

基于聚类的二维向量场可视化

王少荣^{1,2)}, 汪国平^{2,3)}

¹⁾(北京林业大学信息学院 北京 100083)
²⁾(北京市虚拟仿真与可视化工程技术研究中心 北京 100871)
³⁾(北京大学图形与交互技术实验室 北京 100871)
(shaorongwang@hotmail.com)

摘 要: 已有的二维流场可视化中,鞍点等临界点是最重要的特征之一.文中从一个新的角度提出一种基于流线聚类的二维向量场可视化方法.首先生成采样流线集合,然后将流线聚类,最后引入共轭法向量场和流线密度矩阵对同一个类的流线进行加速排序.在此基础上,提出 3 种可视化应用:抽取每一类的代表流线进行向量场的流线简洁表达;根据流线之间距离进行多分辨率均匀流线表达;生成权值图,增强基于纹理的向量场可视化.实验结果表明,该方法具有良好的鲁棒性,可视化效果优于已有的方法.

关键词: 向量场可视化;聚类;向量场简化;密度矩阵
中图法分类号: TP391

Clustering Based 2D Vector Field Visualization

Wang Shaorong^{1,2)} and Wang Guoping^{2,3)}

¹⁾(School of Information Science and Technology, Beijing Forestry University, Beijing 100083)
²⁾(Beijing Research Center of Virtual Simulation and Visualization, Beijing 100871)
³⁾(Graphics and Interactive Technology Laboratory, Peking University, Beijing 100871)

Abstract: Critical points such as saddle point are one set of the most important features in 2D vector field visualization methods. A novel clustering based approach is proposed in this paper. First, a set of streamlines are generated, and are clustered into different groups. Conjugated normal vector field and dense matrices are introduced to accelerate the sorting of the streamlines in each group. Thereafter, different post-processes can be performed according to various visualization applications: a streamline simplification that works for each group; a multi-resolution evenly-spaced streamline placement based on streamline proximity; an enhanced texture based visualization based on the weighted matrix. Experiments show that our results are satisfactory and robust.

Key words: vector field visualization; clustering; vector field simplification; dense matrix

向量场在科学计算和工程分析中描述了许多非常重要常见的物理现象.向量场可视化是理解向量场运动变化规律的有力工具,在计算和工程分析中发挥着重要作用,已被广泛应用于航空航天、地质勘探、仿真、临床医学等领域,具有非常重要的实际意

义和研究价值.
在过去数年的研究中已经出现了很多向量可视化的方法,最常用的有基于流线的可视化和基于纹理的可视化方法.基于流线的可视化方法提供了向量场的一个稀疏表达,突出流线的特征以重点关注

收稿日期:2014-05-02;修回日期:2014-07-23. **基金项目:**北京林业大学新进教师科研启动基金(BLX2011019);国家“九七三”重点基础研究发展计划项目(2010CB328002);国家自然科学基金(61173080). **王少荣**(1975—),男,博士,讲师,CCF 会员,主要研究方向为计算机图形学、虚拟现实、可视化;**汪国平**(1964—),男,博士,教授,博士生导师,CCF 会员,主要研究方向为计算机图形学、虚拟现实、人机交互.

显著特征,并且易于与其他可视化方法进行协同.流线可视化的质量严重依赖于流线种子点位置选取、所放置流线的数量以及长度等.已有的工作中,种子点的选取和向量场的拓扑结构密切相关,基于拓扑结构的流线可视化一直是流线可视化中的重要研究方向.二维向量场的临界点大致可以分为汇聚或者发散类型、循环类型和鞍点类型.已有工作中,鞍点往往是十分重要的临界点.文献[1]提出基于鞍点的流线可视化方法,文献[2]专门包含了一种鞍点类型的模板,文献[3]首先对鞍点附近区域进行处理以提高流线可视化的效果.

本文从另一个角度来考察向量场的特征.首先用种子点采样的方法生成一个向量场的完整流线集,然后根据流线的起始点位置将这些流线聚类,则整个流线集被分成若干个流线类.在这种观察角度之下,临界点中的鞍点并不是最重要的特征.鞍点区域可以看成几个不同类别之间形成的区域,如果把每一个流线类的边界找出来,鞍点区域也随即被刻画出来,并不需要进行额外的讨论.这种做法具有如下特点:

1) 流线分类是完备的,不会遗漏任何区域,聚类也不会重复,因而对向量场的刻画是完备的.

2) 流线的端点是临界点或者在边界上.通过分析每一类流线的端点,同样也是分析临界点的属性;此外,根据端点和流线类之间的关联,同样可以得到拓扑关系.

3) 对于同一类别的流线,它们的端点都是相邻的,因而同一类中的流线性质的也是类似的,可以从中抽取出具有代表性的流线来展示向量场.

4) 和已有方法相比,在表达临界点信息方面本文方法具有一定优势.已有的方法一般利用事先给定的模板,而这种模板在某些情况尤其是多个临界点靠近的情况下往往会失效.本文从相邻的流线类抽取出靠近边界的流线,鞍点等临界点就在这些流线围成的区域内.

完整流线集聚类后,对于每一个流线类的所有流线按照空间的位置关系进行排序;然后根据不同应用的需要进行后续处理,以满足不同的可视化要求.由于完整流线比较多,且基本上覆盖了整个向量场,因此流线聚类的结果也可以用于代表向量场的属性.排序之后的流线可以根据流线之间的距离进行抽取流线等操作,因而可以根据不同可视化应用的需求抽取出符合要求的最佳流线组合.此外,流线排序之后可以计算相邻流线之间的距离,根据距离

可以生成权值对基于纹理的向量场可视化结果进行增强.

已有工作也有基于聚类的向量场可视化方法. Shene 等^[4]提出一种基于层次聚类的三维流线可视化方法, Oeltze 等^[5]使用多种聚类方法进行聚类.和已有工作相比,本文方法聚类流线的种子点均匀采样,并且采样数量较大,用于聚类的流线更能代表向量场.本文使用流线端点之间的距离进行聚类,同一类流线性质的完全一致,缺点是可能导致聚类的数目相对较多.本文方法和已有工作相比最大不同之处是对聚类后结果的后续处理方式不一样,本文方法对同一类流线按空间拓扑进行排序,然后进行进一步的后续处理,这样可以做相应的后续处理,如选择代表性流线以及生成纹理权值图.

