引　论

在高等几何和微积分课程中，读者已学习过许多不同类型的曲线和形状，诸如直线、圆、椭圆和抛物线等。这些形状都有一个共同的基础，也即它们可被定义成代数方程(如 $x^2+y^2=r^2$)或微分方程的解集。虽然我们可以用这些形状来表示周围的世界，但在任何自然物体中，实际上并不存在完美的直线或光滑曲线，它们常常是不规则的，是纯数学曲线的衍生物。因而，虽然这些可微曲线是模拟现实物体的有用近似，但它们不能为某个物体提供完整的模型。

业已证明，有一类数目很大的不可微曲线(不能定义成微分方程解集的曲线)可为许多自然存在的几何形状给出更完整的模型。Benoit Mandelbrot 发明了"分形"(fractal)这个词来描述这些曲线。分形曲线都有一个称为自相似性的共同性质，意思是曲线的子集在放大时看上去就像整条曲线。如果注意看一下四周，就会发现许多存在这类性质的物体。一颗树的每个分枝看上去像它所连接的父分枝，而星座中的一簇星星看上去就像一簇星系。本书中，将介绍如何为许多不同类型的物体建立分形模型，并且还可以了解到如何用这些模型来解决许多不同种类的用更传统的方法难以解决的问题。随书提供的软盘为用户在计算机上研究分形几何的丰富内容提供了必要的工具。

相信读者已看过许多带有分形对象和复数方程的精致而漂亮图片的书籍，可能还正在考虑如何用这些方法来解决实际问题。关于分形图像的编程书早已有许多了，现在在各种书、杂志和日历中都有许多可供探讨、思索和欣赏的图片。实际上，可通过图形说明数学模型的内在美、对称性和复杂性，这种图像的激增也使人们对通常的数学重新产生了浓厚的兴趣。不仅如此，分形除了纯数学外还有许多直接应用。从模拟和虚拟现实到信号处理和数据压缩、再到远程遥感和地理模拟，分形编程技术已在范围广阔的应用领域中日益得到运用。

虽然分形提供了大量数学模拟工具，但绝不意味着只有它们才是可用的。在本书中，将介绍隐含在分形发展背后的概念如何被推广成更一般的方法，概言之即"过程模型"(procedural model)，它允许我们模拟更多的现象。还将介绍如何把图形显示用来表示令人好奇的图形和用作虚拟化，解释用其他方法实际上不能表示的分析结果的强大工具。

分形研究促进了对许多不同类型的复杂系统的研究。许多自然现象——从模拟原子核的相互作用到星系碰撞——对试图求解数目巨大的微分方程的经典方法而言太复杂，以致不能模拟，而且人们也常常只是对这样的系统的定性行为而不是纯定量行为更感兴趣。

正如在本书通篇将要看到的，过程模型提供了一种表示非同一般复杂系统形状的精巧而简洁的方法。实际上，读者将会看到复杂行为是如何用组合简单而且更重要的、易理解的变换集和迭代方程的一小组规则来模拟的。通过在计算机上编码这样的模型，我们既可以静态(看一幅图片)也可以动态地(在屏幕上观察系统演变)观察结果，然后把它们和所要了解的任何系统进行比较。

从非常实际的意义上讲，只有计算机才使分形及与其相应的数学研究成为可能。因为分形曲线是如此复杂，以致用手工来绘制实际上是不可能的。只有计算机才能进行必要的计算

来画一条分形曲线。分形的过程模型从本质上来讲是递归产生的对象。在计算机上有效地表现它们需要用一种很好地支持递归的计算机语言。迄今为止,理想的语言一直是C语言,尤其是C++已成为最理想的面向对象的编程语言,它们为分形编程提供了功能强大的新特征。本书所附软盘充分利用了C++语言面向对象特征的优点,设计出了精巧而高效的代码。

自分形被开始使用以来,它们就一直和计算机图形学紧密地联系在一起。就此面言,本书中几乎所有代码都力图以这种或那种方式来显示图形图像。许多应用程序用分形和过程模型来模拟三维数据集。因此,本书中的软件既包括二维的又包括三维的图形绘制例程。该三维图形软件包使用方便,适用于显示各种各样的图形对象,甚至书中不曾提及的分形对象。此外,该软件包还包括一个用于 visualizing 云和其他无定形三维数据集的以卷提供的软件包。

使用像C++这样而向对象的语言来进行分形编程,是和计算机图形学中的两大最新发展趋势密切相关的:面向对象的编程和分形模拟。对图形尤其是三维图形而言,C++是一种非常好的语言,因为图形是根据已存在对象来描述的。图形对象以某种方式绘制,诸如平移或旋转,以呈现在屏幕上。用面向对象的术语来说,这些操作代表改变对象的"方法"(method)。读者会在书中看到许多使用分形方法定义图形对象,从面产生对象的特定颜色、构造或外观的例子。用分形和过程模型把对象组合在一起,是模拟许多变化多端的系统和复杂图形环境的一种很有效的方法。

0.1 读者所需内容

本书是按便于阅览、自己尝试的方式写的。因此,读者要亲自操作一台 PC 机以练习这些演示程序。随书提供的磁盘上给出了本书软件的完整源代码。为了有效地使用本软件,读者必须已有C或C++编程经验,并可在PC机上编译和执行C++代码。本书所有代码开发都是在 MS-DOS 下用 Borland C++ 3.1 版本进行的,但已尽各种努力以消除代码中的 Borland 具体特征,力求和 AT&T C++ 2.0 标准版本一致。少量与设备相关的图形卡例程是以易适用于其他环境的分立代码模块提供的。为了成功地运行演示程序,读者需要有一台带有浮点协处理器的 286 类型或更好的 PC 机,还需要有一张标准 VGA 图形卡。本软件特别适于用 ATI Graphics Ultra Pro SVGA 卡以获得高分辨率显示和实 24 位映像。其中有一个和描述分形模拟技术的所有章节都有关的演示程序,该程序说明了各个概念并给出了一个引导读者编制自己程序的工作程序例子。把样本程序复制下来,然后就可以用它们进行实验来产生新的效果和增强型演示程序。

