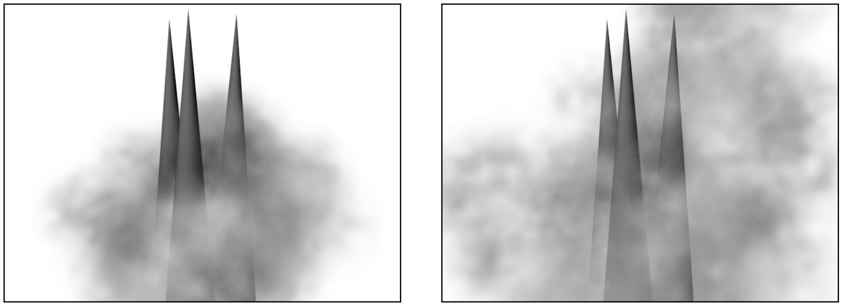

Figure 6-10. Particle systems can create effective animated smoke effects.

Color plate 8 uses a volumetric fog light to project both fog and image through the upper-right corner of the composition.

Another situation in which a localized atmospheric effect may be necessary is when **smoke** is present. Because of the density of smoke, even a faint light passing through it will make the smoke visible. A common technique for rendering smoke is particle systems, which we discuss in much more detail in section 7.8. As with volumetric fog lights, the goal is to account for the tiny particles that float in the air and deflect the light. Like all the techniques discussed in this section, any parameters you set can be animated. In Figure 6-10, we see two moments in the life of a particle-system smoke effect. In (b) the smoke has expanded and risen relative to its original configuration in (a).

6.3 Fractals

In the 1970s, a French mathematician named Benoît Mandelbrot began refining an unusual new branch of mathematics, which has turned out to be extremely rich in possibilities for computer graphics. He termed this new mathematics **fractal** mathematics, because it involves the possibility of writing mathematical equations for geometries of fractional dimensionality. Normally, you think of dimensionality in terms of whole numbers: a point in space is 1-dimensional, a drawing on a piece of paper is 2-dimensional, a wooden box is 3-dimensional, and so on. In Mandelbrot's system, it is possible to think of something that is 1.76-dimensional, for instance, or 2.24-dimensional.

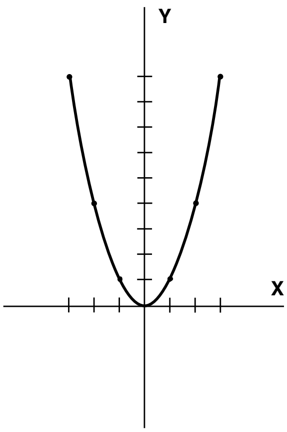

Figure 6-11. Any mathematical equation can be visually plotted. Here a simple equation is plotted as a parabola.

It is common for mathematicians to make visual representations, or graphs, of their mathematical formulas. To take a simple example, the visual representation of the equation $y = x^2$ is a parabola (Figure 6-11). This curve can be plotted by listing several *x* values, calculating the *y* value for each *x* value, and then drawing a mark at each (*x,y*) combination.

The images that result when fractal equations are plotted are much more complex than this simple example, and mathematicians working with fractal

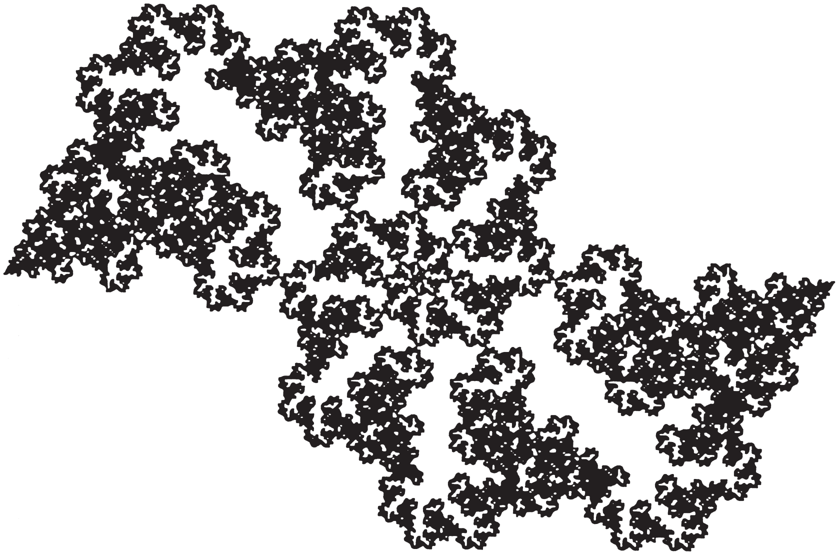

Figure 6-12. A pattern generated by plotting an equation of fractal mathematics. (Bathsheba L. Grossman, 1985)

mathematics soon discovered that visual representations of their fractal equations produced some very intricate, unexpected, and beautiful results. Most interestingly, they found that graphs of fractal equations produce imagery that captures some of the characteristics of natural phenomena. That is, fractal imagery often has an irregularity and an unpredictability, an intricacy of detail, and a similarity of detail to overall form reminiscent of forms found in nature (Figure 6-12).

