一个实时水彩画渲染的系统

计算机图形国际会议论文集 · CGI--7月 2005

DOI: 10.1109/CGI.2005.1500426 - 资料来源 · IEEE Xplore

一个实时水彩画渲染的系统

杰里米-伯吉斯 "计算机图形和视觉
研究组 计算机科学系
奥塔哥大学

Geoff Wyvillt
计算机图形和视觉研究小组
奥塔哥大学计算机科学系

Scott A. King:I: 计算机与数学科学
德克萨斯A&M-Corpus Christi

ABSTRACT

我们提出了一个实时渲染场景的系统，以产生类似水彩画的图像。我们的系统使用对象空间渲染技术和图像平面后期处理的混合技术，以达到非常好的效果。我们通过图形处理单元（GPU）编程、硬件加速的阴影量和使用
在各种不同的任务中，都会有噪音。我们设法通过结合以前的图像空间和物体空间的方法来规避这两者中发现的问题。

关键字。 NPR，实时，油漆，水彩画。

!.简介

非逼真渲染（NPR）是指可用于制作与真实世界不完全相似的场景的技术系列。标准的、现实的图形技术不能产生许多NPR领域所需的图像特征。
我们感兴趣的领域是绘画式渲染--
以绘画的风格渲染场景。绘画的一些特征在标准的渲染方法中是缺失的。

- 某些绘画风格中固有的素描性。笔触的存在。
  由画布上的颜料而非现实世界中的光线所引起的复杂的色彩互动。

试图对这些特征进行建模的NPR方法通常在完成的图像上工作[1][2][31也就是说，系统将被提供一个渲染的场景或照片，然后它将处理它，使其类似于一幅画。我们的方法在物体空间中做了很多工作。

在图像平面内进行绘画式渲染有两个关键问题，我们已经通过对象空间处理过了技巧。首先，它通常非常缓慢。这种处理的速度不是图像平面技术的必要特征。这种方法通常很慢，因为每个像素都需要大量的处理。第二个问题是，后处理步骤通常只有单帧包含的信息。正因为如此，产生的壮举，如刷子
描写，都是不连贯的。

我们提出了一个混合系统，用于渲染水彩画风格的图像。我们的系统在硬件中工作，并在对象空间中做大部分的"艰苦工作"，因此速度非常快。对象空间技术几乎自动产生帧与帧之间的一致性。

我们的系统首先通过对象空间方法产生关于场景中每个像素的各种信息，然后对这些信息进行后处理步骤，产生最终的画面。在这一节中，我们2研究了真正的水彩画的特点，考察了以前在绘画式渲染方面的工作，并讨论了在GPU上工作的限制和好处。在第3节中，我们研究了如何使用视频游戏中常见的技术快速产生阴影，在第4节中，我们从头到尾研究了我们的绘画层算法。在第5节中，我们研究了我们的结果，并在第5节中简要讨论6了我们到目前为止的成功，并着眼于未来的工作。

2. 背景材料

2.1. 真正的水彩画

为了明确我们所要达到的目的，我们现在介绍一些真正的水彩画技巧（图1）。

基本的水洗。水洗是一种单一的颜料层。一个重要的变化是渐进式水洗，其中一种颜料随着水洗的进行而变薄。
创建一个梯度。
上釉。这是用大量非常薄的水洗画的过程，以产生颜色 "发光 "的效果。
湿中带湿。Tilis是指我们在潮湿的表面上作画。湿中有湿的画法会使颜料在某种程度上无法预测地扩散。
阴性绘画。在传统的水彩画中，从不使用白色。如果需要白色，则在纸上留出空白，因此称为阴性画。

*
电子邮件：jburgess@cs.otago.ac.nz
t 电子邮件：geoff@cs.otago.ac.nz
*我的*电子邮件：sking@sci.tamucc.edu

国际计算机图形学会议（2005CGI'05）论文集
6月22-24日，美国纽约州石溪市2005,。
0-7803-9330,9/05/$20.00 ©2005 IEE

