文章编号:1671-4598(2021)07-0269-06 DOI:10.16526/j.cnki.11—4762/tp.2021.07.052 中图分类号:TP391.41 文献标识码:A

新一代载人航天器显示仪表的图形加速算法研究

张庆熙，夏加高，李文新
（兰州空间技术物理研究所，兰州 730000）

摘要：针对航天显示仪表对图形快速绘制的需求，研究现有2D图形绘制算法；提出改进的Bresenham直线绘制方法，利用线段中点进行加速绘制，一次循环可同时绘制两个点，比原算法节省了约30%的计算步骤；提出基于凸多边形的种子填充算法，利用凸多边形的某一单边确定填充种子，记为填充起点，以余下的边作为填充边界，记为填充终点，将所有起点终点进行对应划线完成填充，简化了计算步骤，算法复杂度减半；经过实验验证该算法在绘制复杂仪表图像时在执行时间方面的优越性，在理论和实验上都优于传统算法，能够较好地满足航天显示仪表对二维图形加速的需求，已成功应用到载人飞行器仪表中。

关键词：图形快速算法；显示仪表；图形绘制；图形填充；载人航天

Research on Graphics Acceleration Algorithms for Display Instruments of New Generation Manned Spacecraft

ZHANG Qingxi，XIA Jiagao，LI Wenxin
（Lanzhou Institute of Physics，Lanzhou 730000，China）

Abstract：In response to the demand for rapid graphics rendering of aerospace display instruments，the existing 2D graphics rendering algorithms are studied. An improved Bresenham line drawing method is proposed，which uses the midpoint of the line segment to speed up the drawing. Two points can be drawn at the same time in one cycle，which saves about 30% of the calculation steps than the original algorithm；a seed filling algorithm based on convex polygons is proposed，which uses convex polygons. A certain single side determines the filling seed，which is recorded as the starting point of filling，and the remaining edges are used as the filling boundary，which is recorded as the filling end point. All starting and ending points are marked to complete the filling，which simplifies the calculation steps and reduces the algorithm complexity by half. Experiments have verified the superiority of the algorithm in terms of execution time when drawing complex instrument images. It is better than traditional algorithms in theory and experiment，and can better meet the needs of aerospace display instruments for acceleration of two—dimensional graphics. It has been successfully applied to In the instrument of the manned aircraft.

Keywords：graphics fast algorithm；display instrument；graphics drawing；graphics filling；manned spaceflight

0 引言

随着我国载人航天事业的不断发展，航天显示仪表作为空间站与航天员之间的数据窗口，如何高质高效地进行数据显示一直是一个不可回避的问题。航天显示仪表的存在是为了便于航天员观察空间站当前各系统的各种状态数据，而人眼在感知事物时，当画面切换频率高于40帧/秒，人眼才不会察觉到画面的闪烁，为保证仪表的画面质量，每幅图像从解析到显现保持必须小于25毫秒[1]。空间站在运行过程中，各种系统需要传递给宇航员的信息量极大，为了能够及时展现空间站状态，当数据信息发生变化时仪表图像的切换速度就必须要快，而显示图像的切换不同于幻灯片，每一帧二维图像都需要从点、线、圆、多边形开始绘制并进行图形填充[2]。

新一代航天显示仪表采用了DSP/CPU+FPGA的高速处理器架构。相比于民用的显示仪表，空间站中不适合使用体积更大、能耗更高的图形加速卡（graphics processing unit，GPU），无法使用民用领域一些利用GPU进行图形加速的新算法，所以必须选择适合于当前仪表硬件平台的图形显示算法，对其进行改进，减少运算时间，提高运算效率。此外，受嵌入式平台自身的计算资源总量的限制，图形显示算法应选择在保证不过多占用资源基础上提高处理速度的处理算法。航天仪表控件主要使用二维绘图算法进行绘制，所以改进二维绘图算法，减少绘制所需的计算量，提高仪表人机界面的绘制效率。每一点图像绘制效率的提高，积累到仪表显示上都会使仪表的单位时间显示信息量得到提升、使得仪表的实时性得到加强，使航天员对空间站的运行状态把握更为精准。

收稿日期：2021-05-18； **修回日期**：2021-06-08。

作者简介：张庆熙(1995-)，男，山东烟台人，硕士，主要从事图像处理和计算机视觉方向的研究。

引用格式：张庆熙，夏加高，李文新. 新一代载人航天器显示仪表的图形加速算法研究[J]. 计算机测量与控制，2021，29(7)：269-273.

