

改进边折叠算法的数字孪生模型 几何轻量化设计

孙 军¹,石晓东¹,孙诗华²

(1. 沈阳建筑大学机械工程学院,辽宁 沈阳 110168;2. 中国工业互联网研究院辽宁分院,辽宁 沈阳 110020)

摘 要 目的 基于经典的边折叠算法,提出一种提高数字孪生技术中虚实映射关系实时性与稳定性的数字孪生模型几何轻量化方法,解决数字孪生模型占用内存大的难题。**方法** 首先分析缝袋机器人码垛流水线的数字孪生模型的数据结构;其次基于经典的边折叠算法,引入二次误差度量作为边折叠代价,设计控制折叠因子,并运用多细节层次技术改进边折叠算法;最后设计四组对比实验进行测试。**结果** 简化后的顶点数量平均减少了50.14%;简化后的文件所占用的内存降低了79.08%;符合数字孪生模型的几何形状误差要求并且大大提高了场景动画的帧率。**结论** 笔者所提出的数字孪生模型几何轻量化方法大大减少了模型的几何图元,降低了模型文件所占用的内存,有效地改善了在数字孪生技术中虚拟产品和物理产品之间一一映射关系的实时性与稳定性差的问题,提升了场景实时渲染效果,保持了模型的细节特征和视觉效果。

关键词 数字孪生模型;几何轻量化;边折叠算法;二次误差度量算法;豪斯多夫距离

中图分类号 TP391.9

文献标志码 A

引用格式:孙军,石晓东,孙诗华.改进边折叠算法的数字孪生模型几何轻量化设计[J].沈阳建筑大学学报(自然科学版),2024,40(4):745-755. (SUN Jun,SHI Xiaodong,SUN Shihua. Geometric lightweight design of digital twin model based on improved edge folding algorithm[J]. Journal of Shenyang jianzhu university(natural science), 2024,40(4):745-755.)

Geometric Lightweight Design of Digital Twin Model Based on Improved Edge Folding Algorithm

SUN Jun¹,SHI Xiaodong¹,SUN Shihua²

(1. School of Mechanical Engineering, Shenyang Jianzhu University, Shenyang, China, 110168; 2. Liaoning Branch of China Industrial Internet Research Institute, Shenyang, China, 110020)

Abstract: This paper was proposed to analyze the reasons for the large memory usage of digital twin models, and propose a geometric lightweight method for digital twin models based on the

收稿日期:2022-10-16

基金项目:国家自然科学基金项目(52075348)

作者简介:孙军(1963—),男,教授,博士,主要从事先进数控理论与应用、数字化应用技术等方面研究。

classic edge collapse algorithm to improve the real-time performance and stability of the virtual-physical mapping relationship in digital twin technology. The author first analyzes the data structure of the digital twin model for the sewing bag robot palletizing production line. Secondly, based on the classic edge collapse algorithm, the quadratic error metric is introduced as the cost of edge collapse, and a control collapse factor is designed. Furthermore, the edge collapse algorithm is improved using multi-level of detail technology. Finally, four sets of comparative experiments are designed and conducted for testing. The results show that the number of vertices after simplification are count decreased by an average of 50.14%; The memory occupied by the simplified file is reduced by 79.08%; It meets the geometric shape error requirements of the digital twin model and greatly improves the frame rate of the scene animation. It is proved that the designed geometric lightweight method of digital twin model greatly reduces the geometric primitives of the model, reduces the memory occupied by the model file, effectively improves the real-time and stability issues of the one-to-one mapping relationship between virtual and physical products in digital twin technology, improves the real-time rendering effect of the scene, and maintains the details and visual effects of the model.

Key words: digital twin model; geometric lightweight; edge collapse algorithm; quadratic error measurement algorithm; Hausdorff distance

近年来,全球对于数字孪生技术的关注度持续提升,颇具权威性的IT研究与顾问咨询公司Gartner Group曾在2016—2017年连续将数字孪生技术列为当年的十大战略科技发展趋势之一^[1]。“数字孪生”是美国密歇根大学PLM研究中心的教授迈克尔格里夫斯博士最先提出的,他在2002年的一次演讲中首次提出了数字孪生是由物理产品、虚拟产品以及物理产品和虚拟产品之间的联系三部分组成^[2]。通过物理产品和虚拟产品之间的联系,数字孪生可以实现物理世界和信息世界的互联互通、信息交互^[3]。通过获取真实空间内物理产品的信息,可以建立相对应的数字孪生模型,从而构建出一个与真实空间完全一致的映射空间。在此映射空间中,可以借助实际生产数据模拟出物理实体在真实环境中的行为^[4],完成交互反馈、数据分析、产线优化和预测性维护等功能。数字孪生可以通过传感数据动态感知机械设备的故障与健康状态,乃至预测和判断其性能退化程度及可靠性,进而有针对性地开展管理与维修工作,从而能够有效提升机械设备安全性、可靠性及使用寿命^[5]。

现如今拥有大数据量的复杂装配实体不断涌现,物理产品的模型结构越来越复杂,产品零件数量多,其本身特征多,承载的信息量大,所占用的储存空间也越来越大,这不仅给计算机的内存容量和处理速度带来了新的挑战,同时也由于数字孪生模型数据量的增大使得模型加载速度降低且浏览不流畅^[6],从而大大影响了虚拟产品和物理产品之间一一映射关系的实时性与稳定性。笔者在研究一套缝袋机器人码垛流水线的数字孪生技术时,在构建虚拟空间的过程中发现:整套流水线的模型数量多、承载的信息量大,且其中不乏一些无用且重复的信息。这导致了虚拟空间中的数字孪生模型所占用的内存非常大,在实际应用过程中时常会出现虚拟产品不能与物理产品保持映射关系,滞后和卡顿等现象。因此,需要对数字孪生模型进行轻量化处理。

