

doi: 10.3969/j.issn.1674-8425(z).2020.10.016

本文引用格式:肖雄,张晋西,张太绪,等.基于Solidworks与VB.NET工程图自动调整方法研究及应用[J].重庆理工大学学报(自然科学),2020,34(10):118-125.

Citation format: XIAO Xiong, ZHANG Jinxi, ZHANG Taixu, et al. Research and Application of Automatic Adjustment Method Based on Solidworks and VB.NET Engineering Drawings [J]. Journal of Chongqing University of Technology (Natural Science), 2020, 34 (10): 118 - 125.

# 基于Solidworks与VB.NET工程图 自动调整方法研究及应用

肖雄,张晋西,张太绪,罗双宝,徐磅逸

(重庆理工大学机械工程学院,重庆 400054)

**摘 要:**参数化模型驱动后,与其关联的工程图自动更新时将出现视图布局不合理、视图比例失调、尺寸标注混乱等问题。分析了目前视图比例及其位置调整方法的优缺点,提出了基于视图对角线之差最小化及视图中心偏移量的调整方法,优化了视图比例调整。提出了基于尺寸与视图相对位置及字符标记的调整方法,简化了尺寸调整过程。采用VB.NET及结合数据库技术,实现了基于模板参数化驱动后工程图视图位置、视图比例及各种尺寸标注位置的优化调整,智能化出图,提高了设计效率和质量。

**关键词:**工程图;视图布局;自动调整;VB.NET;数据库

中图分类号:TP391

文献标识码:A

文章编号:1674-8425(2020)10-0118-08

## Research and Application of Automatic Adjustment Method Based on Solidworks and VB.NET Engineering Drawings

XIAO Xiong, ZHANG Jinxi, ZHANG Taixu,

LUO Shuangbao, XU Pangyi

(College of Mechanical Engineering, Chongqing University of Technology,  
Chongqing 400054, China)

**Abstract:** After the parametric model is driven, with the associated drawings automatically updated, there will be problems with unreasonable view layout, misaligned view, and chaotic dimensions. The advantages and disadvantages of the current view scale and its position adjustment method are analyzed. The method of minimizing the difference between the diagonal of the view and the offset of the center of the view is proposed to optimize the view scale adjustment. An adjustment method based

收稿日期:2019-09-13

基金项目:重庆市研究生教育优质课程(91);重庆理工大学研究生创新项目资助(yxc20192025);重庆市基础与前沿研究项目(csts2016jcyjA0444)

作者简介:肖雄,男,硕士研究生,主要从事计算辅助设计研究,E-mail:1169475875@qq.com;通讯作者 张晋西,男,硕士,教授,主要从事计算机辅助设计、机械运动仿真研究,E-mail:zhangjinxi@cqut.edu.cn。

on the relative position of the size and the view, and the character mark is proposed, which simplifies the size adjustment process. Using VB. NET and combined with database technology, based on template parameterization, the optimization adjustment of the drawing view position, view scale and various dimension positions is realized, and the intelligent drawing is improved, which improves the design efficiency and quality.

**Key words:** drawing; view layout; auto-adjust; VB. NET; database

产品参数化三维建模能极大地提高设计和绘图效率,目前这类技术获得了较广泛的应用,但三维模型参数化驱动后,与其关联的工程图自动更新生成时将出现视图比例不合理、视图错位、尺寸标注混乱等问题,因此,需对参数化驱动后的工程图进行调整。目前,国内外学者在工程图调整技术方面进行了大量研究,且取得了一定的成果。王涛等<sup>[1]</sup>根据水平与竖直尺寸的特点,提出一种基于尺寸标注子集划分增量式尺寸自动布局算法,实现了水平与竖直尺寸自动布局。黄飞等<sup>[2]</sup>根据模板工程图中视图定位点坐标实现了视图位置调整,同时提出一种基于尺寸图层分布的尺寸调整方法。李志等<sup>[3]</sup>提出将参数化后视图包络框对角线长度与模板视图对角线长度对比分析的方法,实现了视图比例调整。张求星等<sup>[4]</sup>根据工程图模板中视图包络框中心点坐标实现了视图位置调整。贾鸿盛等<sup>[5~10]</sup>提出一种基于尺寸类别和结构的位置调整方法,实现了线性及角度尺寸的位置调整。但这些方法对视图比例调整时会出现误调等现象,部分尺寸无法调整及位置调整不合理等问题。

