

中图法分类号: TP391.41 文献标志码: A 文章编号: 1006-8961(2011)06-1064-06

论文索引信息: 孙长会, 范茵, 李骞, 黄伟. 增强型 IBFV 2 维矢量场可视化算法 [J]. 中国图象图形学报, 2011, 16(6): 1064-1069

增强型 IBFV 2 维矢量场可视化算法

孙长会, 范茵, 李骞, 黄伟

(解放军理工大学气象学院, 南京 211101)

摘要: 提出一种基于质点平流的增强型 IBFV 可视化算法, 可显著增加 IBFV 算法生成图像的对比度。首先通过质点平流获得一系列的矢量纹理; 然后将这些矢量纹理作为 IBFV 算法中的背景图像, 代替原来的噪声纹理与帧缓存中的纹理进行图像混合生成新图。通过这种方式不仅可以准确反映流场的动态变化, 而且增强了矢量线间的对比, 同时还可以获得较高的绘制速度。

关键词: 矢量场可视化; 质点平流; 纹理映射; 图像混合

Enhanced IBFV 2D vector field visualization

Sun Changhui, Fan Yin, Li Qian, Huang Wei

(Institute of meteorology, PLA University of Science and Technology, Nanjing 211101 China)

Abstract: This paper presents an enhanced IBFV visualization algorithm based on particle advection, it can obviously increase the rendering image contrast. The particle advection is at first performed to get a series of vector textures. Then the vector textures are performed image blending with texture of frame-buffer to generate rendering images, which are treated as the background images of IBFV algorithm instead of noise textures. In this way, it can not only exactly reflect flow dynamic variety, but also enhance the contrast among flow lines. Furthermore, this algorithm can achieve a real-time frame rates as fast as IBFV algorithm.

Keywords: vector field visualization; particle advection; texture mapping; image blending

0 引言

矢量场可视化是科学计算可视化中非常具有挑战性的研究课题之一, 它通过直观的图形、图像来显示场的运动、洞察场的内涵本质和变化规律, 广泛应用于计算流体力学、空气动力学、气象预报等领域。流场是应用最广泛、研究最深入的矢量场之一, 目前主要有 3 类可视化方法: 基于几何形状的方法、基于颜色光学特性的方法和基于纹理的方法。点图标(箭头、有向线段等)、线图标(流线、迹线等)或面图标(流带、流面等)等都属于基于几何形状的方法,

这种方法选定起始点后构造并绘制点、线、面、体等中间图元, 以表示矢量数据的方向和大小。可见, 此种方法易于实现, 最先用于矢量场可视化, 但显示结果依赖于起始点位置的选择, 只能显示初始点相关的流场信息。第 2 类方法的代表是粒子系统, 用粒子的属性来反映矢量场, 其优点是灵活方便, 但有可能丢失场的连续性特征, 而且不宜观察局部细节信息。基于纹理的矢量场可视化综合了前两者的长处, 它以纹理图像的方式显示矢量场的全貌, 描述矢量场的细节, 是一种全局的可视化方法。具有代表性的算法有点噪声法^[1](spot noise), 在早期的矢量场可视化中具有重要地位, 但该算法不适合方向剧

收稿日期: 2009-09-17; 修回日期: 2010-06-29

第一作者简介: 孙长会(1983—), 男, 解放军理工大学气象学院信号与信息处理专业硕士研究生, 主要研究方向为科学计算可视化。E-mail: sunch_nal@126.com。

烈变化的矢量场,纹理细节较粗糙;线积分卷积法^[2](LIC),成像质量较高,纹理细节清晰,但计算耗时;图像平流法^[3](IBFV),绘制速度高,但成像对比度较低,单帧图像纹理细节不清晰。