现阶段模型轻量化的处理手段主要是网格简化技术,先将模型进行网格划分,再通过减少三维模型的点、边和面等几何图元的数量来减少模型数据的存储,从而提升模型的传输和处理效率,以达到提升三维场景实时

渲染效率的目的。常用的网格简化技术主要有顶点聚类法、区域合并法、采样方法和增量式简化法等。其中增量式简化算法最容易实现,增量式简化算法可细分为顶点移除、边折叠和三角形折叠三种几何图元删除法,这类算法复杂度低,且易于控制,能保持模型的几何特征,简化效果比较理想。W. J. Schroeder等^[7]在1992年的SIGGRAPH会议上提出了顶点删除算法,该算法通过评价函数来剔除网格中被评价为不重要的顶点,减少模型的图元,从而达到简化模型的目的。但是,此算法会使网格形成空洞,随后又需要对空洞进行填充,使得整个简化过程非常麻烦。H. Hopp等^[8]在1993年提出了边折叠算法,该算法引入了一个全局函数,通过折叠网格中不重要的边来达到简化的目的,但该算法却丢失掉了模型的部分细节。1994年,H. Bernd^[9]提出了三角形剔除算法,通过一个评价函数来判定三角形网格的重要程度,随后对所有三角形按重要程度大小进行排序并剔除重要程度低的三角形;但是该算法和顶点删除算法一样造成了网格空洞,简化过程也很麻烦。由此可见,在增量式简化算法当中,边折叠算法的简化过程是相对比较简单且高效的。

在上述的三种经典模型简化算法相继被提出之后,众多学者基于经典算法提出了改进算法。Pan Zhigeng等^[10]将二次误差度量作为三角形折叠的代价,使得算法的简化效率进一步提升,但是该算法仍然丢失了模型的部分细节。Y. Mao等^[11]在二次误差的基础上,充分利用了顶点周围的几何信息改进了评价函数,保持了模型边缘的特征,大大提升了简化效果。熊仕勇等^[12]提出了基于半边折叠的多细节层次模型简化算法,该算法同样在二次误差度量的基础上,在评价函数中引入了曲率特征、视觉特征度和折叠程度等评价因子,有效降低了简化模型的几何误差,并且较好地保持了模型的几何细节特征。

基于上述研究,笔者分析了缝袋机器人

码垛流水线的数字孪生模型的数据结构,明确了数字孪生模型占用内存大的原因;再基于经典的边折叠算法,引入了二次误差度量作为边折叠代价,同时设计了控制折叠因子,并运用多细节层次技术改进边折叠算法,提出一种模型几何轻量化方法;最后通过实验验证所提出方法在四个方面的表现;研究表明:该方法大大提高了模型简化的效率,有效改善了在数字孪生技术中虚拟产品和物理产品一一映射关系的实时性与稳定性。

1 数字孪生模型

数字孪生模型都是由三维绘图软件建立的,而所有三维绘图软件所绘制的模型都是由基准、坐标系、建模历史、拓扑关系、特征定义和参数等数据构成。这些大量数据也造成了数字孪生模型结构复杂、信息繁多。因此,造成数字孪生模型占用内存大的最主要原因就是模型的零件包含很多并不影响模型外观的多余信息,并且模型的装配体在进行装配时,重复记录了零件的参数信息。

为了验证这一论点,笔者分别用三种文件保存格式将缝袋机器人码垛流水线Solidworks模型(见图1)的74个零件进行保存格式转换,其中.obj保存格式剔除了模型的建模历史,只以点和面的形式保存模型的外形数据,一定程度上释放了以原始格式保存的模型文件所占用的内存空间,.SLDPRT格式是模型的原始保存格式,.stl格式是一种剔除零件建模历史的二进制文件保存格式。对比结果如图2所示。

