基于流线纹理合成的 2D 矢量场可视化

张 文
(国防科技大学计算机学院,长沙 410073)
李晓梅
(装备指挥技术学院,北京 101416)

摘 要 针对科学试验数据可视化问题,提出了一种 2D 矢量场可视化的方法——流线纹理合成方法,即通过将 1D 纹理映射到流线上,再利用流线纹理来合成可视化图象,因为移动 1D 纹理很容易形成矢量场动画.该方法是利用局部区域内流线的近似平行性,首先依据临界点来设定流线宽度,然后把流线绘制成多条平行流线,再分别将多条不同的 1D 纹理映射到流线上,从而能够加快计算.实际结果表明,由于流线纹理合成的纹理图象上清楚地显示了流线,因而反映了矢量场的方向信息,同时,该方法计算速度快,可以达到交互可视化的要求.

关键词 矢量场可视化 流线纹理合成
中图法分类号:TP91.41 **文献标识码**:A **文章编号**:1006-8961(2001)03-0280-05

2D Vector Field Visualization Based on Streamline Texture Synthesis

ZHANG Wen
(*Institute of Computer Technology, National University of Defense Technology, Changsha* 410073)
LI Xiao-mei
(*Institute of equipment command and technology, Beijing* 101416)

Abstract In 2D vector field visualization, the texture-based methods represent the field structures globally, and have become attractive alternatives. In this paper we presents a new texture-based technique: Streamline Texture Synthesis (STS). Many 1D textures which have high strong directions are synthesized firstly. STS uses a random 1D texture to texture map each streamline, then synthesizes texture mapped steamlines. In local vector field region, the steamlines are approximately parallel, so a streamline can be drawn thicker in order to accelerate computing. The thickness of a streamline is set based on the critical point of a vector field. The thick streamline is rendered as many parallel streamlines, then many 1D textures are used to texture map them. The STS can generate animation by cycling through 1D texture. The analyzed result proves that STS can produce substantially better visual results and STS is fast enough to visualize vector field interactively.

Keywords Vector field visualization, Streamline texture synthesis

0 引 言

矢量场可视化(如流场可视化,动态系统可视化等)是科学可视化研究的一个重要课题,现已广泛应用于天气预报、地质勘探、航空航天、核模拟等领域.虽然矢量场可用箭头、流线、迹线等来实现可视化,但是,大量的箭头图标会导致所生成的可视化图象杂乱无章,而且这种流线、迹线的可视化效果与种子点(流线、迹线的起始点)的选择有关.LIC(Linear Integral Convolution)是一种基于纹理的 2D 矢量场可视化方法[1],它是对输出图象的每个象素点,首先计算一条流线,并用 1D 滤波器沿流线方向卷积白噪声图象,再利用其得到的象素值来合成纹理.FLIC(Fast LIC)则是利用同一流线上卷积积分的重叠,来加快 LIC 计算[2].虽然 LIC 纹理较好地反映了矢量场的结构,但其纹理图象较为模糊,而且,LIC 计算费时,很难达到交互可视化的要求;而 ID(Integrate and Draw)方法是对 LIC 计算出的每条流线赋一随机灰度值[3],这虽使可视化图象有所改善,但不易形成动

基金项目:九五国防预研项目(16.1.4.5)
收稿日期:2000-01-26;改回日期:2000-06-06

画.通过对 FLIC 和 ID 方法的改进,本文提出了一种流线纹理合成(Streamline Texture Synthesis,STS)的方法,它能快速、有效地实现 2D 矢量场的可视化,并且可以快速地实现动画.

1　LIC 纹理合成

LIC 方法用 1D 滤波器沿流线方向来卷积一白噪声图象[1],以合成纹理.假定流线用 $\sigma(s)$ 表示,其中,s 是弧长参数,则输出的 LIC 纹理在点 $x_0=\sigma(s_0)$ 的强度为 $I(x_0)$,于是,LIC 计算可用下式表示

$$I(x_0)=\frac{1}{\int_{-L}^{L}k(s)\,\mathrm{d}s}\int_{s_0-L}^{s_0+L}k(s-s_0)\,T(\sigma(s))\,\mathrm{d}s \quad (1)$$

其中,T 为 2D 白噪声,k 为滤波器(可为常数滤波、三角滤波等),滤波器长为 L.虽然输入的白噪声图象各点不相关,而由于 LIC 计算却使得输出纹理在流线方向上相关,故其输出纹理能显示矢量场的方向信息.对输出图象的每个象素点,LIC 不仅都要计算其流线,而且还要用式(1)计算该点的 LIC 纹理强度.FLIC(Fast LIC)又对 LIC 进行了改进[2],即在矢量场中的同一条流线上,由于相邻采样点的式(1)积分有重叠,因此对于常数滤波器,其后继采样点只要计算非重叠部分即可.如图 1 所示.

