

# 基于SolidWorks的自动焊缝标注的二次开发

张泽南, 仲梁维

(上海理工大学机械工程学院, 上海 200093)

✉1727358290@qq.com; zlvcd@126.com



**摘要:** 在进行大型焊接零部件设计时, 在现有的三维设计软件中设计人员通常采用多实体零件的设计方式进行设计, 但是在工程图中焊缝的标注往往是十分费时和费力的, 同时为了统计模型中的焊缝信息形成相应的焊缝清单, 目前任是人工操作, 大大增加了工作量和焊接件的设计周期。文章主要基于SolidWorks三维设计软件, 采用C#编程语言, 利用SolidWorks对外开放的API函数接口, 结合实际的焊缝识别准则, 实现了自动提取多实体三维设计模型中的焊缝信息, 采用SQLServer数据库对焊缝信息进行存储与读取, 并在工程图中进行自动焊缝标注, 同时导出对应的焊缝清单。大大减少了人工进行焊缝识别和标注的过程, 提高了焊接件的工程图的出图效率, 对于减少焊接件的设计周期有着现实的意义。

**关键词:** 焊缝识别; 自动标注; 多实体零件; 二次开发; SolidWorks

**中图分类号:** TP319 **文献标识码:** A

## The Second Development of Automatic Weld Labeling Based on SolidWorks

ZHANG Zenan, ZHONG Liangwei

(School of Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

✉1727358290@qq.com; zlvcd@126.com

**Abstract:** When designing large welded parts, the existing 3-d design software designers usually design the parts as separate entities, but the drawing of the weld marks tend to be very time-consuming and laborious, and at the same time in order to form the corresponding weld mark list, the task is now a manual operation, which greatly increased the workload and product design cycle. Based on SolidWorks 3-d design software, the paper uses C# programming language, SolidWorks open API interface, and the actual welding seam recognition criteria to achieve the automatic extraction of multi-entity weld information in three-dimension design model, SQLServer database is used to store and save the weld information, carries out automatic weld marking of engineering drawing, and export the corresponding weld listing at the same time. It greatly simplifies the process of manual identification and marking of welding seams, and improves the efficiency of engineering drawing of welding parts, with practical significance for reducing the design cycle of welding parts.

**Keywords:** weld identification; automatic marking; multiple solid parts; secondary development; SolidWorks

### 1 引言(Introduction)

焊接与连接是国家自然科学基金委员会机械工程学科的重要领域, 是零部件成形制造的重要工艺方法, 是实现大到巨型结构小到微纳器件轻量化和功能化的主要途径之一, 对实现装备制造绿色环保和节能减排具有重要意义<sup>[1]</sup>。焊接技术被誉为工业的裁缝, 在工业化发展的进程中, 焊接在船舶、汽车、航空航天、军事等领域发挥着不可替代的作用。21世

纪的制造业正在逐步向“数字化”“智能化”方向发展, 焊接也正顺应着这种趋势, 向自动化焊接的方向快步前进<sup>[2,3]</sup>。在SolidWorks三维设计软件中对焊接件进行设计, 工程图进行标注时, 需要手动标注和识别焊缝, 对于焊缝的信息需要人工进行测量和导出, 当焊接件的实体数量庞大时, 设计人员的工作量大幅度的增加, 且容易出错。SolidWorks是基于特征的三维参数化造型软件, 对于用户的自定义二次开发提

供了良好的支持,在SolidWorks中进行程序化建模时,需要针对SolidWorks进行二次开发,通过SolidWorks API接口,编制能够嵌入SOLIDWORKS的插件,通过调用插件进行程序化绘图<sup>[4]</sup>。运用二次开发技术,结合Visual Studio2012开发平台可实现特定功能的产品开发,可以拓展和弥补SolidWorks自身的不足。

## 2 开发原理(Development principle)

SolidWorks提供了完整的免费的二次开发工具API,凡是支持COM编程的开发工具,如Visual C++6.0(以下简称VC++)、Visual Basic、Delphi等均可直接调用这些API函数开发出用户自己的程序模块,提高设计的效率与正确率<sup>[5]</sup>。为了方便用户进行二次开发,SolidWorks提供了几百个API接口。这些API接口分为两种,即OLE Automations的IDispatch和作为Windows基础的COM(Component Object Model组件对象模型)<sup>[6]</sup>。对应的SolidWorksAPI对象模型如图1所示,它是一个自上而下的树型结构。

