一种基于可编程 GPU 的实时烟雾模拟算法研究

邓定胜
(四川民族学院理工学院 四川 康定 626001)

摘 要 随着国内经济的日益增长和科技的进步,人民的生活质量以及生活水平也得到了逐步提升,同时精神文化需求也越来越强烈,对动画、影视及其相关方面提出了更高的要求。提出一种在可编程 GPU 基础上对烟雾模拟进行有效计算的实时算法。在预先确定的烟雾实时状态的基础上,通过实验实现可编程 GPU 的实时烟雾模拟算法,且以非常自然的烟雾流动来达到向目标的形状和状态进行转换的目的。通过对于可编程 GPU 的实时烟雾模拟算法的相关分析和研究,希望可以对我国相关技术和相关产业的进一步发展提供借鉴价值和指导作用。

关键词 可编程 GPU,烟雾模拟,实施算法,纹理,帧
中图法分类号 TP302 **文献标识码** A

Study on Real-time Smoke Simulation Algorithm Based on Programmable GPU

DENG Ding-sheng
(School of Science and Technology, Sichuan University Nationalities, Kangding, Sichuan 626001, China)

Abstract With the increasing of the domestic economy and the progress of science and technology, people's quality of life and standard of living have also been gradually improved, while spiritual and cultural demand is more and more strong, so higher requirements are required for animation, film and television and related aspects. Therefore, a computer real-time algorithm based on programmable GPU that can control smoke simulation effectively is born, and this study is based on this. Through the experimental simulation in real time, smoke simulation algorithm of programmable GPU can be implemented in real-time status on the basis of predetermined smoke, and can achieve the target of shape and state transform through fairly natural smoke flow. Through the programmable GPU real-time smoke simulation algorithms of correlation analysis and research, certain reference significance and guiding role can be provided for the further development of related technologies and related industry in China.

Keywords Programmable graphics processing unit, Smoke simulation, Implementation algorithm, Texture, Frame

制作动画时必不可少的效果之一便是烟雾这类没有固定形态的特效[2]。烟雾效果广泛应用于影视特效、电脑游戏动画、模拟飞行这些方面。需要用到烟雾特效的场面事实上要比想象之中多不少。不管是处于战斗场景、日常生活,还是处于风景画,全都能够通过它的加入达到提升画面的华丽程度的目的,这个东西非常具有承托的作用[3]。最先的烟雾动画是动画师通过手工绘制而成的,即使此方法允许动画师精准操控动画设计的每一个中间过程,可它需要的工作量实在太大,所以不好捕捉烟雾中像漩涡等这些难以发现的信息。通过计算机可以使烟雾动画效果的生成更方便,还可以减少动画师的工作量。可是因为烟雾运动瞬息万变,形状不好掌控,给研究人员带来了不小的困难。Stam[1] 给出了一种以物理为基础的计算方法即稳定流体模拟法,利用近似求解的方法来算出流体物理的方程式[2],可以达到完全稳定流动的目的。现在的烟雾模拟大多数都以该工作为基础,利用流体物理方程式模拟烟雾流动的过程,达到自动生成看起来比较自然、真实的烟雾效果的目的[4]。

1 可编程 GPU 基础上的实时烟雾模拟原理

本文主要是在国外相关的研究(以 Navier-Stokes 方程为例)基础上进行的,通过一定的调查和研究可以清楚地知道,Navier-Stokes 方程式主要用来定义烟雾流体场的物理模型,在整个模型中主要是通过对方程中的风力项进行一定处理来达到对烟雾状态和形象进行的目的的[5]。其中的风力项由驱动力和聚合力两部分构成,驱动力可以起到促进烟雾按照规定和计划的目标移动的作用,聚合力则是用于防止和避免烟雾流动过程中出现的扩散现象。

由于本文研究的算法不需要保证初始状态和形状与期望状态和形状之间转移是最优的,因此在很大程度上降低了非线性优化工作所带来的大量的计算方面的成本。通过本文的研究和实际的实验,在很大程度上证明了本文所研究的实时烟雾模拟算法可以保证烟雾状态相互转化的实时性和高效性。

