

自由焦点的形状约束词云快速生成方法

王一凡, 吴向阳*, 王吉康, 金鑫

(杭州电子科技大学计算机学院 杭州 310018)
(wuxy@hdu.edu.cn)

摘要: 针对传统形状约束词云生成方法摆词速度慢、摆词方向和摆词中心固定的问题, 提出一种基于热扩散方程和旋转对称场插值的形状约束词云快速生成方法. 首先, 通过计算三角网格上的热扩散方程快速地获得各个位置处的距离场, 并以此生成摆词螺旋线快速产生形状约束词云; 然后, 通过改变热扩散中源点的位置和温度调整词云摆放的焦点位置, 生成不同摆词中心下的布局效果; 最后, 对用户手工交互的约束方向利用旋转对称场插值获得各位置处的摆词方向向量, 生成契合形状特征的布局效果. 实验表明该方法能快速地生成不同布局焦点下的形状特征保持的词云效果.

关键词: 词云; 距离场; 热扩散方程; 旋转对称场

中图法分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2022.19433

A Fast Generation Method for Free-Focus Shape-Aware Wordle

Wang Yifan, Wu Xiangyang*, Wang Jikang, and Jin Xin

(School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018)

Abstract: The existing methods have low efficiency in generating shape-aware wordle and only use specific directions and fixed center when placing words. To address this problem, a fast generation method for shape-aware wordle based on heat diffusion equation and rotationally symmetric field is proposed. First, the distance field at each location is quickly obtained by simulating the heat diffusion process on the triangular mesh, which is then used to guide the rapid growth of Archimedean spirals to place words within a shape. Second, the focus of the word cloud is adjusted by changing the position and temperature of the source in the heat diffusion process, generating word cloud layout under different word placement centers. Finally, the rotationally symmetric field is employed to interpolate the direction constraint from users' interaction to obtain the word placement direction at each pixel, which helps the wordle layout to match the shape feature. The experiment shows that the method can quickly generate feature-aware wordle under different focuses.

Key words: wordle; distance field; heat diffusion; rotational symmetry field

词云^[1]是一种将一段文本转换为单词云, 并根据单词在文档中的重要性确定其字体大小的可视化工具, 它们以概述的方式传递文本中的关键信息. 词云因具有美观性和实用性, 得以大量应用于

报纸、教育、演示和视觉分析等多个方面^[2-4], 成为用户理解文档内容的有效手段.

近年来, 形状约束词云逐渐获到关注, 其特点是文本数据基于形状摆放, 因而它能充分地利用

收稿日期: 2021-10-19; 修回日期: 2021-12-02. 基金项目: 国家自然科学基金(61972122); 浙江省自然科学基金重点项目(LZ22F020015). 王一凡(2003—), 男, 在校学生, 主要研究方向为数据可视化; 吴向阳(1975—), 男, 博士, 教授, 硕士生导师, CCF会员, 论文通信作者, 主要研究方向为数据可视化、三维场景理解; 王吉康(1994—), 男, 硕士, 工程师, 主要研究方向为数据可视化; 金鑫(1998—), 男, 硕士研究生, 主要研究方向为数据可视化.

显示空间,并将文本的主题关联到相应的形状,从而有利于呈现多个话题和增强文字的交流能力,如将介绍人体知识的词云分别按照其内容放置在相应人体部位的轮廓内,可以增强用户对词云内容的理解.在形状词云中,鉴于 ShapeWordle 方法^[5]具有高填充率和字体大小能较真实地反映单词重要性的特点,其可获得较优的词云布局效果,但它基于距离场生成阿基米德螺旋线指导单词摆放,导致摆词速度较慢,无法达到交互的词云布局性能.同时,它的布局中心由于固定于图形轮廓中心而缺少布局的灵活性,但用户的关注点并非总是在轮廓中心,因此让用户能根据需要交互地选择布局焦点具有重要意义.此外,它只以水平和垂直方向摆放单词,而将单词以契合轮廓形状的方向摆放会增加布局的美观性.

针对形状词云布局的上述问题,本文提出了一种基于三角网格上热扩散方法和对称旋转场插值的形状约束词云布局方法.通过三角化图形轮廓区域和计算三角网格顶点处的热扩散方程,快速获得区域中各点处的距离场值,并交互地生成形状约束词云效果.通过调节热扩散过程中的源点位置和源点温度,实现了交互地调整词云布局焦点.利用图像处理技术或手工交互提取图像中的轮廓和线条特征,并通过对称旋转场插值获得与提取的特征一致的摆词方向,增加了词云的美观度和布局的紧凑性.

1 相关工作

词云技术中的一个核心问题是词云的布局方式,不同的布局方式具有不同的美观度,传达不同的信息.包琛等^[6]较详细地总结了词云的布局方式等技术.较简单的布局方式是将单词按阅读习惯从上到下、从左到右排列,而其中的单词顺序按其权重或字母表次序排序,如 SparkClouds^[7]和 parallel tag clouds^[8].圆形布局是另一种重要的布局方式,如 Bielenberg^[9]将重要词置于圆形中心,按重要性将词由内向外布局;Stefaner^[10]基于主元分析和典型关联分析方法将词布置在一个近似二维圆形空间.而在 Wordle^[11]中,螺旋线布局方式占据重要位置,ManiWordle^[12]和 Rolled-out Wordle^[13]从中心位置出发,沿着圆形阿基米德螺旋逐个放置单词;

