

3D 打印机多轴联动插补算法的研究与优化

吴飞, 王梦辉, 李亦能

(武汉理工大学 机电工程学院, 湖北 武汉 430070)

摘要: 针对3D打印过程中,打印机多轴联动插补算法计算效率低的问题和Bresenham算法在3D打印运动控制中的不足,在现有算法基础之上提出阶跃式Bresenham算法、速度自适应算法,分别完成喷嘴的成型扫描运动和浆料的挤出运动.采用两种算法相结合的方式控制3D打印机多轴联动,从而提高3D打印机的插补速度.同时分析了该控制方式在微控制器中的实现流程,并将其移植于微控制器中.本研究设计了直线插补仿真试验和不同算法的3D打印试验,证明了该控制方式在打印精度不变的前提下能够提高打印效率.

关键词: 3D打印;多轴联动;阶跃式Bresenham算法;速度自适应算法

中图分类号: TH 162 文献标志码: A 文章编号: 1005-3026(2024)01-0085-08

Research and Optimization of Multi-axis Linkage Interpolation Algorithm for 3D Printer

WU Fei, WANG Meng-hui, LI Yi-neng

(School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan 430070, China. Corresponding author: WU Fei, E-mail: wufei@whut.edu.cn)

Abstract: Aiming at the problem of low computational efficiency of printer multi-axis linkage interpolation algorithm in 3D printing process, the shortcomings of Bresenham algorithm in 3D printing motion control are analyzed. Step Bresenham algorithm and velocity adaptive algorithm are proposed on the basis of existing algorithms to complete the nozzle forming scanning motion and slurry extrusion motion respectively. The multi-axis linkage of 3D printer is controlled by combining the two algorithms to improve the interpolation speed of 3D printer. At the same time, the implementation flow of the control method in microcontrollers is analyzed, and it is then transplanted into microcontroller. Linear interpolation simulation experiments and 3D printing experiments with different algorithms are designed which prove that the control method can improve the printing efficiency on the premise of constant printing accuracy.

Key words: 3D printing; multi-axis linkage; step Bresenham algorithm; velocity adaptive algorithm

3D打印技术,即增材制造技术,是近几十年来发展最迅猛的技术之一^[1].打印机的控制系统多种多样,其中计算机(PC)+微控制器(MCU)+硬件控制卡的控制系统具有成本低、可独立工作、易于市场化的特点.目前该控制系统逐渐成为市场上主流的控制系統.3D打印的流程大体分为三步:第一步,利用Solidworks,CAD等软件进行打印模型的建立;第二步,通过Cura,Simplify3D等切片软件将三维模型数字化(生成

G代码);第三步,通过解析G代码进行模型打印^[2].3D打印机通过各轴对应步进电机的协同运动来完成不同打印模型的打印.3D打印机协同插补运算主要在微控制器中完成^[3],这一过程主要有以下4个步骤:

- 1) 按照Bresenham算法对打印机各轴进行插补运算,得到各联动轴的移动距离.
- 2) 根据前瞻算法和拐点速度优化算法,得到各线段间的衔接速度.

3) 由前两步可得到各轴移动距离、衔接速度(末速度). 结合 G 代码中提供的相关参数和步进电机的加减速算法得到各轴移动速度.

4) 采用定时器中断方式, 向各轴驱动器内发送相应的脉冲.

目前微控制器运算能力有限, 导致此过程运算效率较低^[4]. 本研究主要针对多轴联动部分进行研究, 鉴于多轴联动算法的缺点, 提出优化方案. 在目前 MCU 计算能力下, 通过对多电机协同插补算法的优化, 来提高插补效率, 从而提升 3D 打印机的运算速度.

1 Bresenham 算法原理和在 3D 打印中的应用

Bresenham 算法是常用于屏幕上直线段绘制的一种插补算法. 该算法将屏幕上各像素点构造一张虚拟网格, 沿着直线段的插补方向逐步插补, 依次计算直线与直线上下相邻网格交点的距离, 按照距离最近原则, 确定像素点的位置, 最终完成插补直线的绘制^[5].

如图 1 所示, 假定 $y = kx + b$ 为直线 AB 的数学方程, 直线上第 i 个像素坐标为 $A(x_i, y_i)$, 则根据 Bresenham 算法式(1)、式(2)可知, 下一个点可能的位置为: $M(x_{i+1}, y_i)$ 或 $N(x_{i+1}, y_{i+1})$. 然后按照同样的方式依次计算即可完成直线的插补.

$$\begin{cases} x_{i+1} = x_i + 1, \\ y_{i+1} = y_i + 1, d > 0.5; \\ y_{i+1} = y_i, d \leq 0.5. \end{cases} \quad (1)$$

式中, $d = |y - y_i|$.