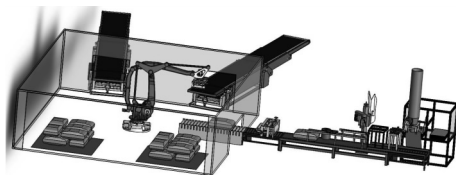


图1 缝袋机器人码垛流水线模型

Fig.1 The model of stacking assembly line with bag sewing robot

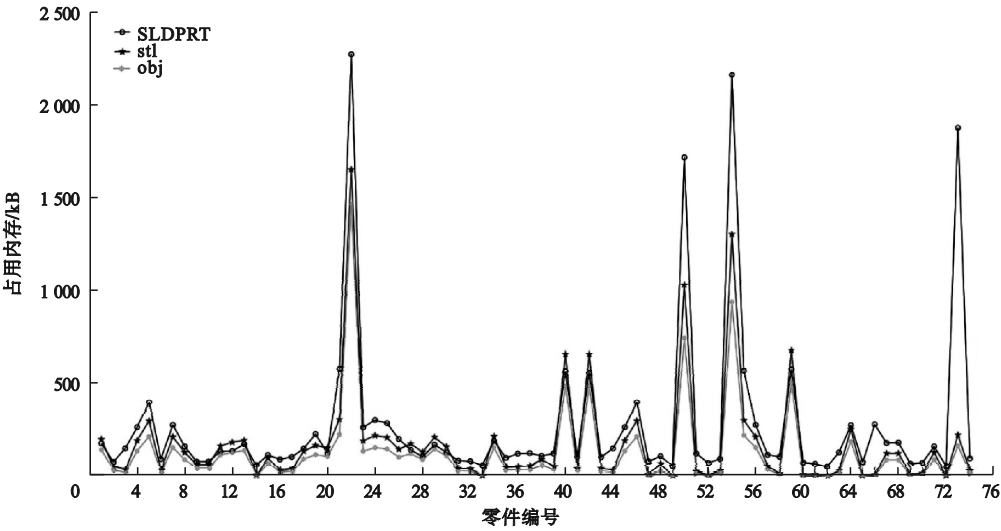


图2 三种文件格式的占用内存对比

Fig. 2 The memory usage comparison of three file formats

由图2可知,. stl格式与. obj格式的简化效果都很明显,尤其. obj文件格式无论模型的原始格式占用的内存是大是小,其简化效果都十分显著,保持着平均58.5%的简化效率,证明了造成数字孪生模型占用内存大的最主要原因就是模型的零件包含很多并不影响模型外观的多余信息。

2 改进的边折叠算法

在找到模型轻量化存储格式的基础上,设计一种改进的模型简化方法。笔者基于经典的边折叠算法,引入了二次误差度量作为

边折叠代价,同时设计了三种控制折叠因子,并运用多细节层次技术对边折叠算法进行了改进。

2.1 传统边折叠算法

传统的边折叠算法遵循这样的折叠准则:以边作为被删除的基本几何元素,将边的两个顶点合并成一个新顶点,同时删除与两个原始顶点相关的边并更新顶点与周围顶点的连接关系,如图3所示。图中, v_1 和 v_2 表示边的两个原始顶点, \bar{v} 表示边的新顶点。边折叠算法需要考虑边被删除的顺序和新的顶点的位置。

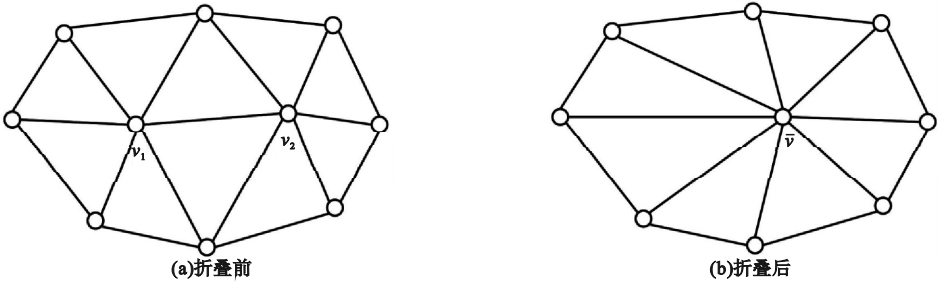


图3 边折叠算法原理

Fig. 3 The folding principle of edge folding algorithm

2.2 二次误差度量算法

二次误差度量算法最先由 M. Garland 和 P. S. Heckbert 提出^[13],使用该算法进行

模型简化可以得到高质量简化模型,并且该简化模型同时具有简化效率高及占用内存小的优点^[14]。

二次误差度量算法是将模型的三角形网格中每个三角形的三个顶点及其相关的所有三角形组成的一个该三角形及其相关三角形的集合 C 。在三维欧氏空间中,三角形及其相关三角形所在的平面方程表示为: $ax + by + cz + d = 0$ 。其中, d 表示任意常数,并且参数 a, b, c 满足 $a^2 + b^2 + c^2 = 1$,这个平面方程可以被定义为 $\mathbf{P} = [a, b, c, d]^T$,顶点 v 到三角形平面的距离平方可以被定义为

$$(\mathbf{P}^T \mathbf{v})^2 = (\mathbf{v}^T \mathbf{P})(\mathbf{P}^T \mathbf{v}) = \mathbf{v}^T (\mathbf{P} \mathbf{P}^T) \mathbf{v} = \mathbf{v}^T \mathbf{Q} \mathbf{v}. \tag{1}$$

其中, $\mathbf{P} \in C$; \mathbf{Q} 为误差矩阵; $(\mathbf{P}^T \mathbf{v})^2$ 为二次误差。

在边折叠算法中引入二次误差度量作为折叠代价,将会给每条边的各个原始顶点设置一个 4×4 的误差矩阵 \mathbf{Q} ,表示从顶点到相邻平面的距离平方和,则每个顶点的折叠误差可表示为

$$\begin{aligned} E(\mathbf{v}) = \mathbf{v}^T \mathbf{Q} \mathbf{v} = & q_{11}x^2 + q_{22}y^2 + q_{33}z^2 + \\ & 2q_{12}xy + 2q_{13}xz + 2q_{14}x + 2q_{23}yz + 2q_{24}y + \\ & 2q_{34}z + q_{44}. \end{aligned} \tag{2}$$

式中: $\mathbf{v} = [x, y, z, 1]^T$ 表示顶点位置; x, y, z 表示顶点坐标,结合式(1)顶点的误差矩阵 \mathbf{Q} 可以表示为

$$\mathbf{Q} = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ q_{41} & q_{42} & q_{43} & q_{44} \end{bmatrix} = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix}. \tag{3}$$

利用折叠误差 $E(\mathbf{v})$,可以很好地解决2.1节所提到的边折叠算法的两个问题,即:按照折叠误差大小来对边的原始顶点进行折叠的优先级排序,折叠误差较大的优先折叠;同时也以这个折叠误差来衡量新产生的顶点 \bar{v} 的位置好坏,新顶点的最优位置就是将这