1 相关工作

和本文相关的工作主要包括基于流线以及基于纹理的二维流场可视化方法.

流线可视化一直是二维流线可视化中重要的研究方向,近年来种子点放置方法一直是研究的重点.这些方法的目标大致分为两大类:一类强调达到均匀放置流线的效果,另一类则凸显出流线的特征. Jobard 等^[6]提出一种均匀分布流线的方法,该方法直观易于理解,但由于采取局部最优策略,无法保证全部流线在图中的均匀分布. Mebarki 等^[7]提出最远点的种子放置方法,其在距离所有已放置流线的最远点处放置种子点,但采用的策略不适合应用于大数据集,另外当密度增大时会出现流线不连续的状况. Turk 等^[8]提出能量函数来生成均匀流线,可以得到很好的效果,但是耗费时间较长. Verma 等^[2]提出一种基于临界点的种子点放置方法,其优点在于临界点附近的特征被很好地展现,缺点是不能保证流线的均匀性.

在流场可视化中还有一个活跃的分支是基于纹理的可视化方法,该方法主要包括点噪声^[9]和 Cabral 等^[10]提出的线卷积分(line integral convolution, LIC)算法. LIC 基于向量方向的相关性对噪声纹理低通滤波,用一个卷积核沿流线方向对噪声纹理进行卷积,从而生成一个可以表达向量场的静态图像.

有很多关于二维向量场简化的技术,其中常见的是聚类. Heckel 等^[11]提出一种自顶向下的简化方法,通过聚类和主成分分析进行迭代优化. Telea 等^[12]提出一种自底向上的简化方法. Du 等^[13]提出一种

用质心 Voronoi 图镶嵌的简化和可视化方法. Shene 等^[4]提出一种基于层次聚类的三维流线可视化方法,其中种子点的选择基于流的显著性. Oeltze 等^[5]通过定义相似性用多种聚类方法进行聚类,选取每一类的代表性流线进行可视化.

2 向量场区域的分类

为了确定向量场的性质,可以对向量场进行分割. 本文使用流线作为向量场的基本描述工具,向量场中一个区域内的性质由该区域内的所有流线来表达. 流线的性质由整个流线所经过位置的向量方向决定,相邻的流线可能具有完全类似的走向,也可能差别很大. 本文希望将具有一致走向的流线聚为一类.

本文中,算法的精度和流线的精度有很大的关系,为了保证计算的精确性,本文使用四阶的 Runge-Kutta 方法来计算流线. 为了方便表述,先给出一个定义. 对于向量场中任给的一点,如果该点的向量不是 $\mathbf{0}$ 向量,则存在一条经过该点的流线. 如果这条流线满足以下情形之一,则称该流线为经过该点的完整流线:1)起点和终点相连,即该流线为闭合曲线;2)流线端点位置处的向量值为 $\mathbf{0}$ 或者流线端点在向量场的边界.

首先均匀生成一个覆盖整个向量场的种子点集合,以这些种子点生成一个完整的流线集合. 给定二维向量场,将其定义域分成 $m \times n$ 个正方形区域,找出每个区域的中心点,以每个中心点作为种子点生成一条完整流线. 为了方便讨论,本文把不同种子点生成的重合流线当成不同的流线,这样得到一个包含 $m \times n$ 条完整流线的完整流线集.

2.1 流线聚类算法

为了对完整流线集的流线进行分类,首先给出 2 条流线能聚到同一类的条件. 如果 2 个点都是边界上的点,或者都是同一种类型的临界点,那么称这 2 个点的类型相同. 如果 2 条流线的起点和终点的类型分别相同,且起点和起点、终点和终点之间的距离小于给定的距离,则称这 2 条流线相似. 对于 2 个集合,如果存在 2 条相似的流线,则称这 2 个集合相似. 本文使用最近邻聚类算法对完整流线聚类,把那些端点之间距离小于给定阈值的流线聚成一类. 聚类算法步骤如下:

Step1. 生成一个完整流线集合 $C = \{l_i, i = 1, 2, \cdots, n\}$.

Step2. 生成一组集合 $S_i = \{l_i\}, i = 1, 2, \cdots, n$, 每个集合包含 1 条完整流线.

Step3. 对于任何 2 个集合 S_i, S_j , 如果它们相似,则将它们合并成一个新的集合 S_k , 并删去 S_i, S_j .

Step4. 重复 Step3, 直到没有可以合并的集合.

该聚类算法的时间复杂度为 $O(n^2)$. 为了减少算法开销,可以先对完整流线集的流线按端点类型进行预处理,分成几个不同的集合;然后对同一个集合分别使用上述聚类算法,从而加速聚类的过程. 阈值的取值和完整流线集种子点之间的间距有关,本文取值为间距的 1.5 倍.

2.2 流线端点的确定

确定流线的起始位置对于流线的聚类非常重要. 如果流线的端点在边界或者临界点附近,端点在边界的情况容易确认;而端点在临界点附近的情况则计算较为复杂,可根据不同临界点的情况进行处理. 流线的端点不可能是鞍点,只能是排斥结点、吸引结点、排斥焦点、吸引焦点和中心点.

对于临界点是中心点的情况,该处流线的起点和终点重合,由于无法找到一个确切的起点或者终点,所以本文采用流线所围绕的中心作为端点;对于排斥结点和吸引结点类型的临界点,流线端点逼近临界点. 流线计算时,如果流线前进速度小于给定的阈值,则可认为到达流线的端点;对于临界点是排斥焦点和吸引焦点的情况,流线以螺旋方式逼近临界点,数值计算时,流线围绕着临界点以比较慢的速度逼近临界点,先计算出所围绕的中心,然后直接以此中心作为端点.

3 基于聚类的向量场可视化

在本文的可视化算法中,需要对聚类后的流线按照空间拓扑顺序进行排序,然后计算相邻流线之间的距离,用于后续的处理.

