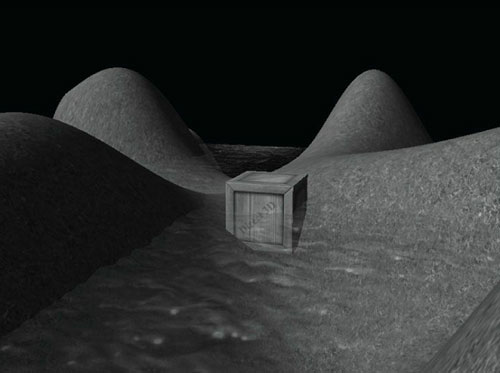
# 9.1 混合方程

考虑图9.1。我们从地形和板条箱开始渲染场景，先绘制地形，再绘制板条箱，使地形和板条箱的像素依次存入后台缓冲区。然后使用混合（blending），将水体绘制到后台缓冲区，使水体像素和后台缓冲区中的地形、板条箱像素融为一体。通过一方式，我们可以透过水体看到地形和板条箱。本章我们主要讲解混合技术，它可以将当前的光栅化像素（也称为源像素）与后台缓冲区中的像素（也称为目标像素）混合（融合）在一起。该技术通常用于渲染半透明物体，比如水和玻璃。

****

**图9.1 半透明的水面。**

**注意**：为了叙述方便，当我们提到后台缓冲时，指的是渲染目标。但是，我们也可以绘制到一个“离屏，off screene”渲染目标中。混合作用在这些渲染目标的方式是相同的，目标像素的值就是之前光栅化后的离屏渲染目标中的值。

**学习目标**

1．理解混合的工作原理以及如何在Direct3D中使用混合。

2．学习Direct3D支持的各种混合模式。

3．了解如何通过alpha分量来控制图元的透明度。

4．学习如何使用HLSL的**clip**函数阻止像素绘制到后台缓冲区。

设**C**src为当前正在进行光栅化处理的第*ij*个像素（源像素）的颜色，**C**dst为后台缓冲区中的第*ij*个像素（目标像素）的颜色。当不使用混合时，**C**src会覆盖**C**dst的值（假设该值已通过深度/模板测试）并成为后台缓冲区中的第*ij*个像素的新颜色。但是当使用混合时，**C**src和**C**dst会被组合为一个新颜色**C**并覆盖**C**dst的值（即，混合颜色**C**会成为后台缓冲区中的第*ij*个像素的新颜色）。Direct3D使用如下混合方程混合源像素和目标像素颜色：

**C** = **C**src ⊗ **F**src ⊞ **C**dst ⊗ **F**dst

**F**src（源混合系数）和**F**dst（目标混合系数）可以是9.3节描述的任何一个值，它们按照各种不同的方式调整源像素和目标像素的比例，实现各种不同的混合效果。运算符⊗是定义在5.3.1节中定义的颜色的分量乘法；运算符⊞可以是9.2节描述的任何一个二进制运算符。

上述混合方程只负责处理颜色的RGB分量。alpha分量要由另外一个方程来处理：

*A* = *A*src*F*src ⊞*A*dst*F*dst

该方程与前一个方程基本相同，只是使用的混合系数和二进制运算符有些区别。我们将RGB和alpha分开的原因非常简单，就是为了独立地、按照不同的方式处理它们。

**注意**：我们很少混合alpha分量，多数情况下，我们只是对RGB分量进行混合。尤其是在本书中，我们从不混合源alpha值和目标alpha值，虽然alpha值会参与我们的RGB混合操作。这主要是因为我们暂时用不到后台缓冲区中的alpha值。只有当某些算法需要使用目标alpha值时，后台缓冲区中的alpha值才有用。