DOI: 10.3901/JME.2020.15.218

一种基于 OpenGL 的螺旋成型加工 轴截面廓形计算方法*

贾康^{1,2} 郑帅³ 洪军^{1,2}

(1. 西安交通大学现代设计与转子轴承系统教育部重点实验室 西安 710049;

2. 西安交通大学机械工程学院 西安 710049;

3. 西安交通大学软件工程学院 西安 710049)

摘要: 轴截面加工廓形的精确计算是螺旋特征加工的一个重要问题。不同于常规的解析法、离散法或布尔运算法等都在三维图形空间内进行求解,从可视化角度将空间几何问题映射到图像空间,提出了一种基于 OpenGL 图形渲染隐含计算的计算方法。首先,对螺旋特征成型加工的数学模型进行构建;然后,对螺旋特征轴截面加工空间的渲染计算原理从图形空间到图形空间的关系进行解析;随后,对计算的完整实现方案分别从图形空间与图像空间之间的映射关系、几何模型空间相交像素信息渲染表征实现以及轴截面廓形的最终解析提取进行研究,并给完整的计算实现框架与流程性函数组。最后以锥铣刀的螺旋槽铣削加工为例,进行了轴截面加工廓形的计算,并通过与解析解进行验证了所提方法的有效性。

关键词: OpenGL; 图形渲染; 螺旋槽; 成型加工; 轴截面廓形; GPGPU

中图分类号: TH164

A Transversal Profile Calculation Method for Helical Forming Machining Based on OpenGL

JIA Kang^{1,2} ZHENG Shuai³ HONG Jun^{1,2}

(1. Key Laboratory of Education Ministry for Modern Design & Rotor-Bearing System,

Xi'an Jiaotong University, Xi'an 710049;

2. School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an 710049;

3. School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049)

Abstract: Accurate calculate the machined transverse profile of part for helical feature manufacturing is a significant question. Unlike the traditional methods such as analytical ones, discrete ones and Boolean ones solve problems in 3-D space, through mapping the transvers profile identification problem from 3-D space to image space based on the visibility theory, a OpenGL-assist implicit graphic rendering calculate method is developed. At first, the mathematical model for the forming machining of the helical feature is constructed. Then, the rendering based calculate theory is analyzed through the relationship between the 3-D space and image space. Next, the entire calculate method is detailed presented in the aspects of mapping between the 3-D space and image space, how the operation of the pixel rendering indicates the intersection in 3-D space, and the result extraction from image space to 3-D space. Meanwhile, the complete calculate frame and corresponding function groups are provided. At last, the profile calculation of for a helical groove milling by a taper cutter is taken to proof the effectiveness of proposed method via comparing with the analytical result.

Key words: OpenGL; graphic rendering; helical groove; forming machining; transverse profile; GPGPU

0 前言

等半径等导程的螺旋结构是一种非常重要的结

构形式,在机械产品有非常广泛的应用,譬如螺旋齿轮、压缩机转子、回转类刀具的螺旋容屑槽以及丝杠等。这些螺旋结构由给定的轴截面廓形绕自身回转轴线做等半径等导程的螺旋运动扫掠而成。从加工角度而言,在加工过程中,刀具绕自身高速回转形成主切削运动,刀具相对工件的等半径等导程螺旋运动为进给运动,以成型的方式完成螺旋特征

* 国家自然科学基金(51705403, 51635010)和中国博士后基金(2017M623158)资助项目。20190920 收到初稿,20200409 收到修改稿

的加工。因此, 该轴截面廓形的精确加工对于保障其工作性能起到至关重要的作用。工件螺旋特征的加工涉及三类几何问题: 已知刀具廓形和安装位姿, 求解加工出的螺旋特征轴截面廓形; 已知螺旋特征轴截面廓形和刀具安装位姿, 求解刀具廓形; 和已知刀具廓形和螺旋特征轴截面廓形求解刀具安装位姿。其中, 第一类问题是后两类问题求解的基础, 直接支持刀具安装位置的确定、刀具廓形的验证以及螺旋特征加工质量的分析等。