3.1 流线的排序

任选一个聚类的流线集,对于任意 2 条流线,除了在端点处位置之外,2 条流线之间没有其他交点. 如果把所有的起始点看做同一个点 S ,把所有的终止点看做同一个点 E ,则同一类中的所有流线在二维平面上围出一个封闭的区域 R . 任取集合中的一条流线 l ,则 R 被 l 分成 2 个区域 R_1 和 R_2 , R_1 和 R_2 中的流线相互不相交. 以流线 l 的前进方向为正方向, l 上任取一点 P , 设该处的向量为 $\mathbf{V}(x, y)$, 取 P 位置处向量 $\mathbf{N}(y, -x)$, 则 $\mathbf{N}$ 和 $\mathbf{V}$ 垂直. 称 $\mathbf{N}$ 所指的方向为流线 l 的右侧, $-\mathbf{N}$ 所指方向为 l 的左侧,这样就定义了同一流线类中流线之间的左右顺序,可以将同一类中的流线按从左到右的顺序进行排序.

对任意 2 条流线 l_1, l_2 排序,其中一种简单的做法是在区域 R 中找一条曲线 c 与 l_1 和 l_2 相交,交点分别为 P_1, P_2 ,根据交点情况判断 2 条曲线的顺序.理论上,除了端点处之外任意 2 条流线没有交点,但在实际的数值计算中,由于精度的原因,相邻的流线也可能在端点之外的地方相交,这时交点 P_1 和 P_2 的顺序并不能反映出流线之间的顺序.如果所在区域流线越密集,流线之间的间距越小,则交点不能反映流线拓扑顺序的可能性越大.所以在实际排序过程中,为了避免由于精度带来的排序误差,应尽可能找到合适的曲线 c ,使得 c 和 l_1 和 l_2 相交的交点 P_1 和 P_2 之间的距离尽可能大.

在实际的曲线求交计算中,曲线表达为离散的多个线段构成的折线段.设 2 条曲线包含的线段数量分别为 K_1 和 K_2 ,则判定这 2 条曲线求交运算的数量级为 K_1K_2 ,这个过程需要大量计算,为了加速此计算过程,下面引入共轭法向量场和图像空间以及在图像空间上定义的流线密度矩阵.

3.1.1 共轭法向量场和图像空间

对于大小为 $m \times n$ 的向量场 V ,引入一个同样大小的向量场 V_c .如图 1 所示,给定向量场定义域中任意一个位置 $P(i, j)$,如该处向量为 (x, y) ,令 V_c 中在 (i, j) 处的向量值为 $(-y, x)$,则 V_c 和 V 在相同位置的向量模大小相等、方向相互垂直,称向量场 V_c 为向量场 V 的共轭法向量场.任给一个位置 P ,以 P 为种子点,分别在 V 和 V_c 中生成流线 l 和 l_n ,则 l 和 l_n 在 P 处的切线相互垂直,称 l 和 l_n 在 P 处垂直, l_n 为向量场 V 中过 P 点的法流线, l_n 位于 l 左侧的部分称为左法流线,位于 l 右侧的部分称为右法流线.由定义,法流线与所有和它相交的流线都垂直.对于流线 l_1, l_2 ,如果 l_n 和 l_1 交点为 P_1 , l_n 和 l_2 的交点为 P_2 ,称曲线段 P_1P_2 为法流线 l_n 被 l_1, l_2 截断的曲线,曲线段 P_1P_2 的长度称为截断长度.求交判断时,截断长度越长曲线之间的间距越大,排序越准确.

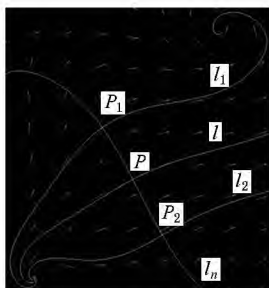


图 1 法流线和截断曲线

把向量场定义域所在的空间称为模型空间.给定放缩系数 s ,定义大小为 $(m \times s) \times (n \times s)$ 的图像空间.模型空间中点 (x, y) 变换到图像空间中点 (x', y') 的变换公式为 $x' = [x \times s], y' = [y \times s]$,其中 $[]$ 为取整运算符.图像空间中的点是离散的,每个点的坐标只能是整数.模型空间中的一条流线,对应像素空间的像素点序列.

为了描述向量场中经过某一个点的流线的疏密程度,在像素空间引入流线密度矩阵 M ,矩阵的每个元素记录图像空间中通过该位置的流线数目.如果 $M(i, j)$ 的值比较大,表明通过该位置的流线比较多;如果 $M(i, j)$ 的值为 0,表明该处没有流线通过.排序的求交判断尽量不要在 $M(i, j)$ 值比较大的位置处进行.

3.1.2 排序算法

为了对流线类排序,过其中一条流线上的某一点 P 引法流线 l_n ,希望 l_n 和这些流线交点之间的间距尽可能大.在图像空间快速计算 l_n 的截断长度,将 P 转换到图像空间 P' , l_n 转换到图像空间得到 l'_n ,从 P' 出发,先沿正向顺序遍历 l'_n ,同时查找 M 在对应位置的密度值,直到密度值为 0(即 l'_n 离开有效区域)或者大于给定阈值(经过该处位置的流线比较多,使用求交排序的错误可能性大),记该处为 E' .再从 P' 出发,反方向重复这个过程,得到 S' ,则曲线 $E'S'$ 为法流线 l'_n 和流线类流线相交的截断曲线.找到合适的 P 和 l_n ,使得生成的截断曲线 SE 最长,然后让该法流线和流线类求交,并对交点进行排序,即可得到流线的顺序.

本节中对同一个流线类中所有流线排序.首先选取第一条流线,用这条流线把集合分成 Q_L 和 Q_R 2 个有序队列,然后分别对 Q_L 和 Q_R 中的流线进行排序.

Step1. 选第一条流线.

如果流线类的端点不是临界点,则端点在图像空间中对应着一段轮廓.从轮廓(如果 2 个端点都不是临界点,则选长的轮廓)中挑出中间的一个像素 P' ,从经过该像素的流线中任意取一条流线 l , l 左侧的流线加入 Q_L , l 右侧的流线加入 Q_R .经过 P 的所有流线同时加入 Q_L 和 Q_R .

