펴 볼 수 있다.

Fig. 9에서는 실제 3D 프린팅된 물체와 시각화 기법을 통해 구성된 결과물을 비교하였다. 또한 출력물의 레이어 수와 각각의 시각화에 소요된 시간을 기록하였다. 일반적으로 레이어의 수에 비례하여 시각화 시간은 증가 하며 본 논문의 시각화 기법은 Intel CPU E5-2697 환경에서 단일 쓰레드만으로 10개의 레이어를 처리하는데 대략 0.7초 정도의 시간을 소요했다. 우리의 시각화 기법은 Gaussian kernel을 사용한 경우 보다는 25% 정도의 성능저하를 보인다. 하지만 여러 도형 모델들



(a) Printed object (b) Normal (c) Gaussian kernel (d) Our method

Fig. 9 Comparison of actually printed objects (a) and rendering results. The Fig. indicate the number of layers of actually printed object and time taken to visualize by Gaussian kernel (c) and our method (d)

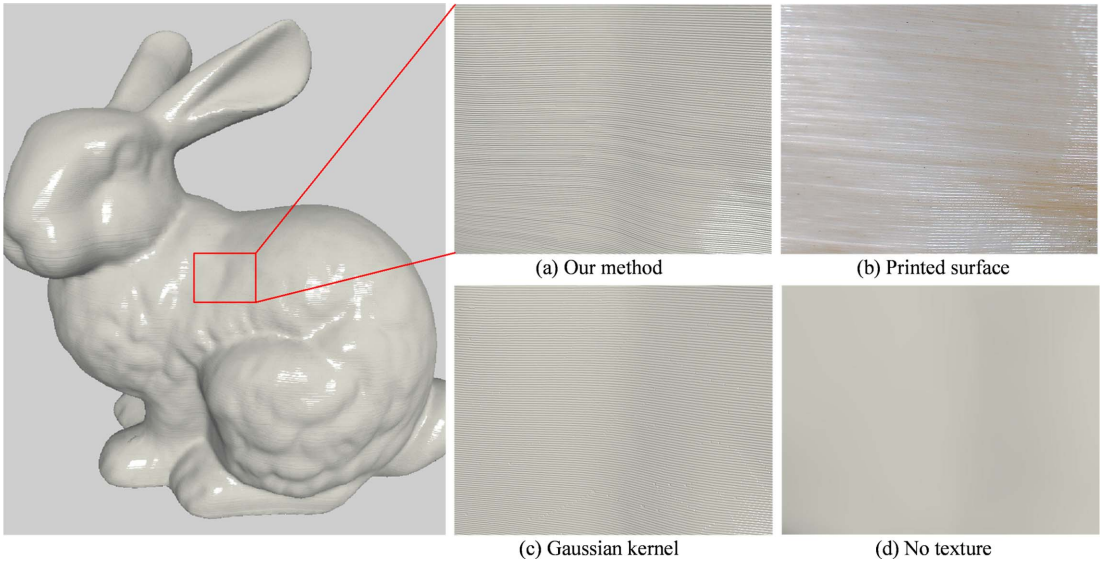


Fig. 10 Close-up view of the Stanford Bunny model visualized by our method

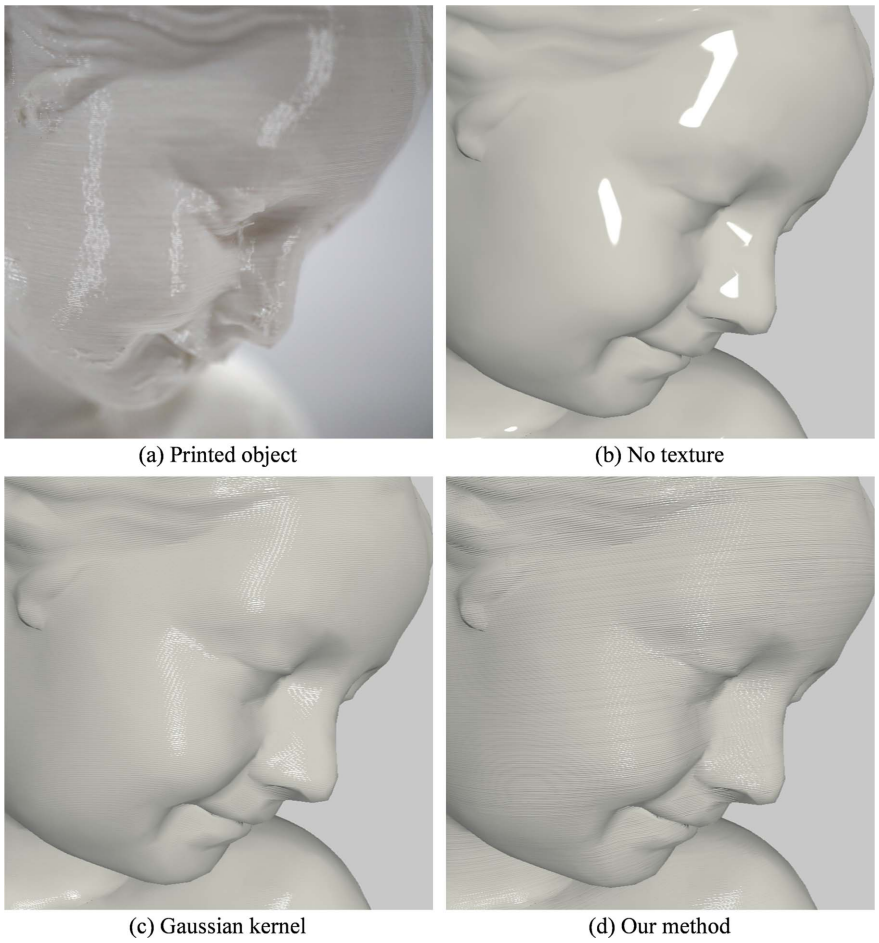


Fig. 11 Comparison of the photo of printed object and the rendered images

을 통해 비교해 보았을 때, Gaussian kernel을 사용한 경우 보다 본 논문의 시각화 기법이 보다 더 실제 출력물과 가까운 결과를 보여준다.

Fig. 10은 본 논문의 기법으로 생성된 결과(a)와 실제 출력물의 표면(b)을 각각 확대하여 비교하고 있다. Gaussian kernel을 사용할 경우(c)나 일반적인 렌더링 결과(d)와 비교하였을 때 우리의 기법이 실제 출력물에 더 근접한 결과를 보여준다. Gaussian kernel 함수를 이용할 경우에는 식 (8)과 같이 텍스쳐의 픽셀 값을 결정하였다.

$$v_p = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}h^2} \tag{8}$$