当前, 很多高级商业 CAD 软件可通过扫描运动完成已知刀具廓形和安装位姿的螺旋特征轴截面的廓形计算, 一些专用磨床自带的 CAM 系统^[1]也可通过输入指定类型的目标廓形文件完成刀具廓形设计等。其本质主要采用解析法、离散法和 CAD 辅助法这三类方法。其中, 解析法采用空间包络原理^[2-3], 刀具空间运动包络体部分的外表面与加工的螺旋特征表面重合, 在任一时刻, 刀具与螺旋特征为空间线接触, 该接触曲线上每一点对应的刀具切削速度矢量和刀具表面法线矢量垂直。基于此, 进一步将该接触线转换到工件轴截面内确定出加工的廓形。然而, 由于刀具廓形的复杂程度不定, 很可能造成刀具接触线出现局部不连续, 需要分段进行求解, 计算复杂^[4-7]。离散法求解的核心在于将刀具运动扫描体等效为一系列空间离散位置的同时, 将刀具看作沿轴向的一系列薄片, 从而求解出每一时刻刀具每一廓形点与工件轴截面之间的交点, 这些交点的外包络即为加工出的等效廓形^[8-11]。离散法不受制于刀具廓形奇异性的制约, 能够广泛地应用于螺旋特征轴截面加工廓形的计算, 并支持加工刀具位姿的计算搜索等问题^[12]。然而, 这些算法涉及到多组/多重超越方程的求解, 计算量大, 且计算精度取决于模型的离散精度。基于 CAD 开发包辅助的加工仿真计算方法^[13-17]通过几何实体之间的布尔运算回避了复杂的空间计算, 但涉及到商用 CAD 软件进行二次开发, 工作量较大。此外, 其计算精度由 CAD 软件底层的算法决定。综上可知, 解析法通用性较差, 离散法计算量大而基于 CAD 仿真的方法受制于软件接口且工作量较大。

这三类问题求解方法在三维的模型空间内开展, 针对现有的计算效率和方法通用性等问题, 从计算模式和计算平台角度转换思考角度, 可借鉴 GPU 内嵌的图形渲染管线另辟蹊径。图像渲染管线的工作原理可以简述为首先将三维模型投影到二维平面(Vertex 处理), 然后将二维平面的投影图像转换为离散的像素点, 并进行每一像素点的渲染计算和

深度排序等(Fragment 处理)。GPU 的多核架构天然的适用于像素处理这种单指令多线程的计算任务, 具有优良的计算性能。因此, 在 90 年代逐渐兴起将一些典型计算映射为图形渲染任务, 通过图形渲染管线来执行, 如典型的碰撞检测^[18]、刀具加工路径检测^[19]等。伴随着 GPU 并行计算能力逐渐强大, 逐渐发展出 GPGPU^[20], 其编程平台已经脱离常规面向图像的 OpenGL 而发展出 CUDA 和 OpenCL 等, 在高性能计算领域得到广泛应用。螺旋特征的轴截面廓形求解问题具有典型的模型遮挡关系, 这非常适图形渲染管线中模型转化为二维离散像素表达的思想, 因此可将本问题的求解过程映射到离散图像渲染。OpenGL^[21]作为一种图形渲染标准, 具有良好的跨平台能力, 并且能够支持底层渲染流程管线的控制。当前目标问题的计算流程中涉及的大量计算与逻辑控制能够通过 OpenGL 完美的实现, 无需过多的人为数据建模与主动控制等, 易于封装为普适的算法块, 因此本文选择 OpenGL 平台确保研究的通用性。

为保证螺旋特征轴截面加工廓形计算的通用与高效, 本文探索性的提出一种基于 OpenGL 的螺旋特征加工轴截面计算方法, 将问题求解从传统的模型空间转换到图像空间, 借助 OpenGL 的标准函数巧妙地将加工廓形计算组织到图形渲染管线, 以离散像素点的方式将大量的计算通过硬件高效完成, 可为近似类型问题提供一种新求解方法的探索。本文首先对螺旋特征加工的通用数学模型进行描述, 然后给出基于 OpenGL 的图像空间问题计算原理, 并给出详细计算流程与方法, 最后通过一个加工任务进行计算验证以表明本文方法的有效性。

1 螺旋特征加工数学模型

螺旋特征的加工可以通过车削(或拉削)完成, 亦可经由回转类刀具如铣刀、砂轮实现。加工配置如图 1 所示, 工件坐标系 $O_w-X_wY_wZ_w$ 与工件固联, Z_w 轴与工件回转轴重合, X_w 轴和 Y_w 轴由右手准则确定; 刀具坐标系 $O_c-X_cY_cZ_c$ 固联于刀具上, 令坐标系原点 O_c 与刀尖点重合, 回转类刀具以 Z_c 轴与刀具回转轴重合, 车削刀具以 Z_c 轴垂直于前刀面, X_c 轴和 Y_c 轴由右手准则确定。在初始安装时有, 工件坐标系原点 O_c 到刀具坐标系原点的矢量 $L_d=[L_{d,x}, L_{d,y}, L_{d,z}]^T$, 刀具坐标系相对工件坐标系的旋转矩阵为 R_w^c 。由此可确定出刀具坐标系向工件坐标系的转换矩阵 M_w^c , 将刀具表面 S_c 转换到工件坐标系中。