如果流线类的所有端点都是临界点,任意选取一条流线 l ,根据长度均匀取点 $P_i(i=1, \dots, n)$,过 P_i 生成一条法流线 l_{ni} .选取有最长截断曲线的法流线 l_n ,找出 l_n 被截取部分的中点,从经过该中点的流线中任取一条流线 l 作为第一条流线.剩下的所有流线同时加入 Q_L 和 Q_R .

将 l 放入已排序流线队列 Q_0 .令 $l_c = l$.

Step2. 对 Q_L 排序.

Step2. 1. 将 l_c 转换到图像空间,在图像空间中根据曲线长度对 l_c 均匀取点 $P_i(i=1,\cdots,n)$,过每个 P_i 生成一条左法流线 l_{ni} ,选取最长截断长度的法流线 l_n .

Step2. 2. l_n 和 Q_L 中未排序的流线求交并排序,直到经过交点 C 的流线过密(如果 C 在图像空间对应位置 C' ,那么经过 C' 的流线数超过给定的阈值,即 M 在该处的值比较大).将排好序的流线做好已排序标记,依次加入已排序流线队列 Q_o 的头部,同时移出 Q_L .

Step2. 3. 如果 Q_L 中还有未排序流线,取出 Q_o 中的第一条流线 l_F ,令 $l_c=l_F$,转 Step2. 1.

Step2. 4. 取 Q_o 中的最后一条流线 l_L .令 $l_c=l_L$.

Step3. 对 Q_R 排序,处理过程与 Q_L 类似.

3.2 流线距离

本文使用流线之间的距离来描述流线的相似度.2 条流线之间的间距并不是处处一样的,使用 2 条流线一组对应点之间的距离来描述流线之间的相似程度.将排好序的流线转换到图像空间,在图像空间上按照流线长度均匀取点,从这些点引法流线,分别计算法流线和流线的交点距离,得到一组距离,对这些距离加权平均作为流线之间的距离,用于相似度的计算.

4 实验结果及分析

对流线进行聚类并对同一类的流线排序后,可以有不同的向量场可视化方法.下面分别介绍向量场的简洁表达、向量场的多分辨率均匀表达,以

及对基于纹理可视化结果增强等方面的应用.本文实验在 Intel® Core™ i7-4770K CPU@3.50 GHz, Windows 7(64 位)环境下完成,所有算法均并行处理.其中,生成完全流线集的计算时间和向量场大小成正比,海洋数据集耗时 5 s,其他数据集耗时均在 1 s 以内;流线聚类算法是最耗时的步骤,算法效率和数据集规模相关,海洋数据集耗时 20 s,其他数据集耗时不超过 2 s;排序算法的速度比较快,海洋数据集耗时 5 s,其他数据集耗时不超过 1 s.

4.1 基于流线的简洁表达

对向量场进行简洁表达是向量场可视化的一种重要方法.由于大数据的向量场流线比较多,如果过多流线同时显示,不容易突出向量场的重要区域,另外过多的流线会出现显示混乱的情况.本文使用基于流线聚类结果进行简洁方式表达,对聚类并排好序的每一个流线类,根据流线之间的距离选出中间位置的流线作为代表流线,流线的粗细由该流线类包含流线的数目决定,数量越多,流线越粗.图 2 所示为全球流场可视化,流场大小为 720×360 .图 2 a, 2 b 所示为基于箭头的可视化方法,分别使用了不同的分辨率表达;图 2 c 所示为基于流线的简洁表达,不同颜色的流线代表不同区域的流场,流线的粗细代表流场的相对强弱.本文算法的可视化效果要优于简单基于箭头的可视化效果.图 2 d 所示为基于流线的简洁表达与 LIC 算法融合的结果,是对基于纹理可视化方法的补充和增强.

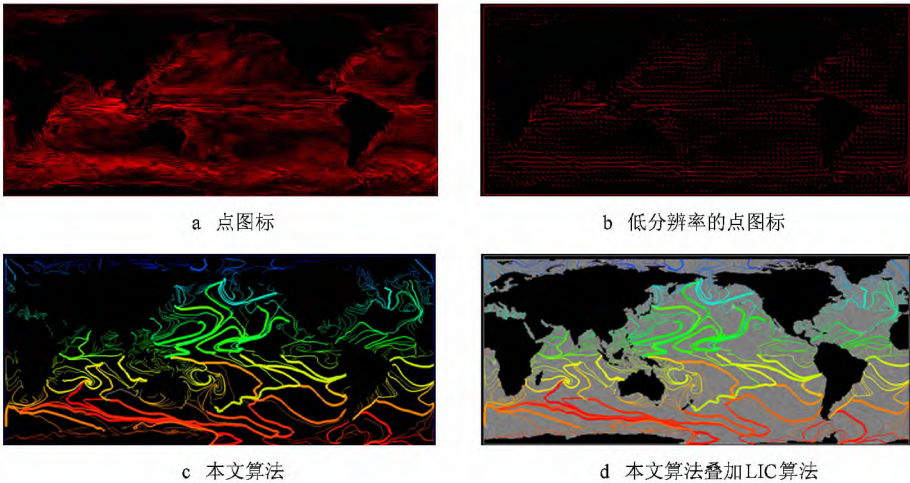


图 2 全球洋流可视化

4.2 向量场的均匀表达

向量场的均匀表达一直是许多研究人员感兴趣的工作^[6,14-15].一般认为,均匀放置的流线有助于用户快速获得整个向量场的流模式,而任意放置的流

线会导致用户的注意力分散.

