**狼人**

|  |  |  |  |
| --- | --- | --- | --- |
| 微软研究所 | 多伦多大学 | Richard Szeliski微软研究院 | 迈克尔F.科恩微软研究所 |

# 摘要

本文讨论了一种新的方法，即捕捉合成的和真实世界的物体和场景的完整外观，表示这些信息，然后用这种表示从新的相机位置渲染物体的图像。与ComputerVision中传统使用的形状捕捉过程和计算机图形中传统使用的渲染过程不同，我们的方法不依赖于几何表示。我们取而代之的是采样并重建一个4D函数，我们称之为Lumigraph。Lumigraphis是完整全光功能的一个子功能，它描述了光在各个方向上的各个位置的流动。使用Lumigraph，对象的新图像可以快速生成，而不依赖于场景或对象的几何或照明复杂性。本文讨论了一个完整的工作系统，包括样本的采集，Lumigraph的构建，以及从这种新的表现形式得到的图像的后续趋势。

# 1      介绍

在计算机图形学中创建虚拟环境或对象的过程首先是对环境中对象的几何和曲面属性以及任何灯光进行建模。环境的图像随后从虚拟摄像机的有利位置呈现出来。为了开发计算机辅助设计系统，允许对复杂的几何和材料属性进行规范，已经付出了很大的努力。类似地，大量的工作已经被用于生产模拟光在虚拟环境中的传播以创建真实图像的系统。

尽管做出了这些努力，但要重现现实世界中许多复杂的几何结构和微妙的灯光效果仍然是困难或不可能的。通过直接从真实世界中获取对象的几何和材质特性，可以绕过建模问题。这种方法通常涉及相机、结构光、测距仪和机械传感设备（如三维数字化仪）的一些组合。成功后，可以将结果输入渲染程序，以创建真实对象和场景的图像。不幸的是，这些系统仍然无法完全捕捉到几何和材料特性的小细节。现有的渲染方法在忠实再现真实世界照明的能力方面也继续受到限制，即使给定了精确的几何模型。



访问微软研究院时所做的工作。

|  |
| --- |
| 允许免费制作本作品的部分或全部或个人或课堂使用的数字或硬拷贝，前提是复制品不是为了盈利或商业利益而制作或分发的，且副本在第一页上附有本通知和完整的引文。若要以其他方式复制、重新发布、在服务器上发布或重新分发到列表，则需要事先获得特定的许可和/或收费。  ©1996 ACM-0-89791-746-4/96/008…3.5美元 |

Quicktime VR[6]是最早提出可以跳过传统建模/渲染过程的系统之一。相反，一系列捕捉到的环境地图允许用户从空间中的固定点环视场景。你也可以通过不同的视图来创建三维模型的轮廓。Chenand Williams[7]和Werner等人[30]通过模拟像素从一个摄像机位置移动到另一个位置时像素的运动（即光流）来研究图像之间的平滑插值。在全光学建模[19]中，McMillan和Bishop讨论了如何在柱面图像的立体对中找到每个像素的视差。给定视差（大致相当于深度信息），他们可以移动像素从新的有利位置创建图像。在[14]中讨论了使用立体对平面图像的类似工作。

本文扩展了Quicktime虚拟现实和全光建模，进一步发展了捕捉环境中一个区域内光的完整流动的思想。这种流动用一个充气函数来描述[1]。气动光学函数是一个五维量，描述了在每个3D空间位置和每个2D方向上的光流动。在本文中，我们讨论了捕获和表示一个全光函数的计算方法，以及使用这种表示从任意角度渲染环境图像的计算方法。

与Chen和Williams的视图插值[7]和McMillan和Bishop的全光模型[19]不同，我们的方法不明确依赖于任何光流信息。这些信息在实际操作中通常很难获得，特别是在具有复杂可见性关系或镜面反射面的环境中。然而，我们确实使用近似的几何信息来提高低采样密度下重建的质量。以前的基于流的方法隐式地依赖于漫反射表面反射，允许它们使用来自单个图像的像素来表示来自不同视点的单个几何体的外观。与此相反，我们的方法定期对全腔光学函数进行采样，因此对反射特性没有任何假设。

如果我们只考虑离开有界物体的光的子集（或等效地进入有界的空白区域），沿着任何光线的辐射保持不变这一事实使我们可以将全光函数的关注范围缩小到四维。本文首先讨论了这个4D函数的表示，我们称之为Lumigraph。然后，我们讨论了一种用廉价的手持式相机对全光功能进行采样的系统，并将捕获的光“显影”成Lumigraph。最后，本文描述了如何使用纹理映射硬件在虚拟摄像机模型下从任意视点快速重建图像。Lumigraph表示也适用于合成对象，允许我们对复杂模型的完整外观进行编码，并以独立于模型复杂性的速度重新呈现对象。我们提供了合成和真实序列的结果，并讨论了目前正在进行的使系统更有效的工作。



我们假设介质（即空气）是透明的。

# 2      代表

## 2.1     从5D到4D

气动函数是5个变量的函数，表示位置和方向。如果我们假设空气是透明的，那么沿着光线穿过空白区域的辐射率保持不变。如果我们进一步限制我们的兴趣在光离开一个有边界物体的凸面上，那么我们只需要沿着物体周围的某个表面来表示全光函数的值。立方体因其计算简单而被选中（见图1）。在空间中的任何一点上，可以通过沿着光线向后追踪穿过空白空间到立方体的表面来确定沿任何方向的光线的辐射度。因此，由于物体而产生的气动光学函数可以简化为4维。

以前曾有过将气动光学功能限制在某些周围表面的想法。在全视差全息立体图[3]中，物体的外观是通过移动相机沿着某个表面（通常是平面）捕捉到的，捕捉到的是一组2D照片。这个阵列然后被转移到一个单一的全息图像，它可以显示三维物体的外观。本文报道的工作从全息立体图中提取了许多概念。