个折叠误差控制到最小的位置。为了找到这个最优位置,对式(2)的折叠误差中的 x, y, z 求偏导数,令其为0。即:

$$\frac{\partial E(\mathbf{v})}{\partial x} = \frac{\partial E(\mathbf{v})}{\partial y} = \frac{\partial E(\mathbf{v})}{\partial z} = 0. \tag{4}$$

将得到线性方程组:

$$\begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{v} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \tag{5}$$

如果这个线性方程组有解,则这个解就是新顶点的最优位置,如果没有解,则从边的两个原始顶点以及边的中点当中选择能使式(2)的值最小的点作为折叠后的新顶点。

但是二次误差度量算法也有它的局限性。首先,由于模型边缘相邻的三角形网格较少,算法会在模型边缘产生较大的偏差;其次,二次误差度量算法并没有考虑曲面特性,比如法线、曲率等,这使得具有复杂曲面的模型在运用该算法后的简化效果差强人意。M. Garland 等^[15]在提出二次误差度量算法后也针对算法的缺陷进行了改进,但改进之后的误差矩阵的维数增多,使得算法的计算变得非常繁琐。为了弥补二次误差度量算法的缺陷,使得算法更加适合为数字孪生模型进行简化,笔者引入了三种控制折叠因子对二次误差度量算法的误差矩阵进行了改进。

2.3 控制折叠因子

2.3.1 顶点局部邻接区域面积因子

笔者引入顶点局部邻接区域面积因子,借助三角形网格的面积大小来判断模型的特征明显程度^[16]。一般情况下,网格模型在具有显著模型特征的区域三角形网格分布较密集,并且在该区域内的三角形网格面积会比较小;相反,在模型特征不明显的区域,三角形网格的分布比较稀疏并且面积较大。在模型特征比较明显的区域进行过多次数的边折叠后,会使得模型的特征细节有所丢失,这样就对模型的外形造成了影响甚至是改变。

为了在保持模型外观特征的情况下对模型进行简化,有必要将顶点局部邻接区域面积因子引入作为控制折叠因子。

顶点局部邻接区域由顶点周围的三角形围绕而成(见图4),顶点A的局部邻接区域由三角形 T_0 、 T_1 、 T_2 、 T_3 和 T_4 和组成。从图4中可以看出,顶点A的局部邻接区域面积是明显比顶点B和顶点C的小的,说明顶点A所在的区域是模型特征较明显的区域,如果优先对其进行折叠将会影响模型的外观特征。

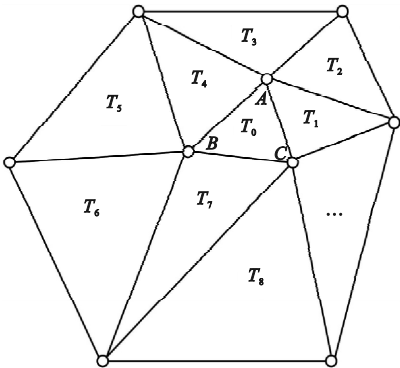


图4 顶点局部邻接区域

Fig. 4 Local adjacency region of the vertices

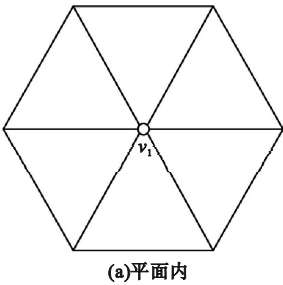
将顶点局部邻接区域面积因子表示为

$$S_v = W \frac{S_{\text{neighbour}}}{S_{\text{total}}} = W \frac{\sum_{T_i \in U(v)} S_{T_i}}{\sum_{T_i \in C(v)} S_{T_i}} \quad (6)$$

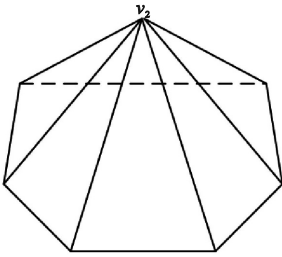
式中: S_v 为顶点 v 的顶点局部邻接区域面积因子; $S_{\text{neighbour}}$ 为顶点 v 的所有邻接区域三角形的面积之和; S_{total} 为顶点 v 所在平面的所有三角形的面积之和; W 为顶点局部邻接区域面积因子的权重,用来根据不同的模型外观来调节因子的控制力度。

2.3.2 顶点视觉重要度因子

顶点视觉重要度决定了模型顶点的凹凸几何特性,如图5所示。图5(a)中顶点 v_1 在平面的水平面内,而图5(b)中顶点 v_2 在平面的水平面之上。顶点 v_2 处体现了模型的凹凸几何特性,如果顶点 v_1 和 v_2 以相同的折叠准则进行折叠,势必会让模型的几何特征受到改变,消除模型的棱角,影响了模型的外观。因此,笔者将顶点视觉重要度因子引入作为控制折叠因子,来保持模型顶点的凹凸特性。



(a)平面内



(b)平面外

图5 顶点的凹凸特性

Fig. 5 Bump property of the vertices

将顶点视觉重要度因子表示为

$$V_v = \left\| \frac{\sum_{T_i \in C(v)} S_{T_i} \cdot \vec{n}_{T_i}}{\sum_{T_i \in C(v)} S_{T_i} \cdot \vec{n}} \right\| \quad (7)$$

式中: S_{T_i} 为顶点 v 所在平面的各个三角形面积; \vec{n}_{T_i} 为顶点 v 所在平面的各个三角形的单位法向量; \vec{n} 为平面内任意一个三角形的单位法向量。式(7)表明,当 V_v 趋近于1时,