针对当前工程图自动调优技术存在的不足,在参考前人的基础之上,就上述问题,提出了完善的视图比例调整方法、基于视图包络框中心点偏移量的视图位置调整方法及基于尺寸与视图相对位置及字符标记的调整方法。这些方法优化了工程图的调整过程,实现了对三维模型参数化驱动后工程图的自动调整及批量出图。

## 1 视图比例调整

本文将依据视图包络框对角线长度实现视图比例调整。视图比例、视图位置及尺寸调整与视

图包络框之间有着直接或间接关系,视图尺寸与视图包络框之间无约束关系,若先尺寸调整,再视图比例调整,包络框的大小将发生变化,从而导致尺寸位置再次混乱。因此,在参数化模型驱动后,需以视图比例、视图位置、尺寸位置的先后顺序对工程图进行调整。

在 SolidWorks 工程图中,包络框紧密嵌套在视图周围且大小不可更改。包络线位置及其参数如图 1 所示,其中点 $(X_{\min}, Y_{\min})$ 及 $(X_{\max}, Y_{\max})$ 分别为包络框左下角及右上角坐标,坐标数据可通过 View 对象中 GetOutline 函数获得。视图对角线长度计算如式(1)所示。

$$L = \sqrt{(X_{\max} - X_{\min})^2 + (Y_{\max} - Y_{\min})^2} \quad (1)$$

运用 View 对象中 GetName2 函数可获得视图名称。通过 View 对象中 ScaleRatio 函数获得视图当前比例,以及通过 UseSheetScale 函数判断是否采用图纸比例。在工程图模板制作完成后,通过上述函数提取包络框对角点坐标、对角线长度、视图比例及名称等信息,同时写入 Excel 数据库。

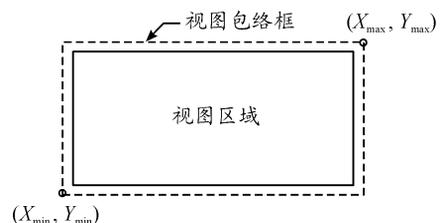


图 1 视图包络线示意图及其参数

参数化模型驱动后,通过 Sheet 对象中 GetViews 函数获得所有视图。根据视图数组得到包络框对角点坐标、对角线长度及视图比例等信息。为解决视图比例失调问题,在依据图纸比例规则下,提出更新后视图对角线长度与模板中该视图对角线长度之差达到最小的调整方法。因此,可

建立所有对角线长度之差和的函数,即:

$$F = \sum_{i=1}^k \left( \left| L'_i \frac{V_s}{V'_s} - L_i \right| \right) + \sum_{j=1}^m \left( \left| L'_j \frac{V_{s_j}}{V'_{s_j}} - L_j \right| \right) \quad (2)$$

式中: $L_i$ 、 $L'_i$ 为采用图纸比例的视图更新前后视图对角线长度, $V_s$ 为更新前图纸比例, $V'_s$ 为更新时图纸比例, $L_j$ 、 $L'_j$ 为采用自定义比例的视图更新前后的视图对角线长度, $V_{s_j}$ 为更新前自定义比例, $V'_{s_j}$ 为更新时的自定义比例, $k$ 为采用图纸比例的视图个数, $m$ 为采用自定义比例的视图个数。从而将视图比例的调整转化为求图纸比例及非图纸比例来使函数  $F$  最小的优化问题。这样能避免对采用图纸比例的视图依次调整时造成已调整视图比例的再次失调。

遗传算法(genetic algorithm)是模拟生物在自然环境中的遗传和进化的过程而形成的自适应全局优化搜索算法。因此,本文采用遗传算法求解视图比例,求解流程如图2所示。

