



Melhorias na geração do código G do algoritmo Hotfill

Enhancements in the GCode generation of the Hotfill algorithm

Rodrigo Yuske Yamauchi¹, Neri Volpato², Ricardo Dutra da Silva³,
Rodrigo Minetto⁴

RESUMO

A estratégia de preenchimento de manufatura aditiva denominada de *Hotfill* utiliza um algoritmo que preenche padrões zigue-zague para otimizar o resfriamento entre filamentos. No entanto, essa enfrentava problemas na geração de códigos de máquina (GCode), tais como falta de parâmetro de retração, ausência de pontos de contorno e ajuste impreciso do parâmetro de extrusão. Este trabalho apresenta aprimoramentos nessa estratégia para permitir a impressão e análise experimental da mesma. Assim, o algoritmo *Hotfill* foi atualizado, implementando-se códigos para inclusão dos pontos de contorno e parâmetros de retração/extrusão. Além disso, foram adicionados comandos para aquecimento e desligamento da plataforma de impressão. Os resultados demonstraram uma geração eficiente do GCode com a estratégia *Hotfill*, corrigindo as imprecisões anteriores e melhorando a qualidade das peças impressas.

PALAVRAS-CHAVE: extrusão com restrição de tempo; geração de código G; manufatura aditiva.

ABSTRACT

The additive manufacturing filling strategy called *Hotfill* uses an algorithm that fills zigzag patterns to optimize inter-filament cooling. However, it faced problems in the generation of machine codes (GCode), such as lack of retraction parameter, absence of points contour and inaccurate adjustment of the extrusion parameter. This work presents improvements in this strategy to allow printing and experimental analysis of the same. Thus, the *Hotfill* algorithm has been updated, implementing codes to include contour points and retraction/extrusion parameters. In addition, commands for heating and shutting down the printing platform were added. The results demonstrated an efficient generation of GCode with the *Hotfill* strategy, correcting inaccuracies and improving the quality of printed pieces.

KEYWORDS: Time-constrained extrusion; GCode generation; Additive manufacturing.

INTRODUÇÃO

A estratégia *Hotfill* (NAKONETCHNEI, 2022) envolve um algoritmo que procura um caminho para preencher em um padrão zigue-zague na tecnologia de extrusão de material (VOLPATO, 2017), de modo a garantir que o tempo de resfriamento entre as deposições de filamentos adjacentes permaneça abaixo de limites de tempo definidos pelo usuário.

No entanto, ao se tratar do código G (linguagem padrão para a maioria das impressoras 3D) gerado pela estratégia, não era possível utilizá-la eficientemente na impressão de peças uma vez

¹ Bolsista da UTFPR. Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil. E-mail: rodrigoyuske@alunos.utfpr.edu.br. ID Lattes: 3721894179909007.

² Docente no DAMEC. Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil. E-mail: nvolpato@utfpr.edu.br. ID Lattes: 8414652619232683.

³ Docente no DAINF. Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil. E-mail: rdsilva@utfpr.edu.br. ID Lattes: 8512085741397097.

⁴ Orientador, Docente no DAINF. Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil. E-mail: rminetto@utfpr.edu.br. ID Lattes: 8366112479020867.



que o algoritmo necessitava de ajustes finos. As principais imprecisões observadas eram: falta de parâmetro de retração para os *jumps* (pulos) realizados entre os trechos de *raster*, falta dos pontos de contorno e cálculo impreciso do parâmetro de extrusão de material (E).

Assim, atualizou-se o algoritmo *Hotfill* para funcionar com as últimas versões do software RP3 (Rapid Prototyping Process Planning), que é um software de planejamento de processos de manufatura aditiva desenvolvido pelo NUFER (Núcleo de Manufatura Aditiva e Ferramental) da UTFPR-CT. Este software é responsável por calcular os pontos a serem usados pelo algoritmo *Hotfill*, o qual reordena totalmente esses pontos. Este trabalho apresenta as melhorias implementadas.

MÉTODOS

O *Hotfill* foi implementado em linguagem *Python*, baseado na técnica de programação dinâmica (NAKONETCHNEI, 2022), e como um projeto dependente do RP3, que é responsável por fornecer os pontos calculados com padrão de preenchimento zigue-zague a serem então reordenados pelo algoritmo *Hotfill*.

Anteriormente, o código GCode gerado com esse algoritmo não possuía contornos para as peças a serem produzidas, possuíam desvios no cálculo do parâmetro de extrusão E (próprio do GCode e que indica a quantidade de material a ser extrudado pelo bico), e não considerava a retração do bico extrusor para os saltos feitos entre trechos de *raster*, o que implicava em imperfeições nas peças nas regiões das trajetórias desses saltos e inutilizava as impressões feitas com a técnica.

