

文章编号:1007-5321(2019)06-0091-07

DOI:10.13190/j.jbupt.2019-152

一种数据驱动的三维流场流线特征化筛选方法

熊光正¹, 黄智濒¹, 戴志涛¹, 杨武兵²

(1. 北京邮电大学 智能通信软件与多媒体北京市重点实验室, 北京 100876;

2. 中国航天空气动力技术研究院, 北京 100074)

摘要: 传统的流线可视化方法因视线遮挡和数据密集难以刻画流场特征, 难以应对大规模数据, 为此, 从数据驱动的思路出发, 提出了一种筛选三维流线的算法, 实现对大规模精细流场的特征刻画. 该算法对广泛撒点取得的流线集进行特征化, 通过计算流线上各点的特征, 并以此为依据对流线进行分段; 基于所有分段的几何特征构建一组特征向量, 并利用词袋方法建立一组词向量; 以词向量为基础计算流线间的几何特征相似度, 以评估各个流线间的相似性, 实现对流线的筛选. 通过在特定流线的查询和整体流线流场的压缩这 2 个典型应用场景上的应用, 检验了该方法对流线筛选的效果.

关 键 词: 数据驱动; 流线分段; 流场可视化; 词袋; 大规模精细流场

中图分类号: TP391.41

文献标志码: A



A Data Driven Characteristically Filtering Method for 3D Flow Field

XIONG Guang-zheng¹, HUANG Zhi-bin¹, DAI Zhi-tao¹, YANG Wu-bing²

(1. Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia, Beijing 100876, China;

2. China Academy of Aerospace Aerodynamics, Beijing 100074, China)

Abstract: With the wide application of fluid mechanics, more and more large-scale fine flow fields have emerged. Overlapping streamlines and dense fields make it hard to use the traditional streamline visualization methods to characterizes the flow fields or process with large-scale fine flow fields. Based on the idea of data-driven, this paper presents an algorithm to implement the characterization of large-scale fine flow fields. The algorithm characterizes streamlines obtained by widely spreading seed points, calculates the features of each point, segments the streamlines based on the features, and then constructs a set of feature vectors and a set of word vectors. Then, the algorithm calculates the geometric feature similarity between streamlines to evaluate streamline similarity and achieves streamlines filtering. Two typical application scenarios, streamline query and flow field compression, verify the proposed method.

Key words: data driven; streamline segmentation; flow field visualization; bag-of-word; large-scale fine flow field

三维流场数据被广泛应用于气象、航空航天等科学计算及工程应用领域, 其具有较大的规模和精细的结构. 流线是流场数据的直观表现, 分析人员

借助流场可视化直观地观察流场. 基于流线的几何可视化方法在各类流场可视化场景得到了广泛的应用.

收稿日期: 2019-07-10

基金项目: 2019 年中央基本业务费项目(RC201958)

作者简介: 熊光正(1999—), 男, 硕士生.

通信作者: 黄智濒(1978—), 男, 讲师, E-mail: huangzb@bupt.edu.cn.

基于流线的可视化方法使用^[1]“撒点—追踪—可视化”三步走策略,先在流场中选取若干种子点作为流线追踪的起点,再使用流线追踪算法获取流线,最后在流场空间中绘制出每条流线,既能展示每条流线的数据,又能展现流场随时间的变化.随着计算机模拟技术的发展,流场的规模不断攀升,其精细化程度亦稳步提高.对于大规模精细流场数据,若选取的种子点过于稀疏,则追踪所得的流线往往不足以完整描述流场的细节;若撒点过于密集,则生成的流线过多,可能引发流线重叠、流线遮挡等问题,前者引入了冗余的流线信息,后者对分析人员造成了干扰,二者均降低了流场图像的质量.

为了得到清晰、精确的流场图像,可以引入数据驱动的概念,通过广泛撒点、对追踪所得的流线进行特征化处理,并量化流线间关系,实现流线的查询和流场的压缩.基于流线上各点的特征,对流线进行分段,再引入嵌入算法,将分段嵌入成词向量.基于词向量在空间中的距离刻画流线间的距离,基于特征考虑流线内各点、各分段间的相互关系.本算法结合流线的空间位置关系和流线内各个分段的距离,将流场嵌入至特征词向量,借此提高后续任务的性能.