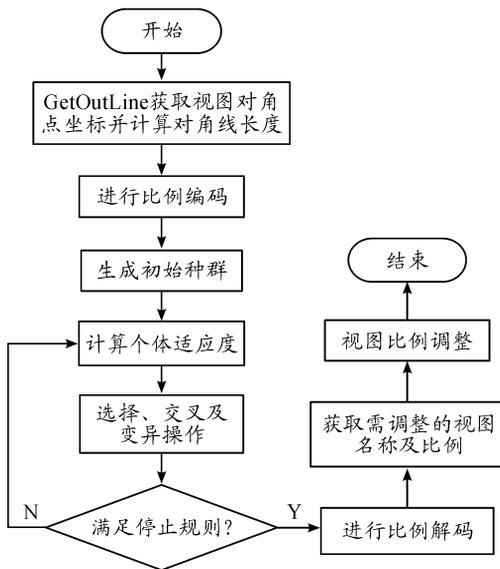


图2 视图比例求解流程框图

在参数化模型驱动后,计算各视图对角线长度,根据制图比例标准及当前图纸比例产生一组图纸比例,根据各自定义视图当前比例产生相应自定义比例数组,对图纸比例及自定义比例进行相应编号,再进行对应编码产生初始种群。

通过式(2)计算适应度,采用“轮盘赌”方式选择优良个体。该方法是一种基于比例的选择,利用各个体适应度所占比例决定其子孙保留的可能性。对选择的个体进行交叉及变异,若满足迭代次数或最优个体适应度值的变化达到给定阈值就停止运算。

根据求解结果得到需调整的视图名称及其比例,通过 ActivateView 及 ScaleRatio 函数调整需要修改比例的视图。

## 2 视图位置调整

在 SolidWorks 工程图中,每个视图都有一个定位点及一个包络框中心点。如果在视图中添加草图,包络框将会自动调整大小以包括这些草图。如果定位点  $P$  及中心点  $O$  在视图中处于如图3所示位置,则会出现定位点与包络框中心点不重合的现象,若将更新后的视图定位到模板视图定位点,视图四周调整的增量将会不均匀。因此,采用定位点增减包络框中心点偏移量的方法实现视图位置的调整。

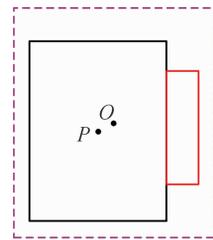


图3 定位点及中心点分布示意图

运用 View 对象中 Position 函数获得视图定位点。通过 GetOutline 函数获得包络框对角点坐标,并通过计算得到包络框中心点坐标  $(X_c, Y_c)$ ,即:

$$\begin{cases} X_c = (X_s - X_2)/2 \\ Y_c = (Y_s - Y_2)/2 \end{cases} \quad (3)$$

式中: $X_2$ 、 $Y_2$ 分别为视图包络框左下角顶点的  $X$  及  $Y$  坐标; $X_s$ 、 $Y_s$ 分别为视图包络框右上角顶点的  $X$  及  $Y$  坐标。

在 SolidWorks 工程图模板制作完成后,运用 View 对象中相应函数分别获取视图定位点、中心

点及视图名称等信息,并储存到 Excel 数据库中。视图比例调整后,通过遍历方式获取视图名称及包络框中心点坐标  $O'$ ,然后将该视图名称与 Excel 表中名称进行匹配,再将  $O'$  与所匹配的视图名称下包络框中心点坐标  $O$  进行比较。视图位置调整示意图如图 4 所示,其中视图包络框中心点坐标的偏移量  $x_{len}$ 、 $y_{len}$  的计算如下所示:

$$\begin{cases} x_{len} = (X_s' - X_z')/2 - X_c \\ y_{len} = (Y_s' - Y_z')/2 - Y_c \end{cases} \quad (4)$$

式中: $X_z'$ 、 $Y_z'$  为参数化驱动后视图包络框左下角顶点的  $X$  及  $Y$  坐标, $X_s'$ 、 $Y_s'$  为视图包络框右上角顶点的  $X$  及  $Y$  坐标。

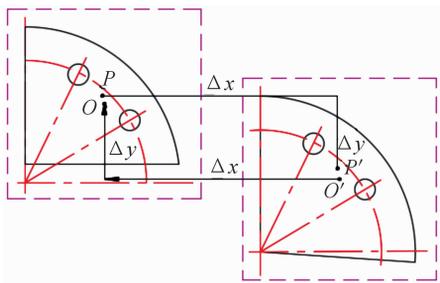


图 4 视图位置调整示意图

通过计算可得视图新的定位点  $P(X_p, Y_p)$ 。 $P$  点坐标的计算为:

$$\begin{cases} X_p = X_p' - x_{len} \\ Y_p = Y_p' - y_{len} \end{cases} \quad (5)$$