0.2 本书的组织

本书由五部分构成。在第一部分中,将介绍书中其余部分所需要的所有编程工具,了解分形研究是如何发展的以及为什么需要所有这些编程工具。第二部分介绍一维和二维的基本分形方法。第三部分研究如何把分形一般化为过程模型,它可应用于建立各种现象类型的更完善和有用的模型。另外,第三部分还介绍了三维图形模块以及如何建立地形的逼真模

型。第Ⅳ部分解释如何在模拟、数字信号压缩等许多应用中使用分形和过程模型。最后，在第五部分中，讨论了过程模拟的发展趋势，重点突出分形可能有助于解决的主要应用和问题。

0.3 历史背景

在第一章中，将介绍导致人们发现分形的若干发展过程。由于概念全是新的，因此，分形的思想并不是突然呈现在人们头脑中。事物总是有前因后果，自 Mandelbrot 首次使用"分形"这个词以来，在把分形模拟方法用于解决各种问题和拓宽基本概念，从而包纳更多应用方面，均已有许多人对分形研究作出了贡献。本章回顾了这些发展，它们导致了分形今天的局面。

0.4 用C++编程

了解了分形历史之后，第二章说明用来编制本书代码的一些C++技术。本书假定用户已具备一些用 C 或 C++编程的经验，第二章阐述用来完成随书提供的磁盘上的全部代码的编程技术和风格。此外，第二章还介绍了许多有用的编程实践，它们将有助于编制更清晰和可读的代码。像任何高级编程语言一样，对于实现某个既定算法，C++也提供了许多可能的途径。第二章提出了一些读者可以参考的指导思想，它们将帮助把本书的代码用于自己的目的。按照这些要点，将能编制高效的、容易理解的代码。

0.5 分形图形

虽然分形算法一般来说常是让人感兴趣的，但它们的主要吸引力还一直是在屏幕上制作图形。第三章描述了在 PC 显示器上显示二维和三维图形的软件工具，对各个C++函数的目的和操作都作了简要阐述。在各章末，对该章中讨论的所有有用的图形函数均作了小结。除了支持本书中的演示程序之外，这些图形函数也为显示读者自己的二维和三维对象提供了工具。用这些图形函数，结合本书后面章节中介绍的工具，便可以制作几乎不限数目的精美图片和动态模拟画面了。

0.6 仿射变换

本书讨论的所有算法，其要素之一是仿射变换，由于它们的简单性和一般性，我们几乎到处使用它们。例如，使用单个变换矩阵的某个步骤可以把图形对象进行平移、旋转或缩放。许多分形算法可以简单地描述成各种仿射变换的重复应用。第三章中，将介绍仿射变换的性质、如何使用它们以及C++如何提供一种容易把矩阵运算插入到代码中去的紧凑而可读的方法。第三章末对在更高级的算法中非常有用的若干实用函数进行了描述。

0.7 统计问题

分形模拟技术的主要应用之一是模拟存在于自然界中的、表面看上去是随机的模式。统计模拟是产生读者可能看到过的绝大多数分形图像的关键组元。第四章中，我们将学习如何产生大量不相关的伪随机数。“伪随机”(pseudorandom)这个词指产生的序列具有这样的性质：在给定用于产生序列的初始条件下，该序列总可以被精确地再次产生。随后我们就可以用这些伪随机数序列来产生具有指定分布的随机值，如高斯分布(钟形曲线分布)或泊松分布(随机点取样)。把这些“随机”数当作输入，就可以产生蔚为壮观的、逼真的二维、三维形状和结构。

0.8 分形曲线

至此我们已具备了所有必要的软件工具，可以开始制作分形图像了。第五章以最简单的情况作为开头：一维，或更确切地说，单参数分形。为了产生分形曲线，引入了自相似性这个基本概念，并把它融入各种方法中。自相似性是指某条曲线在不同尺度下看上去都是“相同”的。分形的原始定义是用“初始元”(initiator，起始形状)和“生成元”(generator，变换该起始形状的规则)来表示的。根据这个基本定义，就能明白更一般的过程模型是如何定义的。这些过程模型允许应用多条规则，产生严格来说不是分形但同样复杂而精致的复杂形状。读者还可以学会如何把随机数引入曲线，从而制作看上去更自然的形状。

我们广泛地使用仿射变换来表明简单操作的组合是如何产生具有高度复杂性和深度的图像的。实际上，许多制作基本分形曲线的生成元规则可视作多个仿射变换的组合应用。读者还可以看到如何把生成元规则推广到非仿射的情况以产生甚至更为复杂和精致的曲线。最后，第五章用一个演示程序作为小结，该程序允许用户交互制作自己的分形形状。

0.9 实心分形

第五章所述的初始元和生成元的基本绘图方法在第六章中又作了推广，它允许模拟由多维实心对象，如 Sierpinski 垫等组成的分形形状。尽管分形方面的大多数题目注重于画分形形状，但也可能以其他方式来使用分形对象，如模拟扩散这样的过程。第六章引入了C++对象类，它允许我们以更一般的方法使用分形对象。在这种方法中，画图只是可能使用分形对象的方式之一。进而，读者将明白为什么从图形对象基类获得分形对象类，并允许组合分形对象，而如果没有C++面向对象的优点，这种组合将是非常困难的。第六章以演示分形对象的柔性为结尾，该演示程序可以交互式地组合对象，产生无穷无尽的、颜色不同的新形状。

