

les de partida. Para establecer una representación de intersección de dos árboles octales, construimos el nuevo árbol utilizando los octantes donde los dos objetos se superponen. De forma similar, en una operación de diferencia, buscamos las regiones ocupadas por un objeto y no por el otro.

Se ha desarrollado un gran número de otros algoritmos de procesamiento de árboles octales. Las rotaciones tridimensionales, por ejemplo, se realizan aplicando las transformaciones a las regiones espaciales representadas por los octantes ocupados. Para localizar los objetos visibles de una escena, podemos determinar en primer lugar si cualquiera de los octantes frontales están ocupados. Si no, proseguimos con los octantes situados detrás de los octantes frontales. Este proceso continúa hasta que los octantes ocupados son localizados en la dirección del punto de vista. El primer objeto detectado en la dirección de cualquier trayectoria de punto de vista a través de los octantes espaciales, desde la parte frontal hasta la parte trasera, es visible, y la información de dicho objeto se puede transferir a una representación mediante un árbol cuaternario para su visualización.

8.22 **ÁRBOLES BSP**

Esta estrategia de representación es similar a la codificación mediante árboles octales, excepto que ahora dividimos el espacio en dos particiones en lugar de ocho en cada etapa. Mediante un árbol de **particionamiento binario del espacio** (binary space-particioning; **BSP**), subdividimos una escena en dos partes en cada etapa mediante un plano que puede estar en cualquier posición y con cualquier orientación. En una codificación mediante un árbol octal, la escena se subdivide en cada paso mediante tres planos perpendiculares entre sí, alineados con los planos de coordenadas cartesianas.

En la subdivisión adaptativa del espacio, los árboles BSP pueden proporcionar un particionamiento más eficiente, ya que podemos posicionar y orientar los planos de corte para acomodarnos a la distribución espacial de los objetos. Esto puede reducir la profundidad de la representación de la escena mediante un árbol, en comparación con un árbol octal y, por tanto, reducir el tiempo de búsqueda en el árbol. Además, los árboles BSP son útiles para identificar la superficie visible y para el particionamiento del espacio en algoritmos de trazo de rayos.

8.23 **MÉTODOS DE GEOMETRÍA FRACTAL**

Todas las representaciones de objetos que hemos considerado en las secciones anteriores usaban métodos de geometría euclídea; es decir, las formas de los objetos se han descrito mediante ecuaciones. Estos métodos son adecuados para describir objetos fabricados: aquellos que tienen superficies suaves y formas regulares. Pero los objetos naturales, tales como las montañas y las nubes, tienen características irregulares o fragmentadas, y los métodos euclídeos no proporcionan representaciones realistas de tales objetos. Los objetos naturales se pueden describir de forma realista mediante **métodos de geometría fractal**, en donde se utilizan procedimientos en lugar de ecuaciones para modelar los objetos. Como es lógico pensar, los objetos definidos mediante procedimientos tienen características bastante diferentes de los objetos descritos con ecuaciones. Las representaciones de geometría fractal de objetos se aplican habitualmente en muchos campos para describir y explicar las características de fenómenos naturales. En gráficos por computadora, utilizamos los métodos fractales para generar visualizaciones de objetos naturales y de sistemas matemáticos y físicos.

Un objeto fractal posee dos características básicas: detalle infinito en cada punto, y una cierta *autosimilitud* entre las partes del objeto y las características totales del objeto. Las propiedades de autosimilitud de un objeto pueden presentar formas diferentes, dependiendo de la representación que elijamos del fractal. Describimos un objeto fractal mediante un procedimiento que especifica una operación repetida para producir el detalle en las subpartes del objeto. Los objetos naturales se representan con procedimientos que se repiten teóricamente un número infinito de veces. Las visualizaciones gráficas de objetos naturales se generan, por supuesto, mediante un número finito de pasos.

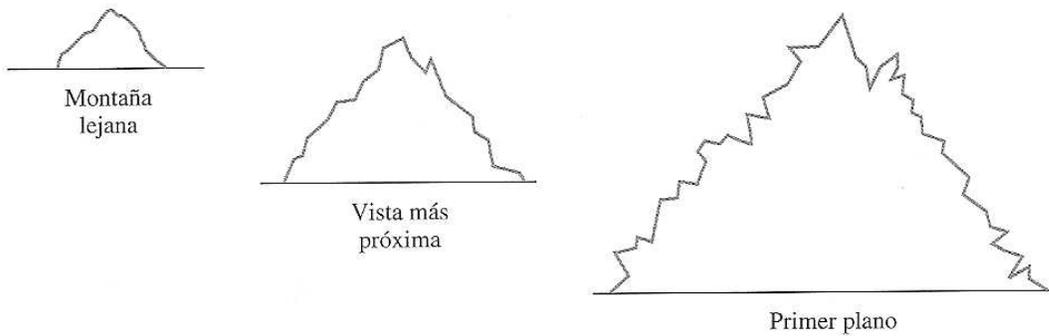


FIGURA 8.67. Apariencia escarpada del contorno de una montaña con diferentes niveles de ampliación.

Si ampliamos una forma continua euclídea, no importa cómo sea de complicada, podemos finalmente obtener la vista ampliada para verla con contornos más suaves. Pero si ampliamos un objeto fractal, continuamos viendo cada vez más detalles en las ampliaciones sin obtener finalmente una apariencia del objeto con contornos suaves. El contorno de una montaña frente al cielo continúa teniendo el mismo aspecto escarpado a medida que la vemos desde una posición cada vez más cercana (Figura 8.67). A medida que nos acercamos a la montaña, el detalle más pequeño de los salientes y de las rocas se hace visible. Acercándonos aún más, vemos los contornos de las rocas, después las piedras y a continuación los granos de arena. En cada paso, el contorno revela más curvas y vueltas. Si tomamos los granos de arena y los analizamos con un microscopio, veríamos de nuevo el mismo detalle repetido a nivel molecular. Formas similares describen las costas y los bordes de plantas y nubes.

Para obtener una vista ampliada de un fractal, podemos seleccionar una parte del fractal para su visualización dentro de un área de visualización del mismo tamaño. Entonces llevamos a cabo las operaciones de construcción del fractal en aquella parte del objeto y visualizamos el mayor detalle de aquel nivel de ampliación. A medida que repetimos este proceso, continuamos visualizando cada vez más detalles del objeto. Como consecuencia del detalle infinito inherente a los procedimientos de construcción, un objeto fractal no tiene un tamaño definido. Cuando añadimos más detalle a la descripción de un objeto, las dimensiones crecen sin límite, pero las amplitudes de las coordenadas del objeto permanecen confinadas dentro de una región finita del espacio.

Podemos caracterizar la cantidad de variación del detalle de un objeto mediante un número llamado *dimensión fractal*. A diferencia de la dimensión euclídea, este número no es necesariamente un entero. La dimensión fractal de un objeto se denomina a veces *dimensión fraccional*, que es la base del nombre «fractal».

Los métodos fractales han resultado ser útiles para modelar una gran variedad de fenómenos naturales. En aplicaciones gráficas, las representaciones fractales se utilizan para modelar el terreno, las nubes, el agua, los árboles y otra flora, las plumas, el pelo y variadas texturas superficiales, y a veces sólo para elaborar patrones atractivos. En otras disciplinas, los patrones fractales se han encontrado en la distribución de estrellas, islas de ríos y cráteres de la Luna; en la lluvia; en las variaciones de la bolsa; en música; en el tráfico; en la utilización de la propiedad urbana; y en los límites de las regiones de convergencia en técnicas de análisis numérico.

Procedimientos para generación de fractales

Un objeto fractal se genera por aplicación repetida de una función de transformación específica a puntos dentro de una región del espacio. Si $\mathbf{P}_0 = (x_0, y_0, z_0)$ se selecciona como posición inicial, cada iteración de una función de transformación F genera niveles sucesivos de detalle mediante los cálculos:

$$\mathbf{P}_1 = F(\mathbf{P}_0), \quad \mathbf{P}_2 = F(\mathbf{P}_1), \quad \mathbf{P}_3 = F(\mathbf{P}_2), \quad \dots \quad (8.105)$$

Por lo general, la función de transformación se puede aplicar a un conjunto de puntos específico o a un conjunto inicial de primitivas, tales como líneas rectas, curvas, áreas de color o superficies. También, pode-

mos utilizar procedimientos deterministas o aleatorios. La función de transformación se podría definir utilizando transformaciones geométricas (cambio de escala, traslación, rotación), o mediante transformaciones no lineales de coordenadas y parámetros de decisión estadísticos.

Aunque los objetos fractales, por definición, contienen detalles infinitos, aplicamos la función de transformación un número finito de veces, y, por supuesto, los objetos que visualizamos tienen dimensiones finitas. Una representación procedimental aproxima un fractal «real» a medida que el número de transformaciones se incrementa para producir cada vez más detalle. La cantidad de detalle incluido en la visualización gráfica final de un objeto depende del número de iteraciones realizadas y la resolución del sistema de visualización. No podemos visualizar variaciones de detalle que sean menores que el tamaño de un píxel. Pero podemos ampliar de forma repetida porciones seleccionadas de un objeto para ver más de sus detalles.

Clasificación de fractales

Los fractales **autosimilares** poseen partes que son versiones reducidas en tamaño del objeto entero. Comenzando por una forma inicial, construimos las subpartes del objeto aplicando un parámetro de escala s a toda la forma. Podemos utilizar el mismo factor de escala s para todas las subpartes, o podemos utilizar factores de escala diferentes en partes del objeto diferentes de escala reducida. Si también aplicamos variaciones aleatorias a las subpartes de escala reducida, se dice que el fractal es *estadísticamente autosimilar*. Las partes entonces tienen las mismas propiedades estadísticas. Los fractales estadísticamente similares se utilizan habitualmente para modelar árboles, arbustos y otras plantas.

Los fractales **autoafines** poseen partes que se forman mediante parámetros de escala diferentes, s_x , s_y , y s_z , según direcciones de coordenadas diferentes. Y podemos también incluir variaciones aleatorias para obtener fractales *autoafines estadísticamente*. El terreno, el agua y las nubes se modelan habitualmente mediante métodos de construcción de fractales autoafines estadísticamente.

Los **conjuntos invariantes de fractales** se forman mediante transformaciones no lineales. En esta clase de fractales se incluyen los fractales *autocuadráticos*, tales como el conjunto de Mandelbrot (formado con funciones cuadráticas en el espacio complejo), y los fractales *autoinversos*, contruidos mediante procedimientos de inversión.

Dimensión fractal

La cantidad de variación de la estructura de un objeto fractal se puede describir mediante un número D , llamado **dimensión fractal**, que es una medida de la aspereza, o de la fragmentación, del objeto. Los objetos con mayor aspecto dentado presentan mayores dimensiones fractales. Un método para generar un objeto fractal consiste en establecer un procedimiento iterativo que utiliza un valor seleccionado de D . Otra técnica es determinar la dimensión fractal a partir de las propiedades deseadas de un objeto, aunque, por lo general, puede ser difícil calcular la dimensión fractal. Los métodos para calcular D se basan en conceptos desarrollados en ramas de las matemáticas, particularmente de topología.

Mediante analogía con la subdivisión de un objeto euclídeo se obtiene una expresión de la dimensión fractal de un fractal autosimilar construido con un único factor de escala s . La Figura 8.68 muestra las relaciones entre el factor de escala y el número de subpartes n de división de un segmento unidad de línea recta, un cuadrado unidad y un cubo unidad. Para $s = \frac{1}{2}$, el segmento de línea unidad (Figura 8.68(a)) se divide en dos subpartes de igual longitud. Con el mismo factor de escala, el cuadrado de la Figura 8.68(b) se divide en cuatro subpartes de igual área, y el cubo (Figura 8.68(c)) se divide en ocho subpartes de igual volumen. En cada uno de estos objetos, la relación entre el número de subpartes y el factor de escala es $n \cdot s^D = 1$. En analogía con los objetos euclídeos, la dimensión fractal D de objetos autosimilares se puede obtener a partir de:

$$ns^D = 1 \tag{8.106}$$

Resolviendo esta expresión en D , la **dimensión de similitud fractal**, tenemos:

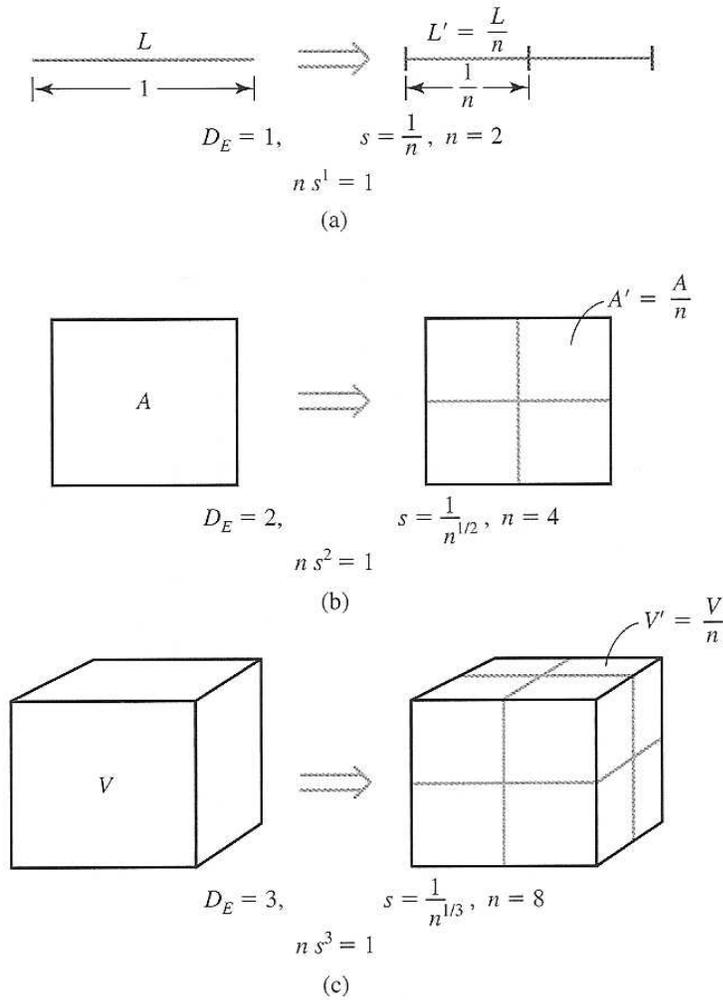


FIGURA 8.68. Subdivisión de una línea unidad (a), un cuadrado unidad (b), y un cubo unidad (c). La dimensión euclídea se representa como D_E , y el factor de escala de cada objeto es $s = \frac{1}{2}$.

$$D = \frac{\ln n}{\ln(1/s)} \tag{8.107}$$

En un fractal autosimilar construido con diferentes factores de escala en las diferentes subpartes del objeto, la dimensión de similitud fractal se obtiene a partir de la relación implícita:

$$\sum_{k=1}^n s_k^D = 1 \tag{8.108}$$

donde s_k es el factor de escala de la subparte k .

