

文章编号: 2096-1618(2023)04-0404-06

基于剖面轮廓点投射算法的管道交互式程序化建模方法

王晓龙, 涂宇, 祁泽凡, 刘应浒, 何晓曦

(成都信息工程大学软件工程学院/四川省信息化应用支撑软件工程技术研究中心, 四川 成都 610225)

摘要:现代工业管道建筑管网布线复杂精细,对设备模型曲面的精细化表达有更高的要求。同时,追求网格模型的精确与高效的开发效率是模型开发人员的迫切愿望,对工业管道进行精确的图形描述和简化工业管网的工作流水线成为该领域的研究重点。提出一种改进的基于样条的管道剖面轮廓点位置投射计算方法,从现有的非分割管道网格中提取曲线控制节点,为用户提供可编辑的初始框架并完成管道网络的拼接。实验结果表明,该方法可用于工业领域中各类型管道或类似设备模型的仿真建模。构建框架和组件易于编辑,对非专业用户友好,具有高效性和高容错率。

关键词:样条;剖面轮廓点位置计算;纵截面偏移曲线;样条控制点检测;程序化建模

中图分类号:TP312

文献标志码:A

doi:10.16836/j.cnki.jcui.2023.04.005

0 引言

在工业仿真中,管道作为其工业设备体系中重要的组成部分,管道建模需要对实际的管道布线进行精确的表示。相对于传统构造立体几何(constructive solid geometry)和分解表示^[1]的建模方式,工业管道的建模流程其立体几何由众多曲面网格构成,不适用分解表示和构造表示这两种建模方案。因此,本文采用更能表现由许多曲面(三角形、样条)粘合起来形成封闭的空间区域的边界表示法(boundary representation)^[1]。

在网格程序化阶段,对于不同弯曲程度的管线进行精确的曲线拟合,该方法基于样条的建模方法,将管道基础框架描述为多个样条,用户可以通过拖动管道框架的样条节点进行编辑。然后,通过基于样条的网格生成方法生成脊状网格,该环节针对不同类型截面的管道网格进行程序化网格生成。最后,建筑物其余部分的网格通过沿框架实例化预制建筑组件来生成。该方法对新手用户友好,管道模型易于编辑。但是,仍然存在一些缺陷。文献[2-3]介绍的基于样条的网格生成方法只能生成管状网格,当样条中有急性拐点时,生成的网格会发生扭曲。管道转角处的网格存在扭曲穿模问题,因此编辑具有许多顶点的管道接口对用户来说仍然很困难。本文提出了一种交互式 and 程序建模方法来解决上述缺点,允许非专业用户在几分钟内创建所需的工业管道模型,并使管道和组件易于编辑。主要贡献如下:

(1)提出一种新的基于样条曲线的管道剖面轮廓

顶点位置算法,通过曲线拐点位置的网格顶点合理计算,使直线转角处能有效避免穿模情况的发生、曲线转角处网格模型有良好的曲面表现。同时,引入纵截面偏移曲线,丰富管道表面的曲面变化。

(2)提出一种基于BVH(bounding volume hierarchies)^[4]改进的碰撞检测校验方法,用以管理场景中的样条节点的同时加速节点的碰撞检测,使用户的节点编辑操作通过碰撞检测机制精准地定位目标位置。这一交互方案简化了用户的操作流程,提高了建模效率。

1 相关工作

现有的程序化建筑建模技术可以分为4类:基于规则、数据驱动、基于草图和基于形状语法的方法。本文提出一种新的基于样条的交互式建模方法。

1.1 基于样条的交互式建模

样条曲线所构造的管道在针对不同模型重建时的沿样条轨迹方向曲面自然过渡(无扭曲且接缝自然),需要一个完善的解决方案。分段多项式构造的曲面表示^[5]常被应用于复杂曲面。计算机辅助设计(CAD)中边界表示对于生成具有近似曲面边界的非结构化三角形网格是曲面生成的重点^[6-9]。Ims等^[10]设计一种基于程序化几何重建将线性非结构化网格转换为弯曲的高阶网格,从线性表面网格重建高阶曲面,同时保留原始低阶曲面的尖锐特征边缘。与现有的表面重建方法不同,Jiao等^[11]提出局部拟合的加权平均和连续移动帧。这些方法都基于加权最小二乘多项式拟合并保