0.10 模拟自然

使用分形模拟的主要动力之一是模拟自然物体的模式和特征。在许多植物中可找到具

有此类模式的经典例子。例如，树的每个分枝总是牢牢地生长在它的父分枝上。这种内在的自相似性，使得适合于用自相似分形来模拟植物从单颗种子开始的生长和产生新类型植物的植物演变。第七章介绍了对建立植物生长和植物演变的精确模型方面的研究。使用过程模拟方法，可以在合理近似的程度上模拟包括植物生长在内的许多过程，甚至模拟一颗种子如何最终繁衍覆盖住整个区域。

但植物构造只是用分形来模拟自然的一个小例子。第七章中，我们还将看到两维分形的其他例子，如星系中的星分布等。正如即将要介绍的，有许多种图像可以用分形模拟技术来制作。还有，通过在模型中的控制参数小集合中加入随机变量，即使用同样基本的方法，也可以显著地改变图形的外观。只规定一些参数就能产生如此多种多样的图像的能力，是我们使用分形的主要原因。第七章以一个演示程序结尾，用它可帮助制作一幅分形森林图像，图像的背景是布满分形星星的天空。

0.11 复数显示

前几章所述的过程模型为制作分形曲线提供了一种手段，但它们绝不是唯一的方法。第八章中，将介绍显示屏幕可变换成复平面图。屏幕的每一像素代表一个初始的复数 z。每个值随后被代入一个简单方程以产生新的值，过程不断重复。直至终止条件满足，方程的迭代过程才停止。满足终止条件的迭代次数确定了屏幕上各像素所显示的颜色。通过研究迭代序列的作用，我们就可以了解到所用方程系统的全部特征和稳定性。

第八章开头阐述了显示 Mandelbrot 和互补的 Julia 集的方法。据此读者可以探索当迭代方程改动、终止条件改变及确定像素颜色的显示方法发生变化时所产生的情况。改变迭代规则，即使微小变化，仍能对显示的模式产生令人吃惊的影响。但除了介绍简单地建立深奥的模式之外，还将介绍用于生成这些图像的技术是如何借此来模拟许多现实图像的。第八章以一演示程序结束，它允许用户交互地观察复平面的一些有趣的功能，包括 Mandelbrot 集和它的互补 Julia 集。

0.12 离散分形

有些场合下要用到严格意义上讲不是分形的递归过程模型。尽管如此，这样的模型仍可以产生相当令人感兴趣的、有用的图像，它们可应用于某些不适合用“纯”分形的场合。第九章将介绍使分形绘制过程一般化从而包含更多精致结构和递归规则的各种方法。

大多数分形模型考察的是点连续模式。第九章中，将介绍一些处理只产生离散值的量化分形。第九章中列举了一个离散分形（在曲线的任意点处仅采用少量值且不必连续），给出相当复杂的映射问题解的例子。

第九章还介绍了如何在曲线匹配中运用分形这个问题。经典的曲线匹配问题是寻找一条通过或至少接近已知集的“最合适”的曲线。在模拟随机过程时，我们通常要建立一条和固定点匹配的曲线，那些固定点通常是通过某些外部测量获得的。曲线本身代表了某些已测量到的信号，因此，模型曲线仍必须具有产生被测信号过程的随机或分形特性。第九章展示了在连续和离散情况下建立这类一维曲线的若干方法。

0.13 三维问题

分形模拟的一个主要应用是产生三维对象的构造和外观。第十章列举了三维分形模拟的两个典型应用。第一个是研究如何建立用于模拟地形的逼真模型，包括土地管理和其他地理方面的应用。随后用第九章介绍的技术对地形模型进行了推广，以便把低分辨率的真实地形测量和高分辨率的分形地形结合起来，这为建立实际场所的逼真模型提供了非常强有力的技术。第十一章介绍如何使用随书提供的三维软件包显示我们自己制作的新地形。

地形模拟本质上是个二维分形过程。当然，有许多三维对象对地形模拟技术不适用。例如，第七章所述的植物模拟可以方便地扩展成三维以建立非常逼真的模型。第十章将介绍模拟另一个非常熟悉的自然景物，即云的方法。与较为传统的多边形模拟方法相反，这里的云是用不同尺度的球三维堆积来模拟的。对无定形云而言，这类模拟要比任何多边形模拟合适得多。第十章结尾给出了产生云的一些例子。

0.14 分形应用

尽管上述这些分形模型是有用的结构，但人们仍未看到它们的任何实际应用。在本书的这部分中，将介绍一些用分形和过程模型来解决某些有意思的问题的方法。

0.14.1 生命保险分形

如第七章和第十章所述，我们经常要用分形模型来模拟或“制作”自然景观。在第十一章中，将介绍分形绘图技术是如何用到飞行模拟应用中去的。飞行模拟程序的目标是模拟飞行员在实际飞行器中将要经历的环境。飞行模拟程序视觉系统试图产生对飞行员而言是外部世界的、尽可能逼真的描绘。通过使用这样的视觉系统，飞行员们可以得到现实世界中具体机场的认识训练。因此，该视觉系统需要模拟飞行员们在世界各地许多飞机场上将要看到的景像。第十一章叙述所采用的各种技术，其中用分形模型来制作既逼真又协调的机场景像，而飞行员可能在距飞机场 20 英里远处，也可能在跑道上正在着陆。

为了生成某个地区的分形地图，我们把第九章叙述的关于离散值分形的技术（从小离散值集合中选择的分形算法）推广到二维。这种把实际地图的数字化数据和分形生成特征组合起来的技术，为描绘任意分辨率和尺寸的复杂景观提供了一种非常有效的方法。

在大多数应用中，我们不是通过如第十章所示的改变表面厚度、用分形模拟来模拟表面的粗糙度，就是用它通过调制颜色改变表面的构造。第十一章介绍如何推广模拟技术来改变其他属性，如透明度、亮度和反射性等，从而产生更逼真的效果，如跑道上的着色条或模拟的跑道标识。随后，我们把分形方法与照相这样的其他技术进行了比较，表明为什么使用分形可以大大减少生成此类景观所必需处理的数据量。