이와 같이 우리는 Gaussian kernel을 사용하여 단순한 형태의 G-Code 경로를 텍스쳐에 반영하였고 Perlin 노이즈를 이용한 시각화 기법과 비교하였다. Fig. 9과 Fig. 10에서 볼 수 있듯이 Perlin 노이즈가 실제 출력물 표면과 유사한 패턴을 생성하는데 영향을 미친다는 것을 확인 할 수 있다. 단순히 Gaussian kernel을 이용한 방법보다 Perlin 노이즈를 응용한 우리의 기법이 실제 출력물과 더 흡사한 결과를 보여준다.

Fig. 11은 최종적으로 복잡한 물체를 3D 프린팅한 출력물과 본 논문의 시각화 기법을 통한 결과를 비교한 예제이다. (a)는 실제 출력물을 나타내며 (b)는 본래 모델을 렌더링한 결과이다. 그리고 (c)는 Gaussian Kernel, (d)는 우리의 시각화 기법을 각각 적용한 결과이다.

5. 결론 및 향후 연구계획

본 연구에서는 G-Code에서 경로의 위치 값만을 추출하였고 텍스쳐를 생성하여 출력물을 시각화하였다. 따라서 3D 프린터의 여러 가지 옵션(속도, 사출량 등)에 따른 변화를 반영하지는 못했다. 또한 모델 표면에 인접한 경로를 계산하기 위하여 G-Code파일을 파싱하고 분석하는 과정이 요구되고 있지만 이는 슬라이싱 과정에서 쉽게 처리 될 수 있을 것이라 생각된다. 따라서 본 논문의 시각화 기법을 슬라이서에 응용한다면, 슬라이싱하는 과정에서 3D모델의 출력 결과를 미리 보기 할 수 있을 것으로 기대된다.

우리는 실시간 고속처리를 위하여 G-Code 경로를 직접 렌더링하지 않고 텍스쳐를 생성했다. 때문에 서포터(Supporter)의 형태를 시각화 할 수는 없었으므로 정확하게 3D 프린팅 결과를 시각화하기 위해선 서포터 경로까지 고려되어야 한다. 그리고 텍스쳐링을 이용한 방법은 표면에서 돌출된 G-Code 경로까지 시각화하지 못하기 때문에 외곽선 부분을 관찰 할 때 정확한 품질을 보여주지는 못한다. 추가적인 텍스쳐링과 쉐이딩 기술이 적용된다면 이런 문제를 해결 할 수 있을 것이라 된다. 또한 본 논문에서 사용된 Perlin 노이즈 기법은 3D 프린팅의 불규칙한 패턴을 시각화하기 위해 적용되었고 노이즈 값을 계산하는 과정에서 3D 프린터 하드웨어의 정량적 특성(모터의 회전 수 노즐의 온도 등)은 고려되지 않았다. 이런 특성을 고려하여 노이즈 값을 계산한다면 지금 보다 더 정밀한 시각화 결과가 나올 것으로 기대된다.