全球照明研究人员已经使用“表面受限的全光函数”来有效模拟包含复杂几何物体的环境区域之间的光传输。在一个立方体的表面上，围绕着某个区域，这个信息就是模拟光从该区域到所有其他区域的全部信息[17]。在照明工程的背景下，这一思想已被用来建模和表示物理照明的照明。Ashdown[2]描述了一种机架，用于沿着感兴趣灯具周围的球体移动相机。捕捉到的信息可以用来表示全局照明模拟中的光源。阿什当将表面受限的全光函数的概念追溯到莱文[15]。

Katayama等人描述了本文中报道的一个有限版本。[11] 一。在他们的系统中，摄像机沿着轨道移动，捕捉到某个物体的一维图像阵列。然后，这些信息被用来从空间中的其他点生成物体的新图像。由于它们只捕捉沿一条直线的全光函数，因此只能获得水平视差，并且在该直线上引入了畸变。最后，在与我们同时进行的工作中，Levoy和Hanrahan[16]代表了一个4D功能，它允许从空间的任何地方对物体进行无失真的全视差视图。

## 2.2     4D Lumigraph的参数化

有许多潜在的方法可以参数化Lumigraph的四个维度。我们采用了类似于数字全息立体图的参数化方法[9]，也被Levoy和Hanrahan[16]采用。我们从一个立方体开始组织一个Lumigraph，并且在不丧失一般性的前提下，只考虑立方体的一个正方形面（完整的Lumigraph是由六个这样的面构成的）。



我们只考虑函数的快照，这样就省去了时间。在不损失通用性的情况下，我们也只考虑单色函数（在实践中有3个离散的颜色通道），消除了考虑波长的需要。此外，我们忽略了动态范围的问题，从而将自己限制在某个有限范围内的标量值。

类似地，我们可以在一个空凸区域内重建完整的全光函数，只需将其表示在空白区域的边界上。在区域内的任何点上，通过找到该方向与区域边界的交点，可以找到从任何方向进入的光。

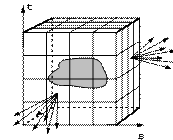


图1：立方体的表面保存了由于封闭对象而产生的所有辐射信息。

|  |  |  |
| --- | --- | --- |
|  | **s** | 射线的4D参数化  **美国** |
| （b） 4D空间的二维切片  雷（s，u） |  | （c） 二维光线空间  **美国** |

图2:Lumigraph的参数化

我们选择一个简单的立方体面参数化，正交轴平行于标有和的边（参见图1）。使用与标有and轴的平面平行的第二个平面参数化方向（图2）。因此，4D Lumigraph中的任何一个点都可以通过它的四个坐标来识别，这四个坐标是贯穿第一个平面并在第二个平面相交的坐标（见图2中的射线）。我们将原点放在平面的中心，轴垂直于平面。飞机位于。完整的Lumigraph由六对这样的行星组成，它们的法线沿着，，，和方向。

有时考虑两个2D类似物与4D Lumigraph是很有启发性的。图2（b）显示了4D Lumigraph的二维切片，它指示了轴和轴。图2（c）显示了2D光线坐标中相同的排列方式，其中光线映射到点（例如，光线（）），点映射到线。

图3显示了Lumigraph的参数化与任意图像中的像素之间的关系。给一个陆-



更精确地说，光线空间中的线表示穿过空间中某个点的光线集。

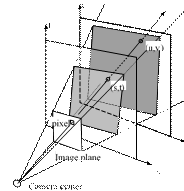


图3:Lumigraph和任意图像中像素的关系

migraph，可以生成一个任意的新图像，每个像素都用适当的值着色。相反地，给定一些任意的图像和相机的位置和方向，每个像素可以被视为Lumigraph值的样本，用于构造Lumigraph。

两个平行平面参数化有许多优点。给出了射线的几何描述，计算它的坐标在计算上很简单；人们只需找到它与两个平面的交点。此外，利用现代工作站上硬件内置的纹理映射操作，可以快速完成从该参数化进行的重建（见第3.6.2节）。最后，在这个参数化中，当一个视点沿着平面沿直线移动时，在平面上的投影是几何对象轨迹上的点沿着平行的直线。这使得计算效率很高，可以计算几何点（即光流）的视在运动，并对Lumigraph应用深度校正。

## 2.3     4D参数化的离散化

到目前为止，Lumigraph已被讨论为一个未知的，连续的，四维函数，在超三次域和标量范围内。要将这样的对象映射到计算框架中，需要离散表示。换句话说，我们必须选择函数所在的有限维函数空间。为此，我们在每个维度中选择一个离散的细分，并将一个系数和基本函数（重建内核）与每个4d网格点相关联。

在和中选择Subdivision，在和中选择Subdivision将生成和平面上的点网格（图4）。网格点用编入索引，位于。网格点用编入索引，位于

. 一个4D网格点被索引。此网格点的数据值（实际上是一个RGB三元组）称为

### 2.3.1    依据的选择

我们将每个网格点与一个基函数相关联，以便将连续Lumigraph重构为线性和，其中存在于由基选择定义的空间中的有限维Lumigraph。

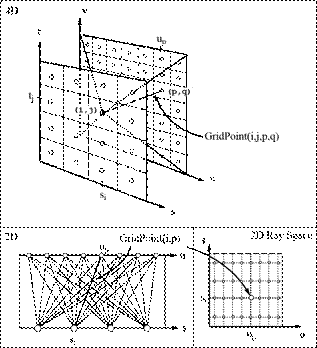


图4:Lumigraph的离散化

例如，如果我们选择常数基函数（即，在最接近相关网格点的4D区域中，值为1，在其他地方为零的4D框），则Lumigraph为分段常数，并采用最近网格点的系数值。

类似地，一个四边形的基本函数在网格点处可用，并在所有相邻的网格点上衰减。因此，的值从构成该点所在的超立方体的16个网格点进行插值。

我们选择使用四边形基是因为它的计算简单性和连续性。然而，由于这个基不受Nyquist频率的限制，因此相应的有限维函数空间不是平移不变的[24]，因此网格结构在我们的结果中会稍显明显。