0.14.2 越小越好

分形模拟更令人感兴趣的应用之一，是它在减少图像存储容量需求或更简洁地说就是图像压缩方面的应用。通过 Dr. Michael Barnsley 和 Dr. Alan Sloan 的开创性工作，分形研究

人员已认识到，构造自相似图像，比如叶子或植物，可以由一小组点通过仿射变换重复映射而产生。这为减小此类图像的存储容量需求带来了巨大的优势。这样，就不必存储像素阵列，而只需要存储一些起始点和作用到这些点上的变换规则即可。在第十二章中，将介绍压缩和释放一维信号和二维图像的基本技术。

0.14.3 混沌

在过去的五年里，混沌和非线性系统研究迅猛发展，主要是因为可用计算机系统来处理分析此类系统所需要的大量数据。一般来说，混沌系统就是系统的输出对其输入极端敏感的系统。例如，考虑在山顶上平衡的大理石，只要稍加用力，它会很快地从山顶上翻滚下来，翻滚的方式取决于许多难以预测甚至不可能预测的小事件。

第十三章列举了若干常见的混沌系统以及用来分析它们的一些技术。非线性和混沌系统频繁出现在从物理到医学的很多应用领域。许多更为传统的分析技术，如 Fourier 分析，人们观察不到产生所研究信号的过程。正如在第十三章中将要介绍的，混沌研究为分析复杂系统和解释系统的作用提供了另一种更直观的方法。

0.15 未来应用

第十四章介绍如何把全书所述的各种分形模拟和过程模拟技术用于其他应用领域，同时还将介绍这些方法是如何用于其他学科以及分析技术是如何提高的。

直到目前，大多数分形模拟方法的提出和改进都是为了提高计算机生成图像的逼真程度。但是，可以从另一个不同的角度来看待前面所述的棘手问题，这样的看法正在形成，即可以认为，信号或图像表面的复杂性是简单组元复杂的相互作用或重复组合的结果。这和宇宙中所有物质是由质子、电子和中子组成的情况十分相似。分形、过程模型和混沌的应用，就是为了探索控制复杂系统的组元规则。分形提供了这些由生成大量种类的图像和信号所产生的问题的很有希望的研究方法，它们通常和我们在现实世界中看到的情况是一致的。

本书中给出的工具和演示程序为读者自己研究分形过程提供了许多不同的途径。利用二维和三维图形例程，可以制作高度复杂的漂亮图形。过程模拟例程设计中的面向对象方法，允许把该软件方便地用于许多不同的应用。我们的目的是让读者知道，分形绝不只是用来制作漂亮图形，它们还可以应用于许多类问题。目前整个领域尚处于萌芽阶段，还有待进行许多开发和提高。总而言之，正如大量有关文献所表明的那样，分形既是娱乐型的，又是实践型的。因此，可以尽情地使用此软件，去制作书籍和日历上的所有图片，但是请记住，那些所谓的漂亮图形也正在日益得到更多的实际应用。

第一章　分形历史

1975年，Benoit Mandelbrot 在他的《Les Objets Fractals：Forme，Hasard et Dimension》（在1977年翻译成英文是“Fractals：Form，Chance，and Dimension”）一书中最早使分形普及化。在他的书中，Mandelbrot 发明了“分形”（fractal）这个词来描述一类递归定义的，可产生现实和超现实图像的曲线。自该书出版以来，分形在计算机图形方面的研究和应用已达到了这样的程度：即使一般的计算机用户，也会毫无疑问地碰到过以日历、杂志封面或计算机作品等形式出现的分形图像。

分形的高度复杂性和精细结构使其具有很大的艺术价值，不但如此，分形还有许多其他的应用，远不止制作漂亮图片。利用分形算法，可以建立植物、海岸线、山和其他自然对象的逼真模型；而这些用其他方法来模拟是非常困难甚至是不可能的。分形提供了一种紧凑而有效地存储复杂对象详细结构的方法。用少量数据存储某个对象的详细描述的能力，使分形成为存储压缩算法的理想替代者，比如第十二章中介绍的分形图像压缩技术。对分形的深入研究，也促使研究人员去研究相关的领域，像“混沌”（chaos），即显然是复杂方程的随机行为；以及“复杂性”（complexity），即关于系统实际复杂程度特点的研究。本书不但展示了分形的许多应用，而且更为主要的是，为读者提供了应用分形的编程工具。在深入实际分形编程之前，我们先考察一下分形的发展历史，了解分形是如何到达目前状况以及分形的未来是非常必要的。

1.1　开　端

像所有的杰出思想一样，分形也不是神奇地突然出现的，而是建立在许多数学家和其他科学家的努力工作基础之上的。Mandelbrot 把历史片断凝聚起来，发现了一些真正有洞察力的数学知识。但在他的研究之前，实际上已有不少前人的工作，为他的研究奠定了坚实的基础。在 Mandelbrot 发现分形前一百多年，若干数学家就已研究了被认为是连续性和可微性的病态曲线和几何集合。Georg Cantor（1845—1918）就是这样的一位数学家，他的关于病态集的工作为今天的许多数学理论铺平了道路，包括现代集合论的开创。Cantor 进行了集合论的研究，对连续、封闭和稠密这样的性质取得了较深和更严格的理解。为了测试这些集合理论的有效性，促使他研究了这些不同一般的集合。

对平面曲线的经典看法是把曲线上的点(x,y)看作是函数相关的，最简单类型的曲线可定义为$y=f(x)$。引入参数t，并把x和y写成t的函数，可以获得更一般的一维曲线。函数$x(t)$和$y(t)$不是已明确地给定的，就是某个微分方程组的解。19世纪的数学家和物理学家的流行观点是，不论那些方程是如何复杂，任何现象都可以被描述成某组微分方程的解，科学家的任务就是寻找描述所研究的特定系统的方程组。但这种方法的问题是，它会马上促使科学家们不得不去求解极其难以求解的方程组，而如果没有超级计算机来至少找到近似的数值解，求解这些方程将会十分困难。