所述算法是一种新颖的流线特征化筛选算法,基于流线的几何特征将流线分为若干段单一特征的流线分段,基于流线上各点的特征值将流线分段嵌入至分段特征向量,使用主成分分析方法对分段特征向量进行降维,并使用 K 均值算法对降维后的分段特征向量进行聚类.算法引入词袋模型,将流线上各对分段视为表达式,基于各表达式的空间特性对其赋权,将各表达式加权累加以计算流线的特征词向量.算法基于TF-IDF算法对各表达式赋权,并以此权重计算流线间的相似度.最后,算法引入贪心法,计算流场内的主要流线以压缩流场,改善流场可视化的效果.

1 相关研究

早期研究着重于选取种子点.为了描述撒点策略,Ye等^[2]引入撒点模板以表述目标周围种子点的分布.关键点的位置显著影响流场的特征,其基于关键点间的位置关系使用不同的撒点模板来优化图像.Edmunds^[3]对流场中的向量进行聚类,分析各点流向的结构,据此建立撒点模板.郭雨蒙等^[4]提

出了流线相似度引导种子点分布的并行处理方法.Marchesin等^[5]使用流线的几何特征指定流线分布.李春鑫等^[6]引入基于支持向量机的预处理模型来控制流线的疏密.然而,这样的策略要求撒点前掌握流场的特征,不适用于精细流场.Li等^[7]引入人工输入来区分主次流线.流线可能经过具有不同特性的区域,故引入分割点的概念,将不同特性的流线切分成两段分段.用户指出分割点作为正样本,使用支持向量机等学习用户分类,从而有监督地分类流线分段,最后获取完整流线的特性.Yu等^[8]建立流线集合的层次化结构,在不同尺度下选择流线,基于流线间的位置关系自底向上地建立层次化的流线间结构.

借此,引入数据驱动的概念,通过广泛撒点获取大量流线数据,用有监督或无监督的策略学习流线的特征,从而建立流线间的关系网络,进而对流线进行筛选、获取高质量的流场图像.此类算法通过计算流线上各点的空间关系,如最短距离、最长距离、平均距离或狗绳距离等距离评估流线对的相似度.

基于流线的几何特征可以得出流线间的相似度.Ross等^[9]提出将流线转化为向量,计算向量间的相似度,来计算流线间的相似度.Liu等^[10]考虑流线的子结构,将流线视为子结构的集合,将子结构视为字符,流线即为字符串,计算字符串间的相似度即可得到流线间的相似度.Li等^[11]引入词袋模型来计算流线间的相似度.Han等^[12]基于卷积神经网络实现流场数据的压缩.Wang等^[13]提出基于输入的流线搜索方法.Li等^[14]基于点的特征向量,采用词袋模型建立流线的特征向量.

为了提高计算速度,Tang等^[15]引入并行计算来加速计算.其在三维流线相似度模型下,并行撒点,计算相似度,提高了整体计算效率.

深度学习技术亦对流线表示、查询和流场压缩有所帮助.Han等^[16]通过两阶段深度学习方法实现流场的压缩.其对各点的速度进行插值后,建立低分辨率的流场,再模仿图片超分辨率算法,用卷积神经网络重建高分辨率流场.Han等^[17]提出基于自动编码器的流线特征提取模型,其FlowNet框架使用自动编码器获取流线低维特征向量,据此计算流线间相似度.图1所示为后续流程的整体框图,其中虚线框内的最后一步见2.1节.

