$$\hat{C}_i = \left(1 - \hat{A}_{i-1}\right) C_i + \hat{C}_{i-1},$$
$$\hat{A}_i = \left(1 - \hat{A}_{i-1}\right) A_i + \hat{A}_{i-1},$$

where

$$\hat{C}_i$$

and

$$\hat{A}_i$$

are the accumulated color and opacity from the front of the volume.

The compositing equations (Equations 5 and 6) are easily implemented with hardware alpha blending. For the Over operator, the source blending factor is set to 1 and the destination blending factor is set to (1 – source alpha). For the Under operator, the source blending factor is set to (1 – destination alpha) and the destination factor is set to 1. Alternatively, if the hardware allows for reading and writing the same buffer, compositing can be performed in the fragment shading stage by projecting the proxy polygon vertices onto the viewport rectangle (see Section 39.5).

To blend opaque geometry into the volume, the geometry needs to be drawn before the volume, because the depth test will cull proxy fragments that are inside objects. The Under operator requires drawing the geometry and the volume into separate color buffers that are composited at the end. In this case, the depth values from the geometry pass are used for culling fragments in the volume rendering pass.

39.5 Advanced Techniques

This section describes techniques for improving the quality of rendering and creating volumetric special effects.

39.5.1 Volumetric Lighting

The local illumination model presented in the previous section adds important visual cues to the rendering. Such a simple model is unrealistic, however, because it assumes that light arrives at a sample without interacting with the rest of the volume. Furthermore, this kind of lighting assumes a surface-based model, which is inappropriate for volumetric