

模拟海水

杰里·泰森多夫

1 引言和目标

这些笔记旨在为计算机图形程序员和艺术家介绍模拟，动画和渲染海水环境的方法CG水已经成为一个共同的工具，在视觉效果的工作，在各级计算机图形，从印刷媒体到故事片。几种商业产品可用于几乎任何计算机平台和工作环境。一些视觉效果公司继续扩展和改进这些工具，寻求生成更高质量的表面几何形状，复杂的交互和更引人注目的图像。为了让艺术家利用这些工具获得最大的收益，他或她熟悉概念，术语，一点海洋学和目前的状态是很重要的。

正如电影《水世界》和《泰坦尼克号》以及自1995年以来制作的其他几部电影中的开创性努力所证明的那样，CG水的图像可以以高度的真实感生成。然而，这种现实主义水平主要限于相对平静，良好的海洋条件。具有大量喷雾、破碎波、泡沫、飞溅和尾流的条件正在改善并接近相同的逼真外观。

目前在计算机图形学中流行三种通用方法来模拟流体运动，特别是水面。这三种方法都以某种方式与许多应用的核心基本Navier-Stokes方程相关。计算流体动力学（CFD）是近年来受到广泛关注虽然这种技术的许多版本已经存在了一段时间，但Stam[1]，Fedkiw和Foster[2]以及许多其他人最近的CFD方法的主要缺点是计算是在被称为欧拉框架的2D或3D网格中结构化的数据上执行的。这种网格化架构限制了可以计算的组合范围和流细节例如，目前，使用CFD波浪在接近浅海滩时破碎进行模拟是可行的然而，该模拟不能模拟海湾大小的海洋区域，同时也模拟分解到喷雾形成的细节，这仅仅是因为所需的网格点的数量非常大。

第二种方法显示出巨大的希望，被称为光滑粒子流体动力学（SPH）。这是一个完全不同的方法来解决Navier-Stokes方程。SPH设想流体的体积由小的重叠区域组成，每个区域的中心携带一定量的质量和动量。这些区域被允许在流体内四处移动，并且由于压力、应变、重力等而经受力。但现在每个区域的中心作为一个粒子，Navier-Stokes方程被转换成离散粒子的运动方程。这种方法被称为拉格朗日框架（与CFD中的欧拉框架相对）。流体体积未绑定到栅格几何体。SPH是一种非常有用的模拟方法，适用于存在显著飞溅或爆炸的情况，甚至已用于模拟固体中的开裂[3]。使用隐式方法来构造流体的表面，已经实现了标准的计算机图形应用，例如倒水[4]。

最后，第三种方法是这些笔记的重点，与CFD和SPH方法不同，这一方法专注于模拟水体表面运动的更窄目标。表面模拟通常由CFD和SPH方法生成，但表面是通过添加到这些流体模拟的算法生成的，例如通过跟踪称为水平集的隐式表面类型。在选择专注于表面结构和运动，我们消除了计算流体力学和SPH方法的计算和分辨率限制

耗氧物质当然，当我们消除这些计算时，我们已经失去了某些类型的真实运动，最明显的是表面的破裂与拓扑结构的强烈变化。代替该计算，我们用海洋现象学知识和计算灵活性的混合物代替，以实现其他人实际上无法实现的现实类型的表面从这个意义上说，这种现象学方法应被视为CFD和SPH方法的补充

一般来说，读者应该离开这个材料与

1. 了解海洋表面传播的重要物理概念，最重要的是色散的概念和色散关系的类型。
2. 了解一些生成/动画水面高度场的算法，适用于模拟像风暴潮一样大的波浪和像微小毛细管一样小的波浪；
3. 了解水面反射和折射的基本光学过程；
4. 介绍了海水的颜色过滤行为；
5. 介绍复杂的照明效果，即焦散和godrays，当阳光穿过粗糙的表面进入下面的水量时产生；以及
6. 一些经验法则，哪些选择使好看的图像和什么是权衡的质量与计算资源。提供了一些示例着色器，示例渲染演示了讨论的内容

在深入研究之前，我首先想更具体地说明我们在这些笔记中涵盖（或不涵盖）海洋环境的哪些方面。图1是由水、空气和云的模型产生的海洋景观的渲染云层发出的光从表面反射回来在最左边，也存在太阳闪光。水的通常带蓝色的颜色是由于蓝色天光的反射，以及从体积散射后从水中出来的光。虽然这些笔记没有处理云和空气的建模和渲染，但讨论了来自云和空气的天光如何从水面反射或折射。这些注释将告诉您如何使用图中所示的细节和质量为海浪创建高度场位移映射曲面。注释还讨论了水下环境的几个效应以及如何建模/渲染它们。主要的四种效果是阳光（也称为godray）、水下表面的焦散、光散射造成的模糊和滤色。

海洋环境还有许多其他复杂而有趣的方面这些包括破碎波、喷雾、泡沫、水中物体周围的尾迹、来自撞击表面的物体的飞溅物以及整个海洋-大气环境的全局照明。关于这些主题有大量我已经包括了一个简短的一节先进的修改基本波高算法，产生波涛汹涌的波浪。该修改可以可行地导致破碎波的表面部分的完整描述，并且也可能用于驱动喷雾和泡沫动力学。

当然，在计算机图形学和海洋学领域，有大量关于海洋表面模拟和动画的文献计算机图形学中最早描述水波的是Fournier和Reeves [12]，他们使用称为Gerstner波的水面模型模拟了波浪出现在其上的海岸线。在同一个问题上，Darwin Peachey[13]



图1：海洋环境的真实图像

提出了一个变化的这种方法使用的基础形状以外的正弦曲线。

在海洋学文献中，海洋光学在20世纪40年代成为一个热门的研究课题。平方根邓特利在1963年发表的论文[17]中包含了与计算机图形学相关的光学数据今天继续工作。光学海洋学领域已经发展成为一门成熟的定量科学，具有分支学科和许多不同的应用。Curtis Mobley[18]撰写了一篇关于科学现状的优秀评论。

在这些讲座中，我们采取的方法来创建表面波接近Masten, Watterberg和Mareda[11]所概述的方法，尽管该技术在光学海洋学界发表论文之前已经使用多年。这种方法从快速傅里叶变换（FFT）处方合成一片海浪，具有用户可控的大小和分辨率，并且可以在更大的域上无缝地平铺补丁包含许多倍频程的正弦波，所有加起来，在每个然而，正弦振幅和相位的混合物来自于海洋的基于经验的统计模型。使这些正弦曲线看起来像波而不仅仅是一堆正弦波的原因是使用的大量正弦曲线、正弦曲线的相对振幅以及使用色散关系的动画。我们研究了正弦曲线的数量和分辨率对渲染图像质量的

在下一节中，我们开始讨论海洋环境，广泛介绍全球照明问题。该环境的辐射度方程看起来很像任何其他辐射度问题的方程，尽管一些环境组件的体积特性使一般实现相当复杂。然而，我们简化的问题，忽略了一些相互作用，并取代其他与遥感数据生成的模型。

第4节介绍了创建海洋表面实现的实用方法我们提出了两种方法，一种是基于一个简单的模型的水结构和运动，和一个基础上总结了大量的正弦波的振幅是相互关联的基础上的实验证据。这种第二种方法使用快速傅立叶变换（fft）技术进行求和，并且已经有效地用于商业广告、电视和电影的项目中。

在讨论了水表面的结构和动画之后，我们着重讨论了水的光学性质与

图形问题。首先，我们讨论空气-水界面的相互作用：反射和折射。这给我们留下了一个简单但有效的Renderman风格的着色器，例如，适合在BMRT中接下来，探索水下环境的光学特性

最后，请记住，这些笔记是一个活的文件。一些 的 的讨论 的 各种 top-ics仍然非常有限和不完整。如果您发现问题或有其他问题，请随时与我联系 jerry@finelightvisualtechnology.com。 的本课程文档的最新版本位于<http://www.finelightvisualtechnology.com>

2 海洋环境

就我们的目的而言，海洋环境仅由四个组成部分组成：水面、空气、太阳和水面以下的水量。在本节中，我们以数学方式和示意性方式跟踪从光源到相机的通过环境的光的流动。通常，这里的光能传递方程与任何其他光能传递问题一样是耦合的。然而，在合理的程度上，耦合可以被截断，并且简化的光能传递问题具有相对快速的解决方案。

由相机看到的光取决于来自源（即，光源）的光能流。太阳和天空）到表面并进入相机。除了来自表面的直射太阳光和天光的镜面反射之外，入射光的一部分透射通过表面。最终，一部分透射光被水体积散射，通过界面返回并进入空气中。在表面处反射或折射的一些光可以第二次撞击表面，从而产生更多的反射和折射事件。在某些观看条件下，多次反射和折射可能对图像产生明显的影响。然而，就我们而言，我们将一次忽略来自表面的多个反射或折射。这不仅使算法和计算更容易和更快，而且在大多数观看条件下相当准确，并产生视觉上逼真的图像。

在表面上方的环境中的任何点处，包括在相机处，来自任何方向的总光强度（辐射）具有三个贡献：

$$L_{\text{above}} = rL_s + rL_a + t_u U_u, \quad (1)$$

术语定义如下

r 是从海洋表面上的点到相机的反射的菲涅耳反射率

t_u 是来自海洋体积的光 L_u 的透射系数，在表面处折射到相机中。

L_s 是直接来自太阳的光的量，通过大气，到达海洋表面上的点，在那里它被表面反射到相机。

L_a 是（漫射）大气天光

L_u 是刚好在表面之下的光，其通过表面透射到空气中。

方程1有意地以一种速记的方式书写，隐藏了对空间位置和光传播方向的依赖性。

虽然等式1看起来具有相对简单的结构，但是项 L_s 、 L_a 和 L_u 原则上可以彼此具有复杂的依赖性，以及对反射率和透射率具有复杂的依赖性。

有大量的研究文献详细调查了这些依赖性[19],但我们在这一点上不会追求这些定量方法。但是我们可以继续使用相同的简写符号的同时进一步阐述耦合。来自太阳的直射光 L_S 是

$$L_S = L_{TOA} \exp\{-\tau\}, \quad (2)$$

其中 L_{TOA} 是在大气顶部的直射阳光的强度,并且 τ 是针对阳光的方向和地球上的点的大气的“光学厚度”。漫射大气天光 L_A 和上升光 L_U 两者都可以被写为两项的和:

$$\begin{aligned} L_A &= L^0_-(L_S) + L^1_-(L_U) \\ L_U &= L^0_+(L_S) + L^1_+(L_A) \end{aligned} \quad (3)$$

$$\quad \quad \quad (4)$$

$$\quad \quad \quad U \quad U$$

这些方程揭示了问题的潜在复杂性。虽然 L_A 和 L_U 都依赖于直射阳光,但它们也相互依赖。例如,穿透到海洋中的光的总量来自直射阳光和来自大气阳光。一些进入海洋的光被海洋中的微粒和分子散射,回到大气中。一部分上升的光线又被散射到大气层中,成为天光的一部分,照射在地表上,如此循环往复。这是光能传递中的一个经典问题。与其他无线电问题相反,它对于这种情况并不是特别特殊的,除了上涌光因为它来自体积多次波

散射

出于这些注释的目的,我们解决光能传递问题的方法很简单:由于上升流的贡献代表了对阳光的“第三次”依赖性,因此认为天光仅依赖于来自太阳的光;并且用基于对海洋的科学观察的经验公式来完全替换用于 $L-U$ 的方程,该经验公式仅取决于直接的阳光和指示水的类型和透明度的一些其它参数。

在水面下,辐射度方程具有示意形式

$$L_{\text{低于}} = tL_D + tL_I + L_{SS} + L_M, \quad (5)$$

定义见

t 是在表面上的每个点和角度处透射通过水面的菲涅耳透射率。

来自太阳的“直接”光穿透到水中。

L_I 来自

来自太阳和大气的单次散射光,在到达任何点之前在水体中散射一次。

多重散射光。这是在体积中经历更多散射事件的单次散射光。

就像上面的水的情况一样,这些术语都是以相对复杂的方式相互关联的例如,单个散射光取决于直接光和间接光:

$$L_{SS} = P(tL_I) + P(tL_D) \quad (6)$$

其中量 P 是其自变量的线性函数算子,包含关于单个散射事件和散射光在其从散射光穿过时的衰减的信息。

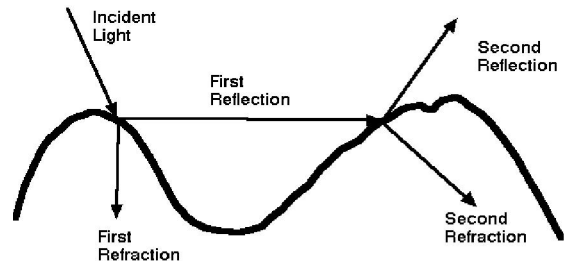


图2: 多次反射和透射

通过空气-水界面。

指向摄像机。类似地,多次散射光取决于单次散射:

$$L_M = G(tL_I) + G(tL_D). \quad (7)$$

功能示意量 P 和 G 是相关的,因为多重散射只是一系列单散射。从形式上讲,这两者具有操作符依赖性,其形式

$$\begin{aligned} GP &= 1 + P + \frac{1}{2}PP + \frac{1}{6}PPP + \dots \\ &\sim \frac{1}{1 - P} \exp(P). \end{aligned} \quad (8)$$

在这一点上,示意性表示可能已经失去了它的有用性,因为卷积类算子的复杂(并且在这里没有定义)含义,并且因为 G 关于 P 的表达式已经创建了关于指数 P 的甚至更示意性的视图。所以现在我们将离开示意性表示,并在剩下的时间里继续进行更具体的数量。

这里提出的正式的图解讨论确实有一个数学上和物理上精确的对应物。辐射传输的研究领域已经被大量的研究人员应用到水光学领域一段时间了。引用的参考文献是进一步了解信息的极好读物。

如前所述,在某些情况下,还有一个对海洋渲染很重要的光能传递场景,但我们不会考虑它。这种情况如图2所示。沿着追踪光行进方向的箭头的轨迹,我们看到有时到达表面(从上方或下方)的光可以不止一次地反射和/或透射通过表面。大量产生这种行为的条件是:波的高度必须相当高,并且观察波的方向或光源的方向必须几乎掠过表面。波浪越高,光源或相机需要的掠射越少。这种现象已在实验和计算机模拟中进行了研究。这是相当好理解的,我们将从这一点上忽略它。

3 表面运动的数学、物理和实验

在这一节中,我们来看看我们试图解决的数学问题。我们通过应用一系列近似法大大简化了数学。我们怎么知道这些近似值是好的?经过几十年的海洋学研究,海洋表面的运动已经得到了表征

通过测量、模拟、数学分析和实验。我们在本节中应用的近似值并不完美，在现实世界中许多情况下它们会崩溃。但它们在海上的大多数条件下工作得非常好。为了让你对它们的工作原理有一些了解，我们在第4节中展示了一些已经完成的实验工作，这些工作清楚地表明了这些近似在现实世界中的工作。