En la Figura 8.68, consideramos la subdivisión de formas simples (línea recta, rectángulo y caja). Si tenemos formas más complejas, incluidas las líneas curvas y los objetos con superficies no planas, la determinación de la estructura y las propiedades de las subpartes es más difícil. Para formas de objetos generales, podemos utilizar *métodos de cobertura topológica* que aproximan las subpartes del objeto mediante formas simples. Una curva subdividida, por ejemplo, se podría aproximar mediante secciones de línea recta, y una superficie con *splines* subdividida se podría aproximar mediante pequeños cuadrados o rectángulos. Otras formas de cobertura, tales como círculos, esferas y cilindros, también se pueden utilizar para aproximar las características de un objeto dividido en partes más pequeñas. Los métodos de cobertura se utilizan habitualmente en matemáticas para determinar propiedades geométricas tales como la longitud, el área, o el volumen de un

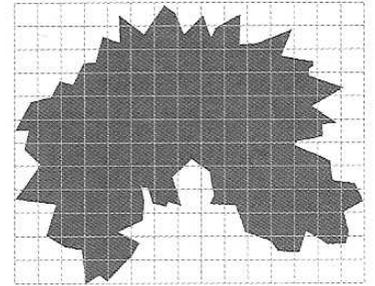


FIGURA 8.69. Caja de cobertura de un objeto de forma irregular.

objeto complejo mediante la suma de las propiedades de un conjunto de objetos de cobertura más pequeños. También podemos utilizar los métodos de cobertura para determinar la dimensión fractal D de algunos objetos.

Los conceptos de cobertura topológica se utilizaron originariamente para ampliar el significado de las propiedades geométricas a formas no estándar. Una ampliación de los métodos de cobertura que utiliza círculos o esferas condujo a la idea de *dimensión de Hausdorff-Besicovitch*, o *dimensión fraccional*. La dimensión de Hausdorff-Besicovitch se puede utilizar como la dimensión fractal de algunos objetos, pero por lo general, es más difícil su evaluación. De forma más habitual, la dimensión fractal de un objeto se estima mediante *métodos de cobertura con cajas* que utilizan rectángulos o paralelepípedos. La Figura 8.69 ilustra la idea de caja de cobertura. Aquí, el área dentro de los límites irregulares se puede aproximar mediante la suma de las áreas de los pequeños rectángulos de cobertura.

Los métodos de cobertura mediante cajas se aplican determinando en primer lugar las amplitudes de las coordenadas de un objeto, después subdividiendo el objeto en un número de pequeñas cajas utilizando los factores de escala proporcionados. El número de cajas n que es necesario para cubrir un objeto se denomina *dimensión de caja*, y n está relacionada con la dimensión fractal D . En objetos autosimilares estadísticamente con un único factor de escala s , podemos cubrir el objeto con cuadrados o cubos. A continuación contamos el número n de cajas de cobertura y utilizamos la Ecuación 8.107 para estimar la dimensión fractal. Para objetos autoafines, cubrimos el objeto con cajas rectangulares, ya que las direcciones diferentes se cambian de escala de forma diferente. En este caso, estimamos la dimensión fractal utilizando tanto el número de cajas n como los parámetros de transformación afín.

La dimensión fractal de un objeto es siempre mayor que la dimensión euclídea correspondiente (o dimensión topológica), que es simplemente el menor número de parámetros necesarios para especificar el objeto. Una curva euclídea tiene una sola dimensión ya que podemos determinar sus puntos con un parámetro, u . Una superficie euclídea es bidimensional, con parámetros de superficie u y v . Y un sólido euclídeo, que requiere tres parámetros en cada especificación de coordenadas, es tridimensional.

En una curva fractal que se encuentra completamente dentro de un plano bidimensional, la dimensión fractal D es mayor que 1 (la dimensión euclídea de una curva). Cuanto más cercano es D a 1, más suave es la curva fractal. Si $D = 2$, tenemos una *curva de Peano*; es decir, la «curva» rellena completamente una región finita del espacio bidimensional. Para $2 < D < 3$, la curva se intersecta a sí misma y el área se podría cubrir con un número infinito de veces. Las curvas fractales se pueden utilizar para modelar los límites de objetos naturales, tales como las costas.

Las curvas fractales espaciales (aquellas que no se encuentran completamente dentro de un único plano) también tienen dimensión fractal D mayor que 1, pero D puede ser mayor que 2 sin intersección consigo misma. Una curva que rellena un volumen del espacio tiene dimensión $D = 3$, y una curva en el espacio que se intersecta a sí misma tiene una dimensión fractal $D > 3$.

Las superficies fractales tienen habitualmente una dimensión dentro del rango $2 < D \leq 3$. Si $D = 3$, la «superficie» rellena un volumen del espacio. Y si $D > 3$, hay una cobertura de superposición del volumen. El terreno, las nubes y el agua se modelan habitualmente con superficies fractales.

La dimensión de un fractal sólido se encuentra habitualmente en el rango $3 < D \leq 4$. De nuevo, si $D > 4$, tenemos un objeto que se autosuperpone. Los sólidos fractales se pueden utilizar, por ejemplo, para modelar las propiedades de las nubes como densidad de vapor de agua o la temperatura dentro de una región del espacio.

Construcción geométrica de fractales deterministas autosimilares

Para construir geoméricamente un fractal determinista (no aleatorio) autosimilar, comenzamos por una forma geométrica dada, llamada *iniciador*. Las subpartes del iniciador se reemplazan a continuación por un patrón, llamado *generador*. A modo de ejemplo, si utilizamos el iniciador y el generador mostrados en la Figura 8.70, podemos construir el patrón de copo de nieve, o curva de Koch, mostrada en la Figura 8.71. Cada segmento de línea recta del iniciador se reemplaza por el patrón del generador, que consta de cuatro segmentos de línea de igual longitud. Después se cambia de escala el generador y se aplica a los segmentos de línea del iniciador modificado, y este proceso se repite un número de pasos. El factor de escala en cada paso es $\frac{1}{3}$, por lo que la dimensión fractal es $D = \ln 4 / \ln 3 \approx 1.2619$. También, la longitud de cada segmento de línea del iniciador se incrementa en un factor $\frac{4}{3}$ en cada paso, de modo que la longitud de la curva fractal tiende a infinito a medida que se añade más detalle a la curva (Figura 8.72). La Figura 8.73 ilustra los patrones adicionales de generador que se podrían utilizar para construir curvas fractales autosimilares. Los generadores de la Figura 8.73(b) y (c) contienen más detalle que el generador de la curva de Koch y tienen mayores dimensiones fractales.

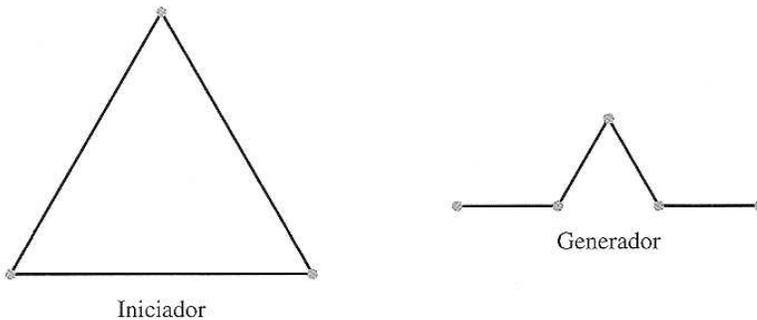


FIGURA 8.70. Iniciador y generador de la curva de Koch.

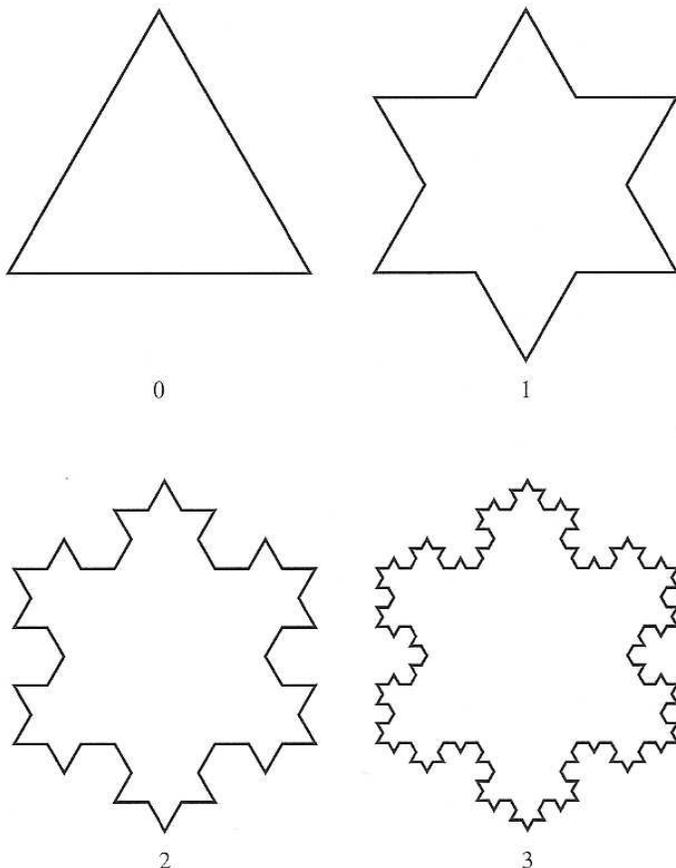


FIGURA 8.71. Tres primeras iteraciones de la generación de la curva de Koch.

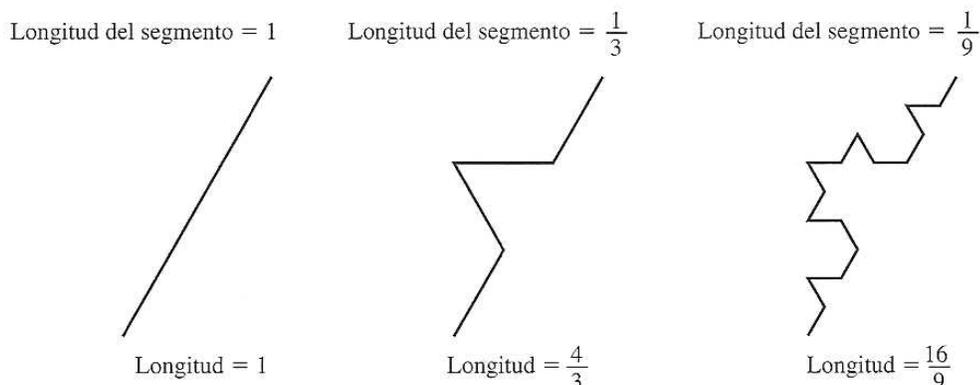


FIGURA 8.72. La longitud de cada lado de la curva de Koch se incrementa en un factor de $\frac{4}{3}$ en cada paso, mientras que las longitudes de los segmentos de línea se reducen en un factor de $\frac{1}{3}$.

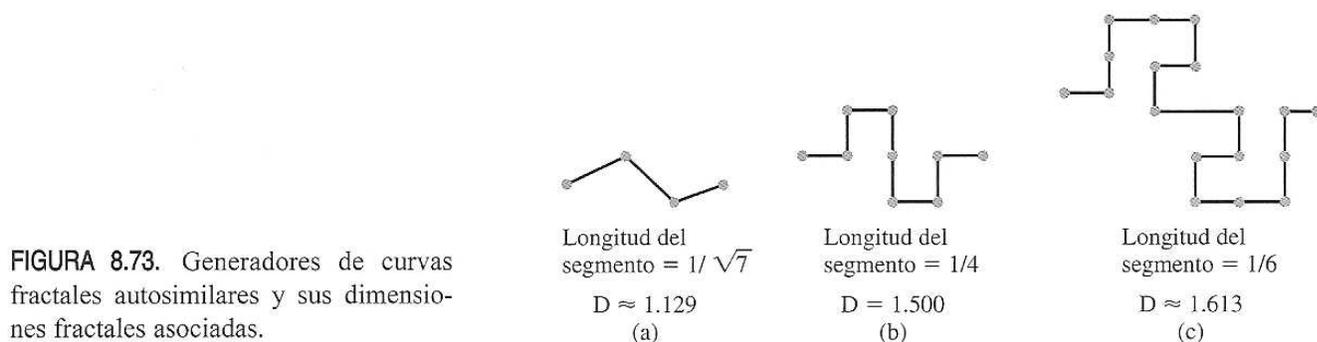


FIGURA 8.73. Generadores de curvas fractales autosimilares y sus dimensiones fractales asociadas.