式中: $X_p'$ 、 $Y_p'$  分别为参数化模型驱动后视图定位点的  $X$  及  $Y$  坐标。

然后通过 View 对象中 Position 函数将当前视图定位到新定位点  $P$ ,从而实现对工程图中所有视图位置的重新定位。

### 3 尺寸调整

参数化模型驱动后,工程图中的视图大小及位置将发生变化,由于尺寸标注与视图之间不存在几何约束关系,进而导致其与视图变化不同步,将会出现如图 5 所示尺寸布局不合理等问题。为解决上述问题,在基于参数化模板工程图的前提下,提出新的调整思路:根据尺寸类型及布局特

点,记录下模板工程图布局合理时尺寸与包络框或基准点的相对位置,将数据记录于 Excel 中,在参数化更新后,利用这些数据与包络框对角点或基准点坐标计算出尺寸标注的坐标,最后实现尺寸标注自动调整。

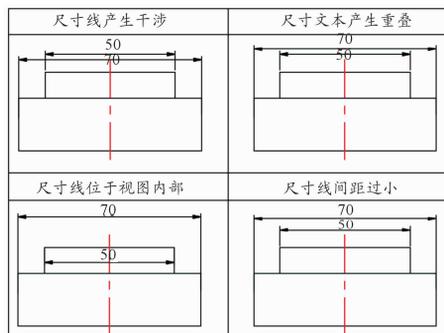


图 5 视图布局问题示意图

#### 3.1 尺寸类型、结构及布局

在工程图中,尺寸标注以水平尺寸与竖直尺寸所占比例最大,此外还有角度及倾斜尺寸标注等。尺寸标注是由延伸线、尺寸文本线、尺寸文本及箭头等元素所组成,尺寸结构及其组成如图 6 所示。

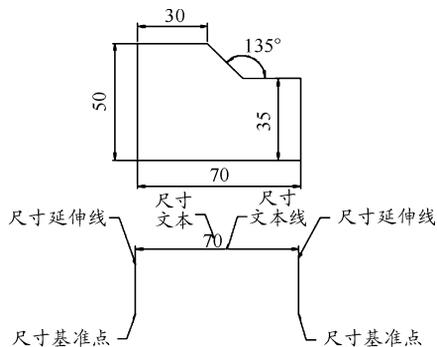


图 6 尺寸结构及其组成示意图

在工程图中,尺寸标注位置将会因视图具体形状的不同而不同。通过对整个视图尺寸布局特征的研究,尺寸布局可分为如图 7 所示两种情况。虚线为视图包络框,一种情况是尺寸布置在包络框外部,另一种情况则是尺寸布置在包络框内部。而对于角度及倾斜等类型的尺寸也可归为这两种情况。

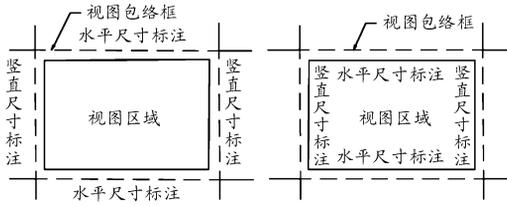


图7 尺寸布局分布示意图

### 3.2 尺寸调整原理及参数计算

尺寸调整是利用工程图更新前尺寸的位置坐标来对更新后的尺寸位置进行调整的过程。尺寸标注之间的重叠及干涉等布局问题可简化为如图8所示的问题模型。为使 $D_1$ 、 $D_2$ 之间不产生重叠及干涉,根据尺寸标注要求,间距 $\Delta_1$ 保持在5~10 mm,且 $D_1$ 与 $D_2$ 之间的相对位置不变。在制作工程图模板时尺寸之间的间距就已相对固定,参数化更新后只需保证尺寸间距与更新前尺寸间距相等就可使尺寸间不出现重叠及干涉。

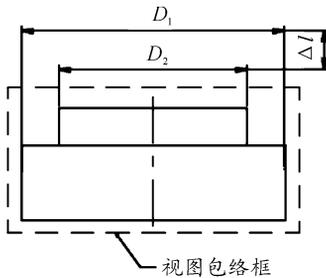


图8 尺寸布局模型示意图

尺寸标注之间的间距在SolidWorks中无法直接获取,需通过计算获得。包络框会随视图大小变化而自动调节且始终围绕在视图周围。因此,水平及竖直尺寸间距 $\Delta d$ 的计算将转化为如图9所示 $\Delta h$ 及 $\Delta v$ 计算。尺寸文本线与尺寸延伸线末端距离在参数化过程中始终保持不变,从而对于倾斜尺寸标注只需使更新前后尺寸延伸线长度不变就能实现合理布置。

