山东大学
SHANDONG UNIVERSITY

毕业论文（设计）

论文（设计）题目：

基于数学表达的隐式费马螺旋线生成算法

姓　　名　周若淼
学　　号　201905130193
学　　院　计算机科学与技术学院
专　　业　计算机科学与技术
年　　级　2019 级
指导教师　赵海森

2023 年 5 月 7 日

摘　　要

费马螺旋线是一种特殊的螺旋结构，它的性质有利于路径规划算法形成全局连续的路径。连通费马螺旋线生成算法是基于费马螺旋线提出的路径规划算法。它能够形成全局连续的、低曲率的路径，用于填充任意的连通 2D 区域，已被应用于机器人轨迹规划、数控加工、FDM 打印等多个领域。但目前的算法也存在一些问题：一、存在欠填充区域，应用于 3D 打印时会降低打印质量；二、连通费马螺旋线缺乏像 Peano 曲线或 Hilbert 曲线一样的规律性和数学严谨性。

本文对已有的连通费马螺旋线生成算法提出了可能的改进方向，试图改善目前算法存在的问题。本文的主要工作是：一、重点对螺旋线的生成方式做了探究，尝试了基于距离场生成螺旋线及基于线性组合生成螺旋线两种方式；二、对一些问题提出了设想，以期为后续研究提供可能的方向。本文对算法改进的贡献是：一、通过使用数学表达式生成费马螺旋线的方式，提高费马螺旋线的数学严谨性。二、使用改进方法生成的螺旋线，其能填充的区域具有新的形态，它有两个突出的“接口”，或许能够更有利于区域间螺旋线的连接。三、后处理阶段用于曲线平滑，会增多欠填充区域；本文对螺旋线生成阶段的曲线平滑方法做出了设想，若能够实现，可取消后处理阶段，减少欠填充区域，提高打印质量。

关键字：路径规划；费马螺旋线生成；数学表达

ABSTRACT

Fermat spiral is a special spiral structure, and its properties facilitate path planning algorithms to form globally continuous paths. The connected Fermat spiral generation algorithm is a path planning algorithm based on Fermat spiral. It can form globally continuous, low curvature paths for filling any connected 2D region, and has been applied in multiple fields such as robot trajectory planning, CNC machining, FDM printing, and so on. However, the current algorithm also has some problems: first, there are underfilled areas that can reduce printing quality when applied to 3D printing; second, The connected Fermat spiral lacks the regularity and mathematical rigor of the Peano curve or Hilbert curve.

This paper proposes possible improvement directions for existing connected Fermat spiral generation algorithms, attempting to improve the existing problems of the algorithms. The main work of this article is as follows: first,The focus is on exploring the generation methods of spiral lines, attempting to generate spiral lines based on distance fields and linear combinations;second, Suggestions were proposed for some issues in order to provide possible directions for future research. The contribution of this paper to algorithm improvement is as follows: first, By using mathematical expressions to generate Fermat helices, the mathematical rigor of Fermat helices is improved. second, The spiral generated by the improved method has a new shape in the area it can fill, and it has two prominent "interfaces", which may be more conducive to the connection of spiral lines between regions. third,The post-processing stage is used for curve smoothing, which will increase the underfilled area; This article proposes a curve smoothing method for the generation stage of spiral lines. If it can be achieved, it can reduce underfilled areas and improve printing quality.

Key Words: Tool-path generation, Fermat spiral generation, Mathematical expression

目　录

第 1 章 绪　论

1.1 选题背景和研究意义

费马螺旋[1]是一种特殊的螺旋结构，它由两个交错的子螺旋在中心点处连接而成，如图 1-2（e）所示。这种螺旋的起点和终点都位于边界处，有利于螺旋之间的连接，进而形成全局连通的路径。

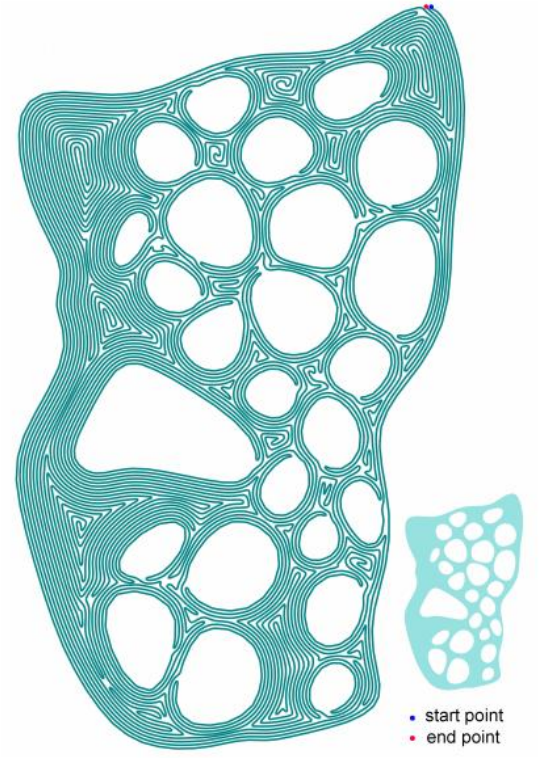

图 1- 1 连通费马螺旋线[2]

赵海森[2]等提出的连通费马螺旋线路径生成算法，是一种基于费马螺旋的路径规划算法。该算法能够形成全局连续、低曲率、等路径间距的路径，用于填充任意的连通 2D 区域，如图 1- 1 所示。算法流程是：首先将要填充的区域分成一些子区域；之后在各子区域分别填充螺旋线；并在区域间进行螺旋线的连接。该算法已被应用于机器人轨迹规划、数控加工、FDM 打印等多个领域。但是，目前的算法仍然存在一些问题，如：存在欠填充区域，应用于 3D 打印时会降低打印质量；连通费马螺旋线缺乏规律性和数学严谨性等。

本文的工作是，对现有的连通费马螺旋线路径生成算法提出可能的改进方向。主要包括：一、重点对螺旋线的生成方式做了探究，尝试了基于距离场生成螺旋线及基于线性组合生成螺旋线两种方式；二、对一些问题提出了设想，以期为后

续研究提供可能的方向。本文对算法改进的贡献是：一、通过使用数学表达式生成费马螺旋线的方式，提高费马螺旋线的数学严谨性。二、使用改进方法生成的螺旋线，其能填充的区域具有新的形态，它有两个突出的“接口”，或许能够更有利于区域间螺旋线的连接。三、后处理阶段用于曲线平滑，会增多欠填充区域；本文对螺旋线生成阶段的曲线平滑方法做出了设想，若能够实现，可取消后处理阶段，减少欠填充区域，提高打印质量。

1.2 研究现状

该部分从三方面论述研究现状，分别是目前常用的路径规划算法、连通费马螺旋线路径生成算法及其他螺旋线的生成算法。

1.2.1 路径规划算法

平面区域填充是数控加工、3D 打印等领域的重要环节。路径规划算法的选择将直接影响工件质量。如图 1- 2 所示，目前常用的路径规划算法主要有：平行线填充路径、轮廓偏置填充路径、螺旋扫描填充路径、分形扫描填充路径、分区扫描填充路径等。本部分将从 3D 打印方面介绍各路径规划算法。

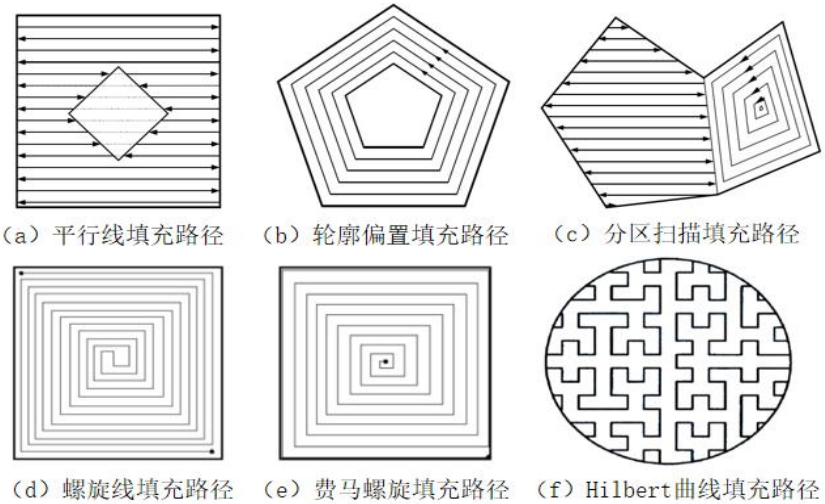

图 1- 2 路径规划算法

平行线扫描填充路径[3]是使用一组等距的平行线填充区域的方法。这种方法对平行线与轮廓进行求交，仅保留轮廓内部的线段作为打印路径。通过打印喷头按一定顺序沿平行线段往复移动，完成填充。这种方法控制简单，并有着成熟可靠的算法实现。能够迅速填充不含内轮廓的区域。但也存在一些问题：在打印存在

空腔的复杂区域时，打印喷头需要在空腔处停止给料，行走一段空行程后，继续打印。这会使打印材料在区域边缘缺失或堆积，降低打印精度；空行程的存在也会降低打印效率。同时，打印方向相同会使材料收缩应力方向相同，容易使产品出现翘曲变形。

轮廓偏置填充路径[4]通过将轮廓线向打印区域内部进行等距偏移形成打印路径。打印路径都在区域内部，会缩短空行程，提高打印效率。由于打印路径的方向在不断变化，会使收缩应力分散，降低产品翘曲变形的可能。但也存在一些问题：对复杂模型求解偏置线时，会出现偏置线自交、偏置线自交等情况，为了避免这种情况，偏置线的求解算法会十分复杂，这会降低处理效率。

螺旋扫描填充路径[5]是使用螺旋线进行区域填充的方法。是本文重点研究的路径规划算法。该方法的路径方向在打印过程中不断发生变化，能够使打印材料收缩应力分散，提高工件平整度。

分形扫描填充路径是使用分形曲线进行区域填充的方法。分形曲线具有局部和整体相似的特点，通过不断迭代，生成填充路径。Hilbert 曲线是一种常用分形曲线，它通过自我复制产生打印路径。如图 1- 2（f）所示，Hilbert 曲线能够产生连续的打印轨迹，这会提高打印效率。Hilbert 曲线的路径方向在不断发生变化，这会提高产品平整度。但是该曲线具有许多急弯，在急弯附近打印喷头会降速，降低打印效率；也会使填充不均匀。

分区扫描填充路径是一种复合的填充方法，他将填充区域分成几部分，根据不同区域的特性，选择不同的填充方法，取长补短。翟晓雅[6]等将填充区域分为轮廓层、过渡层、内层，通过在轮廓层使用轮廓偏置算法、在内层使用连通费马螺旋线生成算法，在保证区域边界的同时，能够提高打印速度。很好地融合了各种路径规划算法的优势。

综上，评价一个路径规划算法的优劣，应该从算法运行效率、打印效率、产品质量等几方面进行考量。优化一个路径规划算法可以从以下几方面进行:减少打印路径中空行程的长度；降低打印喷头的开关次数；减少路径中包含的急弯数，尽量使用长的、低曲率的路径；降低路径生成算法的复杂度等。