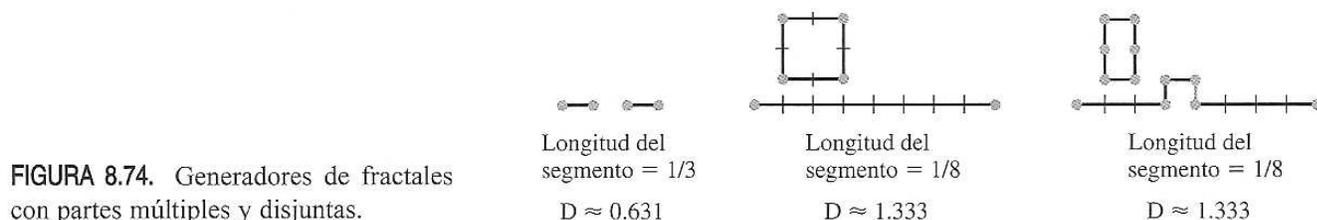
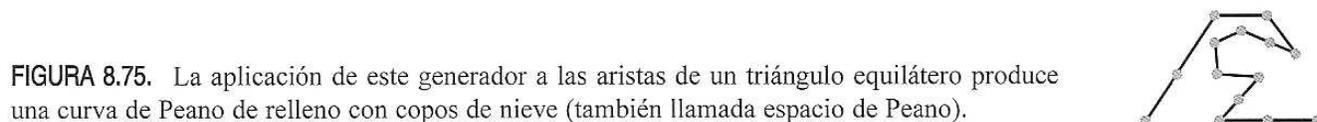


FIGURA 8.74. Generadores de fractales con partes múltiples y disjuntas.



También podemos utilizar generadores con múltiples componentes disjuntos. En la Figura 8.74 se muestran algunos ejemplos de generadores compuestos. Podríamos combinar estos patrones con variaciones aleatorias para modelar varios objetos naturales que presentan múltiples partes desconectadas, tales como distribuciones de islas a lo largo de la costa.

El generador de la Figura 8.75 contiene segmentos de línea de longitud variable y se utilizan múltiples factores de escala para la construcción de la curva fractal. Por tanto, la dimensión fractal de la curva generada se determina a partir de la Ecuación 8.108.

Las visualizaciones de árboles y otras plantas se pueden construir mediante métodos de construcción geométrica autosimilar. Cada rama del contorno del helecho mostrada en la Figura 8.76(a) es una versión cambiada de escala de la forma del helecho total. En la parte (b) de esta figura, el helecho se sombrea totalmente con un giro aplicado a cada rama.

Como ejemplo de la construcción de un fractal autosimilar de las superficies de un objeto tridimensional, cambiamos de escala el tetraedro regular mostrado en la Figura 8.77 con un factor $1/2$, después colocamos el

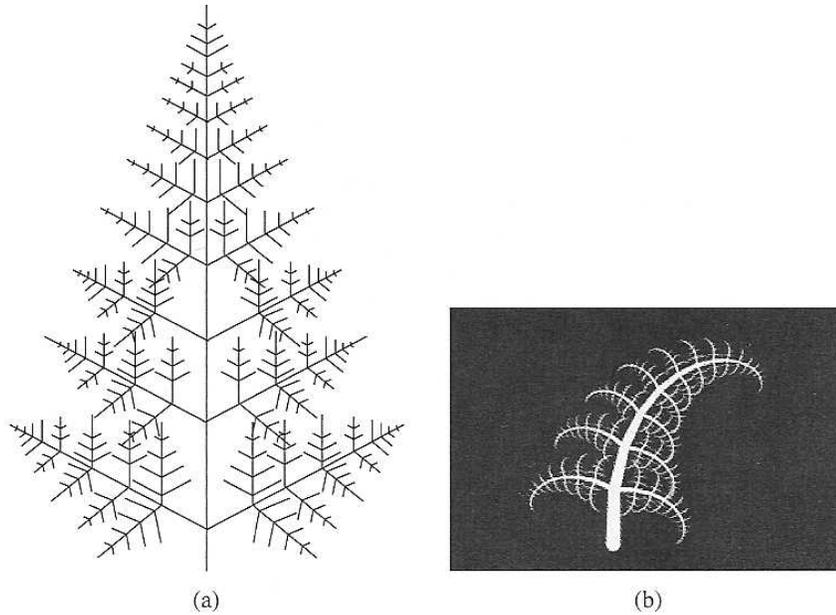


FIGURA 8.76. Construcciones autosimilares de un helecho. (Cortesía de Meter Oppenheimer, Computer Graphics Lab, New York Institute of Technology.)

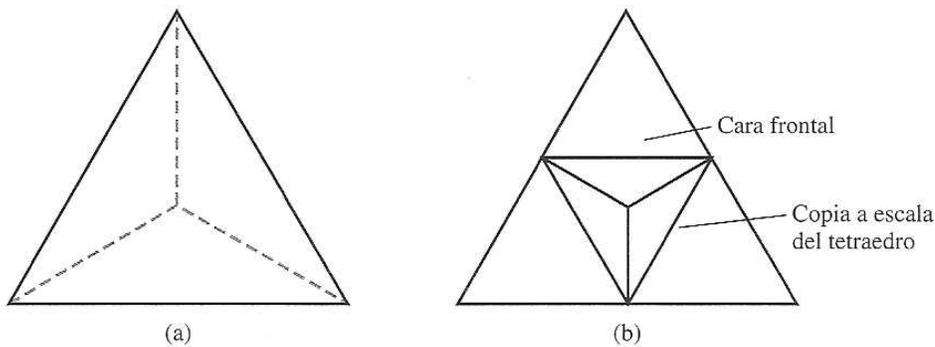


FIGURA 8.77. Cambio de escala del tetraedro (a) en un factor de $1/2$ y posicionando la versión con el cambio de escala aplicado en una cara del tetraedro original produce la superficie fractal mostrada en (b).

objeto que hemos cambiado de escala en cada una de las cuatro superficies originales del tetraedro. Cada cara del tetraedro original se convierte en seis caras más pequeñas y el área de la cara original se incrementa en un factor $3/2$. La dimensión fractal de esta superficie es:

$$D = \frac{\ln 6}{\ln 2} \approx 2.58496$$

que indica que es una superficie bastante fragmentada.

Otro modo de crear objetos fractales autosimilares consiste en perforar agujeros en un iniciador dado, en lugar de añadir más área. La Figura 8.78 muestra algunos ejemplos de objetos fractales creados de este modo.

Construcción geométrica de fractales estadísticamente autosimilares

Para introducir variabilidad en la construcción geométrica de un fractal autosimilar, podríamos seleccionar de forma aleatoria un generador en cada paso a partir de un menú de patrones. O podríamos construir un fractal autosimilar calculando pequeños desplazamientos de las coordenadas mediante pequeñas variaciones aleato-

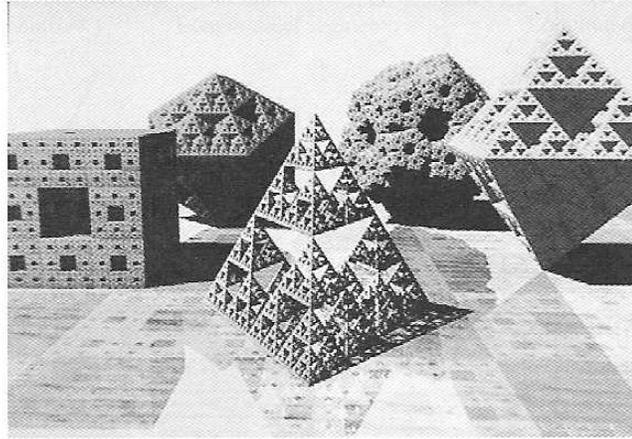


FIGURA 8.78. Fractales autosimilares tridimensionales formados con generadores que sustraen subpartes de un iniciador. (Cortesía de John C. Hart, Department of Computer Science, Universidad de Illinois en Urbana-Champaign.)

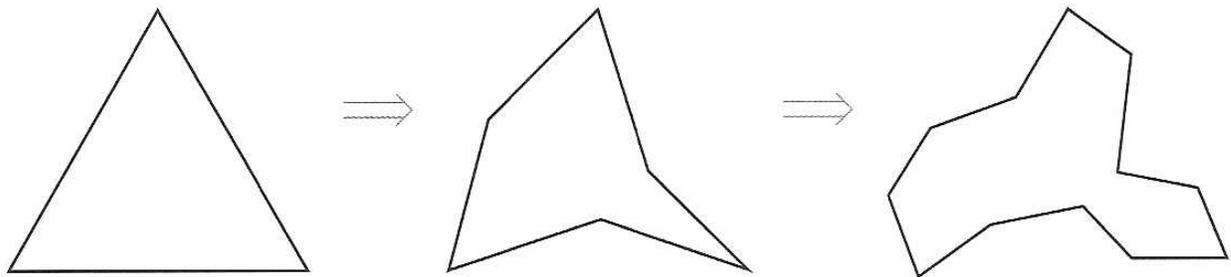


FIGURA 8.79. Un patrón modificado de «copo de nieve» que utiliza un desplazamiento aleatorio del punto medio.

rias. Por ejemplo, en la Figura 8.79 utilizamos una función de distribución de probabilidad para calcular desplazamientos variables del punto medio en cada paso de la creación de un patrón aleatorio de copo de nieve.

En la Figura 8.80 se muestra otro ejemplo de este método. En esta visualización se hace un cambio de escala aleatorio de parámetros y ramificación aleatoria de direcciones para modelar los patrones de la venas de una hoja.

Una vez que se ha creado un objeto fractal, podemos modelar una escena utilizando varias instancias transformadas del objeto. La Figura 8.81 muestra la instanciación de un árbol fractal con rotaciones aleatorias. En la Figura 8.82, se muestra un bosque fractal empleando varias transformaciones aleatorias.

Para modelar las formas nudosas y retorcidas de algunos árboles, podemos aplicar funciones de giro así como de cambio de escala para crear las ramas aleatorias y autosimilares. Esta técnica se ilustra en la Figura 8.83. Comenzando por el cilindro con tapas de la parte izquierda de esta figura, podemos aplicar transformaciones para producir (secuencialmente de izquierda a derecha) una espiral, una hélice y un patrón con giro aleatorio. En la Figura 8.84, se muestra un árbol modelado mediante giros aleatorios. La corteza de árbol de esta visualización se modela utilizando mapas de abultamiento (*bump mapping*) y variaciones fractales brownianas de los patrones de abultamiento. Los métodos de generación de curvas fractales brownianas se estudian en la siguiente sección, y los métodos de mapas de abultamiento se exploran en la Sección 10.17.

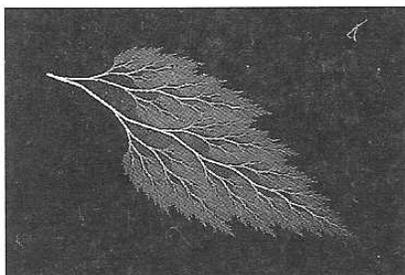


FIGURA 8.80. Construcción aleatoria y autosimilar de ramificación de venas de una hoja en otoño. (Cortesía de Peter Oppenheimer, Computer Graphics Lab, New York Institute of Technology.)

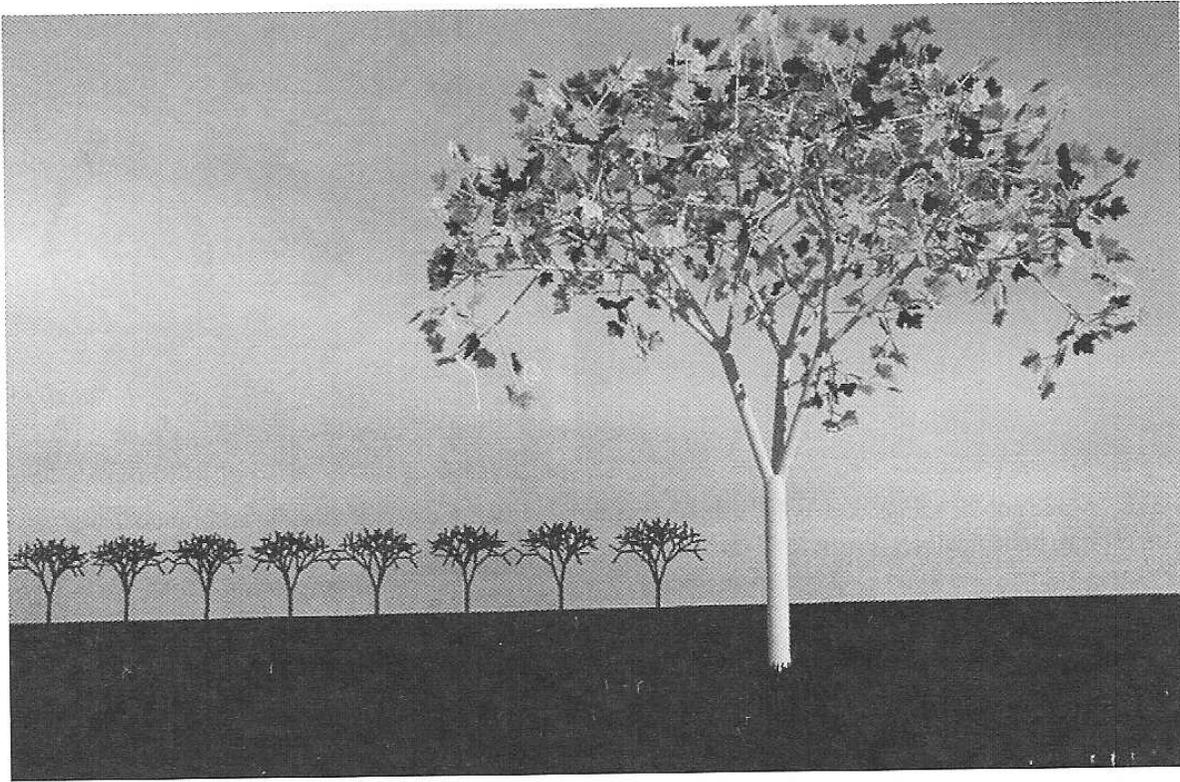


FIGURA 8.81. Modelado de una escena que emplea instanciación múltiple de objetos. Las hojas fractales se añaden a un árbol en posiciones transformadas de forma aleatoria, y varias instancias rotadas y con cambio de escala del árbol se utilizan para formar una arboleda. El césped se modela mediante instancias múltiples de conos verdes. (Cortesía de John C. Hart, Department of Computer Science, Universidad de Illinois en Urbana-Champaign.)