证 C^0 连续性,使网格达到较高的精度并对尖锐特征进行综合处理。

上述方法仅针对从现有低阶网格计算生成高质量的高阶网格,在动态生成管道网络时计算效率依然是需要解决的问题。

为得到边界表示合理的模型,另一种思路则是简化原始模型^[12-13]。为获得简化模型,一种标准方法是在原始 CAD 模型^[14-15]之上定义虚拟拓扑。一种类似 CAD 的简化方法,强调自动化和后部网格划分需求,考虑了网格约束拓扑^[16-17]。这些简化框架已被证明在许多实际应用中是成功的。在简化过程中,这些框架可能会出现一些几何翘曲问题。这些问题可以通过保持原始 CAD 和仿真模型之间的关键节点之间的联系来解决^[18]。关键思想是将简化步骤存储在连接设计和仿真几何的高级框架中,促进 CAD 去特征化^[18-19]。

与生成高阶网格模型或简化原始网格模型的方案不同,Yang 等^[20]提出一种基于 Semi-NURBS 形式的 T 样条的曲面融合框架。该框架渐进式的拟合方案基于八叉树的自适应网格细分,无须构建有限元方法中使用的大型矩阵,在需要精细变形的区域进行自适应细化。该方法中细分区域需要在其算法参数域中插入控制点,这也导致在模型构建过程中细分区域的控制点需要更精准的选取。

Huang 等^[3]提出了一种新颖的基于样条的交互式程序化建模方法。为用户提供了一个基于样条的标准化网格挤出框架,用户通过两类控制点编辑框架:位置控制点和弯曲控制点。用户完成编辑后,建模系统从构件库中实例化预制构件,并将其逐一放置到框架上,得到结果模型。在样条算法的基础上完成平滑拼接和组合,以满足管道在工业化体系中的设计要求。改进后的系统仍有很大的局限性。该框架无法构建沿特定方向挤出立面网格,如栅栏、台阶等模型。

1.2 碰撞检测加速结构

由于围绕移动几何体重新调整包围盒相对简单,层次包围盒(BVH)成为一种流行的加速结构,适用于光线追踪渲染的动画场景。然而,如果场景中的对象随着动画的进行而移动或重新排列,那么这种更新后的树的遍历效率会迅速下降,导致渲染时间的急剧增加和帧率的相应降低。Kopta 等^[4]提出了一种通过树旋转有效地扩展动态场景重新达到同步的方法,用于离线更新静态场景的 BVH 效率,是一种快速、轻量级的增量更新算法。然而,在实现大规模碰撞检测和面对同一位置的多个不同对象时,由于返回大量的碰撞节点,若需要获取特定节点而进行遍历校验,其检测效

率会随着返回的节点数量的不断上升而下降。

本文解决了目前工业管道建模中的交互式和程序化方法的弱点,建筑和构件由样条和根据样条挤出网格的组件来描述,用户可以实时针对单条管道编辑模型,从整体形状到局部构件并通过节点之间的碰撞实现实时吸附以达到准确的位置。改进的建模框架支持不同样条类型的管道,并支持自定义截面配置文件来读取剖面 and 纵截面的分段多项式等相关数据,用户只需通过图形工具对剖面 and 纵截面进行轮廓的绘制编辑,进一步降低了建模的难度。本文的网格挤出方法是完全自动化的,不需要制作任何代理模型,用户可以专注于模型编辑。

2 管道表面建模

基础样条曲线的生成和顶点所组成的实例化网格是本文建模系统中主要的曲面建模操作,是后续部分研究工作的基础。

样条网格生成允许用户自定义样条曲线,然后根据样条曲线生成管道网格。对于直线路径的管道、柱体等模型,使用样条类型 Linear 绘制直线路径即可;对于具有平滑转角的路径,则可以选择 3 次贝塞尔样条^[21]对拐点网格进行编辑等。本文使用 3 次贝塞尔对曲线进行描述,这种曲线是直观且易于控制变化的,可以有效地描述弯曲的形状。