1 二维绘图算法研究内容

二维图形绘制是仪表显示系统的基础，图形算法是一个专门的研究方向。每当图形算法运行速度有微小的提升，对于整个系统的运行都能够有节约大量时间资源的增益。一般地，二维图形算法研究内容方向可以分为两大类：图元（点、线、圆、多边形）算法研究和图形填充算法研究。

直线的绘制算法，主要有数值微分法、中点画线法、Bresenham 等几种经典的算法，近年来也提出一些新的算法，如多像素行直线生成算法[3]，双步算法进行直线绘制[4]。这些新算法在特殊的应用场景有较好的效果，但是程序实现有些难度。因此，本文在经典的直线绘制算法基础上进行研究。

圆的绘制算法较多，内接正多边形逼近算法、圆正交正多边形逼近算法和 8 分圆生成算法，也有人提出椭圆离散直线位移加速法[5]，圆弧快速绘制的方法[6]。这些算法的主要思想是利用圆的对称进行图形加速。

多边形算法的基础是直线算法，各种多边形的快速生成算法本质是直线算法的延伸[7]。

图形填充算法研究方向较多，经典有种子算法、边界扫描法等。张毅等人提出基于边界跟踪的任意区域填充算法[8]；邱国清提出了基于种子算法的环形扫描填充法[9]。通过对比分析，这几种填充算法的效率提高并不明显，主要是依靠硬件水平的提升来实现快速填充，所以本文在经典的种子算法上进行研究。

绘制到屏幕的图形需要有位置信息，离不开屏幕坐标系的设定[11]。坐标系的设置主要有两个方面内容，即原点位置和坐标轴方向。本文采用如图 1 的屏幕坐标系，屏幕的左上角为原点，向右为 X 轴正方向，向下为 Y 轴正方向。

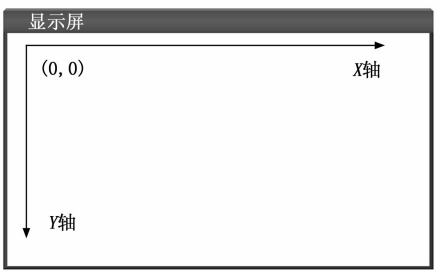

图 1 屏幕坐标系

2 图元加速算法研究

图形绘制过程中，由于图元（点、线、圆、多边形）的不同，因而需要采取相应不同的算法，本节主要就不同图元的绘制算法进行阐述，并提出可行的加速改进算法。

2.1 直线加速算法研究

直线绘图是 2D 绘图中最基本的绘图功能，是绘制其它图形的基础，如果直线绘制的效率高，则会提高整个仪表显示系统的运行速度。

绘制直线一般有三种函数，横线、竖线和斜直线[12-14]。横线、竖线的算法比较简单，但斜直线的算法比较多，时间和空间复杂度各不相同。其中最常见的算法有 Bresenham 直线生成算法[15]，在工程上应用广泛。