本文受国家自然科学基金项目(11461058),四川省教育厅自然科学重点项目(17ZA0295),四川民族学院 2017 年应用型示范课程项目(sfkc201705)资助。
邓定胜(1978—),男,副教授,主要研究方向为算法设计,E-mail:307729837@qq.com。

2　可编程 GPU 基础上的实时烟雾模拟发展历程

对烟雾进行一定的动态效果处理,是近年来计算机动画领域新兴的一个较为热门的研究方向[6]。

在二阶粒子系统的基础上,Fiore 允许相关的动画师可以自行定义流体(烟雾)的运动轨迹及其加速线,赋予了动画师较高程度的自主权[7]。在 Fiore 模型中,动画师可以预先准备出一些可以随着时间的不断变化而产生相应变化的二维路径,随后可以通过相关系统的加工将这些二维的路径扩展为三维的路径,进而生成动画,同时对于其流体(烟雾)的运动速率可以通过加速线进行一定的调节。虽然,该模型可以在很大程度上起到方便烟雾等气体的上升和飘动的作用,但其发生作用的同时也需要大量的人机交互。

Treuille 提出了一种在关键帧技术的基础上对烟雾进行动态处理的模拟算法。在其算法中,关键帧包括两方面,分别是密度关键帧和速度关键帧,分别表示在某个具体的时间 t 用户所确定的密度 p_t^* 和速度 u_t^*,当前烟雾密度 p_t 和速度 u_t 应该尽最大的可能与之相接近。为了在真正意义上达成用户所需的效果,该模型中的模拟算法通过某种方式添加了风力项,通过风力项的作用使得烟雾可以向着关键帧的状态和形象转变[8]。

Stam 则通过对梯度的计算和优化来模拟和发挥出风力的作用,通过其风力项的流动来切实达到关键帧的要求。

McNamara 引入了相关的伴随方法来对烟雾的动态变化进行模拟。同时,这种算法也应用了梯度的非线性技术来对烟雾的运动进行调节,随后使用相关的伴随方法来最大程度上降低系统的依赖性,从而提高了整个过程中的时间效率,但是较大的局限性是需要非常大的存储量。

在 Treuille 相关研究的基础上,Fattal 应用了新的烟雾方式,即目标驱动。该算法应用一系列的期望状态和形状将关键帧因素取代,进而来模拟烟雾的变化过程。其中的驱动力可以促进烟雾向着期望的密度分布和运动,而其聚合力降低了最优算法而带来的高额成本。

3　可编程 GPU 基础上的实时烟雾模拟的主要算法及其方程

本文的相关算法主要是在 Fattal 的工作基础上,将实时烟雾的流动速度以及相关的密度等参数转换成具有一定纹理颜色的通道,随后通过使用计算机的 GPU 对已经传送到相关像素程序中的烟雾进行预算,最终就可以得到可以有效实时模拟烟雾过程的算法[9]。

3.1　其流体物理的方程定义及含义

从相关的流体力学出发,我们知道 NS 方程是用来对流体进行表达形式中最为全面、均衡的微分方程式。一个不存在任何粘性的流体场的 NS 方程式可以用以下的公式进行表示:

$$\frac{\partial u}{\partial t}=-u\cdot\nabla u-\nabla p+f \tag{1}$$

$$\nabla\cdot u=0 \tag{2}$$

方程(1)也可以称为不可压缩的流体动力方程式;同时,方程(2)也可以称作不可压缩的流体连续方程式。在上述的两个方程中,字母 p 代表压强,字母 f 代表风力,u 代表速度,"·"表示矢量点积,"▽"代表微分算子。方程(1)中存在的对流项 $-u\cdot\nabla u$ 主要表示相关的流体(实时烟雾)随着其速度场进行水平的相关传送和改变;而方程式(2)左半部分中的 $\nabla\cdot u$ 代表相关过程中存在的速度场散度,通过这个方程式可以知道该过程中的速度场不存在散场,也就是说该过程中的流体具有一定的连续性。该模拟算法中的实时烟雾可以根据其浓度场的变化进行相关的描述,而其中的浓度场会伴随着其模拟算法中流体场变化发生一定程度的平移和变化,如方程(3)所示:

$$\frac{\partial p}{\partial t}=-u\cdot\nabla p \tag{3}$$

在该实时烟雾的模拟算法中,f 代表的风力是其算法中对实时的烟雾进行处理的最为主要的手段和方法。这种在可编程 GPU 基础上的实时烟雾模拟的实时算法是通过风力来促使模拟中的烟雾从当前的状态和形状向着期望的状态和形状进行转变。但是我们必须注意到该模拟算法中的风力因素包括着两方面的内容,分别是驱动力和聚合力,驱动力可以起到促进模拟算法中的实时烟雾向着期望的状态和形状进行转换的作用;而其风力中的聚合力可以在很大程度上防止模拟算法中的实时烟雾向着期望的状态和形状进行转换发生扩散现象[10]。

在 Fattal 表述的算法中,将流体动量方程中 f 代表的风力转换为 $F(p,p^*)$代表的驱动力,如方程(4)所示:

$$\frac{\partial u}{\partial t}=-u\cdot\nabla u-\nabla p+\nabla v_f F(p,p*)-V_d u \tag{4}$$

同样地,该模拟算法中的驱动力 $F(p,p^*)$的主要作用也是将实时烟雾的当前状态和状态转换成预期的目标状态和形状;$-v_d u$ 代表着动量衰减因素,主要可以起到保证该模拟算法方程式的动量是守恒的。

另一方面,该模拟算法中的实时烟雾状态和形状在向期望的状态和形状进行相关转换的过程中,会因为其中的扩散作用导致其烟雾的浓度发生变淡和衰减,进而影响到其图像的清晰度,因此必须要对方程(3)添加其聚合力 $G(p,p^*)$,用来防止和避免实时烟雾向目标烟雾状态和形状转换的过程中的烟雾扩散现象,如式(5)所示:

$$\frac{\partial p}{\partial t}=-u\cdot\nabla p+v_g G(p,p^*) \tag{5}$$

其中,v_f 起着对其模拟算法中的驱动力进行的作用;v_d 则是可以对其动量的衰减因素进行一定程度的控制;v_g 则是其模拟算法中的聚合力的非负参数[11]。

3.2　驱动力

其模拟算法中驱动力必须要完成两方面的工作:

(1)促进模拟算法中的实时烟雾的当前状态和形状即 $p(x,t)$向着期望的状态和形状 $p^*(x)$进行转换;

(2)当其实时烟雾的相关转换完成以后,模拟算法中的烟雾应该处于静止的状态,用相关的方程式表示 $u=\frac{\partial u}{\partial t}=0$,这可以在很大程度上保证其相关的转化过程是稳定的。因此,我们必须注意到,其模拟算法中的实时烟雾由当前的目标状态和形状向着期望的状态和形状转换的过程中,伴随着实时烟雾的当前状态向目标状态的不断接近,模拟算法中的驱动

力所产生的动量必须要相应地逐渐减小，进而达到保证其实时烟雾在相关的转换完成时烟雾的相关运动是静止的。

由于该模拟算法中的实时烟雾的期望状态的梯度 ∇p^* 总是会在很大程度上指向烟雾的密集处，因此，我们可以将其梯度 ∇p^* 的所在方向认为是实时烟雾的流动方向，那么模拟算法中任意点处的驱动力都可以说是与该梯度有关系的，用相关的方程式进行表示，即：

$$F(p,p^*)=\alpha\nabla p^* \tag{6}$$

为了防止和杜绝在模拟算法中发生梯度为零的情况，就必须要通过一个高斯核与 p^* 做卷积对其进行一定的模糊处理，通过这个过程所得到的 $\widetilde{p}^*$ 可以在很大程度上保证模拟算法中各处梯度都处于非零的状态。该高斯核的相关定义可以用一个方程式进行表示，即：

$$g(x)=e^{-X^TX/\sigma^2} \tag{7}$$

根据该模拟算法的相关需求，可以对 $\widetilde{p}^*$ 的相关梯度进行一定的归一化处理，那么方程(7)就可以修改成为：

$$F(p,p^*)=\widetilde{\alpha}\frac{\nabla\widetilde{p}^*}{\widetilde{p}^*} \tag{8}$$