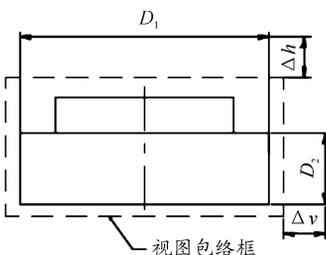


图9 尺寸布局参数示意图

在SolidWorks工程图中,对于不带延伸线的线性、角度、直径及半径标注,因尺寸文本线端点与尺寸基准点重合,使文本线能够随视图变化自动协调,从而不需调整。

在工程图模板制作完成之后,首先通过IDisplayDimension对象中的Type2函数对尺寸进行类别判断,若为线性尺寸,再通过GetArrowHeadAtIndex和GetLineAtIndex3函数分别获取尺寸箭头顶点坐标及尺寸标注中所有直线端点坐标,从而判断该线性标注有无延伸线,若有尺寸延伸线,再通过GetLineAtIndex3函数获取线性尺寸及角度尺寸的一条延伸线两端点坐标,起始坐标记为 $(X_q, Y_q)$ ,终点坐标记为 $(X_m, Y_m)$ ,尺寸文本线中心点坐标通过GetPosition函数获得,记为 $(X_c, Y_c)$ 。通过GetOutline函数可分别获取视图包络框左下角及右上角顶点坐标,记为 $(X_z, Y_z)$ 和 $(X_y, Y_y)$ 。

对于线性尺寸,若尺寸延伸线两端点坐标满足式(6),则为水平尺寸,若满足式(7),则为竖直尺寸,否则为倾斜尺寸。

$$X_z = X_y \quad (6)$$

$$Y_z = Y_y \quad (7)$$

对于竖直线性尺寸,若满足式(8),则该尺寸布置在包络框左侧,若满足式(9),则该尺寸布置在包络框右侧。若尺寸在包络框左侧, $\Delta v$ 的计算如式(10)所示,并用 $X_0$ 标记,若尺寸在包络框右侧, $\Delta v$ 的计算如式(11)所示,并用 $X_1$ 标记。

$$X_c \leq X_z \quad (8)$$

$$X_c \geq X_y \quad (9)$$

$$\Delta v = X_z - X_c \quad (10)$$

$$\Delta v = X_c - X_y \quad (11)$$

对于水平线性尺寸,若满足式(12),则该尺寸布置在包络框上侧,若满足式(13),则该尺寸布置在包络框下侧。若尺寸在包络框上侧, $\Delta h$ 的计算如式(15)所示,用 $Y_0$ 标记,若尺寸在包络框下侧, $\Delta h$ 的计算如式(14)所示,用 $Y_1$ 标记。

$$Y_c \leq Y_z \quad (12)$$

$$Y_c \geq Y_y \quad (13)$$

$$\Delta h = Y_z - Y_c \quad (14)$$

$$\Delta h = Y_c - Y_y \quad (15)$$

对于竖直尺寸,若满足式(16),则尺寸布置在视图内部,尺寸文本线中心点到基准点距离  $\Delta h$  的计算如式(18)所示,同时用 XY0 标记。对于水平尺寸,若满足式(17),则尺寸布置在视图内部,尺寸文本线中心点到基准点距离  $\Delta v$  的计算如式(19)所示,同时用 XY1 标记。

$$X_z < X_c < X_y \quad (16)$$

$$Y_z < Y_c < Y_y \quad (17)$$

$$\Delta h = X_q - X_c \quad (18)$$

$$\Delta v = Y_q - Y_c \quad (19)$$

对于倾斜的线性尺寸,用  $Q_x$  标记,若延伸线两端点坐标满足式(20),则用 Xu1 标记,若满足式(21),则用 Xu2 标记,若满足式(22),则用 Xb1 标记,若满足式(23),则用 Xb2 标记,延伸线长度  $l$  的计算如式(24)所示。