而 Seyfert 等^[14]提出的 dynamic word clouds 以方形螺旋线查找下一个摆词位置.Wang 等^[5]提出一种能适应任意形状的改进螺旋生成方法,生成的布局能契合任意形状.力引导布局也较广泛地应用于词云布局中,其布局结果具有紧凑且无重叠的特点.Wang 等^[4]在力引导中利用单词间斥力和单词到中心的引力保证词云的合理布局.Shu 等^[15]基于 D3.js 库实现了一个自底向上的层次力引导布局图,在连续帧间保持稳定性的同时,带来了一定的随机性.力引导中常利用单词间的语义相关性定义它们之间的力,因而较适合应用于语义词云布局.Cui 等^[16]分析了词云之间的关系,并利用自适应的力引导布局保持上下文间的语义关系.Wang 等^[17]使用力引导优化用户评价中的语义关系布局,以保证评价中关键词之间的语义关系.此外,将词云以河流图的形式进行布置,并结合交互和动画辅助可以展示文本主题随时间的演变过程,如 TIARA 系统^[18].而 Knittel 等^[19]使用金字塔布局揭示文本内容的时变过程,其将文本标签的纵轴高度映射为时间范围,用不同标签横轴位置之间的距离表示它们的语义相近性.

形状约束词云将词云限制在特定形状内,以达到提高美观度和传递特定信息的作用,如 Tagxedo^①和 WordArt^②允许用户将词云放入特定的形状内,但由于对个别单词进行了放大和丢弃,因此无法保证数据的真实性.Chi 等^[2]利用包含单词不重叠、单词不超过边界、分布均匀等约束条件的刚体动力学系统,将单词布局限制在目标形状内.Wang 等^[5]通过修正摆词螺旋线的切向量和法向量,使生成词云布局能适应任意形状.Maharik 等^[20]提出了一种数字显微照片生成技术,利用文字生成图像中的各个区域,形成文字马赛克效果.利用地理形状约束词云布局能使词云包含特定的地理含义,如 Geo word clouds^[21]依据地理位置信息放置单词,生成形如地图中的地区形状的地理词云.Predominance tag maps^[22]在地图上展示具有地理特征的数据,以显示各局部地理区域中最显著的数据类别.Reckziegel 等^[23]以词云展示了具有地理信息的话题的兴衰过程.Afzal 等^[24]给出了一种自动排版图生成方法,通过文字形成地图特征的词云,并在其上展示交通密度、犯罪率和人口数据等信息.本文采用 Wang 等^[5]的摆词方式生成紧凑的形状约束词云,但本文优

① <http://www.tagxedo.com>

② <http://www.wordart.com>

化求解了每个位置处的摆词方向, 使摆词效果更加符合形状轮廓特征, 形成更加美观的布局效果。

词云交互和编辑功能可以改变词云的布局和外观, 提高词云使用的自由度。编辑交互主要包括重绘制和重排。重绘制改变单词的颜色等外观, 没有破坏词云的布局; 而词云重排包括对单词的添加、删除、移动、缩放和改变字体等操作, 它可能会破坏原有的词云布局, 因而需要再调整布局。ManiWordle^[12]和 Rolled-out Wordle^[13]提供了对词云布局的灵活控制能力, 允许用户操作整个布局或移动、旋转和删除单个单词。WordlePlus^[3]通过笔和触控设备让用户可以操作单词的摆放和聚集。当删除词云中单词时, WordlePlus 会自动使用词云边界附近的单词填充删除后留下的空白。EdWordle^[4]将每个单词看作一个刚体, 对刚体施加单词间吸引力和朝向画布中心的吸引力, 每次编辑完成后, 利用刚体动力学系统达到力学平衡以完成布局调整。ShapeWordle^[5]采用了 EdWordle 的部分编辑技术, 允许用户编辑大单词的位置、颜色、大小和摆放方向, 大单词编辑完成后再填充小单词。本文利用三角网格上的热扩散方法加速了距离场计算, 能更快速地生成形状约束词云布局, 并且能交互地改变布局焦点。

2 本文方法

形状感知的 Wordle 方法^[5]计算词云布局区域内各点距离图形轮廓边缘的距离场, 并以此生成与轮廓形状匹配的形状感知螺旋线, 然后从区域中心出发沿着螺旋线向外依次摆词, 形成紧凑的形状约束的词云布局。螺旋线运动方向计算公式为

$$\begin{pmatrix} dx \\ dy \end{pmatrix} = md\theta N_p + rd\theta T_p \quad (1)$$

其中, N_p 和 T_p 为像素 $p(x, y)$ 处距离场的单位法向量和单位切向量; $r = \sqrt{x^2 + y^2}$ 表示像素 $p(x, y)$

到阿基米德螺旋线起始点的距离; m 为控制相邻螺旋线之间距离的参数。阿基米德螺旋线的运动方向受 N_p 和 T_p 控制。

为了在尖锐特征等高曲率区域也能生成特征保持的螺旋线, 螺旋线的前进步长需考虑螺旋线的局部曲率, 式(1)可重写为

$$\begin{pmatrix} dx \\ dy \end{pmatrix} = \frac{mRd\eta}{r} N_p + Rd\eta T_p \quad (2)$$

其中, R 为局部曲率半径; η 为用户指定的角速度参数。

上述词云生成方法产生了较好的形状约束词云效果, 但存在如下缺点。首先, 摆词方向只采用水平和垂直方向 2 种, 摆词结果没有匹配轮廓形状特征, 产生的词云美观度不足; 其次, 上述方法在求距离场的梯度场 N_p 和切向场 T_p 时要多次遍历整个图像空间, 并且摆词时的词重叠检测需耗费较多时间, 影响了摆词的时间效率, 难以达到交互的词云布局性能; 最后, 上述方法产生的螺旋线都是从距离场大的中心区域往距离场小的边缘区域生长, 因此限制了摆词的中心位置, 缺少布局的灵活性。