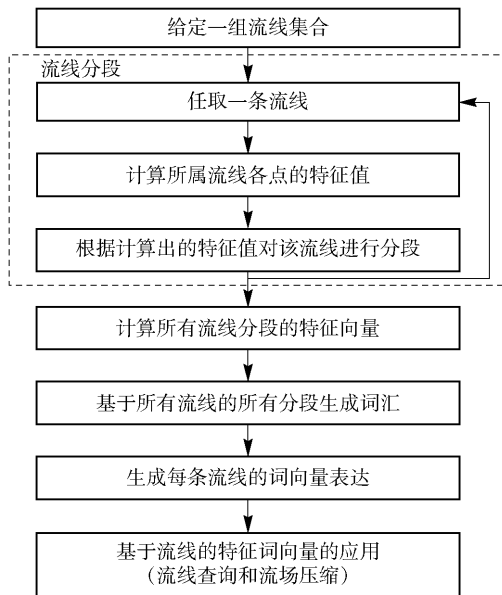


图1 数据驱动的三维流场流线特征化筛选方法

2 数据驱动的三维流场流线特征化筛选方法

2.1 流线分段

2.1.1 计算流线各点的特征值

使用4个特征描述流场中的流线和流线上的点。对于每一个特征,计算各点的特征值,进而计算该特征的特征向量,最后将各个特征的向量与流线的规模向量相连,得到流线的特征向量。

曲率描述流线的弯曲程度。记 x', y' 为 x 和 y 的一阶导数, x'', y'' 为 x 和 y 的二阶导数,有

$$\kappa = \frac{|x'(t)y''(t) - x''(t)y'(t)|}{(x'^2(t) + y'^2(t))^{\frac{3}{2}}} \quad (1)$$

挠率描述了空间曲线远离平面的程度。记 x', y' 为 x 和 y 的一阶导数, x'', y'' 为 x 和 y 的二阶导数, x''', y''' 为 x 和 y 的三阶导数。计算公式如下:

$$\tau = \frac{x'''(y'z'' - y''z') + y'''(x'z'' - x''z') + z'''(x'y'' - x''y')}{(y'z'' - y''z')^2 + (x'z'' - x''z')^2 + (x'y'' - x''y')^2} \quad (2)$$

曲折度描述曲线偏离直线的程度,其中 l 为流线的长度, S 和 E 分别为流线的起始端点和终止端点的坐标,故 $\|S - E\|$ 为其直线距离。有

$$t = \frac{l}{\|S - E\|} \quad (3)$$

速度方向熵表征流线方向的变化。在空间球坐标系中,流线速度的方向为一单位向量,其与单位球上的点一一对应。将单位球分成 M 个面积相等的

曲面,则每个方向映射到一个曲面。为曲面编号,则速度方向可用编号表示,流线速度方向可用正整数序列表示,该序列的熵为流线速度方向熵。序列中的各个正整数为信号,记 p_i 为各采样值的频率,则有

$$e = - \sum_{i=1}^M p_i \log(p_i) \quad (4)$$

2.1.2 根据特征值对流线进行分段

平直流线的特征弱,而曲折流线具有显著的特性,其特征与曲率的变化关系紧密,故基于曲率进行分段。该方法关注流线上曲率较大、变化率较大的位置,如受流场影响而剧烈弯曲的部分及其邻域。

曲率较大,且为曲率极值点的点,即潜在的分段中点。将流线的端点加入潜在的分段中点序列,遍历各个潜在的分段中点。以其曲率加权,计算相邻潜在的分段中点间的加权中点,即为潜在的分段点。

设 a 和 b 为一对相邻潜在的分段中点, $d(a)$ 和 $d(b)$ 分别为其距流线起始端点的距离,且 $d(a) < d(b)$,则对应潜在分段点距流线起点的距离 $d(s_{ab})$ 为

$$d(s_{ab}) = d(a) + (d(b) - d(a))\alpha + (d(b) - d(a)) \times (1 - \alpha) \frac{C(a)}{C(a) + C(b)} \quad (5)$$