3.1 伯努利

海洋表面运动的数学公式的起点[14]的第2章提供了从Navier-Stokes的Bernoulli方程的彻底推导我们在这里提供了简短的版本，并请您参考金斯曼

不可压缩Navier-Stokes方程的速度
 $\mathbf{u}(\mathbf{x}, t)$ 在任意时间 t 任意位置 \mathbf{x} 的流体A的方程为

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \mathbf{F} \quad (9)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (10)$$

在这些等式中， $p(\mathbf{x}, t)$ 是流体上的压力，并且 $\mathbf{F}(\mathbf{x}, t)$ 是施加到流体的力。在我们的应用中，力是保守的，因此 $\mathbf{F} = -\nabla U$ 表示某个势能函数 $U(\mathbf{x}, t)$ 。

对于海洋表面动力学，由于保守力（如重力），将运动类型限制为称为势流的一类是值得的。这是一种速度具有梯度形式的情况：

$$\mathbf{u}(\mathbf{x}, t) = \nabla \phi(\mathbf{x}, t) \quad (11)$$

这种限制具有重要的效果，将流动的自由度从速度 \mathbf{u} 的三个分量减少到势速度函数 ϕ 的单个分量。实际上，使用这种梯度形式，四个Navier-Stokes方程转化为两个方程

$$\frac{\partial \phi}{\partial t} + \frac{1}{2} |\nabla \phi|^2 = -p - U \quad (12)$$

$$\nabla^2 \phi(\mathbf{x}, t) = 0 \quad (13)$$

方程12称为伯努利方程。作为Navier-Stokes方程的完全非线性简化，Bernoulli 有关3D中伯努利方程的数值模拟的更多详细信息，请参见[23]。

3.2 线性化

为了我们的目的，我们希望减少伯努利

通过应用以下两个限制，更进一步地计算等式：将运动方程线性化，并将方程的计算限制为表面本身上的点，忽略表面下方的体积。这看起来像是一个极端的限制，但当与一些海洋现象学知识结合时，这种限制非常有效。

第一个限制是线性化伯努利方程。这是简单的任务去除二次项 $1/2 (\nabla \phi)^2$ 。伊利米

在他们的运动中，波浪不是非常剧烈，至少在主...
 ciple.所以伯努利

所有量 ϕ 、 p 和 U 仍在3D点 \mathbf{x} 处求值
 在表面和水体积中。

第二个限制是仅评估水面上的量要做到这一点，我们必须首先描述我们所指的表面。我们将表面视为动态变化的高度场 $h(\mathbf{x}, t)$ ，其仅为水平位置的函数。

时间 \mathbf{x} 和时间 t 。为了方便起见，我们定义了水面高度为零。利用该定义
 的波高，重力引起的势能项 U 为

$$U = g h \quad (15)$$

g 是重力常数，通常为 9.8 米/秒^2 （公制单位）。仅限于水面
 有几个重要的后果-

序列第一个结果是质量守恒。在不可压缩的Navier-Stokes方程中，质量通过质量通量方程

$$\nabla \cdot \mathbf{u}(\mathbf{x}, t) = 0 \quad (16)$$

当我们选择只考虑势流时，该质量守恒方程变为

$$\nabla^2 \phi(\mathbf{x}, t) = 0 \quad (17)$$

如果我们将位置矢量的水平部分标记为 \mathbf{x} ，使得 $\mathbf{x} = (x, y)$ ，并且 y 是指向水体积的坐标，则质量守恒方程限于表面的详细外观如下所示

$$\frac{\partial \phi}{\partial t} + \frac{1}{2} |\nabla \phi|^2 = -p - U \quad (18)$$

现在，当你看到这个方程，发现 ϕ 现在只依赖于表面上的 \mathbf{x} ，你可能会想扔掉方程中的 $|\nabla \phi|^2$ 部分，因为看起来没有依赖性。这将产生无用的结果。反而

在偏微分方程的这个奇怪的世界中，更好的方法是允许 ϕ 是一个任意函数（至少对于这个质量守恒方程），并将 y -导数算子定义为

$$\frac{\partial}{\partial y} = \frac{\partial}{\partial h} \quad (19)$$

因此算子 $\frac{\partial}{\partial y}$ 为零。每当我们需要一个垂直导数时，我们将使用这种方法来计算水面上的任何量。当然，这引入了一个包含平方根函数的不寻常的运算符。

另一个结果是，我们只局限于表面，压强基本上保持不变，我们可以选择让这个常数为0。有了这个和其他的限制，伯努利

$$\phi(\mathbf{x}, t) \quad (20)$$

有一个最后的方程，必须重写这种情况。回想一下，速度势 ϕ 用于计算作为梯度的3D流体速度， $\mathbf{u} = \nabla \phi$ 。速度的垂直分量现在必须使用等式19。此外，流体的垂直速度与表面高度的速度相同。结合这些我们得到

$$\frac{\partial h}{\partial t} + \frac{1}{2} |\nabla h|^2 = -p - U \quad (21)$$

$$\frac{\varphi}{U} = p_{-} - \text{普雷特} \quad (14)$$

这最后两个方程20和21是求解表面运动所需他们可以

也可以转换成一个方程。例如，如果对方程21的时间求导，并用20代替速度势的时间求导，我们得到表面高度演变的单一方程。

$$\frac{\partial^2 h(\mathbf{x})}{\partial t^2} = -g \nabla^2 h(\mathbf{x}, t) \quad (22)$$

这仍然涉及不寻常的操作者²。然而，在IV处花费两个以上的²将使其变为更正常的两个方程的维拉普拉斯算子

$$\frac{\partial^4 h(\mathbf{x})}{\partial t^4} = g \nabla^4 h(\mathbf{x}, t)$$

这种形式通常是建立表面波方程数学解的起点

3.3 分散

事实证明，我们建立的表面高度方程-这一课体现在一个简单的数学关系中，称为色散关系，这是本节的重点。这里我们的目标是从上述数学中获得该简单表达式，理解其一些含义，并证明，即使它看起来不切实际地简单化，色散关系实际上存在于自然海浪中，并且可以通过实验测量。

为了方便起见，我们仅使用方程23中的表面演化方程的版本。可以使用其他两个版本，并且使用稍微不同的一组操纵来得出相同的答案。注意，表面高度的运动方程在表面高度中是线性的。因此，与任何线性微分方程一样，方程的通解是通过将任意数量的特解相加而得到的。因此，我们需要找到一个具体的解决方案。

结果是所有的特定解都有如下形式

$$h(\mathbf{x}, t) = h_0 \exp\{i(\mathbf{k} \cdot \mathbf{x} - \omega t)\} \quad (24)$$

4)

2D向量 \mathbf{k} 以及数字 ω 和 h_0 在这一点上是通用参数。如果我们在方程23中使用这种形式的解，它会变成这样的代数方程：

$$h_0 \cdot \omega^4 - g^2 k^2 = 0 \quad (25)$$

（并且 k 是向量 \mathbf{k} 的幅度）。对于这种解决方案，只有两种可能性：

1. $h_0 = 0$ 。那么表面高度是平的，解就不是很有趣了。
2. $\omega = \pm \sqrt{gk}$ 和 h_0 可以是任何值。这是一个有趣的解决方案。

我们在这里发现的是，整个Navier-Stokes流体动力学问题，简化为水面的演化方程并近似为可以相对容易地求解的某个方程，相当于施加表面高度运动的时间频率 ω 与传播波 $\mathbf{k}=\mathbf{k}$ 的空间范围相连接的约束的单个方程。该关系 $\omega=gk$ 是前面提到的色散关系

特别注意，没有对振幅 h_0 施加约束。但是，如果纳维-斯托克斯方程对振幅没有任何说明，我们如何给它一个值？一种方法是通过高度和其垂直速度施加初始条件。对于下一节中的海洋表面模拟，我们将使用

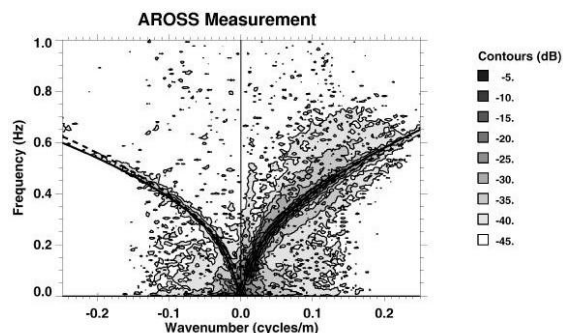


图3：通过3D PSD的切片，显示观察到的波能非常好地遵循深水色散关系

另一种方法，一种统计过程，以产生随机实现的振幅，引导测量的方差属性的波高在开放的海洋。

因此，剩下的问题是如何合理的离散关系，模拟现实的海洋表面波？这就是大量实验研究可以发挥作用的地方。虽然已经有几十年的研究使用放置在水中的设备直接测量波浪运动的海洋波浪特性，在这里，我们看看一些相对较新的研究，涉及测量波浪特性远程与相机在一个平面上。

AROSS[24]是一种安装在小型飞机头部特殊软管中的全色相机附接到摄像机的是导航和GPS仪器，其允许针对每个帧测量摄像机位置、观看方向和定向。当飞机绕着海洋上空的一个点飞行一个圆形轨道后，这些数据可以用来将海洋的图像重新映射到一个公共的参考系中，这样飞机的运动就被去除了（除了照明变化）。这种重新映射允许研究人员在一些数据处理中使用许多帧海洋图像，通常为1-2分钟，以寻找色散关系。

AROSS图像经受的数据处理产生了称为3D功率谱密度（PSD）的东西。这是通过在时间上对图像的时间序列进行傅立叶变换以及在图像的两个空间方向上进行傅立叶变换来获得的。这3个傅立叶变换的输出是对于每个空间和时间频率，与幅度 $h_0(\mathbf{k}, \omega)$ 密切相关的量。然后，以某种方式对这些进行绝对平方和平滑或平均，使得输出是 $h_0(\mathbf{k}, \omega)$ 的统计平均值的数值近似。

但是3D PSD如何帮助我们确定自然界中出现的色散关系？如果图像仅发现色散约束表面波，则对于不满足色散关系的所有 \mathbf{k} 、 ω 值，因此，我们通常期望3D PSD仅在 \mathbf{k} 、 ω 值的窄集合中具有显著

图3显示了由AROSS图像生成的3D PSD图[24]。根据3D体积PSD，该绘图图形是通过体积切片的平面。当像这样切片时，色散关系是切片上的曲线，如两条虚线所示。将数据绘制为PSD强度的等高线，通过右侧的键进行颜色编码在色散关系曲线之后的PSD水平比在其他区域中高得多。这表明，在所有尺度上的表面波的运动包括一个非常强的色散关系的运动风格。当然，还有其他类型的运动，PSD图显示为远离DIS-A的强度水平。



图4: 1986年在加利福尼亚州祖玛海滩附近收集视频数据的地点

persion曲线但弥散运动是这一数据的最强特征。

相对简单的实验可以由任何人进行，只要有摄像机和山顶俯瞰海洋。例如，图4是来自视频片段的帧，示出了水进入加利福尼亚州祖玛海滩附近的海滩。1986年，摄像机位于俯瞰海滩的山上1993年，图中所示的视频帧区域被数字化，以产生仅包含水面的帧的时间序列。

图5示出了来自图像数据的实际3D PSD。沿着我们讨论的色散关系有两个明显的分支，浅水影响没有明显的修改。还有第三个分支，其近似于位于前两个分支之间的直线。对视频的检查表明，该分支来自视频帧的一部分中漂浮在水面上的表面活性剂层，并且以恒定速度移动不包括表层，该数据清楚地表明色散关系的有效性，并表明表面波的线性化模型的有效性。

4 实用海浪算法

在本节中，我们将重点介绍建立海浪高度场的算法和实际步骤。虽然我们将主要采用基于快速傅立叶变换（FFT）的方法，但我们首先介绍一种称为Gerstner波的更简单的描述。这是一个很好的起点，有几个原因：与FFT相比，数学相对较轻，可以引入几个重要的在讨论了郭士纳波之后，我们将讨论更复杂的FFT方法，该方法可以产生更真实的这些被称为线性波当然不是故事的全部，因此我们还讨论了一些方法，海洋学家通过这些方法将描述扩展到

在讨论的过程中，我们将看到风速、表面张力和重力加速度等量是如何进入算法的实际实现的。

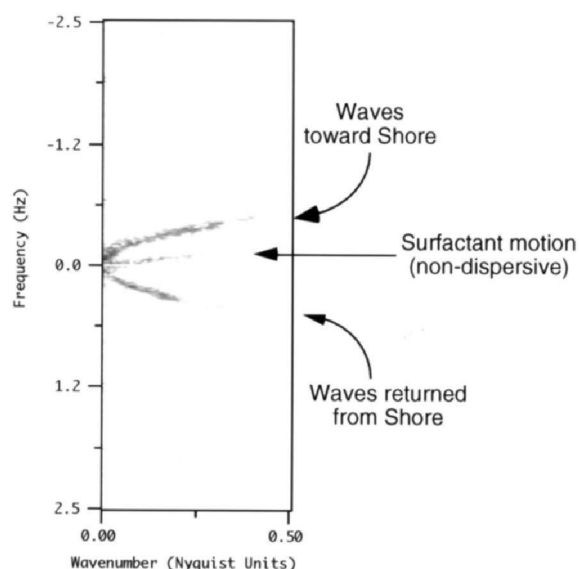


图5: 来自处理后视频数据的3D功率谱密度灰度图的切片。

4.1 郭士纳波

Gerstner波最早是在大约200年前作为流体动力学方程的近似解被发现的。在计算机图形学中的第一个应用似乎是Fournier和Reeves在1986年的工作（前面引用过）。物理模型是根据表面上各个点的运动来描述表面作为一个很好的近似，当波浪经过时，水面上的点如果未受扰动表面上的点标记为 $\mathbf{x}_0 = (x_0, z_0)$ 并且未受高度为 $y_0 = 0$ ，则当振幅为 A 的单个波经过时，表面上的点在时间 t 处移位至

$$\mathbf{x} = \mathbf{x}_0 - (\mathbf{k}/k) A \sin(\mathbf{k} \cdot \mathbf{x}_0 - \omega t) \quad (26)$$

$$\text{产率} = A \cos(\mathbf{k} \cdot \mathbf{x}_0 - \omega t) \quad (二十七)$$

在这些表达式中，称为波矢量的矢量 \mathbf{k} 是指向波的行进方向的水平矢量，并且具有通过下式与波的长度（ λ ）相关的幅度 k ：

$$k = 2\pi/\lambda \quad (28)$$

频率 ω 与波矢量有关，如稍后所讨论的

图6示出了两个示例波形，每个波形具有不同的无量纲振幅 kA 值。对于值 $kA > 1$ ，波是周期性的，并且随着 kA 接近1，在波的顶部处示出变陡对于 $kA > 1$ ，在波的顶部形成环，并且“波表面的内部在外部”，这不是特别期望或现实的

到目前为止，郭士纳波是相当有限的，因为它们是一个单一的正弦波水平和垂直。然而，这可以通过对一组正弦波求和而推广到更复杂的轮廓。选取一组波矢量 \mathbf{k}_i 、振幅 A_i 、频率 ω_i 和相位 ϕ_i ，其中 $i = 1, \dots, N$ ，以获得表达式