目前,针对点噪声法、LIC 法的不足,已经提出了很多改进算法。de Leeuw 和 van Wijk 提出了增强型点噪声^[4]算法,该算法根据矢量场局部方向变化的剧烈程度,对点进行拉伸弯曲,并对结果图像进行滤波,提高了成像质量。Stalling 和 Hege 提出了 FastLIC^[5]算法,该算法基于盒形滤波卷积核,利用同一流线上像素间的相关性,在一次流线积分中通过差补法计算多个像素值,将 LIC 方法的计算速度提高了一个数量级。但有关 IBFV 算法的改进却很少,本文首次提出一种基于质点平流的增强型 IBFV 改进算法,该算法首先通过质点平流获得一系列的矢量纹理,然后将这些纹理高通滤波后与帧缓存中已经过平流的纹理进行图像混合,通过这种方式可以显著提高输出图像的对比度,同时还可以获得较高的绘制速度。

1 IBFV 算法

1.1 IBFV 算法概述

IBFV 算法是一种基于图像平流的算法,最终流场动画的每一帧都是由之前图像和一系列经过滤波的噪声背景图像的卷积得到,是一种用宏观图形表现微观粒子运动的新方法。相比点噪声法、LIC 法,IBFV 法可以充分利用计算机图形的功能,将大部分复杂的显示计算过程通过图形硬件来完成,从而整体获得较高的绘制速度。基于 IBFV 算法的扩展性研究很多, van Wijk 将 IBFV 算法成功应用于曲面流场^[6]的可视化,该算法以三角网格为基本图元,首先将 3 维曲面投影到 2 维平面,然后进行纹理平流与混合,最终实现曲面流场的运动。Telea 与 van Wijk 提出了基于硬件加速的 3 维 IBFV^[7]算法,将 IBFV 算法拓展到 3 维空间,该算法采用一种纹理平流技术,通过向前的纹理平流与原有的纹理进行比例混合,实现流场中粒子的运动。文献^[8]应用 IBFV 算法对空间晶体生长实验中产生的流场数据进行可视化,实现了流场数据的实时动态显示。

1.2 IBFV 算法数学描述

对于给定的 2 维稳定矢量场

$$v(p) = [v_x(p_x, p_y), v_y(p_x, p_y)] \quad (1)$$

式中, $p \in S$ 为空间位置点集, $v(p)$ 为相应位置处的速度, S 为矢量场的定义域。对于该矢量场中的任意质点,它的运动方程满足

$$dp/dt = v(p) \quad (2)$$

1 阶欧拉近似解为

$$p_{k+1} = p_k + v(p_k) \Delta t \quad (3)$$

式中, $k \in N$, 为流场动画的帧数。

假定 $F(p, k)$ 为流场动画帧,采用矢量场定义域 S 对 F 进行网格划分,并将每一个网格单元 $F(p_k, k)$ 看做质点按式(3)进行图像平流,设 $F(p_{k+1}, k+1)$ 为平流后的网格单元。对于 2 维稳定场平流轨迹将为流线,则 F 沿矢量场平流的过程可近似为

$$F(p_{k+1}, k+1) = \begin{cases} F(p_k, k) & p_k \in S \\ 0 & \text{其他} \end{cases} \quad (4)$$

由方程式(4)可知,若 p_k 平流出流场定义域 S ,最终 F 将被平流为全黑(值全为 0)的图像。为了避免这种情况发生,在每次平流过程中,增加背景噪声图像 G 的影响,则有方程

$$F(p_{k+1}, k+1) = (1 - \alpha)F(p_k, k) + \alpha G(p_{k+1}, k+1) \quad (5)$$

式中, $\alpha \in [0, 1]$ 为混合参数,方程式(5)递推得

$$F(p_k, k) = (1 - \alpha)^k F(p_0, 0) + \alpha \sum_{i=0}^{k-1} (1 - \alpha)^i G(p_{k-i}, k-i) \quad (6)$$