$$\begin{cases} Y_m > Y_q \\ X_m > X_q \end{cases} \quad (20)$$

$$\begin{cases} Y_m > Y_q \\ X_m < X_q \end{cases} \quad (21)$$

$$\begin{cases} Y_m < Y_q \\ X_m < X_q \end{cases} \quad (22)$$

$$\begin{cases} Y_m < Y_q \\ X_m > X_q \end{cases} \quad (23)$$

$$l = \sqrt{(X_m - X_q)^2 + (Y_m - Y_q)^2} \quad (24)$$

对于角度尺寸,用  $J_d$  进行标记,通过 IDisplayData 对象中 GetLineCount 函数判断角度尺寸是否有延伸线,若没有延伸线,则不需记录尺寸位置信息,若有延伸线,通过 GetLineAtIndex3 函数获取延伸线起始点坐标,记为  $(X_q, Y_q)$ ,计算出尺寸文本线中心点与起始点的相对距离,计算公式如下:

$$\begin{cases} \Delta x = X_q - X_c \\ \Delta y = Y_q - Y_c \end{cases} \quad (25)$$

在计算尺寸标注相对位置数据时,需通过 GetFullName 函数获取尺寸全名,以便尺寸调整时进行尺寸名称匹配。在尺寸标注相对位置数据计算完成后,需将其储存在表格中作为工程图更新后尺寸调整的数据依据。对于水平及竖直线性尺寸,表格第 1 行储存尺寸全名,第 2 行储存该尺寸

相对位置数据,第 3 行储存该尺寸标记字符。对于倾斜的线性尺寸,表格前 3 行储存的数据跟水平线性尺寸一样,另外在表格第 4 行需储存方位标记字符。对于角度尺寸标注,表格的第 1 行储存尺寸全名,第 2 行储存  $\Delta x$ ,第 3 行储存  $\Delta y$ ,第 4 行储存该尺寸的标记字符。

### 3.3 尺寸位置调整

本文将通过遍历及结合尺寸相对位置的方式实现尺寸调整。先通过 GetOutline 函数获取视图左下及右上对角点坐标,并将这两点坐标赋值给 Object 类型变量  $Z_x$  及  $Y_x$ ,然后通过 IDisplayDimension 对象中的 Type2 函数判断该尺寸类别,若该尺寸为线性或角度尺寸,再通过 GetFullName 函数获取尺寸全名,并将其赋值给 String 类型变量 name,然后定义循环,查找对应数据表中尺寸全名并与 name 变量匹配,若不相等,重复匹配操作,若相等,则提取该尺寸相对位置数据及尺寸标记字符,然后跳出循环。

文本线中心点新定位点  $P$  记为  $(X_n, Y_n)$ ,若该尺寸标记字符与 X0 匹配,再通过 GetArrowHeadAtIndex2 函数获取尺寸线上两端箭头顶点坐标,分别记为  $(J_{x0}, J_{y0})$  及  $(J_{x1}, J_{y1})$ ,则  $P$  点坐标计算公式为:

$$\begin{cases} X_n = Z_x(0) - \Delta v \\ Y_n = (J_{y0} + J_{y1})/2 \end{cases} \quad (26)$$

若该尺寸的尺寸标记字符与  $X_1$  匹配,则  $P$  点坐标计算公式为:

$$\begin{cases} X_n = Y_x(0) + \Delta v \\ Y_n = (J_{y0} + J_{y1})/2 \end{cases} \quad (27)$$

若该尺寸标记字符与  $Y_0$  匹配,再通过 GetArrowHeadAtIndex2 函数获取尺寸线上两端箭头顶点坐标,分别记为  $(J_{x0}, J_{y0})$  及  $(J_{x1}, J_{y1})$ ,则  $P$  点坐标计算公式为:

$$\begin{cases} X_n = (J_{x0} + J_{x1})/2 \\ Y_n = Z_x(1) - \Delta h \end{cases} \quad (28)$$

若该尺寸的尺寸标记字符与  $Y_1$  匹配,则  $P$  点坐标计算公式为:

$$\begin{cases} X_n = (J_{x0} + J_{x1})/2 \\ Y_n = Y_x(1) + \Delta h \end{cases} \quad (29)$$