4实用海浪算法

$$\eta = \sum_{i=1}^N \frac{(\mathbf{k}_i/k_i)}{k_i} A_i \sin(\mathbf{k}_i \cdot \mathbf{x}_0 - \omega_i t + \varphi_i) \quad (29)$$

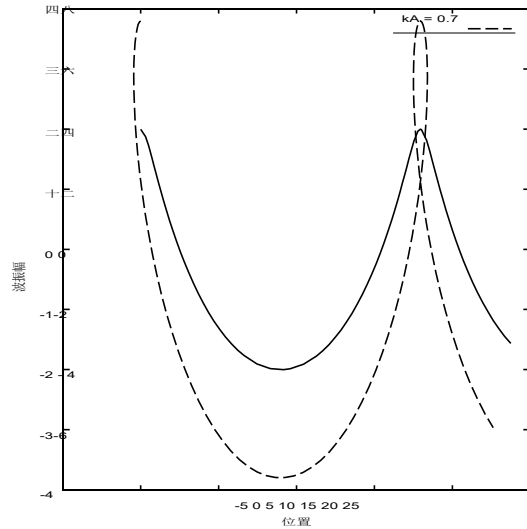


图6: 具有不同相对振幅和波长的两个单模Gerstner波的轮廓。

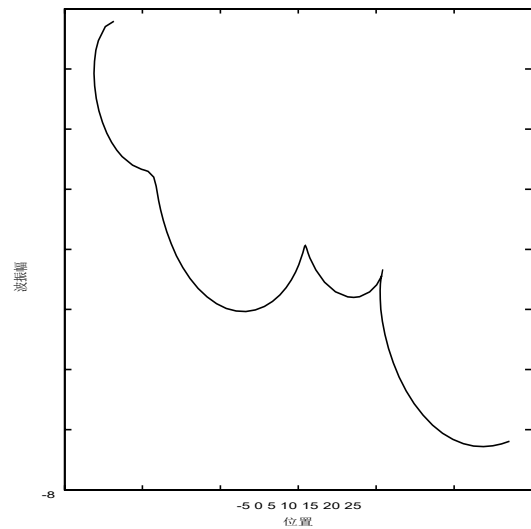


图7: 3模式Gerstner波的轮廓。

$$y = \sum_{i=1}^N A_i \cos(\mathbf{k}_i \cdot \mathbf{x}_0 - \omega_i t + \varphi_i) \quad (三十)$$

图7示出了集中具有三个波的示例。以这种方式可以获得感兴趣的和复杂的形状。

4.2 设置波浪动画：色散关系

Gerstner波的动画行为由为每个分量选择的频率 ω_i 的集合确定。对于水波，在这些频率与它们对应的波矢量的幅度 k_i 之间存在公知的关系。在深水中，底部可能被忽略，这种关系是

$$\omega^2(k) = gk \quad (三十一)$$

参数 g 是重力常数，标称为 $9.8 \text{ m/se } c^2$ 。这种色散关系适用于Gerstner波，并且也适用于接下来介绍的基于FFT的波。

存在修改色散关系的若干条件当底部与波浪的长度相比相对较浅时，底部对波浪具有阻滞作用。对于平均水位以下深度为 D 的海底，色散关系为

$$\omega^2(k) = gk \tanh(kD) \quad (3)$$

2) 注意，如果底部非常深，则 \tanh 的行为函数将该色散关系减小到先前的色散关系。

修改色散关系的第二种情况是表面张力。波长约为1厘米或更小的非常小的波有一个附加项：

$$\omega^2(k) = gk(1 + k^{2L}), \quad (33)$$

并且参数 L 具有长度单位。它的大小是表面张力产生影响的尺度。

使用这些色散关系，很难创建一个连续循环的水面帧序列。

例如，为了使序列在一定量的时间 T 之后重复，需要所有频率都是基频

$$\omega_0 \equiv \frac{2\pi}{T} \quad (三十四)$$

然而，当波矢 \mathbf{k} 分布在规则的格子上时，不可能将色散产生的频率也布置在具有间隔 ω_0 的均匀格子上。

解决方案是不使用色散频率，而是与它们接近的集合。对于给定的波数 \mathbf{k} ，我们使用频率

$$\bar{\omega}(\mathbf{k}) = \frac{\sum \omega(\mathbf{k})}{\sum 1} \quad (35)$$

其中 $[[a]]$ 表示取 a 的值的整数部分，并且 $\omega(\mathbf{k})$ 是感兴趣的 n/y 色散关系。频率 $\bar{\omega}(\mathbf{k})$ 是分散表面的量化，并且水面的动画在时间 T 之后循环，因为量化的频率都是 ω_0 的整数倍。图8绘制了原始色散曲线以及两种选择的量化色散曲线重复时间 T 。

4.3 统计波动模型与傅里叶变换

海洋学文献倾向于淡化郭士纳波作为一个现实主义海洋模型。相反，使用统计模型，结合实验观察。在统计模型中，波高被认为是水平位置和时间 $h(\mathbf{x}, t)$ 的随机变量。

统计模型还基于将波高场分解为正弦和余弦波之和的能力。这种分解的价值在于波的振幅具有良好的数学和统计特性，使得建立模型更简单。在计算上，分解使用快速傅立叶变换(ffts)，这是一种快速评估和的方法。

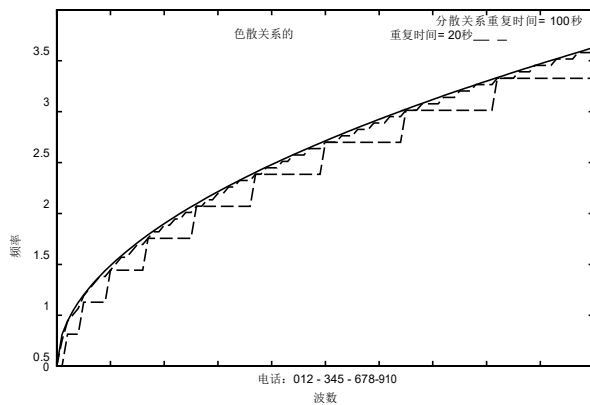


图8: 对于20秒和100秒的重复时间, 连续色散曲线 $\omega = gk$ 注意, 对于较长的重复时间, quantized是对原始曲线的更接近的近似。

波高场的基于 \mathbf{f} 的表示表达了在水平位置 $\mathbf{x} = (x, z)$ 处的波高 $h(\mathbf{x}, t)$ 作为和具有复杂的、随时间变化的振幅的正弦曲线:

$$h(\mathbf{x}, t) = \sum_{\mathbf{k}} \tilde{h}(\mathbf{k}, t) \exp(i\mathbf{k} \cdot \mathbf{x}) \quad (36)$$

钾

其中 t 是时间, \mathbf{k} 是二维向量, 分量 $\mathbf{k} = (k_x, k_z)$, $k_x = 2\pi n/L_x$, $k_z = 2\pi m/L_z$, n 和 m 是整数, 边界为 $N/2 \leq n \leq N/2$ 和 $M/2 \leq m \leq M/2$ 。fft过程在离散点 $\mathbf{x} = (nL_x/N, mL_z/M)$ 处生成高度场。其他点处的值也可以通过切换到离散傅立叶变换来获得, 但是在许多情况下, 这是不必要的并且在此不应用。高度振幅傅立叶分量 $\tilde{h}(\mathbf{k}, t)$ 确定表面的结构。本小节的其余部分涉及在图1中生成随机振幅集。

一种与海洋现象学相一致的方式

出于计算机图形学的目的, 还需要波高场的斜率矢量, 以便找到表面法线、入射角以及光学建模的其他方面。计算斜率的一种方法是通过 \mathbf{f} 网格点之间的有限差分, 由一些2D向量 $\Delta \mathbf{x}$ 水平分离。虽然有限差分在存储器要求方面是有效的, 但它可能是一个很差的近似波的斜率与小波长斜率向量的精确计算可以通过使用更多ffts来获得:

$$g(\mathbf{x}, t) = h(\mathbf{x}, t) = \sum_{\mathbf{k}} \tilde{h}(\mathbf{k}, t) \exp(i\mathbf{k} \cdot \mathbf{x}) \quad (37)$$

钾

根据该 \mathbf{f} 表示, 有限差分方法将用与

$$\exp(i\mathbf{k} \cdot \Delta \mathbf{x}) - 1 \quad (38)$$

对于小波长的波, 其不能很好地近似波高的梯度只要可能, 通过等式37中的 \mathbf{f} 进行斜率计算是优选的方法。

\mathbf{f} 表示在具有水平维度 $L_x L_z$ 的贴片上产生波, 在该贴片之外, 表面是完全周期性的。在实际应用中, 斑块大小在一侧上从10米到2公里不等, 每个方向上的离散样本点的数量高达2048个 (即, 2048个)。网格为 2048×2048 , 或超过400万个波)。贴片可以根据需要在区域上无缝平铺然而, 这种平铺延伸的结果是波场中存在人为的周期性。只要片尺寸与视场相比较大, 该周期性就不明显。此外, 如果相机靠近表面, 使得有效水平线在一个或两个片长度之外, 则周期性在观看方向上将不明显, 但是其可以作为跨视场的重复结构而明显。

海洋学研究表明, 方程36是在开阔海洋中自然发生的风浪的合理表示。对海洋表面的一些波浪浮标、照片和雷达测量的统计分析表明

W/V高度幅度 $\tilde{h}(\mathbf{k}, t)$ 几乎是统计上稳定的。具有空间频谱的独立高斯波动, 由

$$\tilde{h}(\mathbf{k}) = \frac{1}{\sqrt{P(\mathbf{k})}} \sum_{\mathbf{x}} h(\mathbf{x}, t) \exp(-i\mathbf{k} \cdot \mathbf{x})$$

对于由括号表示的数据估计总体平均值 $\langle \tilde{h}(\mathbf{k}) \rangle$ 有几个分析半经验模型的波谱 $P_h(\mathbf{k})$ 。在充分发展的海中, 比毛细波大的风驱动波的一个有用的模型是菲利普斯谱

$$P_h(\mathbf{k}) = \frac{\exp[-1/(kL)^2]}{k^4} |\mathbf{A}|^2, \quad (40)$$

其中 $L = V^2/g$ 是由速度为 V 的连续风引起的最大可能的波浪, g 是重力常数, \mathbf{u} 是风向。 A 是一个数字常量。菲利普斯谱中的余弦因子 k_w^2 消除了垂直于风向的波浪这个模型虽然相对简单, 但在波数

克雷 一个简单的解决方法是抑制小于小长度 LL 的波, 并通过乘法因子修改Phillips谱

$$\exp(-k^2 \ell^2) \sum_{\mathbf{k}} \quad (四十一)$$

当然, 你可以自由地

4.4 建立随机海浪高度场

水波高度场的实现是根据到目前为止详细阐述的原理创建的: 具有规定形式的空间谱的高斯随机数。这在傅立叶域中直接最有效地波高场的傅立叶振幅可以产生为

$$\tilde{h}_0(\mathbf{k}) = \frac{1}{\sqrt{2}} \sqrt{P_h(\mathbf{k})} \xi(\mathbf{k}),$$

其中 ξ_r 和 ξ_i 是来自高斯随机数生成器的普通独立抽取, 平均值为0, 标准差为1。高斯分布的随机数倾向于遵循关于海浪的实验数据, 但是当然可以使用其他随机数分布。例如, 对数正态分布可以用于产生“间歇性”变化的高度场即, 波浪非常高或几乎平坦, 两者之间相对较小。

给定色散关系 $\omega(\mathbf{k})$, 在时间 t 的波场实现的傅立叶振幅为

$$\tilde{h}(\mathbf{k}, t) = \tilde{h}_0(\mathbf{k}) \exp\{i\omega(\mathbf{k}) t\} + \tilde{h}_0^*(-\mathbf{k}) \exp\{-i\omega(\mathbf{k}) t\} \quad (43)$$

这种形式通过将 w “向左”和“向右”传播来呈现复杂的 x 共轭性质 $\hat{h}^*(\mathbf{k}, t) = \hat{h}(\mathbf{k}, t)$ 。除了实现简单之外，该表达式对于计算 $h(\mathbf{x}, t)$ 也是有效的，因为它依赖于ffts，并且因为可以计算任何选定时间的波场而不计算任何其他时间的场。

实际上，傅立叶网格需要多大？选择什么样的尺度范围才是合理的？如果你想更快地生成波高，你会怎么做？让我们看看这些问题。

傅立叶网格应该有多大？ N 和 M 的值可以在16和2048之间，以2的幂为单位。对于许多情况，范围128到512中的值是足够的。对于极其精细的曲面，可以使用1024和2048。例如，在电影《水世界》和《泰坦尼克号》中使用的波场大小为2048-2048，网格点之间的间距约为3厘米。如果值大于2048，则应小心，因为浮点计算的数值精度限制可能会变得明显。

选择什么样的尺度范围才是合理的？这个问题的答案归结为选择 L_x 、 L_z 、 M 和 N 的值。任一方向上的最小平面是 $dx = L_x/M$ 或 $dz = L_z/N$ 。通常， dx 和 dz 不需要低于2cm左右。在这个尺度之下，波浪作用的量与其余的波浪相比是小的。此外，2厘米以下的波的物理行为开始呈现出非常不同的特征，涉及表面张力和“非线性”过程。从谱的形式来看，波长大于 v^2/g 的波被抑制。所以要确保 dx 和 dz 比 v^2/g 小相当大的量（10 - 20%）。1000），否则大部分有趣的波都会丢失。这个...
crot到逼真的波浪（例如，图12（a）与图12（c）相比）的一个优点是使 M 和 N 尽可能大。

如何在最短时间内生成波高场？计算的耗时部分是快速傅立叶变换。运行在1+ GHz CPU上，512 512个FFT可以以接近交互的速率生成。

4.5 示例：高度字段和渲染

我们现在转向使用上面讨论的fft方法创建的波的一些示例。我们将以两种格式显示波：作为灰度级与波高成比例的灰度图像；以及使用几个不同的渲染包的海洋景观的渲染来说明什么是可能的。

在第一组示例中，网格大小被设置为 $M=N= 512$ ，其中 $L_x=L_z= 1000$ 米。风速是在 $V= 31$ 米/秒处的烈风，在 x 方向上移动。还使用了 $l= 1$ 米的小波截止。图9是波高的灰度表示：较亮意味着较高，较暗意味着较低的高度。虽然由这里描述的fft算法产生，但是图9显然不是水高度场。检查图10可能会有所帮助，图10是斜率 x 分量的灰度图。这看起来更像图9的水波。怎么回事？

图9和图10展示了水表面光学的结果，在下一节中讨论：表面结构的可见质量往往受到波浪斜度的强烈影响。我们将详细讨论这一点，但现在我们将总结说，水的反射率是波浪斜率的强函数，以及光线和相机的方向。