经典 Bresenham 直线生成算法的算法步骤如下：首先通过线段的起点（x_0，y_0）和终点（x_1，y_1）得出绘制直线的参考坐标轴，如果 x 坐标差值绝对值大于 y 坐标差值绝对值，则直线沿着 X 轴绘制，反之则沿着 Y 轴绘制；第二步，根据起始点坐标差值的正负号，决定坐标轴的绘制方向；第三步，得出直线斜率；第四步，沿某个坐标轴方向步进，计算另一坐标轴的坐标的误差项；第五步，修正误差项 d：

$$d = kx_{i+1} + b - y_i \tag{1}$$

如果误差项大于 0.5，则另一坐标值加 1 并绘点，同时修正误差项，反之误差项则保持原值。算法的流程图如图 2 所示。

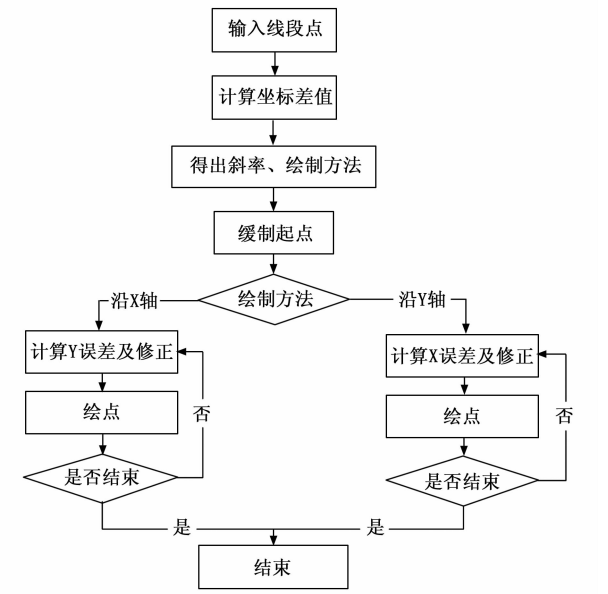

图 2 Bresenham 直线生成算法流程图

为了更好地满足需求，对 Bresenham 直线生成算法进行了改进，具体步骤如下：1）通过线段的起始点判断线段绘制的参考坐标轴；2）得出绘制线段的坐标方向；3）得出绘制线段的斜率；4）得出线段沿绘制坐标方向的中点坐标；5）沿绘制坐标轴方向得出误差项；6）修正误差项 d，同式（1）。如果误差项大于 0.5，则另一坐标值加 1 并绘点，同时向绘制的另一端点减 1 并绘点，即计算出两个坐标值，然后修正误差项，反之误差项保持原值。

算法流程图如图 3 所示。

通过算法流程图可以看出，改进的 Bresenham 直线生成算法每次循环可以绘制出两个点，比 Bresenham 直线生成算法节省循环次数，运行效率和直线的长度相关，在 200

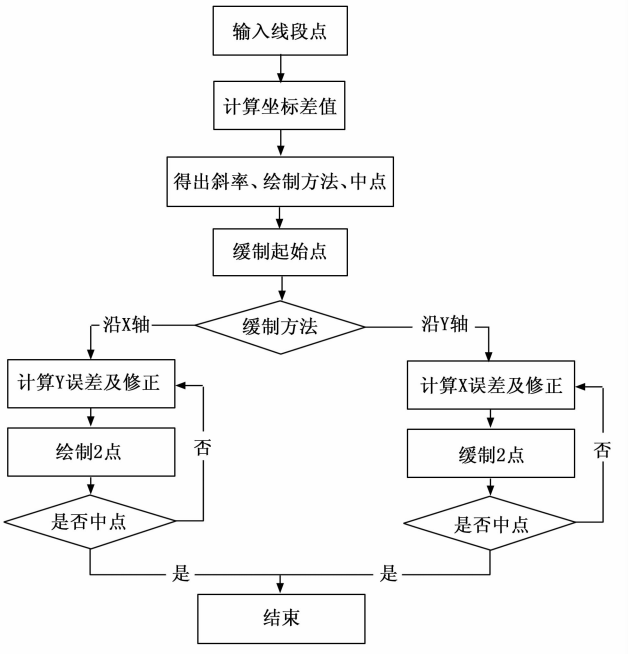

图 3　改进的 Bresenham 直线生成算法流程图

像素点的条件下，减少了 100 次循环，减少了 100 个计算步骤，减少了约 30%的计算次数。

2.2　圆的绘制算法研究

圆是 2D 几何绘图中的基本图形，其生成算法主要有内接正多边形逼近算法、圆正交正多边形逼近算法和 8 分圆生成算法。前两种算法主要是利用线段逼近圆弧的方法生成圆，其逼近的正 N 边形，只有 N 足够大，才能更加逼近圆。