文献[3]介绍了一种基于样条的脊状模型生成方法,它使用预先定义的脊状轮廓沿样条轨迹扫描以生成脊状网格。如图 1 所示,a 模型为管道基础样条,b 模型是对剖面顶点位置未作改进的网格挤出算法所生成的网格模型,c 模型是使用本文方法所生成的网格模型。当样条轨迹具有锐角时,这种方法会产生不均匀的管道网格,主要原因是样条的每个控制节点处的轮廓形状差异,但在文献[2]中的所有控制节点处都使用相同的形状。本文使用一种新的管道剖面轮廓点位置计算方法来解决这个问题并进行一些改进,使其可以描述所需要的任何组件形状。



图1 具有锐角转角的样条管道网格对比

2.1 剖面轮廓点计算

令 N_i 、 T_i 和 B_i 分别为每个样条控制点的法线、切线和正切向量, $i \in [1, n]$ 。管网的轮廓由用户定义。如图 2 所示,首先将垂直于管道轴线的剖切平面轮廓转换为由 T_1 和点 p_1 定义的平面 plane_1 。

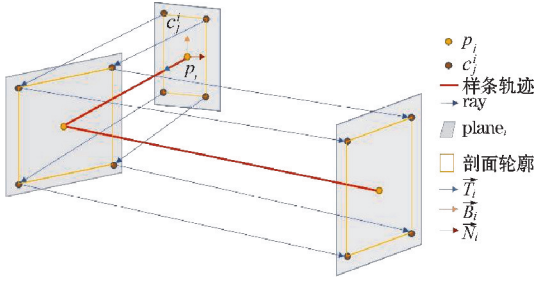


图2 剖面轮廓点位置计算

然后,以每个轮廓控制点 $c_j^i, j \in [1, m]$ 为起点,向 $\overrightarrow{p_i p_{i+1}}$ 方向发射一条射线 ray_j^i ; c_j^{i+1} 是 ray_j^i 和 plane_{i+1} 的交点:

$$c_j^{i+1} = c_j^i + \frac{-\mathbf{T}_{i+1} \cdot (p_{i+1} - c_j^i)}{-\mathbf{T}_{i+1} \cdot \mathbf{d}} \cdot \mathbf{d} \quad (1)$$

式中 $\mathbf{d} = \overrightarrow{p_i p_{i+1}} / \|\overrightarrow{p_i p_{i+1}}\|$ 。 plane_{i+1} 的剖面轮廓由剖面间的点 $\{c_1^{i+1}, c_2^{i+1}, \dots, c_m^{i+1}\}$ 定义。最后,将用户定义的剖面控制点和所有计算出的剖面轮廓顶点连接后得到管道网格。图1中c模型显示了使用本文方法生成的管道网格,只能生成普通管道形状。为了使该方法更加通用,在管道网格中加入了纵截面偏移曲线和纵截面轮廓约束。

2.2 纵截面偏移曲线

为了使沿管道轴线方向的剖切挤出面能够自由定义,引入纵截面偏移曲线,偏移曲线是用户定义的曲线 $y(x), x \in [0, 1]$ 。对于管网的任何顶点 c_j^i ,将其沿方向 $\overrightarrow{p_i c_j^i}$ 进行偏移以产生网格顶点:

$$c_{i,j}^{\text{disp}} = c_j^i + \varphi \cdot y\left(\frac{i}{n}\right) \cdot \frac{\overrightarrow{p_i c_j^i}}{\|\overrightarrow{p_i c_j^i}\|} \quad (2)$$

其中 φ 是缩放系数,曲线 $y(x)$ 为用户自定义曲线函数,用于表达平行方向于 $\overrightarrow{p_i c_j^i}$ 的偏移变量, n 是样条的总结点数。

2.3 纵截面轮廓约束

方向约束用于根据用户定义 \mathbf{B}_i 和 $\overrightarrow{p_i c_j^i}$ 之间的角度阈值 $\delta, \delta \in [0, \pi]$,选择待偏移的顶点和待偏移顶点的方向。对式(2)进一步改进,得到约束结果 $c_{i,j}^{\text{constrain}}$:

$$c_{i,j}^{\text{constrain}} = c_j^i + \varphi \cdot y\left(\frac{i}{n}\right) \cdot \mathbf{W} \cdot \overrightarrow{\mathbf{D}} \quad (3)$$

$$\overrightarrow{\mathbf{D}} = \begin{cases} \frac{\overrightarrow{p_i c_j^i}}{\|\overrightarrow{p_i c_j^i}\|}, & \delta > \frac{\pi}{2} \\ \mathbf{B}_i, & \delta \leq \frac{\pi}{2} \end{cases} \quad (4)$$

$$\mathbf{W} = \begin{cases} 1, & \alpha \leq \delta \\ 0, & \alpha > \delta \end{cases} \quad (5)$$