### 2.3.2    投影到选定的基础上

如果连续的Lumigraph，并且选择了有限维Lumigraph的基础，我们仍然需要定义into的投影（即，我们需要找到导致

这是最接近的指标）。如果我们选择距离度量，那么投影是通过对内积给出的基函数[8]的对偶积分来定义的，

（一）

在箱基的情况下。四边形基函数的对偶更为复杂，但这些基函数足够接近它们自己的对偶。

在用核函数进行低通滤波后，可以将此投影解释为点采样。哈雷[9]在全息立体图的背景下进行了这种解释。我们也可以将这种投影解释为将一个物理或合成的“倾斜”相机放置在网格点上，其光圈与双线性基相对应，像素位于

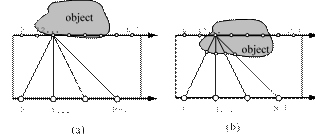


                图5：平面分辨率的选择

用双线性滤波器进行抗锯齿处理。这种类比在[16]中得到了应用。

在图16中，我们展示了从Lumigraphs生成的图像。几何场景由一个部分立方体组成，前面是粉红色的脸，后面是黄色的脸，地板上是棕色的脸。这些lumigraph是使用两种不同的求积方法来近似方程1，并使用两组不同的基函数，即常数和四边形。在（a）和（c）中，只有一个样本被用来计算偏头痛系数。在这些例子中，可以看到一些艺术品。计算了In（b）和（d）的数值积分。很明显，最好的结果是用一个完全求积的方法，得到四元逼近基函数。

### 2.3.3    分辨率

一个重要的决定是如何设置分辨率，并且，最好地平衡效率和从Lumigraph重建图像的质量。和的选择受到这样一个事实的影响，即我们期望对象的可见表面比平面更靠近平面。在这种情况下，平面的分辨率与最终图像分辨率密切相关，因此选择接近最终图像分辨率效果最好（我们考虑128到512的分辨率范围）。

从平面上的一个网格点观察lumigraph的2D子集（见图5（a））。如果物体的表面正好位于平面上的一个网格点上，那么离开该点的所有光线都代表物体表面上某个位置的辐射函数的样本。即使物体表面偏离平面，如图5（b）所示，我们仍然可以期望平面上的函数保持平滑，低分辨率就足够了。因此，比预期低得多的分辨率可以产生良好的结果。在我们的实现中，我们使用16到64之间的值。

### 2.3.4    几何信息的使用

假设物体的基本函数行为，关于物体几何的知识给我们提供了有关关联的lumigraph函数的相关性的信息，并且可以用来帮助定义我们的基本函数的形状。

考虑二维Lumigraph中的光线（图6）。距此光线最近的网格点是。但是，网格点和可能包含更接近于的真实值的值，因为这些网格点表示的光线与IntersectionWith相交。这意味着调整基本函数的形状。

假设我们知道第一次与物体表面相交的深度值。那么对于给定的，我们可以计算出与同一几何体相交的光线的对应关系-

|  |  |
| --- | --- |
| z射线（s、u） |  |

图6：射线深度校正

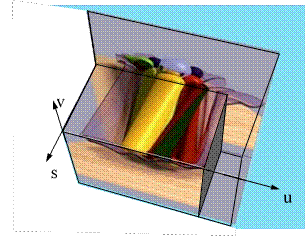


                   图7:Lumigraph的切片

对象上的ric位置作为原始光线。让深度在平面和平面上。然后通过检查图6中的相似三角形可以找到交叉点，

（二）

在光线空间（图6（b））中绘制的图6（a）中查看相同的情况是很有启发性的。在这个图形中，三角形就是射线

，圆圈表示离散的Lumigraph附近的网格点。穿过的对角线表示物体在前后移动时的交点的光流（在本例中为水平运动2d）。这条线与和的交点分别发生在和处。

图7显示了图19中使用的光线跟踪果盘的Lumigraph三维子空间切片。像素运动在这个空间中沿着直线流动，但如果场景包含透明度，可能会出现多个运动。流线的斜率对应于沿流线追踪的对象上的点的深度。注意函数是如何沿着这些流线连贯的[4]。

我们期望Lumigraph沿着光流线是平滑的，因此使基函数相应地适应它们的形状是有益的。重新映射和赋值并执行此整形。在有限元方法中，形成基函数的支撑以使其与被逼近函数的结构紧密匹配的思想被广泛地应用于有限元方法中。例如，在用于图像合成的光能传递方法中，元素的网格适应于有关发光函数的知识。



假设能见度没有变化。

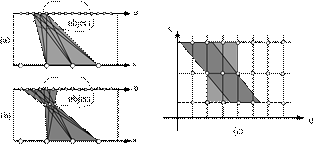


图8：（a）支持未修正的基函数。（b） 支持深度校正基函数。（c） 在光线空间中支持两个基函数。

     新的基函数由

找到并使用方程式2，然后进行计算，即

新的深度校正基虽然形状复杂，但仍然是系数的线性和，各贡献基函数的权重仍然是统一的。然而，基不再表示为简单盒或hatsasbefore的张量积。图8显示了二维几何空间和二维光线空间中未修正（浅灰色）和深度校正（深灰色）基本函数的支持。请注意，与未校正的基准相比，深度校正基准的支撑如何使对象的曲面穿过更窄的区域。