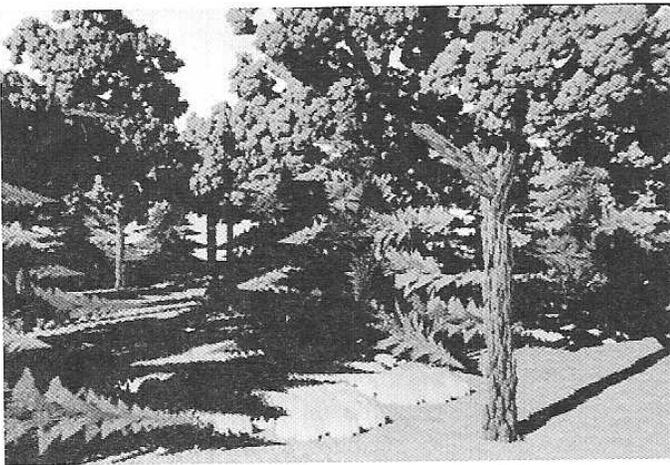


FIGURA 8.82. Un bosque fractal creado con instancias múltiples de hojas, agujas de pino, césped y corteza de árbol. (Cortesía de John C. Hart, Department of Computer Science, Universidad de Illinois en Urbana-Champaign.)

Métodos de construcción de fractales afines

Podemos obtener representaciones altamente realistas del terreno y de otros objetos naturales utilizando métodos con fractales afines que modelan las características de los objetos como el *movimiento browniano fraccional*. Este es una ampliación del movimiento browniano estándar, una forma de «paseo aleatorio», que describe el movimiento en zigzag y errático de partículas de gas u otro fluido. La Figura 8.85 ilustra la trayectoria de un paseo aleatorio en el plano xy . Comenzando por una posición dada, generamos un segmento de línea recta según una dirección aleatoria y con una longitud aleatoria. Otra línea aleatoria se construye a continuación desde el extremo de esta primera línea, y el proceso se repite un número concreto de segmentos de línea.

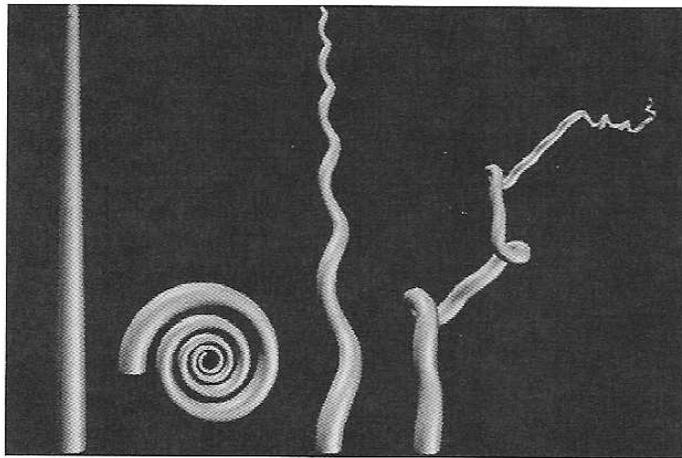


FIGURA 8.83. Modelado de ramas de árboles mediante giros en espiral, helicoidales y aleatorios. (Cortesía de Peter Oppenheimer, Computer Graphics Lab, New York Institute of Technology.)

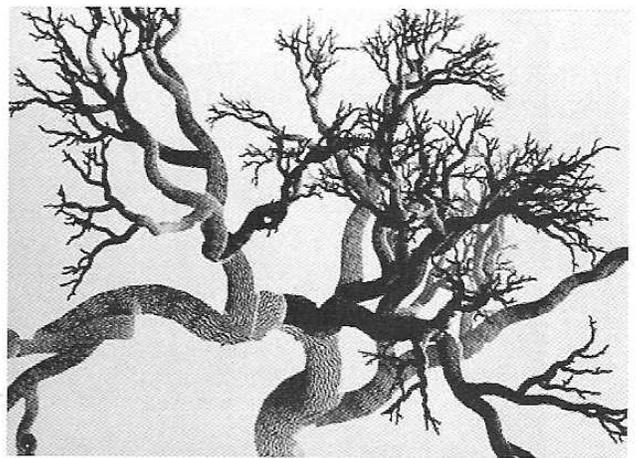


FIGURA 8.84. Ramas de árbol modeladas con serpenteos aleatorios. (Cortesía de Peter Oppenheimer, Computer Graphics Lab, New York Institute of Technology.)

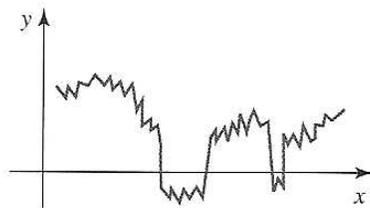


FIGURA 8.85. Un ejemplo de movimiento browniano (paseo aleatorio) en el plano xy .

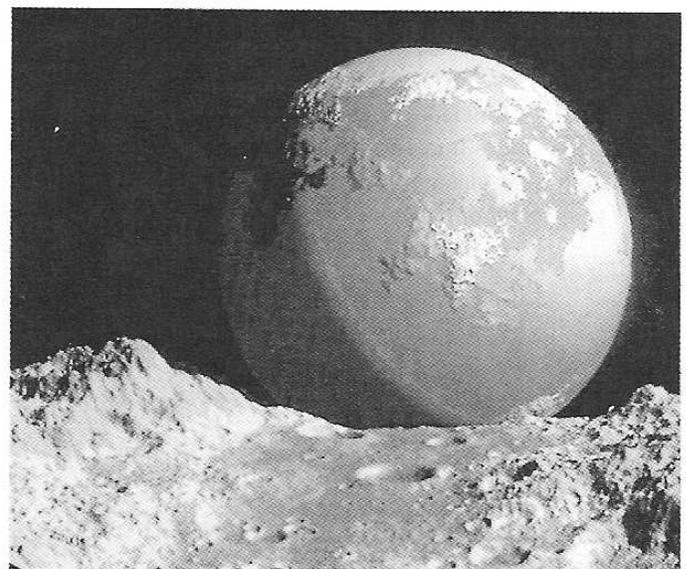


FIGURA 8.86. Un planeta con movimiento browniano observado desde la superficie de un planeta con movimiento browniano, con cráteres añadidos, en primer plano. (Cortesía de R. V. Voss and B. B. Mandelbrot, adaptado a partir de *The Fractal Geometry of Nature* de Benoit B. Mandelbrot (W. H. Freeman and Co., Nueva York, 1983).)

El movimiento browniano fraccional se obtiene al añadir un parámetro adicional a la distribución estadística que describe el movimiento browniano. Este parámetro adicional modifica la dimensión fractal de la trayectoria del «movimiento».

Una única trayectoria browniana fraccional se puede utilizar para modelar una curva fractal. Y con una matriz bidimensional de alturas brownianas fraccionales aleatorias sobre una cuadrícula de un plano de tierra, podemos modelar la superficie de una montaña uniendo las alturas para formar un conjunto de parches de polígonos. Si las alturas aleatorias se generan sobre la superficie de una esfera, podemos modelar las monta-

ñas, los valles, y los océanos de un planeta. En la Figura 8.86 el movimiento browniano se utilizó para crear las variaciones de altura sobre la superficie del planeta. Las alturas se codificaron en color de manera que las alturas más bajas se pintaron en azul (los océanos) y las alturas más elevadas en blanco (nieve sobre las montañas). El movimiento browniano fraccional se utilizó para crear las características del terreno en primer plano. Los cráteres se crearon con diámetros aleatorios y en posiciones aleatorias, utilizando procedimientos con fractales afines que describen fielmente la distribución de los cráteres observados, de las islas de los ríos, de los patrones de lluvia y de otros sistemas similares de objetos.

Al ajustar la dimensión fractal en los cálculos del movimiento browniano fraccional, podemos variar el escarpado de las características del terreno. Los valores de la dimensión fractal próximos a $D \approx 2.15$ producen características realistas en las montañas, mientras que valores próximos a 3.0 se pueden utilizar para crear paisajes extraterrestres de apariencia inusual. También podemos cambiar de escala las alturas calculadas para hacer más profundos los valles e incrementar la altura de los picos de las montañas. En la Figura 8.87 se muestran algunos ejemplos de características del terreno que se pueden modelar mediante procedimientos fractales. En la Figura 8.88 se muestra una escena modelada con nubes fractales sobre una montaña fractal.

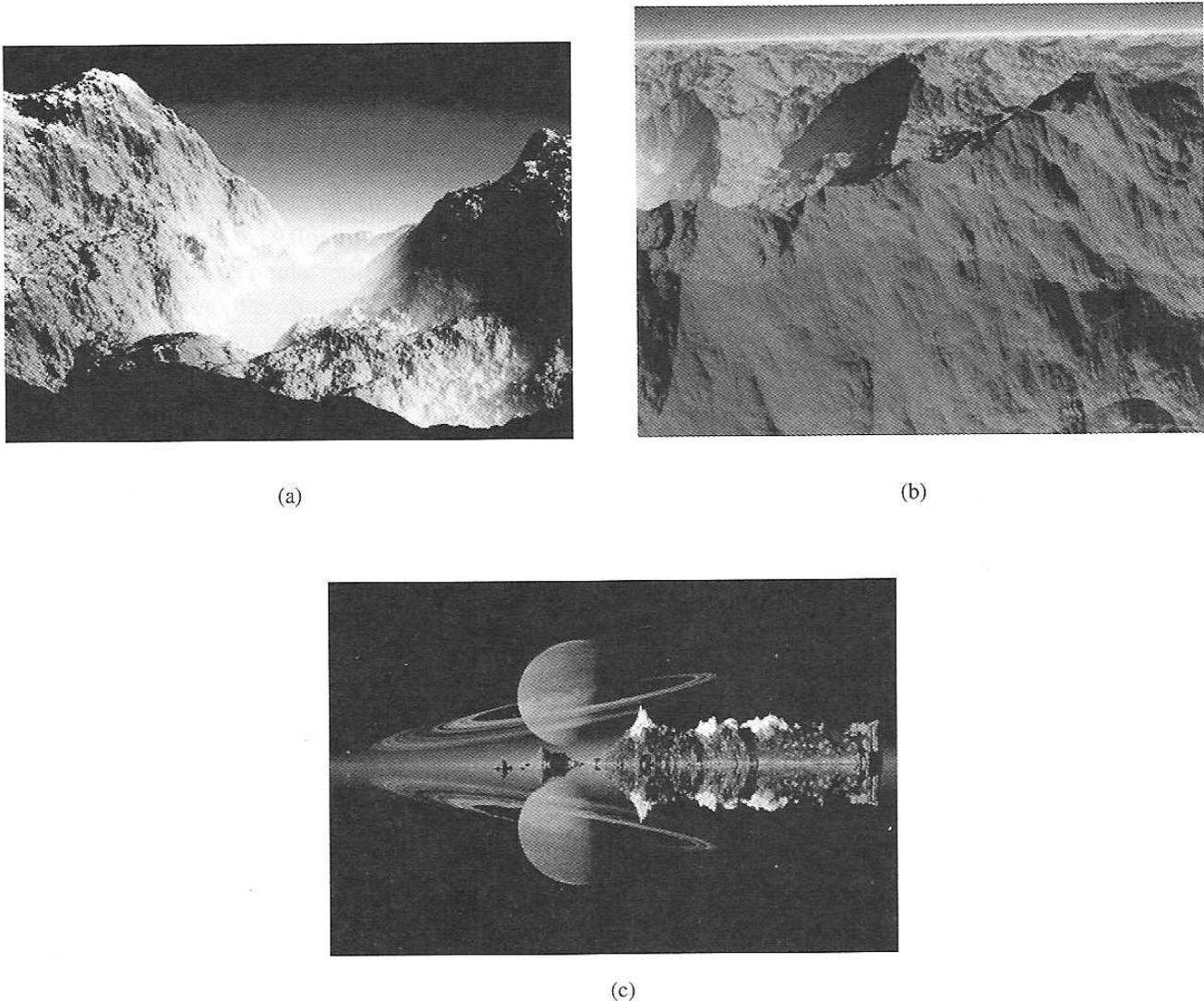


FIGURA 8.87. Variaciones de las características del terreno modeladas mediante movimiento browniano fraccional. (Cortesía de (a) R. V. Voss and B. B. Mandelbrot, adaptado a partir de *The Fractal Geometry of Nature* de Benoit B. Mandelbrot (W. H. Freeman and Co., New York, 1983); y (b) y (c) Ken Musgrave y Benoit B. Mandelbrot, *Mathematics and Computer Science*, Universidad de Yale)



FIGURA 8.88. Una escena modelada mediante nubes fractales y montañas fractales. (Cortesía de Ken Musgrave y Benoit B. Mandelbrot, *Mathematics and Computer Science*, Universidad de Yale.)

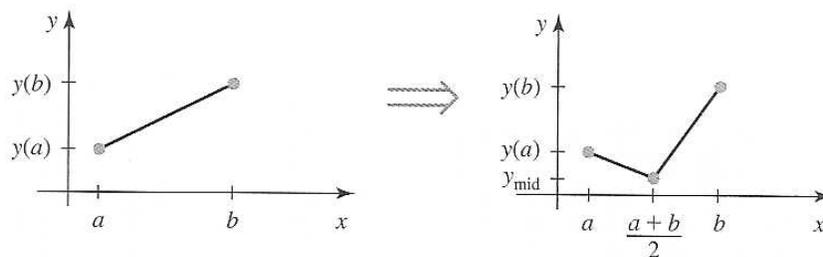


FIGURA 8.89. Desplazamiento aleatorio del punto medio de un segmento de línea recta.

