

面向高密度印刷电路板圆形特征的快速铺铜算法

方敬苛¹,徐宁¹,张雪琳¹,孙贤伟^{1,*},胡建国²

(1. 武汉理工大学 信息工程学院,武汉 430070;2. 中山大学 深圳研究院,深圳 518057)

摘要: 随着印刷电路板的高度集成化,传统的芯片安装和散热无法满足整体尺寸和重量的要求。铺铜作为 PCB 设计环节中最后进行的关键步骤,在提高电路板性能和有效性方面具有重要作用。本文针对高密度印刷电路板上存在的圆形特征,提出了一种基于完整图形的信息分类和区域计算的快速铺铜算法,并对铺铜过程中产生的孤岛提出了搜索方案。其核心思想如下:首先根据图形信息对大量特征进行区域划分,然后根据碰撞关系对其分簇,基于分簇结果得到外包络边链和孤岛边链,最后通过环和孔的边链关系得到铺铜结果。算法在 3.20 GHz、16 GB 内存的 Windows 10 环境下运行,使用工业用例测试数据进行验证。实验结果表明,与开源的基于拟合多边形的铺铜工具相比,在面向大规模场景时,铺铜结果不依赖于拟合多边形边数,时间缩短了约 60%。

关键词: 印刷电路板;快速铺铜;孤岛搜索;圆形特征;轮廓求解

中图分类号: TP391.41

文献标识码: A

Fast shape filling algorithm for circular features in high-density PCB

FANG Jingke¹,XU Ning¹,ZHANG Xuelin¹,SUN Xianwei^{1,*},HU Jianguo²

(1. School of Information Engineering,Wuhan University of Technology,Wuhan 430070,China;

2. Shenzhen Research Institute,Sun Yat-Sen University,Shenzhen 518057,China)

Abstract: With the increasing integration of printed circuit board,traditional methods are unable to meet the requirements for size and weight. Copper plating,a crucial step in PCB design,plays a significant role in improving the performance and effectiveness of PCB. To address the numerous circular features on large-scale PCB,a fast shape filling algorithm is proposed. This algorithm utilizes graphic information classification and identifies islands that may arise during the copper plating process. The core idea of the algorithm is as follows: firstly,a large number of features are divided into regions based on their graphical information. Then,clustering is performed based on collision relationships. The outer contour chains and isolated island chains are obtained based on the clustering results. Finally,the result is obtained through the relationship between the contour chains of loops and holes. The algorithm is implemented in a Windows 10 environment with a 3.20 GHz processor and 16 GB memory,and industrial test data is used to validate its performance. The experimental results demonstrate that compared to an open-source copper plating tool based on fitting polygons,the proposed algorithm is independent of the number of polygon edges and achieves a time improvement of approximately 60%.

Keywords: printed circuit board;fast shape filling;island search;circular features;envelope chain

0 引言

随着半导体行业和微电子技术的高速发展,印刷电路(Printed Circuit Board,PCB)及其电子元器件在微型化、

集成化的同时,对算力的要求在不断提升。但会伴随产生芯片功耗和发热问题,严重影响电子元器件的寿命和可靠性。铺铜作为电路板设计的重要环节,将主要的地(GND、SGND、AGND)进行有效连接,有助于芯片散热。

基金项目:科技创新 2030—“新一代人工智能”重大项目(2021ZD0114600);深圳基金重点项目“基 20220019 软件自动化设计和生成算法研究”(JCYJ20220818102002005)。

方敬苛(硕士研究生),主要研究方向为电子设计自动化;徐宁(教授),主要研究方向为电子设计自动化、计算机软件与系统;张雪琳(硕士研究生),主要研究方向为电子设计自动化;孙贤伟(副教授),主要研究方向为无线传感网络算法设计、组合密码编码、图论;胡建国(副教授),主要研究方向为集成电路设计、电子设计自动化、物联网与人工智能。

* 通信作者. E-mail:sunxw@whut.edu.cn.

