*出现在96年*

# 光场渲染

Marc Levo y和Pat Hanrahan

计算机科学系

斯坦福大学

## 摘要

许多技术已经被提出通过重新显示先前渲染或数字化的视图来穿越场景。也有人提出了通过扭曲输入图像，利用深度信息或多个图像之间的对应关系在视图之间进行插值的技术。本文描述了一种简单而稳健的方法，通过对现有的图像进行合并和重采样，在没有深度信息或特征匹配的情况下，从任意相机位置生成新的视图。这项技术的关键在于将输入图像解释为4D函数（光场）的2D切片。此函数完全描述了在固定照明的静态场景中，光通过无障碍空间的流动。

我们描述了一个光场的采样表示，它允许高效地创建和显示向内和向外的视图。我们已经从渲染和数字化图像的大阵列中创建了光场。后者是通过安装在计算机控制的机架上的摄像机获得的。一旦创建了光场，就可以通过提取适当方向的切片来实时构建新视图。由于该方法的成功依赖于高采样率，我们描述了一个压缩系统，该系统能够将我们生成的光场压缩到100:1以上，而保真度损失很小。我们还解决了创建过程中的抗锯齿问题，以及切片提取过程中的重采样问题。

**CR类别：**一、 3.2[计算机图形学]：图像/图像生成-数字化和扫描、查看算法；I.4.2[计算机图形学]：压缩-近似方法

**其他关键字：**基于图像的绘制，光场，全息立体图，矢量量化，极线分析

## 1介绍

传统上，三维图形系统的输入是由不同材质的几何图元和一组灯光组成的场景。基于此输入规范，渲染系统计算并输出图像。最近出现了一种新的绘制方法：基于图像的绘制。基于图像的渲染系统从一组预先获取的图像生成环境的不同视图。这种方法有几个优点：

|  |
| --- |
| 地址：盖茨计算机科学大厦3B层邮箱：y@cs.stanford.edu斯坦福大学hanrahan@cs.stanford.edu加利福尼亚州斯坦福，94305http://www-graphics.stanford.edu |

•基于图像绘制的显示算法需要大量的计算资源，因此适合在工作站和个人计算机上实时实现。•交互式观看场景的成本与场景复杂性无关。

•预采集图像的来源可以来自真实或虚拟环境，即数字化照片或渲染模型。事实上，两者可以混合在一起。

这些技术的先行者是使用环境贴图捕捉纹理贴图中的入射光[Blinn76，Greene86]。环境地图记录从各个方向到达某一点的入射光。环境贴图最初的用途是有效地近似环境在曲面上的反射。但是，环境地图也可以用来快速显示从固定位置到可变方向的任何向外看的环境视图。这是苹果QuickTimeVR系统的基础[Chen95]。在这个系统环境中，地图是在场景中的关键位置创建的。用户能够分散地从一个位置导航到另一个位置，并且在每个位置不断地改变观察方向。

基于环境地图的绘制系统的主要缺陷是视点是固定的。放松固定位置约束的一种方法是使用视图插值[Chen93，Greene94，Fuchs94，McMillan95a，McMillan95b，Narayann95]。这些方法中的大多数都需要环境地图中每个像素的深度值，如果环境贴图是合成图像，则很容易提供该值。给定深度值，可以从不同的有利位置重新投影环境地图中的点，以便在多个图像之间扭曲。这种扭曲方法的关键挑战是，当先前被遮挡的区域变得可见时，如何“填补空白”。

在采集到的图像之间进行插值的另一种方法是在两幅图像中找到对应的点[Laveau94，McMillan95b，Seitz95]。如果摄像机的位置已知，这相当于找到相应点的深度值。自动寻找图像对之间的对应关系是立体视觉的经典问题，不幸的是，尽管存在许多算法，但这些算法相当脆弱，不一定总能找到正确的对应关系。

在本文中，我们提出了一种新的技术，它是健壮的和

允许在可能的视图范围内有更大的自由度。光在空间中的自由位置是一种主要的表现手法。在自由空间中，光场是一个4D函数，而不是5D函数。图像是4D光场的二维切片。从一组图像创建光场对应于将每个2D切片插入4D光场表示。类似地，生成新视图对应于提取和重新采样切片。

从光场生成新图像与以前的视图插值方法有很大不同。首先，新图像通常由原始输入图像的许多不同片段组成，并且不需要看起来像其中任何一个。其次，不需要模型信息，如深度值或图像对应关系来提取图像值。第三，图像生成只涉及重采样，一个简单的线性过程。

