

引文格式: 陆凯, 李迎光, 刘旭, 等. 数据驱动飞机结构件加工特征识别方法[J]. 航空制造技术, 2021, 64(8): 32-40, 57.

LU Kai, LI Yingguang, LIU Xu, et al. A data-driven method for machining feature recognition for aircraft structural parts[J]. Aeronautical Manufacturing Technology, 2021, 64(8): 32-40, 57.

数据驱动的飞机结构件加工特征识别方法*

陆凯¹, 李迎光¹, 刘旭², 邓天池¹

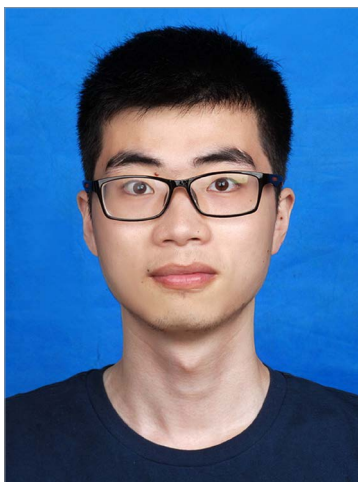
(1. 南京航空航天大学机电学院, 南京 210016;

2. 南京工业大学机械与动力工程学院, 南京 210009)

[摘要] 加工特征识别技术是实现 CAD/CAM 一体化的重要途经, 对提升飞机结构件数控加工质量和效率具有重要意义。飞机结构件结构复杂, 包含大量复杂相交特征, 现有加工特征识别方法识别相交特征时需要预先定义特征相交模式, 难以满足飞机结构件复杂相交特征识别需求。因此, 提出了数据驱动的飞机结构件加工特征识别方法, 将加工特征识别问题转化为图学习问题, 通过构建图神经网络模型从历史工艺数据中自适应学出特征识别规则, 打破了特征相交模式预先定义思路的局限。以典型飞机结构件为测试对象, 孤立特征、相交特征和总的加工特征正确识别率分别达到了 98.11%、94.62% 和 96.18%, 验证了所提方法的有效性。

关键词: 数据驱动; 飞机结构件; 加工特征识别; 相交特征; 图神经网络

DOI: 10.16080/j.issn1671-833x.2021.08.032



陆凯

硕士研究生, 主要研究方向为数字化设计与制造。

框、梁、壁板和接头等飞机结构件通常由数控加工的方式进行制造, 加工质量和效率将直接影响整机性能和生产研制周期。传统的数控加工工艺决策以人工交互模式为主, 由于飞机结构件结构复杂且加工精度要求高, 工艺决策时需手动提取几何形状信息构造加工区域, 不仅重复工作量大、效率低, 而且极易出错。加工特征是具有特定加工工程语义的几何形状, 将加工特征与加工区域构造规则关联, 便可通过加工特征识别技术从零件中自动提取加工特征信息, 进而实现加工区域自动构造。典型的特征识别方法主要有基于图的

特征识别方法^[1]和基于痕迹的特征识别方法^[2]。

基于图的特征识别首先将零件表示成图, 然后将图分解为特征子图, 并利用子图同构算法将特征子图和预定义特征进行匹配。Joshi 等^[3]首次提出了基于 AAG 图的特征识别方法, 通过子图匹配能够高效地识别孤立的特征。然而当多个加工特征相互耦合或相交时, 这些加工特征对应的特征子图不再拥有完整的拓扑边界, 导致从 AAG 分解出的特征子图会缺失了部分节点和边, 造成分解出的特征子图和模板特征不匹配, 这使得相交特征的识别成为了一个

* 基金项目: 国家杰出青年科学基金(51925505)。

难题。对此,虚链接技术得到了大量的研究和发展^[4-8],其核心思想是根据不同特征相交模式制定特定规则,从而重新建立缺失节点和特征子图之间的链接关系,恢复特征子图缺失的节点。然而由于复杂零件相交特征的模式十分庞杂,无法事先完全定义,因此复杂零件相交特征的识别仍然是一个难题^[9]。

基于痕迹的特征识别方法首先定义特征的痕迹,痕迹是一种可以用于推理从而反求出完整特征的线索,且不会因为特征相交而被破坏,常见的痕迹有几何线索^[10]和非几何线索^[11]两种;然后根据痕迹的定义规则构建痕迹生成器,进而从零件模型中提取出所有可能的特征痕迹;最后基于几何推理对痕迹进行补充,得到完整的加工特征。Vandenbrande和Requicha^[12]首次提出了基于痕迹的特征识别方法,能够补充槽、孔等加工特征,但存在痕迹冗余的问题。Gao等^[13]以最小条件子图(MCSG)作为痕迹,并通过图分解生成所有的MCSG,每个生成的MCSG都对应了一个加工特征,从而避免了痕迹冗余。程少杰等^[14]基于“层”定义槽腔特征的痕迹,并利用分层槽腔特征相交的特点,建立几何推理规则补充“层”痕迹,得到完整的槽腔特征。

Wang等^[15]按特征相交改变拓扑方式定义了四类相交特征,并基于特征面搜索规则生成并补充相交特征痕迹。基于痕迹的特征识别方法极大地提升了相交特征识别能力,但由于痕迹的生成和补充算法依赖预先定义的特征相交模式来制定几何推理规则,因而仍难以解决复杂零件的特征相交问题。

对于飞机结构件等复杂零件而言,不同加工特征相交的方式是任意的,因而特征相交的模式几乎是无限的。因此上述方法虽然能在一定程度上识别特定的相交特征,但由于需要预先定义特征相交的模式,难以解决复杂零件的特征相交问题。针对上述情况,本文提出了一种数据驱动的加工特征识别方法,不采用预先定义相交特征模式的思路,而是从历史工艺数据中自适应地学习特征识别规则,来实现对复杂零件相交特征的识别。

数据驱动的加工特征识别方法

本文将加工特征识别问题转化为图学习问题,进而通过构建图神经网络模型从历史工艺数据中学习出特征识别规则。所提方法是一种通用的加工特征识别方法,适用于框、

梁、壁板等多种类型的飞机结构件的特征识别。方法的流程如图1所示,主要包括4个步骤:

Step1: 零件模型的属性图表示。

通过将零件模型表示为属性图,加工特征识别问题则转化为图分解的问题。

Step2: 基于图神经网络的加工特征分离。

建立图神经网络模型 GNN_{split} 将零件属性图分解为孤立特征子图和相交特征子图,其中孤立特征子图直接作为特征识别的结果,而相交特征子图则要要进一步识别。