本文利用三角网格上的热扩散方法快速地获得不同布局焦点下的距离场数据以及它的梯度场法向场, 以快速生成摆词螺旋线。同时, 结合手工交互、图像特征提取和二重旋转对称场插值获得各点处的契合轮廓特征的词云摆词方向。具体流程如图 1 所示, 首先在预处理阶段提取图形轮廓边缘上的离散点, 并以这些点和区域内的随机点三角化整个图形; 其次, 通过热扩散方程计算每个三角形顶点处的距离场, 并对它们进行 OpenGL 插值, 快速地获得各个像素处的距离场值, 进而获得它们的梯度和法向; 然后, 以图像的边缘特征或用户手工交互的线条作为方向约束, 插值生成整个形状区域内的方向场, 它们作为摆放单词的方向, 以生成契合图形轮廓特征的摆词效果; 最后, 利用产

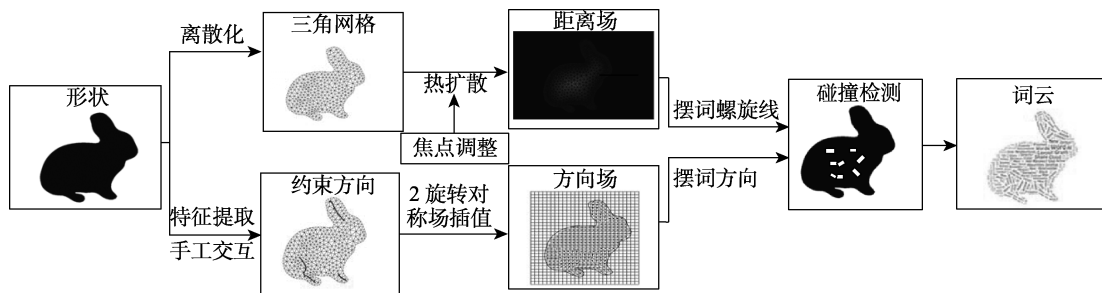


图 1 本文方法流程

生的梯度场和法向场,采用式(2)生成形状感知的螺旋线,在螺旋线的位置并基于上一步生成的摆词方向放置单词.在计算距离场时,用户可以通过自由定义焦点位置改变热扩散的距离场值,从而改变词云摆放的中心,获得不同摆词焦点下的词云布局效果.

2.1 预处理

当用户输入一幅图像后,使用边缘提取算子获取图形轮廓,轮廓区域内为摆词区域.如图 2a 所示,为减少计算量以获得交互的布局效果,对生成的图形区域进行三角网格化,不妨设生成的三角网格包含了 n 个顶点和 m 个三角形.后续的距离场求解(图 2b 和图 2c)和词云方向生成过程均在此网格上进行,而碰撞检测(图 2d)基于像素坐标计算.

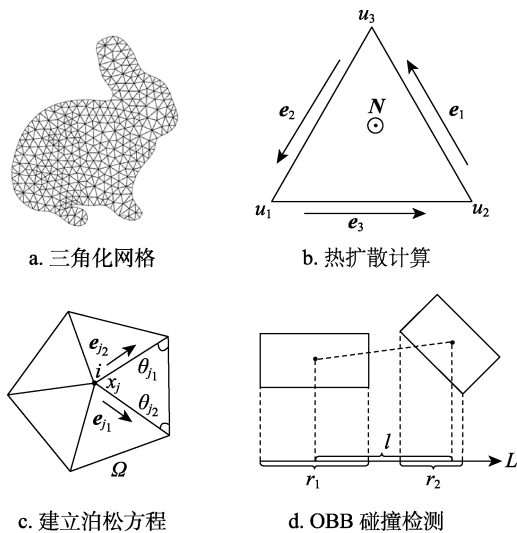


图 2 本文方法模型

2.2 距离场求解

借鉴 Crane 等^[25]方法,本文采用热传播方法求解距离场,同时通过改变热传播中的边界条件实现自由焦点变化下的词云布局效果.基于热传播的距离场求解过程包含 3 个步骤.

Step1. 对设定的某个固定时刻 t , 求解热流方程 $\dot{u} = \Delta u$, 获得各个顶点 i 处的温度 u_i . 在三角网格上上述微分方程求解过程可转换为

$$u = (A + t \cdot \Delta)^{-1} \delta \quad (3)$$

其中, u 为各顶点处的温度组成的向量; A 为 $n \times n$ 对角矩阵, 其对角线上第 i 元素是第 i 个顶点关联的所有三角形面积之和; Δ 为三角网格上的拉普拉斯矩阵; δ 为三角网格顶点处的初始温度值组成的向量, 在轮廓边缘上各点处的值为 1, 其他位置处的值为 0. 式(3)计算了热从轮廓边缘点向内扩散, 经过时间 t 后各个顶点的温度值. 在焦点调整过程中, 需对用户选定的焦点位置

处的初始温度值设定为 0° 甚至更小的负数温度, 如此经过热扩散后, 此处的温度仍然处于最低值, 即距离场值最大. 在上述求解中, 若 t 取值过小, 则热量尚未扩散到图形区域内的各顶点处; 否则, 各个顶点处的温度由于热平衡会相等, 2 种情况下均无法计算各顶点处的距离场值. 本文首先计算所有三角形边长度的平均值 l , 再令 $t = l^2$. 实验表明, 如此设置此参数值能获得较好的实验效果.

Step2. 计算各三角面片上的热扩散场 $X = -\nabla f / \|\nabla f\|$.