Métodos de desplazamiento aleatorio del punto medio

Los cálculos del movimiento browniano fraccional requieren tiempo, ya que las coordenadas de la elevación del terreno sobre un plano de tierra se calculan mediante series de Fourier, que son sumas de términos con senos y cosenos. Los métodos que utilizan la transformada rápida de Fourier (Fast Fourier Transform; FFT) se utilizan habitualmente, pero son aún un proceso lento para generar escenas con montañas fractales. Por tanto, *métodos de desplazamiento aleatorio del punto medio* más rápidos, similares a los métodos de desplazamiento utilizados en las construcciones geométricas, se han desarrollado para aproximar las representaciones con movimiento browniano aleatorio del terreno y otros fenómenos naturales. Estos métodos se utilizaron originariamente para generar cuadros de animación en películas de ciencia ficción que involucraban características inusuales de terrenos y planetas. Los métodos de desplazamiento del punto medio ahora se utilizan habitualmente en muchas otras aplicaciones de gráficos por computadora, entre las que se incluyen las animaciones para anuncios de televisión.

Aunque los métodos de desplazamiento aleatorio del punto medio son más rápidos que los cálculos del movimiento browniano fraccional, producen características de terrenos de apariencia menos realista. La Figura 8.89 ilustra el método del desplazamiento del punto medio para generar una trayectoria de un paseo aleatorio en el plano xy . Comenzando por un segmento de línea recta, calculamos un valor de la ordenada y desplazado de la posición del punto medio de la línea como el valor medio de los valores de la ordenada y del extremo más un desplazamiento aleatorio:

$$y_{\text{mid}} = \frac{1}{2}[y(a) + y(b)] + r \quad (8.109)$$

Para aproximar el movimiento browniano fraccional, seleccionamos un valor para r de la distribución gaussiana de media 0 y una varianza proporcional a $|b - a|^{2H}$, donde $H = 2 - D$ y $D > 1$ es la dimensión fractal. Otro método para obtener un desplazamiento aleatorio consiste en tomar $r = s_r |b - a|$, donde el pará-

metro s es un factor seleccionado de «rugosidad» de la superficie y r_g es un valor aleatorio gaussiano de media 0 y varianza 1. Se pueden utilizar tablas de búsqueda para obtener los valores gaussianos. El proceso después se repite calculando un valor desplazado de la ordenada y para la posición media de cada mitad de la línea subdividida. Y continuamos la subdivisión para obtener un cierto número de segmentos o hasta que las longitudes de las secciones de la línea subdividida son menores que una longitud seleccionada. En cada paso, el valor de la variable aleatoria r disminuye, ya que es proporcional al ancho $|b - a|$ de la sección de la línea que hay que subdividir. La Figura 8.90 muestra una curva fractal obtenida con este método.

Las características del terreno se generan mediante la aplicación de los procedimientos de desplazamiento aleatorio del punto medio a un plano rectangular de tierra (Figura 8.91). Comenzamos asignando un valor de altura z a cada una de las cuatro esquinas (**a**, **b**, **c**, **d** de la Figura 8.91) del plano de tierra. A continuación, dividimos el plano de tierra por el punto medio de cada arista para obtener los nuevos cinco puntos de la cuadrícula: **e**, **f**, **g**, **h** y **m**. Las alturas en los puntos medios **e**, **f**, **g** y **h** de las aristas del plano de tierra se pueden calcular como la media de la altura de los dos vértices más cercanos más un desplazamiento aleatorio. Por ejemplo, la altura z_e en el punto **e** se calcula utilizando los vértices **a** y **b**, mientras que la altura en el punto medio **f** se calcula utilizando los vértices **b** y **c**:

$$z_e = (z_a + z_b)/2 + r_e, \quad z_f = (z_b + z_c)/2 + r_f$$

Los valores aleatorios r_e y r_f se pueden obtener a partir de una distribución gaussiana de media 0 y varianza proporcional a la separación de la cuadrícula elevada a la potencia $2H$, donde $H = 3 - D$ y $D > 2$. Valores más elevados de D , dimensión fractal de la superficie, producen un terreno con más dientes de sierra, mientras que valores más bajos generan un terreno más suave. También podríamos calcular los desplazamientos aleatorios como el producto de un factor de rugosidad por la separación de la cuadrícula por un valor de la tabla de búsqueda de un valor gaussiano de media 0 y varianza 1. La altura z_m de la posición media del plano de tierra **m** se puede calcular utilizando los puntos **e** y **g**, o los puntos **f** y **h**. De forma alternativa, podríamos calcular z_m utilizando las alturas asignadas a las cuatro esquinas del plano de tierra y un desplazamiento aleatorio del siguiente modo:

$$z_m = (z_a + z_b + z_c + z_d)/4 + r_m$$

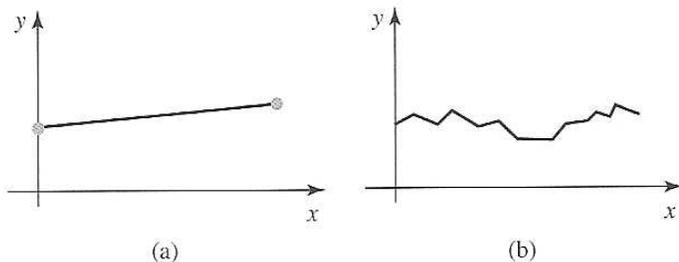


FIGURA 8.90. Una trayectoria de un paseo aleatorio generada a partir de un segmento de línea recta con cuatro iteraciones del procedimiento de desplazamiento aleatorio del punto medio.

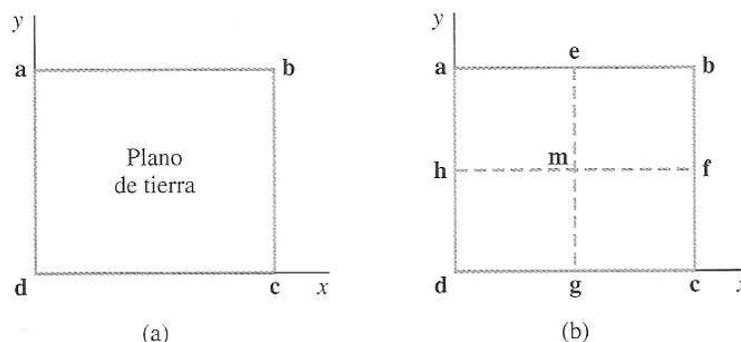


FIGURA 8.91. Un plano rectangular de tierra (a) se subdivide en una cuadrícula de cuatro secciones iguales (b) en el primer paso de un procedimiento de desplazamiento aleatorio del punto medio para calcular las alturas del terreno.

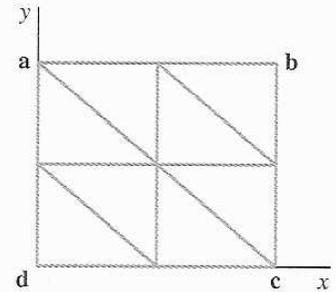


FIGURA 8.92. Ocho parches de superficie formados sobre un plano de tierra en el primer paso de un procedimiento de desplazamiento aleatorio del punto medio para la generación de características del terreno.

Este proceso se repite para cada una de las nuevas cuatro partes de la cuadrícula en cada paso, hasta que la separación de la cuadrícula llega a ser más pequeña que un valor seleccionado.

Los parches triangulares de superficie para la superficie del terreno se pueden formar a medida que las alturas se generan. La Figura 8.92 muestra ocho parches de superficie que se podrían construir en el primer paso de la subdivisión. En cada nivel de recursión, los triángulos se dividen sucesivamente en parches planos más pequeños. Cuando el proceso de subdivisión se ha completado, los parches se somborean utilizando las posiciones seleccionadas de las fuentes de luz, los valores de otros parámetros de iluminación, y los colores y texturas seleccionados de la superficie del terreno.

El método del desplazamiento aleatorio del punto medio se puede aplicar para generar otros componentes de una escena además del terreno. Por ejemplo, podríamos utilizar los mismos métodos para obtener características superficiales de ondas en el agua o patrones de nubes sobre un plano de tierra.

Control de la topografía del terreno

Un modo de controlar la colocación de los picos y de los valles en una escena con terreno fractal que se modela con un método de desplazamiento de punto medio, consiste en restringir las alturas calculadas a ciertos intervalos sobre varias secciones del plano de tierra. Podemos realizar esto designando un conjunto de *superficies de control* sobre el plano de tierra, como se ilustra en la Figura 8.93. Después calculamos una altura aleatoria en cada punto medio de la cuadrícula del plano de tierra que depende de la diferencia entre la altura de control y la altura media calculada para dicho punto. Este procedimiento obliga a las alturas a pertenecer a un intervalo preseleccionado en torno a las alturas de la superficie de control.

Las superficies de control se pueden utilizar para modelar características existentes del terreno en las Montañas Rocosas, o en alguna otra región, construyendo las facetas del plano utilizando las alturas de un plano topográfico de una región concreta. O podríamos establecer las alturas de los vértices de los polígonos de control para diseñar nuestras propias características del terreno. También, las superficies de control pueden tener cualquier forma. Los planos son los más fáciles de utilizar, pero podríamos utilizar superficies esféricas u otras formas curvas.

Utilizamos el método de desplazamiento aleatorio del punto medio para calcular las alturas de la cuadrícula, pero ahora seleccionamos valores aleatorios a partir de una distribución gaussiana en la que la media μ y la desviación estándar σ son funciones de las alturas de control. Un método para obtener los valores μ y σ consiste en hacer ambos proporcionales a la diferencia entre la altura media calculada y la altura de control predefinida en cada punto de la rejilla. Por ejemplo, para el punto de la rejilla e de la Figura 8.91, establecemos la media y la desviación estándar del siguiente modo:

$$\mu_e = z_{c_e} - (z_a + z_b)/2, \quad \sigma_e = s|\mu_e|$$

donde z_{c_e} es la altura de control del punto e del plano de tierra y $0 < s < 1$ es el factor de escala. Los valores pequeños de s , tales como $s < 0.1$, producen una concordancia más ajustada a la envolvente del terreno, y valores más grandes de s permiten mayores fluctuaciones de la altura del terreno.

Para determinar los valores de las alturas de control sobre una superficie de control del plano, en primer lugar, determinamos los valores de los parámetros del plano A , B , C y D . En cualquier punto del plano de tierra (x, y) , la altura en el plano que contiene aquel polígono de control se calcula entoces del siguiente modo:

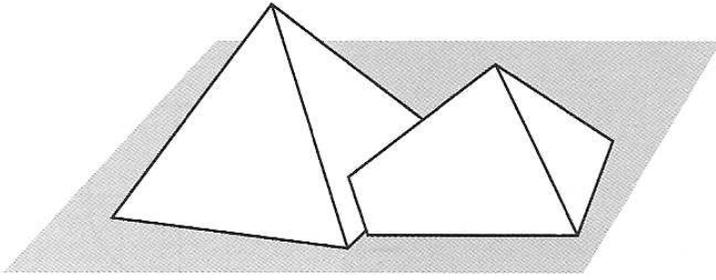


FIGURA 8.93. Superficies de control sobre un plano de tierra.

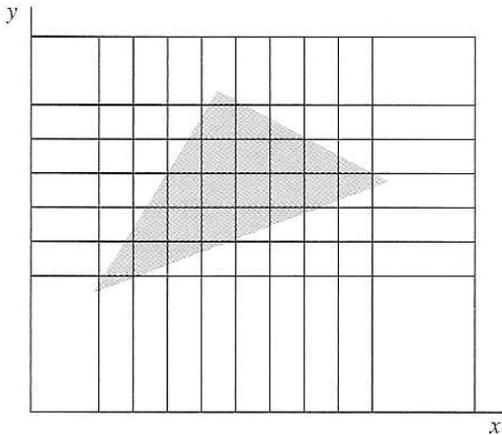


FIGURA 8.94. Proyección de una superficie triangular de control sobre una cuadrícula del plano de tierra.

$$z_c = (-Ax - By - D)/C$$

Se pueden utilizar métodos incrementales para calcular las alturas de control sobre los puntos de la cuadrícula del plano de control. Para llevar a cabo estos cálculos eficientemente, subdividimos el plano de tierra con una cuadrícula más pequeña de puntos xy y proyectamos cada polígono de la superficie de control sobre el plano de tierra, como se muestra en la Figura 8.94. A partir de esta proyección, determinamos los puntos de la cuadrícula que se encuentran por debajo de cada polígono de control. Esto se puede realizar utilizando procedimientos similares a los empleados en el relleno de áreas mediante líneas de barrido. Es decir, para cada «línea de barrido» y de la malla del plano de tierra que cruza las aristas del polígono, calculamos las intersecciones de la línea de barrido y determinamos qué puntos de la cuadrícula están dentro de la proyección del polígono de control. Los cálculos de las alturas de control en estos puntos de la cuadrícula se realizan de forma incremental de este modo:

$$z_{c_{i+1,j}} = z_{c_{i,j}} - \Delta x(A/C), \quad z_{c_{i,j+1}} = z_{c_{i,j}} - \Delta y(B/C), \quad (8.110)$$

donde Δx y Δy son los incrementos de la cuadrícula en las direcciones de los ejes x e y . Este procedimiento es particularmente rápido cuando se aplican métodos de vectores paralelos para procesar los puntos de la cuadrícula del plano de control.

La Figura 8.95 muestra una escena construida utilizando planos de control para estructurar las superficies del terreno, del agua y las nubes sobre un plano de tierra. A continuación, se aplican algoritmos de sombreado de superficies para suavizar las aristas de los polígonos y proporcionar los colores apropiados a la superficie.