Step3: 基于图神经网络的特征痕迹生成。

建立图神经网络模型 GNN_{hint} 将Step2得到的相交特征子图分解为一系列痕迹特征子图,每个痕迹特征子图对应了一个加工特征的痕迹。

Step4: 基于神经网络的特征痕迹补充。

建立神经网络模型 NN_{in} 对Step3得到的痕迹特征子图进行痕迹补充,从而得到完整的加工特征子图。

1 基于属性图的零件模型表示

属性图 G 的定义为: $G=(V, E, H)$,其中 V 是图中节点的集合, E 是图中边的集合,反映了节点之间的连接关系, $H=\{h_1, h_2, \dots, h_n\}$,

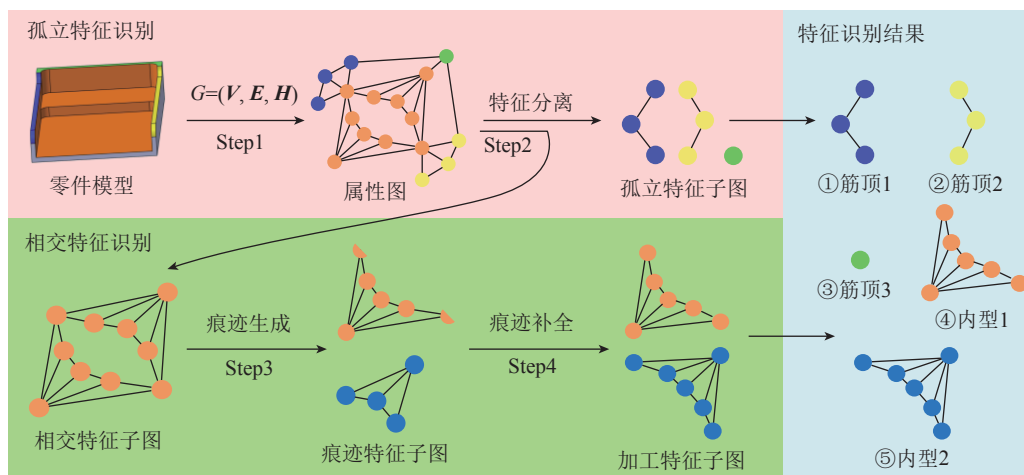


图1 总体流程

Fig.1 Workflow of the proposed method

$H \in \mathbb{R}^{(F \times n)}$, 表示属性矩阵, 它的列向量 h_i 表示第 i 个节点 v_i 的属性向量, $v_i \in V$, F 是节点属性的维度, n 是节点的数量。在计算机存储中, 通常使用邻接矩阵 $A \in \mathbb{R}^{(n \times n)}$ 存储 V 和 E 的关系, 第 i 行 j 列的矩阵元素 A_{ij} 为:

$$A_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \\ 0 & \text{if } (v_i, v_j) \notin E \end{cases} \quad (1)$$

其中 (v_i, v_j) 表示节点 v_i 和 v_j 的连接关系。通过邻接矩阵 A 和属性矩阵 H , 就可以唯一地构建属性图。

对于一个零件模型而言, 零件上所有的面组成了节点的集合 V , 所有的边组成了边的集合 E , 邻接矩阵 A 则由零件上面-面连接关系确定, 如图 2 所示, 黄、绿、紫 3 种颜色节点对应了 3 个筋顶特征, 橙色节点对应了一个槽腔特征, 该槽腔特征由两层的内型特征组成。而属性矩阵 H 则由零件模型中所有面的属性向量组成, 每个属性向量存储了一个面的几何和拓扑信息, 如面的法向量、面类型等, 典型的面属性如表 1 所示。

2 基于图的加工特征识别问题定义

当零件模型表示为属性图之后, 零件中的加工特征则对应了属性图的子图, 称为加工特征子图。图 2 (b) 的属性图包含了 5 个加工特征子图, 分别为 $\{r_1, r_2, r_3\}, \{r_4\}, \{r_5, r_6, r_7\}, \{s_1, s_2, s_3, s_4, i_1, i_2\}, \{i_1, i_2, s_5, s_6, s_7, s_8\}$, 如图 3 所示。加工特征子图按特征是否相交来划分, 可以分为孤立特征子图和相交特征子图。其中孤立特征子图由一个加工特征子图组成, 其拓扑结构不会受到其他加工特征的影响, 因而拥有完整的拓扑边界, 如 $\{r_1, r_2, r_3\}, \{r_4\}, \{r_5, r_6, r_7\}$; 而相交特征子图由两个以上的加工特征子图相交而成, 如 $\{s_1, s_2, s_3, s_4, i_1, i_2\} \cup \{i_1, i_2, s_5, s_6, s_7, s_8\}$, \cup 表示求并集, i_1, i_2 是公共节点。存在公共节点是特征相交的主要原因, 这往往导致加工特征子图失去完整的拓扑边界。

基于图的加工特征识别主要思

路为: 把属性图分解为一个个加工特征子图。以识别图 3 的加工子图为例, 通过合理地删除图 2 (b) 属性图中的一些边, 可以很容易地从属性图分解出图 3 的①②③三个加工特

征子图, 但无法直接得到图 3 中④和⑤两个加工特征子图, 因为④和⑤都需要节点 i_1 和 i_2 , 而通过删除边的方式最多只能保证④⑤中的一个加工特征子图得到节点 i_1 和 i_2 , 另一个加

表 1 典型的面属性
Table 1 Typical face attributes

序号	属性	类型	维度
1	面的法向量	浮点型	3
2	面类型	整型	3
3	面是否有环	整型	1
4	面积周长比	浮点型	1
5	距底面的最大/最小高度	浮点型	2
6	节点的度(边的数量)	整型	1
7	面与面凹/凸连接数量	整型	2
8	面与面相切/不相切数量	整型	2
9	面中直线/圆弧/自由曲线的数量	整型	3
10	面与面相切且凹连接的数量	整型	1
11	面与面相切且凸连接的数量	整型	1
12	面与面不相切且凹连接的数量	整型	1
13	面与面不相切且凸连接的数量	整型	1