为了说明定制光谱模型的一个简单效果，图11是与图11相同的高度场的灰度显示

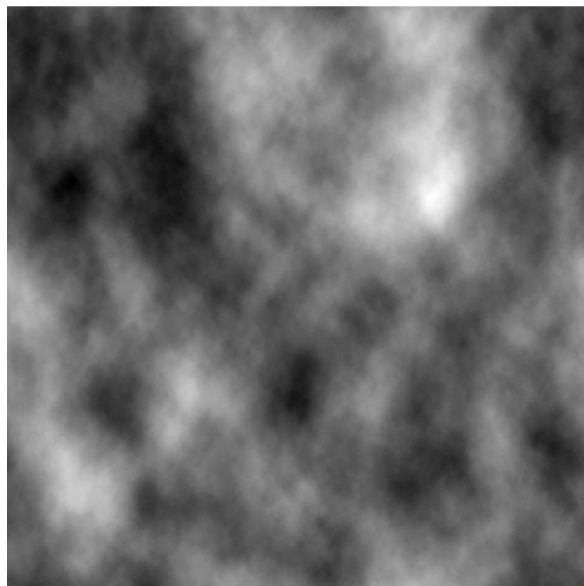


图9：表面波高度实现，以灰度显示

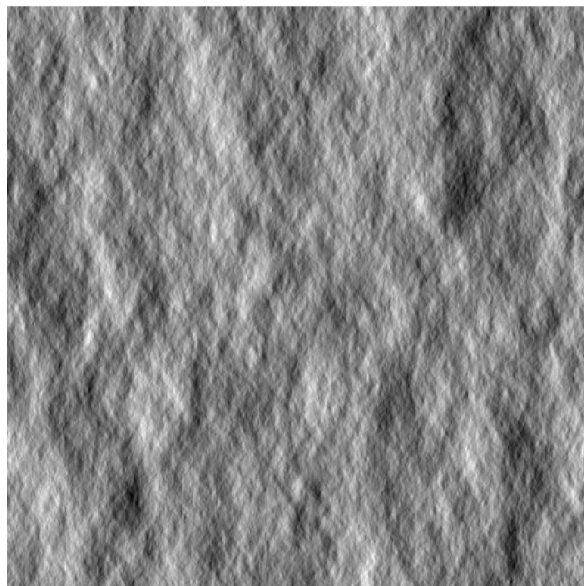


图10：图9中的波高实现的斜率的X分量。

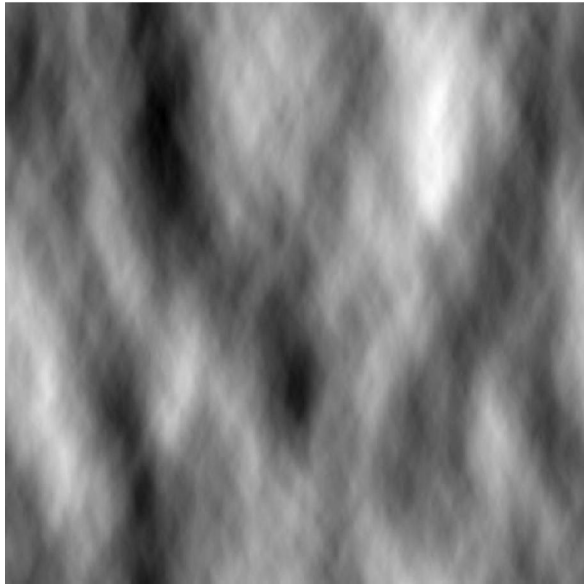


图11：具有增加的方向依赖性的波高实现。

9, 除了方向因子 $|\hat{\mathbf{k}} \cdot \hat{\mathbf{w}}|$ 等式40中的 $k = 2$ 已经被改变为 \mathbf{k} 。表面显然更符合风向。^[6]

高度场的下一个示例使用BMRT中相对简单的着色器, 即符合Renderman的光线跟踪器。着色器将在下一部分中显示。图12示出了水面的三个渲染图, 改变网格数 M 和 N 的大小, 并使小平面大小 dx 和 dz 与 $1/M$ 和 $1/N$ 成比例。因此, 当我们从上图到下图时, 小平面的尺寸变得更小, 我们可以看到渲染中细节数量增加的效果。显然, 更多的波浪细节有助于构建逼真的曲面。

作为最后的示例, 图13是在Arete Entertainment的商业包RenderWorld中渲染的图像。该渲染包括大气和水体积散射光的效果这些将在下一节中讨论。但很明显, 使用 \mathbf{f}_t 预编码从随机数生成的波高场可以产生一些很好的图像。

4.6 惊涛骇浪

在这一节中, 我们将简要地讨论如何创建看起来波涛汹涌的波浪。到目前为止, 由 \mathbf{f}_t 方法产生的波具有圆形的波峰和波谷, 这使它们看起来像是晴天。即使在相当好的天气里, 特别是在大风或暴风雨中, 波浪的顶部也是尖锐的, 底部是平坦的。波浪轮廓的这种斩波的程度足够高的波浪 (例如斜率大于约 $1/6$), 最终在顶部破裂, 产生一组新的泡沫、飞溅、气泡和喷射的物理现象。

该方法的出发点是表面运动的基本流体动力学方程。这些方程用两个动力学场表示: 表面高度和表面上的速度势, 并且从贯穿水和空气的体积的流体的Navier-Stokes描述导出, 包括界面上方和下方。奶精

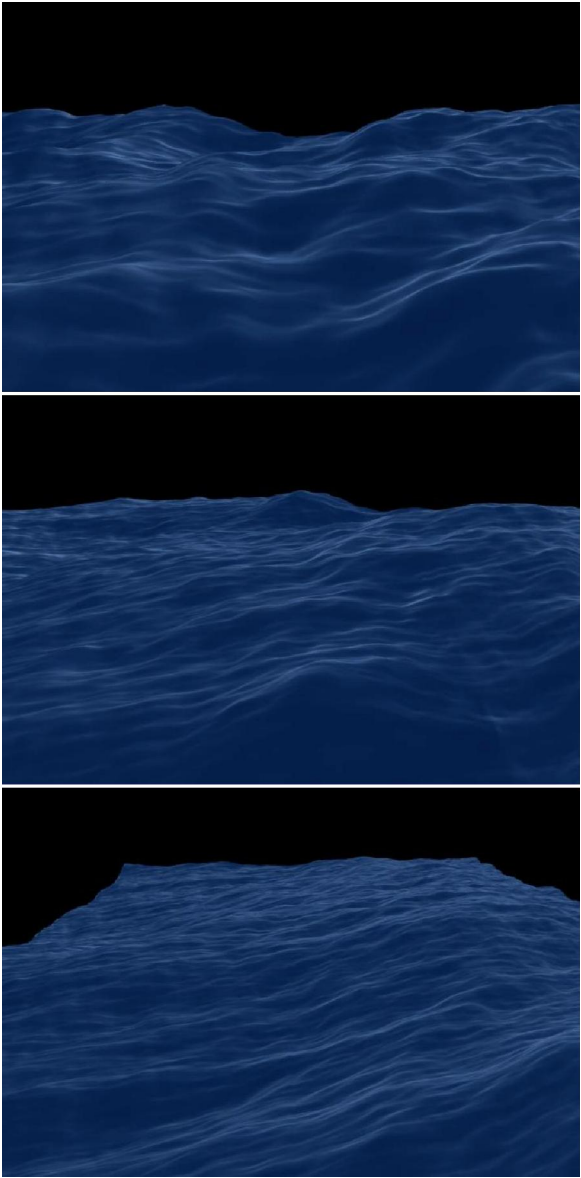


图12：具有（顶部）相当低数量的波（小平面尺寸= 10 cm）的波的渲染, 具有很少的细节;（中间）相当好数量的波（小平面尺寸= 5cm）;（底部）具有最多细节的大量波（小平面尺寸= 2.5cm）。

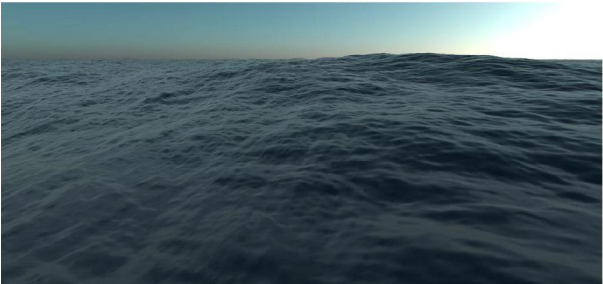


图13：在具有模型大气和复杂阴影的商业包中呈现的波高场的

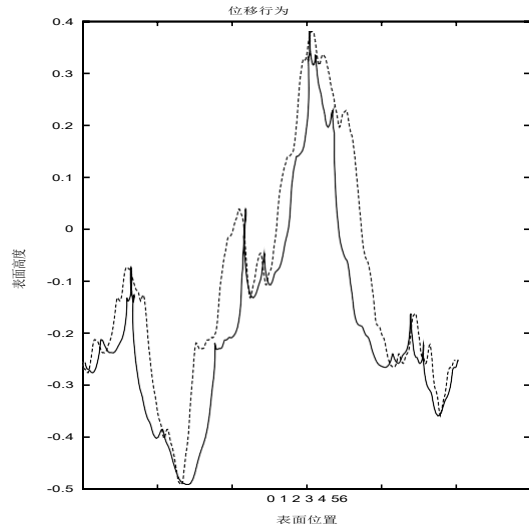


图14: 具有和不具有位移的波高剖面的比较虚曲线是由 \mathbf{m} 表示产生的波高实曲线是使用等式44位移的高度场。

等人[16]开始应用称为“Lie变换技术”的数学方法来生成高程和速度势的“正则变换”序列。这个复杂的数学过程的好处是将高度和速度势转换变换后的情况实际上就是我们讨论过的简单海平面高度场，包括本文中使用的相同色散关系的演变。从那里开始，逆李变换原则上将我们的唯象解转换为动态更准确的解。然而，Lie变换在3维中难以操纵，而在二维中已经获得了精确的结果。基于二维的精确结果，提出了三维解形式的外推：波的水平位移，位移随波局部变化

在 \mathbf{m} 表示中，使用高度场的傅立叶振幅来计算2D位移向量场，如

$$\mathbf{D}(\mathbf{x}, t) = \sum_{\mathbf{k}} \frac{\mathbf{k}}{k} h(\mathbf{k}, t) \exp(i\mathbf{k} \cdot \mathbf{x}) \quad (4)$$

使用这个向量场，表面的网格点的水平位置现在是 $\mathbf{x} + \lambda \mathbf{D}(\mathbf{x}, t)$ ，高度 $h(\mathbf{x})$ 如前所述。参数 λ 不是原始猜想的一部分，而是缩放位移向量的重要性的方便方法。这种推测的解决方案不直接改变波高，而是以依赖于高度场的空间结构的方式扭曲表面点的水平位置。然而，这种扭曲的特定形式实际上锐化了高度场中的峰图14示出了沿着一个方向的波高的轮廓。

在一个模拟的表面。这清楚地表明，该解的位移形式与原算法相似。

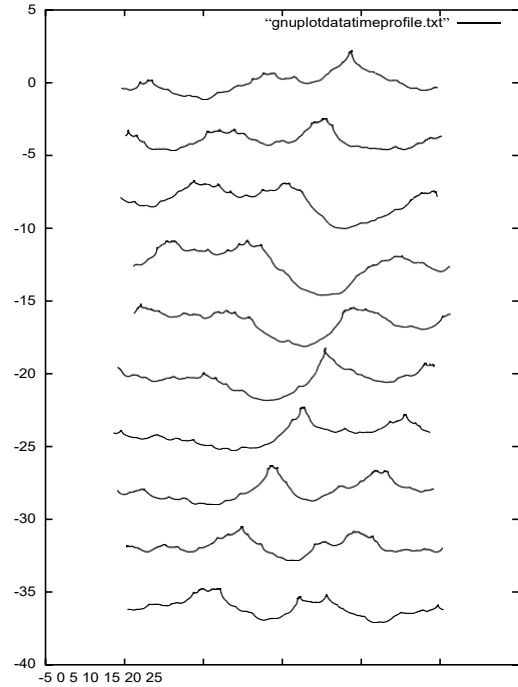


图15: 波浪表面的轮廓的时间序列从上到下，曲线之间的时间为0.5秒。

然而，在那种情况下，应用于正弦形状的位移行为是表征水面结构的主要方法，并且在这里它是对已经有用的波高表示的修改。

图15显示了这些起伏波动在演化过程中的表现。波浪的顶部形成一个尖锐的尖点，它逐渐变圆，不久就消失了。

在图14中可以看到这种生成波涛的方法的一个“问题”。在一些波的顶部附近，表面实际上穿过自身并反转，因此表面的向外法线指向内部。这是因为波分量的振幅可以足够大以产生重叠的这是很容易击败简单地减少

缩放因子 λ 的大小。出于计算机图形的目的我们不会讨论

在这里，除了注意到为了使用该重叠区域，需要简单且快速的测试来确定该效果正在发生之外，如何执行这样的扩展幸运的是，有这样一个简单的测试，其形式为从 \mathbf{x} 到 $\mathbf{x} + \lambda \mathbf{D}(\mathbf{x}, t)$ 的变换的雅可比矩阵。雅可比矩阵是变换唯一性的度量。当位移为零时，雅可比矩阵为

1.当存在位移时，雅可比矩阵具有以下形式

$$J(\mathbf{x}) = J_{xx}J_{yy} - J_{xy}J_{yx},$$

(4)

5) 具有单独的项

$$J_{xx}(\mathbf{x}) = 1 + \lambda \frac{D_x(\mathbf{x})}{D_y(\mathbf{x})}$$

在第4节中讨论的用于构建Gerstner波的Rithm [12]。在

$$J_{yy}(\mathbf{x}) = 1 + \lambda_y$$

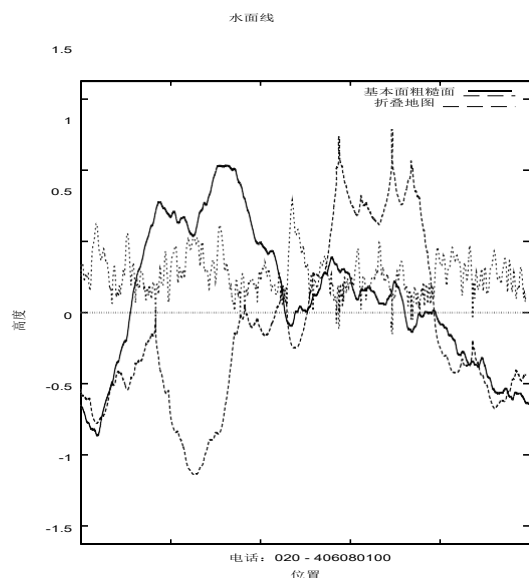


图16: 具有和不具有位移的波高轮廓。还绘制了起伏波剖面的雅可比映射。

$$\begin{aligned} J_{yx}(\mathbf{x}) &= \frac{D_y(\mathbf{x})}{\lambda} \\ J_{xy}(\mathbf{x}) &= \lambda \frac{D_x(\mathbf{x})}{g} = J_{yx} \end{aligned}$$

并且 $\mathbf{D} = (D_x, D_y)$ 。雅可比矩阵表示重叠波的存在，因为其值在重叠区域中小于零。例如，图16绘制了没有位移的基本波、具有位移的波以及用于起伏波的 J 值的轮廓（标记为在起伏的表面中的 λ 利用该信息，如果需要，提取重叠区域并将其用于其他目的应该相对