As fractal mathematics developed, computer graphics researchers began to develop techniques that took advantage of these characteristics by producing both two-dimensional pictures and three-dimensional models. Today, these fractal techniques have become an integral part of all 3D CG systems, and are used for both modeling and rendering.

One of the most important characteristics of fractals is that at any level of detail you can see similarities in the forms. The forms at the most minute level bear a resemblance to the forms at grosser levels, and these in turn look similar to forms at even higher levels. This characteristic, called **self-similarity**, belongs to many phenomena in nature as well. The large-scale crags and peaks of a mountain are similar in shape to the smaller abutments that make up those crags. Individual rocks on the mountainside have a similar "cragginess" to them, but on a much smaller scale. The pebbles that break off from such rocks have similar forms, and if you examine one of these pebbles with a magnifying glass, you would find miniature "crags" similar to the cragginess of the mountain itself.

To see how self-similarity works in a fractal image, take a simple and common example, called the **Koch curve**. You start generating this fractal curve with a simple straight line (Figure 6-13a). Next you break this straight line into a pattern, which is called the **generator** (Figure 6-13b). Breaking each of the four straight lines that make up the generator into yet smaller versions of the generator pattern yields a new pattern consisting of four occurrences of the generator (Figure 6-13c). Subdividing each straight line of that pattern into the generator pattern yields the next **generation** of the curve (Figure 6-13d). You can repeat this process of subdivision indefinitely, creating ever more detailed and refined repetitions of the generator pattern.

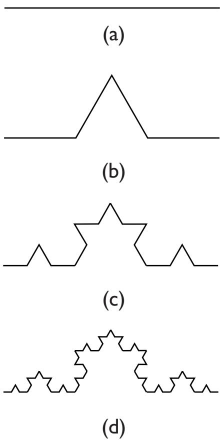

Figure 6-13. A Koch curve exemplifies the two fractal principles of self-similarity and recursion.

Notice that each new generation of the Koch curve is defined in terms of a previous Koch curve. For example, Koch-curve generation number 22 consists of Koch-curve generation number 21 with a generator pattern substituted for each straight line of number 21. In other words, in order to create a Koch curve, you first must create a Koch curve: the definition is **self-referential.**

This process of defining something in terms of itself is called **recursion** and the process is said to be **recursive**. Because it is self-referential, a recursive process in principle can go on forever. A common example of recursion in the physical world is what happens when two mirrors are placed face to face. If you look in the first mirror, you see yourself reflected in the second, which in turns reflects an image of yourself looking in the first mirror and being reflected in the second, which in turn reflects an image of yourself in the second mirror looking at yourself in the first mirror and being reflected in the second, which is being reflected in the first—and so on.

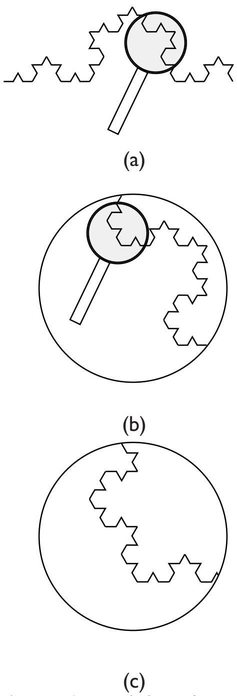

Figure 6-14. A fractal image has, as part of its definition, an infinite amount of detail. No matter how much you magnify an image, you will always see additional detail.

In a fractal curve there can be an unlimited number of generations—that is, it contains an unlimited amount of detail, even if you cannot see it. Beyond a few additional generations, for example, you could not see any more detail in Figure 6-13d. The image on the page is simply too small for your eye to pick up such tiny detail. For this practical reason, therefore, you probably would not attempt to draw too many additional generations. It is important to understand, however, that by definition these additional generations of the curve are there, even if you choose not to draw them.

If you imagine magnifying a section of the curve, you can understand more clearly that no matter how much you magnify a section of a fractal curve, there will always be, if you choose to draw it, more detail to see. Imagine that you place a magnifying-glass over a section of a Koch curve (Figure 6-14a). Notice that even at a magnified level the same amount of detail is available (Figure 6-14b). If you zoom in yet again, as indicated by the magnifying-glass icon, the resultant image again has a full level of detail (Figure 6-14c). The definition of the original curve includes all of these levels of detail—that is, all of these generations.