矩阵 M_w^c 具体如下

$$M_w^c = \begin{bmatrix} R_w^c & L_d \\ 0 & 1 \end{bmatrix} \quad (1)$$

在加工过程中, 刀具相对工件的直线运动速度为 v , 同步的回转运动角速度 $\omega=2\pi L_w/v$, 由此完成

导程为 L_w 的等半径螺旋特征切削。因此, 通过下式确定出任意时刻 t 刀具表面 S_c 在工件坐标系中的坐标 $S_c^w(t)$

$$S_c^w(t) = M(t) \cdot M_w^c \cdot S_c \quad (2)$$

式中, $M(t)$ 为螺旋运动坐标变换矩阵, 具体如下

图1 螺旋特征加工过程模型

$$M(t) = \begin{bmatrix} \cos(\omega t) & -\sin(\omega t) & 0 & 0 \\ \sin(\omega t) & \cos(\omega t) & 0 & 0 \\ 0 & 0 & 1 & vt \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

基于上述加工运动模型, 刀具完全切削过一个工件轴截面(如 $X_w-O_w-Y_w$ 面), 加工出的廓形即为本文待求解的廓形。

2 基于 OpenGL 的螺旋特征轴截面加工廓形计算

2.1 基于计算机图形渲染的轴截面廓形计算原理

计算机图形渲染, 如图2所示, 其本质是将三维模型空间的几何体沿投影方向离散为与屏幕像素对应的二维数组, 借助图像处理固化的一系列渲染算法并行的计算, 将二维数组结果表达图像空间。深度测试作为计算图形渲染流程中一个关键步骤, 完成空间对象沿投影方向位置关系(如大于、小于和等于等)的计算, 直接支持空间对象之间遮挡关系的正确显示。深度测试的计算借助深度缓存 Z-buff 数组实现, 其为与视景体底面离散像素点对应的二维数组, 数组元素类型为 32(或 64)位的整数。图形渲染管线中将视景体沿视线方向的范围归一化为 $[0, 1]$, 如此可实现视景体深度方向 $1/2^{32}$ (或 $1/2^{64}$) 的计算精度, 并最终视景体底面离散像素点对应的 Z-buff 表达。

针对螺旋特征的加工数学模型进行考虑, 可将这种沿视线方向的深度计算应用于轴截面加工廓形

的计算(参见图2): 由于刀具绕工件回转轴做螺旋运动, 令视口方向与工件回转轴平行, 可直观的得到刀具运动扫掠体与工件轴截面之间的垂直视图; 采用正交视图以确保离散化视线的平行, 以将视图渲染任务等效为工件回转轴平行的密集射线与工件轴截面、刀具扫掠体的隐式交点计算问题。因此, 通过合理设置深度测试计算条件, 可以借助图形渲染实现与工件轴截面深度相同(即刀具扫掠体与工件轴截面交线)的像素点的表征, 即在图像空间内完成了目标数学问题的计算。

图2 计算机图形渲染

2.2 计算流程

上述基于图形渲染的螺旋特征轴截面加工廓形计算原理, 需要从计算模型预处理、图形渲染条件设置以及渲染结果后处理计算等方面, 依据图3所

示的相对固化的计算机图形渲染执行。设计完整的计算流程如图 3 所示: 首先, 进行计算模型的预处理, 配置合理的渲染几何模型, 并准备后续计算相关的各类数据结构等; 然后, 依照图形渲染流程的执行流程, 在 OpenGL 环境下进行渲染所需函数的完整流程组织, 并设定相应的计算条件等, 从而借

助固化的图形渲染管线这一隐式的并行计算模式完成螺旋特征轴截面加工廓形的像素元素表征; 最后, 对图像空间的计算结果进行后处理, 将表征出的像素转换到几何三维空间, 然后进行加工廓形的提取及后处理, 试验加工廓形坐标的精确表达。