1.2.2 连通费马螺旋线路径生成算法

该算法能够构造出全局连续的路径，填充任意的连通 2D 区域。算法的具体流

程如下。

首先，对要填充的 2D 区域进行区域划分。方法是，先以路径宽度的任意倍数为值，对该 2D 区域构造等值线；再以各等值线为节点、某种规则[7]下可能相接的等值线之间的距离为边权，构造有权无向图；之后，对该有权无向图做最小生成树。最小生成树中度小于 2 且有边相连的节点为同一区域。

之后，对区域内部构造螺旋线，对区域之间进行连接。方法是，自底向上遍历最小生成树，对区域内部的等值线进行切断和重新连接，形成费马螺旋线；对区域边界的等值线，选取距离螺旋线起始点最近的点进行连接。此时，已构造出全局连续的打印路径。

最后，进行曲线优化。以优化曲线平滑性及均匀路径间距。但是，在该阶段会引起曲线欠填充的问题。所以，在进行算法改进时，考虑去掉该阶段，以优化打印质量。

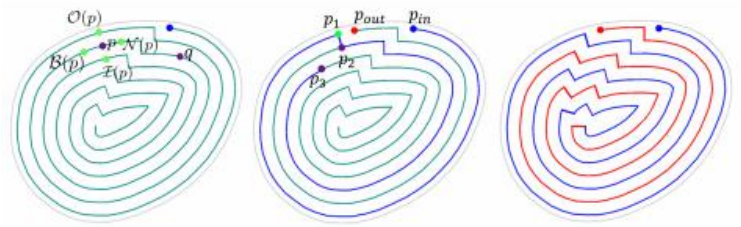

图 1- 3 费马螺旋线生成过程[2]

图 1- 3 展示了在子区域中从螺旋线生成费马螺旋线的过程。对子区域中螺旋线的具体生成方式描述如下。首先定义可螺旋区域，一个可螺旋区域 R，其距离场应该具有单极值。将该区域的打印路径设为π，给出以下符号定义。对任意点$p \in \pi$，如果该点在区域距离场中，存在朝向区域中心的梯度线，将该梯度线与打印路径的交点设为I_p，如果点p接近区域中心，则I_p不存在。同理，将在点p处朝向外侧的梯度线，与打印路径的交点设为O_p，如果点p位于区域边界上，则O_p不存在。I_p和O_p是将等值线转化为费马螺旋的连接点。取间距δ，B_p表示沿打印路径到达p时，在p点前方距离为δ的点；N_p表示沿打印路径到达p时，在p点后方距离为δ的点。令p_{in}为打印路径的起点，p_{out}为打印路径的终点。如图 1- 3，从螺旋线的起点开始逐步构造螺旋线。从起点开始沿打印路径运动到$p_1 = B(p_{out})$，重新

连接路径从 p_1 连接到 $p_2 = I(p_1)$，接着沿打印路径移动到达 $p_3 = B(I(B(p_2)))$，在该点重复上述过程。迭代执行以上过程，移动到区域中心，此时移动方向转为向外。从中心起，沿着未在向内移动时经过的路径移动，重新连接向外的路径，直到到达 p_{out}。以上，完成了零阶费马螺旋线的构造过程。在重新连接过程中生成的“台阶”形，会在后处理阶段进行优化，以使打印路径达到更高阶的连续性。

连通费马螺旋线生成算法能够在任意的连通 2D 区域生成连续的、低曲率、等路径间距的打印路径。但目前的算法也存在一些问题：这种路径规划方式存在欠填充区域，会降低打印质量；连通费马螺旋线缺乏像 Peano 曲线或 Hilbert 曲线一样的规律性和数学严谨性；对连通费马螺旋线的定义是构造性的而不是概念性的。

1.2.3 螺旋线生成算法

目前存在的研究大多是单条螺旋线生成算法。Held M[5]等提出了一种基于单条螺旋的区域填充算法，但该算法存在大量欠填充区域，不利于实际应用。Romero-Carrillo P[8]等基于阿基米德螺旋的线性变形，提出了一种径向等距单螺旋生成算法；该算法通过对内环和外环的线性组合，生成在内环和外环围成的区域内的螺旋线；但是该算法仅对凸区域提供了明确的算法，应用范围存在一定的局限性。汪云海[9]等在可视化领域，提出了能够适应形状的等距螺旋生成算法，用于形成有特定形状的词云；该算法将阿基米德螺旋线的运动方向分解为切向和法向，并基于距离场将其扩展，用于生成其他形状的螺旋线；但是该算法仅能生成与给定区域相似形状的螺旋线，而不是完全符合区域形状，这与路径规划算法的要求有区别。Bieterman M B[10]等基于偏微分方程，提出了在星形区域内的螺旋路径生成算法；该算法通过偏微分方程将边界轮廓向内偏移，之后通过线性插值将这些轮廓连接起来；该算法用于平面铣削工艺，未能将路径完全均匀分布，会存在重复加工或残余加工的区域。Abrahamsen M[11]在多边形区域内通过折线构造螺旋路径；该方法通过计算中轴树获得折线顶点，并通过在尖角插入圆弧的方式平滑生成的螺旋路径；该方法试图在尊重最大间隔距离的同时，最短化螺旋长度，并非等距化螺旋，与本文的研究有差异。林忠威[12]等使用对角曲线连接偏置线来生成螺旋路径；但在区域边缘螺旋线的等距性质变差，不利于实际加工。

也存在一些双螺旋生成算法。曹俊峰[13]等提出了基于双螺旋的混凝土打印算法，该算法通过对轮廓偏置线的连接，生成螺旋线；但算法对含有内轮廓区域的处理较为简单，会降低打印质量。Zhou B[14]等提出了一种光滑连续的双螺旋生成算法，它能够对含有内轮廓的复杂区域处理，并使起始点都位于轮廓边缘；该算法实现高效，但也存在填充不均匀的问题。Steffen[15]等提出了一种基于混合偏置线的双螺旋生成算法，并将两个子螺旋使用 b 样条进行连接；但该算法无法处理具有内轮廓的区域。

以上研究都对本文有所启发。但是，目前对费马螺旋生成算法的研究较少，难以将现有成果直接应用于路径规划算法。所以本课题具有相当难度及开创性。

1.3 论文组织结构

本文共分四章。第一章为绪论，介绍了课题背景及目前的研究现状。第二章详细描述了本文提出的费马螺旋线生成算法的算法流程，展示了算法的运行结果，对比了几种算法的优劣。第三章基于目前的研究成果，对后续的研究提供可能的方向。第四章是对本文的总结。

第 2 章 费马螺旋线生成算法

2.1 概述

本工作采用自底向上的探究流程。首先，思考螺旋线的生成方式，并总结该方式下生成的螺旋线能够填充的区域特点。之后，找到合适的区域划分方式，能将任意的连通 2D 区域，划分成螺旋线能够填充的区域。最后，在各子区域内填充螺旋线，在区域之间进行螺旋线的连接，进而形成全局连续的路径。本章主要对螺旋线的生成方式进行探究。

本章中，首先，基于阿基米德螺旋提出了费马螺旋的数学表示。之后，基于此，尝试扩展螺旋线的表示，以期增多螺旋线能够填充的区域类型。该尝试主要从两方面进行，分别是：基于距离场的螺线生成算法及基于线性组合的螺线生成算法。之后，对几种算法做了对比和总结。

2.2 基于阿基米德螺旋的费马螺旋线表示

阿基米德螺旋[16]是一种被广泛运用的欧几里得螺旋。它的运动轨迹可以描述为：在一端端点固定的直线匀速转动的同时，有一动点从固定端点出发，沿直线做匀速运动[17]。该动点的轨迹即为阿基米德螺旋。极坐标下，阿基米德螺旋可表示为：

$$\mathrm{r}(\theta)=\mathrm{m}\theta+\mathrm{b} \tag{2-1}$$

其中，θ为极角；r 为到原点的径向距离；$\mathrm{b}=\mathrm{r}(0)$为螺线起点到原点的距离；m 控制螺线间距，相邻旋转周期的螺线间距为$2\pi\mathrm{m}$。如图 2- 1（a）（b）所示。

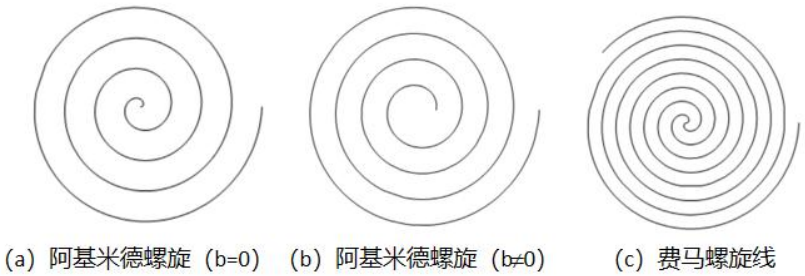

图 2- 1 螺旋线

阿基米德螺旋具有等路径间距的特点，已经在材料设计[19]、医学成像[20]等领

域有了广泛的使用。阿基米德螺旋的这种特点，能够满足路径规划算法对路径连续、平滑、等间距的要求。所以，也可以很好地应用于路径规划算法中。

费马螺旋线是一种特殊的螺旋结构。本文扩展了维基百科中对于费马螺旋的定义[1]，将两个子螺旋在中心点处进行连接的双螺旋结构定义为费马螺旋线。这种螺旋结构具有以下特点：与轮廓偏置填充路径相似，能够适应轮廓边界；螺旋的起始点和终止点可以在边界的任意处，方便了螺旋之间的连接，有利于全局连续路径的生成。这些特点，使得费马螺旋能够应用于路径规划算法中，并已经存在相关研究[2,6]。

本文基于阿基米德螺旋和费马螺旋线的特性，提出了基于阿基米德螺旋的费马螺旋线的数学表示。极坐标方程如下：

$$\mathrm{r}(\theta) = \pm m\theta \tag{2-2}$$

如图 2-1（c）所示，螺旋线由两条运动方向相反的阿基米德螺旋在原点处连接而成。该螺旋线具有恒定的路径间距 $\pi\mathrm{m}$，且在连接点处具有一阶连续性。

2.3 基于距离场的等距单螺旋生成算法

2.3.1 算法描述

汪云海[9]等人研究了阿基米德螺旋轨迹的生成方式，此算法是对该工作的应用。并且，该算法提出了一个新的概念“核”，用以引导螺旋线的生成。

1、阿基米德螺旋

基于上述工作，总结阿基米德螺旋的运动轨迹。螺旋线的生成过程可以看作是，从公式（2-1）中的点 $\mathrm{r}(0)$ 出发，沿螺旋线的运动方向运动，形成的运动轨迹。通过对式（2-1）进行以下操作：将式（2-1）改写成参数方程、在参数方程表示形式下对 θ 进行求导、对求导后的方程进行变形，最终得到阿基米德螺旋的运动方向如下：

$$\begin{pmatrix} \mathrm{dx} \\ dy \end{pmatrix} = md\theta \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} + rd\theta \begin{pmatrix} -\sin\theta \\ \cos\theta \end{pmatrix} \tag{2-3}$$

设 $\mathbf{N} = (\cos\theta, \sin\theta)^T$，$\mathbf{T} = (-\sin\theta, \cos\theta)^T$，则可将螺旋线的运动方向分解为 $\mathbf{N}$ 和 $\mathbf{T}$