本文对每一类流线进行排序,计算同一类中相邻流线之间的距离.因为流线相聚比较近,可以认为这些距离能反映向量场上流线的分布,所以可以根据

流线之间的距离,按照给定的流线间距进行流线的选取. 设给定相邻流线的间距为 d ,算法步骤如下:

- Step1. 将所有流线聚类.
- Step2. 对每一类流线排序. 下面以某个流线类 C 为例来放置均匀流线.
- Step3. 对于流线类 C ,设排序的流线为 $l_1, l_2, \cdots, l_n$, 计算相邻流线之间的距离 $d_1, d_2, \cdots, d_{n-1}, d_i = |l_i l_{i+1}|$; 然后计算每个流线 l_i 到 l_1 的累加距离 $D_1, D_2, \cdots, D_{n-1}, D_i = \sum_{k=1}^i d_k$.
- Step4. 计算所有的相邻流线距离 $s = D_{n-1}$. 计算流线数 $m = \lceil s/d + 0.5 \rceil$, 相邻流线的间距调整为 $d' = s/m$.
- Step5. $k = 0, i = 1$, 记录 l_1 .
- Step6. $D = d' \times 0.5 + d' \times k$.
- Step7. 如果 $D_i \geq D$, 记录 l_i , 执行下一步; 否则, $i = i +$

- 1, 继续执行本步.
- Step8. $k = k + 1$, 如果 $k < m$, 转 Step6.
- 因为流线已经排好序, 流线选择时只需要按照顺序对所有流线进行一次遍历. 如果总流线数为 n 条, 则共需要进行 n 次比较, 算法的时间复杂度为 $O(n)$, 可以进行实时的多分辨率表达.
- 流线生成算法的质量依赖于一致性、连续性和覆盖率. 对于连续性, 本文中过种子点生成的流线是完整流线, 完整流线是所有过该种子点的流线中最长的流线. 一致性和覆盖性并没有一个客观标准, 本文通过视觉效果对实验结果进行比较. 下面将本文工作与 Turk 等^[8]、Jobard 等^[6]、Mebarki 等^[7]、Verma 等^[2]的工作进行比较.

图 3 所示为不同分辨率的流线放置算法比较,

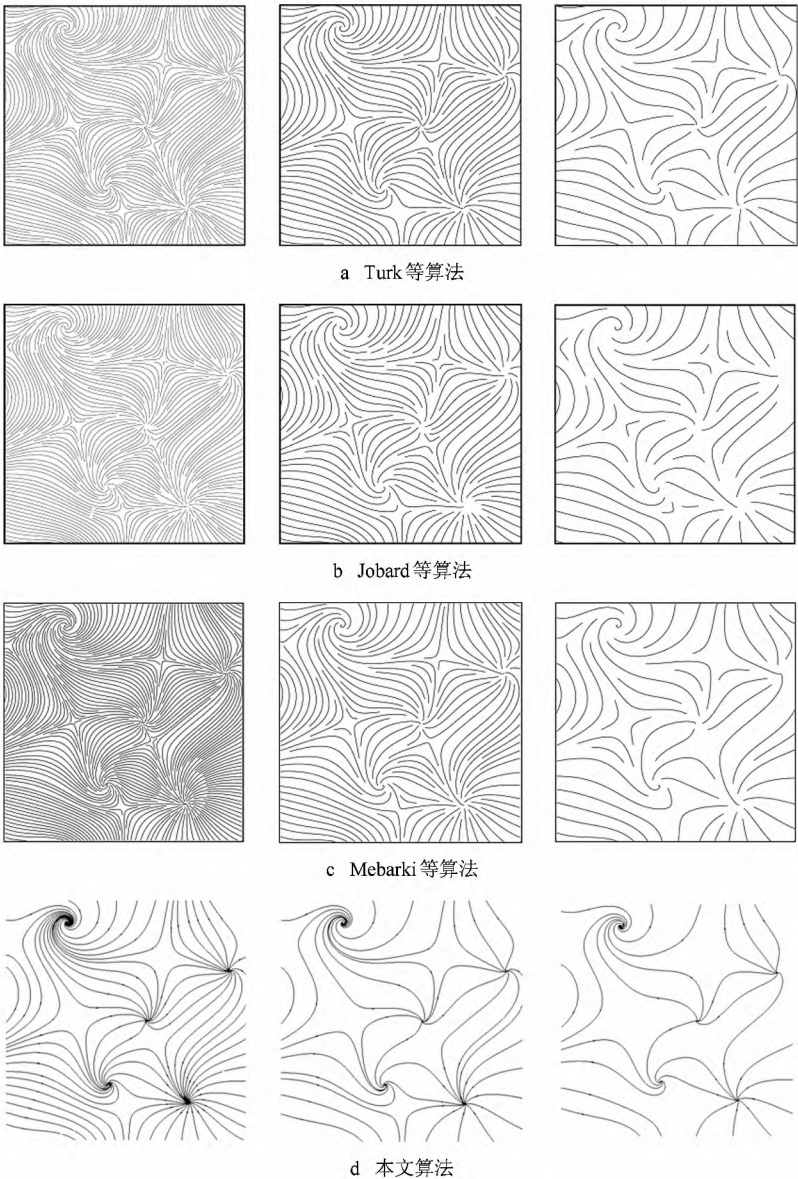


图 3 不同分辨率的流线放置算法比较

其中所用的向量场大小为 50×50 ; 每种算法从左向右使用的流线数目依次减少. 从图 3 可以看出, 关于连续性, 本文算法与 Jobard 等算法和 Turk 等算法差不多; 比 Mebarki 等算法更好一些, 它们的结果更倾向于在某些位置生成短的分开的流线, 而在实际的可视化中, 为了表达流的连贯性, 往往期望在这些位置放置一些连续的长的流线. 因为本文算法没有强制要求产生完全均匀的流线, 所以不会产生过短的流线, 引起中断的视觉效果. 图 3 d 中分别使用间距 1, 2, 3 得到 66, 36, 26 条流线. 本文算法使用较少的流线表达出较多的特征, 从图 3 可以看出, 本文

算法结果的流线放置更加均匀. 对比图 3 第 3 列可以看出, 在流线较少的情况下, 本文算法对鞍点特征的表达更好.

图 4 所示为本文算法和 Verma 等算法进行比较. 图 4 a 所示为 Verma 等算法的结果, 可以看出, 在较少流线的情况下, 很多的临界点并不能清楚地被标志, 尤其是鞍点位置展示得不够清晰. 图 4 b 所示为本文算法使用间距 1, 2 的结果, 可以看出, 在间距较大、流线数目较少的情况下, 临界区域依然比较清晰, 说明本文算法在连续性和一致性方面表现更好.