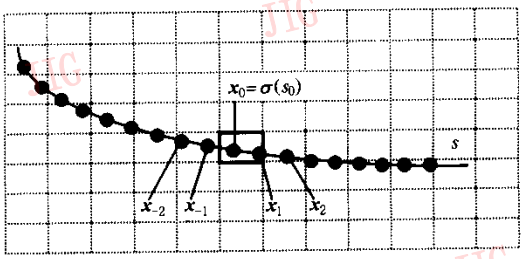


图 1　FLIC 示意图

对于 FLIC,式(1)的离散形式为

$$I(x_0)=\frac{1}{2n+1}\sum_{i=-n}^{i=n}T(x_i) \quad (2)$$

其中,$x_i=\sigma(s_0+ih)$

流线积分时,以 x_0 为初始点,向两个方向计算,h 为积分步长.当计算出 $I(x_0)$ 后,则沿同一流线,其他的采样可用下式计算

$$I(x_{i\pm1})=I(x_i)+\frac{1}{2n+1}[T(x_{i\pm(n+1)})-T(x_{i\mp n})] \quad (3)$$

若每一采样点对应输出图象的一个象素点,并将计算强度值赋给该象素点,于是就命中一次该象素.FLIC 对未命中的每个象素,再以该象素中心对应的矢量场位置来计算一条一定长度的流线,然后沿同一条流线采样,命中多个象素点,这样就使 FLIC 计算比 LIC 约快一个数量级.ID 方法是对同一条流线命中的所有象素赋予同一随机象素值[3],以代替卷积计算,从而加快了可视化计算.

2　流线纹理合成

2.1　STS 算法流程

STS 的主要思想是采用 1D 纹理映射的方法来绘制流线,以得到流线所命中象素点的值,最后合成可视化输出图象.由于考虑到矢量场局部区域内流线是近似平行的,因此可将流线绘制成一条具有一定宽度的曲线,以使得同一条流线命中更多的象素点,从而来加快可视化的计算过程.另外,在流线绘制时,通过移动 1D 纹理,即可以形成矢量场动画.STS 算法流程如下:

```
FOR (输出图象的每一象素点 p) {
  IF (p 未被命中) {
      a.沿 p 计算一条流线;
      b.依据 p 在矢量场中的位置,确定流线的绘制宽度;
      c.采用 1D 纹理映射,进行宽度流线绘制(对多次命中
        的象素点进行值累加,并记录命中次数);
  }
}
FOR (输出图象的每一象素点 p)
      p 的象素值 = p 的累加象素值/p 的命中次数;
对输出图象进行高通滤波、直方图均化等后处理.
```

下面将详细讨论 STS 算法.由于流线的计算是一个给定初始位置求解微分方程的问题,因此,一般采用 Euler 方法、Runge-kutta(RK)方法.本文采用四阶嵌入式 RK 方法来计算流线,并给定规则网格点的矢量值,再采用双线性插值方法以得到任一点的矢量值.对于 STS 算法流程中的象素扫描次序,文献[2]讨论了逐行扫描、分块扫描、半随机 SOBOL 序列等 3 种扫描次序.

实践证明,采用半随机 SOBOL 序列,对 STS 效果最好.另外,为了提高可视化效果,可以对输出图象进行高通滤波、直方图均化等后处理.为了增强流线之间的差异,可采用高通滤波器,同时,利用查找表方法快速实现滤波卷积[4].