Fractales autocuadráticos

Otro método para generar objetos fractales consiste en aplicar repetidamente una función de transformación a los puntos del plano complejo. En dos dimensiones, un número complejo se puede representar como $z = x + iy$, donde x e y son números reales e $i^2 = -1$. En el espacio tridimensional y de cuatro dimensiones, los

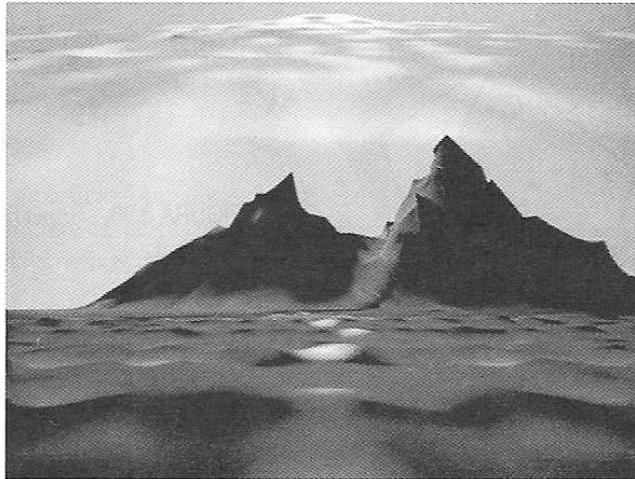


FIGURA 8.95. Una escena compuesta modelada mediante un método de desplazamiento aleatorio del punto medio y superficies planas de control sobre un plano de tierra. Las características de la superficie del terreno, del agua y las nubes se modelaron y sombreadon de forma separada, después se combinaron para formar la imagen compuesta. (Cortesía de Eng-Kiat Koh, Encentuate, Inc., Cupertino, California.)

puntos se representan mediante cuaternios. Una función cuadrática compleja $f(z)$ es aquella que implica el cálculo de z^2 . Podemos utilizar algunas funciones autocuadráticas para generar formas fractales.

Dependiendo del punto inicial seleccionado para iterar, la aplicación repetida de una función autocuadrática producirá uno de los tres resultados posibles:

- El punto transformado puede diverger hacia infinito.
- El punto transformado puede converger hacia un punto límite finito, llamado punto de atracción (atractor).
- El punto transformado permanece en el límite de alguna región.

Como ejemplo, la operación cuadrática no fractal $f(z) = z^2$ en el plano complejo transforma los puntos según su relación con el círculo unidad (Figura 8.97). Cualquier punto z cuyo módulo $|z|$ sea mayor que 1 se transforma mediante una secuencia de puntos que tienden hacia infinito. Un punto con $|z| < 1$ se transforma hacia el origen de coordenadas. Los puntos que están originariamente en el círculo, $|z| = 1$, permanecen en el mismo. Aunque la transformación z^2 no produce un fractal, algunas operaciones cuadráticas complejas generan una curva fractal que es la zona límite entre aquellos puntos que se mueven hacia el infinito y aquellos que tienden hacia un límite finito. Un límite cerrado fractal generado mediante una operación cuadrática se denomina *conjunto de Julia*.

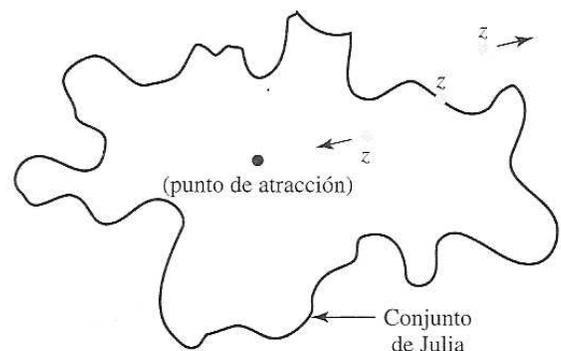


FIGURA 8.96. Posibles resultados de la aplicación repetida de una transformación autocuadrática $f(z)$ al plano complejo, dependiendo de la posición del punto inicial seleccionado.

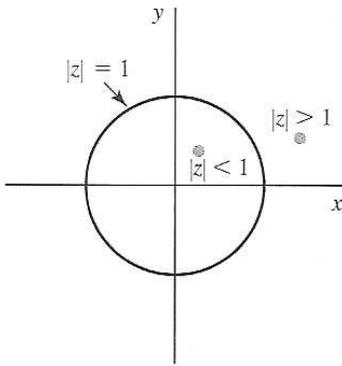


FIGURA 8.97. Un círculo unidad en el plano complejo. La función compleja cuadrática no fractal $f(z) = z^2$ mueve los puntos que se encuentran en el interior del círculo hacia el origen, mientras que los puntos situados fuera del círculo se mueven más lejos del círculo. Cualquier punto inicial del círculo permanece en el círculo.

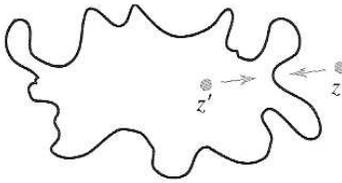


FIGURA 8.98. Localización de la curva frontera fractal utilizando la función auto-cuadrática inversa $z = f^{-1}(z')$.

Por lo general, podemos localizar el límite fractal de una función cuadrática comprobando el comportamiento de los puntos seleccionados. Si un punto se transforma de modo que diverge hacia infinito o converge hacia un punto de atracción, podemos probar con otro punto próximo. Repetimos este proceso hasta que finalmente localizamos un punto del límite fractal. A continuación, la iteración de la transformación cuadrática genera la forma del fractal. En transformaciones simples del plano complejo, un método más rápido para localizar los puntos de la curva fractal consiste en utilizar la inversa de la función de transformación. Entonces un punto inicial seleccionado dentro o fuera de la curva convergerá a un punto de la curva fractal (Figura 8.98).

Una función rica en fractales es la transformación cuadrática:

$$z' = f(z) = \lambda z (1 - z) \tag{8.111}$$

donde λ es una constante compleja. En esta función, podemos utilizar el método de la inversa para localizar la curva fractal. En primer lugar reorganizamos los términos para obtener la ecuación cuadrática:

$$z^2 - z + z'/\lambda = 0 \tag{8.112}$$

La transformación inversa es entonces la fórmula cuadrática:

$$z = f^{-1}(z') = \frac{1}{2} (1 \pm \sqrt{1 - (4z')/\lambda}) \tag{8.113}$$

Utilizando operaciones aritméticas complejas, resolvemos esta ecuación en las partes real e imaginaria de z del siguiente modo:

$$x = \text{Re}(z) = \frac{1}{2} \left(1 \pm \sqrt{\frac{|\text{discr}| + \text{Re}(\text{discr})}{2}} \right) \tag{8.114}$$

$$y = \text{Im}(z) = \pm \frac{1}{2} \sqrt{\frac{|\text{discr}| - \text{Re}(\text{discr})}{2}}$$

donde el discriminante de la fórmula cuadrática es $\text{discr} = 1 - 4z'/\lambda$. Se puede calcular e ignorar unos pocos valores iniciales de x e y (por ejemplo, 10) antes de que empezemos a dibujar la curva fractal. También, ya

que esta función produce dos posibles puntos transformados (x, y) , podemos elegir de forma aleatoria el signo más o menos en cada paso de la iteración siempre y cuando $\text{Im}(\text{discr}) \geq 0$. Cada vez que $\text{Im}(\text{discr}) < 0$, los dos posibles puntos están en el segundo y tercer cuadrante. En este caso, x e y deben tener signos opuestos. El programa siguiente proporciona una implementación de esta función autocuadrática. En la Figura 8.99 se dibujan dos curvas de ejemplo.

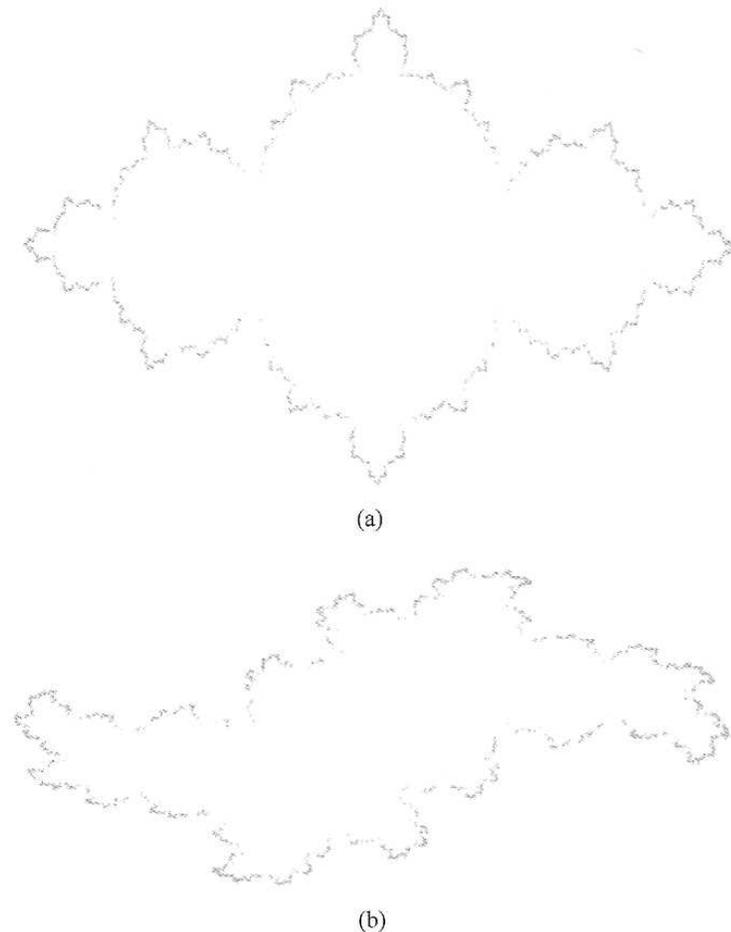


FIGURA 8.99. Dos curvas fractales generadas con la inversa de la función $f(z) = \lambda z(1 - z)$ mediante el procedimiento `selfSqTransf`, utilizando (a) $\lambda = 3$ y (b) $\lambda = 2 + i$. Cada curva se ha dibujado con 10000 puntos.

```
#include <GL/glut.h>
#include <stdlib.h>
#include <math.h>

/* Establece el tamaño inicial de la ventana de visualización. */
GLsizei winWidth = 600, winHeight = 600;

/* Establece los límites de las coordenadas del plano complejo. */
GLfloat xComplexMin = -0.25, xComplexMax = 1.25;
GLfloat yComplexMin = -0.75, yComplexMax = 0.75;

struct complexNum
{
    GLfloat x, y;
};
```

```

void init (void)
{
    /* Establece el color de la ventana de visualización en blanco. */
    glClearColor (1.0, 1.0, 1.0, 0.0);
}

void plotPoint (complexNum z)
{
    glBegin (GL_POINTS);
        glVertex2f (z.x, z.y);
    glEnd ( );
}

void solveQuadraticEq (complexNum lambda, complexNum * z)
{
    GLfloat lambdaMagSq, discrMag;
    complexOum discr;
    static complexOum fourOverLambda = { 0.0, 0.0 };
    static GLboolean firstPoint = true;

    if (firstPoint) {
        /* Calcula el número complejo: 4.0 dividido por lambda. */
        lambdaMagSq = lambda.x * lambda.x + lambda.y * lambda.y;
        fourOverLambda.x = 4.0 * lambda.x / lambdaMagSq;
        fourOverLambda.y = -4.0 * lambda.y / lambdaMagSq;
        firstPoint = false;
    }
    discr.x = 1.0 - (z->x * fourOverLambda.x - z->y * fourOverLambda.y);
    discr.y = {->x * fourOverLambda.y + z->y * fourOverLambda.x};
    discrMag = sqrt (discr.x * discr.x + discr.y * discr.y);

    /* Actualiza z, comprobando para evitar la raíz cuadrada de un número
       negativo. */
    if (discrMag + discr.x < 0)
        z->x = 0;
    else
        z->x = sqrt ((discrMag + discr.x) / 2.0);
    if (discrMag - discr.x < 0)
        z->y = 0;
    else
        z->y = 0.5 * sqrt ((discrMag - discr.x) / 2.0);

    /* Para la mitad de los puntos, utiliza la raíz negativa,
       * situando el punto en el cuadrante 3.
       */
    if (rand ( ) < RAND_MAX / 2) {
        z->x = -z->x;
        z->y = -z->y;
    }
}

```

```

    /* Cuando la parte imaginaria del discriminante es negativa, el punto
    * debería estar en el cuadrante 2 o 4, para invertir el signo de x.
    */
    if (discr.y < 0)
        z->x = -z->x;

    /* Completa el cálculo de la parte real de z. */
    z->x = 0.5 * (1 - z->x);
}

void selfSqTransf (complexNum lambda, complexNum z, GLint numPoints)
{
    GLint k;

    /* Salta los primeros puntos. */
    for (k = 0; k < 10; k++)
        solveQuadraticEq (lambda, &z);

    /* Dibuja el número específico de puntos de transformación. */
    for (k = 0; k < numPoints; k++) {
        solveQuadraticEq (lambda, &z);
        plotPoint (z);
    }
}

void displayFcn (void)
{
    GLint numPoints = 10000; // Establece el número de puntos que hay que dibujar.
    complexNum lambda = { 3.0, 0.0 }; // Establece el valor complejo de lambda.
    complexNum z0 = { 1.5, 0.4 }; // Establece el punto inicial del plano
                                // complejo. */

    glClear (GL_COLOR_BUFFER_BIT); // Borra la ventana de visualización.

    glColor3f (0.0, 0.0, 1.0); // Establece el color de los puntos en azul.

    selfSqTransf (lambda, z0, numPoints);
    glFlush ( );
}

void winReshapeFcn (GLint newWidth, GLint newHeight)
{
    /* Mantiene una relación de aspecto de 1.0, asumiendo que
    * el ancho de la ventana compleja = altura de la ventana compleja.
    */
    glViewport (0, 0, newHeight, newHeight);

    glMatrixMode (GL_PROJECTION);
    glLoadIdentity ( );
}

```

```

    gluOrtho2D (xComplexMin, xComplexMax, yComplexMin, yComplexMax);

    glClear (GL_COLOR_BUFFER_BIT);
}

void main (int argc, char** argv)
{
    glutInit (&argc, argv);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
    glutInitWindowPosition (50, 50);
    glutInitWindowSize (winWidth, winHeight);
    glutCreateWindow ("Self-Squasing Fractal#");

    init ( );
    glutDisplayFunc (displayFcn);
    glutReshapeFunc (winReshapeFcn);

    glutMainLoop ( );
}

```

En la Figura 8.100 se proporciona un gráfico tridimensional de las variables x , y y λ correspondiente a la función autocuadrática $f(z) = \lambda z(1 - z)$, donde $|\lambda| = 1$. Cada sección recta de este gráfico es una curva fractal en el plano complejo.