由于该模拟算中的实时烟雾驱动力的大小必须与其实时烟雾在向着目标状态和形状进行转化时的实际烟雾浓度呈正相比，因此可以将其模拟算法中的实时烟雾的驱动力定义为以下的方程式：

$$F(p,p^*)=\widetilde{p}^*\frac{\nabla\widetilde{p}^*}{\widetilde{p}^*} \tag{9}$$

3.3 聚合力

从理论上来说，驱动力即 $F=(p,p^*)$ 可以将实时烟雾的当前状态和形状向期望状态或形状的转变，但是在具体的数值计算的过程中因为存在着随着实时烟雾的实际状态改变的相关数值的衰减，所以仅仅依靠式(3)进行的平行移动或改变无法实现其实时烟雾的状态和形状的改变，这就要求必须要添加一种可以消除烟雾扩散的因素，也就是聚合力，用相关的公式可以表示为：$G(p,p^*)$。

在实际的数值计算中，可以把实时烟雾的当前状态和形状存在的密度与期望状态和形状的密度的差值定义为方程式 $e(x,t)=p(x,t)-p^*(x)$。从其相关的式子中可以看出，其实时烟雾两种不同状态的密度差值越大，代表着相关的数值衰减和损耗就越严重。因此这就要求必须在实时烟雾的转化过程中对其实际存在的密度差进行一定的扩散处理，使其数值等于零，这一举措可以在很大程度上保证其实时烟雾的两种不同状态的质量是守恒的。

在数值计算中，可以在其满足纽曼边界的基础上，对其存在的密度差实行扩散方程的处理，用方程式可以表示为：

$$\begin{cases}\dfrac{\partial e}{\partial t}=\nabla^2 e\\[2ex]\dfrac{\partial e}{\partial n}=0\end{cases} \tag{10}$$

在实时烟雾的目标状态恒定的基础上，将密度差的方程式代入相关的扩散方程，可以得到另一个方程式，即：

$$\frac{\partial p}{\partial t}\nabla^2(p-p^*) \tag{11}$$

在实际的数值运算中，为了最大程度地避免和防止实时烟雾向期望状态和目标转换的过程中出现的扩散作用而导致的相关数值的衰减，可以用方程式表达出其聚合力，即：

$$G(p,p^*)=\nabla^2(p-p^*) \tag{12}$$

同时，为了确保模拟计算中的聚合力只在实时烟雾密度和浓度大的地方出现，可以代入 $\widetilde{p}^*$ 和 p 因素对实时烟雾的密度和浓度进行一定的调节，那么其聚合力就可以表示为：

$$G(p,p^*)=\nabla[p\widetilde{p}^*\nabla(p-p^*)] \tag{13}$$

3.4 基于可编程 GPU 的数值模拟

GPU 是近几年来新兴的一种全新技术手段，用来作为一个流处理器。GPU 可以通过一定的技术手段实现对通用计算机方面的处理。当前技术手段下的可编程 GPU 可以在真正意义上分别实现对其顶点级和像素级的处理，也就是说它们可以进行编程，都可以取 32 位的浮点精度，同时对其顶点级的相关编程还可以进行分支操作和一定的循环操作。而相比于顶点编程来说，像素编程还具备着其顶点编程所不具备的优点：