说明所属平面的所有三角形的法向量相近,模型的该区域比较平坦,几何特征比较不明显,见图5(a)中的顶点 v_1 ;相反,当 V_v 趋近于0时,说明所属平面的所有三角形的法向量差距较大,模型的该区域凹凸不平,几何特征比较明显,见图5(b)中的顶点 v_2 。

2.3.3 顶点曲率因子

顶点曲率描述的是曲线上某一点的弯曲

程度,是衡量网格模型几何特征显著性的重要指标。顶点的曲率越大,弯曲程度越明显,模型的几何特性也就越显著。但是用一般的方法计算顶点曲率时,并没有考虑到顶点的邻域三角形形状对顶点法向量的影响,所以一般的方法并不适用于复杂模型的简化工作。T. Surazhsky 等^[17]和 E. Magid 等^[18]引入了一种 Gauss-Bonnet 定理来计算曲率的有效方法,该方法在计算曲率时考虑到了顶点的邻域三角形的影响,后来也有很多研究者参考这一方法,均取得了很好的效果。笔者参考 Gauss-Bonnet 定理,并将计算得到的曲率作为控制折叠因子之一,对模型的几何特征进行了保护。

Gauss-Bonnet 定理的一个重大贡献就是表述了曲面的曲率和拓扑之间的联系。定理指出:曲面上某个区域的高斯曲率的曲面积分称为总曲率,测地三角形(黎曼球面几何中的三角形)的总曲率等于它的曲面内角和与平面内角和的差,即:

$$\iint K dA = \oint d\theta - \sum_{i=1}^N \theta_i. \quad (8)$$

式中: K 为顶点的高斯曲率; dA 为曲面微元; N 为顶点的邻域三角形数量; θ_i 为与顶点相交的第*i*个三角形的角度; $\oint d\theta$ 在两种不同的情况下有不同的值,当顶点在网络的内部时, $\oint d\theta$ 值为 2π ;当顶点在网络的边缘时, $\oint d\theta$ 的值为 π 。

假设顶点的高斯曲率在局部邻域中为常数,则由式(8)可得:

$$K = \frac{3(\oint d\theta - \sum_{i=1}^N \theta_i)}{\sum_{i=1}^N S_i}. \quad (9)$$

其中, $\sum_{i=1}^N S_i$ 为顶点的邻域三角形面积之和。

顶点曲率因子可表示为

$$K_v = |K|. \quad (10)$$

引入控制折叠因子后的改进误差矩阵可表示为

$$Q^* = \frac{K_v}{S_v V_v} Q. \quad (11)$$

2.4 多细节层次技术

多细节层次(Levels of Detail, LOD)技术由 J. Clark 率先提出,是一种根据物理产品模型的节点在视觉环境中所处的位置及其重要度来决定模型渲染的资源分配,降低模型不重要部分的细节度,从而高效地对模型进行渲染,释放内存压力的技术。为了合理地分配数字孪生模型的渲染资源并有效维护其边界特征,笔者将多细节层级技术引入到改进的边折叠算法当中。层次分配方法如下:

(1)第一层次是模型的边界层次。笔者将模型边界处的控制折叠因子翻倍,将模型边界处的折叠优先级顺序延后,从而保持模型的边界特征。

(2)第二层次是模型看不见的部分,比如一个零件和另一个零件装配后被遮挡的部分、模型和地面接触而看不到的部分。笔者先将这一层次的控制折叠因子缩减为原来的十分之一,再将这一层次的边折叠算法迭代次数增多,尽量多的删除掉这些看不见部分的几何元素,从而增强简化效果,提高渲染效率。

(3)第三个层次就是模型剩下的部分。在这一层次中,既不放大改进的误差矩阵,也不增多算法的迭代次数,按照上述设计的改进边折叠算法进行简化即可。

2.5 改进的模型几何轻量化流程

改进的模型几何轻量化流程如图6所示,其实现过程如下:

Step 1 将模型文件转化为.obj文件格式;

Step 2 设置好算法的迭代次数;

Step 3 遍历各个三角形网格,计算其面积和法向量并分类保存;

Step 4 计算各个顶点的控制折叠因子

和误差矩阵;

Step 5 对各个顶点进行层次判断,如果是看不见的点,则将控制折叠因子翻2倍,并将迭代次数扩大1.5倍;如果是边界点,则将控制折叠因子缩减为原来的1/10;其他情况则不做任何附加操作;

Step 6 计算各个顶点对的改进误差矩阵;

Step 7 计算各点对的折叠代价,并从大到小进行排列,形成一个点对的集合;

Step 8 取出集合中代价最大的点对进行边折叠,确定新顶点位置;

Step 9 更新顶点,生成新的三角形网格;

Step 10 判断是否达到迭代次数,如果没有达到迭代次数则回到 Step 8,如果达到了则进行 Step 11;