零具有对称性，只计算圆上一部分的值，再通过对称性将值变换到其他象限可以极大的减少计算量。目前工程上常用的是八分圆生成算法和中心圆生成法。所谓八分法，只需要计算八分之一的图形坐标，其它部分通过映射即可以得出坐标，从而减少运算量。

对于第一象限内，靠近 Y 轴的八分之一圆弧（弧上点满足 $0\leqslant x\leqslant y$）如图 4，采用 Bresenham 直线算法的思想，其基本的方法是利用判别变量来判断选择最近的像素点，判别变量的数值仅仅用一些加、减和移位运算就可以计算出来。Bresenham 画八分圆算法也是用一系列离散的点来近似描述一个圆弧。

画圆时，第一个点取（0，R），设第 i 步已经确定点（x_i，y_i），下一步需要确定 A（x_{i+1}，y_{i+1}）位置，这个点只可能是 B（x_i+1，y_i）或者 C（x_i+1，y_i-1），由 B 和 C 到 A 的距离之差 d 大小决定。将 d 进行近似计算：

$$d=(x_i+1)^2+y_i^2-R^2-(R^2-(x_i+1)^2+(y_i-1)^2)=2(x_i+1)^2+2y_i^2-2y_i-2R^2+1 \quad (2)$$

将（0，R）代入上式可得到 $d=3-2R$。

当 $d\geqslant 0$ 时，下一点就选为 C 点，将 $y_{i+1}=y_i-1$ 代入式（2）得到：

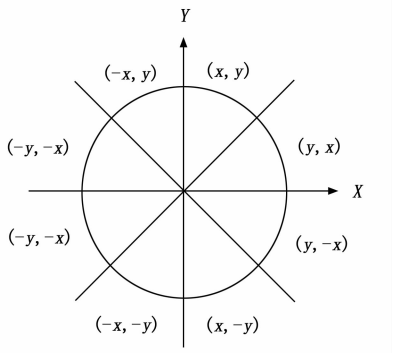

图 4　八分圆示意图

$$d_{i+1}=4(x_i-y_i)+10 \quad (3)$$

反之，选 D 点，将 $y_{i+1}=y_i$ 代入式（2）：

$$d_{i+1}=4x_i+6 \quad (4)$$

在计算每个（x_i，y_i）时，都利用对称性将区域外的其他 7 个点同时得到。

依次迭代，直到 $x\geqslant y$ 停止计算。

Bresenham 八分圆算法的流程图如图 5。

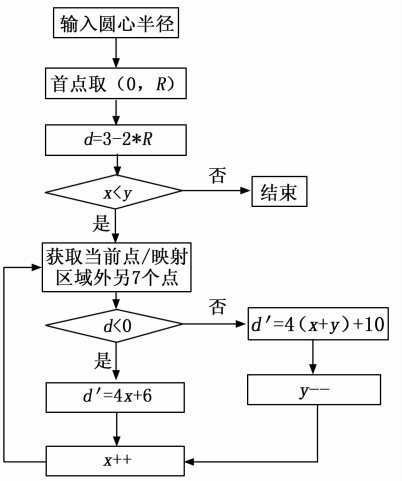

图 5　Bresenham 八分圆算法流程图

2.3　多边形绘制

在仪表显示系统中，规则图形只是图形的一部分，还有大量的图形是不规则图形，很多不规则图形可以通过多边形显示。