其中 α 为一个0~1的常数,描述允许被用于分段的距离占相邻潜在分段中点距离的比重。

基于曲率的变化率计算潜在的分段点,将二者所得的潜在的分段点序列合并,得到最终的序列。

将流线的起终点加入最终的潜在分段点序列,并从流线的起点开始遍历序列。以流线的起点为初始分段起点 s ,以序列中的第2个点为初始分段终点 e ,遍历各点,若分段起点与分段终点间的距离 $d(e) - d(s)$ 大于最小分段长度阈值,则将分段 (s, e) 插入序列,并依次将分段起点 s 设为该分段终点 e ,再将分段终点 e 设为新的分段起点 s 的后一个潜在的分段点,直至遍历完所有潜在的分段点。若遍历至流线终止端点时,当前分段的长度 $d(e) - d(s)$ 依然小于最小分段长度阈值,则将其与分段序列中的最后一个分段合并。

2.2 计算流线分段的特征向量

引入金字塔模型以计算流线分段的特征向量。将各分段上各特征点的特征值自流线起始端点方向至流线终止端点方向排列,形成分段的几何特征向量 (x_1, x_2, \dots, x_l) 。将该几何特征向量作为底层,依

照规则 $x_i^j = \frac{x_i^{j-1} + x_{i+1}^{j-1}}{2}$ 迭代计算其上一层的几何特征向量, 即 $(x_1^1, x_2^1, \dots, x_{l-1}^1), (x_1^2, x_2^2, \dots, x_{l-2}^2), \dots, (x_1^j, x_2^j, \dots, x_{l-j}^j), \dots, (x_1^{l-1})$. 最后, 建立完整的几何特征金字塔结构.

自上至下, 自左至右依次选取 N 个元素, 将其顺序排列, 形成对应分段的几何特征向量 $(x_1', x_2', \dots, x_N')$. 依次计算各个特征的几何特征向量, 并计算分段的规模特征向量 (l, l_d) , 将其连接, 形成分段的特征向量.

2.3 基于流线分段生成词汇

基于词袋模型, 使用词汇表示流线中各分段的特征向量, 进而表现流线的特征. Li 等^[14] 提出了基于点的特征向量建立流线的词袋模型, 但其使用的点的特征数较少, 故其词袋模型的词汇数量相对有限, 降低了模型的性能. 而所述模型基于流线分段建立词袋模型, 故词汇更多, 性能亦更高.

将上述流线分段的特征向量分别映射至一个词汇, 引入聚类映射. 考虑到流线分段的特征向量维度较高, 常见的聚类算法难以对其准确聚类, 在聚类前先对流线分段的特征向量执行主成分分析, 将其降维至 K 维低维空间, 再进一步聚类. 将 K 维特征向量使用 K -Means 聚集到 V 个聚类中心上.

2.4 生成流线的词向量表达

引入与每个词汇一一对应的独热向量, 其长度即为词汇集合的大小 V . 取定 $V=32$, 采用算法 1 计算流线的词向量表达.

算法 1 计算流线的词向量表达

$S := 0$ // 初始化流线维特征词向量为零向量

For streamline s in dataset:

For segment a in streamline:

$\theta_a :=$ 流线分段 a 的独热向量

For segment b in streamline:

If $a \neq b$:

$\theta_b :=$ 流线分段 b 的独热向量

$e_i = \theta_a \times \theta_b^T$ // 表达式 i 的特征词向量

$w_i = \frac{|d_a - d_b|}{l}$ // 表达式 i 的权重

$S := S + w_i e_i$ // 加权累加表达式的特征词向量

对每一个流线分段, 使用其词汇对应的独热向量 θ 表示其特征. 对于每条流线, 计算其空间敏感的词袋特征词向量 B . 在该模型下, 将流线上的一

对分段称为一个表达式, 考虑到本模型中共有 V 个不同的词汇, 每个表达式均对应 2 个流线分段, 即 2 个词汇, 故本模型中共存在 $V \times V = 32 \times 32 = 1\,024$ 种不同的有序词汇组合, 因此, 在实验中, 使用大小为 $V \times V = 1\,024$ 维的词向量 E 描述表达式的特征, 即

$$e_i = \theta_a \times \theta_b^T \quad (6)$$

其中 a 和 b 分别表示表达式中的 2 个流线分段. θ_a 与 θ_b 分别为表达式中 2 个流线分段对应的独热向量. 各个不同的词汇组合与 e_i 中的不同元素一一对应.