(1)能够实现对模拟计算中的实时烟雾纹理数据进行一定程度的访问；

(2)其像素编程的相关数据处理能力远远超过顶点级的数据处理能力；

(3)像素编程下的实时烟雾的填充速度也远远高于顶点编程下的三角形传输速度；

(4)相对于顶点编程来说，像素编程可以支持更多数量的渲染管道数。

因此，本文主要研究的可编程 GPU 基础上的实时烟雾模拟研究的全过程均是采用像素编程的手段进行的，具体的过程如下：首先必须对该模拟算法中的所有变量进行一定的纹理化处理，这一过程可以通过在严格按照用户输入的矩形区域的实际大小的基础上进行实时烟雾的离散化处理来实现。这样可以实现模拟算法中的一切变量都纹理化，同时该模拟算法中纹理化的相关变量的每一个纹元的 RGBA 值都可以代表该模拟算法中每一个离散节点上的实时烟雾的速度和密度等参量，可以在时刻 t 的速度场的基础上，根据上文中的公式计算出时刻 $t+\nabla t$ 分布在的整个模拟算法中的速度场[12]。在对其进行一定的二维处理的过程中，上述公式可以在求出其速度的同时计算出其实时烟雾的密度和浓度的分布情况，并且可以由此绘制出实际的最终结果。

由于本文研究的模拟算法是通过将风力因素作为驱动力来促使实时烟雾由目标状态和形状向期望目标状态和形状进行转换的，因此就必须对其计算过程中的 GPU 添加风力项。在实际运算中，可以直接取出与实际的像素程序相对应的数据展开运算，如图 1 中的直接取出项所示，其中的黑色方块部分代表模拟计算中的一个具体的节点变量，相关计算在输入的矩形几何体中进行。

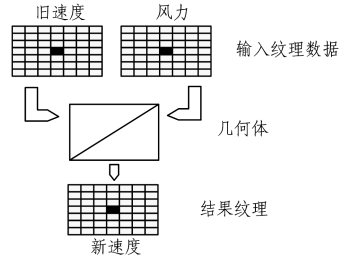

图 1 可编程图形处理单元上风力项的实现过程

3.5　模拟计算中的数值模拟步骤

在上述基础上的数值模拟必须经过一定的步骤才可以真正的实现。

(1)施加实施烟雾向目标状态和形状的驱动力:$\frac{\partial u}{\partial t}=v_f F(p,p^*)$;

(2)添加模拟计算中实际存在的动量衰减项:$\frac{\partial u}{\partial t}=-v_d u$;

(3)计算速度场平移:$\frac{\partial u}{\partial t}=-u\cdot\nabla u$;

(4)利用泊松方程$\nabla^2 p=\nabla u$来计算出模拟计算中的压力场∇p,随后使用其压力场对速度场进行一定的影响和修正,使得其速度场存在的扩散度等于零[13];

(5)在模拟计算中的速度场的影响下,实时烟雾的密度场会发生一定的平移和变化,用方程式表示即为:$\frac{\partial p}{\partial t}=-u\cdot\nabla p$;

(6)对其模拟计算中添加相关的聚合力来防止出现驱动力的存在而导致的实时烟雾扩散现象:$\frac{\partial p}{\partial t}=v_g G(p,p^*)$。

4　可编程 GPU 基础上的实时烟雾模拟的实验结果

本文研究的模拟算法使用 Dell Precision380 工作站(奔腾 3.2GHz CPU,Nvidia Quadro Fx540 显卡,1GB 内存)这款硬件平台来实现,所使用的软件平台较多,分别是开放图形库 OpenGL、可编程图形硬件语言 Cg 和 Visual C++等。本文所实现的实时烟雾状态和形态的转换步骤和具体内容如下:

(1)利用相关的计算机技术制作出两个 raw 格式的图片分别作为实时烟雾的当前状态、形状和期望的状态、形状,其中实时烟雾的当前状态和形状为骑着扫把的女巫,而目标的状态和形状为一只由女巫变化而成的猫。

(2)将制作的 raw 格式的图片放置在预先规定好的网格空间中,同时根据其网格单元的中心点统计出相应网格中的所有像素信息,这样可以形成该网格条件下的实时烟雾浓度,也就是女巫和猫所分别所在的网格包含的烟雾密度和浓度,其中作为背景的网格是不存在烟雾密度的。