감사의 글

이 논문은 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음(R-20150217-000045, IoT 디바이스를 지원하는 HTML5 기반의 실감미디어 서비스 프레임워크기술 개발). 또한 2015년도 한국CAD/CAM학회 하계학술대회에서 발표한 논문을 수정 및 확장한 논문임.

References

1. Mizuno, S., Okadayy, M. and Toriwakiy, J., 1999, An Interactive Designing System with Virtual Sculpting and Virtual Woodcut Printing, *Computer Graphics Forum*, 18(3), pp.183-194.
2. Mizuno, S., Kobayashi, D., Okada, M., Toriwaki, J.I. and Yamamoto, S., 2003, Virtual Sculpting with a Pressure Sensitive Pen, *In ACM SIGGRAPH 2003 Sketches & Applications*, pp.1-1.
3. Mello, V.B., Jung, C.R. and Walter, M., 2007, Virtual Woodcuts from Images, *In Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia*, pp.103-109.
4. Jun, J.U., Jung, J.S., Kim, S.K., Kim, J.A., Kwon, H.J. and Ha, M.K., 2002, A Study in the Improvement in Surface Roughness of Rapid Prototype ad FDM, *The Korean Society of Mechanical Engineers*, pp.195-200.
5. Ha, M.K. and Jun, J.U., 2002, Influence of

Injection Amount Variation on Surface Roughness at FDM, *The Korean Society for Power System Engineering*, 6(2), pp.54-59.
6. Jun, J.U., Kwon, H.J., Kim, S.K., Kim, J.A., Jung, J.S. and Ha, M.K., 2002, Prediction of Sphere Surface by the Theoretical Area Error at FDM, *Korean Society for Precision Engineering*, pp.262-265.
7. Perlin, K., 2002, Improving Noise, *ACM Transactions on Graphics (TOG)*, 21(3), pp.681-682.
8. Perlin, K., 1985, An Image Synthesizer, *ACM Siggraph Computer Graphics*, 19(3), pp.287-296.
9. Cook, R.L. and DeRose, T., 2005, Wavelet Noise, *ACM Transactions on Graphics (TOG)*, 24(3), pp.803-811.
10. Elinas, P. and Stürzlinger, W., 2000, Real-time Rendering of 3d Clouds, *Journal of Graphics Tools*, 5(4), pp.33-45.
11. Pi, X., Song, J., Zeng, L. and Li, S., 2006, Procedural Terrain Detail Based on Patch-lod Algorithm, *In Technologies for E-Learning and Digital Entertainment*, pp.913-920.
12. Valentino, J. and Goldenberg, J., 2003. Introduction to Computer Numerical Control (CNC). Englewood Cliffs: Prentice Hall.
13. Cura User Manual-Ultimaker, https://ultimaker.com/download/170/Cura_User Manual_v1.0.pdf
14. KISSlicer Start Guide, http://www.kisslicer.com/files/KISSlicerQuickStart.pdf
15. Slic3r User Guide, http://lpfrg-latina.com/wp-content/uploads/2013/01/Slic3r_guide.pdf
16. RepRap G-Code, http://reprap.org/wiki/G-code/kr
17. Blinn, J.F., 1978, Simulation of Wrinkled Surfaces, *In Proceedings SIGGRAPH 78*, pp.286-292.
18. Kilgard, M.J., 2000, A Practical and Robust Bump-mapping Technique for Todays GPUs Game Developers Conference, Advanced OpenGL Game Development.
19. Jin, Y.A., He, Y. and Fu, J.Z., 2013, An Adaptive Tool Path Generation for Fused Deposition Modeling, *In Advanced Materials Research*, 819, pp.7-12.
20. Turner, B.N., Strong, R. and Gold, A.S., 2014, A Review of Melt Extrusion Additive Manufacturing Processes: I. Process Design and Modeling, *Rapid Prototyping Journal*, 20(3), pp.192-204.
21. Turner, B.N. and Gold, S.A., 2015, A Review of Melt Extrusion Additive Manufacturing Processes: II. Materials, Dimensional Accuracy, and Surface Roughness, *Rapid Prototyping Journal*, 21(3), pp.250-261.
22. ALMOND 3D Printer, http://opencreators.net/
23. three.js documentation, http://threejs.org/docs/

임 재 광

2012년 동국대학교 컴퓨터공학과 학사
2014년 동국대학교 컴퓨터공학과 석사
2016년 동국대학교 컴퓨터공학과 박사 수료
2016년~현재 (주)텍스터 스튜디어 R&D팀 연구원

홍 정 모

2000년 한국과학기술원 기계공학 학사
2002년 한국과학기술원 기계공학 석사
2005년 고려대학교 전산학 박사
2005년~2007년 Stanford University (USA) 박사후 연구원
2008년~현재 동국대학교 교수

장 승 호

2015년 동국대학교 컴퓨터공학과 학사
2015년~현재 동국대학교 컴퓨터공학과 석사과정