在方程式(6)中,第 1 项代表初始图像对结果的影响,当初始图像为全黑图像(即 $F(p_k, k) = 0$)或者 k 值取很大时,第 1 项可以忽略不计。由此可得

$$F(p_k, k) = \alpha \sum_{i=0}^{k-1} (1 - \alpha)^i G(p_{k-i}, k-i) \quad (7)$$

该式表明,图像中点 p_k 的颜色值可由图例 $G(p, i)$ 沿着通过 p_k 的流线积分卷积得出,卷积核为 $\alpha(1 - \alpha)^i$,即将现有图像沿流场方向平流,然后与背景图像相混合得出新的图像,重复这个迭代过程,将得到矢量场的动态描述。一般情况下,卷积长度为 32 时,即可获得稳定的矢量场动画,卷积长度也可称为图像平流周期。

该算法实现的关键是合理地选取混合参数 α 和背景图像 G 。在实际应用中, α 将被初始化为常数, α 值越大,背景图像 G 的影响将增加,矢量线间对比度将会提高,但矢量场的方向性将会被减弱。对于背景图像 G ,如果平流过程中始终不变,则最终平流的结果图像将是静态的,不能表现出矢量场动

态的效果。因此,每次平流都必须有不同的背景图像 G , van Wijk 通过研究认为,通过方波信号滤波并且相位变化的白噪声纹理将是合适的选取对象,即准备 M 幅不同的背景图像循环使用,这样就可以得到矢量场的动态显示。背景图像 G 中 p 点的颜色值可由下式计算得出

$$G(p, k) = \text{Square}((256k \bmod M + \text{Noise}(p)) \bmod 256) \quad (8)$$

式中, k 为平流帧数, M 为图像平流周期, Noise 为白噪声纹理。

通过分析式(7)(8),并与 LIC 算法进行对比,可以得出通过该算法获得的结果图像对比度低的主要原因,即参与卷积的每一项 $G(p_{k-i}, k-i)$ 的灰度值,并不像 LIC 算法那样来自同一幅白噪声纹理,而是来自一个变化的多幅背景噪声纹理。因此,矢量方向相关性被降低,导致纹理细节不清晰。

2 增强型 IBFV 算法

图 1 为算法生成第 i 帧流场动画的流程图,首先初始化质点集,包括质点的位置、速度、颜色等。其次将网格单元沿流场方向平流,计算网格单元每个顶点平流位置。再次进行网格单元纹理映射,即将原网格单元内的图像映射到平流后得到的新网格单元内^[9]。若已生成的动画帧数 i 小于图像平流周期 M ,则进行质点平流,构造背景图像,否则直接调用通过质点平流已经构造的背景图像。最后将背景图像与缓存中纹理图像进行图像混合,产生新图并显示。

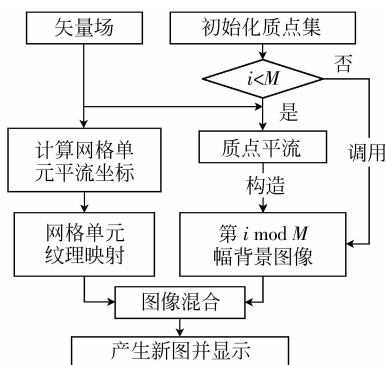


图 1 算法流程图

Fig. 1 The workflow of the algorithm

在该算法中,质点位置为矢量场相应网格点坐标,即在每个网格点上摆放一个质点,质点速度为相

应网格点坐标处的矢量值,质点颜色为随机白噪声生成的灰度值。在质点平流期间,算法主要计算集中在质点管理、卷积等操作,计算较耗时,相应的流场动画绘制速度较慢;平流结束后,算法主要计算集中在纹理映射及图像混合等操作,而纹理映射及图像混合均为 OpenGL 的标准操作,已得到了图形硬件的广泛支持,因此,流场动画绘制速度非常快。