方向，如图 2-2 所示。**N** 和 **T** 可以看作是，在点 (x,y) 处，以 r 为半径、原点为圆心所形成的圆的单位法向量和单位切向量。这样的圆也可以看作是，在原点形成的距离场中，距离为 r 的等值线。式（2-3）指明了在点 (x,y) 处时，下一步该运动到点（$x+dx,y+dy$）。这是通过在距离场切线方向运动 $rd\theta$，在距离场法线方向运动 $md\theta$，两运动叠加得到的。按此种方式，绕原点运动一周，外扩的距离是 $2\pi m$。

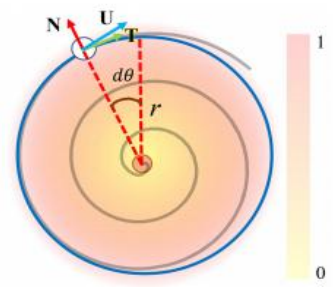

图 2-2 螺旋线的运动方向[9]

2、计算距离场

距离场是一种被广泛应用的形状表示方法，已被应用在边绑定[18]等领域。距离场定义了任意点到形状的最短距离：

$$\varphi(p\in R^2,\Omega)=\min_{q\in\Omega}\|p-q\| \tag{2-4}$$

其中，p 是二维平面上的点；Ω 是形状曲线，在圆形距离场中可以指圆心；q 是形状上的任意点。φ 在形状上值为 0，在向外扩展的过程中逐渐增大。

3、扩展螺旋线的表示

在一点形成的距离场下，能够生成阿基米德螺旋。对该性质进行合理外推，用以形成其他形状的螺旋线。本工作中，提出了一个新的概念“核”，它指的是一段开放的曲线。“核”形成的距离场，能够引导螺旋线运动，形成其他形态的螺旋线。螺线上任意点的运动方向如下：

$$\begin{pmatrix}dx\\dy\end{pmatrix}=md\theta\mathbf{N}+rd\theta\mathbf{T} \tag{2-5}$$

其中，N 是“核”距离场梯度方向，T 为与 N 垂直的方向。N、T 均为单位向量。螺

线从“核”的一端出发，绕该点旋转一周，外扩的距离大约是$2\pi m$，从而实现近似等距。

4、优化

对式（2-5）进行优化。在不同曲率的区域使用相同的步长$d\theta$，会使高曲率区域得不到精确刻画。将式（2-5）中$rd\theta$项改写为$Rd\eta$，用以适应不同曲率的区域。其中，R 是局部曲率半径，随曲率的升高而降低；$d\eta$是用户给定的角速度参数。将式（2-5）改写为：

$$\begin{pmatrix} dx \\ dy \end{pmatrix} = md\theta\mathbf{N} + Rd\eta\mathbf{T} \tag{2-6}$$

式（2-6）表明，螺线运动过程中，在曲率高的地方沿切向方向运动慢，曲率低的地方运动快。所以，该式能够适应不同曲率。

2.3.2 实现细节

算法的输入包括“核”形状、角速度参数$d\eta$、与路径间距相关的 m、迭代次数 num；输出为等距单螺旋线。从“核”的一端开始逐步生成螺旋线，每次迭代螺旋线前进一步。一次迭代的过程如下：在当前点(x,y)处，按式（2-6）进行计算，移动到点$(x+dx,y+dy)$处。如图 2-3 所示，该过程共分两步。首先沿切向方向移动$Rd\eta$，即图 2-3 中由点 A 运动到点 B；再沿法线方向移动$md\theta$，即图 2-3 中由点 B 运动到点 C。在这个过程中，为了防止螺旋线的过度移动，要设置最大切向移动距离。核心算法伪代码见算法 2-1。

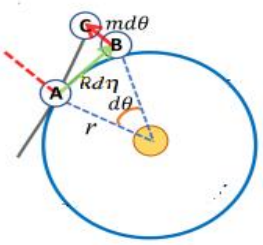

图 2-3 一次迭代的运动过程

算法 2-1 等距单螺旋线生成算法

等距单螺旋线生成算法

```
INPUT：core：核；d_eta：角速度参数；m：路径间距；num：迭代次数；
OUTPUT：spiral：螺旋线
1:   start_point = core 任意端点
2:   pre_point = start_point
3:   for i = 1 to num do
4:       计算在 pre_point 处的单位切向量 T、单位法向量 N、局部曲率 R
5:       now_point = pre_point + R*d_eta*T
6:       d_theta=<pre_point - start_point , now_point - start_point>
7:       now_point = now_point+m*d_theta*N
8:       spiral.add ( now_point )
9:   return spiral.
```

2.3.3 运行结果

在路径宽度为4π、用户角速度为$\frac{\pi}{2}$、迭代次数为 500 次、切向最大运动距离为 5 时，图 2- 4 为运行结果。其中，灰色曲线为输入的“核”，蓝色曲线为生成的螺旋线。

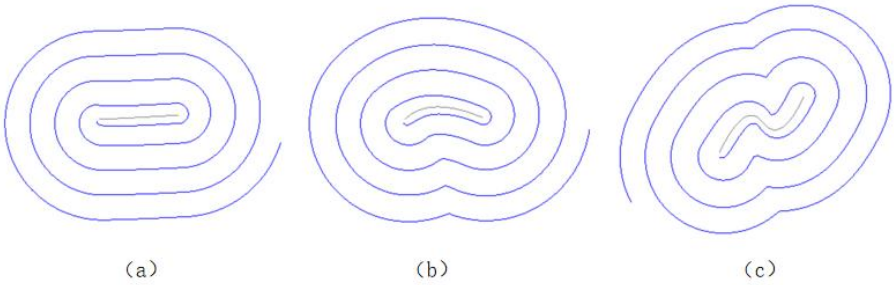

图 2- 4 等距单螺旋算法运行结果

由图 2- 4，螺旋线在外围区域能够实现近似的等距，但在中心区域有过填充现象。

2.3.4 算法评价

基于距离场的螺线生成算法，能够生成近似等距的单螺旋，一定程度上符合该课题的要求。但也存在一些问题。

连续性不足。算法给出了在螺线上任意点，到下一点的运动方向。每次运行该算法，螺线前进一步，通过多次迭代生成完整的螺旋线。所以，该螺旋线是由许多离散的点连接而成的，仅具有零阶连续性。而路径规划算法需要较为平滑的填充曲线。

能够形成的螺线形状较少。该算法仅能形成与“核”距离场相似的螺线形状，无法生成对侧同时凸或同时凹的螺旋线。例如，图 2- 5 所示的图案，就无法由该算法生成。在此后的工作中，考虑扩充“核”的定义：“核”可以是一段闭合的曲线。

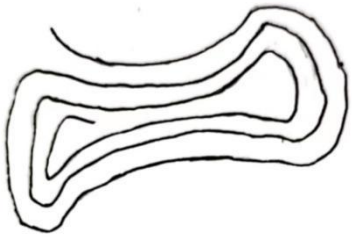

图 2- 5 该算法无法生成的螺旋线形状示意图

“核”与螺旋线能够填充区域的对应关系难以寻找。正如阿基米德螺线无法准确填充正圆形，“核”引导下生成的螺旋线形状也不完全等同于“核”距离场形状。这种生成方式下的螺线形状受距离场局部曲率、用户输入角速度、算法迭代次数、步长等多方面影响。而路径规划算法需要填充曲线能够较好地保持边界特征。

扩展到等距双螺旋困难。算法基于距离场能够生成等路径间距的单螺旋结构。但目前未找到合适的方式，使得相连的两条螺旋间路径等间距。

综上，本文并未将基于距离场的等距单螺旋线生成算法扩展，用以生成费马螺旋线。也未做进一步研究。

2.4 基于线性组合的费马螺旋线生成算法

2.4.1 算法描述

在基于距离场的等距单螺旋生成算法中，认为螺旋线是由起点开始沿运动方向逐渐移动形成的。本节将从另一角度思考螺旋线的生成过程，认为螺旋线是由线性组合生成的。

1、阿基米德螺旋

阿基米德螺旋可以看作是由两个圆线性组合形成的[8]。对式（2-1）进行变形得到：

$$\begin{pmatrix} \mathrm{x}(\theta) \\ y(\theta) \end{pmatrix} = \frac{\theta}{2\pi k}\begin{pmatrix} (2\pi km + b)\cos\theta \\ (2\pi km + b)\sin\theta \end{pmatrix} + (1 - \frac{\theta}{2\pi k})\begin{pmatrix} b\cos\theta \\ b\sin\theta \end{pmatrix}, \theta \in [0, 2\pi k] \qquad (2\text{-}6)$$

其中，k 为螺旋线旋转的圈数；$\begin{pmatrix} (2\pi km + b)\cos\theta \\ (2\pi km + b)\sin\theta \end{pmatrix}$为外圆；$\begin{pmatrix} b\cos\theta \\ b\sin\theta \end{pmatrix}$为内圆。随$\theta$值的逐渐增大，螺旋线的运动从内圆环某点开始，到外圆环某点结束。设$\mathbf{c}(\theta) = \begin{pmatrix} x(\theta) \\ y(\theta) \end{pmatrix}$，$\mathbf{d}_{out}(\theta) = \begin{pmatrix} (2\pi km + b)\cos\theta \\ (2\pi km + b)\sin\theta \end{pmatrix}$，$\mathbf{d}_{\mathrm{in}}(\theta) = \begin{pmatrix} b\cos\theta \\ b\sin\theta \end{pmatrix}$，则式（2-6）可改写为：

$$\mathbf{c}(\theta) = \frac{\theta}{2\pi k}\mathbf{d}_{out}(\theta) + (1 - \frac{\theta}{2\pi k})\mathbf{d}_{in}(\theta), \theta \in [0, 2\pi k] \qquad (2\text{-}7)$$

螺旋线在相邻转弯处的间距为$|\frac{1}{\mathrm{k}}(\mathbf{d}_{out}(\theta) - \mathbf{d}_{in}(\theta))|$。因为在式（2-7）中，$|\mathbf{d}_{out}(\theta) - \mathbf{d}_{in}(\theta)|$恒定，所以阿基米德螺旋线全局等距。如图 2- 6 所示，图中绿线与 x 轴正方向的夹角为θ，与外圆环和内圆环的交点分别为$\mathbf{d}_{out}(\theta)$和$\mathrm{d}_{in}(\theta)$。

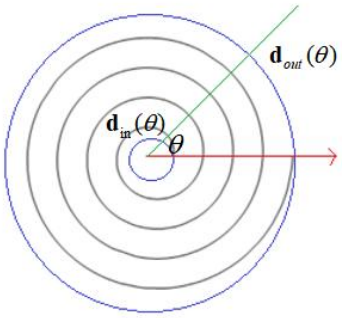

图 2- 6 线性组合生成阿基米德螺旋示意图

2、扩展螺旋线的表示

式（2-7）中，$\mathbf{d}_{in}(\theta)$和$\mathbf{d}_{out}(\theta)$是一对对应关系，两点之间的线段与螺线形成数个交点，在线段上相邻交点间的距离相等，所以阿基米德螺旋线能实现径向等距。考虑将$\mathbf{d}_{out}$和$\mathbf{d}_{in}$表示的封闭曲线类型进行扩展，用以形成其他形态的螺旋线。在外环$\mathbf{d}_{out}$固定时，按$|\mathbf{d}_{\mathrm{out}}(\theta) - \mathbf{d}_{in}(\theta)|$全局恒定的规则寻找内环$\mathbf{d}_{in}$，之后使用式（2-7）形成螺旋线。该方法中，在外环$\mathbf{d}_{out}$中心处任取一点作为原点，由原点向四周引射线。与 x 轴正方向夹角为θ的射线，与$\mathbf{d}_{out}$的交点为$\mathbf{d}_{out}(\theta)$。由$\mathbf{d}_{out}(\theta)$沿

