

Mixing chamber를 사용하여 그라데이션 색상을 구현하기 위한 G-code 생성기 개발

노경석 · 서해원 · 김태영 · 이용구[†]

광주과학기술원 기계공학부

Development of a G-Code Generator for Color Gradation Generations in a Mixing Chamber FDM 3D Printers

Kyoung-Seok Noh, Hae-Won Seo, Tae-Young Kim, and Yong-Gu Lee[†]

School of Mechanical Engineering, Gwangju Institute of Science and Technology (GIST)

Received 12 September 2016; received in revised form 18 October 2016; accepted 25 October 2016

ABSTRACT

The recent 3D printing industry has been focusing on developing 3D printers to fulfill the user's need to bring more colorful and realistic outcomes. Several 3D printers have deployed multiple extruders to print different colors. However, this method has a limit on its availability of colors. To solve this, recent research is focused on using mixing chambers to mix the possible colors to provide more color availability. In this paper, we discuss the environment and algorithm behind the development of G-code which allows a gradation effect of the mix of two filaments. The generation algorithm to make gradient G-code has been implemented in Cura using C ++ and Python.

Key Words: 3D Printing, Cura, G-code, Mixing chamber.

1. 서 론

3D 프린팅의 발전으로 인해 다양한 방법으로 3D 조형물을 출력할 수 있는 기술을 갖추게 되었다. 이에 3D 프린팅은 일반적인 기계 가공 분야뿐만 아니라 식품, 패션 등에도 적용되어 우리의 생활 전반적인 분야에 영향을 미치고 있다. 3D 프린팅은 재료의 종류와 적용하는 방식에 따라 다양한 기술이 존재하며, 이 가운데 대중들에게 널리 알려진 3D 프린팅 방법으로는 소결형의 SLS(Selective

Laser Sintering) 방식과 압출형의 FDM(Fused Deposition Modeling) 방식^[1,2]이 있다. 특히 FDM 방식의 경우 고체 상태의 필라멘트를 사용하여 출력을 하는데, 이 방식은 필라멘트가 녹아 노즐을 통해 압출되어 경화된 얇은 막을 쌓으므로 제작비용과 시간 면에서 효율적이며, 타 방식에 비해 장치의 구조와 프로세스가 간단하기 때문에 장비 가격과 유지보수 비용이 낮아 3D 프린팅 기술 대중화를 주도하고 있다.

또한 최근에는 FDM 방식의 3D 프린터의 기술이 점차 발전함에 따라 다수의 노즐을 배열하거나 mixing chamber를 이용하여 다양한 색상이나 재료를 출력^[3-5]할 수 있는 방법들이 등장했다. 이에

[†]Corresponding Author, lygu@gist.ac.kr

사용자는 자신이 원하는 색상으로 3D 조형물을 2차 가공없이 생산할 수 있게 되었다. 하지만 보편화된 FDM 3D 프린터 방식인 다중 노즐 방식으로는 필라멘트 고유의 색만을 독립적으로 출력할 수 있다는 단점이 있다.

따라서 본 연구에서는 색을 다양한 비율로 혼합 가능한 mixing chamber 방식의 FDM 3D 프린터를 사용하여 그라데이션 색상을 출력할 수 있는 G-code 생성기를 개발하였다. 일반적인 FDM 3D 프린터가 단색, 혹은 2가지 이상의 색상을 독립적으로 출력하는 반면, 본 연구에서 제안하는 슬라이서는 그라데이션 색상을 출력할 수 있게 지원함으로써 보다 더 사용자의 요구에 가까운 색을 결과물로 출력해낼 수 있다는 장점이 있다.