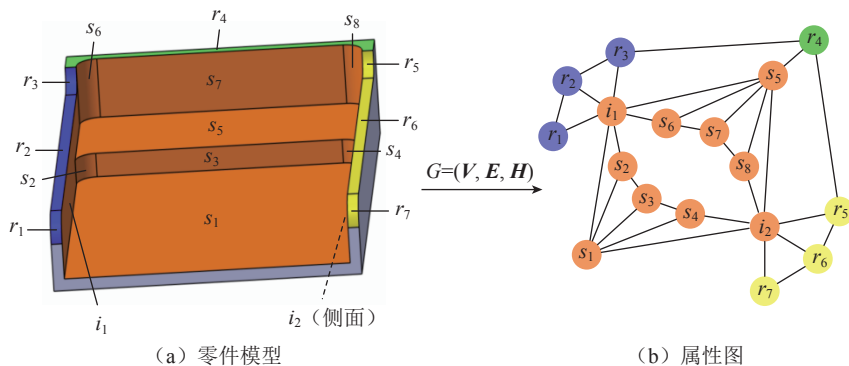


图 2 零件模型的属性图表示

Fig.2 Part model representation based on attributed graph

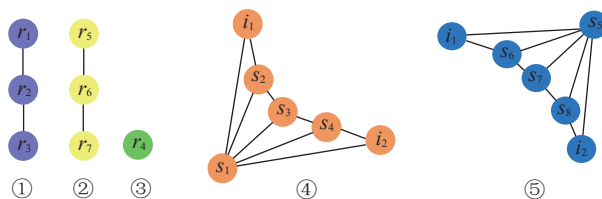


图 3 加工特征子图

Fig.3 Subgraphs corresponding to machining features

工特征子图将缺失 i_1 和 i_2 两个节点, 本文称这种缺失节点的加工特征子图为痕迹特征子图, 它以子图的形式保留了原来完整加工特征存在的几何线索, 该线索可以用来补全痕迹特征子图中缺失的节点, 从而得到完整的加工特征子图。

本文属性图的分解过程分为了两个阶段: 第一阶段, 只从属性图分解出孤立特征子图, 而保证剩下的相交特征子图的完整性, 实现孤立特征子图和相交特征子图分离。第二阶段, 再从相交特征子图分解出痕迹特征子图, 以此作为完整特征留下的几何线索对痕迹特征子图缺失的节点进行补全, 得到完整的加工特征子图。

3 基于图神经网络的加工特征分离

加工特征分离的主要任务是将属性图分解为孤立特征子图和相交特征子图, 如图 4 所示。图 4 (a) 是零件属性图, 其中黄、绿、蓝 3 类节点对应了 3 个孤立特征子图, 橙色节点对应了一个相交特征子图, 图上的边被分为“1”和“0”两种类别, “1”号表示需要打断的边, “0”号表示需要保留的边。图 4 (b) 是打断“1”号类型边之后的分离结果, 实现了加工特征分离。

因此, 通过对边进行打断/保留二分类即可实现加工特征分离。然而传统基于凸边打断规则的图分解算法无法满足分离孤立特征和相交特征的要求, 因为相交特征子图的内部存在凸边, 打断其内部的凸边破坏了相交特征子图的完整性。本文建立了基于图神经网络的边二分类模型(GNN_{split}), 使用带有边分类标签的样本数据进行训练, 从而学出特征分离所需的边打断/保留的分类规则。 GNN_{split} 模型由聚集层、表示层以及密集层组成, 模型的输入为组成零件属性图的邻接矩阵 A 和属性矩阵 H 。属性矩阵 H 首先输入到 GNN_{split} 模型, 在聚集层根据邻接矩

阵 A 让每个节点聚集其邻居节点的属性, 在表示层使用聚集后的节点属性来表示边的属性, 最后在密集层将边的属性向量映射到边的二分类别。 GNN_{split} 模型的结构如图 5 所示。

3.1 聚集层

聚集层的主要任务是实现邻居节点属性的聚集, 属性聚集是学习属性图中拓扑信息的重要方式。通常一个图神经网络模型通常会构建 2~3 个聚集层从而充分学习拓扑信息。对于每一个聚集层, 属性图上的每个节点都会吸收其邻居节点的属性一次, 以聚集节点 v 的邻居属性为例:

$$h_{N(v)}^k = AGG\left(\left\{h_u^{k-1}, \forall u \in N(v)\right\}\right) \quad (2)$$

式中, k 表示第 k 个聚集层, 即第 k 次属性聚集。 $h_{N(v)}^k$ 是节点 v 的邻居聚集向量, 它聚集了节点 v 的邻居属性, $N(v)$ 是节点 v 的邻居节点的集合, h_u^{k-1} 表示邻居节点 u 经过 $k-1$ 次聚集后的属性向量, 特别地, 当 $k=1$ 时, $h_u^{k-1}=h_u$ 是初始输入的属性向量, $h_u \in H, H \in \mathbb{R}^{F \times n}$ 。AGG 是聚集函数, 决定了节点 v 聚集邻居节点属性的方式, 本文使用了注意力聚集函数来实现自适应聚集, 即, 它使用注意力系数决定每个邻居属性所占的比例,