上述讨论的是 $0 \leq k \leq 1$ 的情况, 若 $k > 1$ 则调换 x 和 y 的位置即可.

光栅化是 Bresenham 算法的基本理念, 而 3D 打印机各轴的驱动力来自各轴对应的步进电机, 各轴的运动距离和运动速度由步进电机控制器接收到的脉冲决定. 易知步进电机的步进特性亦符合光栅化的思想^[6-7].

3D 打印过程中, MCU 通过不断解析数据模型来同时控制 X, Y, Z, E_1, E_2 等多个电机协同运动. 根据光栅化的思想, 假设存在一个轴 M , 其中 M 为 X, Y, Z, E_1, E_2 的最大运动距离, 以 M 轴为主方向, 使用 Bresenham 算法分别对 MX, MY, MZ, ME_1, ME_2 进行二维插补运算, 通过这种将二维扩展到多维的方式来实现 3D 打印机多轴联动控制.

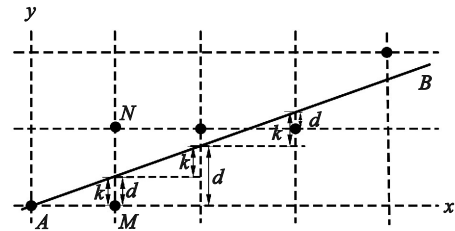


图 1 Bresenham 算法直线插补轨迹

Fig. 1 Straight line interpolation trajectory of Bresenham algorithm

2 阶跃式 Bresenham 算法与速度自适应算法

3D 打印的运动过程可以一分为二: 喷嘴扫描运动和浆料的挤出运动. 喷嘴的扫描运动决定了打印坯体的形状, 打印过程中, 除换层外, 喷嘴一直做平面内直线插补运动; 浆料的挤出运动决定了打印坯体的质量, 若喷嘴扫描运动速度大于浆料挤出速度则会产生“欠堆积”现象, 若喷嘴扫描运动速度小于浆料挤出速度则会产生“过堆积”现象. “欠堆积”、“过堆积”都会影响打印坯体的成型质量, 所以在 3D 打印过程中浆料的挤出速度与喷嘴扫描运动速度的匹配变得尤为重要.

2.1 阶跃式 Bresenham 算法

由第 1 节介绍可知, 在插补过程中, 主轴每运动一步需要做一次判定, 且判定式中涉及浮点运算, 严重影响了微控制器的插补计算效率. 如果在插补过程中经过一次判定可以确定多个进给步或者判定式中只进行整数运算, 这样在相同的微控制器下, 直线的插补计算效率就会大幅度提高.

图 2 假设第 i 个点坐标为 A , 与之对应的误差项是 d_i , 其中 $d_i = |EH| - 0.5$, 当 $d_i > 0$ 时, 下一点坐标为 N , 否则为 E . 王宝仁等^[6]为避免判别式中涉及除法和小数运算, 通过数学推导整理得到 Bresenham 算法判别式为

$$d_{i+1} = \begin{cases} d_i + 2(dy - dx), d_i > 0; \\ d_i + 2dy, d_i \leq 0. \end{cases} \quad (3)$$

初始误差项: $d_0 = 2(y_0 dx - xdy_0) - dx$.

下述以斜率为 k ($0 < k \leq 1$) 的直线 (x 轴为主运动方向) 为例进行讨论, 对于 $k > 1$ 的情况, 将 y 轴视为主运动方向, 采用同样的插补方法亦可获得.

假设 3D 打印机喷嘴在 x - y 平面内进行直线插补运动, 若 x 轴为主方向, 依据 Bresenham 算法

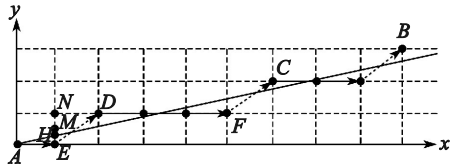


图 2 阶跃式 Bresenham 算法直线插补轨迹

Fig. 2 Trajectory of linear interpolation of step Bresenham algorithm

进行直线插补计算时, x 轴前进多步, y 轴前进一步. 若定义 x 轴前进的步数为步进台阶的宽度, 定义 y 轴前进一步为前进了一个步进台阶, 引入步进台阶概念后, 在直线插补的路径规划中, 使用 Bresenham 算法插补得到的各个插补点会组成一系列步进台阶, 如图 2 所示, 在此基础之上得到如下两个推论:

推论 1 直线段经过 Bresenham 算法插补计算后, 将得到一系列步进台阶, 除了开始和终止步进台阶, 其他各步进台阶的宽度 n 应满足:

$$F \leq n \leq F + 1.$$

式中: $F = dx/dy$, 其中 dx/dy 在 C 语言中表示 dx 除以 dy 取整, $dx = x_1 - x_0$, $dy = y_1 - y_0$.