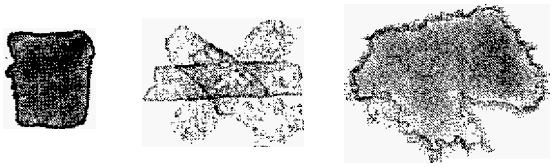


图1：用实际的异色颜料制作的三块色板。左边是基本的水洗，中间是上光，右边是湿润的In wel 参见色板。

2.2. 以前的工作

画家的渲染主要关注的是图像平面。我们可以把这归因于真实绘画的基本"平面"性质--所有的绘画都必须在画布或某种页面上创作。许多绘画效果，如笔触和流体模拟，更容易在图像中模拟和创造。比起在物体空间中，在平面上更有优势。

Curtis等人[I]设计了一种非常成功的产生水彩画图像的算法。使用一个基于物理的colout模型，以及复杂的模拟的该系统产生的2.1.图像确实类似于某些风格的水彩画，但它可以创造出任何在本节中讨论的现实世界中的水彩画效果。

然而，Curtis等人的系统非常慢，而且只是半自动的--它需要用户的协助才能达到很好的效果。此外，Curtis等人对如何将他们的系统扩展到动画中*提供*了非常稀少的讨论。

Barbara Meier提出的一个系统非常不同，因为它根本不涉及创建水彩画场景，而是通过将笔触移动到物体空间来狡猾地解决动画中的一致性问题[4]。笔触是通过使用随机分布的粒子在物体上播种的。在Meier的系统中，笔触总是按照预期的方式进行，而不会出现不可预知的移动。

我们自己的系统从Eric Lum和Kwan-Liu Ma创建的另一个系统中获得了很多灵感，该系统吸收了Curtis等人和Meier{6}的一些优秀工作。他们的系统在物体空间中提供类似画笔的纹理，并使用Curtis等人的论文中发现的颜色模型合成多层油漆。我们的方法与Lum和Ma提出的方法不同，我们简化了笔触生成，并增加了一个后处理图像平面的步骤。我们简化笔触生成的结果与水彩画中的笔触更接近。我们的方法产生了与真正的水彩画更相似的效果，并保持了显示物体轮廓的特性。我们的后处理步骤使我们能够获得真实水彩画中的特定效果，如边缘变暗和物体边缘不直。Lum和Ma使用光线追踪将图像绘制到屏幕上，而我们使用光栅化、OpenGL API和可编程的图形硬件。

2.3. GPU编程。

我们尽可能地在调制解调器显卡的GPU上实现我们的系统。为了使一些术语清晰明了，我们在此简要介绍一下我们能做什么。

基本上，有几种类型2的GPU程序，称为着色器。第一个是顶点着色器，它对多边形模型的顶点进行操作；第二个是片段着色器。着色器，它对栅格化的像素进行操作（图2）。顶点着色器允许我们改变位置、非线性和纹理：每个顶点的坐标数据，并且可以向下面的片段着色器输出有限的信息。片段着色器只允许我们修改深度和颜色，但是在栅格化的像素中存储非颜色信息是很常见的[10]。

此外，在一些用于GPU的高级着色语言中，图形相关的功能已经被简化，如Cg、HLSL和Sh [I1]。特别是向量和矩阵被视为第一类类型，这使得图形中的许多操作更容易处理。

在GPU上工作的主要好处是现代图形卡提供的性能。

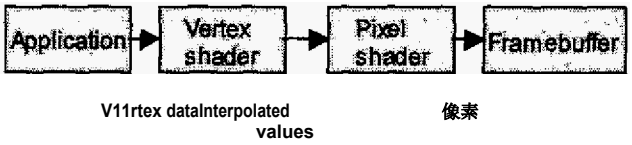


图2：Cg图形帮助线

3. 硬件上容忍的阴影。

正如Lum和Ma所展示的，使用光线追踪的最大好处之一是，阴影是作为基本方法的直接扩展提供的。当使用基于栅格化多边形对象的API时，情况就不是这样了。而产生阴影是比较困难的。有两个常用的影子生成方法，这两种方法都可以在硬件中得到加速--影子映射[12]和影子卷[5]。我们决定，对于我们的目的来说，影子卷是更可取的。尽管阴影贴图提供了柔和的阴影，但当贴图的分辨率不足时，或者当光线和投射阴影的表面之间的角度太低时，它们也会产生伪影。此外，阴影体的概念简单和通用性也吸引了我们。