2.2　1D 纹理合成

为了获得可视矢量场的方向信息,并使得可视图象能很好地刻画流线,进行流线映射的 1D 纹理必须具有很强的方向性,即象素间需具有相关性.由

式(3)可知 在一流线上,由于 FLIC 计算的相邻象素值相差$\frac{1}{2n+1}[T(x_{i\pm(n+1)})-T(x_{i\mp n})]$,且这是一个小的随机值,因此,可以采用随机的方法来得到一方向性很强的 1D 纹理,即首先随机产生一纹理值(对灰度纹理,纹理值在[0,255]之间)作为 1D 纹理中一点的值,然后以此点为起点,将前一点的纹理值加上一个$[-\delta,+\delta]$间随机值来获得下一点纹理值,这样即合成了 1D 纹理,但在计算过程中,控制纹理值不要超出上述纹理值的范围,而且加上的 δ 越大,合成纹理的方向性就越不强,对灰度纹理,δ 一般取 0 ~ 60 效果比较好.实际上,若 δ 等于 0,则 STS 合成的图象与 ID 一致.为了避免对每条流线都要合成 1D 纹理,可以先计算多条具有一定长度的 1D 纹理,以形成具有很强水平方向性的 1D 纹理模板,而且这种模板大小可以调节,如图 2.这样纹理映射时,即可随机选取一行 1D 纹理,来映射到流线.

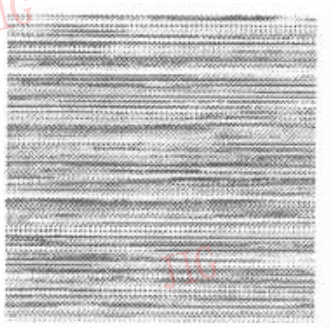


图 2 1D 纹理模板

2.3 宽度流线绘制

当流线绘制到输出图象空间时,STS 即利用 1D 纹理映射来得到该流线所命中的象素点的值,其简单映射方法是,首先从纹理模板中随机选出一条 1D 纹理,然后将其象素点一一对应到流线命中的象素点上,而且在流线绘制时,可能多次命中同一象素点,STS 则将每次得到的象素值进行累加,并记录命中次数,最后将累加值除以命中次数,即得到输出象素值 而在进行宽度流线绘制时,则可将流线绘制成多条平行流线,且一条流线占据一个象素宽度,然后从纹理模板随机选出连续的多条 1D 纹理分别映射到各条流线上(如图 3),这样,同一流线则命中了更多的象素,显然会加快计算速度.

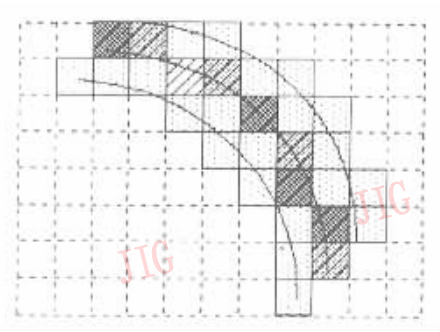


图 3 宽度流线纹理映射

但随着流线宽度加大,将使得可视化图象的分辨率下降,因而不能很好地反映矢量场的细节,这是因为,实际上在矢量场特征点附近,流线不具有近似平行性.由于临界点是矢量为零的点,它能表明矢量场的特征细节[5],因此,为了更好地刻画矢量场的特征,可以依据临界点来自动设置流线的宽度,而且距离临界点越近的流线,宽度越小,这样,流线的宽度就确定了流线绘制时平行流线的条数.由于矢量场中任一点的矢量是通过网格单元顶点的矢量插值得到的,因此若要使网格单元中存在临界点,则对网格单元顶点的各个分量,必须同时存在正、负.这样对于任一网格单元,即首先可以通过检测其任意两个顶点分量的符号来确定其是否是临界点候选单元,即如果存在一对顶点,且它们的每个分量都存在正、负,那么这个网格单元就是一个临界点侯选单元 然后,将侯选单元对分,再对小的单元检测,直到定位到结果图象的象素单元.实际上,可以简单地将侯选的矢量场网格单元直接对应到可视化结果图象上,以得到一定的象素区域,然后将此区域进行一定的扩大,并将构成区域中流线的宽度设定为 1 个象素,其余就可指定宽度.