证明 假设第 i 个步进台阶的台阶宽度为 $n(n = 1, 2, \dots)$, 设该台阶的每一个进给步判定量为 $d_1 \sim d_n$, 设第 $i-1$ 个步进台阶的进给步判定量为 d_k , 若第 $i-1$ 个步进台阶有多个进给步, 设第 $i-1$ 个步进台阶的进给步的上一个进给步的判定量为 d_{k-1} , 由上述差补算法可知:

$$d_k = d_{k-1} + 2dy, \quad (4)$$

$$d_{n-1} = d_k + 2(n-1)dy - 2dx, \quad (5)$$

$$d_n = d_{n-1} + 2dy. \quad (6)$$

其中: $d_{k-1} \leq 0; d_k > 0; d_{n-1} \leq 0; d_n > 0$.

将式(4), 式(5)代入式(6)中得到

$$d_n = d_{k-1} + 2(n+1)dy - 2dx. \quad (7)$$

所以 $d_{k-1} + 2(n+1)dy - 2dx > 0$, 又因为 $d_{k-1} \leq 0$, 所以 $2(n+1)dy - 2dx > 0$, 得到 $n > dx/dy - 1$, 又因为 n 为正整数, 所以 $n \geq F$.

由于 $d_{n-1} = d_k + 2(n-1)dy - 2dx \leq 0$, 且 $d_k > 0$, 故 $2(n-1)dy - 2dx < 0$, 则 $n < dx/dy + 1$, 又因 n 为正整数, 故 $n \leq F + 1$, 证毕.

由推论可知, 插补计算生成的一系列步进台阶, 除了第一个、最后一个外, 其他步进台阶宽度为 F 或者 $F + 1$, 如果前一个步进台阶最后一次进给步的判定量为 d_0 , 那么根据上述推导可知, 只要计算出当前步进台阶第 F 个进给步的判定量 d_1 , 然后根据 d_1 的符号就可以确定该步进台阶的

宽度, 即 $d_1 = d_0 + 2(dy - dx) + 2(F + 1)dy$. 如果 $d_1 \leq 0$, 则步进台阶宽度为 $F + 1$, 同时计算得出此步进台阶最后一步的判定变量, 即 $d_1 = d_1 + 2dy$. 否则步进台阶宽度为 F , 步进台阶最后一步的判定变量为 d_1 . 这样实现 F 或 $F + 1$ 次进给只需判断一次.

上述阶跃式 Bresenham 算法, 通过一次判定可以确定多个进给步, 提高了计算效率, 但是其中直接用 $F = dx/dy$ 来求, 涉及除法和浮点运算, 此时插补计算的效率会受到一定的影响. 为解决这一问题, 本研究提出了一种求 F 的推论.

推论 2 直线段经过 Bresenham 算法插补计算后, 将得到一系列步进台阶, 若第一个步进台阶宽度为 n , 则 $2n - 2 \leq dx/dy \leq 2n$.

证明 假设采用 Bresenham 算法生成的一系列步进台阶的第一个步进台阶宽度为 n , 根据推论 1 易知, 该步进台阶最后一个进给步的判定量 $d_n > 0$, 倒数第二个进给步的判定量 $d_{n-1} \leq 0$, 由于 3D 打印机喷嘴起始点与坐标系交点重合, 所以 $d_0 = -dx$, 插补线段的起始进给步的判定变量为 $d_1 = 2dy - dx$, 则:

$$\begin{cases} 2dy - dx + (n-2)2dy \leq 0, \\ 2dy - dx + (n-1)2dy > 0. \end{cases}$$

可得 $2n - 2 \leq dx/dy < 2$, 证毕.

由上述两个推论可知, 直线段经过 Bresenham 算法插补计算后, 将得到一系列步进台阶, 除第一个、最后一个外, 其他步进台阶的宽度值只有三种情况: $2n - 2, 2n - 1, 2n$. 即在直线插补过程中, 只需要对这三种情况进行选择, 打印机喷嘴就可以一次进给一个步进台阶.

假设第 i 个步进台阶的判定量为 d_0 , 此时的步进台阶宽度为 $2n - 2$, 判定量为 $d_1 = d_0 + 2(dy - dx) + 2(2n - 3)dy$. 若 $d_1 > 0$, 该步进台阶的宽度为 $2n - 2$; 若 $-2dy < d_1 \leq 0$, 该步进台阶宽度为 $2n - 1$; 否则该步进台阶宽度为 $2n$, 这样确定步进台阶的宽度只需一次或两次判断, 直线插补运算效率大大提高. 在连续插补过程中只需对步进台阶宽度进行实时的改变及准备下一个步进台阶宽度和判定量的计算即可. 图 3 为阶跃式 Bresenham 算法的直线插补流程图.