但是，通过更仔细地研究这种折叠准则，我们可以从这些折叠波中学到更多的东西。雅可比矩阵来自于一个 2×2 矩阵，它度量了起伏波动映射 \mathbf{x} 的局部唯一性 $\mathbf{x} + \lambda \mathbf{D}$ 。该矩阵通常可以根据本征向量和本征值写为：

$$J_{ab} = J_- \bar{e}_a + J_+ \bar{e}_b, \quad (a, b = x, y) \quad (46)$$

其中 J_- 和 J_+ 是矩阵的两个特征值，排序使得 $J_- J_+$ 。相应的正交特征向量是 \bar{e} 和 \bar{e} 关于 $\text{iv el } y$ 。从这个表达式中，雅可比矩阵就是 $J = J_- J_+$ 。

折叠的标准是 J_0 意味着 $J_- < 0$ 和 $J_+ > 0$ 。因此，最小特征值就是折叠点的实际信号。此外，特征向量 \bar{e} 指向折叠发生的水平方向所以，处方- 现在的问题是观察最小特征值何时变为负值，并且折叠波的排列平行于最小特征向量。

我们可以用一个例子来说明这个现象。图17和图18示出了海洋表面的两个图像，一个没有波涛汹涌，另一个强烈地施加了波涛汹涌这两个曲面除了起伏波算法之外是相同的。图19示出了沿着穿过表面的切片的两表面的波形轮廓。最后，波浪的剖面是

与最小本征值一起绘制在图20中，显示了折叠和 J_- 的负值之间的明确联系。

顺便说一句，计算此特征值和特征向量矩阵是快速的，因为它们具有解析表达式为

$$\lambda_{\pm} = \frac{1}{2} (J_{xx} + J_{yy}) \pm \frac{1}{2} \sqrt{(J_{xx} - J_{yy})^2 + 4J_{xy}^2} \quad (47)$$

对于特征值和

$$\bar{e}_{\pm} = \frac{1}{\sqrt{q_{\pm}^2}} \begin{pmatrix} 1 \\ q_{\pm} \end{pmatrix} \quad (48)$$

与

$$\bar{q}_{\pm} = \frac{J_{xx} - J_{yy}}{J_{xy}} \quad (49)$$

对于特征向量。

5 扰动的交互波

在第4节中描述的基于傅立叶的水面演变方法有几个限制，使得在真正的交互式应用中不可行。基本上，傅立叶方法在一次FFT计算中计算各处的波高。如果不完全更改计算，则无法选择在曲面栅格 你要么得到整个曲面，要么不计算它。这使得它非常

难以定制从一个位置到另一个位置的波传播问题 因此，如果你想把一些任意形状的物体放入水中，沿着一些用户构建的路径移动它，FFT方法很难计算形状的波浪响应

和物体的运动因此，如果你想让一个看起来很奇怪的船在水中移动，也许会上下移动并改变形状，FFT方法并不是最容易使用的方法此外，如果你想有一个浅的底部，有一个倾斜的海滩或水下海丘或一些其他变化的深度的水，FFT方法再次是一个挑战的使用。

幸运的是，在过去几年中，已经出现了一种替代方案，其允许响应于与水中物体的相互作用和/或可变的底部深度来快速构建水面。该方法的核心是一种根本不使用FFT方法来计算传播的在参考文献[22]中描述了这种交互式方法的基本数学，并将其称为 *iWave*。在这里，我们快速地浏览一下这种方法，并展示一些实际应用的示例。

在其核心，*iWave*方法返回到波高的线性化方程22。我们可以将时间导数转化为时间步长 Δt 的有限时间差，并得到波高的显式表达式：

$$\begin{aligned} h(\mathbf{x}, t + \Delta t) &= 2h(\mathbf{x}, t) - h(\mathbf{x}, t - \Delta t) \\ &\quad - g(\Delta t)^2 \nabla^2 h(\mathbf{x}, t) \end{aligned} \quad (50)$$

该形式可用于明确地将表面波高度从一帧推进到下一帧。当然，困难的部分是如何计算最后一项，即拉普拉斯算子的平方根

解决方案是使用卷积。波高保持在点的矩形网格上（其尺寸不必是2的幂），并且我们利用以下事实：数据值网格上的任何线性运算都可以转换为某种卷积。数值实现的细节包含在[22]中。

但*iWave*真正令人惊叹的特性是，波浪与水面上物体的相互作用是用一个非常简单的

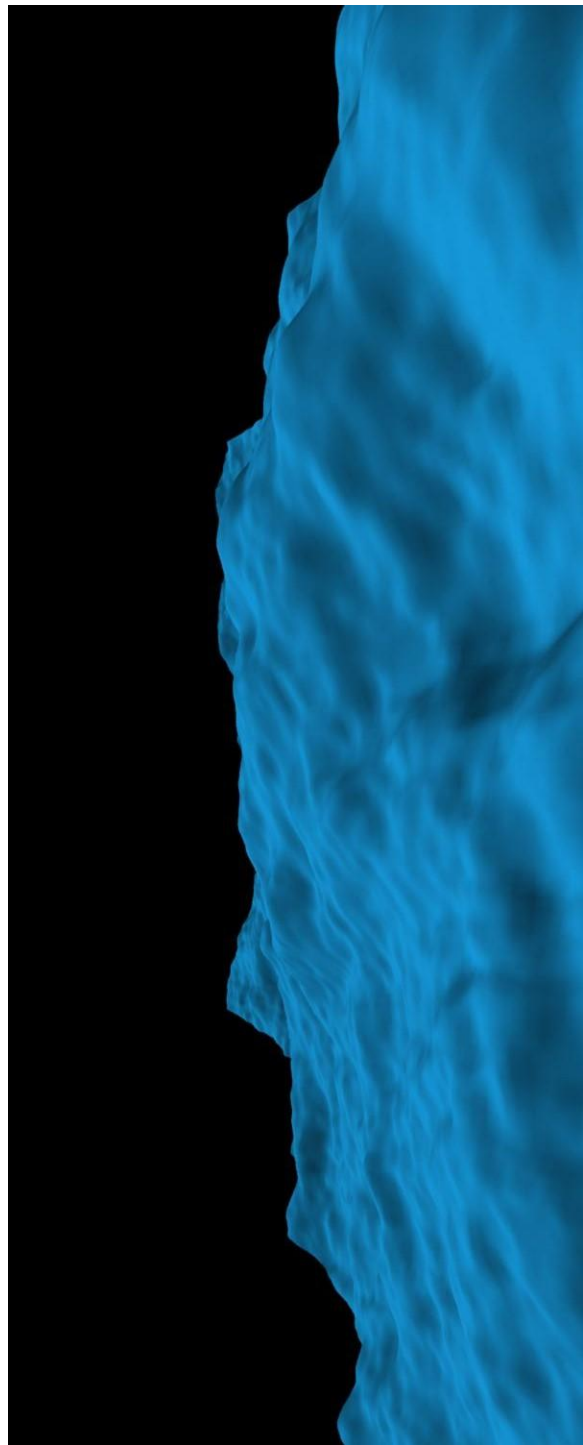


图17：没有应用choppy算法的模拟波浪表面。使用通用塑料着色器在BMRT中渲染。

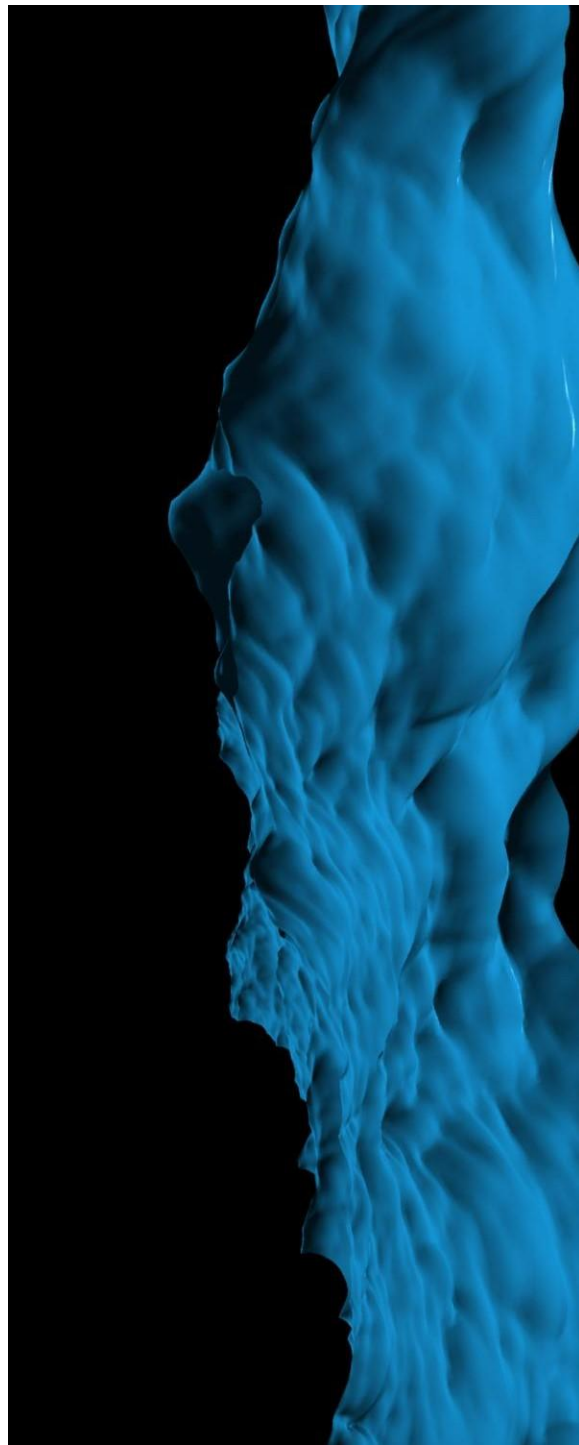


图18：施加强斩波的相同波浪表面。使用通用塑料着色器在BMRT中渲染。

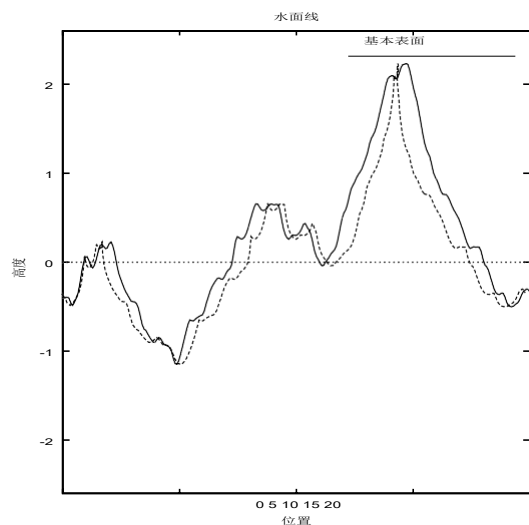


图19：两个曲面的剖面，显示了不连续贴图的效果。

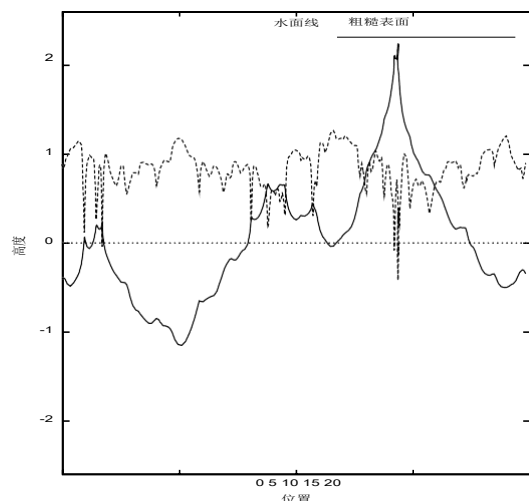


图20：起伏表面轮廓和最小特征值的图。表面的褶皱的位置显然与特征值为负的位置相同。

相当于图像掩蔽的线性操作。为此目的，水中的对象被描述为掩模值的网格，0表示对象存在，1表示对象不存在，并且沿着对象的边缘，0和1之间的值用作抗锯齿。将此遮罩应用于波高栅格，可以有效地移除对象所在栅格点处的波。通过这个简单的过程，入射到物体上的波以逼真的方式反射图21是来自序列的帧，示出了使用iWave与若干不规则形状的对象计算的波高。

通常，模拟与水面上的物体相互作用的波浪应涉及边界条件的仔细处理，使水运动与几何形状边界上的物体相匹配，并强制执行无滑动条件。这是一个复杂且耗时的计算，涉及一定数量的数字巫术。令人惊讶的是，如上所述，像掩蔽波高的简单过程不应该是足够的。然而，使用iWave程序，在模拟中发生了正确的目前尚不清楚这些相互作用在定量上有多准确图22示出了具有与船体相互作用的波浪的高分辨率计算的渲染场景。

5.1 浅水修改

基于FFT方法的波浪模拟可以通过使用等式32中的色散关系来模拟浅水效应。这只适用于平底。这将是很好的模拟波传播到一个海滩，或意大利面浅地下海山，或超过潜艇是刚刚低于表面。iWave方法是做这些事情的好方法。回想一下，iWave方法将数学运算

$$g \sqrt{-2h} \quad (51)$$

转换为卷积，因此有效地 $g \sqrt{-2h}$ 成为卷积核。深度为 D 的浅底将该项改为（与色散关系比较）

$$\frac{g}{\sqrt{-2h}} \cdot \frac{g}{-D^2} \sum h$$

该操作也可以转换为卷积，其中

$\frac{g}{\sqrt{-2h}} \tanh \frac{2D}{h}$ 成为卷积核。

如何将其应用于可变深度的底部？卷积核是一个13x13矩阵[22]。因此，我们可以在深度值 D 的范围内构建一个内核集合。在网格上的每个点处，通过从预构建值的集合进行内插来针对该网格点处的实际深度构建定制内核。图23是来自使用该技术的模拟的波高。底部从右侧的深度倾斜到左侧边缘的深度0。此外，右侧还有一座地下海山。