2.4 动画合成技术

流线绘制时,通过移动 1D 纹理,即可得到不同帧的象素,并可以形成矢量场动画.由于 1D 纹理象素间具有一定的相关性,且各帧图象间象素也相关,因此可保证动画的时空相关性.但因为 1D 纹理缺乏周期性,因而直接循环播放帧图象,则使得周期间的动画有跳动.于是,本文采用下面的帧合成方法,实现了很好的动画效果.为了得到 N 帧图象,可以通过移动 1D 纹理,来计算出 $2N$ 帧图象,再通过下式将 $2N$ 帧图象混合成 N 帧图象

$$I'(x_0,i)=w_1(i)I(x_0,i)+w_2(i)I(x_0,i+N) \quad (4)$$

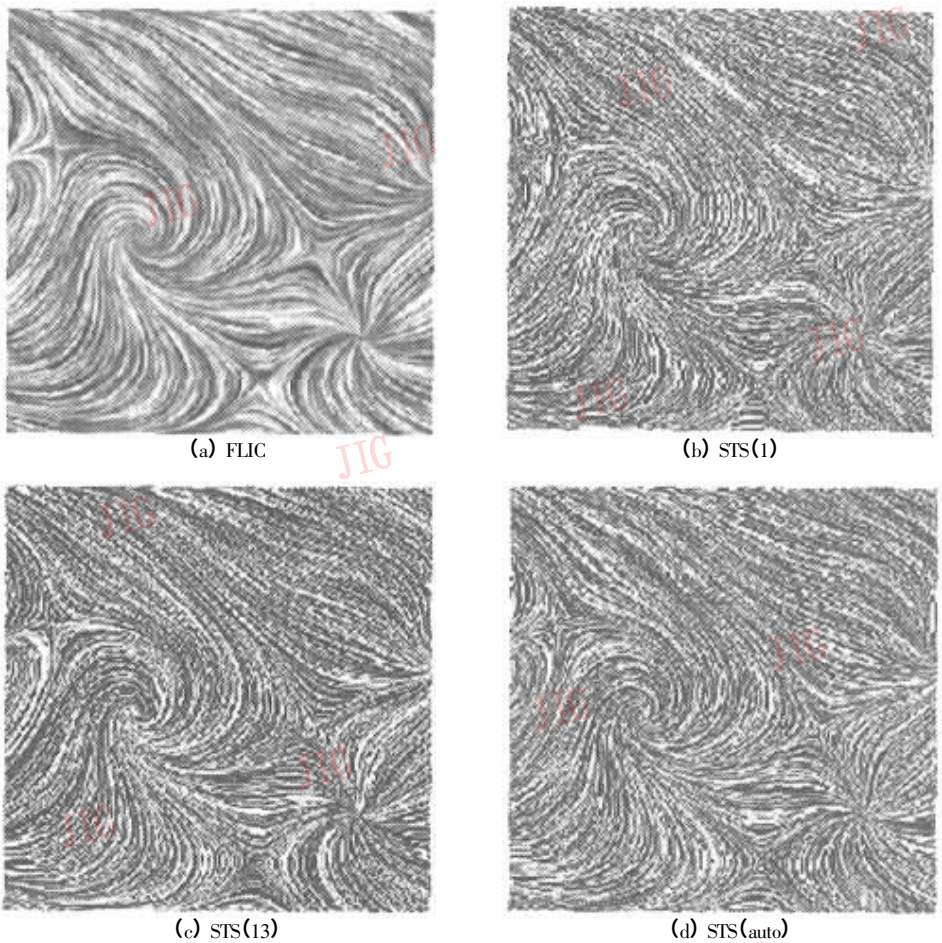
其中,$w_1(i)=i/(N-1)$,$w_2(i)=1-w_1(i)$,$i=0,1,\cdots,N-1$,$I'(x_0,i)$表示第 i 帧输出图象 x_0 点的

象素值,$I(x_0,i)$,$I(x_0,i+N)$表示计算的 $2N$ 帧图象的象素值.如果依据速度大小改变 1D 纹理移动的大小,则可以形成变速动画,且动画速度反映矢量大小.由此可见,这种动画更好地反映了矢量场的方向性.