参考文献[1]的实验结果表明,PCB 表面金属层对热辐射作用的贡献最大,有铜 PCB 元器件散热能力明显高于无铜 PCB。同时,铺铜还可以降低地线阻抗,降低电磁辐射干扰(Electro Magnetic Interference, EMI)。利用计算机辅助设计工具进行物理自动化设计,在仿真阶段就对 PCB 电路板上的铺铜量进行测量,可以有效降低芯片设计成本。

目前,国内外对仿真设计中的铺铜算法研究仍然较少。参考文献[2]提出了基于多边形的快速消除重叠部分的算法,与几何的并运算类似,通过预处理方式减小后续栅格化工作量。参考文献[3]提出了一种基于渐进点扩展算法的铜层自动化设计优化方法,将铜层元素栅格化为离散单元,基于贪心策略对铜元素进行扩张,形成铜层。然而,这些算法受网格大小的影响,最终生成的铜层结果易产生偏差。

同时,计算几何上的研究和发现对本文算法也有一定启发。参考文献[4]针对二维图形提出了基于元素的布尔运算方法,将线架图形转换为实体数据结构,在集合上进行运算,执行时无需考虑交点和顶点。参考文献[5]提出了基于位图法的几何求并运算,将图形看作一个由 0、1 组成的矩阵,内部区域设置为 1,外部区域用 0 表示,填充过程可以转换为 0—1 的逻辑过程。参考文献[6]~[9]基于多边形的方法计算两两图形之间的交点,通过判断出点和入点分析特征并根据具体运算决定结果走向。然而,此类算法仅支持多边形,对于圆形,需要利用圆内接正多边形去逼近圆,将其近似切割成多段直线段后才能处理。PCB 电路板上往往存在许多 Pin、过孔等圆形信息,已有算法不能适应大规模集成电路版图。

本文针对大规模 PCB 上存在多密度圆形特征的特点,在区域划分的基础上提出了基于图形信息分类和计算的整版铺铜算法,包括外包络边链构成和孤岛查询过程。传统的铺铜方法需要大量经验和手动操作去优化结果,本文算法在完成铜皮构建的基础上可以实现尖角的自动平滑,并搜索版图中存在的孤岛,避免了人工修补铺铜区域的时间消耗。铺铜结果在保证残铜率的同时具有更高的精度,与如今常用的开源工具 Kicad 中的铺铜功能相比,采用非离散化的方法也使该算法时间开销更小。

1 问题描述

1.1 相关定义

铺铜是根据选定区域用铜箔连接同网络下的 Pin 和 Track 等元素的过程,该选定区域必须是一个闭合的面。整版铺铜是在满足设计规则要求的前提下,通过给定的板框外轮廓信息与板框内的 Pin 信息求解得到最后的铺铜结果。铺铜仿真示意图如图 1 所示。

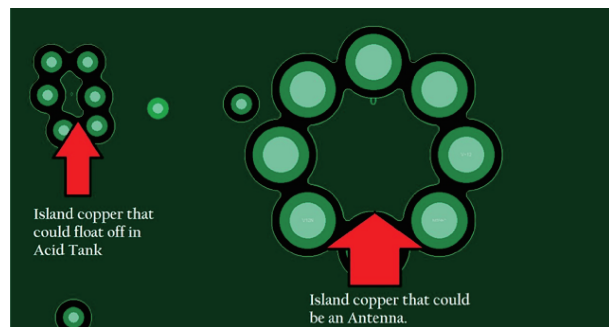


图 1 铺铜仿真示意图

Fig. 1 Simulation diagram of copper

由工业软件的行业标准^[10]可知,铺铜过程需要满足的设计规则如下:

- ① 线网连接正确性。与铜皮属于同一线网的 Pin 需要与铜皮相连,属于不同线网的 Pin 铜皮必须避让。
- ② 间距。铜皮边界在避让不同线网特征时需保持最小间距 $d_{\min}$ 。
- ③ 平滑。为避免天线效应,铺铜结果应尽量平缓,不能出现尖角,通常采取倒角操作。

整版铺铜的优化参数^[10]如下:

- ① 孤岛。又称死铜,指在电气特性上没有与任何地方连接的铜箔,是由于器件之间的间距小于避让距离对铜皮造成割裂产生的。

孤岛的存在不利于工业生产,若无法及时发现铺铜过程中产生的孤岛,则会增加生产成本,所以需要算法寻找孤岛。

- ② 残铜率。即内层铺铜面积与整板面积的比率。在符合工业制造标准的前提下,根据残铜率指标,可铺设铜皮处尽可能铺铜。

1.2 数学模型

根据 Jordan 闭合曲线定理,在二维平面上的任何一条简单闭合曲线都会将平面分成 3 个区域:边界区域、内部区域和外部区域。

多边形的表示可以由一连串边链组成。多边形分为两种类型:岛和孔。岛型有一个面积有限的内部区域和一个面积无限的外部区域。孔型具有面积有限的外部区域和面积无限的内部区域。通常规定一个顺时针方向为标准方向,根据方向是否相反描述两者区别^[8]。由此,从数学角度表达铺铜问题:岛型多边形和孔型多边形的共同区域形成铺铜区域,即在电路板边界基础上,挖去铜皮避让形成的孔,得到的几何图形为最终铺铜结果。几何模型示意图如图 2 所示。