图 3 计算流程解析

2.3 详细计算实现

本部分依据上述的计算流程, 对基于图形渲染的螺旋特征加工轴截面廓形的计算流程进行各阶段详细算法的设计, 以形成算法模板为该类问题提供通用解决方案。

2.3.1 计算模型与数据预处理

由螺旋特征的加工成型过程可知, 当刀具运动扫掠体完全切削过工件指定轴截面才能够完整得切出目标廓形。因此, 在工件坐标系中准备数据如下。

(1) 指定工件轴截面 S_w : $z=z_w$ 。

(2) 设定刀具运动时间范围 t_s 到 t_e , 按照式(2)构建刀具运动扫掠体 $\{S_c^w(t) | t=[t_s, t_e]\}$, 要求刀具扫掠体完全切过工件指定轴截面 S_w 。

从渲染计算需求和完整的计算角度出发, 还需要准备如下数据。

(3) 设定视景体空间: 在工件坐标系中, 定义视景体左下角近点 $E=[x_E, y_E, z_E]^T$ 和右上角远点 $F=[x_F, y_F, z_F]^T$, 要求 $z_E < z_w < z_F$ 。

(4) 视口沿水平和竖直方向的离散像素数量 N_x 和 N_y 。离散像素点的数量决定着图像空间有效计算范围内交点的密化精度, 即工件轴截面的离散化精度。

(5) 图像空间计算结果存储所需的二维数组 $Buf_i[N_x][N_y]$, $i=1, 2$ 。

2.3.2 基于图形渲染的隐式计算

依据图形渲染管线的工作流程及 OpenGL 的标准函数, 设计算法块进行实现。图 3 中所示的图形渲染隐式计算包含的各环节对应的计算实现与函数设置描述如下(函数的具体参数说明参见文献[18]):

(1) 模型与视口设置。

首先将渲染对象移动到合理的渲染位置, 然后设置视线方向。这里渲染任务都在工件坐标系中统

一处理, 由后续视景体保证。视口方向设置为与工件回转轴线平行, 即沿着 Z_w 轴, 通过如下 OpenGL 函数完成

`glLookAt(*, *, *, 0, 0, 1, *, *, *)`;

(2) 投影变换。

投影变换使能对象在视景体空间中的映射模式。这里采用正交投影以保证正确计算, 并将视景体空间的设置一并完成, 采用 OpenGL 函数如下

`glOrtho(x_E, x_F, y_E, y_F, z_E, z_F)`;

(3) 渲染计算。

在图形渲染过程中, 各类渲染条件的合理设置是实现本文目标数学问题求解的关键。深度测试根据空间每一个对象点的深度值, 基于当前设置的深度测试规则(一般为大于等于通过), 与该对象点对应像素点的深度测试缓存(Z-buffer)进行比较, 如果通过测试则使用当前对象点的深度值更新该像素点的深度缓存。

设计图 4 所示操作以像素点的方式实现刀具扫掠体与轴截面加工廓形交线的表征计算: 设置深度测试条件为大于等于, 渲染后刀具扫掠体在工件截面正方向的 A1 部分能够显示, 可表征刀具扫掠体被工件截面遮挡部分的交线 C1, 然而另一侧交线 C2 由于被刀具扫掠体遮挡, 无法得到表征; 随后, 设置深度测试条件为小于等于, 即等价于视线反向, 刀具扫掠体与工件轴截面的遮挡关系互换, 可以完成 I 中被刀具遮挡交线 C2 的识别, 但 II 中已识别交线 C2 在 I 中被遮挡。结合两次渲染结果, 可得到 A1 和 A2 的交集 A3 为刀具在工件截面的切除部分, A3 的边界线就是加工廓形。

然而, 直接提取 A1 和 A2 区域的深度测试缓存(Z-buffer)并进行数值计算, 会引入较大的计算误差。因此, 通过引入模版测试操作将 A1 和 A2 区域的通