阶跃式 Bresenham 算法使直线插补运算效率大大提高, 但其只适用于二维平面直线插补, 并不能直接应用于 3D 打印机多轴联动插补. 由上述可知, 3D 打印过程中除换层外, 喷嘴的成型扫描运动一直在做平面二维直线插补运动^[8]. 本研

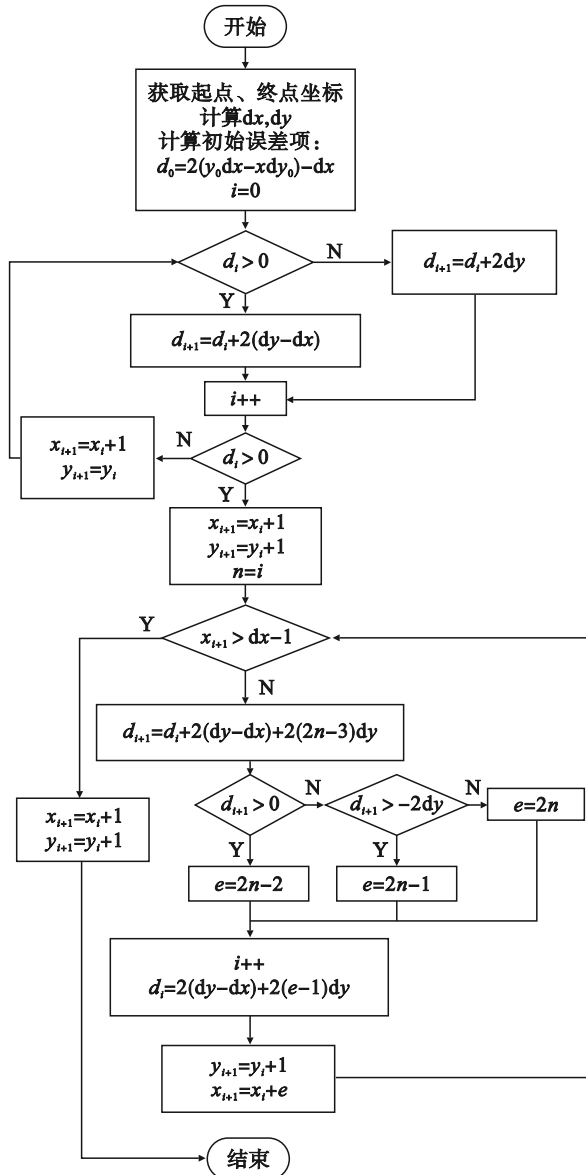


图3 阶跃式 Bresenham 算法直线插补流程
Fig. 3 Linear interpolation process of step Bresenham algorithm

究采用阶跃式 Bresenham 算法来完成喷嘴扫描的运动控制。

2.2 基于扫描速度的挤出速度自适应控制算法

3D 打印过程中, 打印机喷嘴在做连续短距离加减速运动, 其中打印机喷嘴的运动速度包括三个重要的速度节点: 起始速度、匀速速度、终止速度(衔接速度)。打印机喷嘴移动一段距离后, 根据阶跃式 Bresenham 算法可计算得出 3D 打印机各轴所需的运动距离, 经过相应的前瞻算法得到各轴的起始速度、终止速度^[9]。然后根据式(8)和式(9)所示的梯形加减速算法公式确定打印机喷嘴在运动中各个位置的运动速度。

$$\left. \begin{aligned} X_1 &= \frac{(v^2 - v_1^2)}{2 \times a_{\max}}, \\ X_2 &= \frac{(v^2 - v_2^2)}{2 \times a_{\max}}, \\ X_3 &= S - X_1 - X_2. \end{aligned} \right\} \quad (8)$$

$$\left. \begin{aligned} t_1 &= \frac{v_2 - v_1}{a_{\max}}, \\ t_2 &= \frac{v - v_2}{a_{\max}}, \\ t_3 &= \frac{X_3}{v}. \end{aligned} \right\} \quad (9)$$

其中: v, v_1, v_2 分别为当前插补路径的匀速速度、起始速度、终止速度; a_{\max} 为加减速过程最大加速度; S 为当前路径距离; t_1, t_2, t_3 分别为当前路径长度的加速时间、减速时间、匀速时间; X_1, X_2, X_3 分别为当前路段上的加速路程、减速路程和匀速路程。

3D 打印过程中, 经常会出现由于打印参数设置不当导致挤出丝不均匀的现象。为避免该现象的发生, 本文提出了速度自适应算法, 通过调整挤出步进电机的进给速度来控制喷嘴内浆料的挤出量。