A fim de contornar as imprecisões apontadas anteriormente, buscou-se refazer a função de cálculo do parâmetro E , com base no cálculo feito pelo RP3. Assim, para cada par de pontos em um mesmo trecho de *raster* soma-se, a um valor acumulativo de extrusão, o valor calculado pela função de cálculo do parâmetro E . A fórmula para calcular a saída dessa função é dada pela equação 1.

$$E = \frac{V}{A}, \quad (1)$$

onde A é a área da seção transversal do filamento de alimentação utilizado, dada pela equação 2:

$$A = \pi/4 \cdot D^2, \quad (2)$$

onde D = diâmetro do filamento, e V é o volume de material necessário a ser extrudado, assumindo que a seção transversal do material extrudado seja um retângulo de dimensões Z e $(W - Z)$ somado com dois semicírculos de diâmetro Z . Esse volume é dado pela equação 3:

$$V = L \cdot (|W - Z| \cdot Z + \pi/4 \cdot Z^2), \quad (3)$$

onde L = distância entre o par de pontos referente a trajetória de deposição, W = largura do filamento extrudado de *raster*, e Z = espessura da camada. Com o cálculo do parâmetro E feito, para adicionar ao código G as trajetórias de contorno da peça, bastou incluir, no início de cada camada, o código respectivo dos pontos de contornos já calculados pelo RP3.

Como em cada salto entre trechos de *raster* em uma camada não deve haver a extrusão de material, é necessário que haja retração de material antes de começar o salto e, então, retornar a



extrusão após finalizado. Assim, foi adicionado o parâmetro de retração/extrusão E nos saltos entre trechos de *raster*. Isso se aplica tanto para o preenchimento com padrão zigue-zague quanto para os contornos de cada camada. Dado um valor de retração definido pelo usuário, no início de cada salto é feita a redução do parâmetro E acumulativo. Uma vez realizado o salto, a mesma quantidade de material retraída é extrudada (somando novamente ao parâmetro E).

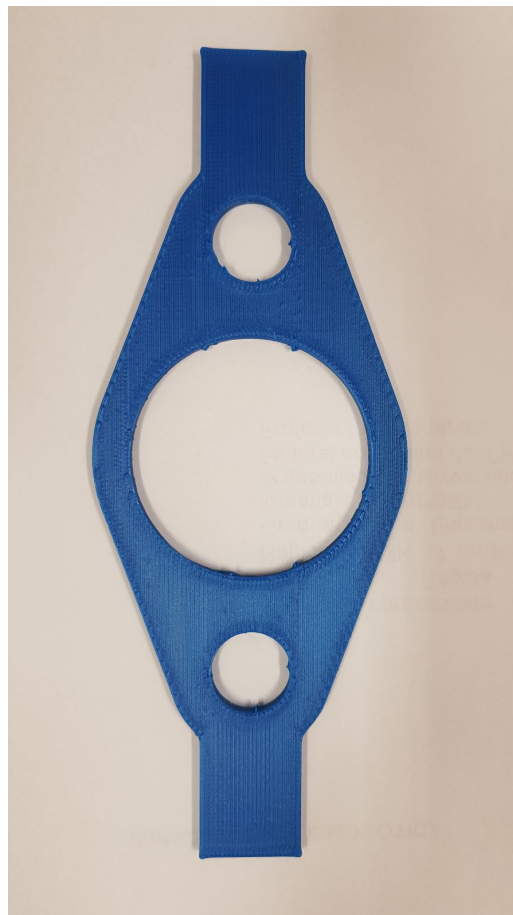
Além disso, como as máquinas trabalhadas pelo NUFER utilizam a plataforma aquecida, adicionaram-se os comandos no GCode para opcionalmente aquecer a plataforma antes da impressão e para desligar o aquecimento na finalização da impressão.

Por fim, anteriormente o código G gerado era configurado para coordenadas relativas para o filamento, o que causava distorções perceptíveis nas bordas com extrusão consideravelmente maior que a desejada. Propôs-se, assim, a mudança para coordenadas absolutas para o filamento, a qual corrigiu o problema.

RESULTADOS

A fim de comparar o método *Hotfill* (atualizado) com o padrão de impressão zigue-zague do RP3, tem-se a impressão da figura 1 sem a aplicação do algoritmo *Hotfill*.