以下为问题描述的相关结构:

- ① 线, $Line(start\ point, end\ point)$ 。由起点和终点构成的边。

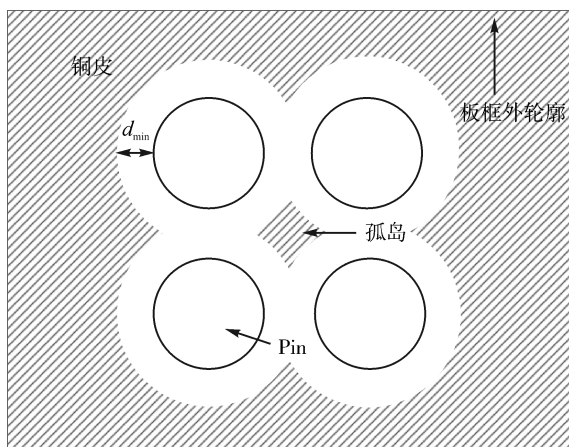


图 2 几何模型示意图

Fig. 2 Schematic diagram of geometric model

② 弧, $Arc(center, r, startpoint, endpoint, clock)$ 。由圆心、半径、弧段起点、弧段终点和时钟方向共同构成的边。

③ 环, $List = \{r_1, \dots, r_n, r_1\}$, 其中 $r_i \in \{Arc, Line\}$, $i = 1, 2, \dots, n$ 。环是由多段边组成的序列, 通常用环形链表存储。现边的起点为上一条边的终点, 现边的终点为下一条边的起点, 形成一个封闭的环。

④ 边界和孔, 按特定时钟方向排列的边形成的环, 且两者方向不同。若边界为逆时针, 则孔为顺时针。

⑤ 铜皮, $B = \{List_1 \cup List_2, \dots, List_{m-1} \cup List_m\}$ 。由边界和孔以及其顺逆时钟关系组成的边链结构, 一个铜皮只包含一个边界, 但可以存在多个孔。

2 本文算法

2.1 算法概述

首先, 对数据进行初始化。通过区域划分对输入的点集信息进行处理, 并利用聚类算法对数据分类, 得到多个初始簇。

然后, 对簇结果用包络边链生成算法寻找外包络, 其中, 边链优化保证结果满足边链平缓的约束, 再利用孤岛搜寻算法找出所有存在的孤岛。

最后, 将板框外轮廓、包络边链生成算法和孤岛搜寻算法生成的边链整合并输出铜皮结果。

2.2 数据预处理

为满足 PCB 版图上特征与铜皮之间不覆盖, 并保证一定避让距离的约束, 需要对原几何轮廓进行等距离外扩, 外扩距离为最小间距。圆形外扩保留圆心信息, 更新半径即可。

根据相对位置关系将圆的类型分为两类并进行以下定义:

① 圆集群。若干圆形成的簇, 圆集群内的圆之间存在碰撞关系或者连通关系。

② 孤立圆。又称离群圆, 与其他圆不存在碰撞关系或连通关系, 没有形成集群的圆。

圆分类和对应边链示意图如图 3 所示, 两类圆对应两类外包络边链。圆集群对应外包络边链为类 1, 孤立圆对应外包络边链为类 2, 两种外包络边链的生成算法不同, 具体生成算法将在下一节介绍。

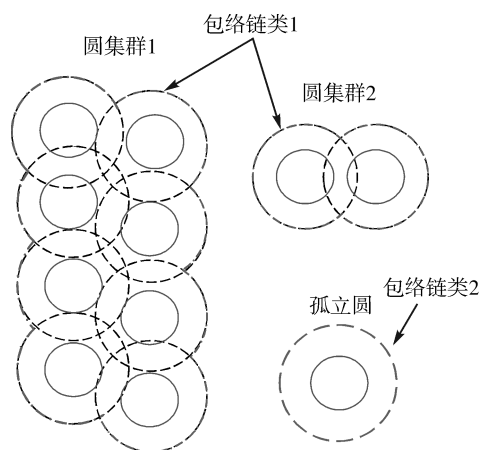


图 3 圆分类和对应边链示意图

Fig. 3 Schematic diagram of circle classification and corresponding edge chains

数据预处理主要包括两个阶段: 领域表构建和聚类阶段, 得到分类后的若干圆集群和孤立圆。

在聚类前通过区域划分得到圆的邻域表, 可以对海量圆信息进行筛选, 能有效减小聚类过程中对圆的搜索次数。领域表(neighbors)构建算法如下:

参数. 已知两圆 c_i, c_j . $eps = r_i + r_j$, $distance(c_i, c_j)$: C_i 和 C_j 之间的圆心距离

输入. 圆形集合 $C = \{c_1, c_2, \dots, c_n\}$

输出. 所有圆集群集合 $G = \{g_1, g_2, \dots, g_m\}$ 和孤立圆集 $C' = \{c_{j_1}, c_{j_2}, c_{j_3}, \dots, c_{j_i}\}, j_i \in \{1, 2, \dots, n\}$

Step1. 将原始圆外扩 d_{min} , 更新 $r = r + d_{min}$

Step2. 根据圆心信息构建二维 Kd-Tree

Step3. 圆的领域表生成

Step3.1. 遍历 Kd-Tree, 计算节点边界范围, 边界范围为该节点对应圆的两倍半径。比较树中每一层维度值是否在边界范围内, 若不满足, 则在子树中寻找, 满足条件节点进入步骤 3.2

Step3.2. 进一步比较是否满足 $distance \leq eps$, 满足则将其加入该圆的 neighbors 中

Step3.3. 重复步骤 3.1~3.2, 直到获得所有圆相应的领域表

Step4. 聚类算法得到分簇结果和孤立圆

其中, 算法采用的是 Euclidean 距离, 若给定圆 $C_1(x_1, y_1)$ 和 $C_2(x_2, y_2)$, 具体公式如下:

$$\text{distance}(c_1, c_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (1)$$

聚类算法基于 DBSCAN 算法^[11]思想, DBSCAN 聚类结果中的簇和噪声点对应圆集群和孤立圆。

2.3 外包络边链生成

如图 3 所示的两类圆对应的包络边链有如下定义, 采取的外包络边链生成算法不同。

① 圆集群生成边链类 L1, 需要考虑圆与圆之间的覆盖关系导致的边链切断和删减。

② 孤立圆生成边链类 L2, 由于孤立圆与其他特征间没有碰撞关系, 不会出现边链转换情况, 因此外扩圆形成的环即为孔。

本节针对圆集群边链提出边链生成算法, 分为外围圆寻找、边链生成和边链优化与改进 3 个步骤。

2.3.1 外围圆寻找

已知, 边链类 L2 的生成只与圆集群中的边界圆相关, 本步骤通过外围圆寻找算法找到参加边链形成的圆, 并按逆时针方向输出, 为后续边链形成做准备。

输入. 分簇结果 $G = \{g_1, g_2, \dots, g_m\}$ 中的 g_m

输出. 逆时针排序的圆链表 $L = \{c_1, c_2, \dots, c_n\}$

Step1. 令比较向量 $V = (1, 0)$, 初始化圆簇中所有圆 $visit = 0$

Step2. 找到簇中最左下角的圆记为初始圆 c_{begin} , 令圆

$c = c_{begin}$

Step3. 遍历圆 c 的领域表, 与其中 $visit = 0$ 的圆得到若干圆心向量

Step4. 计算 Step4 中圆心向量与比较向量间的夹角

Step5. 记最小夹角的圆 c_{cur} 和对应向量为 Q_{min} , 并令圆 c_{cur} 的 $visit = 1$

Step6. 更新比较向量 $V = Q_{min}$, $c = c_{cur}$

Step7. 重复 Step3~Step6, 直至回到圆 c

Step8. 得到最终的结果链表 L 。

寻找外围圆示意图如图 4 所示, 已知该集群为某簇中的无序圆, 按以上规则寻找可得到按逆时针排布的循环链表 $C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_4 \rightarrow C_5 \rightarrow C_6 \rightarrow C_7 \rightarrow C_8 \rightarrow C_1$ 。