影子卷使用的原则是，每一个区域都是被某个物体遮挡的像素属于空间中没有光投射的体积。创建这些体积是相当容易的，但我们如何利用光栅管道来弄清哪些像素在某个特定的阴影中，哪些不在？

要为一个给定的光源创建阴影体积，我们所做的就是把每个物体的剪影边缘从光源挤压到无限远，并在每个末端用一个多边形盖住。因为我们必须在使用这些阴影多边形的过程中对其进行光栅化处理，所以在顶点着色器中即时处理比在软件中计算剪影边缘更容易。顶点着色器会检查一个顶点是否相对于光线的"正面"，然后做两件事中的一件。如果顶点朝向前方，它就留在原地。如果不是，它就被挤压到无限大。在《Cg手册》[1OJ]中可以找到一个这样做的着色器样本。

在阴影量过程结束时，我们知道哪些像素在阴影中，哪些不在。我们的实现只提供了一个光源。我们可以通过在纹理中存储每个光源的模版缓冲区来调整我们的系统以使用多个光源。但这并不太有用，因为绝大多数的水彩画只包括一个光源。

请注意，通常在阴影体积渲染中，我们对每个光源进行一次处理，并在没有阴影的地方加上该光源的贡献。因为我们需要知道阴影的位置（第4节），所以我们要用相反的方法。

关于影子卷的更详细讨论，见Lengyel[5]。

4. 我们的方法

我们的方法是基于这样的想法：真正的水彩画家很少在某一点上准确地画出他或她想要的颜色。而不是涂抹几层来获得所需的效果。我们的系统是这个想法的一个非常简化的版本，对任何给定的点只包括三层油漆。

首先，我们有一个由一组多边形模型和一个场景描述决定的场景。每个物体都必须有一个颜料和一个变换矩阵，以便在空间中定位、缩放和旋转。用户需要做的就是在场景描述文件中指定这些特征中的每一个，剩下的就由系统来完成。理想情况下，这些对象是相当简单的。例如，第5节中的两个杯子场景中的杯子由10000个面组成，该场景可以每秒渲染25次。增加场景的多边形数量会导致性能下降，但除此之外并不影响系统的运行。为了渲染场景，我们的系统包括四个关键步骤。

1. 确定阴影（见第3节）。
2. 确定涂料的厚度和颜色。
3. 进行图像平面后处理。
4. 复合材料层。

4.1 涂料层计算

与Lum和Ma的系统一样，我们的系统有三层颜料：漫反射层、阴影层和纹理层[6]。每一层最好都能代表真实的水彩画中可能出现的不同类型的层。需要注意的是，这里的颜色是指颜料（见第4.3节），而不是RGB三要素。

对于一个给定的像素，漫反射层是分配给该物体的颜料的统一厚度（图3），它只是想代表该物体的实际颜色。例如，如果我们看的是像素（30，40），在这一点上要画一个粉红色的花瓶，那么漫反射层将是一个统一厚度的粉红色颜料。厚度是一个相对值，其中'1.0'是用于Kubeika-Munk模型的单层涂料的基本厚度（4.3节）。虽然一个单位厚度是方便的，但对于以少量水洗为基础的水彩画来说，它是不适当的深色。我们发现，使用0.5的基础厚度可以提供最令人满意的结果。

阴影层是第二层，由一个场景的统一颜料组成，厚度不一（图3）。凡是有完全阴影的地方，这一层的厚度都是0.1。在Phong照明方程产生一些黑暗，但不是完全阴影的地方，我们允许阴影厚度在0.1和00.2之间变化。 0.1.