(3)根据本文所采用的实时烟雾模拟算法,在相关的 GPU 中计算出每个时间步长∇t后的网格中心的实时烟雾的浓度变化,最终可以得到一个新的、仍旧处于网格中的烟雾密度和浓度分布。

(4)根据通过上述程序得到的新图案的浓度分布情况,并采用一定的插值处理,最终可以得到整个图像中不同像素的具体浓度,并根据其像素的浓度计算和确定其像素的 RGBA 值,最终可以得到一幅全新的图像。

该模拟算法的主要运行程序如图 2 所示。

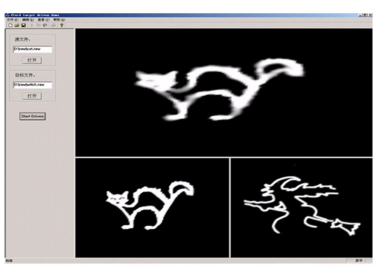

图 2　系统界面情况

图 2 中,实时烟雾初始状态和形状以及期望的状态和形状分别为图片中下半部分的右边部分和左面部分,图片的上半部分则是其转换过程中所呈现的烟雾状态和形状。图 3—图 8 是在实际转换的过程中的截图,整个转换的过程大约历时 3 s,在很大程度上实现了实时转化,在转化的过程中大约生成了 80 帧图像。

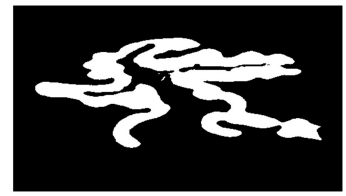

图 3　第 3 帧时的图像

图 4　第 25 帧时的图像

图 5　第 60 帧时的图像

图 6　第 120 帧时的图像

图 7　第 150 帧时的图像

图 8　第 200 帧时的图像

结束语　在我国目前的影视特技和相关的计算机动画中对于实时性的要求非常高,每秒生产的图像不得低于 25 帧,并且要求其处理的对象必须能够对每一帧画面都进行实时的

播放，本文所研究的可编程 GPU 基础上的实时烟雾模拟就可以达到其要求。

本文主要研究了可编程 GPU 基础上的实时烟雾由初始状态和形状向着目标状态和形状转换的全过程，并阐述了其所应用的全部公式。同时在其转换的全过程中应用了先进的 GPU 技术对整个算法进行了加速。本文研究的模拟算法是将其中的实时烟雾的速度和密度整合成了具备一定纹理的颜色通道，然后将其传送平移到相关的像素程序中，最后通过 GPU 完成相关的计算和处理。整个系统可以在 3 s 内生成 80 帧的图像，在很大程度上满足了其实时性的要求和标准，同时还在很大程度上维持了烟雾自然流动的特性，适合应用到交互的动画设计系统中，用以进行实时图形仿真工作。

参 考 文 献

[1] 湛永松，杨明浩，石民勇，等. 基于可编程 GPU 的实时烟雾模拟[J]. 工程图学学报，2008，29(4)：60-64.

[2] 周强，彭俊毅，戴树岭. 基于可编程 GPU 的实时景深模拟[J]. 系统仿真学报，2006，18(8)：2219-2221，2238.

[3] 龙鹏程，罗月童，邹俊，等. 基于可编程 GPU 的可视化技术在中子学分析中的应用研究[J]. 核电子学与探测技术，2010，30(8)：1042-1045.

[4] 柏承双，蒋林，周建伟. 异构 SoC 图形器中可编程剪裁器的设计与实现[J]. 电子技术应用，2015，41(2)：51-54.

[5] 周建伟，韩俊刚，李涛，等. 异构多核 SoC 可编程图形系统硬件设计[J]. 电子技术应用，2014，40(7)：40-43.

[6] 季卓尔，张景峤. 基于可编程 GPU 的骨骼动画算法及其比较[J]. 计算机工程与设计，2008，29(21)：5550-5554.

[7] 包达，罗立民. 基于可编程 GPU 的快速傅立叶变换[J]. 生物医学工程研究，2008，27(2)：111-113.