2. 선행 연구

컬러 3D 프린팅이 다양한 산업 분야에서 적용되면서 이에 관한 연구가 꾸준히 진행되고 있다. Reiner 등은 3D 프린팅으로 조형할 수 있는 물체에 두 가지 색상을 이용하여 텍스처를 맵핑하는 연구를 진행하였다. 이에 Reiner 등은 레이어의 색상을 서로 번갈아 지정하면서 tool path를 엇갈리게 생성할 수 있는 G-code 생성기를 개발하여 checkboard 형상의 맵핑 효과를 통해 두 가지 색상만으로 물체의 색상을 표현하였다^[6]. 그리고 Hergel 등은 다중 노즐 방식의 3D 프린터를 이용할 때, 융해된 필라멘트가 다른 색상 영역으로 침범하는 문제를 개선하기 위해서 다른 색상 영역으로 침범하는 tool path를 최소화 할 수 있는 G-code 생성기를 개발하는 방법에 대해 연구하였다^[7]. 또한 이미 다수의 3D 프린터 제작 회사에서는 다양한 색상을 표현하기 위해서 필라멘트를 섞어 사용하려는 움직임을 보이고 있다. 3D4C(Felix Chan) 프린터는 파랑(Cyan), 자주(Magenta), 노랑(Yellow), 검정(Key=Black)의 4가지 색상을 섞어서 사용함으로써 CMYK 색상을 이용하여 다양한 색상을 표현할 수 있는 3D 프린터를 생산하였다. 이외에도 M2030X(MakerPi)는 두 가지 색상에 대해 그라데이션 효과를 줄 수 있는 3D 프린터를 개발한 상태이다. 하지만 M2030X의 경우 자체 개발한 슬라이서를 GUI에 내장하여 암호화하는 방식으로 사용자의 접근을 제한하고 있다. 그리고 Builder의 3D 프린터는 Fig. 1과 같이 일반적인 슬라이서로 생

성한 G-code를 별도의 소프트웨어를 통해 한번 더 가공하여 그라데이션 색상을 출력할 수 있는 G-code를 생성한다. 하지만 이 방식은 모델 정보와는 무관하게 이미 생성되어 있는 G-code를 변환하여 사용하므로 슬라이싱 과정에서 모델 형상 정보를 이용할 수 없어 활용 가능성이 낮다.

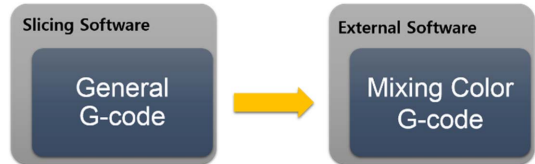


Fig. 1 The method of creating mixing color G-code in Builder

또한 현재 대부분의 컬러 3D 프린터의 경우 2가지 이상의 필라멘트 색을 섞어서 더욱 많은 색상을 표현한다. 하지만 색상을 섞어서 사용할 때, 프린터 공급자가 제공하는 필라멘트 공급 비율을 그대로 이용하므로 블렌딩 품질을 조절할 수 없어 사실상 사용자가 원하는 색상을 이용할 수 없는 실정이다. 따라서 본 논문에서는 자체적인 슬라이서 개발을 통해 두 가지 색상의 필라멘트를 이용하여 두 색상 사이의 그라데이션 효과를 적용할 수 있는 G-code 생성기에 대한 연구를 수행하였다.

3. 시스템 구성

3.1 Hardware

본 연구에서는 제안하는 그라데이션 G-code를

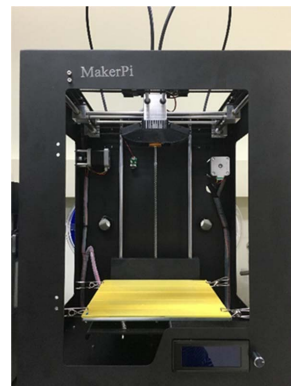


Fig. 2 M2030X(MakerPi) 3D printer with mixing chamber

개발하기 위해서 필라멘트를 섞을 수 있는 M2030 (MakerPi) 3D 프린터를 이용했다. Mixing chamber 를 장착한 이 모델은 두 개의 필라멘트를 주입하고 하나의 extruder를 통해 필라멘트를 섞어서 압출할 수 있는 기능을 갖추었다. 그리고 그라데이션 효과를 검증하기 위해 서로 다른 두 가지 색의 PLA 필라멘트(Print Temp. 190~220°C)를 사용하였다.

3.2 Software architecture

본 연구에서는 프린팅 헤드의 움직임을 조절하는 G-code를 생성하기 위해서 Cura의 슬라이싱 Engine과 GUI를 사용하였다. CuraEngine은 상용 CAD 소프트웨어를 사용하여 모델링한 파트를 레이어 단위로 슬라이싱 하여 G-code로 변환하는 역할을 한다. 본 연구에서는 STL 확장자의 CAD 파일을 모델로 사용하였다. CuraGUI에서는 사용자가 임의로 지정한 헤드, 플랫폼 온도와 같은 환경 설정 값이나 3D 프린터의 기기 정보 혹은 모델 파트의 위치 등을 설정할 수 있다.