Step 11 生成并输出新的网格模型。

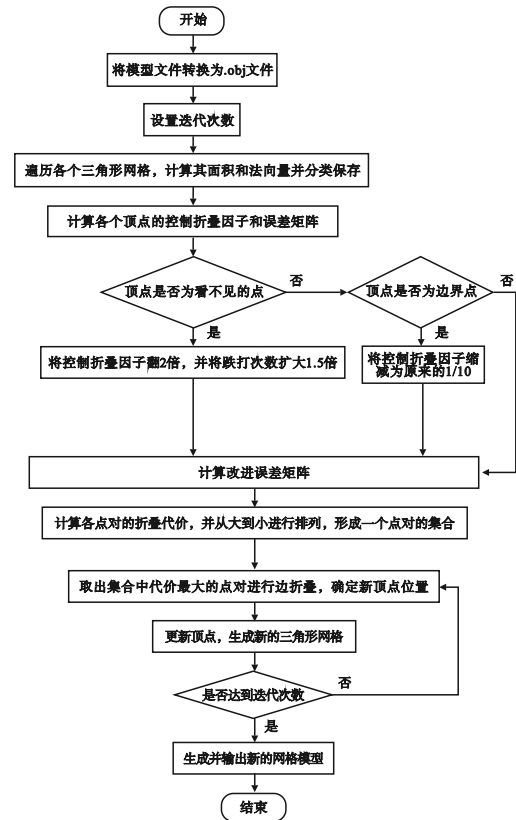


图6 模型几何轻量化流程

Fig. 6 The lightweight process of model geometry

3 基于几何形状的误差度量方法

在对原始的数字孪生模型进行几何轻量化之后不可避免的会产生简化误差,而如何度量简化误差也是模型轻量化研究的关键技术。由于对数字孪生模型进行几何轻量化的目的是为了解决在数字孪生技术中虚拟产品和物理产品的一一映射关系的实时性与稳定性差的问题,所以数字孪生模型的简化误差大小可以从几何形状方面度量。

笔者引用了一种基于几何形状的误差度量方法来度量数字孪生模型的简化误差。该方法从模型的几何形状出发,引用了豪斯多夫距离公式来衡量原始模型和简化模型之间的形状误差。

豪斯多夫(Hausdorff)距离^[19]定义:给定欧氏空间中的两个点集 $A = \{a_1, a_2, \dots\}$, $B = \{b_1, b_2, \dots\}$, 则豪斯多夫距离表示为

$$H(A, B) = \max[h(A, B), h(B, A)]. \quad (12)$$

$$h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\|. \quad (13)$$

$$h(B, A) = \max_{b \in B} \min_{a \in A} \|b - a\|. \quad (14)$$

豪斯多夫距离可以用来测量两个模型之间的最大形状偏差,并且其度量值比较准确、全面,但是其计算比较复杂。笔者在进行误差度量时,只针对模型几何细节特征明显、视觉重要度高的部分运用豪斯多夫距离公式进行误差度量,既节省了计算量,又保证了误差度量值的有效性。

4 实验

通过四组对比实验验证所设计方法的有效性。这四组对比实验分别从模型简化效果、文件大小、几何形状误差以及帧率进行了对比分析。实验设备是一台具备 Windows 11 系统、AMD Ryzen 5 5500U 六核处理器、2.10 GHz、16 GB 内存、AMD Radeon (TM) Graphics 显卡的计算机。实验对象是缝袋机器人码垛流水线的10组数字孪生模型。本次试验将前文所述的顶点局部邻接区域面积因

子的权重 W 分为两个档次:对于传送带、机械臂、原料库和滚筒流水线这类形状较为复杂的模型, W 设为 1;对于缝纫机、围栏、袋子和成品堆这类形状不复杂的模型, W 设为 0.5。算法的迭代次数:传送带(15 000 次),机械臂(15 000 次),缝纫机(1 500 次),滚筒流水线一(3 000 次),滚筒流水线二(1 000 次),滚筒流水线三(5 000 次),围栏(10 次),原料库

3 500 次,袋子(10 次),成品堆(100 次)。

将 10 组数字孪生模型导入 3ds Max 软件中,方便显示模型的三角形网格数量,用笔者提出的简化方法对模型进行几何轻量化,如图 7 所示,其中 Q 表示简化前,H 表示简化后。对比简化前后的三角形网格数量与顶点数量、文件大小和几何形状误差,如表 1 所示。