在我们的系统中，我们使用了深度校正的四边形基函数。修正后的四边形基础中的值使用以下计算方法计算：

四边形近深度校正（s，t，u，v，z）

/\*网格间距\*/

      周围的四个人

            周围的四个人

返回结果

图17显示了使用未校正和深度校正的基函数从Lumigraph生成的图像。深度校正使用162个多边形模型完成，以近似原始70000个多边形。近似值是使用网格简化程序生成的[10]。这些图像显示了深度校正如何减少图像中的伪影。

# 3      Lumigraph系统

本节讨论了许多与创建Lumigraph和从中生成图像相关的实际实现问题。图9显示了系统的框图。这个过程开始于用手持式相机拍摄图像。从已知标记

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 捕获 |  | 利比 |  | 压缩 |  | 重建图像 |



图9:Lumigraph系统

在图像中，估计相机的位置和方向（其姿势）。这提供了足够的信息来创建一个近似的几何对象，用于值的深度校正。更重要的是，每个图像中的每一个像素都充当了全光函数的样本，并用于估计离散Lumigraph的系数（即，发展Lumigraph）。或者，合成对象的Lumigraph可以通过在渲染系统中集成一组投射路径来直接生成。我们只简单地讨论压缩问题。最后，给定一个任意的虚拟摄像机，对象的新图像被快速渲染。

## 3.1     合成场景捕获

创建合成场景的Lumigraph很简单。通过将虚拟针孔相机的中心放在轴上向下看，并使用正方形作为胶片位置来定义成像截锥，可以为每个网格点捕获单个样本的每Lumigraph系数。使用这种倾斜透视相机渲染图像会产生Lumigraph系数。此图像中的像素值（索引）用于光度系数。为了对内核执行积分，每个系数的多条光线可以通过抖动摄影机和像素位置来平均，并使用

. 对于光线跟踪渲染，我们使用了生成建模包[25]提供的光线跟踪程序。

## 3.2     捕捉真实场景

计算真实物体的Lumigraph需要从大量的视角获取物体图像。实现这一点的一种方法是使用一个特殊的运动控制平台，将真实的摄像机放置在与网格点重合的位置和方向上[16]。虽然这是一个合理的解决方案，我们感兴趣的是用一个普通的手持摄像机获取图像。这样就形成了一个简单而廉价的系统，并且可以将适用范围扩大到更大的场景和对象。

为了达到这个目标，我们必须首先校准摄像机，以确定方向和图像坐标之间的映射关系。接下来，我们必须在每个图像中识别出特殊的校准标记，并根据这些标记计算相机的姿势。为了使Lumigraph的深度校正插值成为可能，我们还希望恢复对象的粗略几何模型。为此，我们使用蓝屏技术将每个输入图像转换成asilhouet，然后根据这些二进制图像建立一个体积模型。

### 3.2.1    摄像机标定与姿态估计

摄像机标定和姿态估计可以看作是一个过程的两个部分：确定屏幕像素和光线之间的映射。与此过程相关联的参数自然分为两组：定义相机姿势（刚性旋转和平移）的外部参数和定义三维相机坐标在屏幕上的映射的内在参数。后一种映射不仅包括从三维坐标到未失真的透视（针孔）投影

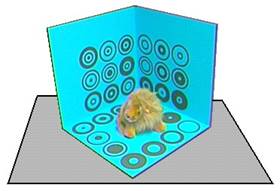


图10：捕获阶段

图像坐标，还有一个径向失真变换和最终平移和缩放到屏幕坐标[29，31]。

我们使用一个固定镜头的相机，因此在整个过程中，内在参数保持不变，只需在数据采集开始前估计一次。然而，外部参数会不断变化，需要为每个新的视频帧计算。幸运的是，在给定固有参数的情况下，可以用较少的校准点高效而准确地完成这项工作。为了计算内参数和外参数，我们采用了一种由Tsai[29]开发并由Willson扩展的算法

[31]。

一个专门设计的阶段提供校准数据的来源（见图10）。舞台有两个以直角固定在一起的墙和一个可以与墙分离并以度增量旋转的底座。放置在这样一个活动基座上的物体可以从上半球的各个方向观看。舞台背景被涂成青色，以便以后进行蓝屏处理。30个标记，每一个都由几个同心环组成，颜色较深的青色，沿着侧面和底部分布。这个数字足够高，可以进行非常精确的相机内部校准。在外部相机校准过程中，只有8个或更多的标记需要可见才能可靠地计算出一个姿势。

通过首先将图像转换为二进制（即黑色或白色）图像，可以在每个图像中定位标记。双阈值运算符将所有图像像素分成三组，这些组由强度阈值和分隔开。亮度低于的像素被视为黑色，亮度高于的像素被视为白色。强度介于和之间的像素只有在有一个黑色邻居时才被视为黑色，否则它们被视为白色。然后在二值阈值图像中搜索连接的组件[23]。具有相似重心的连接组件集可能是标记的候选对象。最后，利用每个标记的半径比来唯一地识别标记。为了帮助用户正确地采样观察空间，实时视觉反馈将显示相机在视图空间中的当前位置和过去位置（图11）。标记跟踪，姿态估计，反馈显示和帧记录大约需要1/2秒每帧在一个SGI索引。

## 3.3     三维形状近似

从自然图像中恢复三维形状信息一直是计算机视觉研究的热点。许多这些技术都假设一个特别简单的形状模型，例如，一个多面体中心，所有边都是可见的。其他技术，如立体匹配，产生稀疏或不完整的深度估计。为了生成完整的、封闭的三维模型，已经尝试了几种方法。一系列技术建立三维体积模型

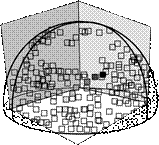


图11:image capture stage的用户界面显示了观察球体上当前和以前的摄像机位置。用户的目标是“绘制”球体。



图12：分割图像加体积构造

直接从被观察对象的轮廓中提取[21]。另一种方法是将可变形三维模型拟合为稀疏的空间数据。尽管经过20多年的研究，从图像中可靠地提取三维几何信息（不使用主动照明和定位硬件）仍然是难以捉摸的。