为方便研究, 本研究设计了喷嘴的挤出结构并搭建了 3D 打印机, 如图 4 所示。3D 打印过程中, G 代码中的进给量代表耗材的挤出量。根据精准控制原则可知, 3D 打印机喷嘴移动一段距离所需的材料量与步进电机通过挤出齿轮控制的挤出量必须相等。图 5 为 3D 打印喷嘴内浆料的挤出示意图, 假设 3D 打印的液体浆料为密度一定的不可压缩的流体, 则步进电机进给量 E 与浆料挤出丝长度 S 的关系为

$$S \times h \times b = \pi \times \frac{D^2}{4} \times E. \quad (10)$$

其中: b 为浆料的挤出宽度; h 为挤出丝的层高; D 为喷嘴内径。长度为 S 的挤出丝要求推杆的进给位移量 E 由式(11)计算得

$$E = \frac{v}{\frac{\pi D^2}{4}} = S \times \frac{4hb}{\pi D^2}. \quad (11)$$

根据精确控制原则, 在 3D 打印过程中, 打印机喷嘴的移动速度应与浆料的挤出速度实时相等。忽略机械运动的滞后性, 可以推导出挤出步进电机的进给量与打印喷嘴移动距离的关系, 其中打印喷嘴移动距离 S 与步进电机的进给量 E 成正比例。为方便计算引入了速度匹配因子 λ :

$$\lambda = \frac{4hb}{\pi D^2}.$$

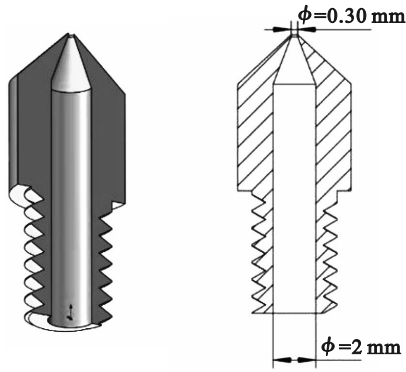


图 4 喷嘴结构
Fig. 4 Nozzle structure

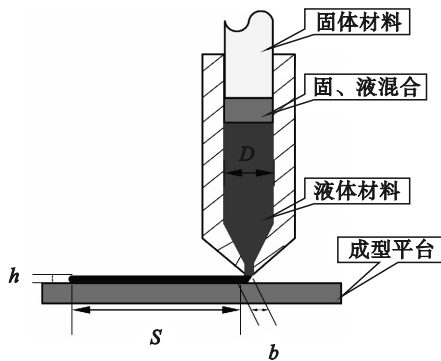


图 5 浆料挤出示意图

Fig. 5 Schematic diagram of slurry extrusion

挤出步进电机的进给速度、进给加速度和进给距离可通过打印喷嘴各阶段的运动速度、加速度、运动路径分别乘以速度匹配因子 λ 计算得到,进而求得浆料的实时挤出速度,从而完成打印喷嘴的运动速度与喷嘴内浆料的挤出速度的自适应匹配,很好地避免了“欠堆积”“过堆积”的现象。

3 多轴联动插补算法在 3D 打印机中的应用

微控制器控制步进电机协同运动分为两个阶段:第一阶段,计算出各轴运动所需的脉冲数和脉冲频率;第二阶段,将相应的脉冲数和脉冲频率发送到各轴对应的驱动控制器中。为实现程序的模块化,提出路径规划器和路径执行器的概念,分别对应程序中的.c文件。在打印过程中微控制器不断地从SD卡或上位机中读取G代码,然后调用路径规划器,在路径规划器中通过阶跃式Bresenham算法和速度自适应算法得到打印机各轴的运动状态,并将相应信息保存到内存中,然后调用路径执行器。路径执行器从固定内存中读取路径规划的信息,并向电机驱动器中发送相应的脉冲。

取任意直线路径进给的G代码规划为例(曲

线路径可以分割成多段直线路径),微控制器先从SD卡或者上位机读取一条G代码,然后调用路径规划器,在路径规划器中首先计算目标点坐标与当前喷嘴坐标点的距离,然后采用2.1小节提到的阶跃式Bresenham算法,计算得出完成该直线段插补时 x,y 两轴各自所需移动的距离,读取G代码中保存的最大匀速速度,由此计算出这一路径段的速度曲线。本研究以梯形加减速算法为例,如式(8)、式(9)所示,通过该加减速算法分别计算出路径中加速距离、匀速距离、减速距离。为方便后期对数据的调用,定义一个结构体block,如图6所示。用于存储不同插补直线路段的长度值、初末速度、相应的运动时间等关键数据。