3　实际结果分析

本文在 Pentium Ⅲ 450 处理器、128M RAM 的微机上,利用 C++ 和 Java 两种语言实现了上述算法.首先利用 Java 语言实现了 STS 的 Java Applet,然后通过浏览器,用户就可以交互地实现自己的矢量场数据的可视化.由于 Applet 需要读入矢量场数据,因此需要使用能够允许 Applet 访问文件的浏览器,如 HotJava 或 AppletViewer.下面对利用 C++ 实现的实际结果分析.

对于一测试的矢量场(TVF),图 4 是分别利用 FLIC、STS(1)、STS(13)和 STS(auto)实现的可视化图象,其中,括号内数字表示用象素点度量的流线宽度,auto 表示依据临界点来自动设置流线宽度.从采用的 FLIC 计算结果可以看出,LIC 纹理图象较好地反映矢量场的结构,但流线间的对比度不强,且较模糊(图 4(a));STS(1)图象则使得流线间的对比度得到增强,并很好地刻画了矢量场的方向(图 4(b));图 4(c)中流线宽度为 13 个象素点,从图中看到,场特征点附近的流线杂乱不清,已丢掉了一些细节;而图 4(d)由于依据临界点调整了流线的宽度,因此在临界点附近也能清楚地反映流线的方向性.对于相关文献[1]、[2]、[3]中的一些矢量场实例也可以得到上述效果.表 1 是利用 FLIC、STS(1)、STS(auto)3 种算法对不同矢量场实例进行计算的时间比较.从表中可以看出:STS(auto)计算时间比 FLIC 提高了 53%81%,而比 STS(1)提高了 28%60%.由于不同矢量场的结构不同,因此算法改进的性能也不一样.表 1 还给出了利用本文所述的动画合成技术,由 STS(auto)合成 10 帧图象所需的计算时间,由表 1 可见,STS(auto)动画合成的效率比较高.由此可见,STS 的计算速度完全可以达到交互可视化的要求.



(a) FLIC　(b) STS(1)　(c) STS(13)　(d) STS(auto)

图 4　利用 FILC、STS(1)、STS(13)和 STS(auto)实现的可视化图象

表 1 FLIC、STS 和 ESTS 计算时间的比较

矢量场	可视化图象的分辨率(pixel)	计算时间(s)			
		FLIC	STS(1)	STS(auto)	动画合成
TVF	256×256	0.99	0.39	0.22	0.77
水分子的静电场(water)	512×512	2.47	1.26	0.52	3.57
苯分子的静电场(benzene)	512×512	3.73	1.76	1.21	4.34
绕圆柱的水流(cylinder)	800×200	0.94	0.61	0.44	1.48
双极子天线场(dipol)	512×512	3.46	1.64	0.66	4.01

4 结 论

STS 能够快速、交互地实现 2D 矢量场的可视化,同时合成图象较好地显示了流线的方向性.若将 1D 纹理映射到迹线上,则 STS 可很容易地用于可视化时变场.若将矢量场从物理空间转换到计算空间,并在计算空间上计算 STS 可视化纹理图象,然后将图象再映射到物理空间,则可以快速地实现 3D 曲面流的可视化.由此可见,STS 是一种有效的矢量场可视化方法.

参 考 文 献

1 Cabral B, Leedom L. Imaging vector fields using line integral convolution. In: Proceedings of SIGGRAPH '93, Anaheim California, 1993: 263272.

2 Stalling D, Hege H C. Fast and resolution-independent line integral convolution. In: Proceedings of SIGGRAPH '95, Los Angeles California, 1995: 249256.

3 Risquet C. Visualizing 2D flows: Integrate and draw. In: Proc. Of the 9th Eurographics Workshop on Visualization in Scientific Computing, Germany, 1998.

4 Wolberg G, Massalin H. Fast convolution with packed lookup tables. Graphics Gems IV, Academic Press, 1994.

5 Helman J L, Hesselink L. Visualizing vector field topology in fluid flows. IEEE Computer Graphics and Applications, 1991, 11(3): 3646.

张 文 1970 年生,1998 年在国防科技大学电子工程学院获硕士学位,现为该校计算机学院博士生.主要研究方向为科学计算可视化算法与系统.

李晓梅 1938 年生,现任装备指挥技术学院教授,国防科技大学计算机学院兼职教授,博士生导师.从事大规模并行计算、科学计算可视化研究.