Otra operación cuadrática que produce una gran variedad de formas fractales es una transformación z^2 ligeramente modificada. En este caso, el fractal es el límite de la región alrededor del conjunto de valores complejos z que no divergen frente a la transformación cuadrática:

$$\begin{aligned}
 z_0 &= z \\
 z_k &= z_{k-1}^2 + z_0 \quad k = 1, 2, 3, \dots
 \end{aligned}
 \tag{8.115}$$

Por tanto, en primer lugar seleccionamos un punto z del plano complejo, después calculamos el punto transformado $z^2 + z$. En el paso siguiente, calculamos el cuadrado de este punto transformado y se lo añadimos al valor original de z . Repetimos este procedimiento hasta que podamos determinar si la transformación es divergente o no.

Los matemáticos han sido conscientes de las características inusuales de tales funciones cuadráticas durante algún tiempo, pero estas funciones eran difíciles de analizar sin computadoras. Después del desarrollo de la computadora digital, el límite de convergencia de la transformación 8.115 se dibujó con una impresora de líneas. A medida que las capacidades de las computadoras digitales se incrementaron, fue posible una investigación gráfica más profunda. Posteriormente, utilizando técnicas de gráficos por computadora más sofisticadas, Benoit Mandelbrot estudió ampliamente esta función, y el conjunto de puntos que no divergen frente a la transformación 8.115 se conoce como el **conjunto de Mandelbrot**.

Para implementar la transformación 8.115, en primer lugar seleccionamos un área rectangular del plano complejo. Los puntos de este área se mapean a continuación a píxeles codificados en color dentro de una ventana de visualización de un monitor de vídeo (Figura 8.101). Los colores de los píxeles se eligen según la velocidad de divergencia del punto correspondiente del plano complejo frente a la transformación 8.115. Si el módulo del número complejo es mayor que 2, entonces divergirá rápidamente cuando se calcula repetidamente su cuadrado. Por tanto, podemos establecer un bucle para repetir las operaciones de cálculo del cuadrado

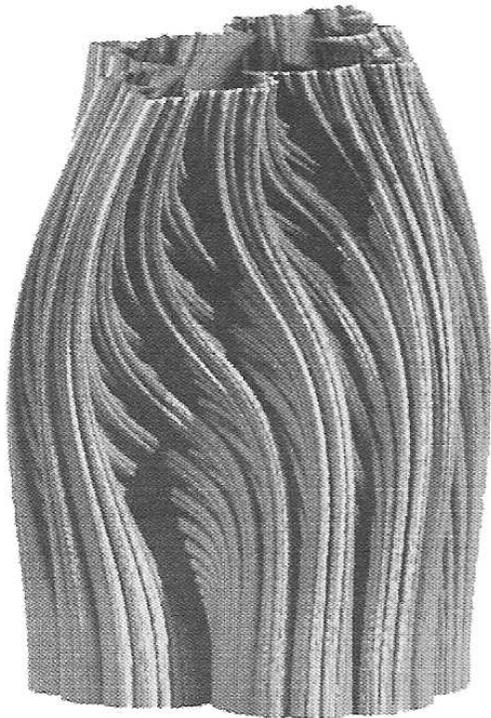


FIGURA 8.100. La función $f(z) = \lambda z(1 - z)$ dibujada en tres dimensiones, con valores normalizados de λ que varían según el eje vertical. (Cortesía de Alan Norton, IBM Research.)

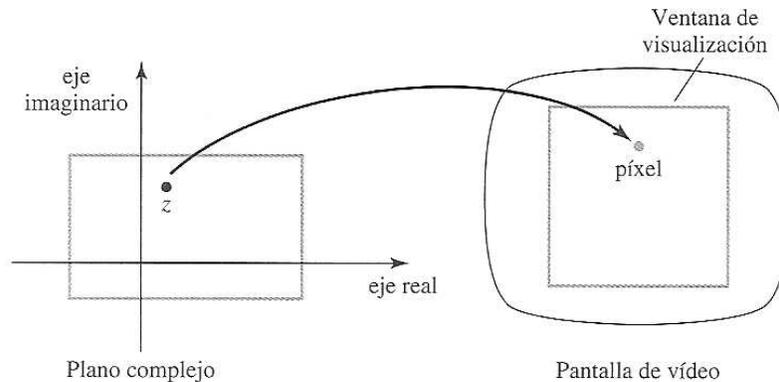


FIGURA 8.101. Mapeo de puntos desde un área rectangular del plano complejo a píxeles codificados con color dentro de una ventana de visualización.

hasta que el módulo del número complejo sea mayor que 2 o hayamos alcanzado un número preseleccionado de iteraciones. El número máximo de iteraciones depende de la cantidad de detalle que queramos para visualizar y del número de puntos que haya que dibujar. Este valor se establece a menudo en algún valor entre 100 y 1000, aunque se pueden utilizar valores más bajos para acelerar los cálculos. Con valores más bajos del límite de iteración, sin embargo, tendemos a perder cierto detalle a lo largo de los límites (conjunto de Julia) de la región de convergencia. Al final del bucle, seleccionamos un valor de color según el número de iteraciones ejecutadas en el bucle. Por ejemplo, podemos colorear de negro el píxel si el número de iteraciones alcanza el valor máximo (un punto no divergente), podemos colorear el píxel de rojo si el número de iteraciones es próximo a 0. Se pueden seleccionar otros valores de color según el valor del número de iteraciones dentro del intervalo que varía desde 0 al valor máximo. Eligiendo diferentes mapeos de color y diferentes partes del plano complejo, podemos generar una gran variedad de visualizaciones dramáticas de los puntos de la vecin-

dad de la frontera fractal que encierra los puntos no divergentes. En la Figura 8.102(a) se muestra una elección para codificar con color los píxeles de la región alrededor del conjunto de Mandelbrot.

En el programa siguiente se proporciona una implementación de la transformación 8.115 para visualizar el conjunto de puntos de convergencia y sus fronteras. La parte principal del conjunto de convergencia está contenido dentro de la siguiente región del plano complejo.

$$-2.00 \leq \operatorname{Re}(z) \leq 0.50$$

$$-1.20 \leq \operatorname{Im}(z) \leq 1.20$$

Podemos explorar los detalles a lo largo de la frontera del conjunto de Mandelbrot seleccionando regiones rectangulares del plano complejo sucesivamente más pequeñas de modo que podamos ampliar áreas seleccionadas de la visualización. La Figura 8.102 muestra una visualización codificada en color (aunque la figura se muestra en blanco y negro) de la región alrededor del conjunto de convergencia y una serie de ampliaciones que ilustran algunas características notables de esta transformación cuadrática.

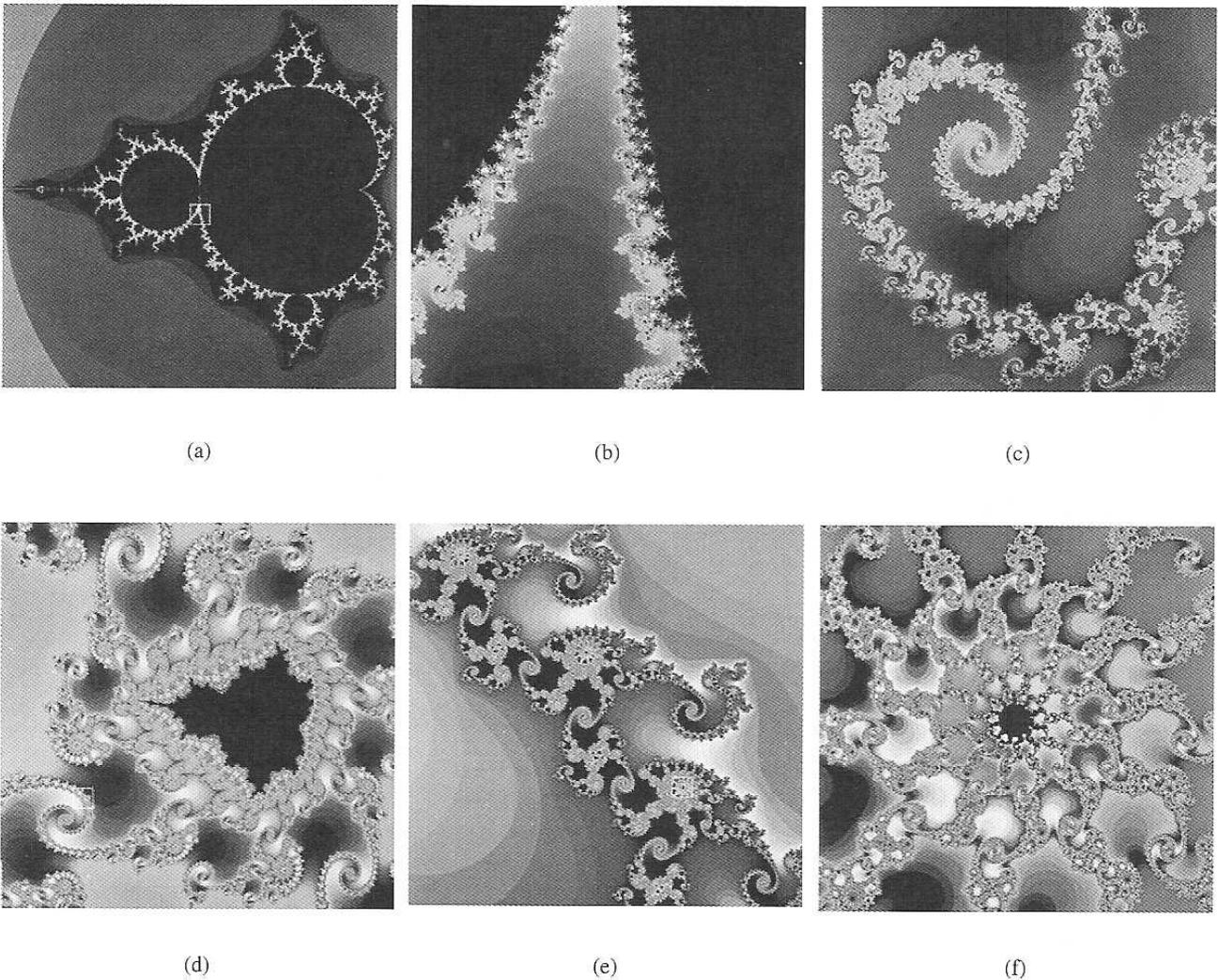


FIGURA 8.102. Ampliación de las fronteras fractales de la transformación 8.115. Comenzando por una visualización del conjunto de Mandelbrot, la región negra de (a) y sus áreas circundantes, ampliamos regiones seleccionadas de la frontera desde (b) hasta (f). El contorno blanco de la caja muestra el área rectangular seleccionada en cada ampliación sucesiva. En cada paso se eligen diferentes combinaciones de color para mejorar los patrones visualizados. (Cortesía de Brian Evans, Vanderbilt University.).

```
#include <GL/glut.h>

/* Establece el tamaño inicial de la ventana de visualización. */
GLsizei winWidth = 500, winHeight = 500;

/* Establece los límites del área rectangular del plano complejo. */
GLfloat xComplexMin = -2.00, xComplexMax = 0.50;
GLfloat yComplexMin = -1.25, yComplexMax = 1.25;

GLfloat complexWidth = xComplexMax - xComplexMin;
GLfloat complexHeight = yComplexMax - yComplexMin;

class complexNum {
public:
    GLfloat x, y;
};

struct color { GLfloat r, g, b; };

void init (void)
{
    /* Establece el color de la ventana de visualización en blanco. */
    glClearColor (1.0, 1.0, 1.0, 0.0);
}

void plotPoint (complexNum z)
{
    glBegin (GL_POINTS);
        glVertex2f (z.x, z.y);
    glEnd ( );
}

/* Calcula el cuadrado de un número complejo. */
complexNum complexSquare (complexNum z)
{
    complexNum zSquare;

    zSquare.x = z.x * z.x - z.y * z.y;
    zSquare.y = 2 * z.x * z.y;
    return zSquare;
}

GLint mandelSqTransf (complexNum z0, GLint maxIter)
{
    complexNum z = z0;
    GLint count = 0;

    /* Sale cuando  $z * z > 4$  */
    while ((z.x * z.x + z.y * z.y <= 4.0) && (count < maxIter)) {
```

```

    z = complexSquare (z);
    z.x += z0.x;
    z.y += z0.y;
    count++;
}
return count;
}

void mandelbrot (GLint nx, GLint ny, GLint maxIter)
{
    complexNum z, zIncr;
    color ptColor;

    GLint iterCount;

    zIncr.x = complexWidth / GLfloat (nx);
    zIncr.y = complexHeight / GLfloat (ny);

    for (z.x = xComplexMin; z.x < xComplexMax; z.x += zIncr.x)
        for (z.y = yComplexMin; z.y < yComplexMax; z.y += zIncr.y) {
            iterCount = mandelSqTransf (z, maxIter);
            if (iterCount >= maxIter)
                /* Establece el color de los puntos en negro. */
                ptColor.r = ptColor.g = ptColor.b = 0.0;
            else if (iterCount > (maxIter / 8)) {
                /* Establece el color de los puntos en naranja. */
                ptColor.r = 1.0;
                ptColor.g = 0.5;
                ptColor.b = 0.0;
            }
            else if (iterCount > (maxIter / 10)) {
                /* Establece el color de los puntos en rojo. */
                ptColor.r = 1.0;
                ptColor.g = ptColor.b = 0.0;
            }
            else if (iterCount > (maxIter / 20)) {
                /* Establece el color de los puntos en azul
                oscuro. */
                ptColor.b = 0.5;
                ptColor.r = ptColor.g = 0.0;
            }
            else if (iterCount > (maxIter / 40)) {
                /* Establece el color de los puntos en
                amarillo. */
                ptColor.r = ptColor.g = 1.0;
                ptColor.b = 0.0;
            }

            else if (iterCount > (maxIter / 100)) {