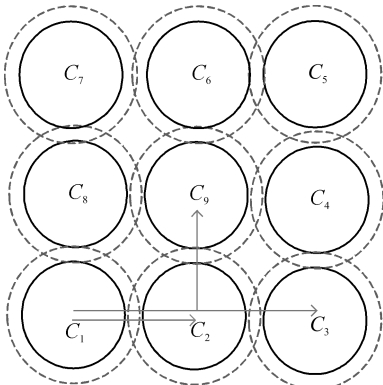


图 4 寻找外围圆示意图

Fig. 4 Schematic diagram of searching for peripheral circles

2.3.2 边链生成

外围圆算法已保证圆的逆时针排列顺序, 边链形成过程可以简化为两圆之间的关系。

若两圆之间相切, 则相切点即为弧段的起点或止点。若两圆之间相交, 则需根据相交的交点关系选取需要添加到链表里的弧形边链点, 最终形成环。所以, 选取哪条弧形边链添加到环形链表中, 是本节需要解决的主要问题。

已知圆与圆若相交, 则必存在两个交点, 圆与圆的交点可以通过已知信息和余弦定理得到。

通过簇的走向和位置关系将两交点分为内部点和外部点, 其中, 外部点为需要保留的最终形成弧段的交点。

外部点判断示意图如图 5 所示, 记 P_1, P_2 为 C_1 和 C_2 的交点, 记 C_1C_2 与水平轴的夹角为 α , C_1P_1 和 C_1P_2 与水平轴的夹角为 β_1 和 β_2 。

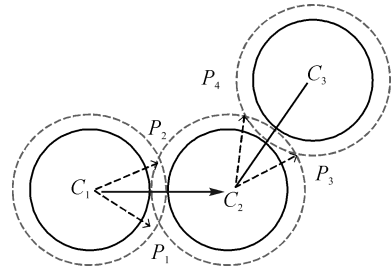


图 5 外部点判断示意图

Fig. 5 Schematic diagram of external point judgment

两交点可分为外部点和内部点(相切则无需判断), 其判定规则如下:

若 $\alpha > \beta_i, i \in \{1, 2\}$, 则对应 P_i 点为内部点。

若 $\alpha < \beta_i, i \in \{1, 2\}$, 则对应 P_i 点为外部点。

由此, 判定点类型后, 保留外部点, 并根据一个圆上的两个外部点和圆信息形成弧段, 更新循环链表信息, 生成边链。

2.3.3 边链优化与改进

在边链形成过程中, 两圆交点处会形成尖角, 不符合铺铜要求, 需要对该结果进行优化和处理, 主要步骤包括倒角和删除自相交, 在得到链表的过程中同步进行。

① 倒角。根据平滑需求, 利用公共外切圆形成的弧段平滑尖角。两圆的外切圆存在两个, 需要判断其中哪一个圆为外部圆, 外部圆上的弧段作为平滑弧保留。判断外部圆和内部圆的方法与外部点和内部点相同, 在此不作赘述。判断结束后, 根据已知两圆信息和选定的平滑圆半径参数求取外切圆。

如图 6 所示, 已知 $\cos \alpha = \frac{|AP|}{|AC|}$, 可以求得角 α 。其

中, $|AC| = r_A + r_C$, $|AP| = \frac{1}{2}|AB|$ 。

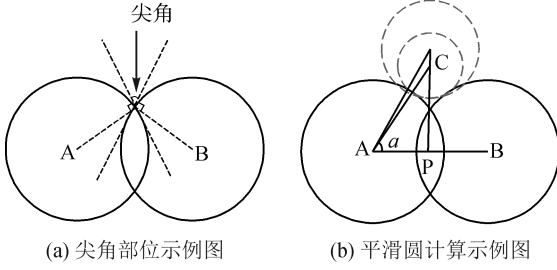


图 6 倒角过程示意图

Fig. 6 Schematic diagram of chamfering process

圆 C 的圆心求解公式如下：

$$\begin{cases} x_C = x_A + (r_A + r_C) \cos \alpha \\ y_C = y_A + (r_A + r_C) \sin \alpha \end{cases} \quad (2)$$

由此得到平滑外切圆,根据外切圆切点和圆信息得到平滑边链,插入原边链中并更新边链。经过处理后两圆之间的边链会由三段弧组成,形成平滑的闭合环。

② 删除自相交。在寻找外部平滑圆弧时,如果圆的分布出现“K”字形(见图 7),即排布出现内凹情况,此时的平滑弧计算会出现错误,可能会出现自相交情况,需要对自相交线段删除并再次平滑。删除和再平滑步骤如下:

Step1. 遍历外部点连接形成的线段,检测任意 2 条线段之间是否有交叉

Step2. 若无交叉,则未出现自相交情况,结束本次处理;否则,出现自相交情况,进行下一步

Step3. 提取交叉线段,并将对应弧段删除,重新调用平滑算法,由此可得出无自回路现象的圆形簇外轮廓线

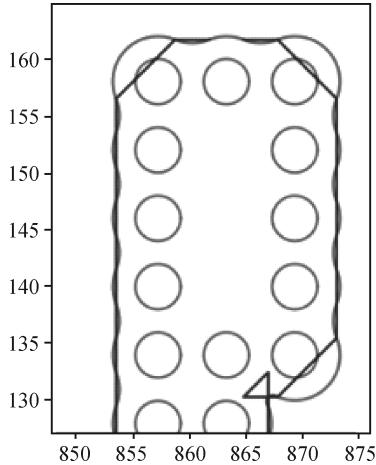


图 7 “K”字型自相交回路连线示意图

Fig. 7 Schematic diagram of K-shaped self-intersect circuit connection

2.4 孤岛搜索

输入: 簇中所有集群圆 $\{C_1, C_2, \dots, C_i\}$

输出: 孤岛边链集合 L

Step1. 计算圆 C_i 与其领域圆的交点记为 (A_{ij}, B_{ij}) , 则该圆

上所有交点为集合 $D_i = \{A_{i1}, \dots, A_{ij}, B_{i1}, \dots, B_{ij}\}$

Step2. 按顺时针排列 D_i 中的点, 记为集合 V , 表示为 $V = \{F_{i1}, F_{i2}, F_{i3}, \dots, F_{ij}\}, F_{ij} \in D_i$

Step3. 由 V 中相邻点之间是否存在点对关系, 构造一个无向图 $G = \{V, E\}$, 构造规则见式(3)

Step4. 在无向图 G 中搜索环

Step5. 根据无向图中环的顶点与圆的映射关系得到环对应的边链 $List_n$, 将该边链加入 L

Step6. 重复 Step3~Step4, 直至所有环都访问

Step7. 输出孤岛边链集合 L

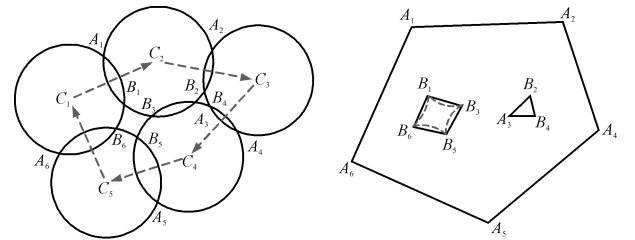
其中, 图的构造规则如下:

$$E(F_{i_1 j_1}, F_{i_2 j_2}) = \begin{cases} 1, & \text{if } i_1 = i_2, j_2 = j_1 + 1, \\ & \text{and } F_{i_1 j_1}, F_{i_2 j_2} \text{ not pair} \\ 0, & \text{else} \end{cases} \quad (3)$$

若相邻的两个交点属于成对关系, 即由同样的两个圆相交产生, 则两个交点在 G 中无边相连。若相邻的两个交点属于非成对关系, 则两个交点在 G 中有边相连。

圆簇无向图生成示意图如图 8 所示, 已知图 8(a) 圆簇内圆和圆上交点信息, 根据图构造规则, 可得图 8(b) 无向图。在图中寻找环。以环 $B_1 \rightarrow B_3 \rightarrow B_5 \rightarrow B_6$ 为例, 其中 4 条边与圆的映射关系可以对应圆 C_2, C_4, C_5, C_1 。根据顶点和对应的圆可以形成由 4 段圆弧构成的环, 如图 8(b) 中虚线所示, 即为该环寻找到的孤岛。

已知基于拟合多边形计算^[2]的算法时间复杂度为 $O(n^2)$ 。与之相比, 本文算法从完整的图形信息出发, 并通过区域划分和图形分类减少图形交点计算, 预处理区域划分和分簇部分的时间复杂度为 $O(n \log n)$ 。其中, n 的大小为圆形个数。后续边链生成算法的时间复杂度为线性。同时, 拟合多边形算法的计算时间和精度受拟合边数的影响, 而本文算法不存在该问题。