在真实的浅水传播中，该模拟中有三个重要的行为

1. 浅水区的波比深水区的波具有更大的振幅。
2. 当波浪接近海滩时，它们堆积在一起并具有更高的空间频率。
3. 海面下的山引起波浪的绕射

除了这些功能，iWave还可以计算其他量，如尖波和表面速度。

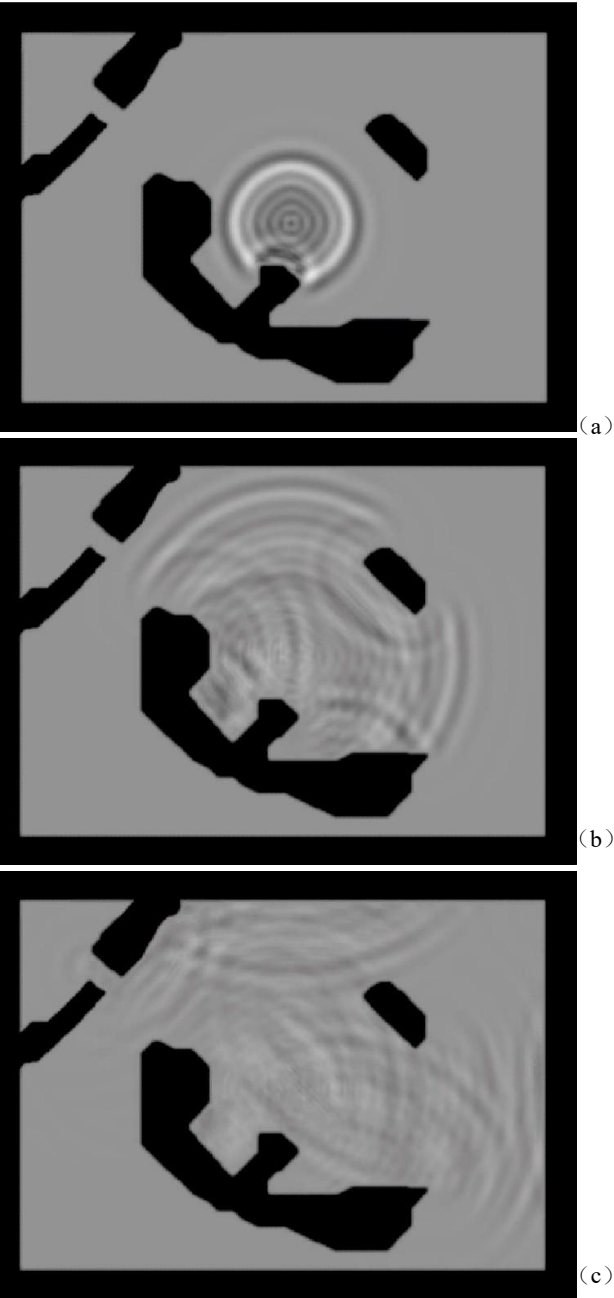


图21: iWave模拟中的一系列帧，显示了水中物体反射的波浪。对象是黑色区域。在左上角，当波在窄通道中传播并出现在拐角区域时，也发生衍射

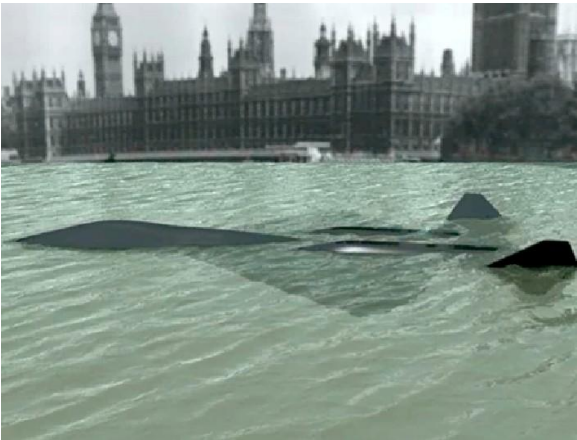


图22: 来自模拟和渲染的帧，显示了波浪与船的相互作用。

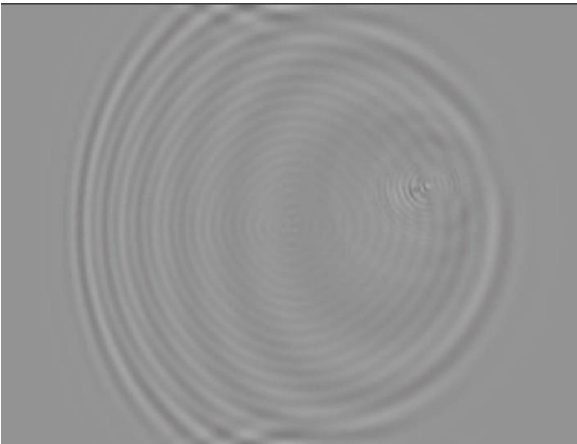


图23: 来自具有可变深度浅底的iWave模拟的帧。

6 表面波光学

海洋表面的光学行为已经相当清楚，至少对于我们在这些笔记中考虑的静止波结构来说是这样。从根本上说，海洋表面是一个近乎完美的镜面反射器，具有很好理解的反射率和透射率函数。在本节中，这些属性进行了总结，并结合到一个简单的着色器渲染。存在表面看起来不是镜面反射体的情况。特别是，从离相机很远的波浪反射的直射阳光 这是由于收集的波小于相机在大距离处可以分辨的波。该机制有点类似于固体表面中导致BRDF的Torrance-Sparrow模型虽然海洋中闪光模式的研究是由Cox和Munk在许多年前开创的，但我所知道的这种BRDF行为的第一个模型是在20世纪80年代早期开发的在本节的最后，我们介绍了概念和条件，陈述了结果，忽略了中间的分析 and 推导。

在这些注释中，特别是在本节中，我们完全忽略一种光学现象：极化极化效应在像水表面的边界界面处可能很强。然而，由于大多数计算机图形学都忽略了极化，我们将继续这一悠久的传统。当然，感兴趣的读者可以找到关于空气-水界面极化效应的文献。

6.1 镜面反射和透射^{测试}

从上方或下方入射到空气-水界面的光线
被分成两个分量：以折射角继续通过界面的透射光线强度

这两种光线中的每一种的反射率和透射率系数减小。这里我们讨论了两条出射光线的方向。在下一小节中讨论系数

6.1.1 反射

众所周知，在完美镜面反射中，反射光线和入射光线相对于表面法线具有相同的角度这适用于所有镜面反射（忽略粗糙化效果），与材质无关。我们在这里建立了一个紧凑的表达出射反射光线。首先，我们需要建立一些符号和几何量。

海洋表面上的三维点可以由水平位置 \mathbf{x} 和波高 $h(\mathbf{x}, t)$ 标记为

$$\mathbf{r}(\mathbf{x}, t) = \mathbf{x} + \mathbf{j} h(\mathbf{x}, t), \quad (53)$$

其中 \mathbf{j} 是指向正上方的单位向量。在点 \mathbf{r} 处，表面的法线直接从表面斜率 $g(\mathbf{x}, t) \equiv \nabla h(\mathbf{x}, t)$ 计算为

$$\mathbf{n}_S(\mathbf{x}, t) = \frac{-\mathbf{g}}{\sqrt{1 + \mathbf{g}^2}} \quad (54)$$

对于从方向 \mathbf{n} 在 \mathbf{r} 处与表面相交的光线，反射光线的方向可以仅取决于入射方向和表面法线。此外，如前所述，表面法线和反射光线之间的角度必须与入射光线和表面法线之间的角度相同。你可以自己验证反射方向 \mathbf{n} 是

$$\mathbf{n}(\mathbf{x}, t) = \mathbf{n} - 2\mathbf{n}_S(\mathbf{x}, t) (\mathbf{n}(\mathbf{x}, t) \cdot \mathbf{n}_S(\mathbf{x}, t)).$$

6.1.2 传输

不幸的是，透射光线的方向并不像反射光线那样简单地表示。在这种情况下，我们有两个指导原则：透射方向仅取决于表面法线和入射方向，以及将入射角和透射角的正弦与两种材料的折射指数相关联的斯涅尔定律。

假设入射光线来自折射率为 n_i 的两种介质中的一种（对于空气， $n=1$ ，对于水， $n=4/3$ ），并且透射光线在折射率为 n_t 的介质中。对于与法线成角度 θ_i 的入射光线，

$$\sin \theta_t = \sqrt{1 - (\mathbf{n} \cdot \mathbf{n}_S)^2} = |\mathbf{n} \times \mathbf{n}_S| \quad (56) \text{ 透射光线将与}$$

$$\sin \theta_t = |\mathbf{n} \times \mathbf{n}_S|. \quad (57)$$

斯涅耳

$$n_i \sin \theta_i = n_t \sin \theta_t. \quad (58)$$

我们现在有了推导传输方向所需的所有部件。方向向量只能是 \mathbf{n} 和 \mathbf{n}_S 的线性组合。它必须满足Snell这足以获得表达式

$$\mathbf{n}(\mathbf{x}, t) = \frac{n_t}{n_i} \mathbf{n} + \Gamma(\mathbf{x}, t) \mathbf{n}_S(\mathbf{x}, t) \quad (59)$$

其中函数 Γ 定义为

$$\Gamma(\mathbf{x}, t) \equiv \frac{n_i}{n_t} \mathbf{n} \cdot \mathbf{n}_S(\mathbf{x}, t) \pm \frac{n_t}{n_i} \sqrt{1 - (\mathbf{n} \cdot \mathbf{n}_S)^2} \quad (60)$$

当 $\mathbf{n} \cdot \mathbf{n}_S < 0$ 时， Γ 中使用加号，当 $\mathbf{n} \cdot \mathbf{n}_S > 0$ 时，使用减号。

6.2 菲涅耳反射率和透射率

伴随着通过界面的反射和透射过程的是描述其效率的一对系数。反射率 R 和透射率 T 通过在界面处没有光损失的约束而相关这就导致了

$$R + T = 1. \quad (61)$$

R 和 T 的表达式推导是基于电介质的电磁理论。我们将不进行推导，而只是写出解

$$R(\mathbf{n}, \mathbf{n}_S) = \frac{1}{2} \left(\frac{\sin^2(\theta_t - \theta_i)}{\sin^2(\theta_t + \theta_i)} + \frac{\tan^2(\theta_t - \theta_i)}{\tan^2(\theta_t + \theta_i)} \right) \quad (62)$$

$$(55)$$

请注意，此表达式对于曲面任一侧的入射光线方向均有效

6. 表面波光学
图4-14 显示了入射光的反射率的曲线图。
作为到表面的入射角的函数。该图从0度的掠射角延伸到90度的垂直入射应当清楚的是，图像上反射率的变化是水的“纹理”或感觉的重要来源。请注意，反射率是相对于波法线的入射角的函数，其进而与表面的斜率直接相关。因此，我们可以预期，一个强大的贡献者的纹理的水面是模式的斜率，而波高的变化主要作为一个波隐藏机制。 这就是为什么

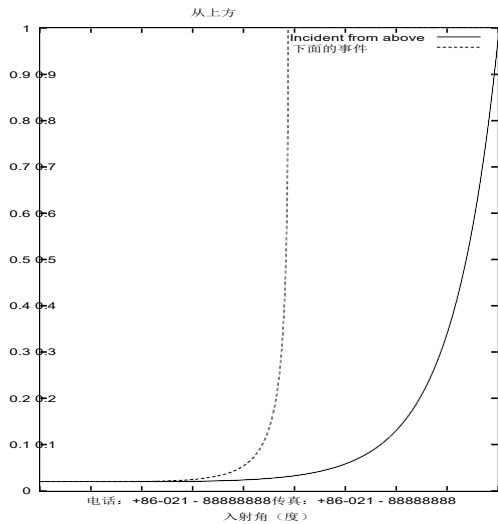


图24：从空气向下到水面的光的反射率，作为光的入射角的函数。

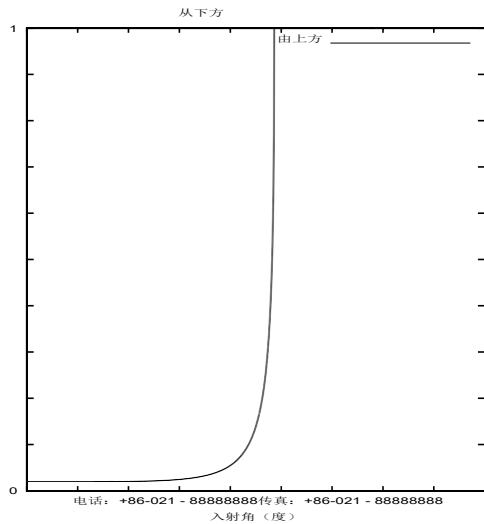


图25：来自水面下方的光的反射率，作为光的入射角的函数。

表面坡度比波高更接近于呈现的水，正如我们在前面讨论图10时看到的。

当入射光线来自水面以下时，反射率和透射率存在重要差异图25再次示出了作为入射角的函数的反射率在这种情况下，反射率在接近41度的相当大的角度处达到1。在低于该入射角的入射角下，反射率为1，因此没有光透射通过界面。这种现象是 **完全的内反射**，只要在游泳池里游来游去就能看到。全内反射开始的角称为

布鲁斯特

$$\sin \theta_B = \frac{n_2}{n_1} \quad (6.3)$$

或 $\theta_B = 48.6^\circ$ 。在我们的图中，这个角度是 $90 - \theta_B = 41.1^\circ$

6.3 为Renderman

从到目前为止的讨论中，渲染必须模仿的最重要的特征之一是表面的纹理，这是由于反射率和透射率的强烈斜率依赖性在本节中，我们将仅使用这些功能构建一个简单的Renderman兼容着色器有着色器经验的读者将知道如何立即扩展这一个。

着色器利用Renderman界面已经提供了一个内置的菲涅耳量计算器的事实，该计算器使用表面法线、入射方向向量和折射率提供R、T、n和n空气到水情况的着色器如下所示：

```
曲面水彩着色器 (
    颜色上升流= color (0, 0.2, 0.3); 颜色天空
    = color (0.69, 0.84, 1); color air =
    color (0.1, 0.1, 0.1); int n = 1.34;
    float Kdiffuse = 0.91;
    string envmap = "";
```

```

    )
{
    浮动反射率vector nI =
    normalize (I); vector nN
    = normalize (Ng);
    float costhetai = abs (nI. nN);
    float thetai = acos
    (costhetai);
    float sinthetat =sin (thetait) /nSnell;
    int n = int n (int n); if (thetait ==
    0.0)
    {
        反射率= (nSnell - 1) / (nSnell + 1);反射率=反
        射率 * 反射率;
    }
    其他
    {
        float fs = sin (thetat-thetait)
            / sin (thetat +
            thetai); float ts = tan
            (thetat-thetait)
            / tan (thetat + thetai);反射率= 0.5
            * (fs*fs + ts*ts);
    }
    向量dPE = P-E;
    float dist = length (dPE) * Kdiffuse;
    dist = exp (-dist);

    如果 (envmap!= "")
    {
        public int findDuplicate ();
    }
    Ci = dist * (反射率 * 天空
                + (1-反射率) * 上升流)
        + (1-dist) * 空气;
}
```

