random numbers, stored as RGBA components, and blurred slightly to hide artifacts caused by trilinear interpolation. Multiple copies of the noise texture are used for each slice at different scales during rendering. Per-pixel perturbation is computed as a weighted sum of the individual noise components. To animate the perturbation, a varying offset vector can be added to the noise texture coordinates in the fragment shader.

The first technique uses the four per-pixel noise components to modify optical properties of the volume after the transfer function has been evaluated. This results in materials that appear to have irregularities. By selecting which optical properties to modify, different effects are achieved.

The second method uses noise to modify the location of the data access in the volume (Kniss et al. 2003). In this case, three components of the noise texture form a vector, which is added to the texture coordinates for the data texture lookup. Figure 39-14 illustrates how volume perturbation is used to add intricate detail to coarse volumetric models.

39.6 Performance Considerations

Texture-based volume rendering can easily push the performance limits of modern GPUs. This section covers a few considerations specific to volume rendering on GPUs.

39.6.1 Rasterization Bottlenecks

Unlike most graphics applications, texture-based volume renderers use a small number of relatively large geometric primitives. The rasterizer produces many fragments per primitive, which can easily become the bottleneck in the pipeline. In addition, unlike opaque objects, the transparent proxy geometry used in volume rendering cannot leverage the early depth-culling capabilities of modern GPUs. The rasterization bottleneck is exacerbated by the large number of slices needed to render high-quality images. In addition, the frame buffer contents in the compositing stage have to be read back every time a fragment is processed by the fragment shader or the compositing hardware.

For these reasons, it is important to draw proxy geometry that generates only the required fragments. Simply drawing large quads that cover the volume leads to a very slow implementation. The rasterization pressure is reduced by making the viewport smaller, decreasing the sample rate, using preintegrated classification (Engel et al. 2001), and by