CuraEngine과 CuraGUI는 Fig. 3와 같은 구조로 데이터를 전송한다.

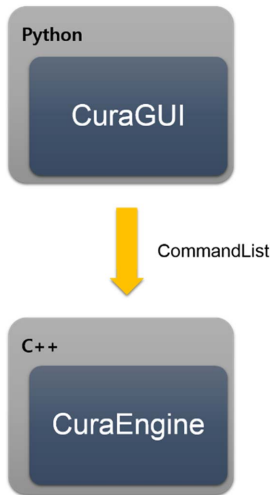


Fig. 3 The connection between CuraGUI and CuraEngine

Cura에서는 소켓 통신을 이용하여 STL 데이터와 프린팅 환경설정 값을 전달한다. CuraGUI에 사용자가 입력한 옵션 정보를 CuraEngine으로 전달하는 소켓 통신 방법은 다음과 같은 절차를 따른다.

- 1) CuraEngine과 CuraGUI 소켓 연결
- 2) Settings에 프린팅 설정 값 저장
- 3) Settings 정보를 commandList에 저장
- 4) commandList를 CuraEngine에 소켓 통신을 이용하여 전송

CuraGUI에서 입력된 프린팅 환경설정 값들은 commnadList로 저장되어 CuraEngine으로 전달된다. commandList에는 헤드의 프린팅 속도, 노즐의 온도와 같은 하드웨어 옵션을 비롯하여 컬러 출력과 관련된 컬러 모드, 그라데이션 비율과 같은 옵션도 포함되어 있다. 이때 전달하는 commandList의 구조는 Fig. 4과 같다.

...	...
[3]	'-s'
[4]	'initialSpeedupLayers=4'
[5]	'-s'
[6]	'minimalFeedrate=10'
...	...
[23]	'-s'
[24]	'colorA=0'
[25]	'-s'
[26]	'colorMixing=2'
[27]	'-s'
[28]	'colorB=100'
[29]	'-s'
...	...

Fig. 4 The information of commandList

Fig. 4과 같이 commandList를 작성하여 CuraEngine 프린팅 정보를 전달하고 나면 CuraEngine의 각 클래스들은 전달받은 정보를 토대로 G-code를 생성하게 된다. G-code 생성에 관여하는 클래스는 크게 3가지다. Fig. 5에서 확인할 수 있는 것처럼 G-code를 생성하기 위해서는 fffProcessor, settings, gcodeExport 클래스의 변수와 함수를 이용해야 한다. fffProcessor는 CuraGUI에서 받은 데이터들을 처리하여 실질적으로 G-code를 작성하는 클래스이다. Fig. 5와 같이 fffProcessor는 writeGcode() 함

수를 사용하여 G-code를 작성하게 된다. G-code의 현재 레이어 수를 의미하는 `currentLayer`와 총 레이어 개수를 의미하는 `totalLayers`은 CuraEngine에서 G-code를 생성하는데 필요한 변수이다. 그리고 `settings` 클래스는 G-code 작성이 필요한 환경 값을 설정하는 곳으로 `commandList`를 통해 전달 받은 프린팅에 필요한 환경설정 값을 입력 받고, 특히 그라데이션을 출력하기 위해 추가된 `colorA`, `colorB`, `colorMode` 변수를 제어하는 클래스이다. 전달된 값 중 `colorA`, `colorB`는 전체 모델의 높이를 100으로 보았을 때, 그라데이션의 시작과 끝의 높이를 나타내며, `colorMode` 변수 값을 이용하여 단일, 듀얼, 그라데이션 모드를 선택해서 출력할 수 있게 된다. 그리고 `gcodeExport`의 경우 실제로 그라데이션 G-code를 생성하기 위한 알고리즘을 내장하고 있는 부분으로 구체적인 프린터 헤드의 움직임과 필라멘트 압출량에 대한 G-code를 출력한다.