依据各对流线分段的空间距离为各表达式的特征词向量赋权. 将流线中各分段上沿流线长度的中点定义为各分段的中点, 考虑到分段中点距离较远的表达式对流线的特征具有更大的影响, 为分段中点距离较远的表达式赋予更大的权重, 将各分段的中点距其所属流线起始端点的距离称为分段的位置 d_i , 将各表达式的权重定义为

$$w_i = \frac{|d_a - d_b|}{l} \quad (7)$$

其中: l 为流线的长度, d_a 和 d_b 分别为该表达式对应分段的位置.

将各表达式的特征词向量加权后按位相加, 以计算流线 X 的特征词向量, 即

$$B = \sum_{a \in X} \sum_{b \in Z} w_i e_i \quad (8)$$

2.5 基于特征流线的词向量的应用

2.5.1 流线查询

传统流线查询方法依赖传统的流线相似度算法, 考虑流线间的空间位置关系, 如最近距离、各点平均距离等, 只能用于查询目标流线附近的流线, 而不能查找距离目标流线较远, 但形状与目标流线类似的流线, 故可基于流线特征化方法的流线查询算法.

以 TF-IDF (term frequency-inverse document frequency) 模型为各表达式赋权. 采用空间敏感的词袋模型, 着重考虑表达式出现与否, 并为其每次出现赋权, 而非直接计算句子中表达式出现的数量. 因此, 简化 TF-IDF 模型, 以含有各词汇的流线占流线总数的比例作为各词汇的频率 f_i . 在本模型中, 以频率对数的绝对值做为各个表达式的权重, 即

$$W_i = \log \frac{1}{f_i} \quad (9)$$

结合表达式的权重比较流线的相似度. 与上

述流线的特征词向量一致,将各表达式的权重以相同的方式排列成长度为 $V \times V = 1\,014$ 维的词向量 W ,则流线对 X, Y 的不相似度词向量为

$$D(X, Y) = W |B(X) - B(Y)| \quad (10)$$

该不相似度词向量上各元素的绝对值即为该流线上各表达式对的不相似度,差异越大,其特征词向量之差的绝对值越大. 考虑到不相似度词向量的对称性,将流线对的不相似度标量定义为

$$d(X, Y) = \sum_i \frac{V(V+1)}{2} W_i |B(X)_i - B(Y)_i| \quad (11)$$

2.5.2 流场压缩

移除流场中的部分流线,不会显著影响流场的特征,而移除其他流线则会影响流场的特征. 反之,加入特定的流线可显著增加与原始流场的相似度. 反复加入流线即可得与原始流场相似度依次递增的流场,此即一组压缩流场.

引入贪心算法,向空流场加入最能提高压缩流线集内各流线间不相似度的流线. 将压缩流场每对流线的相似度的和称为压缩流线集的相似度,将目标压缩流场的流线条数称为目标流场大小,将不在压缩流线集中而在流场中的流线称为外部流线.

算法2 压缩流线集合

$S := \emptyset$ // 初始化压缩流线集合为空集

$\text{MaxDissim} := 0$ // 初始化最大不相似度为 0

For streamline x in dataset:

For streamline y in S :

If $\text{Dissim}(x, y) > \text{MaxSim}$:

$\text{MaxSim} := \text{Dissim}(x, y)$

$S = \{x, y\}$ // 将压缩流线集视为 $\{x, y\}$

While $\text{card}(S) < N$:

$\text{TmpLine} := \text{None}$ // 初始化当前最不相似流线

$\text{MaxDissim} := 0$ // 初始化最大不相似度为 0

For streamline x in dataset:

$\text{TmpDissim} := 0$ // 初始化不相似度为 0

For streamline y in S :

$\text{TmpDissim} := \text{TmpDissim} + \text{Dissim}(x, y)$

If $\text{TmpDisism} > \text{MaxDissim}$:

$\text{TmpLine} := x$

$S := S \cup \{x\}$ // 将 x 加入 S

3 实验

3.1 数据集预处理

采用 SLVG 数据集、IEEE Visualization 2004

Contest 数据集、carotid 数据集和 velocityMag 数据集,以检验不同场景下的性能.