过设计的渲染逻辑表征在模版缓存中, 由此来回避这些数值计算, 简化计算流程。如此, 设计渲染算

法 I 和算法 II 分别如表 1 所示, 依此进行两次渲染计算。

表 1 基于 OpenGL 渲染计算的算法流程

行号	算法 I	算法 II
1	glClearStencil(0);	glClearStencil(0);
2	glClearDepth(0.0f);	glClearDepth(1.0f);
3	glClear(GL_STENCIL_BUFFER_BIT GL_DEPTH_BUFFER_BIT);	glClear(GL_STENCIL_BUFFER_BIT GL_DEPTH_BUFFER_BIT);
4	glEnable(GL_DEPTH_TEST);	glEnable(GL_DEPTH_TEST);
5	glDepthFunc(GL_GEQUAL);	glDepthFunc(GL_LEQUAL);
6	glEnable(GL_STENCIL_TEST);	glEnable(GL_STENCIL_TEST);
7	glStencilFunc(GL_ALWAYS, 0x0, 0xFF);	glStencilFunc(GL_ALWAYS, 0x0, 0xFF);
8	glStencilOp(GL_KEEP, GL_KEEP, GL_INC);	glStencilOp(GL_KEEP, GL_KEEP, GL_INC);
9	绘制工件轴截面 $S_w()$;	绘制工件轴截面 $S_w()$;
10	glStencilFunc(GL_EQUAL, 0x1, 0xFF);	glStencilFunc(GL_EQUAL, 0x1, 0xFF);
11	glStencilOp(GL_KEEP, GL_KEEP, GL_INC);	glStencilOp(GL_KEEP, GL_KEEP, GL_INC);
12	绘制刀具运动扫掠体 $S_c^w()$;	绘制刀具运动扫掠体 $S_c^w()$;
13	glReadBuffer();	glReadBuffer();
14	Buf ₁ [][]=glReadPixels().	Buf ₂ [][]=glReadPixels().

图 4 面向工件轴截面加工廓形交线求解的渲染计算原理

算法块的具体计算原理如下: 1-3 行将模板缓存初始化为 1, 算法块 I 将深度缓存初始化为最小 0.0f(算法块 II 中为最大 1.0f); 4 行启动深度测试; 5 行设置深度测试条件——算法块 I 设置为大于等于通过(算法块 II 为小于等于通过); 6 行启动模板测试; 7-8 行设置模板测试条件为始终通过, 并对通过测试的模板缓存增加 1; 9 行进行工件轴截面 S_w 的模型绘制(工件轴截面部分对应的深度测试和模板测试通过, 模板缓存更新为 1, 参见图 4 中的步骤

b); 10-11 行重新设置模板测试条件, 使之仅对模板缓存等于 1 的点进行测试, 由此将后续计算限制为仅涉及工件轴截面的局部区域, 设置模板测试通过条件——同时通过深度测试和模板测试的模板缓存增加 1; 12 行进行刀具运动扫掠体的绘制(这时, 算法块 I 中深度大于等于工件轴截面的刀具扫掠体部分满足条件, 模板缓存增加 1; 算法块 II 中深度小于等于工件轴截面的刀具扫掠体部分满足条件, 模板缓存增加 1; 模板缓存状态参见图 4 步骤 c)。

基于如上渲染逻辑控制, 两次渲染将区域 A1 和 A2 都在模板缓存中进行了表征。

(4) 视口变换。

将渲染结果以离散像素点合理的输出到屏幕指定区域, 即对计算数组进行数据对齐等。在 OpenGL 中按照水平和竖直方向离散个数 N_x 和 N_y 设置如下

glViewport(0, 0, N_x , N_y);

(5) 图像显示。

当前图像空间的计算结果已经以离散像素点对应的模板缓存形式确定, 通过 OpenGL 函数进行模板缓存的输出如下

glReadBuffer(GL_FRONT);

glReadPixels(0, 0, N_x , N_y , GL_STENCIL_BIT, GL_UNSIGNED_INT, Buf_i[][]);

2.3.3 图像计算结果后处理

图像空间计算结果的信息隐含在一系列离散像素点中, 需要对切削成型区域 A3(即工件轴截面)的轮廓点进行提取, 然后将其映射到视景体三维空间中, 才能够得到目标加工廓形的坐标。

首先, 识别工件轴截面加工区域 A3 的轮廓。由于 A1 和 A2 区域对应的像素点的模板缓存分别存储于数组 Buf₁ 和 Buf₂, 这两区域对应的模板缓存数值大于 1。为保证廓形识别的完备性, 按照图 5 所示, 沿水平和竖直两个方向, 分别从两端逐行的进行第一个模板缓存大于 1 的像素点的判定, 并记录该点在水平和数值方向的序号(i , j)。这些离散像素点即为 A3 区域的边界点, 亦为加工轴截面廓形。

图 5 图像计算结果后处理原理

提取加工廓形离散像素点后, 将其转换到三维几何空间方能得到加工廓形的坐标。由图形渲染原理可知, 正交投影模式下视景体的底面和渲染出的离散像素图像一一对应。由于视口坐标系为竖直向下为正方向, 所以可知视景体中的 E 点对应图像空间的索引为(1, N_y), F 点对应的索引为(N_x , 1)。因此, 渲染图像中索引为(i , j)的像