Some software packages offer fractal procedures that allow you to directly model a surface. If you now imagine trying to model the mountain described at the beginning of this section by using the nonfractal modeling techniques discussed earlier in Chapter 2, you can see that none of those techniques allows you very successfully to simulate the infinitely craggy quality of the mountain. First, complexity and irregularity make the forms involved tremendously difficult to model by positioning vertices and control points. Second, trying to model these forms in enough detail to make them look convincing both from the altitude of an airplane and from the height of a person walking on the surface of the mountain is impractical.

Moreover, if, through some tremendous exercise of saintly patience, you manage to create such an exceedingly detailed model, you will produce a wasteful rendering situation. Since all of the details of this model have to be dealt with, the rendering process will spend a lot of time rendering detail that is not visible. From the height of the airplane, the renderer will render even the tiniest details, even those details too small to be seen in the final picture.

Fractal models solve both the theoretical problem of whether it is possible to create such a complex model, and the practical problem of how to render it once you have created it. On the theoretical side, fractals are by definition very irregularly shaped and contain an unlimited amount of detail, so a fractal technique can, in fact, solve the problem of how to model a mountain. On the practical side, fractal techniques offer control over how many generations of a pattern are going to be created. Depending on the level of magnification needed, the software creates more or fewer generations, in the same way that you increase or decrease the number of generations—that is, the level of detail—in the Koch curve. The most common technique for applying fractal mathematics to three-dimensional modeling is very similar to the approach illustrated by the Koch curve. This process is called **recursive subdivision.**

Imagine that you have a simple rectangular patch (Figure 6-15a). If you subdivide this patch into four smaller subpatches, and then displace the corners of each patch some irregular amount, you produce the model in Figure 6-15b. If you then subdivide each of these subpatches into four smaller patches, with the corners irregularly displaced, you produce a model with more displacement (Figure 6-15c). By repeating this process many times, you can model a very irregular, terrainlike surface (Figure 6-15d). Further, depending on how much detail you need to see on the surface from a given point of view, using this modeling procedure the modeling software can produce a higher or a lower generation of the fractal geometry. A lower generation is suitable for viewing the terrain from a distance. A higher generation is suitable for a close-up view.

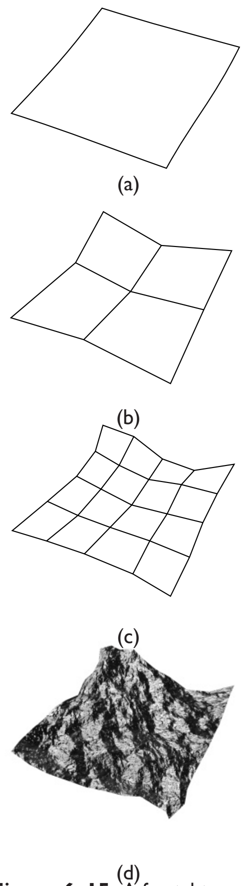

Figure 6-15. A fractal terrain model can be developed by recursively subdividing and displacing a patch.

More commonly, fractal mathematics is used in off-the-shelf software to generate **texture images** that are, in turn, used for a wide variety of rendering applications. Most software packages offer a range of fractal texture

Figure 6-16. Fractal textures are especially useful, both in modeling and in rendering. (*Off the Map*, © 1991 Sylvain Moreau.)

procedures, each adjusted to accomplish a specific purpose. One might make a cloudlike pattern, another a stony-surface pattern, and so on.

A fractal texture might be used for color texture mapping to create an irregular pattern of color. It can also be used for bump texture mapping to create an irregular pattern of bumps on a surface. In color plate 10, the texture of the sand was created with a fractal bump map. Fractal bump mapping can also be useful in conjunction with a fractally-generated displacement map (section 5.7). With this approach, the displacement map creates the large-scale geometry changes, while the bump map (section 3.8) creates the appearance of small-scale bumps.

Applied as a transparency map, a fractal texture can introduce a random irregularity into an atmospheric fog (section 6.2), or can cause the rendering of particles to look smoky (section 7.8). When attempting these effects, you most commonly use fractal procedures to generate solid textures (section 3.9), thereby creating a three-dimensional volume of transparency that varies as you move through the space of the fog or smoke.

The image in Figure 6-16 uses fractal texture mapping in several ways. The material partially covering the woman's face is actually a flat surface with two fractal textures applied—one as a bump map and the other as a transparency map.

6.4 Lighting Subtleties

We have discussed lighting in several other sections of this book. Section 3.3 introduced the basic concepts and techniques of lighting, and in section 6.2 we discussed the use of volumetric fog lights to render smoke or haze within