Figura 1 – Impressão de padrão zigue-zague sem *Hotfill* (i.e., sem otimizações para evitar resfriamento excessivo na peça)

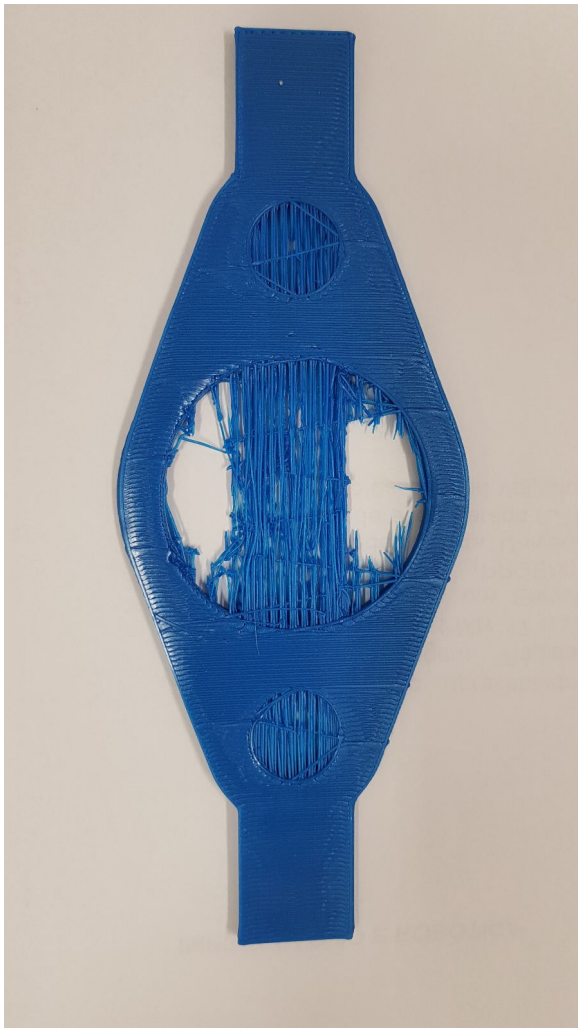


Fonte: Elaborado pelos autores (2022).



A figura 2a mostra uma peça impressa com o algoritmo *Hotfill* previamente às melhorias apontadas neste trabalho, tal que é possível notar diversos depósitos indevidos de filamento em situações que o algoritmo *Hotfill* realiza saltos para cumprir o tempo máximo entre deposições consecutivas. Notam-se imprecisões grosseiras, perceptíveis principalmente nos furos, em saltos entre trechos de *raster* devido à falta do parâmetro de retração e extrusão na operação de salto. Além disso, o código G dos contornos foi inserido manualmente, uma vez que não era feito durante a geração do código. Já a figura 2b mostra a mesma peça feita com os aprimoramentos apontados neste documento.

(a) Impressão com *Hotfill* sem aprimoramentos



Fonte: Elaborado pelos autores (2022).

(b) Impressão com *Hotfill* com aprimoramentos



Fonte: Elaborado pelos autores (2023).

Uma vez que a maioria dos saltos feitos utilizando o algoritmo *Hotfill* ocorrem no furo maior para essa peça, a figura 3 mostra um resultado satisfatório, sem as imprecisões notáveis nos saltos.



Figura 3 – Furo principal da impressão com *Hotfill* com aprimoramentos



Fonte: Elaborado pelos autores (2023).

CONCLUSÕES

Com as melhorias realizadas, foi possível alcançar a geração de código G funcional para o algoritmo *Hotfill*. Apesar de existirem pequenas imprecisões notáveis nos furos, ao comparar com o código GCode ordinário do RP3, sem a estratégia *Hotfill*, notam-se imprecisões similares, as quais provavelmente advêm da própria máquina de impressão utilizada.

Entende-se que essa pesquisa realizou com sucesso a geração eficiente de código G com o algoritmo *Hotfill* e possibilitará a impressão real de peças e, conseqüentemente, comparações práticas com outras estratégias.

Agradecimentos

Aos professores envolvidos ativamente no projeto Dr. Neri Volpato, Dr. Rodrigo Minetto e Dr. Ricardo Dutra. À Me. Elis Nakonetchnei e ao participante do NUFER Eduardo Hungria pelo auxílio no desenvolvimento. À Universidade Tecnológica Federal do Paraná (UTFPR) pela bolsa concedida.

Disponibilidade de Código

O código-fonte desenvolvido está disponível, no momento, somente para parceiros e colaboradores por envolver sigilo de pesquisa.



XIII Seminário de Extensão e Inovação XXVIII Seminário de Iniciação Científica e Tecnológica da UTFPR

Ciência e Tecnologia na era da Inteligência Artificial: Desdobramentos no Ensino Pesquisa e Extensão
20 a 23 de novembro de 2023 - Campus Ponta Grossa, PR



SEI-SICITE
2023

Conflito de interesse

Não há conflito de interesse no presente trabalho.

REFERÊNCIAS

NAKONETCHNEI, Elis C. **HotFill: A Cooling Time Constrained Raster-Fill Planning Algorithm for Extrusion 3D Printing**. 2022. Diss. (Mestrado) – Universidade Tecnológica Federal do Paraná, Curitiba, PR.

VOLPATO, Neri. **Manufatura aditiva; Tecnologias e Aplicações da Impressão 3D**. 1. ed. [S.l.]: Blucher, 2017.