多边形绘制是以直线绘制为基础得到的。其主要思路为将多边形的顶点通过直线的绘制方法连接起来。直线的绘制算法采用改进的 Bresenham 直线生成算法。

多边形其主要算法如图 6 所示。

3　图形填充算法研究

图形填充是在完成图形绘制之后，对于具有相同色彩或灰度值的一定区域进行填充，以完成由线成面的过程[16]。本节主要就经典填充算法进行分析，并提出改进的填充算法。

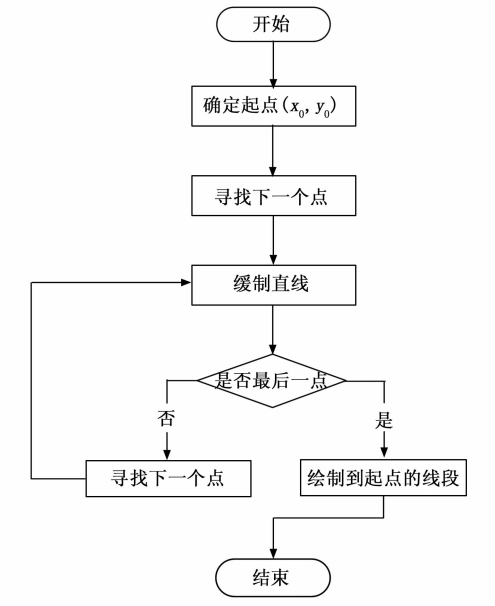

图 6 多边形绘制算法流程图

3.1 种子填充算法介绍

图形填充的算法有很多种[17-18]，其中较为有名的是种子填充算法，该算法思想简单，比较易于实现。

如图 7，种子填充图形算法的主要思想是在一个封闭的几何图形中找到一点，称之为种子，种子按照水平扫描，分别从两个方向扫描，即向左和向右扫描，扫描的终止条件为图形的边界。本行扫描完毕后，再向上或向下寻找新的种子，寻找到的种子条件是必须在图形边界内。

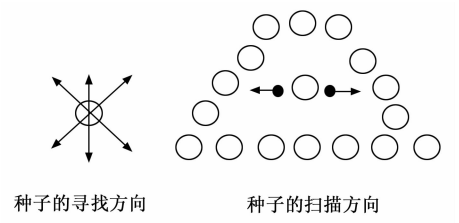

图 7 种子图形填充算法

种子图形填充算法主要有以下几个步骤：1）要计算种子数组的长度，数组的大小为 Y 轴坐标相减，然后种子数组置零；2）根据给定的首个种子按左、右两个方向进行水平扫描，分别扫到两个边界为止；3）种子扫描结束后，在相对应的位置（Y）上记录为 1，表明本行已经扫描结束；4）寻找新的种子，新种子可以向上寻找也可以向下寻找，向上寻找到了边界后，再由最下部向上寻找，种子寻找的条件是必须在封闭图形的范围内，种子可以沿 6 个方向寻找；5）判断种子数组是否已经满了，即种子数据的各个元素均为 1 时，说明填充完毕。种子填充算法的具体流程如图 8 所示。

种子图形填充算法可以适用于任意图形的填充，但是种子的寻找以及扫描边界的确定都需要耗费时间资源，所以种子图形填充算法的效率较低[19]。

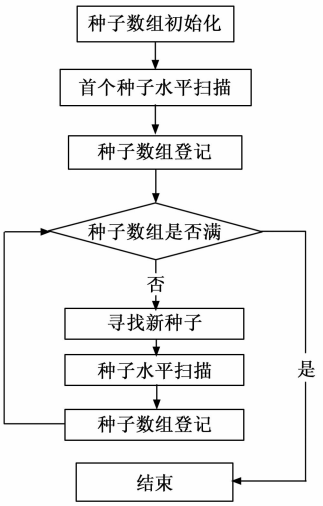

图 8 种子图形填充算法流程图

3.2 基于凸多边形的种子填充算法

在航天仪表显示系统中要经常使用图形填充算法，如果能提高图形填充的绘图效率，则可以提高屏幕的图像刷新率上限[20]，为了实现这一目标，本文提出了基于凸多边形的种子填充算法。