射线方向向原点移动距离 1，所到达的点为$\mathbf{d}_{in}(\theta)$。其中，$\mathbf{d}_{out}$围成的区域为简单凸区域，且曲线$\mathbf{d}_{out}$无自交；距离 1 要小于原点到$\mathbf{d}_{out}$距离的最小值。使用该方法生成的螺旋线，虽然能够满足全局径向等距。但在外环$\mathbf{d}_{out}$长宽比过大时，并不符合认知中等距的定义。且在外环为非凸区域时，生成螺旋线会存在问题。该方法中，中心点的选取也会影响螺旋线的生成效果。所以，考虑改进该方法。

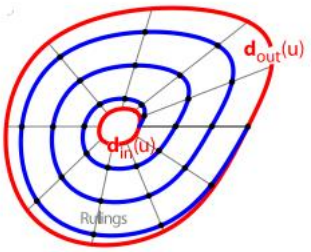

图 2- 7 线性组合生成任意凸的螺旋线

借鉴相关工作[8]，考虑将式（2-7）改写成离散形式。修改$\mathbf{d}_{in}$和$\mathbf{d}_{out}$的对应关系，以使螺旋线更加等距。首先考虑螺线仅旋转一圈的情况。采用离散参数u代替式（2-7）的θ，它的取值范围为 0 到 1。$\mathbf{d}_{in}(u)$和$\mathbf{d}_{out}(u)$在外环和内环上离散选择，是一对对应点，如图 2- 7 所示。此时式（2-7）可改写为$\mathbf{c}(u)=u\mathbf{d}_{out}(u)+(1-u)\mathbf{d}_{in}(u)$。在螺线旋转多圈时，引入变量$j$，表示目前考虑的是第$j$圈。由式（2-7）知，在第$j$圈开始时，$\mathbf{d}_{out}$前的系数应为$\frac{j-1}{k}$，在第$j$圈结束时，$\mathbf{d}_{out}$前的系数应为$\frac{j}{k}$。所以，螺旋在第$j$圈旋转时，$\mathbf{d}_{out}$前的系数应逐渐从$\frac{j-1}{k}$增加到$\frac{j}{k}$，设该系数为$v$，它是关于$u$和$j$的函数，$v$的计算式为$v(u,j)=\frac{j-1}{k}(1-u)+\frac{j}{k}u$。在每一圈中，$u$的取值均为 0 到 1。

综上，式（2-7）的离散形式如下：

$$\begin{aligned}\mathbf{c}_j(u)&=v(u,j)\mathbf{d}_{out}(u)+(1-v(u,j))\mathbf{d}_{in}(u)\\ v(u,j)&=\frac{j-1}{k}(1-u)+\frac{j}{k}u\end{aligned}\tag{2-8}$$

其中，k 是螺旋线旋转的圈数，它与路径间隔相关；j 表示其中的一圈，它是 1 到 k 之间的整数；u 的取值是 0 到 1；$\mathbf{c}_j$表示第 j 圈螺旋线；曲线$\mathbf{d}_{out}$围成的区域是具

有单极值等值线的简单区域；封闭曲线$\mathbf{d}_{\text{in}}$是$\mathbf{d}_{out}$的等值线。$\mathbf{d}_{in}(u)$和$\mathbf{d}_{out}(u)$是一对对应点。在选取对应点时，要满足以下几点：$|\mathbf{d}_{\text{out}}(u)-\mathbf{d}_{in}(u)|$全局近似相等；随 u 的增大，$\mathbf{d}_{in}(u)$和$\mathbf{d}_{out}(u)$的取点单调向逆时针方向移动；$\mathbf{d}_{in}(u)$和$\mathbf{d}_{out}(u)$的连线要全部在$\mathbf{d}_{in}(u)$和$\mathbf{d}_{out}(u)$围成的区域内；对任意不同的 u，$\mathbf{d}_{in}(u)$和$\mathbf{d}_{out}(u)$的连线不能相交。此式也满足螺旋线在相邻转弯处的间距为$|\frac{1}{\text{k}}(\mathbf{d}_{out}(u)-\mathbf{d}_{in}(u))|$。

3、生成另一条子螺旋

式（2-2）能够表示费马螺旋线。由此，式（2-1）的另一条子螺旋可写作：

$$\text{r}(\theta)=-(\text{m}\theta+\text{b}) \tag{2-9}$$

又因为$-\cos\theta=\cos(\theta+\pi)$、$-\sin\theta=\sin(\theta+\pi)$，将式（2-9）写成（2-7）所示的形式为：

$$\mathbf{c}_2(\theta)=\frac{\theta}{2\pi k}\mathbf{d}_{out}(\theta+\pi)+(1-\frac{\theta}{2\pi k})\mathbf{d}_{in}(\theta+\pi),\theta\in[0,2\pi k] \tag{2-10}$$

由此可知，另一条螺旋线的起始点应在已有螺旋线的对向选取。式（2-7）中的$\mathbf{c}(\theta+\pi)$和式（2-10）中的$\mathbf{c}_2(\theta)$会位于同一径向线段上，且两点的连线上无螺旋线经过。此时两点间的距离为$|\frac{1}{2k}(\mathbf{d}_{out}(\theta+\pi)-\mathbf{d}_{in}(\theta+\pi))|$，所以两子螺旋间也能实现径向等距。

扩展到其他形态的螺旋线。另一条螺旋线仍由式（2-8）生成，圆环取点和对应点关系与已有螺旋线相同。但是，另一条螺旋线的取点顺序与已有螺旋线不同，另一条螺旋线应在已有螺旋线的对侧取点。若设已有螺线所对应的外圆环为$d_{out,1}$，另一条螺线对应的外圆环为$d_{out,2}$，则取点的关系为：当$\text{u}>0.5$，$d_{out,1}(u)=d_{out,2}(u-0.5)$；当$\text{u}<0.5$时，$d_{out,1}(u)=d_{out,2}(u+0.5)$。对内圆环的处理类似。在式（2-8）中，若设已有螺旋在第 j 圈为$\mathbf{c}_{1,j}$，另一条螺旋线在第 j 圈为$\mathbf{c}_{2,j}$。则，在$u<0.5$时，$\mathbf{c}_{i,j}(u+0.5)$和$\mathbf{c}_{2,j}(u)$的距离为螺旋线的径向间距，值为$|\frac{1}{2k}(\mathbf{d}_{out}-\mathbf{d}_{in})|$；在$u>0.5$时，$\mathbf{c}_{1,j}(u)$和$\mathbf{c}_{2,j+1}(u-0.5)$的距离为螺旋线的径向间距，

值为$|\frac{1}{2k}(\mathbf{d}_{out}-\mathbf{d}_{in})|$。综上，在$|\mathbf{d}_{\text{out}}(u)-\mathbf{d}_{in}(u)|$全局近似相等时，能够生成近似等距的双螺旋。

4、连接两螺旋线

借鉴相关工作[15]，在螺旋线中心使用 3 次 b 样条连接两螺旋线。该样条曲线具有 G1 阶连续性，共有 7 个控制顶点，如图 2- 8 所示。控制顶点的构造方式如下：$\mathbf{a}_1$、$\mathbf{a}_2$是两螺旋线的起始点；m 是$\mathbf{a}_1$与$\mathbf{a}_2$连线的中点；$\mathbf{a}_1$、$\mathbf{a}_2$分别沿两螺旋线起始点的切线方向移动路径间距的长度得到$\mathbf{t}_1$、$\mathbf{t}_2$；$\mathbf{t}_1+(\mathbf{m}-\mathbf{a}_1)$得到$\mathbf{b}_1$；同理可得$\mathbf{b}_2$。综上，共有$\mathbf{a}_1$、$\mathbf{t}_1$、$\mathbf{b}_1$、m、$\mathbf{t}_2$、$\mathbf{b}_2$及$\mathbf{a}_2$ 7 个控制顶点。以此，构造准均匀 b 样条。

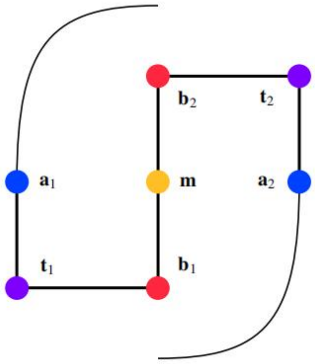

图 2- 8 控制顶点

2.4.2 实现细节

算法输入为外环形状$\mathbf{d}_{out}$，输出为费马螺旋线。算法执行过程描述如下：首先外环向内环做合适距离的偏置，该距离应为两倍路径间距的倍数，得到内环$\mathbf{d}_{in}$；对内环$\mathbf{d}_{in}$和外环$\mathbf{d}_{out}$寻找对应关系，以使$\mathbf{d}_{\text{out}}(u)-\mathbf{d}_{in}(u)$全局近似相等；按式(2-8)生成两条螺旋线，它们起点尽可能在区域的对侧，以保证两螺旋线间距的近似相等；最后，在区域中央使用 b 样条连接两螺旋线。算法核心伪代码见算法 2- 2：

算法 2- 2 费马螺旋线生成算法

费马螺旋线生成算法
INPUT：d_out：外环；m：路径间距； OUTPUT：spiral_1：子螺旋线 1；spiral_2：子螺旋线 2；b_spline：中心曲线 1:　由路径间距 m，计算出内环 d_in

```
2:    寻找内环和外环的对应关系
3:    由式（2-8）形成 spiral_1
4:    起点改为 spiral_1 的对侧，由式（2-8）形成 spiral_2
5:    形成 b_spline
6:    return spiral_1、spiral_2、b_spline.
```

2.4.3 运行结果

在路径宽度为 25，外环为距离场具有单极值的简单凸区域时，运行结果如图 2-9 所示。

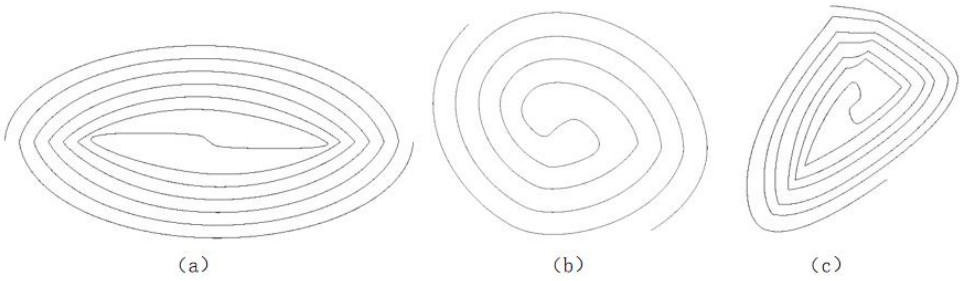

图 2- 9 费马螺旋线算法运行结果

2.4.4 算法评价

基于线性组合的费马螺旋线路径生成算法，能够生成近似等间距的路径。但也存在一些问题。

连续性不足。该算法中，在内环和外环上离散选点的操作，使得螺旋线也是由离散的点组成的，仅具有零阶连续性。

能够填充的区域有限。该算法理论上能够在输入的外环为距离场具有单极值的简单区域时，生成螺旋线。但在目前的代码实现中，寻找对应点的操作存在缺陷，所以仅能对外环距离场具有单极值的简单凸区域生成螺旋线。