最后一层是最有趣的，是"纹理层"。这一层应该是给人以水彩画中的纹理的错觉。在任何时候，它的颜料都与漫反射层相同，但其厚度可以自由变化。这一层的厚度可以通过以下程序找到。

I. 找到这个片段在图像平面上的取样点，x。

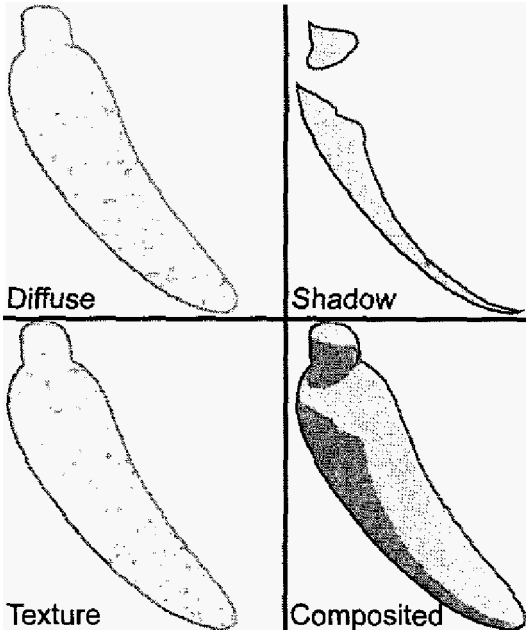


图3：香蕉上的三个颜料层，以及所有三个层的合成。注意笔触是如何跟随纹理层的表面的。另见色板。

2. 使用x对一个双分量的噪声纹理进行采样，并将结果加到x上--这就得到了一个"摇摆"的效果。在我们的实现中，我们使用Wyvill噪声[9]。

3. 找到这一点上表面法线在图像平面上的投影，并将其也加到x上。Titls将使采样的噪声跟随表面（图3）。

请注意，在Lum和Ma的系统中，使用了线积分卷积（LIC，见Interrante**[8]**）来获得物体周围笔触拉伸的效果。我们认为，这既过于昂贵，又能获得不现实的结果。LIC也显得过于人工化--笔触跟随表面的完美性在真实的绘画中没有类似的效果。

所有这些信息都在一个初步的着色器中计算出来。它输出的图像完美地保留了每个像素的所有这些信息，而不做任何处理。每个像素都有以下格式。

红色通道=规范化的物体标识符（id）。当场景被读入时，每个物体都被分配了一个唯一的ID。为了将一个物体ID转换为**1**和0之间的数值，我们用它除以物体的总数。

绿色通道=阴影厚度。蓝色通道=纹理厚度。Alpha通道=深度。

我们不需要漫反射的厚度，因为它反正是均匀的。

4.2.图像平面后处理

用我们的系统制作的原始图像是合理的，但是很明显，在考虑到特定的水彩画目标的情况下，增加一些后期处理会产生更好的结果。特别是有两个关键的特征可以为产生的图像带来很大的帮助。首先，在真正的水彩画中，边缘会变暗。这是因为颜料倾向于向画面的边缘移动。

第二点是，真正的水彩画并不完美，物体形状很少是精确的。

对于边缘效应，我们做的第一件事是对前一帧所有通道的*25个*疏散采样的像素进行高斯平均（图4）。然后，我们找出三个差异
表示下一步要做什么：平均和原始物体ID（i）、阴影厚度（s）和深度（d）之间的各自差异。请注意，物体ID之间的差异被异常地处理。这里的差异只是两个值的小数部分之间的差异。因此，如果原始对象ID是10.0，而平均ID是7.5，我们会发现和0.0之间的绝对差异 0.5.

如果深度低于平均深度，我们就加厚漫反射颜料，如果深度高于平均深度，我们就减薄漫反射颜料，但前提是差值的绝对值超过阈值。不管阈值是多少，如果物体ID有变化，我们会根据深度和物体ID的差异来增减漫反射颜料的比例。最后，我们根据影子平均数和原始影子像素之间的差异，按比例增厚影子层。

做这个边缘暗化和不做这个边缘暗化之间的差别是惊人的，而且大大增加了图像的效果（图5）。

所有的样本点都被X和Y的噪声值所偏移，这导致了物体的不完美外观，其特点是
真正的水彩画。

4.3合成

我们的系统使用Kubeika-Munk颜色模型[7]。在这个模型中，每种颜色实际上都是一种颜料，在所有相关的波长（红、绿、蓝）上都有光的散射和吸收
"值"。这个模型背后的想法是，真正的油漆中的特定颜料看起来是不同的，这取决于它的厚度和下面是什么颜色。
例如，如果一种颜色在白色上应该是非常强烈的，但在黑色上几乎是透明的，Kubeika-Munk颜色模型将准确地显示这一点[7]。