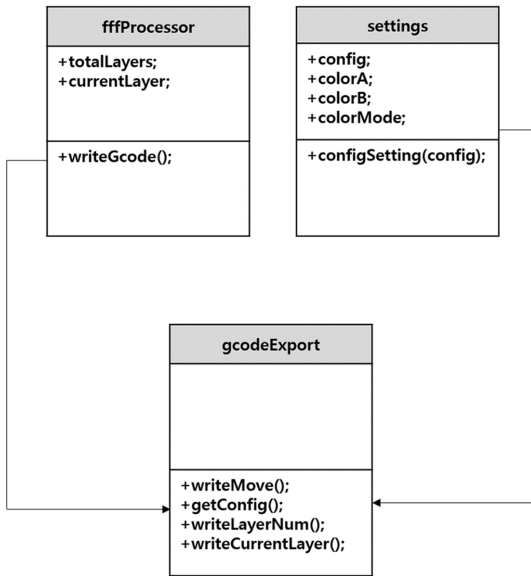


Fig. 5 Class of CuraEngine to make gradient G-code

Fig. 5에서 언급된 변수들이 각기 다른 클래스의 변수이기 때문에 그라데이션 G-Code를 생성하는 `gcodeExport`에서 사용하기 위해서 그라데이션 설정 값을 받기 위한 `getConfig()`와 총 레이어 개수를 전달받기 위한 `writeLayerNum()`, 현재 프린팅하고 있는 레이어의 수를 받기 위한 `writeCurrentLayer()`

를 `gcodeExport` 클래스에 생성한다.

4. 알고리즘

그라데이션 G-code는 일반 G-code와는 달리 두 가지 필라멘트 압출량을 고려하여야 한다. Fig. 6와 같이 단일 컬러 모드의 G-code를 의미하는 Fig. 6(a)와 그라데이션 컬러 모드의 G-code를 나타내는 Fig. 6(b)가 프린팅 헤드의 위치를 결정하는 X, Y의 값에서는 차이가 없다는 것을 확인할 수 있다. 하지만 두 컬러 모드는 필라멘트의 압출량을 표기하는 방법에서 차이가 있다. 단일모드인 Fig. 6(a)의 경우 필라멘트의 전체 압출량을 E로 나타내지만 그라데이션 컬러모드인 Fig. 6(b)의 경우 첫 번째 필라멘트 압출량은 E를 통해 표현되고 두 번째 필라멘트 압출량은 B로 나타낸다.

The position of printing head

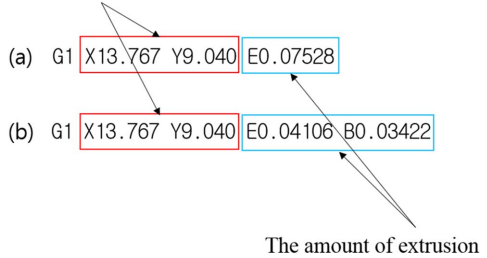


Fig. 6 The difference between one color and gradient color mode

본 논문에서 구현한 그라데이션 알고리즘의 FlowChart는 Fig. 7과 같다. 두 가지 색상의 필라멘트를 이용하는 이 알고리즘은 CuraGUI에 입력되는 `colorA`와 `colorB`의 백분율 값에 따라 모델에 그라데이션이 적용되는 부분을 조절할 수 있다.

알고리즘은 크게 세 부분으로 나눌 수 있다. 첫 번째는 현재 출력하고자 하는 레이어가 그라데이션 적용 구간보다 아래에 있을 경우이다. 이 때는 그라데이션 효과가 적용되지 않고 레이어가 높아질수록 ColorA의 압출량만 누적되어 출력하게 된다. 또한, 이 때 Color B 출력량은 0으로 유지된다.

두 번째 경우는 현재 출력하는 레이어가 그라데이션이 시작되는 레이어와 그라데이션이 끝나는 레이어 사이에 있을 경우이다. 이 때는 현재 레이어의 높이에 따라 첫 번째 필라멘트의 비율을 점

차 낮추고 그만큼 두 번째 필라멘트의 비율을 높이면서 G-code 생성하게 된다. 압출되어야 하는 총량에 현재 출력해야 하는 레이어에 해당하는 변화비율(delta)을 곱하여 첫 번째 필라멘트의 압출량을 구한다. 그리고 두 번째 필라멘트의 양은 압출되어야 하는 총량과 첫 번째 필라멘트의 양의 차로 나타낸다.

마지막 세 번째 경우는 현재 출력하고자 하는 레이어가 그라데이션이 끝나는 레이어보다 높을 때이다. 이 경우에 첫 번째 필라멘트의 압출량은 더 이상 증가하지 않고 유지하며, 두 번째 필라멘트의 압출량을 높이면서 G-code를 생성한다.