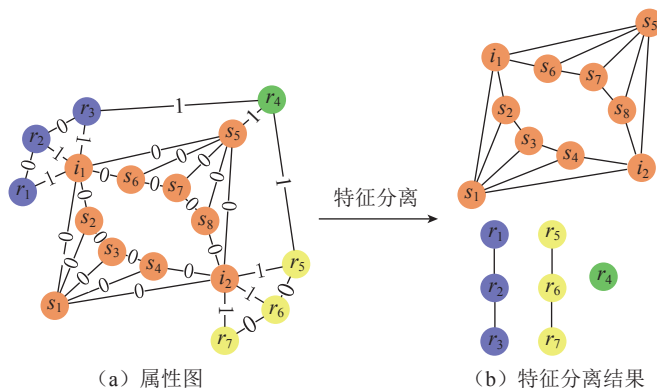


图 4 加工特征分离

Fig.4 Separation of machining features

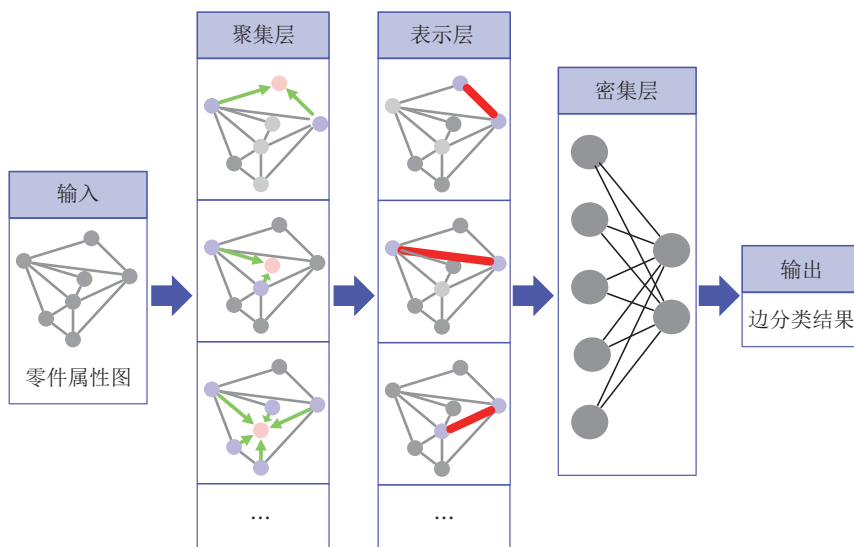


图 5 图神经网络模型结构

Fig.5 Architecture of the graph neural network

并且在模型的训练迭代中不断更新注意力系数^[16]。注意力系数计算方式如下:

$$e_{vu} = \mathbf{a}^T (\mathbf{W} \mathbf{h}_v \parallel \mathbf{W} \mathbf{h}_u) \quad (3)$$

式中, \parallel 表示向量拼接, $u \in N(v)$ 是节点 v 的邻居节点, $\mathbf{a}^T \in \mathbb{R}^{1 \times 2F}$ 是可学习的参数向量, $\mathbf{W} \in \mathbb{R}^{F \times F}$ 是可学习的参数矩阵, 用于对输入向量做线性变换, $\mathbf{W} \mathbf{h}_v \in \mathbb{R}^{F \times 1}$, $\mathbf{W} \mathbf{h}_u \in \mathbb{R}^{F \times 1}$, F' 是线性变换后属性维度。然后利用 softmax 函数对所有邻居的注意力系数 e_{vu} 做规范化处理:

$$\alpha_{vu} = \frac{\exp(e_{vu})}{\sum_{i \in N(v)} \exp(e_{vi})} \quad (4)$$

其中, α_{vu} 是规范化的注意力系数, 决定了多少比例吸收邻居 u 属性, α_{vu} 的值在 0~1 之间, 且 $\sum_{u \in N(v)} \alpha_{vu} = 1$ 。

将式(4)实现的注意力聚集函数带入式(2):

$$\mathbf{h}_{N(v)}^k = \sum_{u \in N(v)} \sigma(\alpha_{vu}^k \mathbf{W}^k \mathbf{h}_u^{k-1}) \quad (5)$$

式中, σ 是激活函数, α_{vu}^k 是第 k 个聚集层的节点 v 对该节点邻居 u 进行规范化的注意力系数, \mathbf{W}^k 是聚集层 k 的参数矩阵, \mathbf{h}_u^{k-1} 是节点 u 在 $k-1$ 层的属性向量。

式(5)只聚集一次邻居节点的属性向量, 称为单头部聚集。有时候为了更好的学习效果, 常使用多头部聚集, 即每层对每个节点聚集邻居节点的属性多次, 进而得到多组邻居聚集向量, 最后将这些向量拼接起来作为该节点在当前聚集层最终的邻居聚集向量, 如式(6)所示, n_{heads} 表示头部的数量, \cup 是拼接符号。在实际任务中, 聚集层头部的数量以及每个头部使用的单元数都作为超参数经过试验和经验确定。

$$\mathbf{h}_{N(v)}^k = \bigcup_{x=1}^{n_{\text{heads}}} \left(\sum_{u \in N(v)} \sigma(\alpha_{vu}^k \mathbf{W}_x^k \mathbf{h}_u^{k-1}) \right) \quad (6)$$

拼接式(6)得到邻居聚集向量 $\mathbf{h}_{N(v)}^k$ 和上一层的聚集向量 \mathbf{h}_v^{k-1} , 得到

节点 v 当前层的聚集向量 \mathbf{h}_v^k , 如式(7)所示。

$$\mathbf{h}_v^k = \mathbf{h}_v^{k-1} \parallel \mathbf{h}_{N(v)}^k \quad (7)$$

最后, 将最后一个聚集层聚集出的节点 v 聚集向量记作 $\mathbf{h}'_v, \mathbf{h}'_v \in \mathbb{R}^{F \times 1}$, 将所有节点聚集向量组成的聚集属性矩阵记作 $\mathbf{H}', \mathbf{H}' \in \mathbb{R}^{F' \times n}$ 。

$$\mathbf{H}' = \{\mathbf{h}'_1, \mathbf{h}'_2, \dots, \mathbf{h}'_n\} \quad (8)$$

除了上述基于多头部的注意力聚集方式外, 还有一些其他优秀的聚集方式或聚集模型可供选择, 如 Geniepath^[17], GraphSAGE^[18]。在实际应用中, 可以通过多次尝试来选择适合实际任务的聚集方式。

3.2 表示层

表示层的功能是使用节点的聚集属性矩阵 \mathbf{H}' 得到边的特征矩阵 $\mathbf{H}_e = \{\mathbf{h}_{e1}, \mathbf{h}_{e2}, \dots, \mathbf{h}_{em}\}$, $\mathbf{H}_e \in \mathbb{R}^{F' \times m}$, m 是属性图边的个数, \mathbf{h}_{ei} 是第 i 条边特征向量, $\mathbf{h}_{ei} \in \mathbb{R}^{F' \times 1}$ 。假设第 i 条边 ei 连接的两个节点为 u, v , 则:

$$\mathbf{h}_{ei} = \mathbf{h}'_u + \mathbf{h}'_v \quad (9)$$

$$\mathbf{h}_{ei} = \{\mathbf{h}'_1, \dots, \mathbf{h}'_v, \dots, \mathbf{h}'_u, \dots, \mathbf{h}'_n\} \cdot [0, \dots, 1, \dots, 1, \dots, 0]^T \quad (10)$$

其中 \cdot 表示矩阵乘法, $\{\mathbf{h}'_1, \dots, \mathbf{h}'_v, \dots, \mathbf{h}'_u, \dots, \mathbf{h}'_n\}$ 即节点属性矩阵 \mathbf{H}' 。