Kubeika-
Munk模型已被证明对油画非常准确，虽然它对水彩画的设计并不完美，但它已被证明是非常成功的。颜料可以被指定为两个RGB三元组；一个在白色上，一个在黑色上。使用成对的RGB三联体而不是所有波长的散射和吸收值来合成和计算相关值的公式可以在Curtis等人[l]中找到。

在我们的系统中，
所有的合成都是在一个像素着色器中进行的。
图中显示3了一个完全合成的对象。

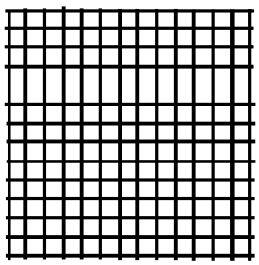


图4：取样网格。所有样本的位置都*被*噪声抵消了。中间的方块是要画的像素，其他灰色方块是其他样本。

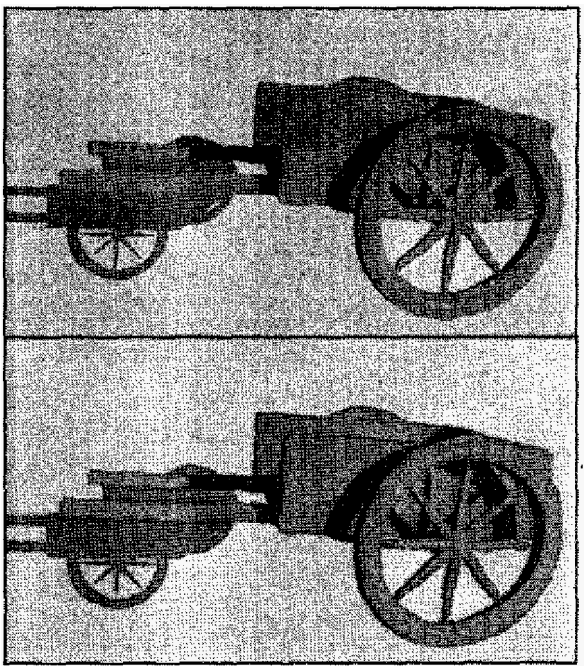


图5：一辆有（下）和无（上）边缘变暗的马车。

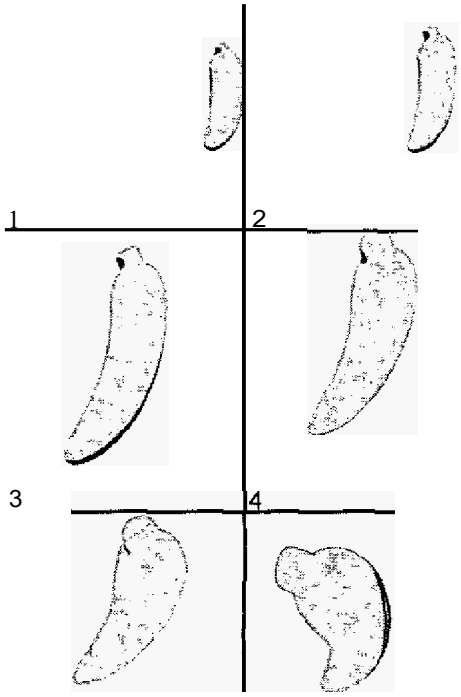
5.RESVLTS

我们的结果很好。我们的系统是基于Eric Lum和Kwan-Liu Ma提出的一个系统。他们的系统在不到一秒钟的时间内渲染128 x 128像素的单一物体。相比之下，我们可以在nVIDIA GeForce 6800上以大约每秒一帧50的512速度渲染512x像素的单个物体。我们的两个杯子的场景（图9）是以512×像素512的速度渲染的，大约每秒一帧25。

我们的系统产生的图像也比以前的一些结果有所改进，看起来更像真正的水彩画。尽管我们的图像比Curtis等人**[1]的图像**具有更少的真实水彩画的特征，**但**它们产生的速度却快得多。评估像我们这样的系统的唯一方法是检查我们的图像--见图和7,8图。 9.