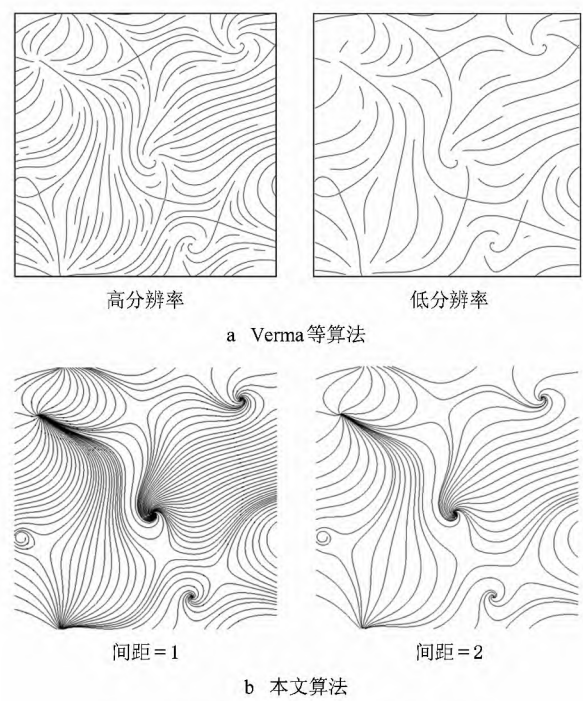


图 4 不同分辨率的流线放置算法比较

本文算法使用完全流线. 一般来说, 较长的流线是比较短的流线更有效的表示方法, 因为后者的间断性削弱了流线的印象. 实验结果也表明了这一点, 尤其在流线数目较少的情况下. 此外, 本文算法可以根据流线之间的间距实时地调节流线的数量, 这是其他方法所不能做到的.

4.3 对纹理可视化结果进行增强

使用纹理表达向量场的好处是可以描述全局的特征, 没有种子点选择的问题, 基于纹理的向量场表达不容易分清向量场不同区域的差别. 为了区分不同区域, 突出每个区域的主体部分, 本文生成一个权值矩阵, 对已有纹理进行加权增强. 给定 $m \times n$ 的向量场 V , 离散系数为 s , 生成 $(m \times s, n \times s)$ 的权值矩阵 M_w 和计数矩阵 M_c . 对于每个区域中的所有流线排

序, 计算相邻流线之间的距离; 根据距离选取中间的流线, 以此流线为基线计算每一条流线到基线的距离, 称为相对距离. 然后进行如下操作:

```
Step1.  $M_w$  和  $M_c$  所有元素清 0.
Step2. 对每一类中的每一条线  $l$ , 设其相对距离为  $d$ , 将  $l$  上的每一个点  $(x, y)$  变换到图像空间离散的  $(x', y')$ .
    for  $i: -k:k$ 
        for  $j: -k:k$ 
             $M_w(i+x', j+y') = M_w(i+x', j+y') * W(i+k, j+k) + F(d)$ ;
             $M_c(i+x', j+y') = M_c(i+x', j+y') + W(i+k, j+k)$ ;
        endfor
    endfor
Step3. 对于每一个  $(i, j)$ , 如果  $M_c(i, j)$  不为 0, 则
```

$M_w(i,j)=M_w(i,j)/M_c(i,j).$

其中, W 为 $(k\times 2+1,k\times 2+1)$ 的模板矩阵,若 k 过小,生成的权值矩阵可能有一些空洞;若 k 过大,平滑效果过强;本文取 $k=1$. $F()$ 为距离函数,本文使用 $F(d)=pow(d,c)$, pow 为幂函数, c 为幂的次数.图 5 所示为 c 分别取 0.3,2.0 所生成的权值图,可以看出, c 越大,不同类之间的对比越强烈,同时空隙也越大.

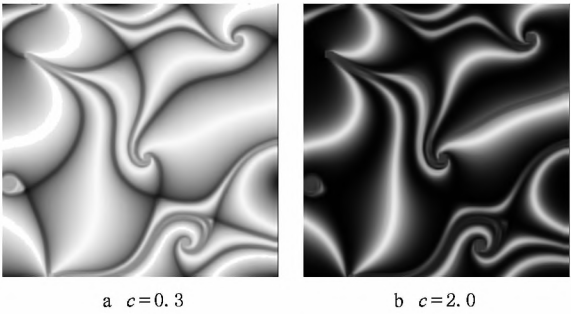


图 5 权值图

图 6 所示为对 LIC 算法生成的纹理进行增强的结果.图 6 中,第 1 行使用的数据和图 3 相同,第 2 行使用的数据和图 4 相同,第 3 行使用的数据大小为 128×128 .对于静态表达方式,本文算法的增强效果比 LIC 效果要好,能清晰地展示出流场不同区域的差别,图 6 中第 3 行的对比尤其明显,LIC 可视化结果几个区域的差别非常不明显.在使用动态表达方式时,本文采用适当的颜色映射,伪彩色增强效果也同样出色.

图 7 所示为海洋可视化效果的比较,向量场数据大小为 720×360 .对于这种比较大的向量场,LIC 算法的效果,尤其是静态表达效果不够清晰,如图 7 a 所示;图 7 b 所示为本文算法结果,可以看出,灰度增强效果把复杂向量场的主要部分呈现出来了;图 7 c 所示为伪彩色增强,该效果借鉴了梵高名画《星月夜》的色调,不但对向量场的相同区域和不同区域进行了增强,也引入了一些艺术的表达效果.