幸运的是，对物体形状的粗略估计就足以大大有助于从Lumigraph捕捉和构建图像。在这个过程中，我们使用了[26]中描述的八叉树构造算法。每个输入图像首先使用蓝屏技术分割成二进制对象/背景图像[12]（图12）。完全封闭对象化的立方体的八叉树表示。在每一个分段图像中，每一个位于八叉树上的体素都被投影到成像平面上，并被测试的数据包含对象的各个部分。如果一个体素落在轮廓之外，它将从树中移除。如果它落在边界上，它会被标记成八个更小的立方体。在对少量图像进行处理后，所有的图像都标记为立方分频。该算法继续进行预设数量的细分，通常为4。生成的三维模型由一组体素组成，这些体素不包含对象（图12）。收集外部多边形，然后使用Taubin的多边形平滑算法对生成的多面体进行平滑处理[27]。

## 3.4     重新定局

如等式1所述，与基函数相关的系数定义为连续Lumigraph函数乘以某个核函数的积分。这可以写成

（三）

实际上，这个积分必须用有限数量的函数样本来计算。来自手持摄像机的输入视频流中的每个像素代表一个样本



从技术上讲，体积是物体视觉外壳的一个超集合[13]。

，Lumigraph功能。因此，无法预先指定或控制域中的采样点。此外，也不能保证输入的样本是均匀分布的。

从这些样本构造Lumigraph类似于多维散乱数据的近似问题。在Lumigraph环境下，这个问题很难解决，原因有很多。由于样本不是均匀分布的，因此不能应用标准的基于傅立叶的采样理论。由于采样点的数量可能很大（），而且由于我们在四维空间中工作，所以求解方程组（如求解薄板问题[28,18]）或构建空间数据结构（如Delauny三角剖分）的成本太高。

除了样本点的数量，数据样本的分布还有两个特点，这使得问题变得尤为困难。首先，采样密度可能相当稀疏，具有较大的间隙区域。第二，取样密度非常不均匀。

第一个问题已经在burt[5]描述的二维cattereddata近似算法中得到了解决。在他的算法中，使用图像金字塔来创建一个低分辨率数据集的层次集。每一个较低的分辨率代表输入数据的“模糊”版本；在较低的分辨率下，数据中的间隙变小。这些低分辨率的数据然后被用来填补高分辨率的空白。

第二个问题，采样密度的不均匀性，已经由Mitchell[20]解决。他解决了在非均匀密度下获得像素值的问题。在这个问题中，当平均采样值时，不希望结果受到采样最密集的区域的过度影响。他的算法通过计算一些较小区域的平均值来避免这种情况。然后，通过将这些层的值平均起来计算像素的最终值。这一平均值不以落在每个地层中的样本数量加权。因此，样本的不均匀性不会使答案产生偏差。

对于我们的问题，我们开发了一种新的分层算法，它结合了这两种算法的观点。与Burt一样，我们的方法使用金字塔算法来填补空白，而像Mitchell一样，我们确保数据的不均匀性不会影响“模糊”步骤。

为了便于表示，该算法在1D中描述，并且只使用一个索引。使用了一组层次化的基函数，其中highestresolution标记为higheresolutions，而lowerresolutions则具有较高的精度。与每一个分辨率相关的是一个权重。这些权重决定了不同分辨率级别的系数是如何组合在一起的。这些权重的使用是我们算法的显著特点。

算法分三个阶段进行。在第一阶段，称为splat，样本数据用于近似方程3的积分，得到系数和权重。在没有样本数据的地区，权重为全部或零。在第二阶段，称为pull，通过合并来自高分辨率网格的系数值，在低分辨率网格的层次集合中计算基函数的系数。在较低分辨率的网格中，间距（权重较低的区域）变得更小（见图13）。在第三阶段，称为推送，来自每个较低分辨率网格的信息与下一个高分辨率网格相结合，在不过度模糊的情况下填补空白-

将已计算出的更高分辨率信息振铃。

### 3.4.1    抛雪球算法

在散开阶段，通过使用以下加权平均值进行蒙特卡罗积分来计算系数-

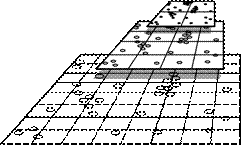


图13:2D推拉。在较低的分辨率下，差距更小。

激励：

（四）



其中表示样本的域位置。如果是，则未定义。如果具有紧支撑，则每个样本只影响一个常数的系数。因此，此步骤以时间线性方式在采样数上运行。

如果样本点是从均匀分布中选择的，则该估计收敛到方程（3）中积分的正确值，并且对于样本点，方差约为

. 这个变化和那个相似

利用重要性抽样得到的，这通常比粗糙的蒙特卡罗估计量小得多。关于这个估计量的全面分析，见[22]。

### 3.4.2    拉

在拉阶段，使用一组更宽的核来推导函数的低分辨率近似值。这些更宽的核是通过将高分辨率的核线性相加来定义的

（）使用一些离散序列。对于线性“帽子”函数，是

低分辨率系数通过组合高分辨率系数来计算。一种方法是计算

（五）



很容易看出，这个公式对应于Burt所用的方法，计算出的结果与应用于更宽核的原始估计量（方程（4））相同。同样，如果采样密度是一致的，这个估计量就起作用了。不幸的是，在进行总体调查时，假设数据是统一采样是不明智的。例如，用户可能在某个特定区域长期使用摄像头。这种非均匀性会使估计量产生很大的偏差。

我们对这个问题的解决方案是将米切尔的推理应用于这一背景，将等式（5）替换为：



该值表示完全饱和，运算符用于将一个上限放在一个高度上



如果的规范化不固定，则使用该值不会导致通用性损失。

采样区域，会影响总和。

拉阶段在时间上以线性方式运行，在所有分辨率上求和基函数的数目。因为每个较低分辨率的基函数密度是基函数密度的一半，所以这一阶段在分辨率下的基函数数量上是时间线性的。

### 3.4.3    推

在推送阶段，使用较低的解决方案来填充重量较轻的高分辨率区域。如果一个较高的分辨率系数具有高的相关置信度（即权重大于1），则我们完全忽略那里的低分辨率信息。如果较高分辨率的系数没有足够的权重，则将低分辨率的信息混合在一起。

为了混合这些信息，函数的低分辨率近似值必须在高分辨率的基础上展开。这是用一个满足条件的序列进行抽样和卷积

.