其中, 温度梯度 $\nabla f = \frac{1}{2A} \sum_{j=1}^3 u_j (N \times e_j)$; A 为面片的面积; N 为它的法向; u_j 为其 3 个顶点处的温度, 它们在 Step1 中获得; e_j 为三角形中第 j 个顶点的对边向量, 如图 2b 所示.

Step3. 解泊松方程 $\Delta \varphi = \nabla \cdot X$ 获得各个顶点处的距离场 φ , 在第 i 个顶点处, 有

$$\nabla \cdot X = \frac{1}{2} \sum_{j \in \Omega} \cot \theta_{j1} (e_{j1} \cdot X_j) + \cot \theta_{j2} (e_{j2} \cdot X_j).$$

如图 2c 所示, Ω 为第 i 个顶点的一环邻域三角形面片. 在三角网格上, 上述距离场求解转换为求解方程组 $\varphi = \Delta^{-1} (\nabla \cdot X)$.

当通过上述方法获得所有三角形顶点上的距离场值后, 本文使用 OpenGL 的 Gouraud 着色方式将距离场值插值至每个像素. 由于对区域进行了三角化, 因此其相对于 ShapeWordle 方法求解过程的计算复杂度大大降低. 其中最大的计算量来自求解各顶点处的 u 和距离场 φ , 但由于矩阵 $A + t \cdot \Delta$ 和 Δ 只有 $n \times n$ 维 (n 远小于像素分辨率), 并且是稀疏矩阵, 因此计算速度快. 一旦获得各个顶点处的距离场后, 利用 Gouraud 着色中的硬件插值, 能快速地获得各个像素的距离场值.

2.3 词云方向生成

为产生与图形轮廓和图像特征相匹配的词云布局方式, 需对图形区域中每个像素产生一个摆词方向向量, 使摆放在此位置的单词按此方向放置. 本文利用用户在区域中手工刷画的线条或通过图像处理提取的图像中的强边缘特征作为约束线条. 在这些约束线条处, 词云摆词方向需与线条的切向方向保持一致. 为确保中心区域的词云以水平方向摆放, 以保证良好的词云可读性, 本文使用中心约束限制中心区域的摆词方向, 即将距离场值大于用户预设阈值 ε 的像素处的单词摆放方向约束为水平方向. 然后, 利用二重旋转对称场方法^[26]逐点优化插值这 2 类约束条件下的摆词方向, 以生成所有顶点处的单词摆放方向.

对三角网格的每个顶点 i 定义其二重旋转对

称场代表向量为 $\mathbf{R}_i = \begin{pmatrix} \cos 2\theta_i \\ \sin 2\theta_i \end{pmatrix}$. 其中, $\theta_i \in [0, \pi]$ 为顶点 i 处的摆词方向角. 利用

$$\min_{\theta_i} \left(\sum_{E_{jk} \in E} \|\mathbf{R}_j - \mathbf{R}_k\|^2 + \sum_{i \in S} \|\mathbf{R}_i - \mathbf{T}_i\|^2 \right) \quad (4)$$

优化获得各个顶点 i 处的方向角 θ_i . 其中, E 为三角网格中所有边的集合; E_{jk} 为连接第 j 和第 k 个顶点的边, $\sum_{E_{jk} \in E} \|\mathbf{R}_j - \mathbf{R}_k\|^2$ 使相连顶点的摆词方向角相似; S 为约束线条经过的顶点和中心约束区域中的顶点的集合; $\mathbf{T}_i$ 为给定的约束摆词方向, $\sum_{i \in S} \|\mathbf{R}_i - \mathbf{T}_i\|^2$ 使约束线条经过的顶点和中心约束区域中的顶点的摆词方向角满足约束条件.

式(4)的优化求解过程可转换为稀疏方程组求解问题, 一旦获得各个顶点处的摆词方向场, 采用类似于第 2.2 节中的 OpenGL 插值方法获得各个像素处的摆词方向. 距离场插值和摆词方向场插值可各使用 2 个颜色通道通过一次 OpenGL 绘制同时进行, 从而提高插值效率.

2.4 碰撞检测

在词云摆放过程中, 新放入的单词不能与已摆放的单词发生重叠, 因而需要进行重叠判断. 单词本身占据一个矩形包围盒空间, 由于本文采用不同的摆词方向, 单词包围盒的两边未必与 x, y 坐标轴平行, 因此不能简单地通过单词包围盒在 x 和 y 坐标轴上的占据范围进行重叠判断. 本文采用层次方向包围盒(oriented bounding box, OBB)碰撞检测方法加快重叠检测过程.

首先对整个摆词区域建立一个矩形包围盒, 并对此包围盒剖分建立层次四叉树; 然后在摆词过程中, 对新放入的单词, 根据其包围盒中心位置和大小找到相应的四叉树节点, 并与节点中已存储的单词包围盒进行碰撞检测. 如果与已有的单词没有发生重叠碰撞, 则将单词摆放在当前位置, 并将它存储在相应的四叉树节点中; 否则沿着螺旋线找下一个摆放位置进行进一步碰撞检测.

如图 2d 所示, 在对 2 个单词进行 OBB 碰撞检测时, 以它们包围盒的边所在的方向作为 4 个投影方向, 如果存在一个投影方向 L , 当 2 个包围盒的边在此投影轴上的长度和 $r_1 + r_2$ 大于 2 个包围盒中心的连线的投影长度 l 时, 则它们不重叠, 否则它们会发生重叠.