素点 P 对应于视景体内的二维坐标 $[x_P, y_P]^T$ 可计算如下

$$\begin{bmatrix} x_P \\ y_P \end{bmatrix} = \begin{bmatrix} j(y_F - y_E)/(N_y - 1) \\ i(x_F - x_E)/(N_x - 1) \end{bmatrix} \quad (4)$$

当前坐标点包括工件轴截面加工廓形和工件外圆局部曲线。因此, 通过判定离散点绕工件回转轴的距离是否小于工件外径 R_w , 剔除工件外圆的局部曲线, 得到螺旋特征轴截面加工廓形。图像计算结果后处理的伪代码如下

For i=1 to N_y Do

For j=1 to N_x Do

If Buf₁[i][j]>1 & Buf₂[i][j]>1 Then

通过式(4)计算 $[x_P, y_P]$;

If $(x_P + y_P)^2 < R_w^2$ Then

存储坐标 $[x_P, y_P]$;

EndIf

Break;

EndIf

EndFor

For j= N_x to 1 Do

If Buf₁[i][j]>1 & Buf₂[i][j]>1 Then

通过式(4)计算 $[x_P, y_P]$;

If $(x_P + y_P)^2 < R_w^2$ Then

存储坐标 $[x_P, y_P]$;

EndIf

Break;

EndIf

EndFor

EndFor

2.4 计算特性分析

本文方法借助 OpenGL 函数基于计算机图形渲染标准流程实现, 且图形空间与图像空间之间的映射计算也较为独立。由此可知, 本文方法针对每一特定问题不需要进行过多的独立数学建模, 仅需要从模型位置等角度确保计算输入, 而核心的计算任务固化为计算机图形渲染操作, 保持独立, 保证了本文算法的通用性。同时, 本文方法实质为离散计算, 回避了解析法中潜在的奇异点等稳定, 进一步确保了本文算法的稳定性。综上可知本文算法非常适合于模块化封装, 便于软件化集成使用。

3 实例验证

按照本文提出的方法针对一个螺旋特征加工任

务的轴截面廓形进行计算, 并将计算结果与解析方法的计算结果进行了精度对比。为表明渲染计算参数的作用, 设计螺旋特征的结构参数、加工配置参数以及两组渲染计算参数见表 3 所示。使用如图 6a 所示的锥形球头铣刀, 其表面 S_c 定义如下

$$f(z) = \begin{cases} (10z - z^2)^{-1} & 0 \leq z \leq 5(1 - \sin(15^\circ)) \\ 5\cos(15^\circ) + (z - 5(1 - \sin(15^\circ)))\tan(15^\circ) & \text{其他} \end{cases} \quad (6)$$

计算场景如图 6b 所示, 工件轴截面 S_w 与工件坐标系中的 $X_w-O_w-Y_w$ 平面重合。刀具扫掠体由 N_c 个离散刀具体组成, 按照运动时间周期 $[t_s, t_e]$ 均匀采样, 并通过式(2)确定。为更加清晰地展示本文算法的计算原理, 将渲染计算中的一些中间结果显示表达。如图 7 所示: 在渲染计算中借助 OpenGL 函数通过设置不同的渲染使能条件, 实质进行了沿工件的轴向正、反方向的两次渲染; 将局部渲染区域内得到的像素按照赋值进行划分, 白色区域为 buf 值=0, 浅色区域为 buf 值=1, 深色区域为 buf 值=2, 由此可以清晰的看到 BUF1 和 BUF2 表达了两个视角下的可视状况; 这两个区域的交集, 即同时满足 buf 值大于 1 的区域, 为刀具在工件轴截面 S_w 内的切削区域, 提取其廓形并剔除外径超出部分即得到工件的轴截面加工廓形。

图 6 工件螺旋特征加工几何模型

为验证本文提出算法的计算精度, 采用解析法构建包络方程, 对表 2 算例计算廓形的每一离散点

$$S_c(z, \theta) = \begin{cases} x = f(z)\cos(\theta) \\ y = f(z)\sin(\theta) \\ z = z \end{cases} \quad (5)$$

式中, $\theta \in [0^\circ, 360^\circ)$, $z \in [0, 20]$, $f(z)$ 为刀具廓形的函数

进行相应的切触点计算, 然后将其转换到工件的轴截面 S_w 内得精确的理论加工廓形。定义本文离散法求解的轴截面点与对应理论加工廓形点之间的距离为本文方法的计算误差。沿加工廓形的径向均匀取 100 个点, 对应的计算误差如图 8 所示。