3.1.1 SLVG 数据集

SLVG 数据集由 2 个的随机生成的流场组成,即 108CP 流场和 5CP 流场,其分别包含 108 个临界点和 5 个临界点. 各采样点组成立方体,其每条边均由 51 个采样点组成,即 $51 \times 51 \times 51$ 个采样点.

3.1.2 IEEE Visualization 2004 Contest 数据集

IEEE Visualization 2004 Contest 数据集为计算机模拟的龙卷风,包含 40 个时间片、 $500 \times 500 \times 100$ 个采样点,覆盖 $2\,139 \times 2\,004 \times 19.8 \text{ km}^3$ 的空间.

3.1.3 carotid 数据集

carotid 数据集的采样点分布在 $76 \times 49 \times 45$ 的空间中,共包含 174 195 个采样点. 本数据集富含漩涡,其流线的曲率大,流场十分复杂.

3.1.4 velocityMag 数据集

velocityMag 数据集的采样点分布在 $57 \times 33 \times 25$ 的空间中,共 47 025 个采样点. 其流线分布在一个不规则三维空间内,流线弯曲而可见明显的临界点.

3.2 实验结果及分析

在各个测试数据集上执行所述算法,观察经管道化后的流线特征.

3.2.1 流线查询

选定一条流线作为目标流线,在流场中查找与其最相似的若干条流线. 在本实验中,查找最相似的 10 条流线,实验结果如表 1 所示. 第 1 组样例取自 carotid 数据集,算法准确地查询到了图形中部环绕关键点的 10 条相似流线;第 2 组样例取自 carotid 数据集,算法查询到了以形状相似的路径环绕各个关键点的 10 条流线;第 3 组样例取自 SLVG 数据集,考虑到流线中下部存在数个弯曲部分,故流线上部的不同的弯曲方向将对应不用的流线特征,依据选取的流线,算法成功查询到了 10 条弯曲方向类似的流线;第 4 组样例取自 SLVG 数据集,与第 3 组样例类似,算法查询到的 10 条流线不仅仅是弯曲方向类似的流线,还是弯曲程度最接近目标流线的流线;第 5 组样例取自 velocityMag 数据集,算法查询到了依次呈现相似的弯折段的 10 条流线;第 6 组样例取自 velocityMag 数据集,与第 5 组类似,算法查询到了与目标流线形状类似的 10 条流线.

表 1 流线查询的实验结果

编号	完整流场	目标流线	相似流线
1			
2			
3			
4			
5			
6			

3.2.2 流场压缩

使用上述流场压缩算法,依次构建大小为 64、32、16、8 线的压缩流线集. 流场压缩的实验结果如表 2 所示. 第 1 组样例取自 SLVG 数据集,算法组合了流经不同路径的流线,在尽可能保留环等流场特征的情况下完成了流场压缩的任务;第 2 组样例取自 SLVG 数据集,算法组合了不同位置成环的流线,准确完成了流线压缩任务;第 3 组样例取自 SLVG

表 2 流场压缩的实验结果

编号	原始流线集	压缩流线集			
		64线	32线	16线	8线
1					
2					
3					
4					
5					
6					

数据集,算法区分了走向不同方向的流线,在压缩流场的同时保持了流线上分布的特征;第 4 组样取自 SLVG 数据集,算法保留了顶部漩涡状的结构;第 5 组样例取自 SLVG 数据集,算法分别保留了顶部的弯折结构、中部的流面和底部的流管特征;第 6 组样例取自 IEEE Visualization 2004 Contest 数据集,算法保留了弯曲程度不同的流线,保持了龙卷风的特征.

4 结束语

所述算法采用数据驱动的策略,使用在三维流场中广泛撒点获取的流线集合,基于几何特征和空间特征将所得流线特征化后,通过加权后的流线特征计算流线间的相似度,继而准确完成了流线查询和流场压缩的任务.