我们首先计算临时值



现在可以将这些临时值与已处于级别的值和值混合。

这类似于在图像合成中执行的混合。

### 3.4.4    几何信息的使用

当使用深度校正基函数时，这种三阶段算法必须稍作调整。在展开阶段，每个取样器必须在第2.3.4节中说明其和值。此外，在推拉阶段，使用深度校正指数，而不是简单地使用基函数与相邻指数组合系数。

### 3.4.5    二维结果

首先将其应用于二维图像，验证了算法的有效性。图18（a）显示了一组来自著名的mandrill图像的分散样本。这些样本是通过随机选取256条线段，并沿着这些线段密集地取样。图像（b）显示应用拉/推算法后得到的图像。图（c）和（d）显示相同的过程，但只有100条采样线。我们的算法在2D图像函数和4D Lumigraph函数上的成功使我们相信它可能还有许多其他用途。

## 3.5     压缩

一个直接的Lumigraph采样需要大量的存储。对于第4节中所示的示例，我们使用空间采样和



这实际上没有米切尔的原始算法那么极端。在这种情况下，他的算法将所有非零权重设置为1。

在这一阶段，可以用方差度量代替权重作为置信度的度量。

我们选择这种类型的采样模式是因为它在很多方面模仿了从手持相机采集的Lumigraph样本的结构。在这种情况下，每个输入的视频图像都是沿2D平面的4D Lumigraph的密集采样。

图像。要以每像素24位存储我们的观察立方体的六个面，需要GB的存储空间。

幸运的是，样本之间有大量的一致性。我们可以将变换码应用于4D阵列，例如小波变换或块DCT变换。在给定几何信息的情况下，我们可以把4Darray看作2D图像数组来处理。然后我们可以从相邻的图像（即相邻位置的图像）中预测新图像。帧内压缩问题与压缩单个图像相同（一个简单的JPEG压缩可以节省大约20:1的成本）。帧间压缩可以比其他压缩利用更多的信息因为我们知道目标是静态的，知道相邻图像之间的摄像机运动，可以预测像素的运动。此外，我们可以利用这样一个事实，即我们有一个二维阵列的图像，而不是一个单一的线性视频流。

虽然我们还没有完成对压缩问题的全面分析，但我们的初步实验表明，压缩比为200:1应该是可以实现的，几乎没有退化。这将存储需求降低到MB以下。显然，使用更复杂的预测和编码方案可以预期得到进一步的改进。

## 3.6     图像重建

Given a desired camera (position, orientation, resolution), the reconstruction phase colors each pixel of the output image with the color that this camera would create if it were pointed at the real object.

### 3.6.1    Ray Tracing

Given a Lumigraph, one may generate a new image from an arbitrary camerapixel by pixel, ray by ray. Foreach ray, the corresponding coordinates are computed, the nearby grid points are located, and their values are properly interpolated using the chosen basis functions (see Figure 3).

In order to use the depth corrected basis functions given an approximate object, we transform the coordinates to the depth corrected before interpolation. This depth correction of the values can be carried out with the aid of graphics hardware. The polygonal approximation of the object is drawn from the point ofviewandwith thesameresolutionasthe desiredimage. Eachvertex is assigneda red, green,blue valuecorrespondingto its coordinate resulting in a “depth” image. The corrected depth value is found by examining the blue value in the corresponding pixel of the depth image for the -faces of the Lumigraph cube (or the red or green values for other faces). This information is used to find and with Equation 2.

### 3.6.2    Texture mapping

Theexpenseof tracing a ray for eachpixelcanbe avoidedby reconstructing images using texture mapping operations. The plane itself is tiled with texture mapped polygons with the textures defined by slices of the Lumigraph: . In other words, we have one texture associated with each gridpoint.

### Constant Basis

Consider the case of constant basis functions. Suppose we wish to renderanimagefrom the desiredcamerashownin Figure 14. The set of rays passing through the shaded square on the plane have coordinatesclosestto thegrid point . Supposethat the

plane is filled with . Then, when using constant basis functions, the shaded region in the desired camera&apos;s film plane should be filled with the corresponding pixels in the shaded region of the plane. This computation can be accomplished by placing a virtualcamera at the desiredlocation, drawing a squarepolygonon the

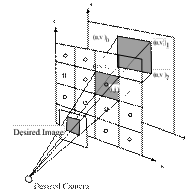


Figure 14: Texture mapping a portion of the st plane

plane, and texture mapping it using the four texture coordinates

           , , , and to index into .

Repeating this process for each grid point on the plane and viewing the result from the desired camera results in a complete reconstruction of the desired image. Thus, if one has an resolution for the plane, one needs to draw at most texture mapped squares,requiring on average, only one ray intersection for each square since the vertices are shared. Since many of the squareson the planeare invisible from the desired camera, typically only a small fraction of these squares need to be rendered. The rendering cost is independent of the resolution of the final image.

Intuitively, youcanthink ofthe planeasa pieceof holographic film. As your eye moves back and forth you see different things at the same point in since each point holds a complete image.