基于凸多边形的种子填充算法继承了种子填充算法的核心思想，根据凸多边形的几何特点，以凸多边形的一条“边”确定填充种子，记为填充起点；以余下的“边”作为图形填充边界，记为填充终点。将所有起点和终点进行对应划线，则将图形填充完成。

基于凸多边形的种子填充算法（以 X 轴为扫描方向）的步骤如下：

1）要根据凸多边形的顶点确定扫描种子的数量，即 X 方向扫描线的数量。根据凸多边形顶点的 Y 轴坐标找出最大值和最小值，种子数量为最大值和最小值之差加 1，设种子数量为 N。同时定义两个坐标数组，扫描起点数组和终点数组，数组的大小均为 N，即 POINT _ START［N］和 POINT _ END［N］。

2）从凸多边形的 Y 轴坐标的最小值顶点，沿其左侧到 Y 轴坐标的最大值顶点进行“画线”操作。“画线”是指调用直线绘制程序，将线的点坐标放置在 POINT _ START［N］数组中。

3）从凸多边形的 Y 轴坐标的最大值顶点，沿其右侧到 Y 轴坐标的最小值顶点进行“画线”操作。“画线”是指调用直线绘制程序，将线的点坐标放置在 POINT _ END［N］数组中，绘点的顺序从 $N-1$ 开始，到 0 结束，点的填充顺序与第二步相反。

4）绘线填充。根据 POINT _ START［N］和 POINT _ END［N］，从 0 到 N－1 依次进行绘线操作，则填充完毕。

对种子图形填充算法和基于凸多边形的填充算法复杂度进行比较。设如图 9 所示图形，分别利用种子图形填充算法和基于凸多边形的填充算法进行填充。

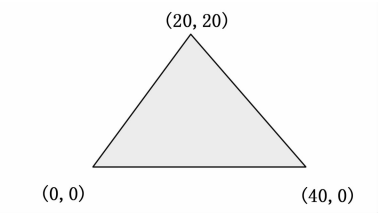

图 9　等腰三角形填充

表 1 是两种填充算法的计算步骤比较，表中的填充算法只计算坐标点的计算步骤，不将坐标点输出到屏幕的时间统计在内。从表 1 中可以看出，图 8 的图形利用基于凸多边形的填充算法比种子填充计算步骤少了 398 步，减少了几乎一半的计算复杂度。

工程实践中，凹多边形也可以分解为若干个凸多边形，利用基于凸多边形的填充算法进行图形填充，从而减少填充的计算步骤。

表 1　填充算法复杂度比较

序号	算法类型	计算步骤
1	种子图形填充算法	820
2	基于凸多边形的填充算法	422

另一方面，有些特殊的凹多边形也可以利用基于凸多边形的填充算法进行图形填充，如图 10 所示。

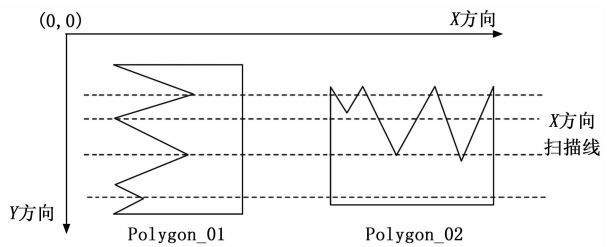

图 10　X（Y）轴有序多边形

图 10 中的多边形 Polygon _ 01 虽然不是凸多边形，但是仍然可以在 X 轴方向利用基于凸多边形的填充算法进行图形填充；而多边形 Polygon _ 02 则不可以利用该算法在 X 轴方向进行填充，但是可以利用该算法在 Y 轴方向进行填充，从而提高图形填充速度。为此，这类特殊的多边形进行了如下定义，称之为 X 轴方向有序多边形或 Y 轴方向有序多边形。