还请注意，我们在动画中保持了良好的连贯性。只有两个特点有损于此。首先，用于降低图像完美度的噪声纹理从不移动。这是一个非常轻微的缺陷，它可以通过将噪声样本平面与原始场景摄像机而不是后处理摄像机对齐来减少。这将消除在世界中移动时的任何不连贯性，尽管旋转摄像机仍然会造成轻微的伪影。

第二个，也是更明显的特征，就是减弱了
动画中的连贯性是指用于笔触纹理的噪声总是与摄像机对齐。相机或场景旋转会导致物体在纹理空间中移动。这可以通过使用立方体映射的噪声纹理来进行修复。我们仍然可以*使用*法线偏移来获得物体轮廓的显示
通过在立方体地图的平面上投射法线来实现。即使不这样做，在我们的系统中，画笔的笔触总是可以预测的，而且看起来一点也不糟糕。图6展示了一个单一物体的简短动画。注意到笔触在大多数情况下是连贯的，并且总是遵循物体的轮廓。



56

图**6**：一个物体的小动画 注意笔触如何保持连贯性。对比度增加了，使笔触更加明显。

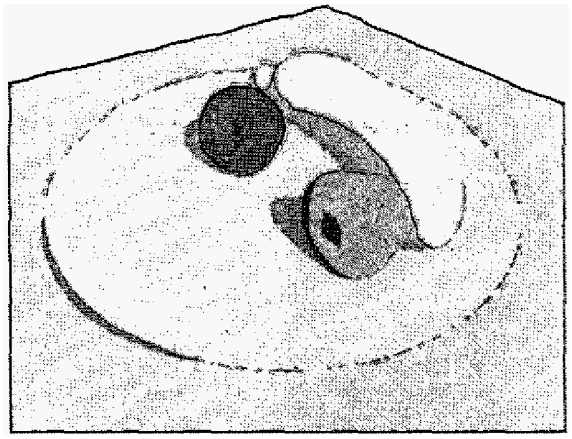


图**7**：用我们的系统渲染的一幅静物。另见颜色**碟子。**

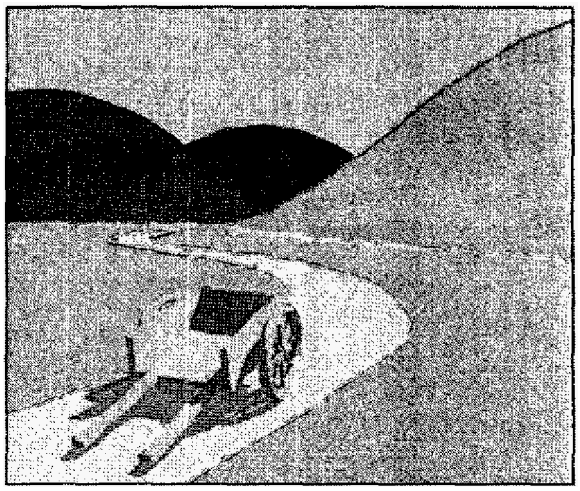
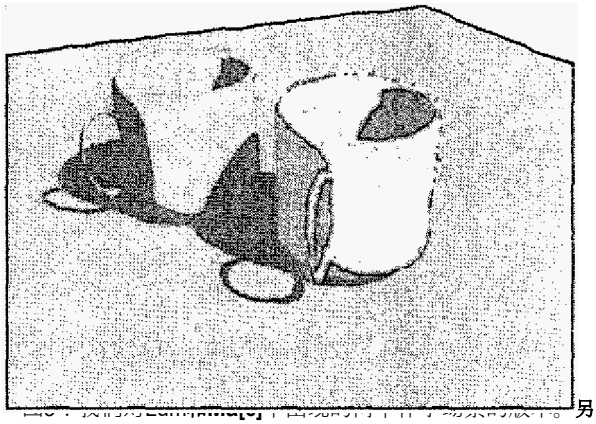


图8：用我们的系统渲染的风景。因此，也是彩色板。



彩图。

6.结论和未来的工作。

我们的系统是，**在**视觉效果方面，它既是对类似系统的改进，又是更快的。调制解调器GPU的强大功能使我们能够创建一个相当完整的系统，该系统仍然允许对场景进行实时导航。通过少量的修改，它还可以实时改变和创建用于这种渲染方式的场景。