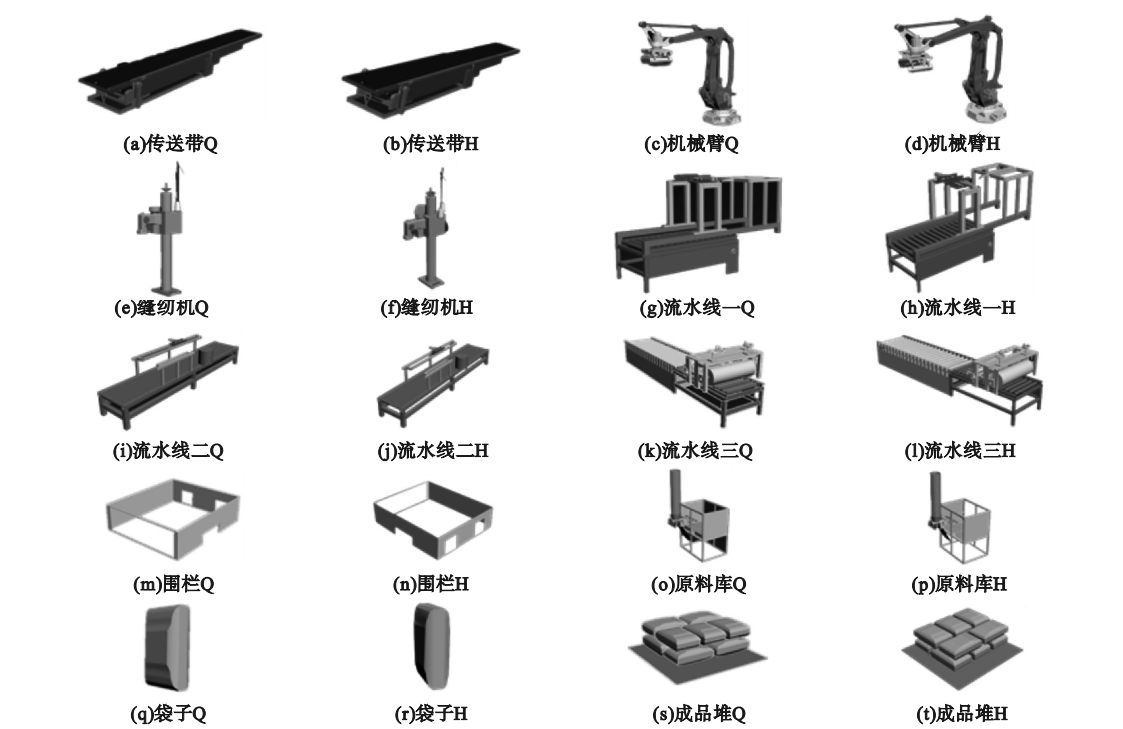


图 7 模型简化前后的对比效果图

Fig. 7 The effect comparison before and after model simplification

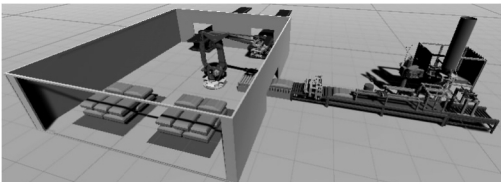
表 1 模型简化前后的对比分析

模型	顶点数量/个		顶点简化率/%	简化前文件大小/kB	简化后文件大小/kB	文件简化率/%	最大误差/cm	最大误差占模型最大尺寸百分比/%
	简化前	简化后						
传送带	100 300	50 141	50.01	23 781	4 035	83.03	3.29	0.41
机械臂	117 156	59 086	49.57	26 782	4 640	82.67	2.31	0.67
缝纫机	11 381	5 975	47.50	2 464	432	82.47	1.35	0.67
流水线一	35 781	18 829	47.38	7 474	1 473	80.29	4.59	1.96
流水线二	8 292	4 146	50.00	1 446	312	78.42	3.40	0.95
流水线三	86 071	64 604	24.94	25 257	5 256	79.19	3.61	1.08
围栏	78	27	65.38	6	2	66.67	6.28	0.62
原料库	41 648	22 248	46.58	7 454	1 694	77.27	5.17	1.47
袋子	120	36	70.00	17	3	82.35	0.76	0.98
成品堆	1 208	604	50.00	195	42	78.46	2.59	0.92

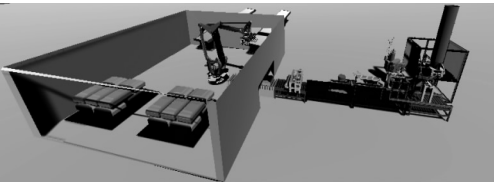
由图7可以看出,模型在使用笔者所设计的简化方法简化之后,并没有失去模型应该保存的外观特征,模型的可用性仍然非常好。由表1可知,从模型的简化效果来看,简化后的顶点数量大大减少,平均减少了50.14%,极大程度地减少了模型的几何图元;从文件大小来看,简化后的文件所占用的内存大大降低,平均降低了79.08%,极大程度地释放了文件所占用的内存;从几何形状误差来看,简化后的误差大小参差不齐,对大型、复杂模型进行简化后,产生的几何形状误差较大,对较小的模型进行简化后产生的误差较小,但所有模型简化后的最大误差占模型最大尺寸的百分比都小于5%,结合陶飞团队^[20]所提出的“数字孪生模型的轻量化准则是为了在数字孪生模型可用、通用的基础

上,通过数字孪生模型速用,进一步满足针对复杂系统的数字孪生建模和模型运行的高效性需求。”的理论,可以得出运用笔者所提出的简化方法简化后的模型符合数字孪生模型的误差要求。

笔者将简化前后的数字孪生模型分别导入了Unity3D引擎当中,生成了两个场景,这两个场景都按照流水线的布局情况进行设置,只是分别采用了简化前和简化后的模型,如图8所示。笔者分别记录了在两种不同的场景下的动画帧率,如图9所示。使用简化前模型的场景动画帧率保持在32.3FPS左右,而使用简化后模型的场景动画帧率保持在69.3FPS左右,对比结果表明模型简化后大大提升了场景的实时渲染效果。



(a)简化前模型搭建的场景



(b)简化后模型搭建的场景

图8 两个 Unity3D 场景
Fig. 8 Two scenes in Unity3D

Statistics	
Audio:	
Level: -74.8 dB	DSP load: 0.3%
Clipping: 0.0%	Streamload: 0.0%
Graphics:	32.3 FPS(31.0ms)
CPU: main 31.0ms renderthread161ms	
Batches: 4714 Saved by batching: 2230	
Tris: 5.6M Verts: 4.6M	
Screen: 801x266-2.4MB	
SetPasscalls: 218	Shadowcasters: 4445
Visible skinned meshes: 0	Animations: 0

(a)简化前模型场景帧率

Statistics	
Audio:	
Level: -74.8 dB	DSP load: 0.3%
Clipping: 0.0%	Streamload: 0.0%
Graphics:	69.3FPS(14.4ms)
CPU: main 14.0ms render thread 12.9ms	
Batches: 2979 Saved by batching: 1785	
Tris: 3.1M Verts: 3.2M	
Screen: 801x266-2.4MB	
SetPasscalls: 206	Shadowcasters: 3043
Visible skinned meshes: 0	Animations: 0

(b)简化后模型场景帧率

图9 两个场景下的动画帧率

Fig. 9 Animation frame rates in two scenes

5 结 论

(1)通过对比缝袋机器人码垛流水线的74个零件模型在不同文件保存格式下占用内存的大小,证明了造成数字孪生模型占用内存大的最主要原因就是模型的零件包含很多并不影响模型外观的多余信息。