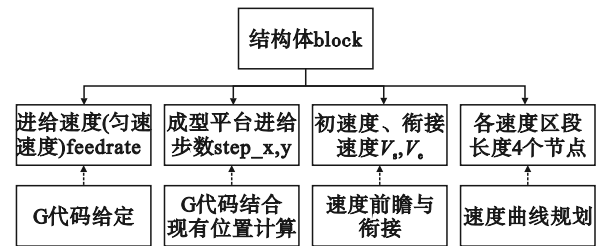


图 6 block 结构体变量示意图

Fig. 6 Schematic diagram of block structure variables

路径执行器中相应的程序通过调取各个直线段对应的结构体就可得到该直线段对应的运动信息,包括插补路径的初速度、末速度(衔接速度)及进给总步数等。通过式(12)计算得到各轴不同运动阶段的实时速度,根据速度自适应算法计算得到与之匹配的挤出步进电机的挤出速度。最后通过微处理器中的定时器发送相应个数和频率的脉冲信号给步进电机驱动器来驱动步进电机协同运动。

$$v(t) = \begin{cases} v + a_{\max}t, & 0 \leq t < t_1; \\ v, & t_1 \leq t < t_2; \\ v - a_{\max}t, & t_2 \leq t < t_3. \end{cases} \quad (12)$$

多轴联动插补算法在 3D 打印控制系统的实现流程为:

- 1) 中断函数主程序不断地读取SD卡或上位机中的G代码。
- 2) 路径规划器采用阶跃式Bresenham算法对G代码中的直线段进行插补运算,得到成型平台各轴的运动距离。
- 3) 路径规划器通过前瞻算法和G代码提供的速度参数计算得到各插补线段的初始速度、衔接速度(末速度),并将路径执行的关键信息存储在内存中(结构体block中)。
- 4) 路径执行器调取block中的关键参数,通

过步进电机加减速算法获得打印机各轴的实时移动速度,然后根据匹配算法计算得出相应的挤出电机的实时速度.

5) 根据各轴对应步进电机的实时速度,通过微控制器中的内置定时器发送相应频率和数量的脉冲信号给步进电机驱动器,来驱动各轴对应的步进电机,完成打印机的多轴联动控制.

4 实 验

4.1 插补效率验证

为了验证阶跃式 Bresenham 算法可以有效地改进插补运算的速度,本研究在 Matlab 软件上分别编写一套常规 Bresenham 算法和阶跃式 Bresenham 算法,通过模拟脉冲的发送,将待插补直线以图像的方式显示出来,并使用 Matlab 中自带的 tic, toc 函数计算程序中不同算法对同一直线段的插补时间.

为提高实验数据的可信度,对同一条直线连续插补 5 次并记录其插补时间,最后取 5 次实验的平均值.

本实验以第一象限内斜率不同的 4 条直线进

行测试:

- 1) $y=x/3$, 起点(0,0), 终点(30,10);
- 2) $y=x/10$, 起点(0,0), 终点(100,10);
- 3) $y=x/50$, 起点(0,0), 终点(500,10);
- 4) $y=x/100$, 起点(0,0), 终点(1000,10).

图 7 为直线 $y=x/10$ 的插补仿真图.

各直线轨迹生成时间如表 1~表 4 所示.

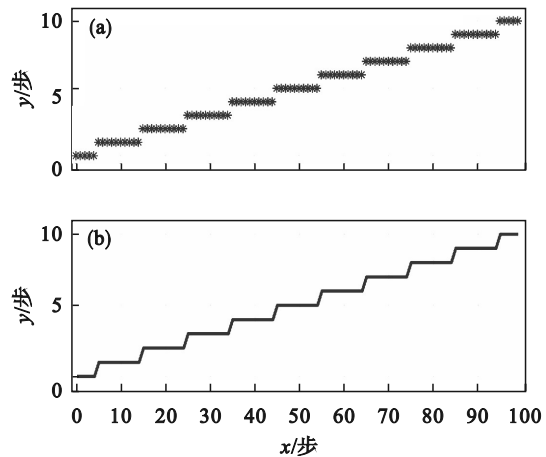


图 7 直线 $y=1/10x$ 插补仿真图

Fig. 7 Interpolation simulation of diagram $y=x/10$

(a)一点形式; (b)一线形式.

表 1 直线 $y=x/3$ 插补轨迹生成时间对比

Table 1 Comparison of generation time of interpolation trajectory $y=x/3$

算 法	1	2	3	4	5	均值
常规 Bresenham 算法	319.715	317.334	315.452	315.184	327.304	318.998
阶跃式 Bresenham 算法	264.132	261.842	259.411	261.634	270.756	259.955

表 2 直线 $y=x/10$ 插补轨迹生成时间对比

Table 2 Comparison of generation time of interpolation trajectory $y=x/10$

算 法	1	2	3	4	5	均值
常规 Bresenham 算法	467.965	460.051	461.567	459.115	460.317	461.803
阶跃式 Bresenham 算法	294.162	291.649	299.319	289.462	304.438	295.806