引入表示向量 $\mathbf{c}_i = [0, \dots, 1, \dots, 1, \dots, 0]^T$, $\mathbf{c}_i \in \mathbb{R}^{n \times 1}$, n 是节点数量, \mathbf{c}_i 有两个位置的值为 1, 分别对应了边 ei 连接的两个节点 u 和 v 。则式(10)可改写为:

$$\mathbf{h}_{ei} = \mathbf{H}' \mathbf{c}_i \quad (11)$$

令表示矩阵 $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_m\}$, 则边的特征矩阵 \mathbf{H}_e 计算方式如下:

$$\mathbf{H}_e = \mathbf{H}' \mathbf{C} \quad (12)$$

3.3 密集层

密集层的功能是实现从边的特征矩阵 \mathbf{H}_e 到边的二分类别的映射。密集层通常由 1~3 层的全连接神经网络组成, 记最后一层输出为 $\mathbf{h}_{\text{out}} = \{\mathbf{h}_{\text{out}1}, \mathbf{h}_{\text{out}2}\}$, 边的分类结果为 $\mathbf{y} = \{y_1, y_2\}$, $y_1 + y_2 = 1$ 。

$$\mathbf{y} = \text{softmax}(\mathbf{h}_{\text{out}}) \quad (13)$$

$$y_1 = \frac{\exp(h_{\text{out}1})}{\exp(h_{\text{out}1}) + \exp(h_{\text{out}2})} \quad (14)$$

$$y_2 = \frac{\exp(h_{\text{out}2})}{\exp(h_{\text{out}1}) + \exp(h_{\text{out}2})} \quad (15)$$

最后使用分类结果 \mathbf{y} 和分类标签 $\bar{\mathbf{y}}$ 的交叉熵定义损失函数 $L(\mathbf{y}, \bar{\mathbf{y}})$:

$$L(\mathbf{y}, \bar{\mathbf{y}}) = -\frac{1}{N} \sum_{i=0}^N \left[\bar{y}_1^{(i)} \cdot \log(y_1^{(i)}) + \bar{y}_2^{(i)} \cdot \log(y_2^{(i)}) \right] \quad (16)$$

其中 $\bar{\mathbf{y}}$ 是 one-hot 编码的边类别标签向量, N 表示样本总数。

4 基于图神经网络的特征痕迹生成

在上一节中, 零件属性图被分解为孤立特征子图和相交特征子图, 为了进一步识别其中的相交特征, 使用痕迹特征子图作为特征的痕迹。

本节的主要任务就是通过图分解将相交特征子图分解为一系列痕迹特征子图, 每个痕迹特征子图对应了唯一的加工特征, 如图 6 所示。图 6(a) 是上一节得到的相交特征子图, 包含了两个加工特征, 分别使用橙、蓝两种颜色节点表示, 其中混合颜色节点表示特征相交造成的公共节点, 相交特征子图内部的边被分为“1”和“0”两类, 其中“1”表示需要打断的边, “0”表示需要保留的边。图 6(b) 是打断类型为“1”的边之后生成的两个痕迹特征子图, 可以看出虽然蓝色加工特征在打断过程中失去了公共节点, 但仍然保留了完整加工特征存在的部分几何信息。

痕迹特征子图的生成过程等价于对相交特征子图中的边进行打断/保留的二分类, 本节同样基于图神经网络算法构建了边二分类模型 (GNN_{hint})。模型训练所需标签构造方式为: 不同孤立特征子图间的边设置为“1”表示打断, 相同孤立特征子图内部的边设置为“0”表示保留, 对于共有节点采取随机划分的策略, 其边的打断和保留取决于该节点划

分给哪个加工特征子图。

GNN_{hint} 的输入为定义相交特征子图所需的邻接子矩阵 A_{sub} 和属性子矩阵 H_{sub} , $A_{sub} \in \mathbb{R}^{n' \times n'}$, $H_{sub} \in \mathbb{R}^{F \times n'}$, 其中 n' 表示相交特征子图所含节点的数量。 GNN_{hint} 模型的结构与上一节点分离加工特征的图神经网络边二分模型 GNN_{split} 一致, 不同之处在于模型架构的超参数不同, 如聚集层的层数等。最后根据 GNN_{hint} 输出的边二分结果, 利用连通子图算法得到一系列痕迹特征子图。

5 基于神经网络的特征痕迹补全

本节根据痕迹特征子图留下的几何线索对痕迹进行补全以得到完整的加工特征子图。显然痕迹特征子图中缺失的节点一定属于其所在相交特征子图, 因此其补全流程为: 遍历相交特征子图中的节点, 并根据几何线索判断每个节点是否属于当前痕迹特征子图, 如果属于则将该节点加入到该痕迹特征子图, 如图 7 所示。

上述痕迹补全的流程可以看成对相交特征子图中所有的节点进行分类, 类别“1”表示节点属于该痕迹, 类别“0”表示节点不属于该痕迹。本文基于神经网络算法建立了节点二分类模型(NN_{fix}), 该模型根据痕迹提供的几何线索判断相交特征子图中每个节点与该线索的关系。

NN_{fix} 模型是由多个全连接层组成神经网络。首先将痕迹特征子图反映的几何线索编码为痕迹向量 h_{clue} , 编码方式为对痕迹特征子图中所有节点的属性向量求平均。接着对于相交特征子图中的每个节点 v , 将其属性向量 h_v 与痕迹向量 h_{clue} 拼接得到 NN 模型的输入向量 h_{in} , 如式 (17) 所示。

$$h_{in} = h_v \parallel h_{clue} \quad (17)$$

其中, $h_v \in \mathbb{R}^{F \times 1}$, $h_{clue} \in \mathbb{R}^{F \times 1}$, $h_{in} \in \mathbb{R}^{2F \times 1}$, F 是属性向量的维度。

经过多个全连接层对 h_{in} 进行多次非线性变换, 并记最后一层全连接

的输出为 h_{out} , 那么类似式 (13~16), 通过 softmax 函数可以计算出节点分类结果, 并构造出类似的交叉熵损失函数。最后根据节点分类结果即可补全痕迹特征子图, 得到完整的加工特征子图。

试验验证

由于不同类型的飞机结构件的工艺数据存在较大差异, 因此为了满足深度学习训练集和测试集数据独立同分布假设, 需要为不同类型的飞机结构件使用各自的工艺数据分别训练数据驱动模型, 从而保证数据的同源性。本文使用框类零件验证了所提方法识别加工特征的能力。