其中: α 是 \mathbf{B}_i 和 $\overrightarrow{p_i c_j^i}$ 之间的角度;当 $\delta \leq \frac{\pi}{2}$ 时,如图3所示的楼梯状模型,只有 p_i 水平上方的控制点做偏移处理,而下方的控制点则被约束不作偏移处理。因此构件的形状,如楼梯、管道连接中用于螺纹连接的外部紧固件、用于法兰连接的连接阀门等可以很容易地定义。

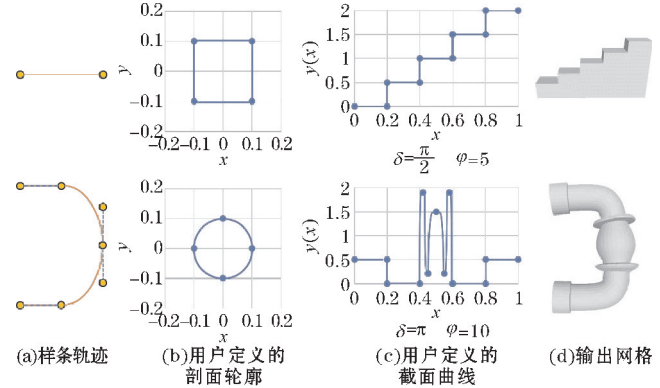


图3 使用样条生成网格模型

3 基于 BVH 的哈希码碰撞检测

为提升在局部包围盒中控制点的查询效率,并解决在同一坐标下的不同控制点的碰撞检测同时避免冲突,基于 Kopta 等^[4]的方法,提出了使用对控制点属性的 xxHash 编码(一种非加密哈希函数)作为控制点的唯一标识,对同一包围盒中不同控制点进行进一步检测筛选。

3.1 生成 HashID

令 P 为控制点属性集合, P_n 为第 n 类属性的值, $n \in \mathbf{N}^*$ 。将控制点各属性分量 P_i (如坐标、单位四元数以及空间对象缩放比例)通过计算后的得到待哈希运算的整形数据。

令初始散列完成后得到的哈希编码为 h ,将上一步得到的整形数据作为参数输入 xxHash 函数后得到该控制点的哈希编码:

$$h(n) = \begin{cases} h(n-1) \cdot \text{prime} + y(P_{n-1}), & n > 1 \\ k_{\text{prime}}, & n = 1 \end{cases} \quad (6)$$

其中, h 在带入公式计算前赋特定散列种子初始值 k_{prime} , prime 作为属性进行非加密哈希散列过程中特定素数。函数 $y(x)$ 返回变量 x 所属类型的哈希码。

最后,将散列初始值 h 作为 xxHash 函数的输入计算,返回得到最终的控制点唯一标识编码,如图4所示。每个样条控制点具有唯一且固定的 HashID 作为其唯一标识,不随控制点属性的变化而变化。因此 HashID 的生成速度不是唯一重要的属性,生成的哈希值遵循具有良好的离散性和随机性属性,以便控制点

都可以最大程度地离散,并将冲突的数量在理论水平减少到最小,遵循生日悖论。

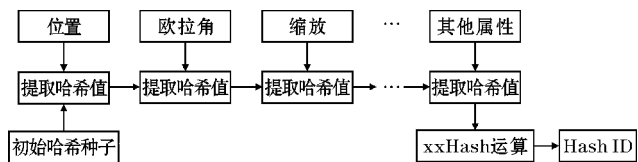


图4 生成样条控制点 Hash ID

3.2 精确碰撞检测

提出的基于样条的交互式建模碰撞检测可总结为以下几个固定步骤:

步骤1 输入待查询控制点,通过 BVH 递归检测返回所处包围盒内的控制点集合 SplinePointList。

步骤2 通过步骤1 输入的待查询控制点的 HashID 进行哈希码校验,得到准确的 BVH 叶节点,排除掉 SplinePointList 中相同位置的控制点。

步骤3 用户在界面中移动控制点,实时更新包围盒大小及 BVH 树结构。

步骤4 BVH 碰撞检测得到包围盒叶节点中所有除用户移动的所有控制点,并遍历计算位置属性。最终,得到碰撞目标并完成吸附功能。

控制节点中 HashID 这一属性的引入,使在遍历查询 BVH 树这一过程中目标节点的准确命中。由 BVH 碰撞返回的叶节点包围盒集合中,通过哈希校验得到精确的节点查询结果,其中包含待获取的各项节点属性。由于加入了哈希码校验,使本文方案相较于传统的递归检测在查询目标控制点并更新 BVH 树阶段可提升部分计算效率。

4 实验及结果

使用的实验编程环境为 Unity3D 2021. 1. 3f1 (64 位)。计算机配置如下: Intel(R) Core(TM) i9-10900 CPU @ 2.80 GHz、32.0 GB 内存、GeForce GTX 1660 SUPPER GPU 和 6 GB 显存。

4.1 模型构建测试

邀请 5 位用户(2 位男性、3 名女性;年龄:22 ~ 26 岁,平均:24.5 岁)使用本文的工具来完成建模任务。所有测试用户是具有 3D 建模工具使用经验的动画公司专业人员,并且至少熟悉一种主流 3D 建模工具。

为用户提供了同一工业区域的三维倾斜摄影数据,这些数据描绘了具有不同网络结构的目标管道,例如不同的管道截面和轴向过渡。要求用户一人一组,分为 5 组。每位用户使用 Houdini、Rhino Grasshopper

以及本文的工具对厂区内特定类型设备进行重建。

每位用户针对一种模型设备分别用上述 3 种工具进行模型重建并导出至 Unity3D 进行渲染,重建结果如图 5 所示。图 6 显示了每个模型的编辑时间。

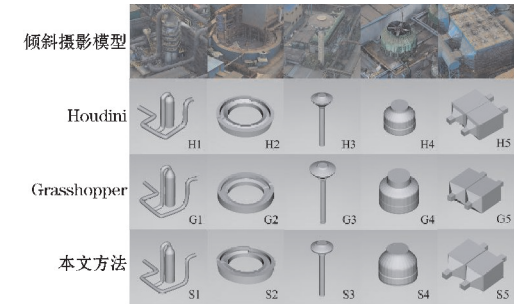


图5 模型重建效果截图

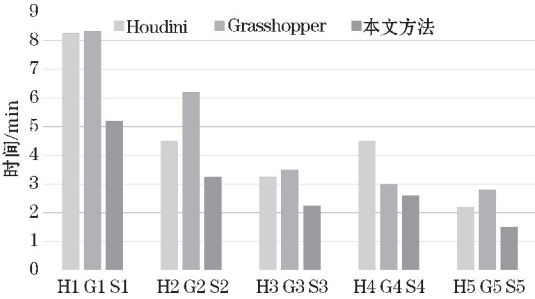
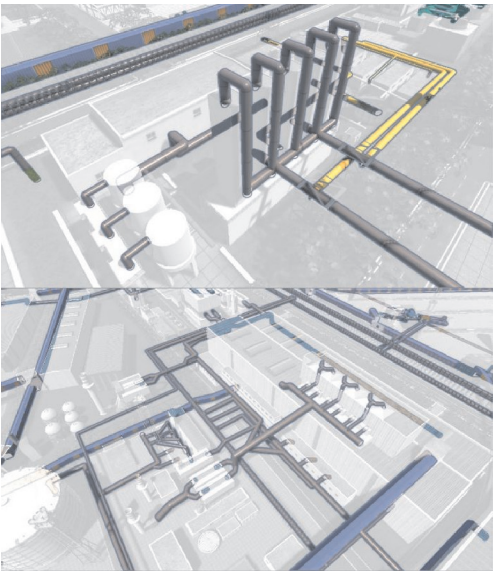