在上述碰撞检测过程中, 假设填满区域的单词个数为 k 个, 考虑利用四叉树减少了检测的次

数, 因此总的检测复杂度为 $O(k \lg k)$, 每次检测需要进行 1~4 次(在 4 条边的方向投影)的重叠检测, 每次重叠检测涉及 4 次投影计算, 因而计算量较小.

2.5 两阶段填词

当填充的单词越来越多, 字体越来越小时, 碰撞检测的代价逐渐增大, 会影响交互效率. 而与此同时, 权重较小的词传递的信息较少, 因此本文采用类似于 ShapeWordle 方法, 着重处理权重较大的大单词, 而对权重较小的小单词用简单方法处理.

填词过程采用不同的策略分两阶段进行. 在第 1 阶段填充权重较大的单词, 此过程采用前述方法逐步进行, 这些大单词形成了词云的整体布局效果, 承担了重要的美学责任. 在焦点调整中, 焦点处的实际距离场值可能较小, 因此可能导致大单词无法填入. 为了解决此问题, 本文会根据原始模型中焦点处实际距离场值与形状中心处的距离场值的比值对词云进行自适应缩小操作, 因此产生的词云会出现字体变小情形. 在实际使用中, 用户也可以关闭此字体缩小功能.

在第 1 阶段完成后, 利用 OpenGL 绘制所有已摆放单词的矩形包围盒, 产生一个与画布相同分辨率的占位掩码. 掩码中的值标注了相应的像素是否已被填充. 第 2 阶段利用小单词填充大单词布局后形成的空隙, 以提高区域中的单词填充率, 并增加美观度. 在此阶段填入的单词只以水平或垂直放置, 因此可以快速确定它所占据的像素空间, 再利用第 1 阶段形成的掩码判断此像素空间中是否有像素已被覆盖, 即新加入的单词是否会与已摆放单词发生重叠. 如果没有重叠, 则在此位置填充此小单词, 并对掩码中相应的覆盖区域内像素设置占位标志; 否则沿着螺旋线查找下一个填充位置.

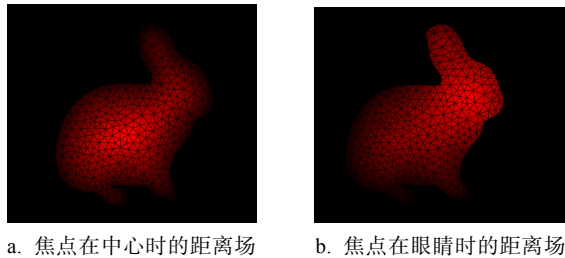
3 实验及结果分析

本节对本文方法和目前形状约束词云效果最佳的 ShapeWordle 方法进行实验对比, 为了保证比较的科学性, 使用相同的图案和文本输入生成相同分辨率的图像. 对比内容包括案例评估和定量评估 2 个方面.

3.1 案例

本文效果图中对大单词用红色颜色显示, 而对第 2 阶段填充的小单词使用蓝色颜色展示. 图 3c 所示为 ShapeWordle 的布局结果, 布局中心位于距离场值最大的兔子身体中心, 此时它的距离场

如图 3a 所示, 颜色越红表示距离场的值越大. 图 3b 所示为本文将焦点移动至兔子眼睛处时的距离场, 此时通过扩散方程获得的距离场在眼睛处最大, 随着与眼睛距离的增大而逐渐均匀变小, 相应的词云摆放效果如图 3d 所示, 较大的词出现在眼睛附近处. 对比 ShapeWordle 的词云布局效果, 本文方法提供了更加灵活的词云整体布局效果, 用户可以自定义词云布局的焦点.



a. 焦点在中心时的距离场

b. 焦点在眼睛时的距离场

c. ShapeWordle^[5]效果图

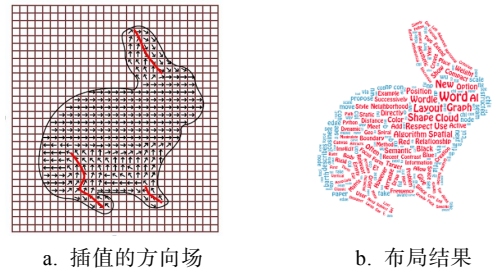
d. 本文焦点改变后的效果图

图 3 不同焦点位置时的效果图对比

如图 4a 所示, 改变大单词的摆词方向, 在兔子身上手工勾勒 3 条红色曲线作为约束方向, 这些曲线体现了兔子的腿部、脚和耳朵的形状走向特征, 同时加上中心约束, 生成的方向场如图中箭头所示. 由于此时焦点在兔子的眼睛处, 虽然其离实际边缘较近, 但通过热扩散计算的距离场值却最大, 因此眼睛附近的摆词方向约束为水平方向. 由于采用二重旋转对称场插值, 任意方向翻转 180° 后的方向与原方向在 2 旋转对称场下是相同的, 因此进行方向插值后会在图中某些地方出现方向突然转向的情况, 但这种结果保证了式(4)能量最小. 在词云布局中为了增加可读性, 对于倒立的单词会加以翻转, 因此 2 旋转对称场中的方向向量较适合用于表示词云摆放方向. 基于图 4a 方向场的词云布局效果如图 4b 所示, 可发现在耳朵、脚等用户交互的区域词云布局较好地表现了形状特征.