定义：在一个平面上，一个多边形与 X 轴任意扫描线相交有且只有两个交点，这个多边形称之为 X 轴方向有序多边形。

定义：在一个平面上，一个多边形与 Y 轴任意扫描线相交有且只有两个交点，这个多边形称之为 Y 轴方向有序多边形。

X 轴方向有序多边形可以利用基于凸多边形的填充算法在 X 轴方向进行图形填充；Y 轴方向有序多边形可以利用基于凸多边形的填充算法在 Y 轴方向进行图形填充。

另外圆是特殊的凸多边形，可以基于凸多边形的填充算法进行图形填充。

4　图形加速算法验证

本文算法的验证平台采用 DSP＋FPGA 平台，该平台为新一代航天显控仪表的通用架构，DSP 为国产 DSP6713，主频为 400 MHz，图形算法均在 DSP 处理器中完成，FPGA 选用国产的 BM3803，由 FPGA 完成对显存的控制输出。

仪表界面一般由若干仪表控制组成，提高复杂仪表控件绘图速度，即可保证整个航天仪表显示系统的显示效率。本文采用了较为复杂的角度仪表控件和电源加载控件进行对比验证。

用经典绘图算法和本文的图形加速算法完成角度仪的图形绘制。图 11 的角度仪表图像，其绘图用时对比如表 2。

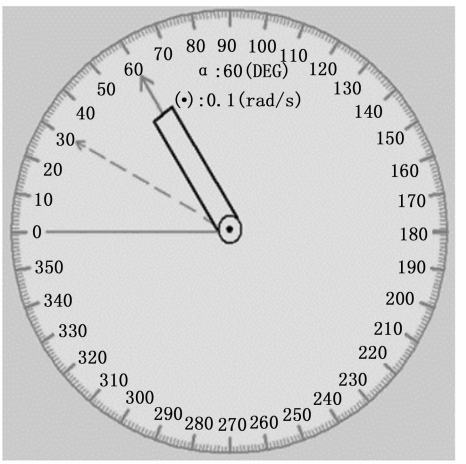

图 11　角度仪表控件

表 2　算法效率比较

序号	算法	用时/ms	效率提升
1	通用算法	8	35%
2	图形加速算法	5.2	

绘图算法和本文的图形加速算法分别生成如图 12 的电源加载控件，其绘图用时对比见如表 3。

表 3　算法效率比较

序号	算法	用时/ms	效率提升
1	通用算法	15	28%
2	图形加速算法	9.8	

通过比较用时，本文的图形加速算法比经典算法有比较明显的速度优势，图形绘制效率有较大的提升，可以满

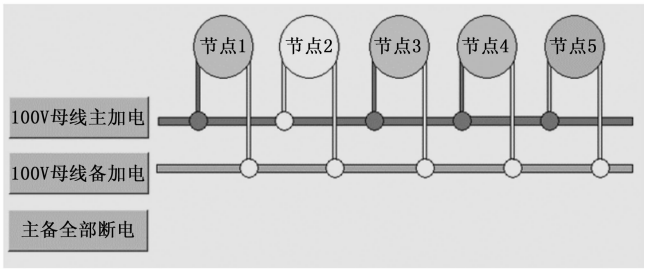

图 12 电源加载控件

足新一代航天器显示仪表的显示要求，已经成功地应用在载人航天器上。

如图 13 所示，该菜单指令界面为利用本文算法渲染生成的空间站某仪表操作交互界面，整个界面包括文字都是利用图形加速算法完成绘制填充，绘制图形精确，色彩填充完整。在绘制不同界面时，在不使用图形加速卡情况下，本算法平均刷新频率比其他图形绘制算法提高 15 Hz，本算法有明显的速度优势，占用计算资源量更少，更适合于空间站显示仪表的图形绘制。