若该尺寸标记字符与 XY0 匹配,再通过 GetArrowHeadAtIndex2 函数获取尺寸线上两端箭头顶点坐标,分别记为  $(J_{x0}, J_{y0})$  及  $(J_{x1}, J_{y1})$ ,并通过 GetLineAtIndex3 函数获取延伸线起始点坐标,记为  $(X_q, Y_q)$ ,则  $P$  点坐标的计算公式为:

$$\begin{cases} X_n = X_q - \Delta v \\ Y_n = (J_{y0} + J_{y1})/2 \end{cases} \quad (30)$$

若该尺寸的尺寸标记字符与 XY1 匹配,则  $P$  点坐标的计算公式为:

$$\begin{cases} X_n = (J_{x0} + J_{x1})/2 \\ Y_n = Y_q - \Delta h \end{cases} \quad (31)$$

若该尺寸标记字符与  $Q_x$  匹配,则表明该尺寸为倾斜线性尺寸标注,先通过 GetLineAtIndex3 函数获取一条延伸线起始点与终点坐标,记为  $(X_q, Y_q)$  及  $(X_m, Y_m)$ ,为保证延伸线长度不变,该延伸线新的终点坐标记为  $(X_{mn}, Y_{mn})$ ,其计算公式如式(32)所示,若尺寸方向标记字符为  $X_{u1}$ ,式中  $X_{mn}$  处取加,  $Y_{mn}$  处取加,若为  $X_{u2}$ ,式中  $X_{mn}$  处取减,  $Y_{mn}$  处取加,若为  $X_{b1}$ ,式中  $X_{mn}$  处取减,  $Y_{mn}$  处取减,若为  $X_{b2}$ ,式中  $X_{mn}$  处取加,  $Y_{mn}$  处取减。

$$\begin{cases} X_{mn} = X_q \pm l \sqrt{\frac{1}{1 + \left(\frac{Y_m - Y_q}{X_m - X_q}\right)^2}} \\ Y_{mn} = Y_q \pm \left| \frac{Y_m - Y_q}{X_m - X_q} \right| \sqrt{\frac{l^2}{1 + \left(\frac{Y_m - Y_q}{X_m - X_q}\right)^2}} \end{cases} \quad (32)$$

再通过 GetLineAtIndex3 函数获取该尺寸标注另一条延伸线起始点坐标,分别记为  $(X'_q, Y'_q)$ ,由于倾斜尺寸的 2 条延伸线平行且长度相等,因此该条延伸线起始点的偏移量也为  $\Delta x$  及  $\Delta y$ ,新的终点坐标记为  $(X'_{mn}, Y'_{mn})$ ,其计算方法同上。

然后该尺寸标注的尺寸文本中心点新定位点  $P$  的计算公式为:

$$\begin{cases} X_n = (X_{mn} + X'_{mn})/2 \\ Y_n = (Y_{mn} + Y'_{mn})/2 \end{cases} \quad (33)$$

若该尺寸的尺寸标记字符与  $J_d$  匹配,则通过 GetLineAtIndex3 函数获取延伸线起始点坐标,则  $P$

点坐标的计算公式为:

$$\begin{cases} X_n = X_q - \Delta x \\ Y_n = Y_q - \Delta y \end{cases} \quad (34)$$

通过 SetPosition 函数将尺寸文本线中心点定位到点  $P$ ,再通过 CenterText 函数对尺寸文本进行居中。

## 4 实例应用效果

本文以风机塔筒的平台底板为例,通过 SolidWorks 三维设计软件进行参数化设计,根据上述工程图调整方法,利用 VB.NET 平台及 SolidWorks API 接口编写程序对底板参数化驱动后的工程图进行自动调整。底板参数化设计后驱动工程图更新,如图 10 所示,视图比例失调,视图位置发生偏移,从而导致视图间或视图与图框产生干涉。线性及角度尺寸文本线出现重叠,部分尺寸文本线位于视图内部的错误情况,线性及角度尺寸延伸线过长且部分尺寸文本严重错位。

视图比例调整关键代码如下:

```
swdraw.ActiveView(Name) '激活指定视图
swview = swdraw.ActiveDrawingView'获取激活的视图
```

```
Dim scal As Object = swview.ScaleRatio'获取视图当前比例
```

```
scal = scal_g'比例赋值
```

```
swview.ScaleRatio = scal'重设视图比例
```