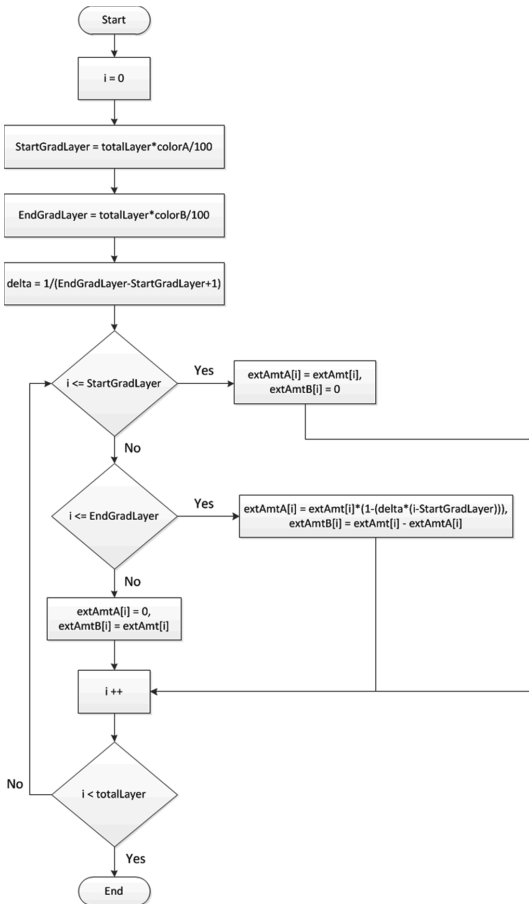


Fig. 7 Gradient color mode algorithm Flowchart

Fig. 7의 알고리즘의 세가지 경우를 그림으로 도시한 결과물은 Fig. 8과 같다. 알고리즘을 통해 설명한 바와 같이 첫 번째와 세 번째 경우에는 각각

ColorA와 ColorB만을 이용하여 출력을 진행하고, 두 번째 경우는 그라데이션 구간이 적용됨을 확인할 수 있다.

Case 1 : current Layer ≤ start Gradient Layer

Case 2 : start Gradient Layer < current Layer ≤ end Gradient Layer

Case 3 : end Gradient Layer < current Layer

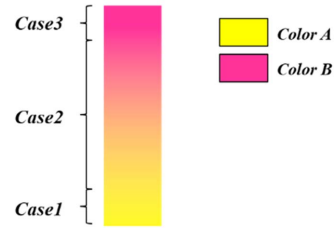


Fig. 8 The result of gradient algorithm in each case

Fig. 9는 CuraGUI에서 colorA에 0, colorB에 100을 입력했을 경우 결과 값이다. 압출량은 단일 컬러를 뽑을 때와 같은 양이지만 레이어가 증가함에 따라 colorA의 압출량은 줄어들고 colorB에 압출량은 늘어나서 파트의 처음 레이어부터 마지막 레이어까지 그라데이션 효과가 생길 수 있다.

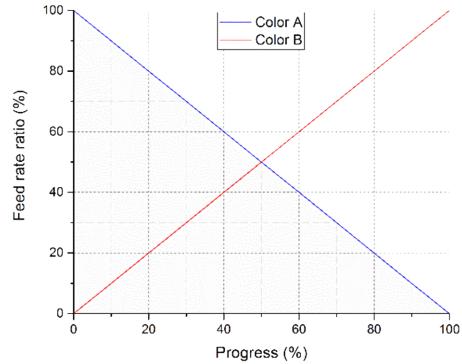


Fig. 9 Progress – Feed rate ratio graph at colorA = 0, colorB = 100

또한 Fig. 10의 경우는 colorA에 30, colorB에 70을 입력했을 경우이다. 이 경우는 앞선 Fig. 9의 경우와는 달리 전체 모델 높이의 0~30%는 colorA로만 출력을 하게 되고, 30~70%는 그라데이션 효과가 적용된다. 그리고 나머지 부분은 colorB로 출력을 진행하게 된다.