图6 模型重建时长统计

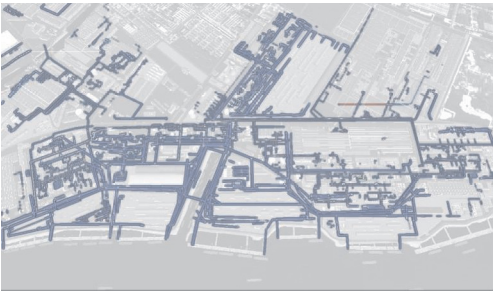
表 1 统计了实验中所涉及的模型的各项几何属性,其中包含模型的顶点数量、三角形数量及网格数据量。

表 1 模型各项属性统计			
模型编号	顶点	三角形	网格/KB
H1	23,116	11,588	1,588
G1	23,756	11,822	1,604
S1	18,532	9,648	1,256
H2	2,254	1,130	170
G2	2,562	1,380	185
S2	1,528	746	125
H3	6,060	3,056	420
G3	5,940	3,012	415
S3	3,984	1,855	302
H4	3,660	1,856	254
G4	3,265	1,654	231
S4	2,135	1,028	126
H5	1,616	808	128
G5	1,564	782	120
S5	1041	597	74

从实验结果可以看出,由于本文方法在模型编辑过程中不需要进行节点流程的编辑,同时引入节点与节点之间的碰撞检测与自动吸附功能,因此构建模型所花费的时间大大缩短。图 7(a) 显示了部分区域的管道模型细节,图 7(b) 是通过本文方法构建的大规模管道网络。



(a) 局部场景



(b) 工业厂区管道网络鸟瞰图
图7 大规模管道网络场景

4.2 空间加速与碰撞检测性能对比

HashID 作为 BVH 碰撞检测机制中判别目标检测域中同一包围盒或相邻包围盒中的目标节点关键属性,使得返回的碰撞节点更加精确并符合预期的交互的实验结果。相关实验条件如下所示:

- (1)实验算法使用 C#语言实现,基于 unity 平台进行渲染。
- (2)测试系统使用 Unity-TestRunner 测试系统进行单元测试。
- (3)哈希编码均采用 32 位存储,参与测试的哈希算法均采用 64 位算法版本。

通过随机生成控制节点对象,并将其加入 BVH 树,之后遍历其中的部分节点修改其节点的位置属性进行碰撞检测并实时更新 BHV 树。实验参照组分别使用 BVH 原始算法和八叉树这两种空间加速算法。实验对比结果如表 2 所示。

表2 空间加速算法运行帧率统计结果 单位:fps

节点数	OurMethod	BVH	Octree
100	768	761	765
500	525	512	526
1000	336	322	319
3000	180	155	141

实验结果表明通过本文算法在较大规模的场景更新和碰撞检测上有一定的优势,当场景控制点对象数量越多,性能提升越明显。

5 总结与展望

提出了一种扩展的交互式和程序化方法来模拟工业设备建筑和建筑。本文方法使用样条网格生成,以完成工业管道和部分设备的建模。同时,改进了现有的 BVH 碰撞目标检测机制,以便非专业用户也可以轻松地编辑样条控制点。

实验结果表明,本文工具具有广泛的应用。没有 3D 建模经验的用户可以使用本文的工具在几分钟内完成对目标建筑的建模,用户还可以对建筑的延展曲面进行编辑,表现出工业设备的丰富细节。与现有方法相比,本文的参数化方法保存的模型文件很小。此外,所改进的基于 BVH 的碰撞检测方法可以显著加快用户的建模速度并提升用户的工作效率。

未来,除了解决用户提出的问题外,还计划做以下尝试:通过读取管线矢量数据并生成管道模型,加快模型开发速度;为模型增加边对象预制件,并实现自定义预制件和动态读取。

参考文献:

[1] Hoffmann C M, Shapiro V. Solid modeling [J]. Handbook of Discrete and Computational Geometry, Third Edition, 2017:1503-1539.

[2] Huang C Y, Tai Y K. Ting tools: interactive and procedural modeling of Chinese ting[J]. The Visual Computer, 2013, 29(12):1303-1318.

[3] Huang C Y, Sheng Y S, Tai Y K. Interactive and procedural modeling of featured chinese architectures[A]. 13th International Symposium on Smart Graphics, 2015:16-28.

[4] Kopta D, Ize T, Spjut J, et al. Fast, effective BVH updates for animated scenes [P]. Interactive 3D Graphics and Games, 2012:2-3.