图 5 展示了鸽子和花瓣 2 个词云布局例子. 其中图 5a 所示为 ShapeWordle 的词云布局效果, 由于加入了小单词, 布局效果紧凑美观, 布局结果与模型形状高度契合; 图 5b 所示为本文改变布局焦点后的效果图; 图 5c 所示为改变了摆词方向后的

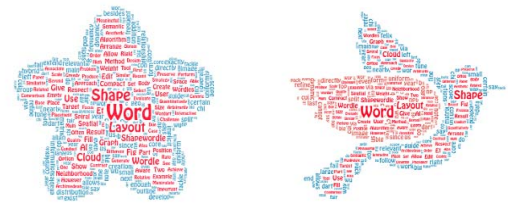
效果图, 词云布局在鸽子羽毛和尾巴处较好地表现了形状特征.



a. 插值的方向场

b. 布局结果

图 4 改变摆词方向效果

a. ShapeWordle^[5]效果图

b. 本文改变焦点后的效果图



c. 本文摆词方向约束后的效果图

图 5 改变焦点和摆词方向后的词云布局效果

图 6 展示了利用图像处理提取的图像特征引导摆词方向的布局效果. 图 6a 是原始图像; 图 6b 是对原始图像抽取的强边缘效果图, 图中只保留了区域内部的较长边缘, 并以这些边缘作为方向约束生成词云布局效果, 如图 6d 所示; 图 6c 是 ShapeWordle 生成的效果图.

3.2 定量评估

表 1 所示为本文方法与 ShapeWordle 方法的词云生成效率对比. 可以看出, 3 种形状的画布分辨率均相同, 离散后的三角网格顶点数量比像素数量小 2~3 个数量级. 在本文方法中, 距离场的求解计算量主要来自 2 次稀疏方程组求解, 方向场的求解计算量主要来源于一次稀疏方程组求解和一次



图 6 自然边缘方向约束前后的效果对比

表 1 本文方法与 ShapeWordle 效率对比

模型	分辨率/像素	顶点数	面片数	时间/ms				总时间/ms	
				距离场	方向场	大单词摆放	小单词摆放	本文	ShapeWordle ^[5]
兔子	900×600	231	307	135	93	235	252	736	5 724
	450×300	231	307	119	85	217	136	611	1 627
鸽子	900×600	327	469	183	116	383	201	1 108	6 138
	450×300	327	469	170	102	362	116	804	1 819
花朵	900×600	198	268	108	72	184	313	857	5 206
	450×300	198	268	97	64	169	207	589	1 314

OpenGL 绘制, 它们与三角形顶点数量相关, 与分辨率的关联性较小; 大单词摆词的计算量包括二叉树的构建与更新、螺旋线生成和碰撞检测, 它们与所摆的单词数量相关; 小单词摆词的计算量来自螺旋线生成、掩码中占位查询以及掩码更新, 因此它们与摆词的数量和分辨率均相关. 在实验中, 不同分辨率下所填入的单词数量保持不变, 单词尺寸按分辨率进行缩放. 由表 1 中数据可知, 本文方法随着分辨率的增加计算时间只是缓慢增加, 在较大分辨率时也能达到交互速度; ShapeWordle 方法中距离场的计算量与像素数量成正比, 由于只有垂直和水平 2 种摆词方向, 因此无须计算方向场, 同时摆放大单词时重叠判断也较简单, 小单词填充与本文方法类似. 表 1 中数据表明, 当分辨率提高时, 距离场计算时间的大大增加会导致词云生成速度快速下降. 之后, 用 3 个度量标准评估词云布局质量.

(1) 布局覆盖率(layout coverage, LC). 用于测量词云布局中的单词所占的空间比, 它定义为图形轮廓内文本颜色的像素数量与所有像素数量的比值. LC 值越接近 1, 意味着布局覆盖越好.

(2) 形状相似度(shape similarity, SS). 用于衡量词云布局与给定图形轮廓的匹配程度. 对于图形轮廓 Ω 上的每个像素 p_i , 在轮廓内找到与之距离最近的文本颜色像素, 记它们之间的距离为 $\rho(p_i)$, 形状相似性度量定义为

$$SS = \frac{1}{n_{\text{contour}}} \sum_{p_i \in \Omega} \rho(p_i)^2.$$

其中, n_{contour} 表示图形轮廓上像素的数量. SS 值越小, 表示词云布局更好地匹配了图形轮廓.

(3) 方向吻合度(direction similarity, DS). 用于衡量词云布局中单词的摆放方向与轮廓形状的吻合程度. 对于轮廓内的每个单词, 求与单词中心距离最近的轮廓点, 记此轮廓点处轮廓线的切向方向为 d_i , 单词的摆词方向为 w_i , 计算每个单词的摆词方向与其最近轮廓点处切向的偏差, 并求和归一化, 即

$$DS = \frac{1}{n_{\text{word}}} \sum_{i=1}^{n_{\text{word}}} \frac{|d_i \cdot w_i|}{s_i}.$$

其中, n_{word} 表示轮廓内单词的数量; s_i 表示单词中心到最近轮廓点的距离, 与外轮廓距离较近的单词具有更大的权重. DS 值越小, 表示单词放置的方向与图形轮廓方向越吻合.

本文使用从英文维基百科中摘录的 112 篇文章中的内容作为词云单词, 并收集了 112 个不同类别的形状, 如动物、植物、运动和数字图形等, 然后随机地为每个文档分配一个形状. 本文方法同时生成未改变焦点和随机改变焦点后 2 种效果图, 统计结果如图 7 所示, 本文未改变焦点布局结果的布局覆盖率优于 ShapeWordle 方法, 原因是本文方法在放置单词时使单词摆词方向与整个图形的轮廓及骨架走向保持一致, 使摆放更加紧凑. 而

ShapeWordle 采用水平或竖直摆放, 导致在一些狭小部位无法放置单词而产生较大空隙. 改变焦点后的布局图的布局覆盖率有所下降, 原因是有些布局焦点选择在距离场较小的位置, 导致字体较大单词放置位置不恰当而产生间隙, 并且距离场变形导致摆词螺旋线间距不均匀, 使有些区域摆词不够紧凑. 本文摆词方向与图形轮廓走向契合也使摆词结果更加贴合轮廓, 因此本文方法的形状相似度也优于 ShapeWordle 方法. 同时, ShapeWordle 方法由于只采用水平和垂直摆词方向, 因此方向吻合度明显小于本文方法.