表 2 螺旋特征加工参数

参数类别	具体参数
螺旋特征参数	导程 $L_w=200$ mm
	旋向=右旋
	外径 $R_w=30$ mm
加工配置参数	$L_d=[0 \text{ mm}, 20 \text{ mm}, 0 \text{ mm}]^T$
	$M_w^c = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 20 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
	$v=15$ mm/s
计算参数	组 I: $E=[12 \text{ mm}, -12 \text{ mm}, -10 \text{ mm}]^T$
	组 II: $E=[12 \text{ mm}, 0 \text{ mm}, -10 \text{ mm}]^T$
	组 I: $F=[32 \text{ mm}, 12 \text{ mm}, 10 \text{ mm}]^T$
	组 II: $F=[32 \text{ mm}, 12 \text{ mm}, 10 \text{ mm}]^T$
	$N_x \times N_y = 1\ 916 \times 946$
	$S_w: z_w=0$
	$[t_s, t_e] = [-1 \text{ s}, 1 \text{ s}]$

由第 I 组视景体参数可知, 视景体底面的像素分辨率为 $24 \text{ mm}/1\ 916 \times 20 \text{ mm}/946 \approx 0.012\ 5 \text{ mm} \times 0.021\ 1 \text{ mm}$ 。对应的计算廓形最大误差约为 0.043 mm , 这表明本文方法计算误差不大于 2 个像素分辨率; 对于第 II 组视景体参数而言, 受制于屏幕分辨率的物理限制, 采用两组视景体进行拼接等效, 这时的视景体底面像素分辨率为 $12 \text{ mm}/1\ 916 \times 20 \text{ mm}/946 \approx 0.006\ 3 \text{ mm} \times 0.021\ 1 \text{ mm}$ 。当前的廓形计算最大误差约为 0.01 mm , 同样表明本文方法的计算精度不大于 2 个像素分辨率。这两组参数的计算精度表明廓形计算精度随着像素分辨率的提高而提高。此外, 这两组计算结果误差沿加工廓形的无规律锯齿形波动正是像素的离散逼近表达造成的, 这是离散方法所有固有的, 其计算精度受制于最小像素分辨率。因此, 可以通过合理的拼接多个视景体或缩减模型渲染区域, 以此来提高像素的分辨率, 从而提高本文方法的计算精度。

图 7 基于 OpenGL 的渲染计算过程

图 8 本文方法计算精度分析

4 结论

本文针对螺旋特征的轴截面加工廓形计算问题,提出了一种基于 OpenGL 标准函数渲染的通用计算方法,主要结论如下。

(1) 本文计算方法将螺旋特征的轴截面加工廓形计算问题等效为三维模型映射到二维离散平面的深度信息判定问题,然后设计出合理的模型渲染预处理方法与图像空间向图形空间的映射后处理方法,与图形渲染管线这一标准隐式渲染流程紧密的

结合起来,并借助标准 OpenGL 函数设计逻辑操作,将计算结果隐含到离散像素中实现求解。

(2) 通过与解析法计算的标准廓形对比,可知本文方法的计算误差不大于 2 个离散像素的分辨率。同时,离散像素的分辨率是决定本文方法计算精度的关键,可以通过缩减视景体渲染范围或多个视景体拼接的方法提升本文方法的计算精度。

(3) 本文方法的求解思路从传统的三维几何空间转向了图形空间,并利用了 GPU 潜在的多核并行计算能力,为三维几何问题的求解提供了一种新的思路和计算示例。

(4) 本文方法一方面本质为离散法,不涉及奇异性等问题,求解稳定;另一方面通过标准的 OpenGL 函数完成,仅涉及三维模型的预处理,易于软件封装使用。

参 考 文 献

- [1] WALTER. Helitronic power—CNC tool grinding machine—for production and resharping[R]. 2008.
- [2] SHETH D, MALKIN S. CAD/CAM for geometry and process analysis of helical groove machining[J]. CIRP Ann.-Manuf. Technol., 1990, 39(1): 129-132.
- [3] ZHANG Wei, WANG Xianfeng, HE Fengbao, et al. A practical method of modelling and simulation for drill fluting[J]. Int. J. Mach. Tools. Manuf., 2006, 46(6): 667-672.
- [4] NGUYEN V, KO S. Determination of workpiece profile and influence of singular point in helical grooving[J]. CIRP Ann.-Manuf. Technol., 2013, 62: 323-326.
- [5] XIAO Silai, WANG Liming, CHEN Z, et al. A new and