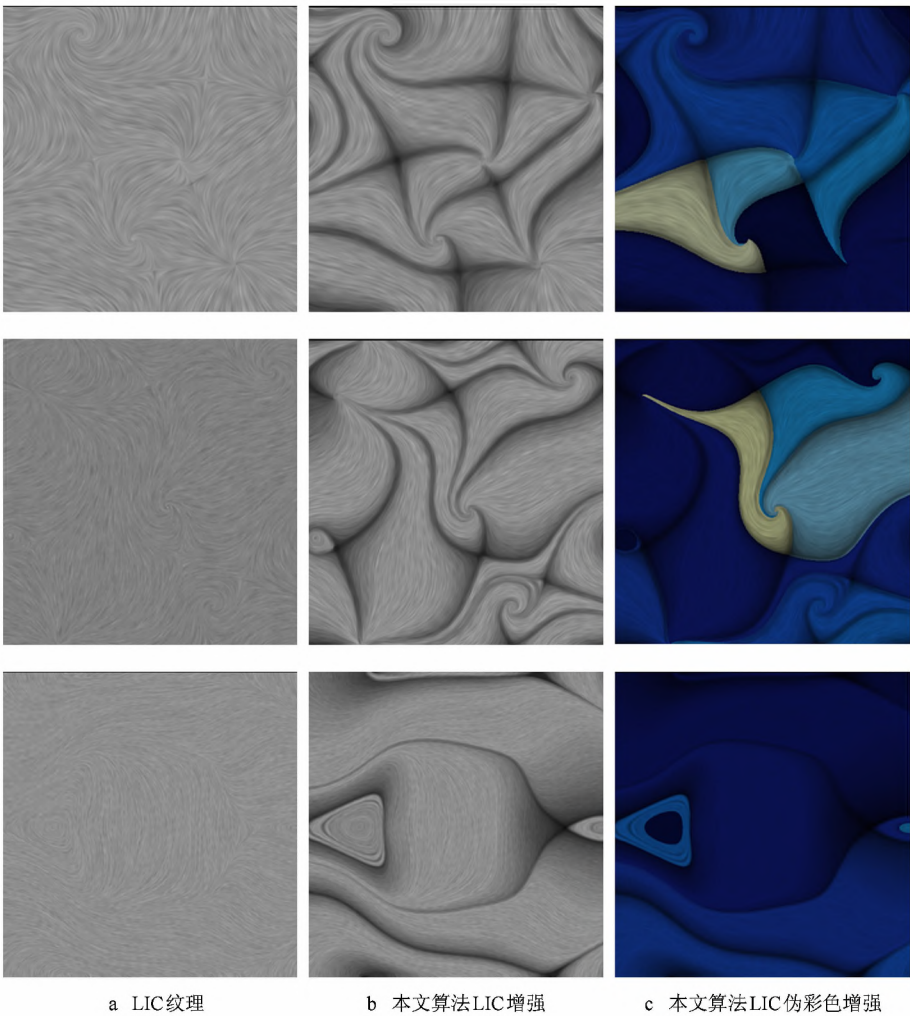


图 6 LIC 可视化和本文增强可视化的比较

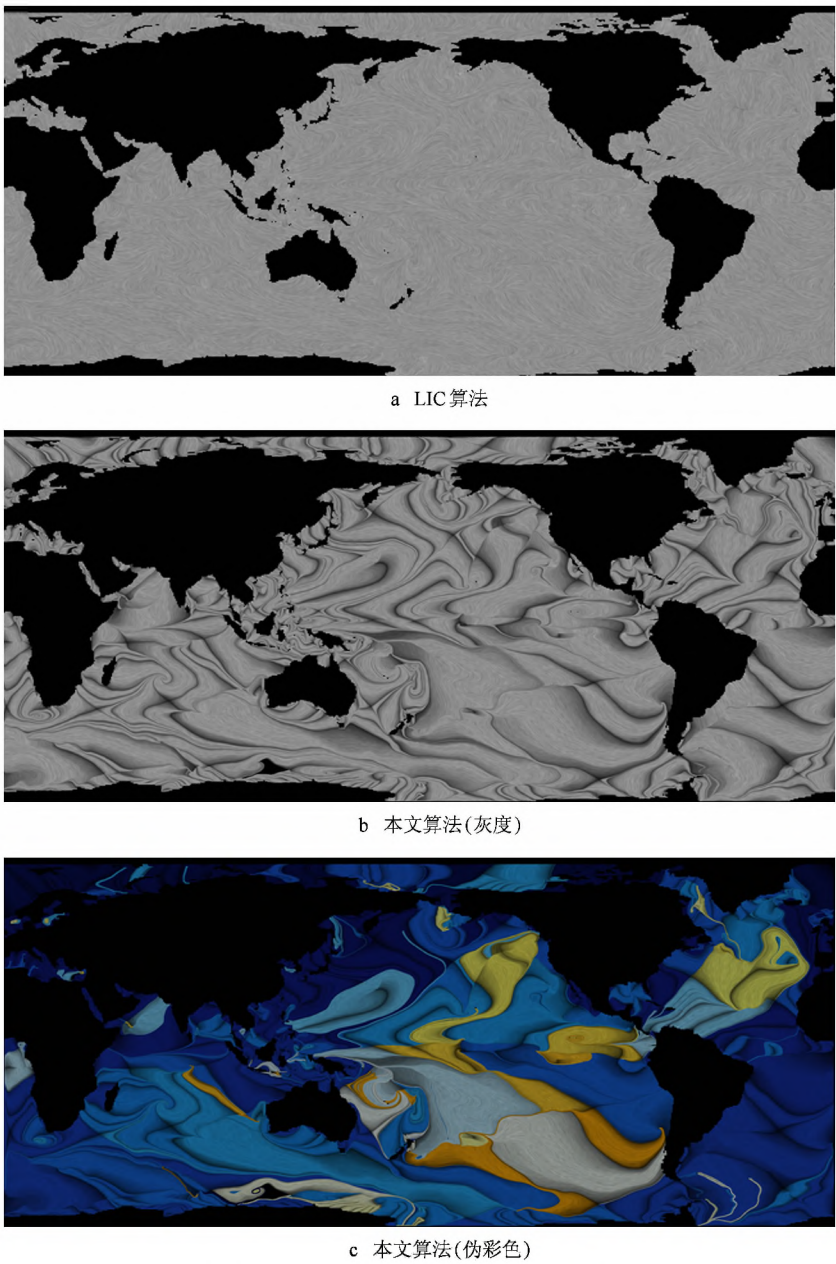


图 7 全球流场的 LIC 可视化和本文增强可视化的比较

5 结 语

本文对向量场中的流线进行聚类,将向量场分成几个流线类,然后对聚好类的流线按照它们的空间中的关系进行排序;对排好序的流线,抽取每一类的中线作为类代表流线,代表流线的宽度表示类的大小,流线类的流线数越多流线越宽.这样处理可以适用于复杂向量场的可视化;在向量场的均匀流线表达中,通过控制流线之间的距离可以实时获取符合给定间距的流线表达.由于本文算法强调每一个流线类的边界,因而边界围成的区域,尤其是鞍点区

域能非常清晰地表达出来;此外,根据流线之间的距离生成权值图,可以用于增强 LIC 可视化效果,灰度方式的增强适用于静态的纹理可视化,伪彩色方式的增强适用于动态的纹理可视化方式.