表 3 直线 $y=x/50$ 插补轨迹生成时间对比

Table 3 Comparison of generation time of interpolation trajectory $y=x/50$

算 法	1	2	3	4	5	均值
常规 Bresenham 算法	610.194	609.567	610.674	608.500	608.198	609.427
阶跃式 Bresenham 算法	307.196	314.756	312.268	307.334	311.468	310.604

表 4 直线 $y=x/100$ 插补轨迹生成时间对比

Table 4 Comparison of generation time of interpolation trajectory $y=x/100$

算 法	1	2	3	4	5	均值
常规 Bresenham 算法	924.648	927.399	969.339	929.085	931.863	936.467
阶跃式 Bresenham 算法	319.462	324.731	321.286	317.613	321.421	320.903

通过对比分析上述 4 组实验结果可知,阶跃式 Bresenham 算法不同程度地节约了插补运算的时间,并且随着插补直线斜率的减小,节约的时间比例增大。

下面对实验结果进行可靠性验证,根据实验结果推理可知:随着插补直线的斜率减小,节约的时间比例增大。在此结论上逆向推导,可以得到当插补直线斜率为 1 时,常规 Bresenham 算法和阶跃式 Bresenham 算法耗时相同。

本实验选取起点、终点坐标分别为 (0,0), (30,30),斜率为 1 的直线作为插补路径,分别采用不同的算法对同一直线进行插补运算,表 5 为直线插补时间,图 8 为插补轨迹仿真图。

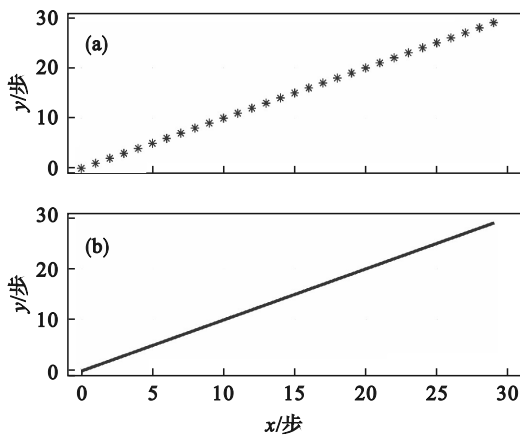


图 8 直线 $y=x$ 插补仿真图

Fig. 8 Interpolation simulation diagram $y=x$
(a)一点形式; (b)一线形式.

在直线插补过程中,阶跃式 Bresenham 算法每判断一次可以确定 n 个进给步 (n 为步进台阶的宽度),如图 8 所示,从插补仿真图可以看出,插补直线的各个步进台阶的宽度均为 1,可知,对于斜率为 1 的直线,在直线插补过程中每前进一步都需要一次判断。这种情况下与通过常规 Bresenham 算法进行直线插补的效果几乎相同。由表 5 的实验结果可知,当待插补直线的斜率为 1 时,两种插补算法消耗时长也几乎相等。证明了上述推论的正确性,从而进一步证明了阶跃式 Bresenham 可以提高直线的插补效率。

4.2 多轴联动算法验证

本研究分别使用常规 3D 打印机和移植有本

研究提出的多轴联动算法的 3D 打印机对同一模型采用不同填充方式进行打印。图 9 的 4 组实验为移植有常规算法的 3D 打印机对同一模型不同填充方式的打印结果。图 10 的 4 组实验为移植有改进算法的 3D 打印机对同一模型不同填充方式的打印结果。