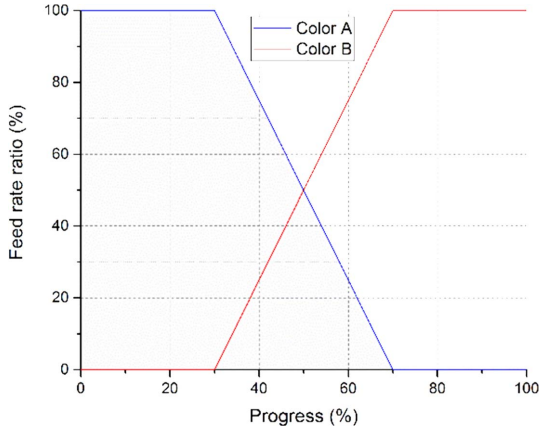


Fig. 10 Progress – Feed rate ratio graph at colorA = 30, colorB=70

5. 결 과

Fig. 11에 colorA와 colorB에 입력되는 값을 조절해가면서 뽑은 출력물이다. Fig. 11(a)는 colorA에 0, colorB에 100을 입력한 결과물이다. 그라데이션 적용 구간의 값이 0~100%까지 설정되었으므로, 출력된 결과물을 보게 되면 처음 레이어부터 마지막 레이어까지 그라데이션 효과가 생긴 것을 확인할 수 있다. 그리고 Fig. 11(b)는 colorA에 20, colorB에 80을 입력했을 때 결과물이다. 이 출력물은 파트의 20~80% 지점에 그라데이션이 생긴 것을 확인할 수 있으며, 그라데이션 효과가 생기지 않은 끝 단에는 각각 colorA와 colorB로 출력력이 진행된 것을 확인할 수 있다. 따라서 파트는 노란색 - 그라데이션 - 분홍색의 총 세 개의 층을 이루게 된다.

Fig. 11(c)는 colorA와 colorB에 각각 50을 입력한 결과이다. 이 모델은 그라데이션 적용 구간의 설정 값이 동일하다. 따라서 그라데이션이 나타나는 부분이 존재하지 않는다. 전체길이의 0~50%인 부분은 colorA에 설정된 필라멘트로 출력되고 50~100%인 부분은 colorB에 설정된 필라멘트로 출력되어 노란색 - 분홍색 총 2개의 층을 이루게 된다. Fig. 11(d)는 colorA에 80, colorB에 100을 입력한 결과이다. 이 모델은 전체 길이의 0~80%인 부분은 colorA에 설정된 필라멘트로 출력되고, 나머지 부분은 그라데이션이 나타나 노란색 - 그라데이션의 총 2개의 층을 이루게 된다.

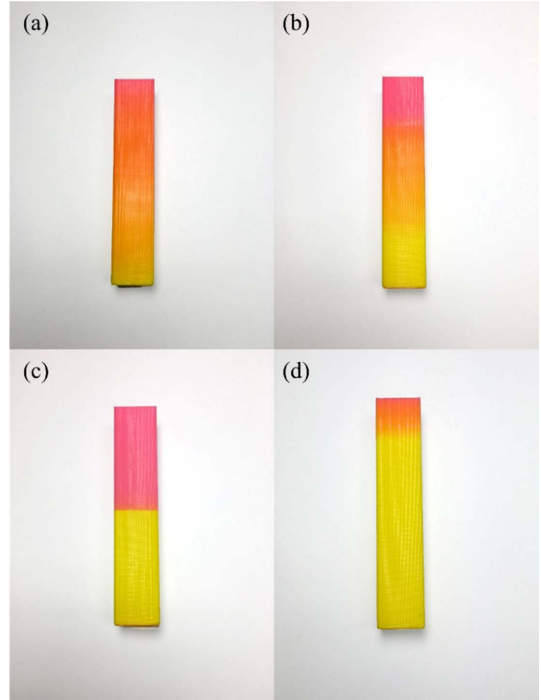


Fig. 11 The result of gradient effect

6. 고찰 및 향후 연구 계획

본 논문에서는 mixing chamber를 갖춘 3D 프린터를 이용하여 다양한 색상을 구현하기 위한 방법으로 두 색상 사이에 그라데이션을 구현할 수 있는 G-code 생성기에 관한 연구를 진행하였다. 이를 구현하기 위해 G-code를 이용하여 mixing chamber에 주입되는 필라멘트의 비율을 조절할 수 있는 알고리즘을 개발하였고, 실제 알고리즘을 이용하여 3D 프린팅 출력을 통해 결과물을 검증하였다. 이를 통해 필라멘트가 주입되는 비율을 G-code를 통해 제어할 수 있는 자체적인 그라데이션 슬라이서를 구현할 수 있었으며, 두 가지 색상 사이의 모든 색을 그라데이션을 통해 구현할 수 있었을 뿐만 아니라 그라데이션의 적용 위치까지 함께 제어할 수 있었다. 또한 알고리즘의 압출 비율을 설정하는 변수 값을 조절하여 각각의 사용 환경에 맞는 블렌딩 품질을 구현할 수 있을 것으로 예상된다. 그뿐만 아니라 본 연구에 적용된 알고리즘을 색상의 혼합이 아닌 재료 물성의 혼합에 적용한다면, 물성이 다른 두 개의 재료 사이에 준

제할 수 있는 혼합된 물성 값 또한 모두 구현할 수 있을 것으로 기대된다.