2.1 质点平流

质点平流思想主要源于不稳定流场可视化, Shen 和 Kao 的 UFLIC^[10] (unsteady flow LIC) 算法从质点平流的角度重新阐释了 LIC 思想,并成功应用于不稳定流场的可视化。一般情况下,设在白噪声图像每个像素点位置都放置一个质点,该质点被释放后将沿着流场的方向流动,每到一个新的位置,都留下“印记”,也就是初始位置像素点的灰度值和对新位置处像素点的贡献值。另外,白噪声图像中的每个像素点又作为一个接收者,保存经过该像素点的质点“印记”。如图 2 所示,质点 P_1 、 P_2 从像素点 A 、 B 出发,沿流场方向被平流到像素点 C , C 作为接收者将保存 P_1 、 P_2 留下的“印记”,即 A 、 B 像素点的灰度值以及贡献值。质点平流的位置可以由欧拉法计算,也可以使用中点法或者精度更高的 4 阶龙格库塔法计算。经过多次平流后,将每个像素点保存的“印记”进行卷积,最终获得流场的矢量纹理图像。对任意像素点灰度值 $I(p)$,若采用盒式卷积核函数进行计算,则计算公式为

$$I(p) = \frac{1}{N} \left(\sum_{i=0}^{i=N-1} T(p_i) \right) \quad (9)$$

式中, $T(p)$ 为该像素点接收到的噪声纹理值, N 为接收到的质点个数。

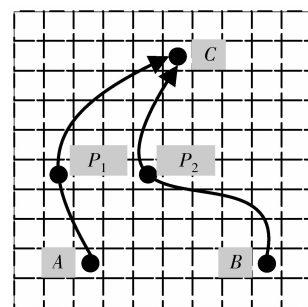


图 2 质点平流

Fig. 2 The illustration of particle advection

如果对所有质点每平流一次,就对每个像素点进行卷积,将获得一系列的矢量图像,此时已平流步

数将是卷积长度。图 3 为平流步数分别为 8、16、24、32 时获得的矢量图像,随着平流步数增加,矢量线越来越平滑,纹理细节逐渐清晰,平流步数为 32

时,图像质量又有所提高,但改善的幅度明显减缓,同时平流步数较大,计算也比较耗时,一般合理的平流步数为 24~32。

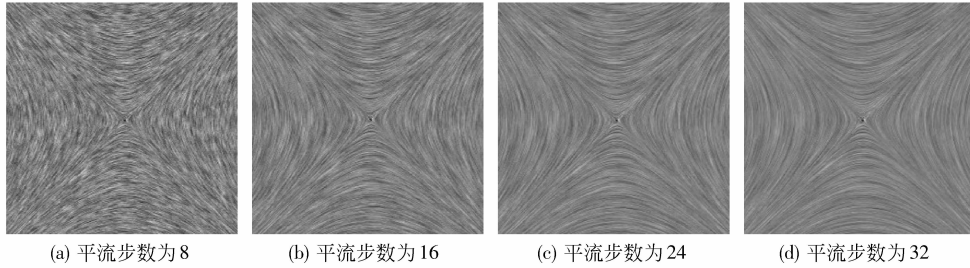


图 3 矢量纹理

Fig. 3 Vector texture with different advection step

2.2 构造背景图像

通过 1.2 节的讨论,已经知道标准 IBFV 算法图像对比度低的主要原因,即由参与卷积的每一项均来自不同的背景噪声纹理导致。可采用将质点平流获得的矢量纹理作为背景图像的方法来改善图像的对比度。则式(7)变为

$$F(p_k, k) = \alpha \sum_{i=0}^{k-1} (1 - \alpha)^i I(p_i) \quad (10)$$

由于卷积项 $I(p_i)$ 是通过质点平流获得像素灰度值的,此时矢量线上的像素点已经具有一定的方向相关性,所以相比原噪声背景图像,纹理细节显然

将变得清晰。即将纹理细节逐渐清晰的矢量图像不断与帧缓存中已平流的图像进行混合,最终获得高质量的流场动态显示效果。

通过质点平流获得的矢量纹理还不能直接应用,否则图像对比度将会非常低。可采用方波信号进行滤波,即

$$I'(p) = \text{Square}(I(p)) = \begin{cases} 0 & I(p) < 128 \\ 255 & \text{其他} \end{cases} \quad (11)$$