为了公平起见，我们确实需要问问自己，我们的结果是否真的像真正的水彩画。我们认为，在许多方面，它们确实很像某些种类的水彩画。这就是说，我们制作的图像仍然缺乏真正的水彩画所必需的东西。这并不是说我们失败了，只是说某些真正的水彩画的某些方面从我们产生的图像中被提取出来了。

我们可以通过各种途径来试图纠正这个问题。最重要的是，对真实水洗的质量和特征进行详细的检查，将有助于改进我们的噪声纹理方法，使其产生更多的茶色水洗的特征。另一个重要的方面是，我们应该

调查是减少平面物体的非常 "平坦的颜色"外观的一种方法。使用深度来减少厚度可能会改善这里的结果。

其次，增加某种 纹 理处理器会有助于对纹理中每种颜色应该被分配的颜料做出"最佳猜测"。有了这个，我们就可以 用一种以上的颜料来制作物体，这将大大增加我们可以制作的场景的质量。这方面的一个延伸是尝试在物体中重现湿中带湿的色彩 混 合。我们相信，这可以通过噪声模糊来实现，这种模糊 的长度与物体上颜料区域中心的距离成比例增加。很明显，我们需要一种方法在我们的后处理步骤中获得这一信息。

最后，为了更接近于真正的水彩画，它如果我们真的要把物体叠加在一起，可能会增加很多。当然，在图层内，这种情况不应该发生，但在真正的水彩画中，发 现一个背景，中间和前景画在上面的情况并不少见。为了获得这些信息，我们可以将其转化为多个纹理，每个纹理只包含一定深度范围内的碎片。

我 们 的结果是非常有希望的。它们都很快速，而且看起来非常好。此外，随着GPU功率的增加，我们 可 以期待更多的可用功率。我们所做的所有补充都会有一些性能上的开销，然而，我们相信这种花费是值得的。

参考文献

[IJ.卡西迪-J-柯蒂斯，肖恩。Andeiron, Josh Seims, Kurt Fleischer, and David Salesin.计算机生成的水彩。在SIGGRAPH'97会议记录中，页面。421--430, 八月 1997.

[2].亚伦-赫兹曼。Paint by RelaJ{lltion.In CGl 2001, pages 47-52, 2001

[3]. Doug DeCarlo和Anthony Santella.Styli7.ation and Abstraction of照片。ACM SIGGRAPH第2002,769-776页。 2002.

[4J.Barbara J. Meier.动画的绘画式渲染。In S(GGRAPH Conference'96 Proceedings, pages 477-484, 1996.

[S].Eric Lengyel. The Mecliani.cs of Robust Stencil Shadows. Gamasutra.com。http://W\vw.gamasutracom/features/200Z10I J/lengyel pfv.hnn，2004年8月7日访问。

[6l Eric B. Lum 和 Kwan-Liu Ma.使用水彩启发的纹理和Ilh.uni.natioll的非写实渲染 在 第 九 届太平洋计算机图形和应用会议（PG'Ol）论文集，IEEE，322-331页 2001.

[7l Chet S. Haase and Gary W. Meyer.为逼真的图像合成建立颜料材料模型。在ACM Transactions on Graphics第305-11,4,335页，1992年。

[8].Victoria Interrante.通过主方向驱动的三维线积分卷积来说明体积数据中 的 曲面形状。在SIGGRAPH会议记录'97中，第109-116页。1997.

[9]. GeoffWyvill." TheNatureofNoise", http://www.cs. ot1go.ac nz/graphics/GeoIDNoisePages/Nature.htlll)，访问日期为I?n/2004。

[10].*Cg工具包用户手册发布* 来自 *1.2*.developer.nvidia.com, nVIDIA公司。 2004.

[11].William R Mark, R Steven Glanville, Kurt Akeley, and Mark J. Kilg！lfd.Cg:一个用类似C语言对图形硬件进行编程的系统。在SIGGRAPH 2003会议记录中，第896-907页。 2003.

[12].Lance Williams.在弯曲的表面上投下弯曲的阴影。在SIGGRAPH会议记录1978中，第270-274页。 1978.