视图位置调整关键代码如下:

```
swview = swdraw.GetFirstView'获取第一个视图
```

```
Dim posi As Object = swview.Position'获取视图当前位置坐标
```

```
posi(0) = posi(0) - xlen'计算视图定位点 X 坐标
```

```
posi(1) = posi(1) - ylen'计算视图定位点 Y 坐标
```

```
swview.Position = posi'设置视图位置
```

```
swview = swview.GetNextView()'获取下一个视图
```

尺寸调整关键代码如下:

```
swview = swdraw.GetFirstView
```

```
swdispdim = swview.GetFirstDisplayDimension
```

( )'获取第一个尺寸

```
swann = swdispdim.GetAnnotation( )'获取尺寸注释
```

```
swann.SetPosition( CenterXY( 0 ), CenterXY( 1 ), 0 )'设置尺寸位置
```

```
swdispdim = swdispdim.GetNext5( )'获取下一个尺寸
```

```
swview = swview.GetNextView( )
```

编写的工程图调整程序能一键且批量处理参数化设计后的工程图,快速准确地调整工程图。运行工程图调整程序对底板工程图进行调整,调整结果如图 11 所示,视图比例、视图布置及尺寸位置出现的问题均已解决。

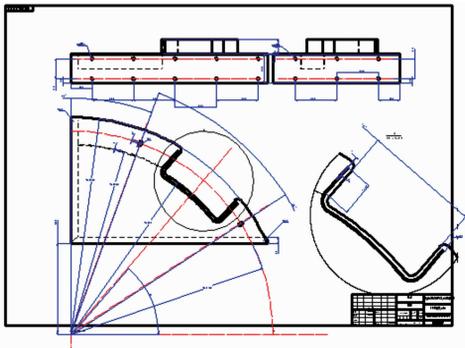


图 10 模型参数化驱动后的工程图结果示意图

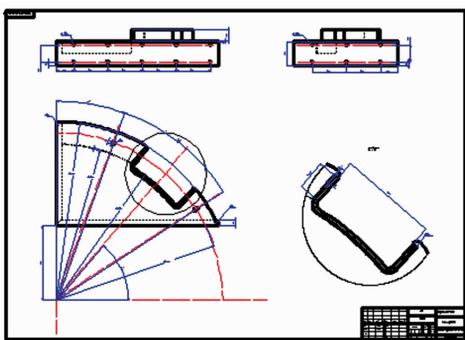


图 11 调整后的工程图结果示意图

## 5 结论

本文中研究了目前三维产品参数化驱动后工程图存在的问题及不足,提出了基于模板工程图的调优方法。利用 VB.NET 平台及 SolidWorks API 实现了参数化驱动 SolidWorks 模型后与其关联的工程图中视图布局及比例、尺寸位置等优化调整。采用本文的工程图调整方法可减少人工的调整过程,在满足工程图质量的同时,缩短了调整优化时间,极大地提高了对工程图批量化处理的速度。

## 参考文献:

- [1] 王涛,莫蓉,万能. 工程图尺寸标注自动布局算法及实现[J]. 航空计算技术,2010,40(2):73-76.
- [2] 黄飞,王宗彦,王乔,等. 基于参数化模板的工程图自动生成与优化调整技术[J]. 煤矿机械,2011,32(5):224-226.
- [3] 李志,安维胜. 基于 SolidWorks 的定制产品工程图自动调整技术[J]. 煤矿机械,2015,36(8):320-322.
- [4] 张求星,赵高晖. 基于 SolidWorks 二次开发中工程图的自动生成和优化[J]. 农业装备与车辆工程,2018,56(11):72-75.
- [5] 贾鸿盛,李盛. 基于类别和结构的工程图尺寸调整技术研究[J]. 现代制造工程,2016(1):73-77.
- [6] 李末. 基于 SolidWorks 的工程图智能生成研究与应用[D]. 大连:大连理工大学,2012.
- [7] 黄启良,王宗彦,吴淑芳,等. 参数化变型设计中工程图调整技术优化研究[J]. 工程图学学报,2011(1):168-173.
- [8] 韩朝,仲梁维. SolidWorks 三维数字化设计工程图技术研究[J]. 现代制造工程,2012(3):42-45.
- [9] 陈乐. 基于 SolidWorks 的工程图快速生成技术研究[D]. 银川:宁夏大学,2015.
- [10] 朱学敏,王宗彦,杨芬,等. 基于免疫遗传算法的工程图自动调优技术[J]. 图学学报,2013,34(2):83-88.

(责任编辑 林 芳)