对图 3 滤波的结果如图 4 所示。

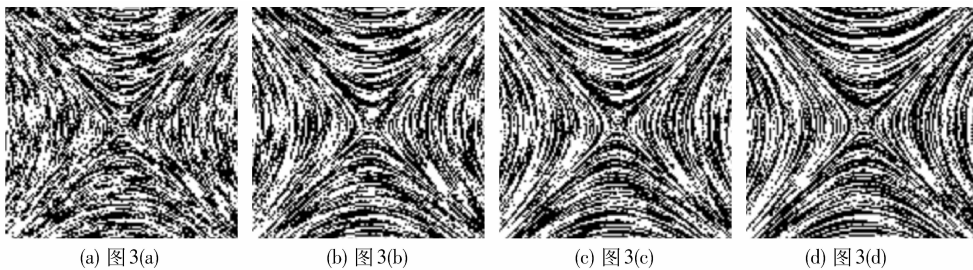


图 4 滤波结果

Fig. 4 The result of filtering

再将经过滤波的矢量纹理逐步混合到缓存中,由于背景图像是一系列由模糊逐渐变为清晰的图像,所以在图像平流过程中将出现图像由清晰变为模糊,随后又逐渐清晰的情况,主要是每个图像平流周期开始后,比较模糊的第 1 幅背景图像将首先被混合进帧缓存的原因。可以通过减小混合因子 α 的方式,减弱背景图像对缓存中纹理的影响,使动画帧间的跳变消失。

3 实验结果

实验环境为普通 PC、Windows XP 操作系统、P8600 2.4 GHz、2 GB 内存和 256 M ATI 3470 显卡,算法基于 VC 6.0 环境调用 OpenGL 函数库实现,显示窗口分辨率为 400 × 400,图像平流周期为 32,算法混合因子 α 为 0.02。目前本文算法主要针对 2 维

稳定场,选用 3 组矢量数据测试该算法,数据 1 为模拟马鞍型矢量场,分辨率为 300×300 ,数据 2 为涡流矢量场,分辨率为 256×256 ,数据 3 为数值预报风场数据,分辨率为 121×125 ,为方便研究及观察,通过双线性插值,数据分辨率为 256×256 。

图 5 为质点平流结束后获得的流场动态显示效果图,图 5 上面 3 幅图为采用标准 IBFV 算法获得的流场动画帧图像,图 5 下面 3 幅图为采用本文算法获得的动画帧图像。从对比效果看,采用本文算法

后,图像对比度明显得到了提高,纹理细节更加清晰,矢量特征更容易区分。表 1 为在不同数据分辨率下两种算法绘制速度对比及所有背景矢量纹理全部生成完毕后的用时。从数据对比看,在质点平流期间本文算法绘制速度较慢,主要原因是质点位置计算以及卷积比较耗时,质点平流结束后,本文算法绘制速度明显得到了提高,与标准 IBFV 算法绘制速度相近,满足交互式实时绘制要求。