颜色有两种贡献：向下照射到曲面上的灯光这第二项将在下一节中讨论这对于天空中的高入射角是重要的，因为反射

6表面波光学
率低而透射率高。

3-23

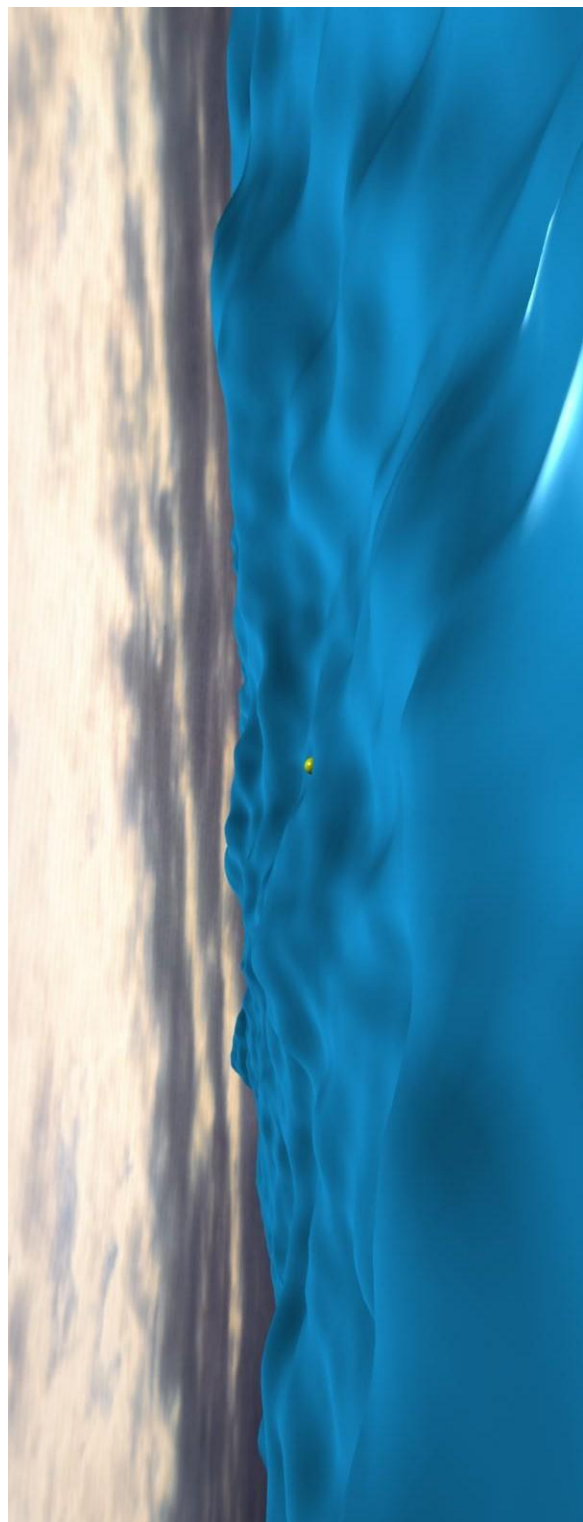


图26：使用通用塑料表面着色器模拟水面。用BMRT渲染。

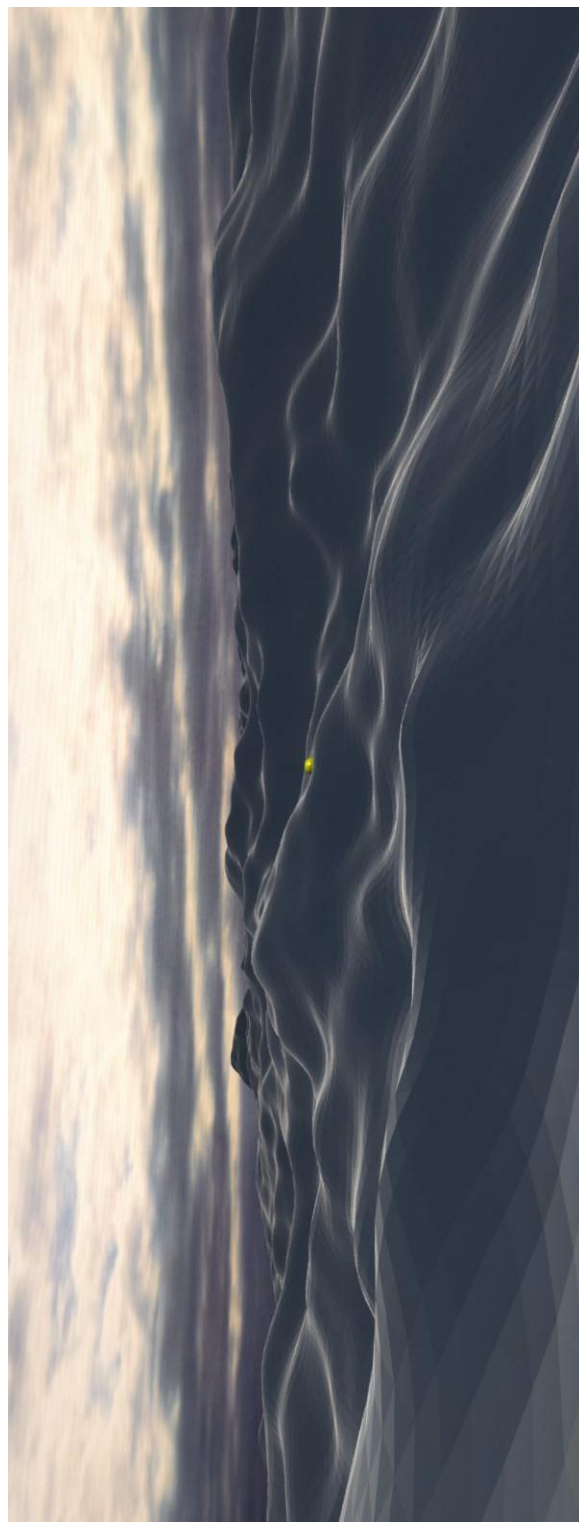


图27：使用真实表面着色器模拟水面。用BMRT渲染。

此着色器用于使用BMRT光线跟踪渲染器渲染图27中的图像。作为参考，完全相同的图像已在26中使用通用塑料着色器渲染。请注意，逼真的水着色器倾向于突出显示波浪的顶部，其中入射角接近90度掠射且反射率高，而波浪的侧面是暗的，其中入射角接近0，反射率低。

7 水量效应

前一节专门讨论了海洋表面的光学行为。在这一节中，我们集中在表面以下的水体积的光学行为。我们首先讨论了水体积对光的主要光学效应，然后介绍了研究人员建立的颜色模型，试图将任何一天的海洋颜色与生物和物理过程联系起来。这些模型是建立在多年的现场测量船舶和同行。最后，我们讨论了两个重要的效果，焦散和阳光，有时很难把握，并产生美丽的图像时，适当的模拟。

7.1 水体的散射、透射和反射

在开阔的海洋中，光被水的体积散射和吸收。这些事件的来源有三种类型：水分子、活的和死的有机物以及非有机物。在世界上大多数远离海岸线的海洋然而，散射主要由有机物控制

为了模拟体积吸收和散射的过程，有五个感兴趣的量：吸收系数、散射系数、消光系数、漫消光系数和体反射率。所有这些系数具有长度倒数的单位，并且表示光随着通过介质的距离的衰减的指数率。吸收系数 a 是光吸收率随距离的变化，散射系数 b 是散射率随长度的变化，消光系数 c 是前两个系数的和 $c=a+b$ ，并且漫射消光系数 K 描述了在考虑吸收和散射过程之后光强度随距离的损失率。 K 和其他参数之间的联系尚未完全理解，部分原因是存在多种根据操作测量来定义 K 的方法不同的方式改变了依赖关系的细节。然而，在非常深的水中存在一个称为渐近极限的条件，在该条件下， K 的所有操作定义都收敛到一个值。 K 的渐近值已经以各种方式建模。有一个数学上精确的结果，比率 K/c 只取决于 b/c ，单次散射反照率，和一些细节的角分布的个别散射分布。图28是用于合理水条件的 K/c 模型的示例。模型已经被生成为 K 的颜色依赖性，最值得注意的是由Jerlov。1990年，Austin和Petzold对光谱模型进行了修订分析，包括新数据，以产生 K 作为颜色函数的精细模型。对于海洋中典型的可见光条件， K 的值范围为0.03/米至0.1/米。一般来说， $K \ll c$ 。

解释这些量以模拟水体效应的一种方法如下：

1. 一束阳光以强度 I （在菲涅耳透射损失一些强度之后）进入水中。沿着长度为 s 的水下路径，路径末端的强度为

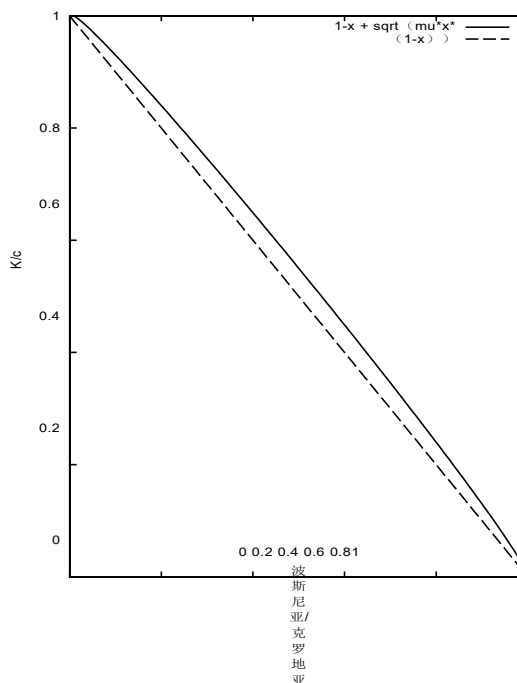


图28：漫消光系数对单次散射反照率的依赖性，归一化为消光系数。

$I \exp(-cs)$ ，即直射阳光的光线尽可能快地衰减。

2. 沿着穿过水的路径，射线的一部分被散射成方向分布。每单位长度射线的散射强度为 b ，因此强度与 $b \exp(-cs)$ 成正比。
3. 从射线散射出的光可能经历更多的散射事件。几乎不可能追踪到他们所有人。然而，该过程的总体结果是将沿着从原始路径到相机的路径的射线衰减为 $b \exp(-cs) \exp(Ks_c)$ ，其中 s_c 是从海洋中的散射点到相机的距离。

用于模拟的第五个感兴趣的量是水体积的体反射率。这是一个旨在允许我们忽略正在发生的细节的量，将体积视为朗伯反射器，并计算体反射率的值该数字对许多因素敏感，包括波面条件、太阳角、水光学特性和角展度的细节。然而，反射率值约为0.04，似乎同意与数据。

7.2 水下POV：折射天光、焦散和日光

现在我们手头有了水下光学特性，我们可以看看海洋中的两个重要现象：焦散和日光。

7.2.1 焦散

在这种情况下，焦散是在水下表面上形成的光图案。因为水面不平坦，成群的

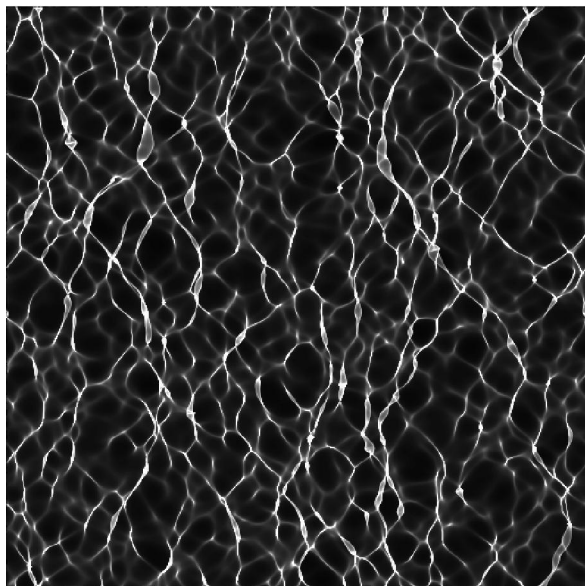


图29：在表面以下的浅深度（5米）处呈现焦散图案

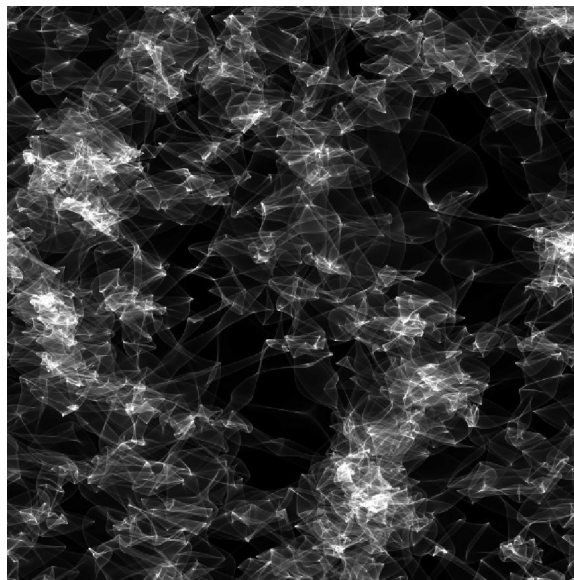


图30：在表面以下的大深度（100米）处呈现焦散图案

入射到表面上的光线要么聚焦要么散焦。因此，在海洋表面以下一定深度的虚拟平面上的点接收来自表面上的光的强度由于深度、原始对比度和其他因素而变化。现在，让我们写下强度

$I = Ref I_0$ ，其中 I_0 为刚好在水面上方量 Ref 是由于波的聚焦和散焦而随虚拟平面上的位置而变化的比例因子，并且被称为**焦散图案**。图29示出焦散图案 Ref 的示例。注意腐蚀性部分-

燕鸥呈现丝状和环状结构。在非常深的深度处，焦散图案甚至更加引人注目，如图30所示。

产生焦散图案的水下光的重要性质之一这实际上是一个简单的想法：假设海洋表面的一小块区域有阳光穿过它进入水中，表面强度为 I 。当我们将该区域映射到更大的深度时，其中的区域量会增加或缩小，但最有可能的增长取决于该区域是聚焦还是散焦。在水中的任何深度处的强度与投影区域的面积的倒数成比例。另一种说法是，如果一束光线发散，它们的强度就会降低，以保持强度时间面积的乘积固定。

模拟的焦散图案实际上可以与真实世界的数据进行（粗略）比较。在20世纪70年代、80年代和90年代发表的一系列论文中，Dera和其他人收集了光的高强度时间序列[21]。作为该数据收集和分析项目的一部分，数据用于生成光强度的概率分布函数（PDF）。图31显示了取自Dera的一篇论文的两个PDF。针对不同的表面粗糙度条件收集两个PDF：较粗糙的水倾向于抑制强度的更多高幅度波动。

图32示出了来自海洋表面的模拟的两个深度处的pdf。由于许多因素，这两组数据与Dera的测量结果不匹配，但最重要的是因为我们没有模拟Dera实验中的环境条件和仪器。尽管如此，图32与图33的相似之处在于：

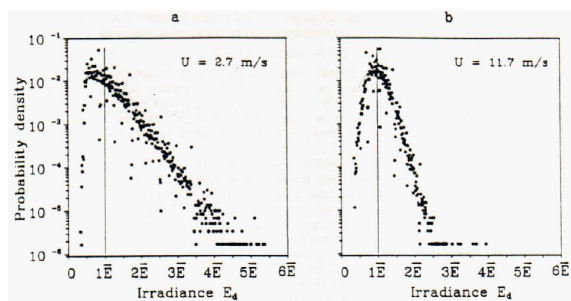


图31：参考文献[21]中由Dera测量的PDF。

德拉

7.2.2 戈德雷

水下阳光，也称为godrays，有一个非常相似的起源焦散。直射阳光进入水体，在表面的不同点聚焦和散焦当光线向下穿过体积时，一些光在其他方向上散射，并且一小部分到达相机。对于相机明显的散射光的累积图案是godray。因此，虽然焦散是直射太阳光的图案，其向下穿透到水体积的底部，但太阳光束是来自水体积中的直射太阳光的那些轴的散射光。图33展示了由向上看的相机看到的太阳光束。