[8] 唐勇，孙晶，吕梦雅，等. 任意交互路径上烟雾运动实时仿真算法研究[J]. 小型微型计算机系统，2016，37(10)：2334-2337.

[9] 侯美华，吴志红，陈楷民. 基于 OSG 的实时烟雾模拟的研究与应用[J]. 计算机工程与设计，2011，32(6)：2088-2091.

[10] 刘斓，徐健锋，刘群. 基于微粒群优化的烟雾扩散的实时研究[J]. 计算机工程与应用，2012，48(2)：179-181，234.

[11] 陈志宏，张正兰. 动画烟雾动态模拟算法研究[J]. 计算机仿真，2009，26(2)：217-220.

[12] 柳有权，刘学慧，朱红斌，等. 基于物理的流体模拟动画综述[J]. 计算机辅助设计与图形学学报，2005，17(12)：2581-2589.

[13] 汪继文，孟强龙. 一种高效实时的烟雾动画模拟方法[J]. 电子技术，2009，36(11)：54-55.

(上接第 598 页)

[5] GUO Y，ZHOU W，LUO C，et al. Instance-based credit risk assessment for investment decisions in P2P lending[J]. European Journal of Operational Research，2015，249(2)：417-426.

[6] 柳向东，李凤. 大数据背景下网络借贷的信用风险评估——以人人贷为例[J]. 统计与信息论坛，2016，31(5)：41-48.

[7] 罗钦芳，丁国维，傅馨，等. 基于“多层次分类”方法的异常 P2P 网贷借款识别[J]. 管理工程学报，2017，31(3)：201-209.

[8] XIA Y，LIU C，LIU N. Cost-sensitive boosted tree for loan evaluation in peer-to-peer lending[J]. Electronic Commerce Research & Applications，2017，24：30-49.

[9] HE H，GARCIA E A. Learning from Imbalanced Data[J]. IEEE Transactions on Knowledge & Data Engineering，2009，21(9)：1263-1284.

[10] HULSE J V，KHOSHGOFTAAR T M，NAPOLITANO A，et al. An exploration of learning when data is noisy and imbalanced[J]. Intelligent Data Analysis，2011，15(2)：215-236.

[11] CHAWLA N V，BOWYER K W，HALL L O，et al. SMOTE：synthetic minority over-sampling technique[J]. Journal of Artificial Intelligence Research，2002，16(1)：321-357.

[12] BREIMAN L. Bagging predictors[J]. Machine Learning，1996，24(2)：123-140.

[13] 刘巧莉，温浩宇，Hong Qin. P2P 网络信贷中投资行为影响因素研究——基于拍拍贷平台交易的证据[J]. 管理评论，2017，29(6)：13-22.

[14] 陈冬宇，朱浩，郑海超. 风险、信任和出借意愿———基于拍拍贷注册用户的实证研究[J]. 管理评论，2014，26(1)：150-158.

[15] 廖理，吉霖，张伟强. 借贷市场能准确识别学历的价值吗？——来自 P2P 平台的经验证据[J]. 金融研究，2015(3)：146-159.

[16] 曾江洪，李文瀚，陈玺. P2P 借款的损失能挽回吗？——基于拍拍贷的实证研究[J]. 科研管理，2016，37(8)：48-57.

[17] 彭红枫，杨柳明，谭小玉. 地域差异如何影响 P2P 平台借贷的行为——基于“人人贷”的经验证据[J]. 当代经济科学，2016，38(5)：21-34.

[18] 胡晏. 信用等级、借款成功率与违约风险——基于“拍拍贷”数据的经验证据[J]. 投资研究，2017，36(8)：143-158.

[19] WEISS G M，PROVOST F. Learning when training data are costly：the effect of class distribution on tree induction[M]. AI Access Foundation，2003.

[20] 魏瑾瑞，吕晓云. Logistic 模型对非平衡数据的敏感性：测度、修正与比较[J]. 统计研究，2016，33(2)：79-85.