### Quadralinear Basis

The reconstruction of images from a quadralinear basis Lumigraph can also be performed usinga combinationof texture mapping and alpha blending. In the quadralinear basis, the support of the basis function at covers a larger square on the plane than does the box basis (see Figure 15(a)). Although the regions do not overlap in the constant basis, they do in the quadralinear basis. For a given pixel in the desired image, values from 16 4D grid points contribute to the final value.

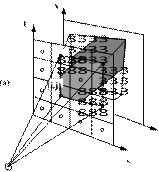
The quadralinear interpolation of these 16 values can be carried out as a sequence of bilinear interpolations, first in and then in . A bilinear basis function is shown in Figure 15(b) centered at grid point . A similar basis would lie over each grid point in and every grid point in .

Texturemappinghardwareonan SGIworkstationcanautomatically carryoutthebilinearinterpolation ofthe texturein . Unfortunately, there is no hardware supportfor the bilinear interpolation. We couldapproximatethebilinearpyramidwith a linearpyramidby drawing the four triangles shown on the floor of the basis function in Figure 15(b). By assigning values to each vertex ( at the center, and at the outer four vertices) and using alpha blending, the final image approximatesthe full quadralinearinterpolation with a linear-bilinear one. Unfortunately, such a set of basis functions do not sum to unity which causesserious artifacts.

A different pyramid of triangles can be built that does sum to unity and thus avoids these artifacts. Figure 15(c) shows a hexagonalregion associatedwith grid point and an associated linear basis function. We draw the six triangles of the hexagonwith at the center and at the outside six vertices . The linear interpolation of values together with the bilinear interpolation of the texture map results in a linear-bilinear interpolation. In practice we have foundit to be indistinguishablefrom the full quad-



The alpha blending mode is set to perform a simple summation.

**(b)**

Bilinear basis centered at (i,j)

**u** 

Linear Basis

Desired Camera

centered at (i,j) Figure 15: Quadralinear vs. linear-bilinear

ralinear interpolation. This process requires at most texture mapped, -blended triangles to be drawn.

### Depth Correction

As before, the coordinates of the vertices of the texture mapped triangles can be depth corrected. At interior pixels, the depth correction is only approximate. This is not valid when there are large depth changes within the bounds of the triangle. Therefore, we adaptivelysubdividethe triangles into four smaller onesby connectingthe midpoints of the sides until they are (a) smaller than a minimum screen size or (b) have a sufficiently small variation in depth at the three corners and center. The values at intermediate vertices are the average of the vertices of the parent triangles.

# 4      Results

We have implemented the complete system described in this paper and have created Lumigraphs of both synthetic and actual objects. For synthetic objects, Lumigraphs can be created either from polygon rendered or ray traced images. Computing all of the necessary images is a lengthy process often taking weeks of processing time.

For real objects, the capture is performed with an inexpensive, singlechipPanasonicanalogvideocamera. Thecapturephasetakes less than onehour. The captureddata is then “developed”into a Lumigraph. This off-line processing, which includes segmenting the imagefrom its background,creatinganapproximatevolumetricrepresentation, and rebinning the samples, takes less than one day of processing on an SGI Indy workstation.

Once the Lumigraph has been created, arbitrary new images of the object or scene can be generated. One may generate these new images on a ray by ray basis, which takes a few seconds per frame at resolution. If one hashardware texture mapping available, then one may use the acceleration algorithm described in Section 3.6.2. This texture mapping algorithm is able to createmultiple frames per second from the Lumigraph on an SGI Reality Engine. The rendering speedis almost independentof the desired resolution of the output images. The computational bottleneck is moving the data from main memory to the smaller texture cache.

Figure 19 shows images of a synthetic fruit bowl, an actual fruit bowl, anda stuffed lion, generatedfrom Lumigraphs. No geometric information was used in the Lumigraph of the synthetic fruit bowl. For the actual fruit bowl and the stuffed lion, we have used the approximate geometrythat was computedusing the silhouette information. Theseimagescanbegeneratedin a fraction ofa second,independent of scenecomplexity. The complexity of both the geometry and the lighting effects present in these images would be difficult to achieve using traditional computer graphics techniques.

# 5      Conclusion

In this paper we have described a rendering framework based on the plenoptic function emanating from a static object or scene. Our method makes no assumptions about the reflective properties of the surfacesin the scene. Moreover, this representationdoesnotrequire usto deriveanygeometricknowledgeaboutthescenesuchasdepth. However,this methoddoesallow usto includeanygeometricknowledge we may compute, to improve the efficiency of the representation and improve the quality of the results. We computethe approximate geometry using silhouette information.

We havedevelopeda system for capturing plenoptic data using a hand-heldcamera,andconvertingthisdatainto aLumigraphusinga novelrebinningalgorithm. Finally, wehavedevelopedanalgorithm for generating new images from the Lumigraph quickly using the power of texture mapping hardware.

In the examples shown in this paper, we have not captured the completeplenopticfunctionsurroundinganobject. Wehavelimited ourselves to only one face of a surrounding cube. There should be no conceptualobstaclesto extendingthis work to complete captures using all six cube faces.

There is much future work to be done on this topic. It will be important to develop powerful compression methods so that Lumigraphs can be efficiently stored and transmitted. We believe that the large degree of coherence in the Lumigraph will make a high rate of compression achievable. Future research also includes improving the accuracy of our system to reduce the amount of artifacts in theimagescreatedby theLumigraph. With theseextensions we believe the Lumigraph will be an attractive alternative to traditional methods for efficiently storing and rendering realistic 3D objects and scenes.