本文使用航空制造企业 8 个框类复杂飞机结构件的历史工艺数据训练了本文数据驱动模型, 并根据实际零件特点设计了 1 个测试零件

用于展示本文方法识别出的加工特征, 这 9 个零件一共包含了 2113 个加工特征, 其中孤立特征 938 个, 相交特征 1175 个, 详细信息如表 2 所示, 表中相交特征数是指包含在相交特征中的加工特征总数。由于零件模型信息以及加工特征信息保存在 CATIA 相应文件中, 本节首先从 CATIA 文件中提取并预处理加工特征数据, 然后划分数据集, 接着介绍 GNN_{split} 、 GNN_{hint} 、 NN_{fix} 3 个模型的训练, 最后给出了试验结果。

1 数据准备

(1) 数据提取。基于 Microsoft VS C++ 以及 CATIA 二次开发组件 CAA 开发了一套数据提取工具, 用于自动从 CATIA 历史文件中提取零件模型的属性图和加工特征。

(2) 预处理。根据加工特征自动识别任务和复杂飞机结构件的特

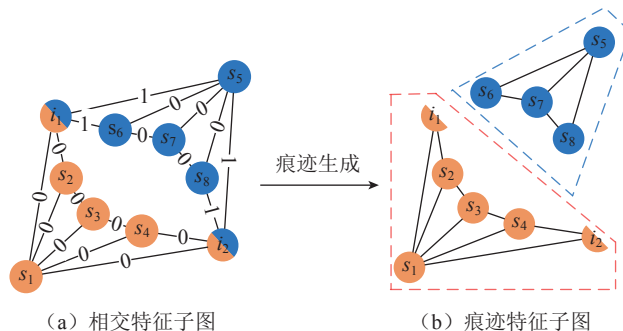


图 6 痕迹特征子图生成过程

Fig.6 Generation of hint feature subgraphs

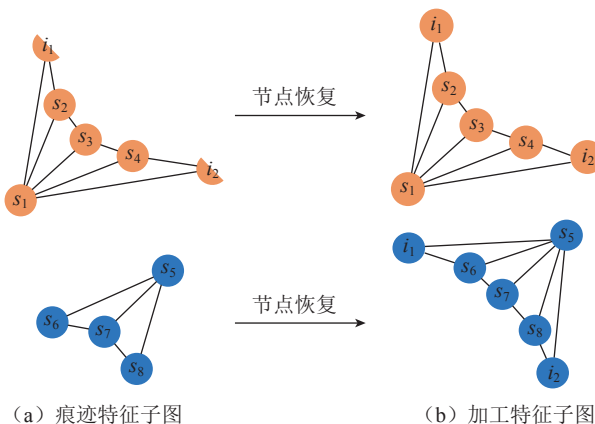


图 7 痕迹特征子图补全

Fig.7 Completion of hint feature subgraph

点,最终选取了9个属性共18个维度,如表3所示。在提取完零件的属性矩阵之后,需要对各个属性做一些处理,通常为了减小异常样本数据的影响,提高模型收敛速度,会对数值较大、分布较广的属性作归一化处理。本文使用了线性归一化方法,让属性的值放缩到0到1之间。以对属性 x 进行归一化为例。

$$x_{\text{norm}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \quad (18)$$

式中, x_{min} 和 x_{max} 分别对应属性 x 在样本中取到的最小和最大值, x_{norm} 是属性归一化后的结果。本文一共对序号为4~9的属性作了归一化处理。

(3) 数据集划分。以1~7号零件为训练集,8号零件为验证集,9号零件为测试集,则3个模型使用的样本数量如表4所示。

2 数据驱动模型训练

GNN_{split} 、 GNN_{hint} 、 NN_{fix} 3个模型均基于Tensorflow^[19]深度学习框架构建,使用训练集数据分别训练了这3个深度学习模型,均使用adam优化器来优化模型可学习的参数矩阵,3个模型结构方面的超参数设置如表5所示。

3 试验结果

图8是 GNN_{split} 、 GNN_{hint} 和 NN_{fix} 3个数据驱动模型在训练集和验证集上的损失函数收敛曲线,分别训练了3200轮、2000轮和3200轮。表6是3个模型的预测精度,其中测试集上的精度分别达到了97.4%、98.2%和96.5%。需要注意的是,在测试阶段 GNN_{hint} 的输入是 GNN_{split} 模型预测得到的相交特征子图分离结果,而 NN_{fix} 的输入是 GNN_{split} 模型预测得到的痕迹特征子图的结果,由于3个模型存在一定的依赖关系,造成测试集的精度相比训练集有所下降。

根据数据驱动模型预测结果,从9号测试零件属性图中分解出所有

表2 历史工艺数据的详细信息
Table 2 Details of the historical data

零件号	面的数量	边的数量	孤立特征数	相交特征数
1	860	2501	79	102
2	1282	3721	120	164
3	1335	3895	120	170
4	685	1995	64	91
5	1553	4548	152	190
6	1236	3601	135	134
7	1242	3631	117	156
8	386	1128	45	38
9	1161	3284	106	130

表3 试验选取的属性选择
Table 3 Selected attributes for test

序号	属性	数据类型	维度
1	面的法向量	浮点型	3
2	面类型	整型	3
3	面是否有环	整型	1
4	面积周长比	浮点型	1
5	距离底面的最大/最小高度	浮点型	2
6	节点的度(边的数量)	整型	1
7	面与面凹/凸边的数量	整型	2
8	面与面相切/不相切边的数量	整型	2
9	面中直线/圆弧/自由曲线的数量	整型	3

表4 数据驱动模型样本数量
Table 4 Size of sample sets of data-driven models

数据驱动模型	训练集	验证集	测试集
GNN_{split}	23892	1128	3284
GNN_{hint}	14114	527	1951
NN_{fix}	7680	630	1954

表5 模型超参数设置
Table 5 Hyper-parameters of models

数据驱动模型	超参数名	超参数设置
GNN_{split}	注意力层数量	3
	注意力层每层所含单元数量	12, 12, 12
	注意力层每层所含注意头部的数量	10, 10, 10
	全连接层每层神经元数量	255, 50
GNN_{hint}	注意力层数量	4
	注意力层每层所含单元数量	12, 12, 12, 12
	注意力层每层所含注意头部的数量	10, 10, 10, 10
	全连接层每层神经元数量	200, 180
NN_{fix}	全连接层每层神经元数量	255, 70, 70, 180