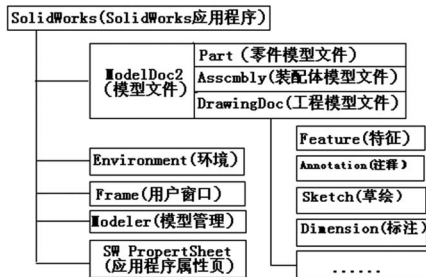


图1 SolidWorks对象模型

Fig.1 SolidWorks object model

SolidWorks的二次开发主要有两种:一种是基于自动化技术而生成的EXE可执行文件,另一种是基于COM开发SolidWorksAdd-in,生成一个DLL插件<sup>[7]</sup>。其中SolidWorks Add-in和SolidWorks程序运行于同一个进程空间,Add-in比主程序有更大的控制力,菜单、工具栏及属性控制页都可以插件形式在SolidWorks中实现。其中,第二种技术运用非常广泛,生成的DLL插件可以融合在SolidWorks设计环境中,增加工具栏的功能,更加便于设计者进行设计操作<sup>[8]</sup>。本文采用的开发方式为C#语言对应的SwAddin插件进行的二次开发,对应开发界面效果如图2所示。

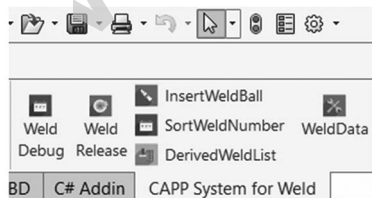


图2 插件界面

Fig.2 The plug-in interface

## 3 三维模型焊缝信息提取(3D model weld information extraction)

虽然本文最终的目的是在工程图中进行焊缝标注,但是在工程图中无法识别出三维的信息,所以首先要在三维模型空间

中提取焊缝的信息,具体实现流程的大体框架如图3所示。

其中关于接头形式的判断,交线的识别,在后面小节有详细的介绍。在SolidWorks三维设计软件中可以通过遍历特征树的技术访问到Cut list切割清单,并通过二次开发可以方便提取其中的实体,通过实体文件夹的创建3D box框的属性,可以在进入cut list后可通过其属性访问实体的厚度,长度,宽度等信息,保存记录作为焊接接头形式判断的依据之一。经过第一步提取可将三维模型中的所有实体对象存储到C#列表中,作为循环遍历的主体部分。

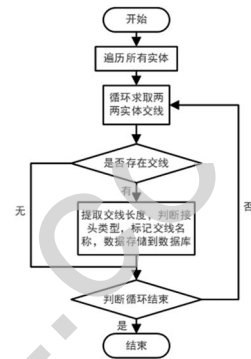


图3 焊缝信息提取流程图

Fig.3 Weld information extraction flow chart

### 3.1 焊接接头形式识别

焊接接头是指两个或两个以上零件要用焊接组合的接点。或指两个或两个以上零件用焊接方法连接的接头,包括焊缝、熔合区和热影响区。熔焊的焊接接头是由高温热源进行局部加热而形成。焊接接头由焊缝金属、熔合区、热影响区和母材金属所组成<sup>[9]</sup>。焊接接头形式可分为对接接头、T型接头、搭接接头、角接头,对应的不同形式的接头类型如图4所示。



图4 焊接接头形式

Fig.4 Welded joint form

本文中只针对对接接头和T型接头进行识别,这里一个十分重要的识别准则为:厚度对厚度的接头为对接接头,厚度对长度或者厚度对宽度的接头形式都为T型接头,对于角接头的处理原则为:如果角接的角度等于90°则作为T型接头处理,否则作为对接接头处理。对于搭接接头的形式出现较少,这里就直接忽略了。此外,本文只针对了两个实体间的焊缝进行了识别处理,可能会出现三个实体同时出现焊缝的情形,这里没有进行焊缝的合并,只是单独按照两两之间进行识别处理,存在一些不足。

### 3.2 实体交线信息提取方法

有了实体信息和接头的判断准则，下面是如何进行判断焊缝，在多实体零件中，焊缝只出现在两个实体的交线部分，所以要对实体之间求取交线。运用SolidWorks二次开发技术，对于实体是属于IBody2这个对象，在这个对象下可以通过GetFaces方法获取该体的所有表面，通过GetEdges方法可以获取该体上所有的边线，通过GetIntersectionEdges方法可以获取两个实体对象之间的交线。对于交线，可以求出交线的长度，作为焊缝长度的信息。