图 13 某仪表操控交互界面

5 结束语

本文基于空间站对提高仪表显示质量、效率的需求提出了完整的改进的航天仪表 2D 图形绘制方法。改进的 Bresenham 直线生成算法和基于凸多边形的种子填充算法比改进前的算法在图形绘制的效率上有明显的提升。整套算法能够准确绘制空间站仪表的操控及显示界面，相比于其它的改进算法，提高了界面图形绘制和显示的实时性，以及画面刷新率的上限，同时规避了不能使用图形加速卡带来的硬件限制问题，能够在一定程度上满足空间站航天仪表对图形加速的需求，在空间站仪表操控显示界面的绘制上具有较好的工程实用性。

参考文献：

[1] 金 韬. 一种机载适航图形显示处理单元的设计与实现 [J]. 电脑知识与技术，2019 (18)：257－260.
[2] 高 静. 高速图形生成系统及其图形算法研究 [D]. 南京：南京航空航天大学，2015.
[3] 贾银亮. 基于 FPGA＋DSP 的飞机座舱综合图形显示技术研究 [D]. 南京：南京航空航天大学，2014.
[4] 刘 超. 基于 FPGA 的二维图形加速算法的设计与实现 [D]. 西安：西安电子科技大学，2015.
[5] 谢周标，周 毅，龙 斌. 二维椭圆硬件加速算法研究及其 FPGA 实现 [J]. 计算机工程与应用，2015，51 (3)：45－49.
[6] 张小永. 基于 FPGA 的圆弧快速生成算法及其应用 [J]. 电子技术与软件工程，2021 (3)：150－151.
[7] 贾庆仁，车德福，修春华. 煤矿地质成图中多边形快速生成算法 [J]. 测绘科学，2016，41 (12)：70－74.
[8] 张 毅，李昌华. 基于边界跟踪的任意形状区域填充算法 [J]. 计算机工程与设计，2015，36 (3)：725－728.
[9] 邱国清. 多边形图形的环状扫描线种子填充算法 [J]. 淮北师范大学学报 (自然科学版)，2017，38 (1)：64－67.
[10] 石志昕. 基本光栅图形生成算法研究 [D]. 济南：山东大学，2007.
[11] 孔全存，李成贵，张凤清. 主飞行仪表图形加速显示系统的 FPGA 设计 [J]. 电子技术应用，2007，33 (4)：62－64.
[12] 阮 航，张义伟，王 炜，舰载显控系统中二维矢量图形加速器的设计 [C] //中国造船工程学会电子技术学术委员会. 中国造船工程学会电子技术学术委员会 2017 年装备设计发展论坛论文集. 中国造船工程学会电子技术学术委员会：中国造船工程学会，2017：6.
[13] WANG J P，et al. Spatial Straight－Line Drawing Algorithm Based on Method of Discriminate Regions—A Control Algorithm of Motors [J]. Energies，2020，13 (19)：5002－5002.
[14] Zhang H R，et al. Seam－based variable－step Bresenham blending method for real－time video mosaicking [J]. Journal of Electronic Imaging，2016，25 (5)：12.
[15] OH Hyun Woo，et al. The Design of a 2D Graphics Accelerator for Embedded Systems [J]. Electronics，2021，10 (4)：469.
[16] 卫洪春. 扫描线多边形区域填充的改进与图像信息提取 [J]. 电子设计工程，2018，26 (12)：172－176.
[17] NATARAJAN K，KUPPUSAMY K. An Enhanced Method for Filling a 2D－Polygon [J]. Digital Image Processing，2010，2 (11)：486－489.
[18] Geraets W G M，DAATSELAAR A N VAN，VERHEIJ J G C. An efficient filling algorithm for counting regions [J]. Computer Methods and Programs in Biomedicine，2003，76 (1)：1－11.
[19] 黄 蕾. 图形图像中多边形区域填充与信息提取算法研究 [J]. 重庆科技学院学报 (自然科学版)，2020，22 (6)：65－69.
[20] CODREA MARIUS C，NEVALAINEN OLLI S. Note：An algorithm for contour－based region filling [J]. Computers & Graphics，2005，29 (3)：441－450.