(a) 圆簇示例图

(b) 圆簇构造的无向图

图 8 圆簇无向图生成示意图

Fig. 8 Schematic diagram of generating undirected graphs for circular clusters

3 实验结果

程序采用 C++ 语言编写, 在 AMD Ryzen 7 5800H with Radeon Graphics 3.20 GHz 环境下进行实验, 算法的实现和调试运行主要是在 Windows 系统下使用 Mi-

crosoft Visual Studio 2019 进行。

本文预处理阶段中的算法优势在于针对大规模数据,通过分区和聚类对数据进行处理,可以有效减少后续计算的时间。该部分算法的时间开销与聚类得到的团簇数量和大小密切相关,如表 1 所列,展示了部分测试用例数据下聚类后的团簇数量和大小。

表 1 聚类算法团簇数量

Table 1 Number of clusters in clustering algorithm

测试用例	元素数量	聚类簇个数	平均簇大小
Case1	20	2	10
Case2	100	8	33.33
Case3	500	15	50
Case4	3 000	33	90.09
Case5	10 000	78	89.743
Case6	30 000	189	123.73
Case7	50 000	221	226.24

部分测试用例元素数量及运行时间如表 2 所列。基于拟合多边形的计算方法精确度会有一定丢失,该指标无法通过实验指标进行证明,所以未在实验表格中列出。可以看到,在元素数量较少时,本文算法时间与拟合多边形的运行时间基本一致,随着元素数量的增加,本文算法时间与拟合多边形方法运行时间相比得到了明显优化,Case5 中的运行时间缩短了约 60%。算法时间对比图如图 9 所示。

表 2 部分测试用例的运行时间

Table 2 Runtime of test cases

测试用例	元素数量	拟合多边形算法运行时间/s	本文算法运行时间/s
Case1	20	0.065	0.0817
Case2	100	0.213	0.3470
Case3	500	8.522	2.4310
Case4	3 000	44.734	21.2530
Case5	10 000	122.110	59.2980
Case6	30 000	258.300	71.5900
Case7	50 000	379.200	104.2000

同时,在实际情况中,拟合多边形算法的时间受拟合多边形的边数影响较大,若要保证高精度,则运行时间将更长。与该算法相比,本文的铺铜算法在大规模数据测试下仍可保证精度和较快的运行时间。