### 3.3 公共面提取法

如果两个实体的交线不止一条，则求取两个实体公共面，通过计算公共面，如果两个公共面都包含厚度尺寸，则认为是对接接头，否则为T型接头。如果两个实体的交线只有一条，这个时候就要求取该包含该交线的两个实体的面，计算两个实体之间包含交线的面之间的夹角，如果夹角为0度、90度、180度中的任何一个值，说明是T型接头，否则为对接接头。现在针对交线不止一条的情况来讨论公共面的寻找的方法：通过IEdge边线对象，可以利用GetTwoAdjacentFaces2方法获取包含该线的面，通过两条交线来获取包含交线的面的对象，这里可能的面的个数为2个、3个或者4个。对于两个面的情形说明两条交线都在面内，这两个面是同一个对象，任意取一个作为公共面，3个和4个的情形需要比较这几个面中哪两个是相同的面，可以利用IFace2对象下的IsSame方法比较两个面是否为同一个面。这样就可以找出公共面。对于只有一条交线的情况，只需要求取两个实体上包含交线的面中任意两个面之间的夹角。获取包含交线面的方法和多条交线的情况一样。

### 3.4 焊缝标记定位

提取到的每一条焊缝需要定位，方便在工程图中辨识每一条线是否为焊缝线的投影。这里可以利用实体命名的方式来进行焊缝对象的标记。可通过PartDoc对象模型下的SetEntityName方法为实体进行命名，其效果如图5所示。

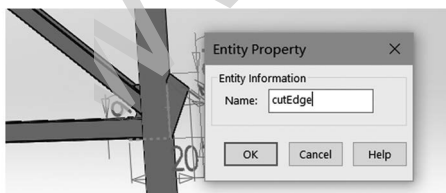


图5 实体命名

Fig.5 Entity named

## 4 工程图自动焊缝标注(Automatic weld marking of engineering drawing)

在三维模型中对焊缝信息进行了提取，对焊缝进行了标定。在工程图中需要进行标注，对于工程图，利用遍历工程视图的方式，通过IView对象下的GetPolylines7方法可以获得工程视图中所有的边线。但是如何将工程视图中的边线和三

维中的边线对应起来，需要进行映射关系转换，找到对应的三维中的线再查看该线是否被标记，二维中的线需要将IEdge对象转化为IEntity对象，通过ModelName属性可以获取对应的实体名称，如果是焊缝则被标记，如果不是则为空。

### 4.1 三维模型到工程图映射关系

在SolidWorks中三维图形变换都是通过变换矩阵来完成的，该变化矩阵如式(1)所示。

$$\begin{bmatrix} a & b & c & n \\ d & e & f & o \\ g & h & i & p \\ j & k & l & m \end{bmatrix} \quad (1)$$

该矩阵一共包含16个元素， $a-i$ 前9个元素构成的 $3 \times 3$ 子矩阵表示旋转矩阵，后面三个元素( $j, k, l$ )为平移矢量， $m$ 表示比例因数，( $n, o, p$ )在这里没有实际意义<sup>[10]</sup>。对于每一个视图IView对象，可以通过ModelToViewTransform获取三维模型到二维视图的转换矩阵，这里需要注意的是在二维视图中遍历到的线段对象是对应三维空间的线段，对应到二维视图中，只需要执行矩阵的乘积运算即可，假设三维空间的一个点 $A(X_a, Y_a, Z_a)$ ，对应到二维视图的坐标点为 $B(X_b, Y_b, Z_b)$ ，则有如式(2)的运算关系：

$$\begin{bmatrix} X_b \\ Y_b \\ Z_b \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c & n \\ d & e & f & o \\ g & h & i & p \\ j & k & l & m \end{bmatrix} \cdot \begin{bmatrix} X_a \\ Y_a \\ Z_a \\ 1 \end{bmatrix} \quad (2)$$

注意这里的坐标点转换为其次坐标的形式，这样可以实现三维到二维的转换，在二维视图对象中对线进行选择，从而进行标注的实现。

### 4.2 定位焊缝标注

对于焊缝标注需要四个步骤：首先在找到相应的焊缝边线对应的二维坐标点之后，可以通过模型对象ModelDoc下的扩展属性下的SelectByID2方法来选中对应的边线；其次就可以插入焊缝标注序号，插入之前需要先创建标注对象，这里同样是通过模型对象下的扩展方法CreateBalloonOptions来创建对象，并且设置相关的属性；然后用InsertBOMBalloon2方法插入创建的标注；最后在插入标注的序号之后再更改相应的属性，设置该标注依附的实体对象。

### 4.3 焊缝标注位置调整

焊缝序号标注完成之后是不整齐的，需要进行位置的调整，达到图纸的要求。这里序号调整的方法是利用视图的边框属性，将焊缝序号调整到边框外围固定的位置，并且按照一定的间隔进行排布。实现方式流程大致为：第一步：通过IView对象的GetOutline方法获取外边框。第二步：遍历获取该视图中焊缝序号的个数，同时获取焊缝标注附着点的坐标。第三步：根据边框尺寸和焊缝序号个数均匀分布焊缝标注。最终实现效果如图6所示。