- accurate mathematical model for computer numerically controlled programming of 4Y1 wheels in 2½-axis flute grinding of cylindrical end-mills[J]. *J. Manuf. Sci. Eng.*, 2013, 135: 41008.
- [6] REN Lei, WANG Shilong, YI Lili, et al. An accurate method for five-axis flute grinding in cylindrical end-mills using standard 1V1/1A1 grinding wheels[J]. *Precision Engineering*, 2016, 43: 387-394.
- [7] JIA Kang, HONG Jun, ZHENG Shuai, et al. An approach on wheel position and orientation calculation for helical broaching tool sharpening[J]. *Int. J. Adv. Manuf. Technol.*, 2017, 92: 1991-2000.
- [8] KO S. Geometrical analysis of helical flute grinding and application to end mill[J]. *Transactions of NAMRI/SME*, 1994, 12: 165-172.
- [9] WU Yuren, FONG Zhonghua, ZHANG Zhengxiang. Simulation of a cylindrical form grinding process by the radial-ray shooting (RRS) method[J]. *Mechanism & Machine Theory*, 2010, 45(2): 261-272.
- [10] 贾康, 洪军, 张银行. 一种拉刀螺旋容屑槽前刀面磨削砂轮安装位姿计算方法[J]. *机械工程学报*, 2019, 55(11): 205-214.
- JIA Kang, HONG Jun, ZHANG Yinhang. An approach on wheel setup calculation for helical rake flank sharpening of broaching tool[J]. *Journal of Mechanical Engineering*, 2019, 55(11): 205-214.
- [11] JIA Kang, GUO Junkang, ZHENG Shuai, et al. A general mathematical model for two-parameter generating machining of involute cylindrical gears[J]. *Applied Mathematical Modelling*, 2019, 75: 37-51.
- [12] KARPUSCHEWSKI B, JANDECKA K, MOUREK D. Automatic search for wheel position in flute grinding of cutting tools[J]. *CIRP Ann.-Manuf. Technol.*, 2011, 60: 347-350.
- [13] LI Guochao, SUN Jie, LI Jianfeng. Process modeling of end mill groove machining based on Boolean method[J]. *Int. J. Adv. Manuf. Technol.*, 2014, 75(5-8): 959-966.
- [14] IVANOV V, NANKOV G, KIROV V. CAD orientated mathematical model for determination of profile helical surfaces[J]. *Int. J. Mach. Tools. Manuf.*, 1998, 38(8): 1001-1015.
- [15] MOHAN L, SHUNMUGAM M. CAD approach for simulation of generation machining and identification of contact lines[J]. *Int. J. Mach. Tools. Manuf.*, 2004, 44(7-8): 717-723.
- [16] LI Guochao, SUN Jie, LI Jianfeng. Modeling and analysis of helical groove grinding in end mill machining[J]. *J. Mater. Process. Technol.*, 2014, 214: 3067-3076.
- [17] KIM J, PARK J, KO T. End mill design and machining via cutting simulation[J]. *Computer-Aided Design*, 2008, 40(3): 324-333.
- [18] 范昭炜, 万华根, 高曙明. 基于流的实时碰撞检测算法[J]. *软件学报*, 2004(10): 83-92.
- FAN Zhaowei, WAN Huagen, GAO Shuming. Streaming real time collision detection using programmable graphics hardware[J]. *Journal of Software*, 2004(10): 83-92.
- [19] WANG Qinghui, LI Jingrong, GONG H. Graphics-assisted cutter orientation correction for collision-free five-axis machining[J]. *International Journal of Production Research*, 2007, 45(13): 2875-2894.
- [20] LUEBKE D, HARRIS M, KRUGER J, et al. GPGPU: General purpose computation on graphics hardware[C]// *Acm Siggraph Course Notes*, ACM, 2006.
- [21] SHREINER D. OpenGL 编程指南[M]. 5 版. 北京: 机械工业出版社, 2006.
- SHREINER D. OpenGL programming guide[M]. 5th ed. Beijing: China Mechine Press, 2006.
-
- 作者简介: 贾康, 男, 1987 年出生, 博士, 助理研究员。主要研究方向为数字化制造与装配, CAD/CAM, 大尺寸空间测量。
E-mail: jiakang@xjtu.edu.cn
- 洪军(通信作者), 男, 1968 年出生, 博士, 教授, 博士研究生导师。主要研究方向为数字化设计、制造与装配, 精密电主轴技术。
E-mail: jhong@mail.xjtu.edu.cn