图1：用实际的水彩画颜料制作的三个样板。左边是基本的**水洗，**中间是上光，右边是湿的。

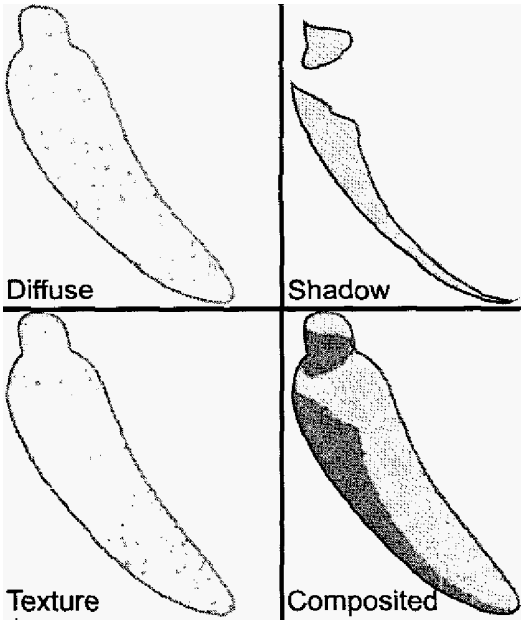


图3：香蕉上的三个颜料层，以及所有三个层的合成。请注意笔触是如何跟随纹理层的表面的。

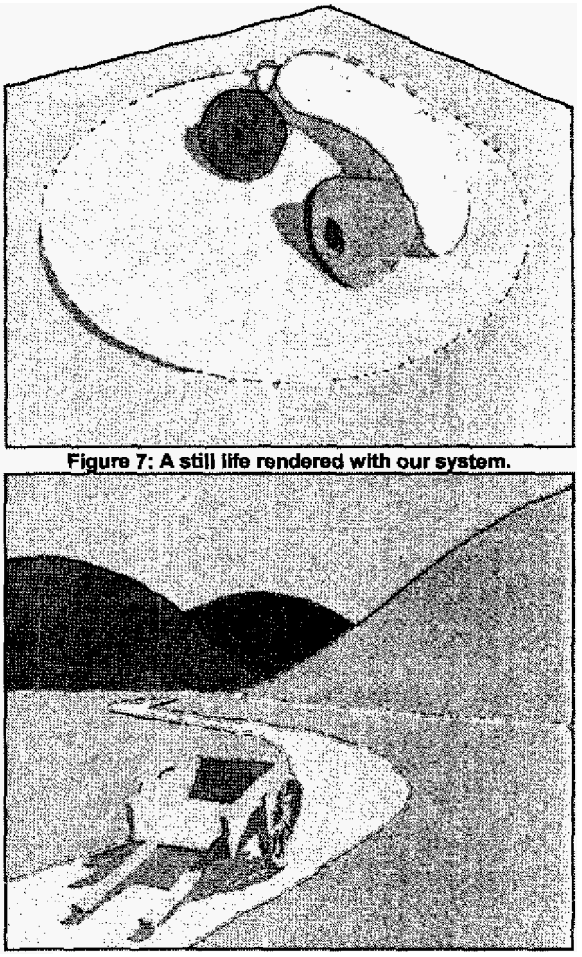


图8：用我们的系统渲染的景观。

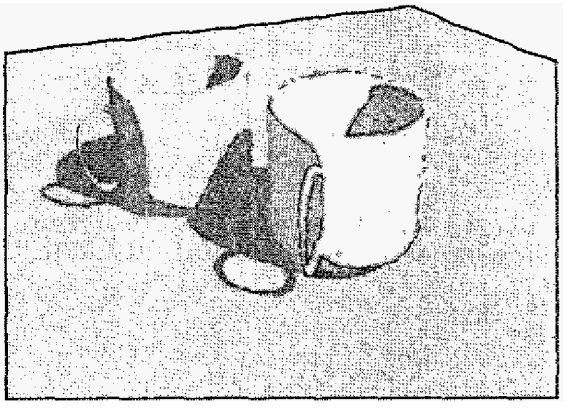


图9：我们对Lum和Ma[6]中出现的两个杯子的版本。