参考文献:

[1] McLoughlin T, Laramee R S, Peikert R, et al. Over two decades of integration-based, geometric flow visualization [J]. Computer Graphics Forum, 2010, 29(6): 1807-1829.

[2] Ye Xiangong, Kao David, Pang Alex. Strategy for seeding 3D streamlines[C]//IEEE Visualization. Minneapolis: [s. n.], 2005: 471-478.

[3] Edmunds M, Laramee R S, Malki R, et al. Automatic stream surface seeding: a feature centered approach[J]. Computer Graphics Forum, 2012, 31(3pt2): 1095-1104.

[4] 郭雨蒙, 王文珂, 李思昆. 基于相似度引导的流线种子点并行分布方法[J]. 系统仿真学报, 2014, 26(9): 2155-2159.

Guo Yumeng, Wang Wenke, Li Sikun. Parallel strategy of streamline seeding with similarity-guiding [J]. Journal of System Simulation, 2014, 26(9): 2155-2159.

[5] Marchesin S, Chen C K, Ho C, et al. View-dependent streamlines for 3D vector fields[J]. IEEE Transactions on Visualization and Computer Graphics, 2010, 16(6): 1578-1586.

[6] 李春鑫, 彭认灿, 高占胜, 等. 一种改进的三维流场数据可视化方法[J]. 武汉大学学报(信息科学版), 2017, 42(6): 744-748.

Li Chunxin, Peng Rencan, Gao Zhansheng, et al. An improved legible method for visualizing 3D flow field[J]. Geomatics and Information Science of Wuhan University, 2017, 42(6): 744-748.

[7] Li Yifei, Wang Chaoli, Shene C K. Extracting flow fea-

- tures via supervised streamline segmentation[J]. Computers & Graphics, 2015(52): 79-92.
- [8] Yu H, Wang C, Shene C K, et al. Hierarchical streamline bundles for visualizing 2D flow fields[C]//IEEE VisWeek 2010 Posters. Salt Lake City: [s. n.], 2010.
- [9] Ross C, Theisel H. Streamline embedding for 3D vector field exploration[J]. IEEE Transactions on Visualization and Computer Graphics, 2012, 18(3): 407-420.
- [10] Liu Shiguang, Song Hange. Streamline querying based on finite substructures[J]. Journal of Visualization, 2019, 22(3): 571-585.
- [11] Li Yifei. Extracting flow features using bag-of-features and supervised learning techniques[D]. Michigan: Open Access Dissertation, Michigan Technological University, 2015.
- [12] Han Jun, Tao Jun, Zheng Hao, et al. Flow field reduction via reconstructing vector data from 3D streamlines using deep learning[J]. IEEE Computer Graphics and Applications, 2019, 39(4): 54-67.
- [13] Wang Zhongjie, Martinez Esturo J, Seidel H P, et al. Pattern search in flows based on similarity of stream line segments[C]//Proc Vision, Modeling and Visualization. Darmstadt, Germany: [s. n.], 2014: 23-30.
- [14] Li Yifei, Wang Chaoli, Shene C K. Streamline similarity analysis using bag-of-features[C]//Proc SPIE 9017, Visualization and Data Analysis 2014. San Francisco, California: [s. n.], 2014: 90170N.
- [15] Tang Bin. Parallel visualization of flow field based on streamline similarity[J]. International Journal of Perfromability Engineering, 2018, 14(4): 717-728.
- [16] Han Jun, Tao Jun, Zheng Hao, et al. Flow field reduction via reconstructing vector data from 3D streamlines using deep learning[J]. IEEE Computer Graphics and Applications, 2019, 39(4): 54-67.
- [17] Han Jun, Tao Jun, Wang Chaoli. FlowNet: a deep learning framework for clustering and selection of streamlines and stream surfaces[EB/OL]. 2018(2018-11-12) [2019-05-23]. <https://doi.org/10.1109/TVCG.2018.2880207>.