未能给出完整有效的寻找对应点算法。螺旋线能否等距严重依赖于对应关系的选取。在后续的研究中，或许可以考虑应用匹配算法[21]。

2.5 总结

2.5.1 算法比较

表 2- 1 对比了三种算法螺旋线生成算法。

表 2- 1 算法对比

名称	连续性	填充区域形态	填充区域是否可以具有内轮廓
原论文方法	0 阶	距离场单极值的区域	可以
基于距离场的等距螺旋线生成算法	0 阶	相似于“核”距离场形态	不可以
基于线性组合的费马螺旋线路径生成算法	0 阶	存在两个“接口”的任意凸区域	不可以

目前的算法都仅具有零阶连续性，在后续的研究中可以考虑使用样条拟合[22]、在尖锐处插入圆弧[23]等方式提高连续性。原论文提供的方法能够填充具有内轮廓的区域，能够填充的区域范围更广。但本文提供了新的螺旋线形态，它具有两个凸出的“接口”，或许有利于区域间的连接。

2.5.2 填充区域特点

设路径间距为 d，则将螺旋线最外圈向外扩展$\frac{d}{2}$，所形成的封闭图形即为该螺旋线能够填充的区域。本章中的路径生成算法所形成的螺旋线能够填充的区域，如图 2- 10 所示。该区域无内轮廓，仅有一圈外轮廓。如图 2- 10（a），可将围成该区域的曲线分为四部分：子螺旋线 1 外扩形成曲线 1、子螺旋线 2 外扩形成曲线 2、连接曲线 1 和曲线 2 的两条曲线分别为曲线 3 和曲线 4。曲线 3 和 4 的特点是：位于区域相对的两侧；长度为 d；由曲线 1 经过曲线 3 到达曲线 2 的过程中，在曲线 3 上的运动方向，与由曲线 1 经过曲线 4 到达曲线 2 的过程中，在曲线 4 上的运动方向相同。所以，如图 2- 10（b）所示区域，并不能由该算法生成的螺旋线填充。

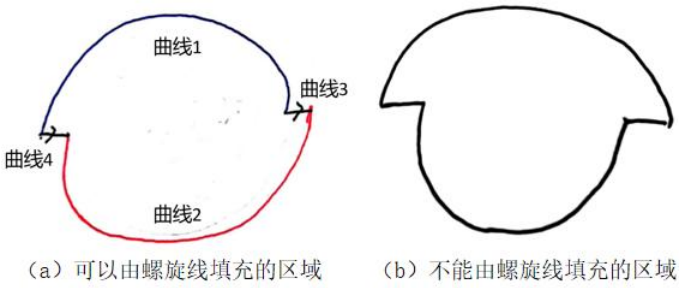

图 2- 10 区域示意图

该区域可以认为有 3 部分，分别是主体部分和两个位于区域对侧“鱼形”区域。

第 3 章 工作展望

本章中，基于上文工作对一些问题提出设想，为后续研究提供可能的方向。具体包括对区域间螺旋线连接方法的设想及对基于偏置操作的螺旋线生成方法的设想。

3.1 基于偏置操作的螺旋线生成方法

在上文研究中，认为螺旋线能够填充的区域如图 3- 1 所示。若在做单个费马螺旋线生成算法时，已知区域边界，对区域边界做偏置操作来形成螺旋线，似乎是更为简单直接的方案。而在路径规划算法中，首先就需要对输入的任意连通 2D 区域作区域划分，将其划分成螺旋线能够填充的区域。此时，单个费马螺旋线的区域边界已知。基于此，算法描述如下。

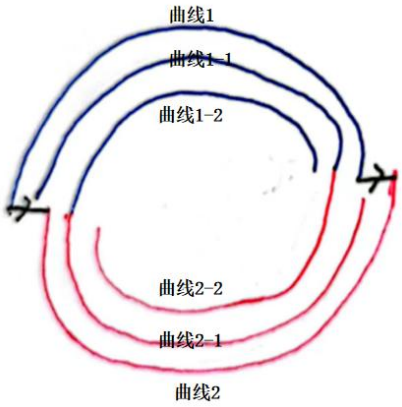

图 3- 1 偏置操作示意图

本文用曲线来表示打印喷头的运动轨迹。假设路径宽度为d，则曲线两侧分别填充宽度为$\frac{d}{2}$的区域。算法首先对图中的曲线 1 和曲线 2 分别向区域内部做距离为$\frac{d}{2}$的偏置线，得到曲线 1_1 和曲线 2_1；之后对曲线 1_1 和曲线 2_1 分别向区域内部做距离为d的偏置线，得到曲线 1_2 和曲线 2_2；此时，曲线 2_2 会与曲线 1_1 相连，曲线 1_2 会与曲线 2_1 相连；之后，不断对得到的新曲线向区域内部做距离为d的偏置线，至区域内部合适位置；最后，将在区域内部的两螺旋线的端点使用合适的平滑曲线连接。

目前已有使用样条曲线实现偏置线的相关研究[24]，或许可以使各段曲线具有平滑性。但在曲线连接点的连续性未知，需要做进一步的研究。

基于偏置操作的螺旋线生成方法，能够在给定区域“接口”不在区域对侧的情况进行填充，如图 3- 2 所示。扩展了螺旋线能够填充的区域类型。

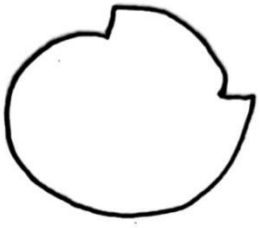

图 3- 2 本方法能够填充的区域示意图

3.2 区域间螺旋线的连接方法

在之前的算法[2]中，要考虑螺旋线之间的连接问题，从而形成全局等距的路径。考虑改进此部分，使各子区域生成的螺旋线能够直接相连。

上文中，螺旋线能够填充的区域具有两个突出的“接口”，可以考虑在各子区域“接口”处，实现螺旋线的连接。若区域划分算法能够像“拼图”一样，实现各子区域的“接口”紧密相接，则生成的各螺旋线或许可以自然连接。基于 3.1 节提出的设想，图 3- 3 展示了一种区域划分的方式，能使螺旋线在区域边界直接相连。

图 3- 3 区域划分示意图

第 4 章 结束语

本文为连通费马螺旋线生成算法提供了可能的改进方向，探究的重点是螺旋线的生成方式。通过对阿基米德螺旋性质的归纳，本文尝试了从两个方面扩展螺旋线能够填充的区域类型，分别是：基于距离场的螺旋线生成算法和基于线性组合的螺旋线生成算法。算法都能生成近似等距的螺旋线，但也存在一些问题。之后，本文对已有的几种螺旋线生成方法做了对比，并基于螺旋线的生成方式，总结了生成方式下的螺旋线能够填充的区域特点。这种螺旋线的生成方式，提供了新的螺旋线形态，或许能够有利于螺旋线的连接。

本文还对之后的工作提供了可能的方向。分别对螺旋线的生成方式和区域间的连接方式提出了设想。在总结了螺旋线能够填充的区域特点后，认为对区域边界做偏置线或许可以更简单有效地生成螺旋线，但由于对偏置线生成算法并不了解，不清楚偏置线生成算法的时间开销，所以本次课题还未进行尝试。基于区域特点，本文还认为可以利用区域“接口”进行螺旋线之间的连接。这些设想都为之后的研究提供了可能的方向。

但是，本文也存在一些问题。如：并未对生成的螺旋线性能定量分析，定量地给出螺旋线欠填充区域和过填充区域与区域总面积的比值。生成的螺旋线仅具有零阶连续性。也未对螺旋线能够填充的区域做出定义。这些问题，期待在后续的研究中得到解决。

参考文献

[1]wikipedia.Fermat’s spiral[EB/OL].(2015-11-28).https://en.wikipedia.org/wiki/%27s_spiral.

[2]Zhao H, Gu F, Huang Q-X, et al. Connected fermat spirals for layered fabrication[J]. ACM Transactions on Graphics,2016,35(4):1-10.

[3]Ding D H, Pan Z X, Cuiuri D, et al. A tool-path generationstrategy for wire and arc additive manufacturing[J]. The Inter-national Journal of Advanced Manufacturing Technology, 2014.73(1-4):173-183

[4]Yang Y, Loh H T, Fuh J Y H, et al. Equidistant path generationfor improving scanning efficiency in layered manufacturing[J]Rapid Prototyping Journal, 2002.8(1): 30-37

[5]Held M,Spielberger C.A smooth spiral tool path for high speed machining of 2D pockets[J].Computer-Aided Design,2009,41(7):539-550.

[6]翟晓雅, 陈发来.分形模型的3D打印路径规划[J].计算机辅助设计与图形学学报, 2018,30(6):13.

[7]孙钊.基于费马螺旋的3D打印路径生成算法研究[D].安徽:中国科学技术大学,2017.

[8]Romero-Carrillo P,Torres-Jimenez E, Dorado R, et al.Analytic construction and analysis of spiral pocketing via linear morphing[J].Computer-Aided Design,2015,69:1-10.

[9]Wang Y, Chu X, Zhang K, et al.ShapeWordle: Tailoring Wordles using Shape-aware Archimedean Spirals[J].IEEE transactions on visualization and computer graphics,2020,26(1):991-1000.

[10]Bieterman M B, Sandstrom D R.A Curvilinear Tool-Path Method for Pocket Machining[J].Journal of Manufacturing Science and Engineering,2003,125(4):149-158.

[11]Abrahamsen M.Spiral tool paths for high-speed machining of 2D pockets with or without islands - ScienceDirect[J].Journal of Computational Design and Engineering,2019,6(1):105-117.

[12]林忠威, 黄常标, 王泽昊.平面轮廓的螺旋填充轨迹生成算法[J].计算机工程与应用, 2015, 51(018):180-185.

[13]曹俊峰.面向3D混凝土打印的双螺旋路径规划算法研究与施工作业系统搭建[D].湖北:华中科技大学,2021.

[14]Zhou B, Zhao J B, Li L, et al.Double spiral tool-path generation and linking method for complex pocket machining[J].Machining Science & Technology,2016,20(2):262-289.

[15]Steffen, Hauth, Lars, et al. Double-spiral tool path in configuration space[J].International Journal of Advanced Manufacturing Technology,2010.

[16]wikipedia.Archimedean spiral[EB/OL].(2023-05-03).https://en.wikipedia.org/wiki/Archimedean_spiral

[17]刘崇军.等距螺旋的原理与计算[J].数学的实践与认识,2018,48(11):10.

[18]Ersoy O, Hurter C, Paulovich F, et al.Skeletonbased edge bundling for graph visualization[J].IEEE Trans. Vis. & Comp.Graphics,2011,17(12):2364-2373.

[19]Rahman N, Afsar M N. A novel modified archimedean polygonal spiral antenna[J]. IEEE Transactions on Antennas and Propagation,2013,61(1):54-61.