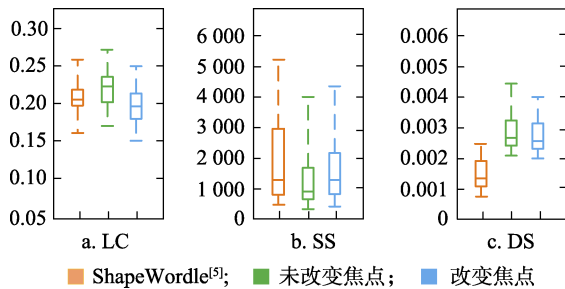


图 7 不同方法的词云布局质量定量对比

3.3 用户调查

邀请多名用户使用本文的词云布局工具, 并和其他词云工具进行比较, 采访他们的使用感受以及对词云布局效果的评价. 在此过程中, 统一使用了马丁·路德·金的《我有一个梦想》的英文演讲稿作为文本数据.

本文招募了 42 名志愿者(25 名男性, 17 名女性). 其中 15 人主修计算机科学, 7 人主修通信工程, 7 人主修工业设计, 另外 13 人为人文学院和英语学院学生, 所有参与者均未使用过词云工具, 其中 16 人在互联网见过其他标签云效果. 每位参与者均在一台 4 核个人计算机上工作, 它带有一个分辨率为 1920×1200 像素的晶宽屏显示屏.

实验 1. 检验本文方法能否反映出焦点位置信息, 给定 10 组本文方法生成的词云布局, 分别由参与者观察分析, 让他们标记出其中认为是词云布局的焦点位置, 然后计算它们与实际词云焦点位置的平均偏差, 即

$$D_1 = \frac{1}{n} \sum_{i=1}^n \sqrt{(x_i - x_{i_focus})^2 + (y_i - y_{i_focus})^2}.$$

其中, $n=10$ 表示 10 组实验数据; (x_i, y_i) 表示参与者标记的坐标; $(x_{i_focus}, y_{i_focus})$ 表示词云布局时的实际焦点坐标. 实验结果显示, 42 位参与者标记的焦点坐标与实际的焦点位置的偏差的期望值为

9.14 像素, 方差为 5.20 像素, 与真实的焦点位置偏差较小, 说明本文的词云布局方法能较好地反映出词云布局的焦点效果.

实验 2. 给定 10 组通过本文方法生成的词云布局 and 相应的 ShapeWordle 方法生成的布局, 分别由这些参与者观察分析, 并向他们提出了如下 3 个问题.

(1) 自由焦点变化是否有必要, 能否产生比 ShapeWordle 更美观的布局效果?

(2) 改变摆词方向是否有必要, 是否产生了比 ShapeWordle 更美观的布局效果?

(3) 是否满意本文方法的交互速度和交互方式?

答案可选择“是”“否”“无法回答”3 个选项.

对于第 1 个问题, 有 23 位参与者选择“是”, 12 位参与者选择“否”, 7 位参与者选择“无法回答”, 表明对自由焦点的布局方式存在较大争议, 选择“是”的参与者给出的主要理由是“给了用户更多的选择”和“有时更加美观”. 对于第 2 个问题, 有 33 位参与者选择“是”, 9 位参与者选择“否”, 无人选择“无法回答”, 说明大部分参与者都觉得改变摆词方向有必要, 他们的主要观点是“能够较好地表现模型的形状特征”和“美观度增加不少”, 而选择“否”的 9 位参与者的反对理由是“有些地方显得线条紊乱, 特征更不明确了”和“可读性没有 ShapeWordle 好”. 对于第 3 个问题, 有 40 位参与者选择“是”, 只有 2 位参与者选择“否”, 理由是“交互速度尚可, 但交互方式最好能增加布局微调功能”.

实验 3. 给定 10 组利用本文方法生成的词云布局图, 每组包含一幅未调节焦点时的布局图和 3 幅不同焦点位置时的布局图. 用户观察后需给出每幅词云布局图中最重要的 10 个关键词, 分别组成关键词集合 $K_{i1}, K_{i2}, K_{i3}, K_{i4} (i=1, 2, \dots, 10)$, 其中 K_{i1} 表示用户从第 i 组中未改变焦点布局图中识别的 10 个关键词组成的集合, K_{i2}, K_{i3}, K_{i4} 表示 3 个不同焦点位置的布局图中识别的关键词集合. 然后对每位用户使用杰卡德距离 J 计算焦点改变对其关键词识别能力的影响, 即

$$C = \frac{1}{30} \sum_{i=1}^{10} \sum_{j=2}^4 J(K_{i1}, K_{ij}).$$

42 位参与者在焦点改变下识别的关键词集与焦点未变时识别的关键词集的杰卡德距离的期望值为 0.77, 相当于有 8.7 个关键词相同, 方差为 2.2 个关键词. 因此, 文本的焦点调节功能基本不影响用户对词云结果的理解.