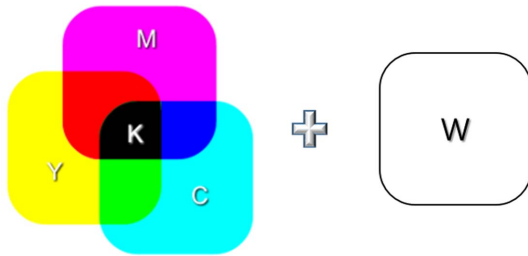


Fig. 12 CMYK + White : subtractive colors for printer

향후에 이를 발전시켜 Fig. 12과 같이 CMYK와 흰 색의 필라멘트를 적용할 수 있는 알고리즘을 구현할 계획이다. 3D 프린터는 종이에 출력하는 것이 아니므로 일반적인 2D 프린터와는 달리 CMYK에 흰색이 추가되어야만 모든 색상을 구현할 수 있다. 따라서 향후에는 본 알고리즘을 토대로 5개의 필라멘트를 공급 받을 수 있는 mixing chamber를 갖추고, 5개의 필라멘트 공급량을 조절할 수 있는 G-code생성기 개발에 초점을 맞춰 연구를 진행할 계획이다.

감사의 글

본 연구는 문화체육관광부 및 한국콘텐츠진흥원의 2015년도 문화기술연구개발지원사업과 [R2015120006, 콘텐츠형 온오프라인 3D프린팅 포

털/센터 구축을 위한 통합솔루션 개발] 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음[R0190-15-2040, 창의소재 활용이 가능한 3D 프린팅 콘텐츠형상관리 시뮬레이터 연구].

References

1. Chua, C.K., Leong, K.F. and Lim, C.S., 2003, Raping Prototyping: Principles and Applications, *Singapore: World Scientific*, doi: 10.1142/5064.
2. Gibson, I., Rosen, D.W. and Stucker, B., 2010, Additive Manufacturing Technologies, Springer, New York.
3. Reiner, T. *et al.*, 2014, Dual-Color Mixing for Fused Deposition Modeling Printers, *In Computer Graphics Forum*, 33(2), pp.479-486.
4. Masood, S.H. and Song, W.Q., 2004, Development of New Metal/Polymer Materials for Rapid Tooling Using Fused Deposition Modelling, *Materials & Design*, 25, pp.584-594.
5. Novakova - Marcincinova, L. and Novak-Marcincin, J., 2012, Testing of Materials for Rapid Prototyping Fused Deposition Modelling Technology, *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, 6(10), pp.2082-2085.
6. Hergel, J. and Lefebvre, S., 2014, Clean Color: Improving Multi-filament 3D Prints, *Computer Graphics Forum*, 33(2), pp.469-478.
7. Macy, B., 2011, Rapid/Affordable Composite Tooling Strategies Utilizing Fused Deposition Modeling, *SAMPE Journal*, 47(4), pp.37-44.



노 경 석

2015년 금오공과대학교 기계시스템
공학과 학사
2015년~현재 광주과학기술원 기계
공학부 석사과정
관심분야: Point cloud, 3D 프린팅,
4D 프린팅



서 해 원

2015년 광운대 컴퓨터소프트웨어
학과 학사
2015년~현재 광주과학기술원 기계
공학부 석사과정
관심분야: 3D프린팅, 4D프린팅,
컴퓨터 그래픽스



김 태 영

2015년 서울시립대 신소재공학과
학사
2016년~현재 광주과학기술원 기계
공학부 석사과정
관심분야: 3D 프린팅, 4D 프린팅



이 용 구

1992년 서울대 기계설계학과 공학사
1994년 서울대 기계설계학과 공학
석사(CAD)
1997년 서울대 기계설계학과 공학
박사(CAD)
1997년~2000년 삼성SDS 책임 연
구원
2000년~2003년 미국 NIST 객원
연구원
2010년~2011년 미국 NIST 객원 연
구원
2003년~현재 GIST 기계공학부 교수
관심분야: 3D 프린팅, 리버스 엔지
니어링, 포토닉스, CAD