[20]Nanni L, Lumini A, Brahnam S.Local binary patterns variants astexture descriptors for medical image analysis[J].Artificial intelligence inmedicine, 2010,49(2):117-125.
[21]杨胜超, 张瑞军.基于二分图最优匹配算法的毕业论文选题系统[J].计算机系统应用,2008,017(007):14-17,34.
[22]张惠敏.欧拉螺线的B样条逼近与数据拟合[D].浙江：浙江大学, 2015.
[23]Chandler P, Rasmussen S, Pachter M. UAV cooperative path planning. 2000.
[24]Tiller W, Hanson E G.Offsets of Two-Dimensional Profiles[J].IEEE Computer Graphics & Applications,1984,4(9):36-46.

致　　谢

在本次毕业论文的写作过程中，我得到了许多帮助和指导，在此表示真挚的谢意。

桃李不言，下自成蹊。首先我要万分感谢我本次论文的指导老师赵海森老师。在本次毕业设计中，老师提供了许多帮助与指导，使得课题能够逐步推进。再次向所有帮助过我的老师表示深深的感谢与崇高的敬意。

岁月虽清浅，时光亦潋滟。同时，也要感谢帮助过我的同学们。感谢他们在我遇到困难时，提供的帮助。在配置环境及代码书写的过程中，同学提供的帮助，总能使我豁然开朗，茅塞顿开。在此，真心的祝愿他们学业有成。

点墨于此，树高千尺不忘根深洪土。感谢我的家人一直在用自己的方式全心全意的爱我、尊重并支持我成长路上的每一个选择。养育之恩，无以回报。

凡是过往，皆为序章，期待和以后更好的你我相遇，星光不问赶路人，时光不负有心人。在追梦道路上，我会追逐自己的本心不断勇往直前！

ShapeWordle: Tailoring Wordles using Shape-aware Archimedean Spirals

Abstract—We present a new technique to enable the creation of shape-bounded Wordles, we call *ShapeWordle*, in which we fit words to form a given shape. T o guide word placement within a shape, we extend the traditional Archimedean s pirals to be shape-aware by formulating the spirals in a differential formusing th e distance fifield of the shape. To handle non-convex shapes, we introduce a mu lti-centric Wordle layout method that segments the shape into parts for our shape -aware spirals to adaptively fill the space and generate word placements. In addit ion, we offer a set of editing interactions to facilitate the creation of semanticall y-meaningful Wordles. Lastly, we present three evaluations: a comprehensive com parison of our results against the state-of-the-art technique (WordArt), case studie s with 14 users, and a gallery to showcase the coverageof our technique.

1 INTRODUCTION

Wordles have proven to be simple but effective in conveying an overview of tex t in an engaging way . As the size of a word represents its frequency (and thus weight), people can easily identify the main theme of the text. In addition,the Wordle algorithm produces an aesthetically pleasing and appealing output by leve raging typefaces and colors. As Wordles have been gaining a tremendoupopularit y, a wide variety of applications have been developed to enable people to create expressive and beautiful wordles .

With the continuous popularity of Wordles on social media, a number of to ols have emerged, such as WordArt and Tagxedo , which allow people to create Wordles in a certain shape by fitting words into it. However, the results do not achieve a high fifill rate and high data fidelity at the same time, asshown by t wo WordArt examples: the word sizes accurately encode the wordfrequencies at t he cost of a loose filling in Fig. 2(a), while Fig. 2(b) achieves atight filling by exaggerating the sizes of some words (e.g., Day and Church).

In addition, we recognize the opportunity of facilitating the creation of sema ntically meaningful Wordles: associating topics into proper parts of a shape ishel pful for presenting multiple topics, and can enhance the communicative power of Wordles . However, existing tools do not provide manipulation capabilities: (i) arrange words of different topics into different parts of a shape, (ii) directly man ipulate individual words, and (iii) fine-tune the Wordle shape.

In this paper, we present *ShapeWordle*, a new technique that allows people to generate shape-bounded Wordles, while preserving data fidelity. The core of o ur technique is a shape-aware Archimedean spiral for guiding the word placemen t within a Wordle. Formulating the Archimedean spiral in a differential form ena bles us to guide it by a distance field, to generate spirals of arbitrary forms, an d thus to fill arbitrary shapes with almost equidistant words. Figs. 2(c,d) show r esults generated by the original Wordle tool WordArt and our ShapeWordle for t he same input text. In contrast to Fig. 2(d), the top and bottom portions in Fig. 2(c) are nearly empty due to the compact spiral placement strategy of Wordle. In addition, the word distribution in Fig. 2(d) is more uniformthan the one in Fi g. 2(a), which was produced using WordArt. Note that for afair comparison we i ntentionally used the same set of words with exactly the same sizes for both ap proaches. To obtain a high fifill rate, even when there are not enough words, Sh apeWordle is able to uniformly scale all words to fill the shape, while preservin g their relative weights (see the example shown in Fig. 2(e)).

Besides, ShapeWordle is able to generate multi-centric layouts by performing a greedy layout strategy to fifill each part of a complex shape with its associat ed words. Meaningful shape parts are obtained by first using automatic image/sh ape segmentation methods and then performing interactive fine tuning.

It is challenging to allow interactive editing while achieving a high filling r ate in a word cloud. To tackle this challenge, we develop a hybrid word represe ntation that allows the editing of important words in a cloud and filling addition al (typically smaller) words in a subsequent step. The division between the two groups can be done by selecting a certain percentage of the most frequent words

(those with the biggest size) or by selecting all words beyond a sizethreshold. Once the words are split and all important words have been arranged,users are a llowed to interactively manipulate each editable word, change the overall shape a nd refine the correspondence between words and shape parts. Dueto this editing functionality, ShapeWordle enables users to create semantically meaningful wordle s such as two examples in Fig. 1, which cannot easily be achieved by existing t ools.

We evaluated our approach by quantitatively measuring the quality of our re sults by computing the layout coverage, layout uniformity and shape similarity. T he results show that our method is capable of producing compact Wordles with shapes highly similar to the given outlines. We also invited 14 people to investi gate ShapeWordle as an authoring tool. Our results demonstrate the advantages o f ShapeWordle over approaches such as WordArt.

In summary our main contributions are:

• We formulate a shape-aware Archimedean spiral to guide and align Wordle lay outs with arbitrarily-given shapes and to facilitate us to create multi-centric Word les, where different words are placed in different parts of the given shape;

• We introduce a set of shape-aware Wordle editing interactions based on the co herent combination of rigid body operations and pixel-based placements; and

• We quantitatively evaluated the quality of the resulting Wordles and conducted case studies with 14 users to illustrate the expressiveness of shapeWordle.

2 RELATED WORK

2.1 Word Cloud Visualization

A complete review of the design space of word clouds is beyond the scope ofth is paper. We refer the readers to Felix et al. Here, we focus our discussionon th e layout problem, i.e., given a set of words with associated weights, create a lay out of words, say with one of the following three options: horizontal,vertical and spatial. The first two options simply arrange words from top to bottom or from left to right in alphabetical order or by their weights. In contrast,the last option does not impose any specific order of positioning the words,

but arranges them subject to various aesthetic and semantic criteria. A classic example is Wordle, which attracts a large number of users in recent years. The core of a Wordle is a greedy algorithm that successively layouts words along a spiral. The original algorithm,however, does not meet many aesthetic and semantic needs of designers,since it does not allow users to manipulate individual words nor pack the words into a target shape. To this end, a variety of advanced layout and editing methods have been proposed in recent years.

Neighborhood graphs. To arrange relevant words next to each other for forming a semantic layout, often a neighborhood graph is constructed with nodes representing words and edges connecting relevant words with weights. Cui et al.implicitly construct such a graph by creating a distance matrix that describes the cosine similarity between words, then place words via a multidimensional scaling of the matrix using a force-directed scheme to reduce the empty space between words. Wu et al. improve this layout by using seam carving to further remove the empty space, while Paulovich et al. [28] extend this algorithm for visualizing the neighborhood relationship between documents and their corresponding word clouds simultaneously. Rather than using an implicit graph, Barth et al. directly incorporate a neighborhood graph into their word clouds. They show that respecting the neighborhood relationship is an NP-hard problem. To overcome the related computational overhead, they present several approximation algorithms and conduct a quantitative comparison using implicit methods,and show that Wordle is the most compact layout. In this paper, our focus is to extend Wordle and shape the Wordle into a target shape,while weakly respecting the neighbor graph.

2.2 Spiral-based Visualization

Spirals are used in many spatial layout processes to compactly arrange objects. Among their many different forms, the Archimedean spiral is a widely-used one for visualization, since it is known to be effective in representing periodicity. Already, Gabaglio employed it to present periodic data. Carlis et al. and Weber et al. independently present the first prototypes of spiral displays, where the color and line thickness are used to encode time series data. Later,Dragicevic and Huot

combine spirals with a clock metaphor and develop a *SpiraClock* system for sh owing upcoming events. By arranging glyphs that encode multiple variables alon g a curve, a spiral can further be employed to reveal multivariatedata, image-bas ed search results, as well as network security data.

On the other hand, Archimedean spirals have been used as the underlying v isual pattern to guide the placement of visual items. Two examples are Wordles and balloon treemaps, where the words and circles are arranged along a spiral, s tarting from the origin using a greedy strategy. These examples, however, follow a conventional circular spiral, hence, they might not effectively fifill an arbitrar y target shape (see Fig. 2(c)). In this work, we generalize the Archimedean spira ls to better adapt the word placement in target shapes. This allows us to optimiz e the generation of Wordles for arbitrary shapes.

3 BACKGROUND

In this section, we fifirst review the Wordle layout algorithm and discuss its two inherent drawbacks that limit its ability to shape a Wordle. After that, we briefl fly describe the Archimedean spiral formulation.

3.1 Wordle Layout Algorithm

Given a list of words and a weight associated with each word, the Wordle algor ithm adjusts the size of each word in proportion to its weight and then represent s the boundary of each word using a splinebased contour. To arrange thewords i n a compact and non-overlapping manner with the more important words closer to the centroid, the algorithm first sorts the words by the weights indescending o rder and then takes the following two steps to place one word at atime:

1. **Initialize:** pick a random position around the center of the canvas (see the or ange dot in Fig. 3(a));

2. **Search-and-update:** create a spiral started from the picked random position,an d search along the spiral for a location to place the next word, such that the ne xt word does not overlap with any already-placed word; then, update theword cl oud with the word placement (see Fig. 3(a) for an illustration of the process).Be ing able to place words in horizontal, vertical or diagonal directions

allows the creation of many variants. While the initial position can be completel y random, the final Wordle might not be very compact.

Drawbacks. Both steps heavily limits the flexibility of Wordle in creating arbitra rily-shaped word clouds. First, the Archimedean spiral always searches for the ne w position in a circular manner, so the generated Wordle cannot effectively com ply with the target shape (see again Fig. 2(c)). Second, picking the initial positio n around a center produces a single-piece Wordle, hindering the creation of mult i-topic word clouds. Overall, these two factors largely attribute to thegeneral blob by shape of most Wordles. In contrast, storytelling word clouds convey semantic s by arranging words into complex, multi-part shapes.There is a gap between the user demands and Wordle functionality.