Cantor 和其他研究人员的努力表明，我们可以构造大量的由有限的一组连续微分方程不能描述的其他类型曲线。所有这些曲线都具有一个共同特征：它们的构造是由递归过程定义的，即从某条称为“初始元”(initiator)的初始曲线，比如一条简单线段开始，然后用一条规则或一组规则，就可以生成曲线的下一阶段。每条规则称为“生成元”(generator)，生成元规则随后作用于新曲线的每个初始元，该过程不断进行，直至无穷。

从效果上来讲，构造这些曲线和在高等几何中所学的构造经典几何对象如等边三角形是相似的。在经典几何中，使用直尺和圆规，根据某个事先确定的过程，就可以从某个起始形状做出新的形状。例如，可以用一条线段构造出一个等边三角形。Cantor 研究了在每个构造步骤中生成的新片断小于起始形状的限制条件下，进行重复构造时所发生的情况。每次一旦生成一组新的片段，就把相同的构造规则作用于这些新的每一个片段，并不断重复。Cantor 感兴趣的是“极限”(limiting)图的性质，也就是说，是无限重复这个构造操作所产生的图的性质。正如他和许多其他人在过去的一百年中所发现的，这些极限曲线具有一些令人非常感兴趣的性质，如具有比产生它们的片断大的维数、模拟像扩散这样的复杂过程的能力等。

1.1.1 关于区间的若干术语

在下面关于 Cantor 集的讨论中，将介绍一些用来规定直线上的区间的数学表示。“开”区间是包含两个端点间所有点但不包括端点本身的集合，用(a,b)表示。例如，(0,1)表示从 0 到 1 的区间，但不包括 0 和 1。“闭”区间包含它的端点，用[a,b]表示。例如，[0,1]包含 0 到 1 间的所有点，同时包括 0 和 1。也可以用[a,b)(包含 a，不包含 b)或(ab](包含 b，不包含 a)表示只包含一个端点的区间。在定义像 Cantor 集这样的复杂集合时，为了确保正确解释每一个点，做这样的区分是必要的，也就是说，一个点不能同时包含于两个分立的区间。

1.2 Cantor 集

Cantor 的成果之一是定义了[0,1]区间上一个具有无数多个点但却不稠密的集合，这个集合就是众所周知的 Cantor 集。集合关于某个点 x 是“稠密的”，是指如果 x 的任意开区间均包含集合中的点(当然，x 必须在该集合中)。例如，区间中的任何有限点集在那个区间中是不稠密的，因为总可以找到一个比两点间的最短距离还要小的区间。“不可数”集是集合中的成员和正整数之间不存在一一对应的集合，意思是无法枚举集中的成员。譬如，在实轴上任何区间都有无数多个点。对某个区间而言，寻找一个可数的无限集并不困难。如对所有大于 0 的整数值 n，序列 1/n 是区间[0,1]的一个可数无限子集，因为除了点 0 附近外，它都是不稠密的。但定义一个不可数的无限子集却比较困难。Cantor 集是由简单的递归过程来定义的：

1. 取区间[0,1]并去掉中间的区间(1/3,2/3)，剩下[0,1/3]和[2/3,1]两个区间；
2. 对剩下的两个区间重复该过程，Cantor 集就是永不会被去掉的点的集合。

图 1-1 举例说明经若干次迭代以后剩下的点。水平线用来表示属于各阶段集合的点集，注意该集在各阶段变得越来越不连续。当然，初看可能不太明显的是，Cantor 集实际上包含无数多个点，即区间[0,1]和 Cantor 集中的点存在一一对应关系。也可以用 Cantor 集构造一些有趣的函数，比如图 1-2 所示的“魔鬼的阶梯”。

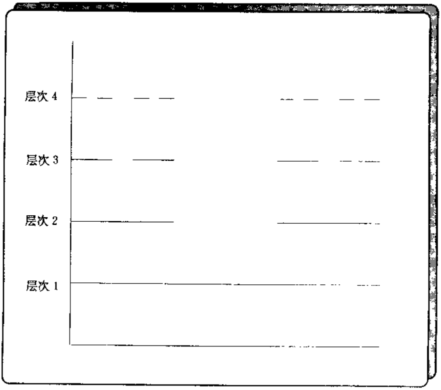

图 1-1 去掉每个区间的中间三分之一而形成的 Cantor 集

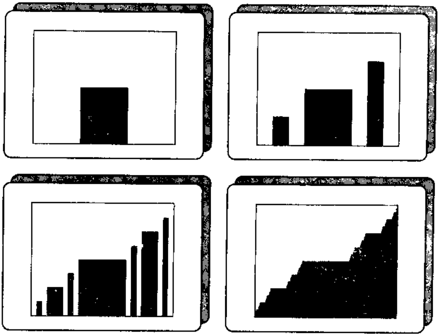

图 1-2 对图 1-1 水平线赋予合适高度而构造成的“魔鬼的阶梯”

"魔鬼的阶梯"是用与构造 Cantor 集相同的过程来递归构成的。每次去掉一个区间，在区间的两个端点间画一条水平线。线的高度按如下确定：如果递归达到层次 k、去掉区间 l，那么该水平线的高度是(2＊l－1)/2k。在第一层次，举例来说，去掉从 1/3 到 2/3 的这个区间，则在高度 1/2 处画一条水平线。而在第二层次，去掉区间[1/9，2/9]，以高度 1/4 画水平线；去掉区间[7/9，8/9]以高度 3/4 画水平线，该过程无限连续进行。所得曲线是一条连续的但在大多数地方是平的分形曲线，因为它完全是由水平线构成的。如果只进行有限次数递归层次的构造过程，则可以得到一组不连续的水平线段。仅仅在无限递归的极限情形下，才能最终获得一条平滑的、从 0 到 1，但没有非水平段的连续曲线。