```

```

        /* Establece el color de los puntos en
        * verde oscuro. */
        ptColor.r = ptColor.b = 0.0;
        ptColor.g = 0.3;
    }

    else {
        /* Establece el color de los
        * puntos en cian. */
        ptColor.r = 0.0;
        ptColor.g = ptColor.b = 1.0;
    }

    /* Dibuja el punto coloreado. */
    glColor3f (ptColor.r, ptColor.g, ptColor.b);
    plotPoint (z);
}
}

void displayFcn (void)
{
    /* Establece el número de subdivisiones en los ejes x e y y las iteraciones
    máximas. */
    GLint nx = 1000, ny = 1000, maxIter = 1000;

    glClear (GL_COLOR_BUFFER_BIT);    // Borra la pantalla de visualización.

    mandelbrot (nx, ny, maxIter);
    glFlush ( );
}

void winReshapeFcn (GLint newWidth, GLint newHeight)
{
    /* Mantiene la relación de aspecto en 1.0, asumiendo que
    * complexWidth = complexHeight.
    */
    glViewport (0, 0, newHeight, newHeight);

    glMatrixMode (GL_PROJECTION);
    glLoadIdentity ( );

    gluOrtho2D (xComplexMin, xComplexMax, yComplexMin, yComplexMax);

    glClear (GL_COLOR_BUFFER_BIT);
}

void main (int argc, char** argv)
{
    glutInit (&argc, argv);

```

```

glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
glutInitWindowPosition (50, 50);
glutInitWindowSize (winWidth, winHeight);
glutCreateWindow ("Mandelbrot Set");

init ( );
glutDisplayFunc (displayFcn);
glutReshapeFunc (winReshapeFcn);

glutMainLoop ( );
}

```

Las transformaciones con funciones complejas, tales como la Ecuación 8.111, se pueden ampliar para producir superficies fractales y sólidos fractales. Los métodos para generar estos objetos utilizan representaciones con *cuaternios* (Apéndice A) para transformar puntos del espacio tridimensional y de cuatro dimensiones. Un cuaternio posee cuatro componentes, un número real y tres números imaginarios. Podemos representar un cuaternio de la siguiente forma, como ampliación del concepto de número del plano complejo,

$$q = s + ia + jb + kc \quad (8.116)$$

donde $i^2 = j^2 = k^2 = -1$. El número real s se denomina también *parte escalar* del cuaternio, y los números imaginarios se llaman *parte vectorial* del cuaternio $\mathbf{v} = (a, b, c)$.

Utilizando las reglas de la multiplicación y suma de cuaternios estudiadas en el Apéndice A, podemos aplicar las funciones autocuadráticas y otros métodos de iteración para generar superficies de objetos fractales. Un procedimiento básico consiste en comprobar puntos del plano complejo hasta que podamos identificar la frontera entre los puntos divergentes y los no divergentes. Por ejemplo, si localizamos en primer lugar un punto (interior) no divergente, entonces comprobamos los puntos vecinos respecto de dicho punto hasta que se identifique un punto (exterior) divergente. El punto interior anterior se guarda como un punto de la superficie frontera. A continuación se comprueban los vecinos de este punto de la superficie para determinar si están dentro (convergen) o si están fuera (divergen). Cualquier punto interior conectado con un punto exterior es un punto de la superficie. De este modo, el procedimiento se autodirige a lo largo de la frontera fractal sin desviar su rumbo lejos de la superficie. Cuando se generan fractales de cuatro dimensiones, los cortes tridimensionales se proyectan sobre la superficie bidimensional del monitor de vídeo.

Los procedimientos para generar fractales autocuadráticos en el espacio de cuatro dimensiones requieren un tiempo considerable de cálculo, para evaluar la función de iteración y comprobar la convergencia o divergencia de los puntos. Cada punto sobre una superficie se puede representar como un cubo pequeño, que proporciona los límites interno y externo de la superficie. La salida de tales programas para proyecciones tridimensionales contienen más de un millón de vértices en los cubos de la superficie. Visualizamos el objeto fractal mediante la aplicación de modelos de iluminación para determinar el color de cada cubo de la superficie. También se aplican métodos de detección de la superficie visible de modo que sólo se muestren las superficies visibles del objeto. Las Figuras 8.103 y 8.104 muestran ejemplos de fractales autocuadráticos de cuatro dimensiones mediante proyecciones en tres dimensiones.

Fractales autoinversos

Se pueden utilizar varias transformaciones geométricas de inversión para crear formas fractales. De nuevo, comenzamos por un conjunto inicial de puntos, y aplicamos repetidamente operaciones no lineales de inversión para transformar los puntos iniciales en un fractal.

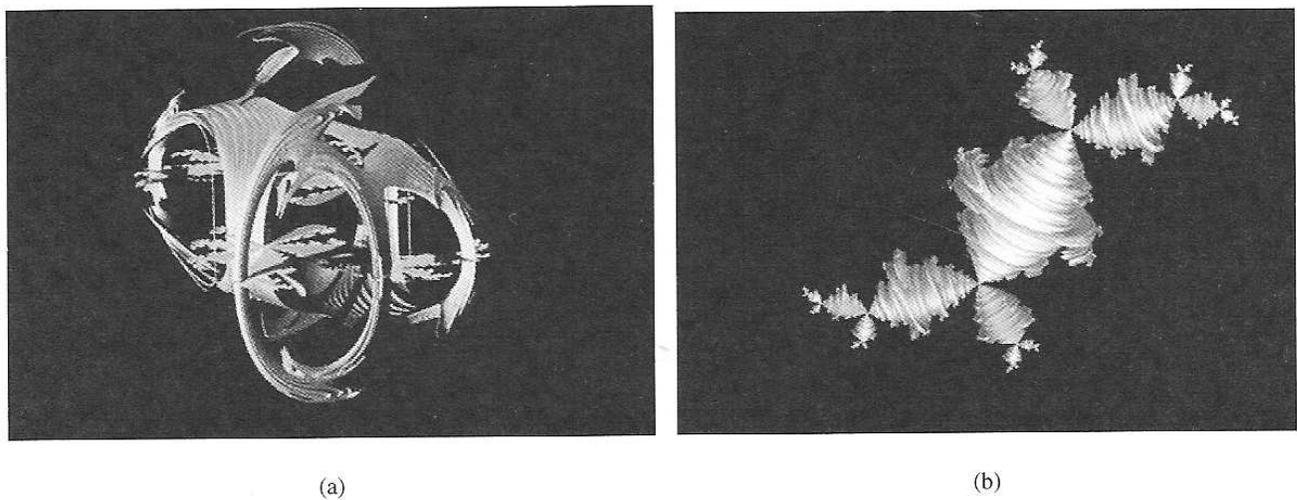


Figura 8.103. Proyecciones tridimensionales de fractales de cuatro dimensiones generados mediante la función autocuadrática representada con cuaternios $f(q) = \lambda q(1 - q)$, que utilizan (a) $\lambda = 1.475 + 0.9061i$ y (b) $\lambda = -0.57 + i$. (Cortesía de Alan Norton, IBM Research.)

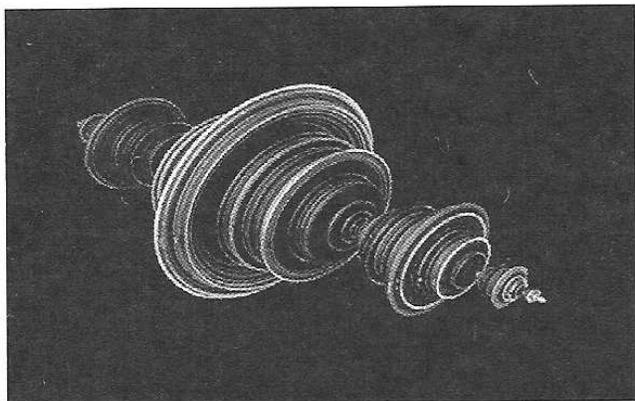


FIGURA 8.104. Una proyección sobre una superficie tridimensional de un objeto de cuatro dimensiones generado mediante una función autocuadrática representada con cuaternios $f(q) = q^2 - 1$. (Cortesía de Alan Norton, IBM Research.)

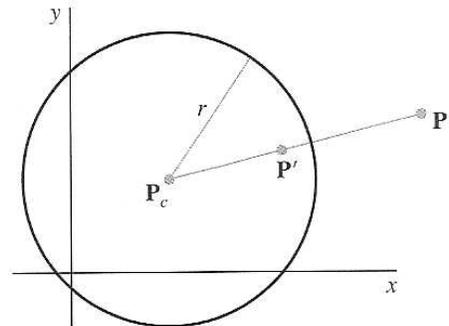


FIGURA 8.105. Inversión del punto P al punto P' situado dentro de un círculo de radio r .

Como ejemplo, consideremos una transformación bidimensional de inversión con respecto a un círculo de radio r y su centro $P_c = (x_c, y_c)$. Un punto P situado fuera del círculo se invierte en un punto P' situado dentro del círculo (Figura 8.105) mediante la transformación:

$$(\overline{P_c P})(\overline{P_c P'}) = r^2 \tag{8.117}$$

donde ambos puntos P y P' se encuentran situados en una línea recta que pasa por el centro del círculo P_c . Podemos usar también la Ecuación 8.117 para transformar los puntos que se encuentran dentro del círculo. Algunos puntos situados dentro se transforman en puntos situados fuera, mientras que otros puntos situados dentro se transforman en puntos situados dentro.

Si las coordenadas de los dos puntos se representan como $P = (x, y)$ y $P' = (x', y')$, podemos escribir la Ecuación 8.117 del siguiente modo:

$$[(x - x_c)^2 + (y - y_c)^2]^{1/2} [(x' - x_c)^2 + (y' - y_c)^2]^{1/2} = r^2$$

También, ya que los dos puntos están situados en una línea que pasa a través del centro del círculo, tenemos que $(y - y_c)/(x - x_c) = (y' - y_c)/(x' - x_c)$. Por tanto, los valores de las coordenadas transformadas del punto P' son:

$$x' = x_c + \frac{r^2(x - x_c)}{(x - x_c)^2 + (y - y_c)^2}, \quad y' = y_c + \frac{r^2(y - y_c)}{(x - x_c)^2 + (y - y_c)^2} \quad (8.118)$$

Por tanto, los puntos situados fuera del círculo se mapean a puntos situados dentro de la circunferencia del círculo; los puntos distantes ($\pm\infty$) se transforman en el centro del círculo. A la inversa, los puntos cercanos al centro del círculo se mapean a puntos distantes situados fuera del círculo. A medida que nos alejamos del centro del círculo, los puntos se mapean a puntos más cercanos a la circunferencia del círculo, situados fuera del mismo. Y los puntos situados dentro cerca de la circunferencia se transforman en puntos situados dentro cerca del centro del círculo. Por ejemplo, los valores de la coordenada x de fuera, dentro del rango que varía desde r a $+\infty$, se mapean a valores x' dentro del rango que varía desde $r/2$ a 0 , para un círculo centrado en el origen, y los valores de x de dentro desde $r/2$ hasta r se transforman en valores dentro del rango que varía desde r hasta $r/2$. Se obtienen resultados similares para los valores negativos de x .

Podemos aplicar esta transformación a varios objetos, tales como líneas rectas, círculos o elipses. Una línea recta que pasa por el centro del círculo es invariante frente a esta transformación de inversión; se mapea en sí misma. Pero una línea recta que no pasa por el centro del círculo se invierte en un círculo cuya circunferencia contiene el centro P_c . Y cualquier círculo que pase por el centro del círculo de referencia se invierte en una línea recta que no pasa por el centro del círculo. Si el círculo no pasa por el centro del círculo de referencia, se invierte en otro círculo, como en la Figura 8.106. Otro invariante frente a la inversión es la transformación de un círculo que es ortogonal al círculo de referencia. Es decir, las tangentes de los dos círculos son perpendiculares en los puntos de intersección.

Podemos crear varias formas fractales mediante esta transformación de inversión comenzando por un conjunto de círculos y aplicando la transformación utilizando diferentes círculos de referencia. De forma similar, podemos aplicar la inversión con círculos a un conjunto de líneas rectas. Se pueden desarrollar métodos de inversión similares para otras formas bidimensionales. Y, podemos generalizar el procedimiento a esferas, planos u otros objetos tridimensionales.

8.24 GRAMÁTICAS DE FORMAS Y OTROS MÉTODOS PROCEDIMENTALES

Se puede utilizar un gran número de otros métodos procedimentales para diseñar formas de objetos o niveles de detalle de la superficie. Las **gramáticas de formas** son conjuntos de reglas de producción que se pueden aplicar a un objeto inicial, para añadir capas de detalle que son armoniosas con la forma original. Las transformaciones se pueden aplicar para alterar la geometría (forma) del objeto, o las reglas de transformación se pueden aplicar para añadir detalles del color o la textura de la superficie.

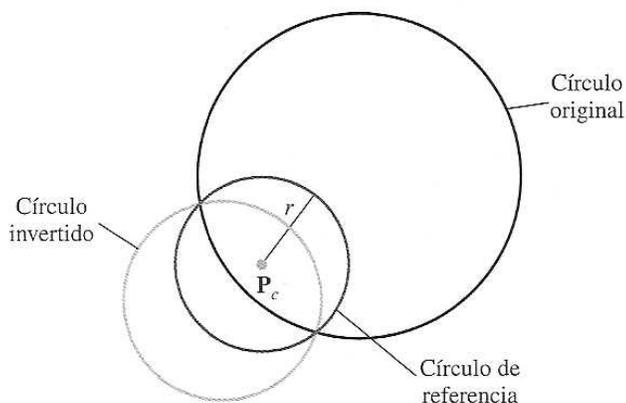


FIGURA 8.106. Inversión de un círculo que no pasa por el origen del círculo de referencia.