4 结 语

本文提出的词云布局方法可以帮助用户快速创建任意形状的、焦点可调的词云布局. 该方法的核心是利用热扩散原理实现快速求解距离场, 并且通过改变热扩散中热源的位置和温度实现焦点调整功能. 在此基础上, 为了取得更加美观的布局效果和传递更加丰富的信息, 本文将传入的图像线条化, 得到图像中的强边缘特征线条, 或手工交互放置约束线条, 然后利用这些线条作为摆词走向的线索, 优化产生整个图像空间中的所有像素处的单词摆放方向, 实现了词云布局与形状特征的匹配.

由于本文方法中单词分布是语义无关的, 无法保证语义关联的单词放置在相邻的位置, 因此在以后的工作中将加入语义关联约束. 另外, 在一些形状特征线条曲率较大的地方, 摆放的单词较杂乱, 主要原因是这些地方摆放的单词长度偏长, 无法实现较真实的弯曲效果, 未来工作中需考虑单词长度和曲率等因素使各个单词能摆放在最合适的位置. 最后, 希望在今后的工作中, 在词云生成工具中加入更多用户交互的功能, 让用户获得更好的用户体验.

参考文献(References):

- [1] Viegas F B, Wattenberg M, Feinberg J. Participatory visualization with wordle[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2009, 15(6): 1137-1144
- [2] Chi M T, Lin S S, Chen S Y, *et al.* Morphable word clouds for time-varying text data visualization[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2015, 21(12): 1415-1426
- [3] Jo J, Lee B, Seo J. WordlePlus: expanding Wordle's use through natural interaction and animation[J]. *IEEE Transactions on Computer Graphics and Applications*, 2015, 35(6): 20-28
- [4] Wang Y H, Chu X W, Bao C, *et al.* EdWordle: consistency-preserving word cloud editing[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2018, 24(1): 647-656
- [5] Wang Y H, Lee B, Chu X W, *et al.* ShapeWordle: tailoring wordles using shape-aware archimedean spirals[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2020, 26(1): 991-1000
- [6] Bao Chen, Wang Yunhai. A survey of word cloud visualization[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2021, 33(4): 532-544(in Chinese)
(包琛, 汪云海. 词云可视化综述[J]. *计算机辅助设计与图形学学报*, 2021, 33(4): 532-544)
- [7] Lee B, Riche N H, Karlson A K, *et al.* SparkClouds: visualizing trends in tag clouds[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2010, 16(6): 1182-1189
- [8] Collins C, Viegas F B, Wattenberg M. Parallel tag clouds to explore and analyze faceted text corpora[C] // *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*. Los Alamitos: IEEE Computer Society Press, 2009: 91-98
- [9] Bielenberg K. Groups in social software: utilizing tagging to integrate individual contexts for social navigation[D]. Bremen: University of Bremen, 2005
- [10] Stefaner M. Visual tools for the socio-semantic web[D]. Potsdam: University of Applied Sciences Potsdam, 2007
- [11] Viegas F B, Wattenberg M, Feinberg J. Participatory visualization with Wordle[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2009, 15(6): 1137-1144
- [12] Koh K, Lee B, Kim B, *et al.* ManiWordle: providing flexible control over Wordle[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2010, 16(6): 1190-1197
- [13] Strobel H, Spicker M, Stoffel A, *et al.* Rolled-out Wordles: a heuristic method for overlap removal of 2D data representations[J]. *Computer Graphics Forum*, 2012, 31(3pt3): 1135-1144
- [14] Seyfert M, Viola I. Dynamic word clouds[C] // *Proceedings of the 33rd Spring Conference on Computer Graphics*. New York: ACM Press, 2017: Article No.7
- [15] Shu X H, Wu J, Wu X K, *et al.* DancingWords: exploring animated word clouds to tell stories[J]. *Journal of Visualization*, 2021, 24(1): 85-100
- [16] Cui W W, Wu Y C, Liu S X, *et al.* Context-preserving, dynamic word cloud visualization[J]. *IEEE Computer Graphics and Applications*, 2010, 30(6): 42-53
- [17] Wang J, Zhao J, Guo S, *et al.* ReCloud: semantics-based word cloud visualization of user reviews[C] // *Proceedings of Graphics Interface*. New York: ACM Press, 2014: 151-158
- [18] Wei F R, Liu S X, Song Y Q, *et al.* TIARA: a visual exploratory text analytic system[C] // *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press, 2010: 153-162
- [19] Knittel J, Koch S, Ertl T. PyramidTags: context-, time- and word order-aware tag maps to explore large document collections[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2021, 27(12): 4455-4468
- [20] Maharik R, Bessmeltsev M, Sheffer A, *et al.* Digital micrography[J]. *ACM Transactions on Graphics*, 2011, 30(4): Article No.100
- [21] Buchin K, Creemers D, Lazzarotto A, *et al.* Geo word clouds [C] // *Proceedings of the IEEE Pacific Visualization Symposium*. Los Alamitos: IEEE Computer Society Press, 2016: 144-151
- [22] Reckziegel M, Cheema M F, Scheuermann G, *et al.* Predominance tag maps[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2018, 24(6): 1893-1904
- [23] Reckziegel M, Janicke S. Time varying predominance tag maps[C] // *Proceedings of the IEEE Visualization Conference*. Los Alamitos: IEEE Computer Society Press, 2019: 231-235.
- [24] Afzal S, Maciejewski R, Jang Y, *et al.* Spatial text visualization using automatic typographic maps[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2012, 18(12): 2556-2564
- [25] Crane K, Weischedel C, Wardetzky M. Geodesics in heat: a new approach to computing distance based on heat flow[J]. *ACM Transactions on Graphics*, 2013, 32(5): Article No.152
- [26] Palacios J, Zhang E. Rotational symmetry field design on surfaces[J]. *ACM Transactions on Graphics*, 2007, 26(3): Article No.55