[5] Lane J M, Riesenfeld R F, Theoretical Development for the Computer generation and Display of Piecewise Polynomial Surfaces[J]. Transactions on Pattern Analysis and Machine Intelligence. IEEE, 1980(1):35-46.

[6] Baker T J. Mesh generation: Art or science? [J]. Progress in Aerospace Sciences, 2005, 41(1):29-63.

[7] Catmull E, Clark J. Recursively generated B-spline surfaces on arbitrary topological meshes[J]. Com-

- puter-aided design,1978,10(6):350-355.
- [8] Owen S J, Staten M L, Canann S A, et al. Q-Morph: an indirect approach to advancing front quad meshing [J]. International journal for numerical methods in engineering,1999,44(9):1317-1340.
- [9] Cass R J, Benzley S E, Meyers R J, et al. Generalized 3-D paving: an automated quadrilateral surface mesh generation algorithm [J]. International Journal for Numerical Methods in Engineering, 1996,39(9):1475-1489.
- [10] Ims J, Duan Z, Wang Z J. Proceedings of The 22nd AIAA Computational Fluid Dynamics Conference [A]. 22nd AIAA Computational Fluid Dynamics Conference,2015:2293.
- [11] Jiao X M, Wang D. Reconstructing high-order surfaces for meshing [J]. Engineering with Computers,2012,28(4):361-373.
- [12] Thakur A-B, Ashis G-G, Satyandra K. A survey of CAD model simplification techniques for physics-based simulation applications [J]. Computer-Aided Design,2009;41(2):65-80.
- [13] White D-R, Saigal S, Owen S-J. Meshing complexity: predicting meshing difficulty for single part CAD models [J]. Eng Comput 2005;21(1):76-90.
- [14] Sheffer A, Bercovier M, Blacker T, et al. Virtual topology operators for meshing [J]. International Journal of Computational Geometry & Applications,2000,10(3):309-31.
- [15] Sheffer A. Model simplification for meshing using face clustering [J]. Computer-Aided Design, 2001,33(13):925-34.
- [16] Foucault G, Cuillière J-C, François V, et al. Adaptation of CAD model topology for finite element analysis [J]. Computer-Aided Design, 2008, 40(2):176-96.
- [17] Foucault G, Cuillière J-C, François V, et al. Generalizing the advancing front method to composite surfaces in the context of meshing constraints topology [J]. Computer-Aided Design, 2013, 45(11):1408-25.
- [18] Shapiro V, Tsukanov I, Grishin A. Geometric issues in computer aided design/computer aided engineering integration [J]. J Comput Inf Sci Eng,2011,11(2):1-13.
- [19] Nolan D-C, Tierney C-M, Armstrong C-G, et al. Defining simulation intent [J]. Computer-Aided Design,2015,59:50-63.
- [20] Yang J M, Zhao G. Surface Blending Using T-splines in Semi-NURBS Form [J]. Computer-Aided Design,2022,146.

An Interactive Procedural Modeling Method for Pipelines based on Profile Point Projection Algorithm

WANG Xiaolong, TU Yu, QI Zefan, LIU Yinhu, HE Xiaoxi

(College of Software Engineering, Chengdu University of Information Technology, Sichuan Province Engineering Technology Research Center of Support Software of Informatization Application, Chengdu 610225, China)

Abstract: Modern industrial pipeline building network wiring is complex and fine, so it has higher requirements for the refined representation of equipment model curved surfaces. At the same time, the accurate and efficient development efficiency of mesh models is an urgent desire of model developers, and accurate graphical descriptions of industrial pipelines and simplification of the working pipeline of industrial pipeline networks have become the focus of research in this field. Therefore, an improved spline-based method for projecting the position of pipeline profile points is proposed in this paper, which can extract curve control nodes from existing non-segmented pipeline meshes, thus providing users with an initial editable framework and completing the stitching of pipeline networks. The experimental results show that the method can be used for simulation modeling of various types of pipelines or similar equipment models in the industry. The construction framework and components are easy to edit, friendly to non-expert users, efficient and highly fault-tolerant.

Keywords: spline; profile point position calculation; profile offset curve; spline control node detection; procedural modeling