"魔鬼的阶梯"及其他类似构造曲线的几何性质，常常要比读者可能最为熟悉的代数和微积分中的曲线复杂得多。"魔鬼的阶梯"是一个复杂结构，但在第八章中将会介绍，在显示器上画此曲线及许多其他递归定义曲线的程序却是很容易生成的。

尽管 Cantor 集有许多有趣的性质和变体，但我们对于这类函数的主要兴趣在于它所具有的独特性质：

1. 这些曲线虽有十分复杂的结构，但可以用非常紧凑的方式表示，即只要存储初始元（通常是像直线段这样的简单几何形状）和构造此曲线的生成元即可；
2. 这些曲线比经典几何学中相对平滑的曲线集合更能精确地表示自然物体中所存在的精细形状和表面；
3. 曲线递归定义，通常没有简单的、封闭形式的表达式（也可以用 $y=f(x)$ 这样的形式表示），因而不能用传统的绘图方法绘制。

由它们的定义得知，因为其递归定义，这些曲线难以用一般方法绘制和分析。在 Cantor 那个时代，递归曲线一般被认为是异常的数学奇事，也没有物理上的对应物，因此深入的研究被认为是没有价值的。但设想那个时代，实际上并没有绘制这些曲线的良好方法，更很少用它们来分析问题，所以这也是一种可以理解的态度。因此，实事求是地说，计算机图形的发明是在此类曲线可以被有效研究之前所取得的最重要的进展。

1.3　其他例子

其他类型递归定义形状的一些引人注目的例子是 Koch 曲线（也称雪花曲线），以 Helge von Koch 命名；读者肯定已经在书、杂志和日历上见到过许多这样的例子。它们都遵循同样的基本规则，例如，它们都从某个起始形状开始，然后施加某个变化规则生成下一阶段。

为制作 Koch 曲线，让我们从一直线段开始，然后 apply 图 1-3 所示的生成元，再从该四边形开始，对四边形每边均 apply 该生成元，生成下一阶段。而 Sierpinski 垫是从一个三角形开始的，如图 1-4 所示，然后用一个和构造 Cantor 集非常相似的过程，也即挖去中心的小三角形。在这两种情况下，过程无限进行来产生最终图形。

注意两个对象间的差别在于：Koch 曲线是由一个一维形状代替另一个一维形状生成的。如图 1-3 所示，生成元总是按比例缩小，以至于它可以精确代替每个线段。在第五章中读者将看到，生成元和初始元之间的比例与所得曲线的维数密切相关。和 Koch 曲线不同，Sierpinski 垫使用二维区域作为它的定义。Sierpinski 垫实质上是 Cantor 集的二维形式。

很容易看到，用不同的初始元和生成元，就可以制作出许多此类对象的变体。尽管如此，

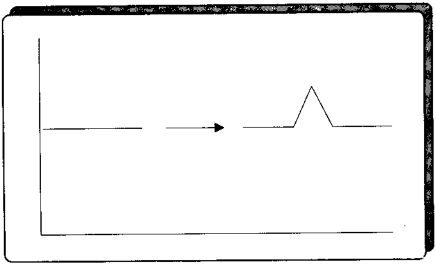

图 1-3　生成 Koch 曲线,也称为雪花曲线

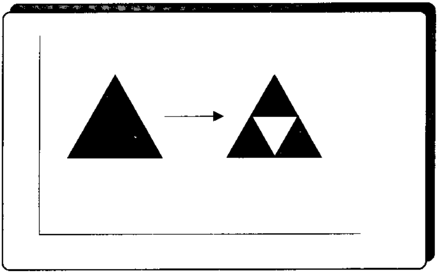

图 1-4　生成 Sierpinski 垫

Koch 曲线、Sierpinski 垫以及它们的无穷无尽的变体,都至少具有一个共同点,即一个称为自相似性的性质。具有自相似性时,若取部分曲线并把它用合适的比例放大,则可以得到原曲线的一个相同的复制品。对 Koch 曲线来说,可以取任意段,放大三倍以获得曲线的复制品。Sierpinski 垫用 2 作为放大因子,这是因为各阶段在垂直和水平方向上都把三角形分成一半。因为对象构造的方式,如本书中所述对象是递归产生对象,自相似性的一个内在性质。

为了进行有用的递归步骤,生成元必须把对象变换成它自身的较小拷贝,然后对每个拷贝重复该操作。因此,生成元必须在它要替换部分的相同位置和位向上进行替换。比如在雪

花曲线中，多边形的每条边均是由与该边位置和位向相同的生成元代替的。同样，Sierpinski垫可以看成是取一个三角形，然后作三个位于该原始三角形中的复制品，相同的操作再作用于每个新形成的三角形。这个过程的每一步均是初始元的一个"仿射变换"(由旋转、缩放、平移和反射组合而成的变换)。这方面的研究产生了一种确定生成元也即一系列作用于对象的仿射变换集合的精巧方法。本书后面章节将解释仿射变换是如何为生成许多应用分形提供非常有效手段的。

1.4 制作空间

像 Koch 曲线这样的递归定义曲线向 19 世纪的许多原则发起了挑战。这些曲线导致的最具革命性的概念之一是改变了维数的直觉定义。大多数人倾向于把一维曲线简单地看作平面图像。这样的曲线没有"厚度"，因此看上去这样的一条曲线将永远不能"充满"平面中某个区域是自然的。一条曲线说是"充满"或覆盖某个区域，指的是能证实该曲线经过区域中每一个点(x，y)。遗憾的是，维数的简单表示在处理无限长曲线时就失效了。1890 年，意大利数学家 Giuseppe Peano 发表了一篇包括连续曲线定义的论文，所提出的连续曲线充满了平面中一个单位正方形。构造这条曲线的方法如图 1-5 所示。