# Acknowledgments

The authors would like to acknowledge the help and advice we received in the conception, implementation and writing of this paper. Thanks to Marc Levoy and Pat Hanrahan for discussions on issues related to the 5D to 4D simplification, the two-plane parameterization and the camera based aperture analog. Jim Kajiya and Tony DeRose provided a terrific sounding board throughout this project. The ray tracer used for the synthetic fruit bowl was written by John Snyder. The mesh simplification code used for the bunny was writtenby HuguesHoppe. Portions ofthecameracapturecodewereimplemented by Matthew Turk. Jim Blinn, Hugues Hoppe, Andrew Glassner and Jutta Joesch provided excellent editing suggestions. Erynn Ryan is deeply thanked for her creative crisis management. Finally, we wish to thank the anonymousreviewers who pointed us toward a number of significant references we had missed.

# References

[1] ADELSON,E. H., AND BERGEN,J. R. Theplenopticfunctionand the elements of early vision. InComputationalModels ofVisualProcessing,LandyandMovshon, Eds. MIT Press, Cambridge,Massachusetts, 1991,ch. 1.

[2] ASHDOWN, I. Near-field photometry: A new approach. Journal of the Illumination Engineering Society 22, 1 (1993),163–180.

[3] BENTON, S. A. Survey of holographic stereograms. Proceedings of the SPIE 391 (1982),15–22.

[4] BOLLES, R. C., BAKER, H. H., AND MARIMONT, D. H. Epipolar-plane image analysis: An approach to determining structure from motion. International Journal of Computer Vision 1 (1987),7–55.

[5] BURT, P. J. Moment images, polynomial fit filters, and the problem of surface interpolation. In Proceedingsof Computer Vision and Pattern Recognition(June 1988),IEEE Computer Society Press, pp. 144–152.

[6] CHEN, S. E. Quicktime VR - an image-based approach to virtual environment navigation. In Computer Graphics, AnnualConference Series, 1995, pp. 29–38.

[7] CHEN, S. E., AND WILLIAMS, L. View interpolation for image synthesis. In Computer Graphics, AnnualConference Series, 1993, pp. 279–288.

[8] CHUI, C. K. An Introduction to Wavelets. Academic Press Inc., 1992.

[9] HALLE, M. W. Holographicstereogramsas discrete imagingsystems. Practical HolographyVIII (SPIE) 2176 (1994),73–84.

[10] HOPPE, H. Progressive meshes. In Computer Graphics, Annual Conference Series, 1996.

[11] KATAYAMA, A., TANAKA, K., OSHINO, T., AND TAMURA, H. A viewpoint independentstereoscopic display using interpolation of multi-viewpointimages. Steroscopic displays and virtal reality sytems II (SPIE) 2409 (1995),11–20.

[12] KLINKER, G. J. A Physical Approach to Color Image Understanding. A K

Peters, Wellesley, Massachusetts, 1993.

[13] LAURENTINI,A. Thevisualhullconceptforsilhouette-basedimageunderstanding. IEEETransactionsonPattern AnalysisandMachineIntelligence16,2 (February 1994),150–162.

[14] LAVEAU, S., AND FAUGERAS, O. 3-D scene representation as a collection of images and fundamental matrices. Tech. Rep. 2205, INRIA-Sophia Antipolis, February 1994.

[15] LEVIN,R. E. Photometriccharacteristicsof light-controllingapparatus. Illuminating Engineering 66, 4 (1971),205–215.

[16] LEVOY,M., AND HANRAHAN,P. Light-fieldrendering.In ComputerGraphics, AnnualConference Series, 1996.

[17] LEWIS, R. R., AND FOURNIER, A. Light-driven global illumination with a waveletrepresentationof lighttransport. UBC CS TechnicalReports95-28,University of British Columbia, 1995.

[18] LITWINOWICZ, P., AND WILLIAMS, L. Animating images with drawings. In Computer Graphics, AnnualConference Series, 1994, pp. 409–412.

[19] MCMILLAN,L., AND BISHOP,G. Plenopticmodeling: An image-basedrendering system. In Computer Graphics, AnnualConference Series, 1995, pp. 39–46.

[20] MITCHELL,D. P. Generatingantialiasedimagesatlowsamplingdensities. Computer Graphics 21, 4 (1987),65–72.

[21] POTMESIL, M. Generating octree models of 3D objects from their silhouettes in a sequence of images. Computer Vision, Graphics, and Image Processing 40 (1987),1–29.

[22] POWELL,M. J. D., AND SWANN,J. Weighteduniformsampling - a montecarlo techniquefor reducing variance. J. Inst. Maths Applics 2 (1966),228–236.

[23] ROSENFELD,A., AND KAK,A. C. DigitalPictureProcessing. AcademicPress, New York, New York, 1976.

[24] SIMONCELLI, E. P., FREEMAN, W. T., ADELSON, E. H., AND HEEGER, D. J. Shiftable multiscale transforms. IEEE Transactions on Information Theory 38 (1992),587–607.

[25] SNYDER, J. M., AND KAJIYA, J. T. Generative modeling: A symbolic system for geometric modeling. Computer Graphics 26, 2 (1992),369–379.

[26] SZELISKI, R. Rapid octree construction from image sequences. CVGIP: Image Understanding 58, 1 (July 1993), 23–32.

[27] TAUBIN, G. A signal processing approach to fair surface design. In Computer Graphics, AnnualConference Series, 1995,pp. 351–358.

[28] TERZOPOULOS, D. Regularizationof inverse visualproblemsinvolvingdiscontinuities. IEEE PAMI 8, 4 (July 1986),413–424.

[29] TSAI, R. Y. A versatile camera calibration technique for high-accuracy3D machinevision metrologyusing off-the-shelfTV camerasand lenses. IEEE Journal of Robotics and Automation RA-3, 4 (August1987), 323–344.

[30] WERNER, T., HERSCH, R. D., AND HLAVAC, V. Rendering real-world objects using view interpolation. In Fifth InternationalConference on Computer Vision (ICCV&apos;95) (Cambridge,Massachusetts, June 1995),pp. 957–962.

[31] WILLSON, R. G. Modeling and Calibration of Automated Zoom Lenses. PhD thesis, Carnegie Mellon University, 1994.