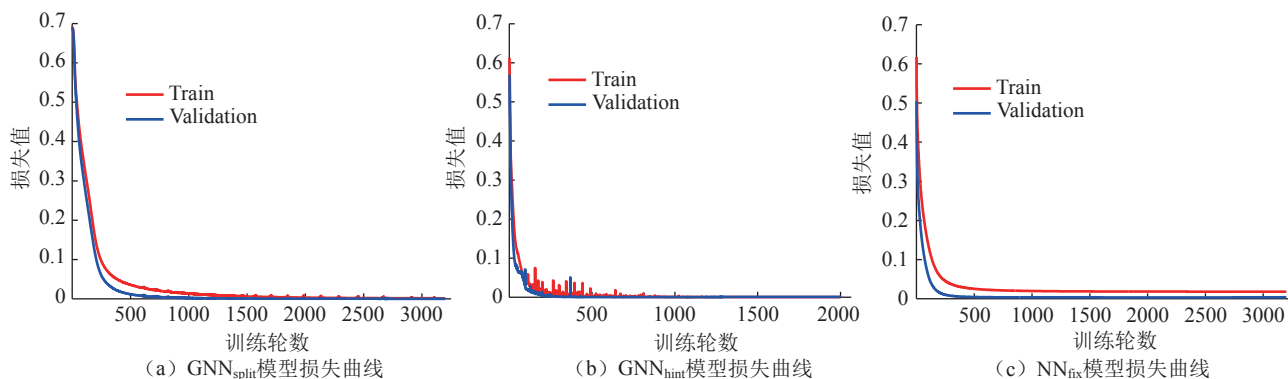


图 8 数据驱动模型损失函数收敛曲线

Fig.8 Convergence of loss function on the proposed data-driven models

表 6 数据驱动模型的精度

Table 6 Accuracy of the proposed data-driven models

模型	训练集精度	验证集精度	测试集精度
GNN _{split}	0.989	0.991	0.974
GNN _{hint}	0.985	0.987	0.982
NN _{fix}	0.974	0.980	0.965

的加工特征子图,鉴于篇幅,选择了其中一个槽腔的识别结果作为显示,如图 9 所示,其中“s”开头节点表示侧面,“b”开头节点表示底面,“h”开头节点表示孔的侧面,“r”开头节点表示筋顶面。

最后通过识别准确率衡量了本文所提算法的识别精度,包含孤立特征识别率(acc_1)、相交特征识别率(acc_2)和总的加工特征识别率(acc_3),分别为 98.11%、94.62% 和 96.18%,如表 7 所示。

$$acc_1 = \frac{correct_{isolate}}{all_{isolate}} \quad (19)$$

$$acc_2 = \frac{correct_{interact}}{all_{intersec}} \quad (20)$$

$$acc_3 = \frac{correct_{isolate} + correct_{interact}}{all_{isolate} + all_{interact}} \quad (21)$$

式中, $correct_{isolate}$ 表示孤立特征识别正确的数量, $all_{isolate}$ 表示总的孤立特征数量, $correct_{interact}$ 表示相交特征识别正确的数量, $all_{interact}$ 表示总的相交特征包含的孤立特征总量。

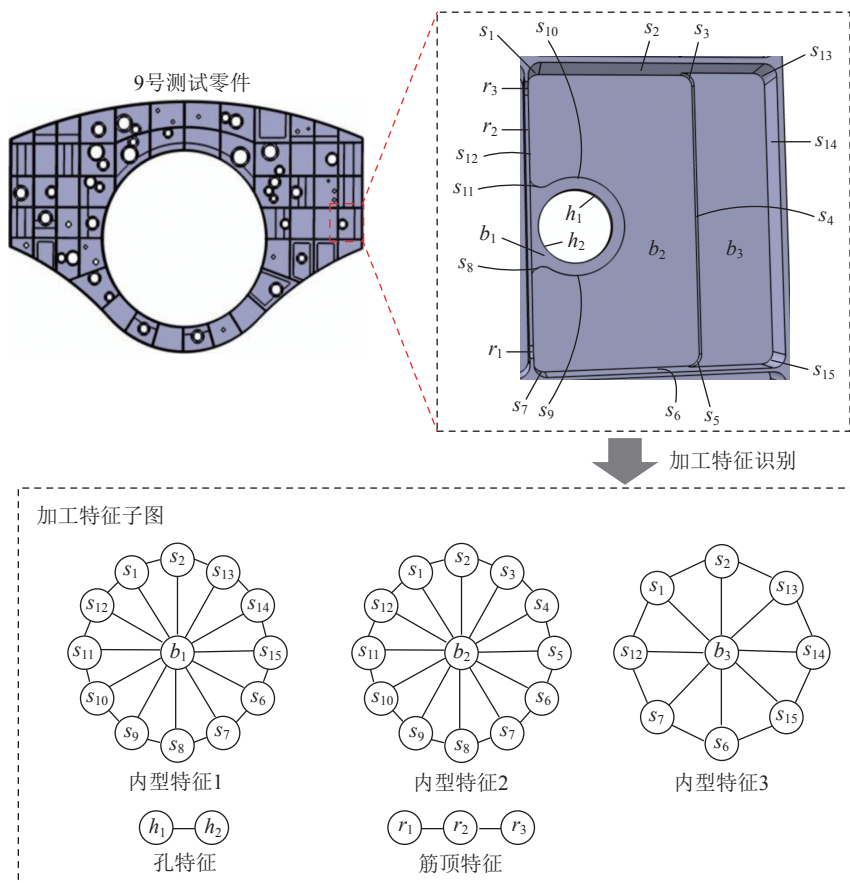


图 9 测试零件的部分特征识别结果

Fig.9 Some feature recognition results of test part

表 7 测试零件的加工特征识别率
Table 7 Correct feature recognition rates of the test part

孤立特征识别率	相交特征识别率	总的加工特征识别率
98.11%	94.62%	96.18%

结论

本文提出了一种数据驱动的飞机结构件加工特征识别方法,从历史工艺数据中学习特征识别的知识。本文的主要贡献如下:

(1) 提出了数据驱动的加工特征识别研究思路,消除对预先定义特征相交模式的依赖,从历史数据中自适应学习特征识别规则。

(2) 提出了基于图神经网络的飞机结构件加工特征识别方法,首先通过特征分离算法得到孤立特征和相交特征,再通过痕迹生成和补全算法进一步将相交特征分解为加工特征的集合,实现相交特征识别。

通过选用典型飞机结构件进行特征识别测试,测试结果表明所提方法具有较高加工特征识别精度,能够识别飞机结构件复杂相交特征,验证了方法的有效性。目前该方法无法保证识别结果完全正确,为了保证工艺编程的可靠性仍需要少量的人工校验。未来将考虑更多的实际零件及历史工艺数据,以此确定最佳的图神经网络结构,并推进所提方法在航空制造企业进行实际应用。

参考文献

[1] ZHU J, KATO M, TANAKA T, et al. Graph based automatic process planning system for multi-tasking machine[J]. Journal of Advanced Mechanical Design Systems and Manufacturing, 2015, 9(3): JAMDSM0034.