3.2 Archimedean Spiral

The Archimedean spiral is one of most widely-used Euclidean spirals,which can be readily defifined in polar coordinates:

$$r(\theta) = m\theta + b \ , \qquad (1)$$

where θ is the polar angle, r is the radial distance from the origin,$b = r(0)$ isth e initial distance of the starting point from the origin, and m controls the spacin g between successive turns. Having a uniform spacing ($2m\pi$) between successive turns is an important and useful characteristic of the Archimedean spiral for man y applications in medical imaging, material design, and digital light processing.Su ch characteristic facilitates an efficient (uniform) space filling (see Fig.3), enablin g the creation of compact Wordles.

4 SHAPE-AWARE WORDLE

In this section, we present how we achieve shape-aware Wordles by extending t he Archimedean spiral to be shape-aware, and by supporting the generation of m ulti-centric Wordle layouts.

4.1 Shape-aware Archimedean Spirals

The Archimedean spiral can also be expressed in Cartesian coordinates, x and y, by using trigonometric functions:

$$\binom{x}{y} = r(\theta)\binom{\cos\theta}{\sin\theta} \ . \qquad (2)$$

Taking the derivatives of Eq. (2) with respect to θ yields

$$\begin{cases} \frac{dx}{d\theta} = m\cos\theta - r(\theta)sin\theta \\ \frac{dy}{d\theta} = m\sin\theta + r(\theta)cos\theta \end{cases} .$$

Actually, （$\frac{dx}{d\theta}$, $\frac{dy}{d\theta}$） is the movement direction (denoted as **U**) of the spiral at (*x, y*) in the 2D space (see Fig. 3(b) for an illustration). Here, we can decompose **U** along **N** = （cosθ，sinθ）^{T} and **T** =（−sinθ，cosθ）^{T}:

$$\mathrm{U} = \mathrm{m}\binom{cos\theta}{sin\theta} + r(\theta)\binom{-sin\theta}{cos\theta} = mN + r(\theta)T \ , \qquad (3)$$

where **N** and **T** are the unit normal vector and unit tangent vector, resp., at poi nt (*x, y*) on a circle of radius $\sqrt{x^2+y^2}$ co-centered with the spiral (see the blue circle in Fig. 3(b)). Such a circle can be interpreted as the isoline of an underl ying distance field $\phi(x, y) = \sqrt{x^2+y^2}$, which measures the Euclidean distance fr om (*x, y*) to the origin.

According to Eq. (3), the movement direction **U** of the Archimedean spiral is go verned by **N** and **T**. To extend the spiral to be shape-aware with a *roughly cons tant spacing* between successive turns, we first compute the distance field (denot ed by ϕ) associated with the input shape (see Fig. 4(a)). Then, we align**N** and **T** with the isolines of ϕ, instead of the isolines of the generic circular distance fif ield φ shown in Fig. 3(b). In this way, when we construct the spiral from a star ting point inside a shape, the spiral can move in a way that follows the shape a nd eventually meets the shape contour (see Fig. 4(d)).

Computing the distance fifield. A distance fifield is an effective shape represent ation that has been used for edge bundling and trail data visualization. It isa sca lar field that specififies the shortest distance to a shape contour specified by a d istance transform $\mathrm{R}^2 \rightarrow \mathrm{R}+$:

$$\varphi(\mathrm{p} \in \mathrm{R}^2, \Omega) = \min_{\mathrm{q}\in\Omega} ||\mathrm{p} - \mathrm{q}|| \ ,$$

where **p** is a point in 2D space, **Ω** is the shape contour, and **q** is any point

on **Ω**. Note that φ is zero on the shape contour and gradually increases towards the center or the medial axis of the shape, and we compute the distance fifield using a linear algorithm [8] with time complexity $O(n)$, where n is the number of points in 2D space.

Extending the Archimedean spiral. To extend the Archimedean spiral to be sha pe-aware, the main question is how to guide the movement of the spiral, orhow to define the movement direction of the spiral at any point **p** in the given shap e. Rather than explicitly constructing the isoline and then computing the isoline normal at **p**, we take the gradient of the distance field as **N**. This strategy can accurately approximate the normal for continuous scalar fields, like φ.Once **N** is available, **T** can be easily obtained because it is a unit vector that is orthogona l to **N**. Then, we can re-write Eq. (3) in a differential form:

$$\binom{dx}{dy} = md\theta N + rd\theta T \ , \tag{4}$$

where r=$\sqrt{x^2 + y^2}$. However, using the same θ at every point in an arbitrary sha pe might not be proper, since the generated spiral might not be able to adapt to regions of high-curvature, e.g., near the corners of the Christmas tree in Fig. 2.

To characterize such sharp features, we consider the local curvature along th e spiral and approximate the curve by small tangential movements (denoted by ds) perpendicular to **N** by $Rd\eta$ and also by $rd\theta$ (see Fig. 4(b) and (c)), where R is the local curvature radius and η is a user-specifified parameter for the angular speed. Doing so, we can write

$$\mathrm{d}\theta = \frac{\mathrm{R}\mathrm{d}\eta}{\mathrm{r}} \ . \tag{5}$$

Note that r in Eq. (4) can also be regarded as the local curvature radius defined on the isolines (concentric circles) of the generic distance field ϕ (see Fig. 4 (c)), while R is the local curvature radius defined on the isolines of the shape-a ware distance field φ. To estimate R at an arbitrary point in φ, we employ the

Hessian matrix; see the supplemental material for details. Putting everythingtogeth er, we can rewrite Eq. (4) as

$$\binom{dx}{dy} = \frac{mRd\eta}{r}N + Rd\eta T \quad . \tag{6}$$

Note that R and r are different at different points, while m and $d\eta$ are constant parameters specified by the user.

In our implementation, we compute dx and dy using Eq. (6) to progressivel y trace out the spiral from the origin. Typically, we set m = 1 and $d\eta$ = π/5, w hile m and R are measured in pixel units on the shape image. Since m isa const ant, the generated spirals can have nearly-uniform spacing ($2m\pi$) between success ive turns. Fig. 5 shows some examples. In the case of a straight curve segment R might become too large, so we empirically set an upper bound of 1.5 pixel u nits as a maximum movement distance along the tangential direction to avoid ex cessive movements. We ran our method on a quad-core PC with a23” LCD wid escreen, an Intel(R) Core(TM) i7-6700K CPU and 16GB RAM. For a wordle of 60 words, our method finishes in less than 5s.

Uniform scaling. Putting our shape-aware Archimedean spiral into the Wordle la yout algorithm enables us to efficiently generate shapeaware Wordles. Fig. 2(d) s hows an example result. However, if the total word areas is far less than the tot al shape area, it is hard to create acompact layout. Hence, we defifine auniform scaling parameter for the words in an attempt to maximize the fill rate, where t he relative weights between words are still preserved. Fig. 2(e) shows the result after we scale up the important words in Fig. 2(d) and regenerate the Wordle.

ShapeWordle：使用具有形状感知能力的阿基米德螺旋来定制词云

摘要

我们提出了一种新的技术来支持创建有形状边界的词云，称之为“ShapeWordle”。在这种技术中，我们将单词摆成给定的形状。为了指导单词在形状中的位置，使其通过利用形状的距离场以微分形式构造螺旋来感知形状。为了处理非凸的形状，我们引入了一种多中心的词云布局方法，将形状分割成若干部分，以便我们的形状感知螺旋线能够自适应地填充空间并生成词的放置位置。此外，我们还提供了一套编辑互动，以促进语义上有意义的词云的创建。最后，我们提出了三个评估结果：我们的结果与最先进的技术（WordArt）的综合比较，对 14 个用户的案例研究，以及一个展示我们技术覆盖面的画廊。

1 引言

事实证明，词云能够以吸引人的方式简单有效地传达文本主旨。由于一个词的大小代表了它的出现频率（也就是权重），人们可以很容易地识别文本的主题。此外，词云算法通过多变的字体和颜色，产生了具有美感和吸引力的输出。由于词云已经获得了巨大的人气，人们开发了各种各样的应用程序，以使人们能够创造出富有表现力的、漂亮的词云。

随着词云在社交媒体上的不断流行,出现了许多工具,如WordArt和Tagxedo ，它们允许人们通过放置单词来形成特定形状的词云。然而，结果并没有同时达到高填充率和高数据保真度，如两个 WordArt 的例子所示：在图 2(a)中，单词的大小准确地编码了单词的频率,而图 2(b)通过夸大一些单词的大小(如 Day 和 Church）的大小来实现紧密的填充。此外，我们认识到有创建有语义意义词云的机会：将主题关联到一个形状的适当部分，有助于呈现多个主题，并可以增强词云的表达能力。然而，现有的工具并不提供操作功能： (i)将不同主题的单词排列到一个形状的不同部分，（ii）直接操作单个单词，以及（iii）微调词云形状。

在本文中，我们提出了“ShapeWordle”，一种能够允许人们生成有形状边界的词云，同时保持数据可靠性的新技术。我们的技术的核心是一个形状感知的阿基米德螺旋，它引导单词在词云中的放置。用微分形式表示阿基米德螺旋，使我们能够通过一个距离场来引导它，生成任意形式的螺旋，从而用几乎等距的单词

填充任意形状。图 2(c,d)显示了由原始的词云工具 WordArt 和我们的 ShapeWordle 为相同的输入文本生成的结果。与图 2（d）相反，由于 Wordle 的紧凑螺旋放置策略，图 2(c)中的顶部和底部几乎是空的。另外，图 2(d)中的单词分布比使用 WordArt 生成的图 2(a)中的单词分布更均匀。请注意，为了进行公平的比较，我们有意为两种方法使用完全相同大小的单词集。为了获得较高的填充率，即使没有足够的单词，ShapeWordle 也能够均匀地缩放所有单词来填充形状，同时保持它们的相对权重（见图 2(e)所示的示例）。

此外，ShapeWordle 能够通过贪心策略生成多中心布局，将复杂形状的每个部分填充与其相关的单词。首先使用自动图像/形状分割方法，然后进行交互式微调，得到有意义的形状部分。

在词云中实现高填充率的同时，允许交互式编辑是具有挑战性的。为了解决这个挑战，我们开发了一个混合单词表示法，允许编辑云中的重要单词，并在后续步骤中填充额外的（通常是较小的）单词。这两组之间的划分可以通过选择一定百分比的最常见的单词（那些大小最大的单词）或通过选择所有超过大小阈值的单词来完成。一旦单词被分割了，所有重要的单词都被安排好了，用户就可以交互式地操作每个可编辑的单词，改变整体形状，并细化单词和形状部分之间的对应关系。由于这种编辑功能，ShapeWordle 允许用户创建具有语义有意义的词，如图 1 中的两个示例，这是现有工具不容易实现的。

我们从计算布局覆盖率、布局均匀性和形状相似性方面进行定量测量结果质量来评估我们的方法。结果表明，我们的方法能够产生与给定轮廓形状高度相似的紧凑文字。我们还邀请了 14 个人来调查将 ShapeWordle 作为一种创作工具。我们的结果证明了 ShapeWordle 比 WordArt 等方法有优势。

总之，我们的主要贡献是：

- 我们制定了一个形状感知的阿基米德螺旋来指导和对齐任意给定形状的文字布局，并方便我们创建多中心的文字，其中不同的单词被放置在给定形状的不同部分；
- 我们介绍了一套基于刚体操作和像素放置的形状感知文字编辑交互；和
- 我们定量评估了结果文字的质量，并对 14 个用户进行了案例研究，以说明 ShapeWordle 的表达性。