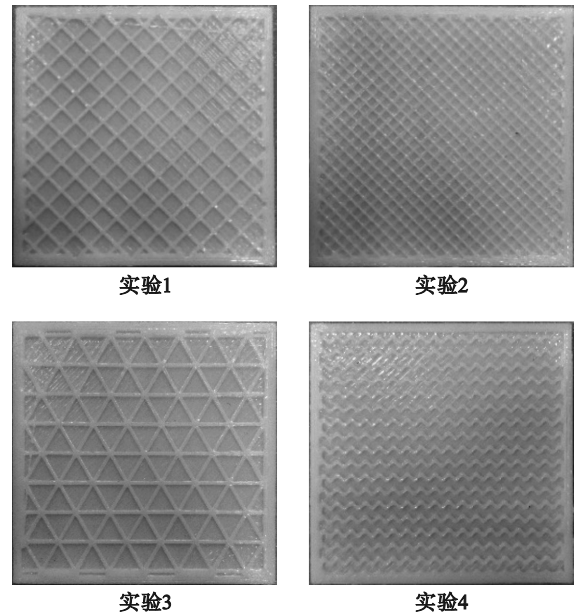


图 9 移植有常规算法的打印机打印结果

Fig. 9 Printing results of the conventional algorithm

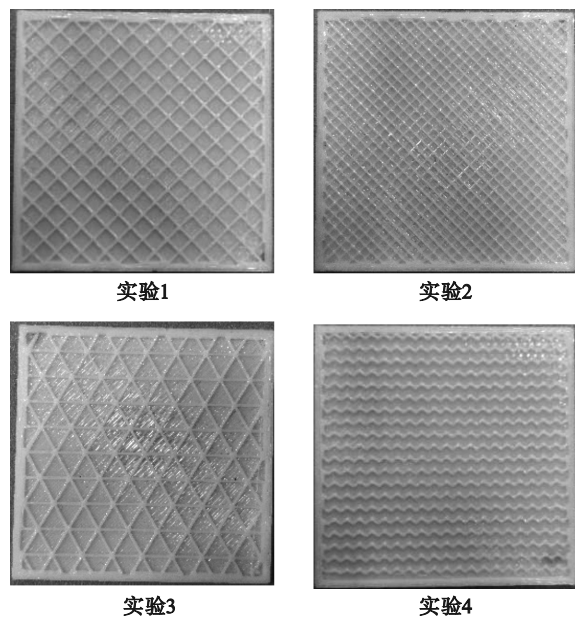


图 10 移植有改进算法的打印机打印结果

Fig. 10 Printing results of the improved algorithm

表 5 直线 $y=x$ 插补轨迹生成时间对比

Table 5 Comparison of generation time of interpolation trajectory $y=x$

算 法	1	2	3	4	5	均值	ms
常规 Bresenham 算法	367.447	370.639	379.707	361.058	364.097	368.590	
阶跃式 Bresenham 算法	368.634	366.134	367.329	363.577	375.601	368.255	

对比图 9 和图 10 中各种填充方式的打印结果知,移植有不同算法的打印机打印出来的坯体质量相等,都符合人们对打印成型件打印质量的要求.采用阶跃式 Bresenham 算法和速度自适应算法来进行多轴联动控制,没有出现浆料挤出丝不均匀的现象,证明了该控制方式的准确性.

5 结 语

本研究提出了一种阶跃式 Bresenham 算法与速度自适应算法相结合的方式控制 3D 打印机的多轴联动.阶跃式 Bresenham 算法除第一和最后一步插补外,其余各步插补过程中,步进一个台阶只需判断一次,大大提高了插补效率.研究了阶跃式 Bresenham 算法在 3D 打印机多轴联动中的不足,提出了速度自适应算法.并通过相应的实验验证,证明了通过阶跃式 Bresenham 算法与速度自适应算法相结合来控制 3D 打印机多轴联动的准确性.

参考文献:

- [1] Srinivasan D, Meignanamoorthy M, Ravichandran M, et al. 3D printing manufacturing techniques, materials, and applications: an overview[J]. *Advances in Materials Science and Engineering*, 2021, 2021: 5756563.
- [2] 余道洋. 3D 打印机 G 代码预处理优化算法[J]. *中国机械工程*, 2019, 30(1): 85-89, 112.
- (Yu Dao-yang. G code preprocessing optimization algorithm for 3D printers[J]. *China Mechanical Engineering*, 2019, 30(1): 85-89, 112.)
- [3] Liu J, Tang J L, Dai C Y. Design of DLP 3D printer control system based on arduino [C]//2020 3rd World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM). Shanghai, 2020: 497-500.
- [4] Lai C K, Lin B W, Lai H Y, et al. FPGA - based hybrid stepper motor drive system design by variable structure control[J]. *Actuators*, 2021, 10(6): 113.
- [5] Dai M, Chen Y, Zheng C Q, et al. Design of multi - step stepper motor coordinated control system based on Bresenham algorithm [C]//2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP). Auckland, 2017: 1-5.
- [6] 王宝仁, 李金松, 吕国娜, 等. 基于 Bresenham 算法的快速直线脉冲增量插补算法[J]. *机床与液压*, 2018, 46(20): 155-158.
(Wang Bao-ren, Li Jin-song, Lyu Guo-na, et al. A fast linear pulse incremental interpolation algorithm based on Bresenham algorithm[J]. *Machine Tool & Hydraulics*, 2018, 46(20): 155-158.)
- [7] Yu K, Zhang Z S, Zhou Z T, et al. A modified Bresenham algorithm for control system of FDM three - dimensional printer [M]//Billingsley J, Brett P. *Mechatronics and Machine Vision in Practice 4*. Cham: Springer, 2021: 125-139.
- [8] Foley J D. *Introduction to computer graphics* [M]. Reading, Mass.: Addison-Wesley, 1994.
- [9] Yu K, Zhang Z S, Zhou Z T, et al. Application of the five-phase S - curve velocity model on FDM three - dimensional printer [C]//2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC). Chongqing, 2020: 1365-1371.