[2] VERMA A K, RAJOTIA S. A hint-based machining feature recognition system for 2.5D parts[J]. International Journal of Production Research, 2008, 46(6): 1515-1537.

[3] JOSHI S, CHANG T C. Graph-based heuristics for recognition of machined features from a 3D solid model[J]. Computer-Aided Design, 1988, 20(2): 58-66.

[4] MAREFAT M, KASHYAP R L. Geometric reasoning for recognition of three-dimensional object features[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990, 12(10):949-965.

[5] TRIKA S N, KASHYAP R L. Geometric reasoning for extraction of manufacturing features in iso-oriented polyhedrons[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1994, 16(11): 1087-1100.

[6] JI Q, MAREFAT M M. Bayesian approach for extracting and identifying features[J]. Computer-Aided Design, 1995, 27(6): 435-454.

[7] JI Q, MAREFAT M M. A Dempster-Shafer approach for recognizing machine features from CAD models[J]. Pattern recognition, 2003, 36(6): 1355-1368.

[8] FOUGÈRES A J, OSTROSI E. Intelligent agents for feature modelling in computer aided design[J]. Journal of Computational Design and Engineering, 2018, 5(1): 19-40.

[9] VERMA A K, RAJOTIA S. A review of machining feature recognition methodologies[J]. International Journal of Computer Integrated Manufacturing, 2010, 23(4): 353-368.

[10] LI W D, ONG S K, NEE A Y C. A hybrid method for recognizing interacting machining features[J]. International Journal of Production Research, 2003, 41(9): 1887-1908.

[11] HAN J H, REQUICHA A A G. Integration of feature based design and feature recognition[J]. Computer-Aided Design, 1997, 29(5): 393-403.

[12] VANDENBRANDE J H, REQUICHA A A. Automatic recognition of machinable features in solid models[D]. Rochester: University of Rochester, 1991.

[13] GAO S, SHAN J J. Automatic recognition of interacting machining features based on minimal condition subgraph[J]. Computer-Aided Design, 1998, 30(9):727-739

[14] 程少杰,李迎光.基于痕迹法的槽腔特征识别方法[J].机械制造与自动化,2008(2):50-52.

CHENG Shaojie, LI Yingguang. A hint-based method of machining feature recognition for cavity[J]. Machine Building & Automation, 2008(2):50-52.

[15] WANG H, ZHANG J, ZHANG X, et al. An oriented feature extraction and recognition approach for concave-convex mixed interacting features in cast-then-machined parts[J]. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 2019, 233(4): 1269-1288.

[16] VELIČKOVIĆ P, CUCURULL G, CASANOVA A, et al. Graph attention networks [EB/OL]. (2018-02-04) [2020-05-02]. <https://arxiv.org/pdf/1710.10903.pdf>.

[17] LIU Z, CHEN C, LI L, et al. Geniepath: Graph neural networks with adaptive receptive paths[C]//Proceedings of the AAAI Conference on Artificial Intelligence. New York, 2019.

[18] HAMILTON W L, YING R, LESKOVEC J. Inductive representation learning on large graphs[EB/OL]. (2020-02-19) [2020-05-02]. <https://arxiv.org/pdf/2002.07962v1.pdf>.

[19] ABADI M, BARHAM P, CHEN J, et al. Tensorflow: A system for large-scale machine learning[EB/OL]. (2016-05-27) [2020-05-02]. <https://www.ixueshu.com/document/80ce6ed80cf3ac1a318947a18e7f9386.html>.

通讯作者: 李迎光,教授、博士生导师,研究方向为航空航天复杂结构件数字化制造与智能制造, E-mail: liyingguang@nuaa.edu.cn.

(下转第 57 页)

Lean Production Line Planning for Airbus A220 Aircraft Door

LIU Min, WANG Dawei, YANG Yanrui, YANG Wubing

(AVIC SAC Commercial Aircraft Company LTD., Shenyang 110000, China)

[ABSTRACT] In order to improve the assembly efficiency of Airbus A220 aircraft door production line and reduce the labor and man-hours, firstly, the assembly time data of door production line was measured by time measurement method based on lean thinking in this paper, and the number of tooling needed to improve the production capacity is determined; Secondly, the Spaghetti diagram was used to analyze the logistics trend of operators in the assembly process, and the possible waste phenomenon is analyzed. Based on the waste phenomenon and the characteristics of door assembly process flow, the door production line is arranged according to the principle of product clustering. At the same time, the layout form of the whole station serial and the local bottleneck station parallel are determined referring to the IE (industrial engineering) method, then the door lean production line is established. Finally, 3D rendering was conducted for the overall layout by 3ds Max software.

Keywords: A220 door ; Lean concepts ; Industrial engineering (IE); Production line; 3ds Max

(责编 一元)

(上接第 40 页)

A Data-Driven Method for Machining Feature Recognition for Aircraft Structural Parts

LU Kai¹, LI Yingguang¹, LIU Xu², DENG Tianchi¹

(1. College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China;

2. School of Mechanical and Power Engineering, Nanjing Tech University, Nanjing 210009, China)

[ABSTRACT] Machining feature recognition is an essential way to realize the integration of CAD/CAM, and is significant to improve the quality and efficiency of CNC machining of the aircraft structural parts, which contains plenty of complex interacting features with complex structure. Existing machining feature recognition methods require predefinition of feature interacting patterns before recognizing interacting features, which is difficult to meet the recognition requirements of complex interacting features of the aircraft structural parts. Therefore, a data-driven method for machining feature recognition for aircraft structural parts is proposed, which transforms the problem of machining feature recognition into a graph learning problem, and adaptively learns feature recognition rules from historical process data by constructing a graph neural network model. The proposed method breaks the limits of the pre-defined feature interacting patterns. With typical aircraft structural parts as test parts, the correct recognition rates of isolated features, interacting features and total machining features reach 98.11%, 94.62% and 96.18%, respectively, which verifies the effectiveness of the proposed method.

Keywords: Data-driven; Aircraft structural part; Machining feature recognition; Interacting feature; Graph neural network

(责编 一元)