本文从一个新的角度提供了多种可视化方式.从实验结果可以看出,本文算法计算复杂度较高,对于更大规模的数据预处理算法需要进一步加速.

下一步计划将本文算法推广到三维向量场的可视化中,并更多地融合其他流场可视化算法,增强流线可视化的表现力;同时,如果流场流线中含有语义信息,则更利于流线聚类方法的高效使用.

参考文献(References):

- [1] Theisel H, Weinkauff T, Hege H C, *et al.* Saddle connectors—an approach to visualizing the topological skeleton of complex 3D vector fields [C] //Proceedings of the 14th IEEE Conference on Visualization. Los Alamitos: IEEE Computer Society Press, 2003: 225–232
- [2] Verma V, Kao D, Pang A. A flow-guided streamline seeding strategy [C] //Proceedings of the Conference on Visualization. Los Alamitos: IEEE Computer Society Press, 2000: 163–170
- [3] Chen Y, Cohen J, Krolík J. Similarity-guided streamline placement with error evaluation [J]. IEEE Transactions on Visualization and Computer Graphics, 2007, 13(6): 1448–1455
- [4] Shene C K, Wang C L, Yu H F, *et al.* Hierarchical streamline bundles [J]. IEEE Transactions on Visualization and Computer Graphics, 2012, 18(8): 1353–1367
- [5] Oeltze S, Lehmann D J, Kuhn A, *et al.* Blood flow clustering and applications in virtual stenting of intracranial aneurysms [J]. IEEE Transactions on Visualization and Computer Graphics, 2014, 20(5): 686–701
- [6] Jobard B, Lefer W. Creating evenly-spaced streamlines of arbitrary density [C] //Proceedings of the 8th Eurographics Workshop in Scientific Computing. Aire-la-Ville: Eurographics Association Press, 1997: 45–55
- [7] Mebarki A, Alliez P, Devillers O. Farthest point seeding for efficient placement of streamlines [C] //Proceedings of the IEEE Conference on Visualization. Los Alamitos: IEEE Computer Society Press, 2005: 479–486
- [8] Turk G, Banks D. Image-guided streamline placement [C] //Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM Press, 1996: 453–460
- [9] van Wijk J J. Spot noise texture synthesis for data visualization [J]. ACM SIGGRAPH Computer Graphics, 1991, 25(4): 309–318
- [10] Cabral B, Leedom L C. Imaging vector fields using line integral convolution [C] //Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM Press, 1993: 263–270
- [11] Heckel B, Weber G, Hamann B, *et al.* Construction of vector field hierarchies [C] //Proceedings of the Conference on Visualization. Los Alamitos: IEEE Computer Society Press, 1999: 19–25
- [12] Telea A, van Wijk J J. Simplified representation of vector fields [C] //Proceedings of the Conference on Visualization. Los Alamitos: IEEE Computer Society Press, 1999: 35–42
- [13] Du Q, Wang X Q. Centroidal Voronoi tessellation based algorithms for vector fields visualization and segmentation [C] //Proceedings of the Conference on Visualization. Los Alamitos: IEEE Computer Society Press, 2004: 43–50
- [14] Liu Z P, Moorhead R J, Groner J. An advanced evenly-spaced streamline placement algorithm [J]. IEEE Transactions on Visualization and Computer Graphics, 2006, 12(5): 965–972
- [15] Wu K Q, Liu Z P, Zhang S, *et al.* Topology-aware evenly spaced streamline placement [J]. IEEE Transactions on Visualization and Computer Graphics, 2010, 16(5): 791–801

作者: 王少荣, 汪国平, Wang Shaorong, Wang Guoping
作者单位: 王少荣, Wang Shaorong(北京林业大学信息学院 北京 100083;北京市虚拟仿真与可视化工程技术研究中心 北京 100871), 汪国平, Wang Guoping(北京市虚拟仿真与可视化工程技术研究中心 北京 100871;北京大学图形与交互技术实验室 北京100871)
刊名: 计算机辅助设计与图形学学报 ISTIC EI PKU
英文刊名: Journal of Computer-Aided Design & Computer Graphics
年, 卷(期): 2014, 26(10)

参考文献(15条)

1. Theisel H;Weinkauff T;Hege H C Saddle connectors an approach to visualizing the topological skeleton of complex 3D vector fields 2003

2. Verma V;Kao D;Pang A A flow-guided streamline seeding strategy 2000

3. Chen Y;Cohen J;Krolik J Similarity guided streamline placement with error evaluation 2007(06)

4. Shene C K;Wang C L;Yu H F Hierarchical streamline bundles 2012(08)

5. Oeltze S;Lehmann D J;Kuhn A Blood flow clustering and applications in virtual stenting of intracranial aneurysms 2014(05)

6. Jobard B;Lefer W Creating evenly-spaced streamlines of arbitrary density 1997

7. Mebarki A;Alliez P;Devillers O Farthest point seeding for efficient placement of streamlines 2005

8. Turk G;Banks D Image-guided streamline placement 1996

9. van Wijk J J Spot noise texture synthesis for data visualization 1991(04)

10. Cabral B;Leedom L C Imaging vector fields using line integral convolution 1993

11. HeckelB;WeberG;HamannB Construction of vector field hierarchies 1999

12. Telea A;van Wijk J J Simplified representation of vector fields 1999

13. Du Q;Wang X Q Centroidal Voronoi tessellation based algorithms for vector fields visualization and segmentation 2004

14. Liu Z P;Moorhead R J;Groner J An advanced evenly spaced streamline placement algorithm 2006(05)

15. Wu K Q;Liu Z P;Zhang S Topology-aware evenly spaced streamline placement 2010(05)

引用本文格式: 王少荣. 汪国平. Wang Shaorong. Wang Guoping 基于聚类的二维向量场可视化[期刊论文]-计算机辅助设计与图形学学报 2014(10)