部分测试用例的铺铜结果如图 10~图 13 所示。

4 结 语

本文提出了一种针对大规模 PCB 上圆形数据的快速

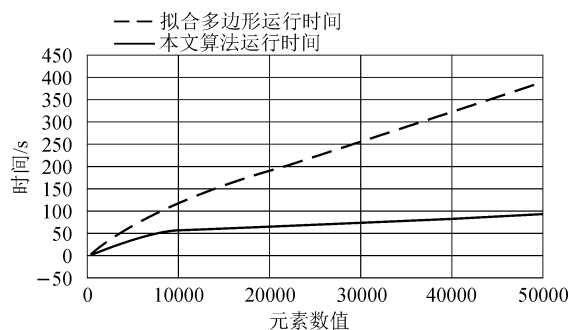


图 9 算法时间对比图

Fig. 9 Algorithm time comparison chart

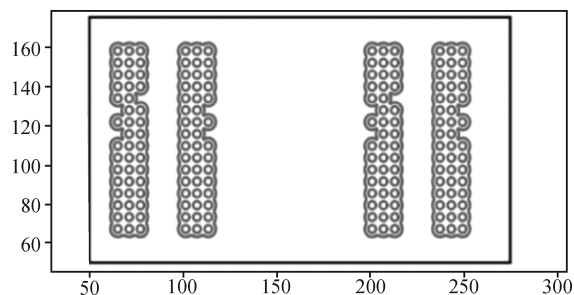


图 10 多圆簇铺铜结果图

Fig. 10 Copper result of multiple circular clusters

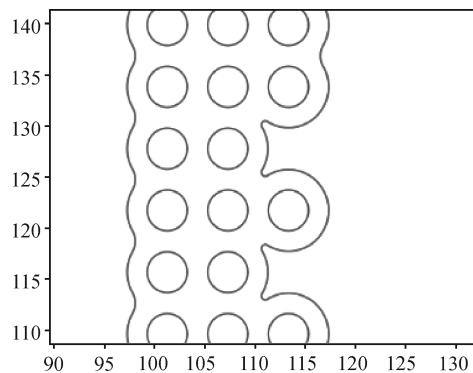


图 11 多圆簇铺铜局部图

Fig. 11 Partial copper view of multiple circular clusters

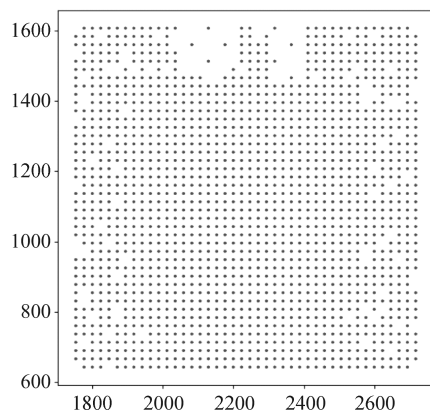


图 12 密集 Pin 铺铜结果图

Fig. 12 Copper result of dense Pin

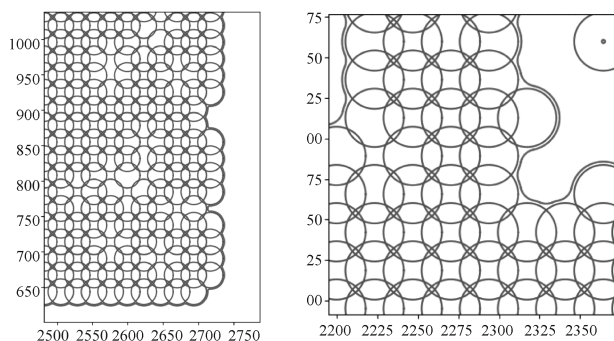


图 13 密集 Pin 铺铜局部图

Fig. 13 Partial copper view of dense Pin

铺铜方法,将铺铜过程转换为满足约束条件的几何计算过程。用外环和内环形式描述 PCB 上的铜皮。同时,算法能够检测铜皮结果中存在的孤岛,为后续删去孤岛的操作提供便利。从实验结果可以看出,与传统拟合圆形的算法相比,精确度更高,且运行时间和计算结果不受拟合和降维过程影响。同时,对孤岛提出了相应的处理方法。然而,目前的算法在传统的多边形基础上仅针对弧的问题进行了改进,对于复杂曲线的情况研究还不完善。在今后的研究工作中,将针对该问题进行深入的研究,尽可能地保证快速铺铜算法能够适用于任意图形。

参考文献

- [1] Y ZHANG. Improved Numerical - Analytical Thermal Modeling Method of the PCB With Considering Radiation Heat Transfer and Calculation of Components' Temperature[C]//in IEEE Access, 2021: 92925 - 92940.
- [2] W Y CHIU, C L LIN. An Efficient De - Overlap Algorithm for PCB Layout Patterns with Circular - Arc[C]//2016 International Computer Symposium (ICS), Chiayi, Taiwan, 2016: 374 - 379.
- [3] Y TIAN, A J FORSYTH. Automated Copper Layer Design

- and Optimization Tool based on Progressive Point Expansion Algorithm for Switch Mode Power Supplies[C]//2021 IEEE 22nd Workshop on Control and Modelling of Power Electronics (COMPEL), Cartagena, Colombia, 2021: 1 - 6.
- [4] FAN JUNFU. DWSI: an approach to solving the polygon intersection - spreading problem with a parallel union algorithm at the feature layer level[C]//Boletim De Ciencias Geodesicas, 2014: 159 - 182.
- [5] J FAN, H HE. Rasterization Computing - Based Parallel Vector Polygon Overlay Analysis Algorithms Using OpenMP and MPI[C]//in IEEE Access, 2018, pp. 21427 - 21441.
- [6] T PECHLIVANOGLU, M ALSAED. MRSweep: Distributed In - Memory Sweep - line for Scalable Object Intersection Problems[C]//2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), Sydney, NSW, Australia, 2020: 324 - 333.
- [7] PAUL R, AMAN M A A, SARKAR A, et al. A combinatorial algorithm to compute set operations on simple isothetic polygons [J]. Multimed Tools Application, 2023: 6647 - 6666.
- [8] Z SHANXIN, Q RUILIAN. An Algorithm for Computing the Union, Intersection and Difference of Two Polygons[C]//2010 Second International Conference on Computer Research and Development, Kuala Lumpur, Malaysia, 2010: 344 - 348.
- [9] MARTÍNEZ, FRANCISCO. A new algorithm for computing Boolean operations on polygons[C]//Comput. Geosci. 2009: 1177 - 1185.
- [10] Cadence 官网. <https://community.cadence.com>.
- [11] MARTIN ESTER, HANS PETER KRIEDEL. A density - based algorithm for discovering clusters in large spatial databases with noise[C]//In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD96). AAAI Press, 1996: 226 - 231.

(责任编辑:薛士然 收稿日期:2023-12-25)