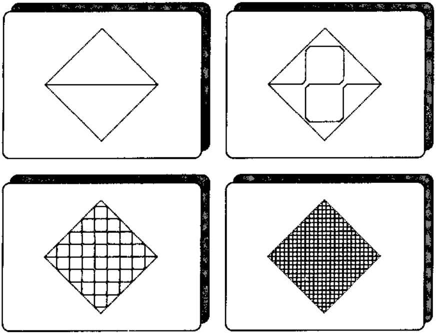

图 1-5 生成平面充填的 Peano 曲线的初始元、生成元和若干次迭代

关于这种基本构造方法，有许多可能的变体。该原始平面充填曲线称为 Peano 曲线。这种基本曲线的构造方法可以很容易地推广到任意有限维数。因此，Peano 曲线可以充满一个三维立方体以及二维正方形。人们会很自然地认为，一维曲线是不可能充满多维空间的。但 Peano 曲线和它的许多变体表明，这种直觉表示对许多分形和递归定义曲线而言是不确切

的。因此，正如Cantor所说明的，定义集合有许多途径；Peano也表明，有许多容易构造的相关曲线可以用来定义集合，即使它们可能因具有一些不平常的性质而难以绘制，诸如不可微连续以及可充满多维空间等。这些性质和在高等微积分课程中所遇到的那些正常的曲线和形状的性质截然不同。

数学维数的表示已由于Peano、Cantor和Sierpinski这样的数学家的努力而发生了彻底改变。目前的结论是，维数是什么有很多种定义方法，究竟选择哪一种取决于所关心的特定问题。我们的主要兴趣是Mandelbrot把分形维数称作什么。第五章将更详细地考察这个概念，目前来说，可以把“分形维数”简单地表示成曲线内在复杂性的度量。例如，Peano曲线的分形维数是2，显示了它的平面充满特性。但就其本身就言，这些研究者的结论看上去仍十分深奥，难以用来解决现实问题。

1.5 自然分形

Mandelbrot在数学领域的主要贡献，就是他认识到了这些递归定义形状在自然界中确实是普遍存在的。他在《The Fractal Geometry of Nature》(自然界的分形几何)一书中，指出了自然存在分形的许多例子。从模拟星系到研究树的叶子和树枝，许多自然界的物体看上去是由在某种程度上和较大物体相似的、较小的东西组合而成的。如在许多种树中，树枝的结构看上去和整颗树的结构非常相似。如果看一下四周，读者还会在所有不同物体中看到许多其他例子。

分形是在任何尺度上都显示自相似性的几何实体。举例来说，可以把Sierpinski垫连续地放大两倍，而最终获得的图形总是一样的。同样，可以把Koch曲线放大三倍，从而获得该曲线的一个新复制品。所有分形都具有一种离散的自相似比例因子，而非分形对象就没有这样的比例因子。如果把平滑的非分形曲线如圆、抛物线或线段的一部分放大，那么放大部分将变得更平滑、更直。而如果把它充分放大，那么放大部分将变成像一条直线。因此，非分形对象总是有一个放大极限，超过此极限，将看不到进一步的结构，如图1-6所示。分形对象没有这样的比例放大极限。无论怎么放大，我们总会看到曲线中更多的、更详细的结构。

自然界中的分形一般不满足任意缩放自相似这个判据。如把某个植物放大足够大的倍数，最后将达到组成植物的分子层次，它当然不具有和植物相同的空间结构。然而，许多物体在若干数量级的尺度上确实显示出自相似性。这个大的自相似尺度范围使这样的物体也十分适合于用分形来模拟。

不像Koch曲线这样严格的几何分形，自然物体不是严格自相似的。也就是说，该物体不是由它自身的较小精确复制品组成的。Mandelbrot引入统计自相似性这个概念，作为自然界物体的更一般和更逼真的模型。在一个“统计自相似的”物体中，组成物体的各部分具有和整体相同的一般结构，只是根据某个比例缩小的精确复制品发生随机变化。从模拟的观点看，这意味着分形的生成元在每个递归层次上是随机变化的。这些变化必须足够小，以保证保持生成元大致看上去仍是相同的。例如，我们可以随机地改变Koch曲线生成元的高度来制作它的一个有趣的变体。整条曲线看上去仍然像Koch曲线，但此时它具有自然物体的更随机的外观特征。统计自相似Koch曲线的一个例子示于图1-7中。

本书将介绍许多统计自相似性的例子。这种模拟复杂对象的方法的魅力是它把一个复

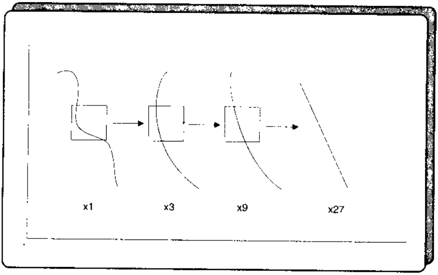

图 1-6　放大光滑的非分形曲线最终形成没有细节或结构的直线

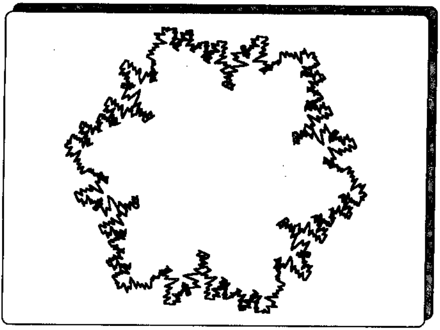

图 1-7　Koch 曲线或雪花曲线，生成器高度随机变化

杂对象分割成了更多可管理的部分(fractal 这个术语就是从拉丁文"frangere"得来的，意思是断裂)。例如，有人只给你一幅 Sierpinski 垫的图像而没有给出任何其他信息，让你写一个把它画在屏幕上的程序，你会觉得这件事很棘手。而分形模拟方法却能提供一种考察和绘制精巧对象的手段。