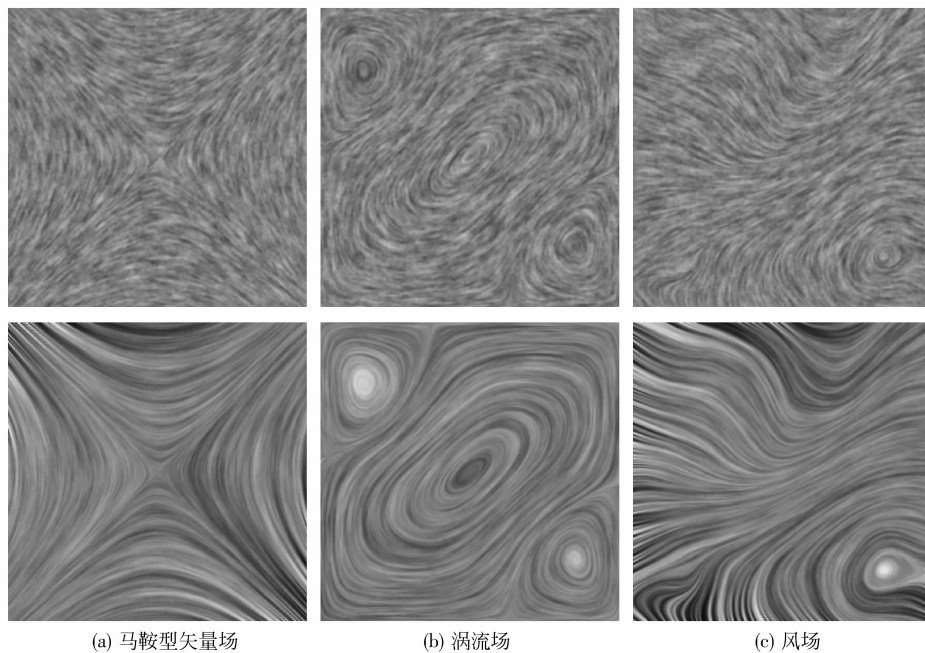


图 5 效果对比

Fig. 5 Contrast of effect

表 1 绘制速度

Tab. 1 Frames per second

矢量数据 分辨率	IBFV 算法	本文算法		
		平流期间 /(帧/s)	平流结束 /(帧/s)	平流用时 /s
256×256	62	9	58	3.837
300×300	42	6	40	5.756

4 结 论

通过对 IBFV 算法进行数学分析,首次得出该算法矢量动画单帧图像对比度低的主要原因,即由参与卷积的每一项均来自不同的背景噪声纹理导致。针对此原因,将质点平流思想引入 IBFV 算法中,用通过质点平流获得的矢量纹理代替噪声纹理

与帧缓存中的纹理进行图像混合以生成新图。实验证明,通过这种方式获得的流场动态效果图像,不仅纹理细节清晰,图像对比度高,同时还可以获得较高的绘制速度,有助于观察流场特征及主要拓扑结构。在质点平流过程中对流场的一些标量属性如大小、角度和曲率等进行色彩映射,以及将算法扩展到非稳定矢量场和曲面矢量场将是下一步研究的重点。

参考文献 (References)

- [1] Van Wijk J J. Spot noise: texture synthesis for data visualization [J]. Computer Graphics, 1991, 25(4): 309-318.
- [2] Cabral B, Leedom C. Imaging vector fields using line integral convolution [J]. Computer Graphics, 1993, 27(4): 263-272.
- [3] Van Wijk J J. Image based flow visualization [J]. ACM Transactions on Graphics, 2002, 21(3): 745-754.

- [4] De Leeuw W C, van Wijk J J. Enhanced Spot Noise for Vector Field Visualization [C/OL] // Proceedings Visualization, 2002. [2009-06-08]. <http://homepages.cwi.nl/~wimc/publications.html>.
- [5] Stalling D, Hege H. Fast and resolution independent line integral convolution [C] // Proceedings of the ACM SigGraph ' 95. New York: ACM SIGGRAPH, 1995: 249-256.
- [6] Van Wijk J J. Image Based Flow Visualization for Curved Surfaces [C/OL]. [2009-06-08]. <http://www.win.tue.nl/~vanwijk/pubs.htm>.
- [7] Telea A, Van Wijk J J. 3D IBFV: Hardware-Accelerated 3D Flow Visualization [C/OL] // Proceedings of IEEE Visualization, 2003. [2009-07-02]. <http://www.win.tue.nl/~vanwijk/pubs.htm>.
- [8] Wu Xiaoli, Shi Meiping, He Hangen. Dynamic flow visualization in space crystal growth [J]. Computer Simulation, 2007, 24 (4): 190-192. [吴晓莉, 史美萍, 贺汉根. 空间晶体生长实验中流场的动态显示研究 [J]. 计算机仿真, 2007, 24 (4): 190-192.]
- [9] Van Wijk J J. IBFV-Supplemental Material ACM SIGGRAPH 2002 [EB/OL]. [2009-09-05]. <http://www.win.tue.nl/~vanWijk/ibfv>.
- [10] Shen H W, Kao D L. A new line integral convolution algorithm for visualizing time-varying flow fields [J]. IEEE Transactions on Visualization and Computer Graphics, 1998, 4 (2): 98-108.