2 相关工作

2.1 词云可视化

对词云的设计空间的完整综述超出了本文的研究范围。我们推荐读者参考 Felix 等人。这里，我们的讨论重点是布局问题，即，给定一组具有相关权重的单词，创建一个单词的布局，比如使用以下三个选项之一：水平、垂直和空间。前两个选项只是按字母顺序或按权重，从上到下或从左到右排列单词。相比之下，最后一个选项并没有规定任何特定的单词定位顺序，而是根据各种美学和语义标准对它们进行排列。一个经典的例子是 Wordle，它近年来吸引了大量的用户。Wordle 的核心是一个贪婪的算法，它沿着螺旋依次排列单词。然而，最初的算法并不能满足设计师的许多美学和语义需求，因为它不允许用户操纵单个单词，也不能将单词打包成目标形状。为此，近年来人们提出了多种先进的布局和编辑方法。

邻域图。为了将相关词相邻排列，形成语义布局，通常用节点表示词构造邻域图，并用带有权值的边连接起来。Cui 等人通过创建一个描述单词之间余弦相似度的距离矩阵来隐式地构造这样的图，然后通过对矩阵进行多维尺度来放置单词并使用力导向布局算法以减少单词之间的空白空间。Wu 等人通过使用接缝雕刻来进一步去除空隙，从而改善了这种布局。而波洛维奇等人扩展了该算法，以同时可视化文档与其对应的词云之间的邻域关系。Barth 等人没有使用隐式图，而是直接将邻域图合并到他们的词云中。他们表明，尊重邻域关系是一个 np 困难的问题。为了克服相关的计算开销，他们提出了几种近似算法，并使用隐式方法进行了定量比较，并显示出 Wordle 是最紧凑的布局。在本文中，我们的重点是将 Wordle 扩展并将其塑造成目标形状，同时弱尊重邻域图。

2.2 基于螺旋的可视化

螺旋在许多空间布局过程中用于紧凑地排列对象。在其许多不同形式的中，阿基米德螺旋是一个广泛用于可视化的螺旋，因为它被认为在表示表示周期性方面是有效的。Gabaglio 已经使用它来提供周期性数据。Carlis 等人和 Weber 等人独立提出了螺旋显示器的第一个原型，其中颜色和粗线用于编码时间序列数据。后来，Dragicevic 和 Huot 将螺旋与时钟隐喻结合起来，并开发了一个螺旋时钟系统来显示即将到来的事件。通过沿着曲线排列编码多个变量的符号，可以进一步使用螺旋来揭示多元数据、基于图像的搜索结果，以及网络安全数据。

另一方面，阿基米德螺旋已经被用作潜在的视觉模式来指导视觉项目的放置。两个例子是 Wordles 和气球树图，其中单词和圆沿着螺旋排列，使用贪婪策略从原点开始。然而，这些例子都遵循了传统的圆形螺旋结构，因此，它们可能不能有效地填充任意的目标形状。在这项工作中，我们推广了阿基米德螺旋，以使单词放置更好地适应目标形状。这允许我们优化任意形状的词云的生成。

3 背景

在本节中，我们首先回顾 Wordle 布局算法并讨论它限制形成形状能力的两个固有缺点。然后，我们简要地描述阿基米德螺旋公式。

3.1 Wordle 布局算法

给定一个单词列表和与每个单词相应的权重，Wordle 算法根据权重调整每个单词的大小，然后使用基于样条的轮廓表示每个单词的边界。为了将单词紧密且不重叠排列，并且将重要的单词靠近中心放置。算法首先按权重降序排序，然后采取以下两步每次放置一个单词：

1. **初始化**：在画布中心周围随机选择一个位置（参见图 3(a)中的橙色点）
2. **搜索和更新**：创建一个从随机位置开始的螺旋，沿着螺旋搜索下一个单词，这样下一个单词就不会与任何已经放置的单词重叠；然后，用单词的位置更新单词云（参见图 3(a)）。

能够将单词放置在水平、垂直或对角线方向上，允许创建许多变体。虽然初始位置是完全随机的，但最终形成的词云可能不是很紧凑。

缺点。这两个步骤都严重限制了 Wordle 在创建任意形状的词云方面的灵活性。首先，阿基米德螺旋总是以圆形的方式搜索新的位置，因此生成的 Wordle 不能有效地符合目标形状（再见图 2(c)）。其次，在中心周围选择初始位置会产生一个单块文字，阻碍了多主题词云的创建。总的来说，这两个因素在很大程度上导致大多数词云的斑点形状。相比之下，讲故事的词云通过将单词排列成复杂的、多部分的形状来传达语义。这是用户需求和 Wordle 功能之间存在的差距。

3.2 阿基米德螺旋

阿基米德螺旋是使用最广泛的欧几里得螺旋之一，它可以很容易地在极坐标中被定义：

$$r(\theta) = m\theta + b \ , \tag{1}$$

其中 θ 是极角，r 是到原点的径向距离，b = r (0)是起点到原点的初始距离，

和 m 控制连续转弯之间的间距，在连续转弯之间具有均匀的间距（2mπ）是阿基米德螺旋在许多应用中的一个重要和有用的特性，如医学成像，材料设计，和数字光处理领域。这种特性便于高效（均匀）空间填充，从而创建紧凑的文字。

4 形状感知词云

在本节中，我们将介绍如何通过扩展阿基米德螺旋到形状感知，并支持多中心文字布局的生成来实现形状感知的词云。

4.1 有形状感知的阿基米德螺旋

阿基米德螺旋也可以用笛卡尔坐标来表示，x 和 y，用三角函数：

$$\binom{x}{y} = r(\theta)\binom{\cos\theta}{\sin\theta} \quad 。 \tag{2}$$

关于 θ，取等式(2)的导数

$$\begin{cases} \frac{dx}{d\theta} = m\cos\theta - r(\theta)sin\theta \\ \frac{dy}{d\theta} = m\sin\theta + r(\theta)cos\theta \end{cases} 。$$

实际上，（$\frac{dx}{d\theta}$，$\frac{dy}{d\theta}$）是二维空间中螺旋在（x，y）处的运动方向（记为 U）。在这里，我们可以沿着 N =（cos θ，sin θ）^{T}和 T=（−sin θ，cos θ）^{T}分解 U：

$$\mathrm{U} = \mathrm{m}\binom{cos\theta}{sin\theta} + r(\theta)\binom{-sin\theta}{cos\theta} = mN + r(\theta)T \ , \tag{3}$$

其中 N 和 T 是在点（x，y）处单位法向量和单位切向量，点（x，y）是在半径为$\sqrt{x^2+y^2}$、与螺旋线共中心的圆上。这样的圆可以解释为一个潜在的距离场的等值线$\phi(x,\ y) = \sqrt{x^2+y^2}$，测量从（x,y）到原点的欧氏距离。根据等式(3)，阿基米德螺旋的运动方向 U 由 N 和 T 控制。为了将螺旋扩展到在连续转弯间具有大致恒定间距的形状感知螺旋，我们首先计算与输入形状相关的距离场（用 ϕ 表示）。然后，我们将 N 和 T 用 ϕ 的等值线进行校正，而不是使用图 3(b)中所示的一般环状距离场ϕ的等值线。这样，当我们从形状内部的起点构造螺旋时，螺旋可以跟随形状移动并最终满足形状轮廓。

计算距离场。距离场是一种有效的形状表示，已用于边缘捆绑和轨迹数据可视化。它是一个标量域，它指定到形状轮廓的最短距离。

$$\Phi(\mathrm{p} \in \mathrm{R}^2, \Omega) = \min_{\mathrm{q}\in\Omega} ||\mathrm{p} - \mathrm{q}|| \ ,$$

其中 p 是二维空间中的一个点，Ω是形状轮廓，q 是Ω上的任意点。需要注意的是，Φ在形状轮廓上为零，并逐渐向形状的中心或中轴增加，我们使用时间复杂度为 O (n)的线性算法[8]计算距离场，其中 n 是二维空间中的点数。

扩展阿基米德螺旋。为了将阿基米德螺旋扩展到形状感知，主要的问题是如何引导螺旋的运动，或者如何在给定形状中的任何一点 p 上定义螺旋的运动方向。我们不是明确地构造等值线，然后计算 p 处的等值线法线，而是取距离场的梯度作为 N。这种策略可以精确地近似于连续标量场的法向，如Φ。一旦 N 可用，T 就很容易得到，因为它是一个与 N 正交的单位向量。然后，我们可以重写等式(3)以微分形式表示：

$$\binom{dx}{dy} = md\theta N + rd\theta T \ , \qquad (4)$$

其中，r=$\sqrt{x^2+y^2}$。然而，在任意形状的每一个点上使用相同的θ可能并不合适，因为生成的螺旋可能无法适应高曲率的区域，例如，在图 2 中的圣诞树的角落附近。

为了描述这些尖锐的特征，我们考虑螺旋线的局部曲率，并用小的切向运动来近似曲线（用 ds 表示），它垂直于 N 可以用 Rdη 和 rdθ *表示，*其中，R 是局部曲率半径，η是用户指定的角速度参数。这样做，我们就可以写作：

$$\mathrm{d}\theta = \frac{\mathrm{R}\mathrm{d}\eta}{\mathrm{r}} \ 。 \qquad (5)$$

请注意，在等式(4)中的 r 也可以看作是定义在一般距离场ϕ的等值线（同心圆）上的局部曲率半径，而 R 是定义在形状感知距离场Φ的等值线上的局部曲率半径。为了估计Φ中任意点上的 R，我们使用了黑塞矩阵；详见补充材料。把一切都放在一起，我们可以重写等式(4)：

$$\binom{dx}{dy} = \frac{mRd\eta}{r}N + Rd\eta T \ 。 \qquad (6)$$

请注意，R 和 r 在不同的点上是不同的，而 m 和 dη是由用户指定的常量参数。

在我们的实现中，我们使用等式(6)计算 dx 和 dy，从原点逐步追踪出螺旋线的轨迹。通常，我们设置 m = 1 和 dη = π/5，而 m 和 R 在形状图像上以像素单位进行测量。由于 m 是一个常数，生成的螺旋在连续匝之间有几乎均匀的间距（2mπ）。图 5 显示了一些例子。在曲线是直线的情况下，R 可能变得太大，所以我们根据经验设置 1.5 像素单位的上界作为沿切向方向的最大运动距离，以避免过度

移动。我们在一台拥有 23 英寸液晶宽屏、英特尔(R)核心（TM）i7-6700K CPU 和 16GB 内存的四核 PC 上运行了我们的方法。对于 60 个单词，我们的方法在不到 5 秒内完成。

统一缩放。将我们的形状感知的阿基米德螺旋放到文字布局算法中，使我们能够有效地生成形状感知的文字。图 2.(d)为一个示例结果。但是，如果总单词面积远小于总形状面积，形成紧凑布局是困难的。因此，我们为单词定义了一个统一的缩放参数，试图最大化填充率，其中单词之间的相对权重仍然保留。图 2(e)显示了我们放大图 2(d)中的重要单词并重新生成 Wordle 后的结果。