(2)基于经典的边折叠算法,笔者设计了一种改进的模型简化方法,该方法可以显著减少模型几何图元的数量,降低模型文件所占用的内存,从而有效地解决了在数字孪生技术中虚拟产品和物理产品之间一一映射关系的实时性与稳定性差的问题。

参考文献

- [1] 肖庆阳, 方建儒, 王龙, 等. 基于数字孪生的全周期智慧车间系统[J]. 组合机床与自动化加工技术, 2021(10): 129–132.
(XIAO Qingyang, FANG Jianru, WANG Long, et al. Full cycle intelligent workshop system based on digital twin [J]. Modular machine tool and automatic processing technology, 2021(10): 129–132.)
- [2] YANG Y, MENG W, ZHU S. A digital twin simulation platform for multirotor UAV [C]//The 7th international conference on information, cybernetics, and computational social systems (ICCSS). Guangzhou, China: IEEE, 2020: 591–596.
- [3] TAO F, QI Q. New IT driven service oriented smart manufacturing: framework and characteristics [J]. IEEE transactions on systems, man, and cybernetics (systems), 2017, 49(1): 81–91.
- [4] 李澍, 郭金刚, 张立洲, 等. 运载火箭发射场数字合练技术[J]. 导弹与航天运载技术, 2019(2): 22–26.
(LI Shu, GUO Jingang, ZHANG Lizhou, et al. Digital training technology for launch vehicle launch sites [J]. Missile and space launch technology, 2019(2): 22–26.)
- [5] 彭宇, 刘大同, 彭喜元. 故障预测与健康管理技术综述[J]. 电子测量与仪器学报, 2010, 24(1): 1–9.
(PENG Yu, LIU Datong, PENG Xiyuan. Overview of fault prediction and health management technology [J]. Journal of electronic measurement and instrumentation, 2010, 24(1): 1–9.)
- [6] 陈龙, 郭军, 张建中. 三维模型轻量化技术[J]. 工矿自动化, 2021, 47(5): 116–120.
(CHEN Long, GUO Jun, ZHANG Jianzhong. Three dimensional model lightweight technology [J]. Industrial and mining automation, 2021, 47(5): 116–120.)
- [7] SCHROEDER W J, ZARGE J A, LORENSEN W E. Decimation of triangle meshes [J]. ACM siggraph computer graphics, 1992, 26(2): 65–70.
- [8] HOPPE H, DEROSE T, DUCHAMP T, et al. Mesh optimization [C]//Proceedings of the 20th annual conference on computer graphics and interactive techniques. Washington, D. C.: IEEE, 1993: 19–26.
- [9] BERND H. A data reduction scheme for triangulated surfaces [J]. Computer aided geometric design, 1994, 11(2): 197–214.
- [10] PAN Zhigeng, ZHOU Kun, SHI Jiaoying. A new mesh simplification algorithm based on triangle collapses [J]. Journal of computer science and technology, 2001, 16(1): 57–63.
- [11] MAO Y, YANG J, ZHU B, et al. A new mesh simplification algorithm based on quadric error metric [C]//The 5th international conference on consumer electronics-Berlin (ICCE-Berlin). Berlin, Germany: IEEE, 2015: 463–466.
- [12] 熊仕勇, 李沁翰, 谢爱荣. 基于半边折叠的LoD三维模型简化算法[J]. 计算机工程与设计, 2016, 37(3): 655–659.
(XIONG Shiyong, LI Qinhan, XIE Airong. Simplification algorithm of LoD three-dimensional model based on half edge folding [J]. Computer engineering and design, 2016, 37(3): 655–659.)
- [13] GARLAND M, HECKBERT P S. Surface simplification using quadric error metrics [C]//Proceedings of the 24th annual conference on computer graphics and interactive techniques. Los Angeles, United States: ACM, 1997: 209–216.
- [14] 王蕊, 李俊山, 刘国庆, 等. 一种保持视觉特征的三维几何模型简化算法[J]. 光电工程, 2008(7): 63–67.
(WANG Rui, LI Junshan, LIU Guoqing, et al. A simplified algorithm of 3D geometric model keeping visual features [J]. Optoelectronic engineering, 2008(7): 63–67.)
- [15] GARLAND M, HECKBERT P S. Simplifying surfaces with color and texture using quadric error metrics [C]// Visualization. Los Angeles, United States: IEEE, 1998: 263–269.
- [16] CHEN Y, LIU C. Quadric surface extraction using genetic algorithms [J]. Computer-aided design, 1999, 31(1): 101–110.
- [17] SURAZHISKY T, MAGID E, SOLDEA O, et al. A comparison of Gaussian and mean curvatures estimation methods on triangular meshes [C]// IEEE international conference on robotics & automation. Cincinnati, United States: IEEE, 2003: 1021–1026.
- [18] MAGID E, SOLDEA O, RIVLIN E. A comparison of Gaussian and mean curvature estimation methods on triangular meshes of range image data science direct [J]. Computer vision and image understanding, 2007, 107(3): 139–159.
- [19] PREPARATA F P, SHAMOS M I. Computational geometry: an introduction [M]. New York: Springer Verlag Press, 1985.
- [20] 陶飞, 张贺, 戚庆林, 等. 数字孪生模型构建理论及应用[J]. 计算机集成制造系统, 2021, 27(1): 1–15.
(TAO Fei, ZHANG He, QI Qinglin et al. Theory and application of digital twin model construction [J]. Computer integrated manufacturing systems, 2021, 27(1): 1–15.)
(责任编辑: 刘春光 英文审校: 范丽婷)