引用

[1] 何士泰，

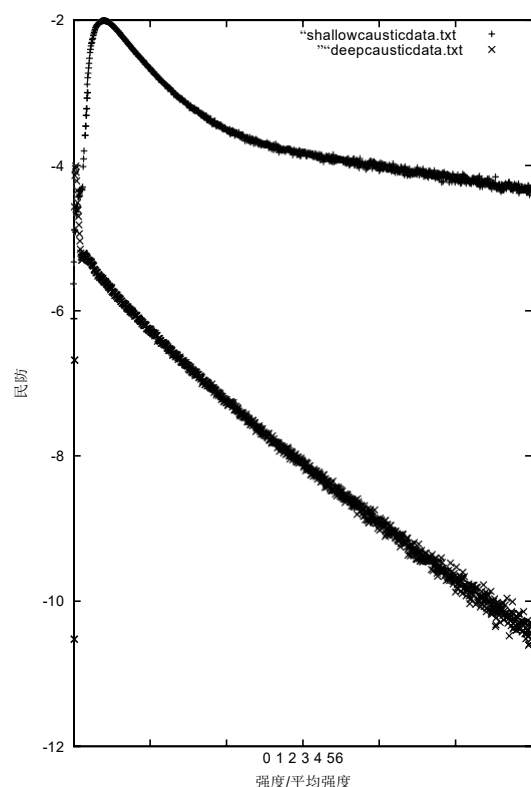


图32：焦散线中光强波动的计算概率密度函数。（上部曲线）浅深度为2米；（下曲线）水深10米。



图33：太阳光束或godrays的渲染，如在光源处直视所见。

- [2] 林志荣,
- [3] 学士 Cheeseman 和 C.P. R Hoppel , <http://www.asc2002.com/summaries/h/HP-08.pdf>
- [4] Simon Premoze, 3.第三章。
- [5] Jeff Odien, 《On the Waterfront》, Cinefex, No. 64, 第96页, (1995年)
- [6] Ted Elrick, “Elemental Images”, Cinefex, No. 70, 第114页, (1997年)
- [7] 凯文·H Martin, “Close Contact”, Cinefex, No. 71, 第114页, (1997年)
- [8] Don Shay, 72, 第82页, (1997年)
- [9] 凯文·H马丁, “病毒: 建设一个更好的博客”, Cinefex, 不. 76, 第55页, (1999年)
- [10] SimonPremož e和MichaelAshikhmin,
- [11] 加里·A Mastin, Peter A. Watterger和John F. Mareda,
- [12] Alain Fournier和William T. Reeves, “海浪的简单模型”, 计算机图形学, 卷. 20, No. 4, 1986, 第75-84页。
- [13] Darwyn Peachey,
- 图形, 卷. 20, 不. 4, 1986, 第65-74页。
- [14] Blair Kinsman , *Wind Waves , Their Generation and Propagation on the Ocean Surface*, Dover Publications, 1984.
- [15] 美国海洋工程课程开发 在 海洋 技术 号 8、 Elsevier Science Publishers B.V. 1992. 另 见 <http://www.dnv.no/ocean/bk/grand.htm>
- [16] 丹尼斯湾Creamer, Frank Henyey, Roy Schult, and Jon Wright, “Improved Linear Representation of Ocean Surface Waves.”J. Fluid Mech, **205**, pp. 135-161, (1989) 中所述。
- [17] 塞伯特角Duntley, 选购配件志是啊**53**, 2, 第214-233页, 1963。
- [18] 柯蒂斯·D莫布里, *光与水: 自然水体中的辐射传输*, 学术出版社, 1994年。
- [19] *平面平行大气层和海洋中的多重散射研究论文集: 方法*. 乔治W. Kattawar, SPIE里程碑系列, **MS 42**, SPIE Opt.工程师按下一九九一年
- [20] R.W. Austin 和 T. 海 洋 光 学 X, Richard W. Petzold , Spinrad, 编, SPIE**1302**, 1990年。
- [21] Jerzy Dera、Slawomir Sagan、Dariusz Stramski, “海面波对阳光的聚焦: 来自黑海的新测量结果, ”*海洋光学XI, SPIE会议录*, 1992年。

- [22] Jerry Tessendorf, Andrew Kirmse, Charles River Media, (2004) .
- [23] Stéphane T. Grilli主页
<http://131.128.105.170/grilli/>.
- [24] AROSS - 机载 远程 光学 聚光灯系统,
<http://www.aross.arete-dc.com>

8 附录：交互式水面的示例代码

本附录中列出的代码是iWave算法的工作实现如下所列，*iwave*油漆看起来像一个粗糙的油漆例程。用户可以在两种模式下进行绘制。当*iwave paint*启动时，它开始在一个小网格（200x200，列出了设置）上运行iWave模拟。这个网格足够小，iWave仿真可以在大约1 GHz或更快的CPU上交互运行。由于启动时没有干扰波，因此模拟运行不明显界面如图34所示。在启动时的默认模式下，绘制会生成以黑色显示并阻挡水波的障碍物。图35示出了已经被涂漆的示例性障碍物。按's绘制扰动时，*iwave*绘制会传播扰动，并从可能已绘制的任何障碍物反射。图36示出了在源已经被涂在障碍物内部并且允许短暂的时间段在障碍物内部传播并离开之后的帧，从而在障碍物的口部处产生衍射图案。

一些有用的键盘选项：

o将绘制模式设置为障碍物绘制。这可以在任何时间选择。

s将绘制模式置于源绘制中。这可以在任何时候选择。

```

/打开/关闭_-----
//
//iwave_paint
//
//演示了生成和交互
//通过允许用户在对象周围波动
//绘制障碍物和源，并观看iwave
//propagation.
//
//作者: Jerry Tessendorf
jerry@finelightvisualtechnology.com
//2003
//
//本软件属于公有领域。
//
/打开/关闭_-----

/打开/关闭_-----
//
//使用方法:
//
// iwave_paint是一个交互式画图程序
//其中用户在水面上绘画，并且
//波的演化和与障碍物的
//在水里。
//
//有两种绘制模式。 键入
//在障碍物绘画模式下的程序。当你
//按住鼠标左键并画图，你
//会看到一个黑色的障碍物。这
//障碍物可以是任何形状。
//
//输入
//mode.php 现在用鼠标左键绘画
// down在水面
//返回 它所产生的波浪会随着你的
//继续画。 波浪从任何
//已绘制或已被绘制
//后来画的。
//
//键入
//
//输入
//在waves中。
//
//按空格键启动和停止wave
// evolution.当进化停止时，你
//可以继续绘制障碍物。
//
//
//这段代码是在Linux下编写和运行的。的
// compile命令是
//
// g++ iwave_paint. C-02 -o iwave_paint -lglut-lGL
//
//
/打开/关闭_-----

```

#包含cmath>

图34：启动时的*iwave*绘制图35：绘制了障碍物的*iwave*绘制图36：在障碍物内绘制源并使其传播出去后的*iwave*绘制窗口

```
#ifndef __APPLE__ #include
GLUT/glut.h>
#else
#include GL/gl.h> // OpenGL本身。#include
GL/glu.h> // GLUT支持库。#include GL/glut.h> //
GLUT支持库。
#endif

#include iostream>

using namespace std;

int i_width, i_height,
size; float *display_map;
浮动* 障碍物; float
*source;

float *height;
float *previous_height
float *vertical_derivative;

float_factor; [13] [13];

int paint_mode;
{ PAINT_OBSTRUCTION, PAINT_SOURCE};

bool_data;
bool toggle_animation_on_off

float dt, alpha, 重力;

floatobjectives_brush[3] [3]; [3]
[3];

int xmouse_prev, ymouse_prev;

//打开/关闭-----
//
//初始化例程
//
//将所有字段初始化为零
void Initialize (float *data, int size, float value)
{
    int i=0; i< size;i++) { data[i] = value;}
}

//计算卷积核的元素void InitializeKernel ()
{
    double dk = 0.01;双
    西格玛= 1.0;双范数= 0;

    for (double k=0;k 10;k+=dk)
    {
        norm += k*k*exp (-sigma*k*k) ;
    }

    int i= 0; i< =6;i++)
    {
        int j=-6; j< =6;j++)
        {
            double r = sqrt ((float) (i*i +
                j*j)) ; double kern = 0;
            for (double k=0;k 10;k+=dk)
            {
                kern += k*k*exp (-sigma*k*k) *j0 (r*k) ;
            }
            kernel[i+6] [j+6] = kern /norm;
        }
    }
}

void InitializeBrushes ()
{
    obstruction_brush[1] [1] = 0.0;
    obstruction_brush[1] [0] = 0.5;
    obstruction_brush[0] [1] = 0.5;
    obstruction_brush[2] [1] = 0.5;
    obstruction_brush[1] [2] = 0.5;
    obstruction_brush[0] [2] = 0.75;
    obstruction_brush[2] [0] = 0.75;
    obstruction_brush[0] [0] = 0.75;
    obstruction_brush[2] [2] = 0.75;

    int [1] [1]= 0;
    [1] [0] = 0.5;
    source_brush[0] [1] = 0.5;
    source_brush[2] [1] = 0.5;
    source_brush[1] [2] = 0.5;
    [0] [2] = 0.25;
    source_brush[2] [0] = 0.25;
    [0] [0] = 0.25;
    source_brush[2] [2] = 0.25;
}

void ClearObstruction ()
{
    int i=0; i< size;i++) {objectivity [i] = 1.0;}
}

void ClearWaves ()
{
    int i=0; i<大小;i++)
    {

```



```

        int count = 0.0;
        previous_height[i] = 0;
        vertical_derivative[i] = 0.0;
    }
}

//-----

void ConvertToDisplay ()
{
    int i=0; i<大小;i++
    {
        display_map[i] = 0.5* (height[i]/scaling_factor + 1.0) * objective [i];
    }
}

//-----
//
// 这两个例程
//
// ComputeVerticalDerivative ()
// Propagate ()
//
// 是iWave算法的核心。
//
// 在Propagate () 中, 我们没有费心处理
// 边界条件 这使得最外层
// 6个像素就像一堵硬墙。
//

void ComputeVerticalDerivative ()
{
    // 第一步: (int x =6;ix<
    //          iwidth-6;ix++) 的内部
    {
        for (int iy=6;iy iheight-6;iy++)
        {
            int index = ix + iwidth*iy;
            float vd = 0;
            for (int x =-6; x =6; x ++ )
            {
                对于 (int iiy=-6;iiy =6;iiy++)
                {
                    int i = ix+iix + iwidth* (iy+iiy);
                    vd += kernel[iix+6][iiy+6] * height[iindex];
                }
            }
            vertical_derivative[index] = vd;
        }
    }
}

void Propagate ()
{
    // 应用阻塞
    int i=0; i< size;i++) { height[i] *= obstruction[i];}

    // 计算垂直导数
    ComputeVerticalDerivative ();

    // 前进表面浮动adt =
    // alpha*dt;
    float adt2 =1.0/ (1.0+adt);
    int i=0; i<大小;i++
    {
        int [i];
        height[i] =height[i]* (2.0+adt) -previous_height[i]-gravity*vertical_derivative[i];
        int [i] * i;
        int [i]= int n; height[i]
        *= objective [i];
        previous_height[i] = temp;
        // reset source each
        step source[i] = 0;
        {\fnSimHei\bordl\shadl\pos (200, 288) }
    }
}

//-----
//
// 绘制和显示代码
//

void resetScaleFactor (float amount)
{
    scaling_factor *= amount;
}

int x, int y, int n, int n
{
    int x = x -1; int
    y = y -1;
    if (xstart 0) { xstart = 0;} if
    (ystart 0) { ystart = 0;}

    int x = x +1;
    int y = y +1;
    if (xend >= iwidth) { xend = iwidth-1;} if
    (yend >= iheight) { yend = iheight-1;}

    if (paint_mode == PAINT_OBSTRUCTION)
    {
        int x = x; ix< = xend; ix++)
        {
            int i = int n; iy< =yend; iy++)
        {
            int index = ix + iwidth* (iheight-iy-1); objective
            [index] *= objective_brush[ix-xstart][iy-ystart];
        }
    }
}

}
else if (paint_mode == PAINT_SOURCE)
{
    int x = x; ix< = xend; ix++)
    {
        int i = int n; iy< =yend; iy++)
    {
        int index = ix + iwidth* (iheight-iy-1);
        source[index] += source_brush[ix-xstart][iy-ystart];
    }
}
}
return;
}

// 打开/关闭 -----
//
// GL和GLUT回调
//
// -----

void cbDisplay (void)
{
    glClear (GL_COLOR_BUFFER_BIT);
    glDrawPixels (i width, i height, GL_LUMINANCE, GL_FLOAT, display_map);
    glutSwapBuffers ();
}

// 动画并显示新结果 void cbIdle ()
{
    if (toggle_animation_on_off) { Propagate ();}
    ConvertToDisplay ();
    int findDuplicate ()
}

void cbOnKeyboard (unsigned char key, int x, int y)
{
    开关 (钥匙)
    {
        案例 '-': 案例 '_':
        resetScaleFactor (1.0/0.9);
        regenerate_data = true; 折断;

        case '+': case '=':
        resetScaleFactor (0.9);
        regenerate_data = true;
        折断;

        案例
        toggle_animation_on_off != toggle_animation_on_off;

        情况
        paint_mode =PAINT_OBSTRUCTION;
        折断;

        案例 ' '
        paint_mode =PAINT_SOURCE;
        折断;

        case
        Initialize (source, size, 0.0);
        break;

        默认值:
        break;
    }
}

void cbMouseDown (int button, int state, int x, int y)
{
    if (button!= GLUT_LEFT_BUTTON) { return;}
    if (state!= return;){ xmouse_prev = x;
    return y; int x, y =
    0;
}

int findDuplicate (int x, int y)
{
}
}

```

附录8: 交互水面的样本代码

3-26

```
xmou
se_p
rev
= x;
retu
rn y;
int
x, y
= 0;
}

//-----

int main (int参数, 字符 ** 参数)
{
    //初始化几个变
    量iwidth =
    iheight =
    200;
    size = i * i;

    dt = 0.03;
    alpha = 0.3;
    重力= 9.8 * dt * dt;

    scaling_factor
    = 1.0;
    toggle_Animati
    on_on_off =
    true;
}
```

```
//为字段分配空间并初始化它们height = new
float[size]; previous_height = new
float[size]; return new float[size]; return new
float[size];
return new float[size];
return new float[size];

System. out. println ();
System. out. println ();

initializeBrushes ();

//构建卷积核InitializeKernel ();

// GLUT例程glutInit
(argc, argv);

glutInitDisplayMode (GLUT_RGBA|double); int n
(int n, int n);

//打开一个窗口
char title[] ="iWave Demo";
int Window_ID = glutCreateWindow (title);

sort (sort);

glutDisplayFunc (&cbDisplay); glutIdleFunc
(&cbIdle); glutKeyboardFunc (&cbOnKeyboard);
glutMouseFunc (&cbMouseDown);
int&findDuplicate ();

return 1;

};
```