在更重要的层次上，生命有机体中的分形结构和生长、演变的基因编码规则吻合良好、

Pixar 公司(Lucas Film Productions 的子公司)前总裁 Dr. Alvy Ray Smith 通过模拟植物生长和演变研究推广了这个思想。有机体从一个初始细胞(初始元),诸如树的种子开始生长。种子内的基因代码为复制细胞和使有机体长大给出指令,通过简单地对每个细胞重复基因指令,有机体便能以自然分形或至少是递归的方式不断长大。

存储一个初始元和一套生成元规则,是比试图编码某个最终结果的完整描述更为紧凑的方法。这种方法也允许单个细胞在它们长大时适应环境。显然,至今描述的分形模拟仍是非常简单的,但通过第七章关于模拟植物的讨论,读者将看到非常复杂的结构是如何在只添加更多一些规则(更多基因代码),以及在不同的发展阶段触发不同的规则时逐渐形成的。

1.6 分形和计算机图形

正如我们早已指出的,手工画递归曲线是很困难的。在没有计算机的时代,M. C. Escher 的工作代表了一些最令人感兴趣的递归定义形状研究成果。他的许多画说明了制作具有无限细节图像的无尽可能性。此外,Escher 观察这些类型的图像以及把它们转画到纸上或木头上的能力也是相当独特的。但由于没有画这些形状的简便方法,对那个时代的数学家、艺术家或科学家来说,使分形对象可视化或研究它们都是非常困难的。

今天,分形的流行完全是由于可以利用高速计算机和计算机图形。计算机是在可以接受的时间内绘制分形模型的唯一现实手段。如在后面章节中将看到的,利用递归定义过程,计算机非常适合于画这些形状。要为重要的是,通过允许控制分形参数的相互作用,我们可以方便地研究各种变换。

Mandelbrot 的工作之所以成功的主要原因之一是他的合作者 Dr. Richard Voss 的努力,后者的云、地形和其他的形状,给出了如何应用分形技术和分形模拟技术来制作自然形状的非常逼真模型的一些最早例子。如同科学上的许多其他发展,人们一旦看到了结果,对图形制作技术的兴趣就迅速增强了。

如今,在计算机图形中如此广泛地使用分形模拟是不足为奇的。计算机生成图像的主要目标之一是制作尽可能逼真的图像。在引入分形绘图技术之前,大多数计算机生成的图像看上去非常人工化。物体表面倾向于显得不自然地平滑和闪光或看上去非常多边形化。各种构造技术,如增添随机噪声,通常可用于弥补这个问题。但是,这些方法,即使是作了改进也仍然不能精确地再现自然外观的结构。

1.7 细节的层次

为了弄清为什么许多计算机图形应用需要分形和过程模型,让我们分析一下画草坪透视图这个问题。大多数三维程序要求有一张要画的多边形清单。为画这个草坪,将不得不为草坪上的所有草定义多边形,或定义草的结构。递常,草的每张叶片的位置是不重要的,只要草的整个外观就行了。而且当你在一英里外时,并不需要模拟后再画草的每张叶片。但是,如果摄像机在草坪上时,就可能想要画草的单个叶片。模拟封闭图片中的每张叶片或者为远视图准备众多的多边形,都会影响你画这幅草坪的积极性。

你所缺乏的,是一种根据你离草坪的距离生成恰好数量草叶的方法。分形可以解决这个

问题，它给出了一个在任何所需要的细节层次上都可以被计算出来的模型，而不要求为无用的多边形准备巨大的存储空间。对任何给定的视点，递归计算分形模型，可以直达到显示设备不能分辨相毗邻草叶这个细节层次。如果视点在远处，那么就不必递归计算得太多。但假设视点在近处，就必须把模型计算到产生单片草叶这个层次。用各种统计技术，还可以把随机偏移加到分形模型中以模仿自然变化。假如和分形生成元相比，该随机偏移是小的，那么制作完后，此图像的整个结构看上去就像草。选择合适的随机偏移量，图像将显得更为逼真。

1.8　分形压缩

正如在后边的章节中将要介绍的，分形和过程模型提供了一种定义复杂结构的非常紧凑的方法。例如，Sierpinski 垫模型可以作为三个仿射变换（矩阵）和一个起始三角形（三个矢量）来存储，增添合适的控制以包含随机变化，便可以根据很小的一簇定义参数制作大量结构和形状。该研究促使 Dr. Barnsley 和 Dr. Sloan 提出了分形压缩概念及其各种实现方法。这个思想是直观的。如果能找到定义特定对象的初始元和生成元，那么只要存储这些极少的信息就行了，而不是对象的整个模型。第十二 章将对分形压缩技术所带来的重大意义作更完整的叙述。

1.9　混　沌

计算机的广泛使用所产生的一个问题是，人们通常模糊地认为，计算机得到的结果总是正确的。但实际上，无论什么时候进行浮点计算，计算机只是使用实际数的近似值。不管用多少位来存储数，总是有些精度和准确度损失。在大多数计算中，这常常是不重要的。但在高度复杂的计算或更常见的迭代计算中，这些精度误差可能产生很大的问题。迭代计算就是把计算的输出用作相同计算的输入。这样的迭代计算可以持续某一循环次数，也可以一直达到某种收敛要求为止。一个简单的例子是下面 x^{10}的计算。计算 x^{10}的一种方法如下所示：

```
float x, y;

y = 1.0;
x = 10.0
// Compute  x^10
for(i=0; i<10; i++)
  y *=x;
```

多重运算的输出 y 被当作 for 循环中的输入。这种方法的问题是每次循环时的误差积累。循环次数越多，最终结果的误差就越大；到最后，结果可能变得基本上没什么意义了。直到近来，由于计算机功能已强大到足以通过成千上万次迭代来计算复杂方程系统，精度损失问题才逐渐变得不太重要了。

假定使用有限精度的算术问题不会消失，接下来最好是研究一下系统对其输入到底有