光场的这种表示类似于计算机视觉中使用的极线体积[Bolles87]和仅水平视差的全息立体图[Benton83]。外极体是由一组图像组成的，这些图像是通过在一个方向上以相等的增量平移摄影机而创建的。这种表示最近被用于执行视图插值[Katayama95]。全息立体图是通过将一片胶片暴露在由侧向移动的照相机捕捉到的一系列图像中而形成的。Halle已经讨论了如何设置相机光圈来正确地获取全息立体图的图像[Halle94]，这个理论也适用于这项工作。加文·米勒也认识到了真实3D显示技术和计算机图形算法之间的潜在协同作用[Miller95]。

使用光场方法在图形工作站上查看三维场景有几个主要的挑战。首先，选择光场的参数化和表示。与此相关的是野外采样模式的选择。第二个问题是如何产生或获得光场。第三，存在快速生成不同观点的问题。这就要求代表光线通过一个点的切片很容易提取，并且切片被正确地重采样以避免最终图像中出现伪影。第四，这种方法的明显缺点是可能需要大量的数据。人们凭直觉怀疑光场是相干的，并且可能被大大压缩。在剩下的部分中，我们将讨论这些问题和我们提出的解决方案。

## 2代表

我们将光场定义为给定方向上某一点的辐射。注意，我们的定义等同于阿德尔森和伯根[Adelson91]引入的全光函数。术语“光场”是由A.Gershun在其经典论文《描述空间中光的辐射特性》[Gershun36]中提出的。[1]McMillan和Bishop[McMillan95b]讨论了将5D光场表示为不同3D位置的全景图像集。

然而，在自由空间（没有遮挡物的区域）中，5D表示可以被减少到4D。这是一个事实的结果，即辐射不会沿一条线变化，除非阻塞。4D光场可以解释为定向线空间上的函数。5D表示的冗余是不可取的，原因有两个：第一，冗余增加了整个数据集的大小；第二，冗余使得从样本中重建辐射函数变得复杂。这种尺寸的减少被用来简化灯具辐射的表示[Levin71，Ashdown93]。在本文的剩余部分，我们将只关注4D光场。

虽然将表示的有效性限制在自由空间似乎是一种限制，但有两种常见的情况下这种假设是有用的。首先，大多数几何模型是有界的。在这种情况下，自由空间是物体凸包外的区域，因此从其凸包外部观察物体的所有视图都可以从4D光场生成。第二，如果我们在建筑模型或室外场景中移动，我们通常是在一个自由空间区域中移动；因此，从该区域内部、区域外对象的任何视图都可能生成。

选择4D光场表示的主要问题是如何参数化有向线的空间。选择参数化有几个问题：

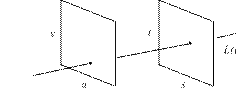
**高效计算。**根据参数计算直线的位置应该很快。更重要的是，为了计算新视图，在给定观察变换和像素位置的情况下，计算线参数应该很容易。

**控制一组线路。**所有线的空间是无限的，但只需要一个有限的线空间子集。例如，在观察一个对象的情况下，我们只需要与对象的凸包体相交的线。因此，在3空间中的实际线和线参数之间应该有一个直观的连接。

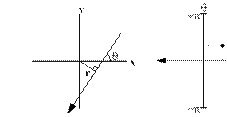
**均匀取样。**在线参数空间中给定等间距的样本，3空间中的线的图案也应该是均匀的。从这个意义上讲，一个统一的采样模式是一个采样间隔中的线数在任何地方都是恒定的。请注意，行数的正确度量与form factor内核[Sbert93]有关。

我们提出的解决方案是通过直线与任意位置的两个平面的交点来参数化直线（见图1）。按照惯例，第一个平面上的坐标系是（u，v），第二个平面上的坐标系是（s，t）。定向线是通过将uv平面上的一个点连接到st平面上的点来定义的。实际上，我们将u、v、s和t限制在0和1之间，因此每个平面上的点都被限制在凸四边形内。我们称之为轻板。直观地说，光板表示光束进入一个四边形并离开另一个四边形。

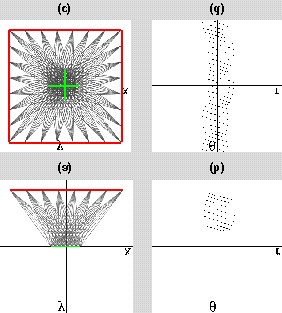
这种表示法的一个很好的特点是其中一个平面可以放置在无穷远处。这很方便，因为可以通过点和方向参数化直线。后者将证明有助于从正交图像或具有固定视野的图像构建光场。此外，如果所有计算都使用齐次坐标进行，则这两种情况可在不增加成本的情况下处理。



**图1：**轻板表示。



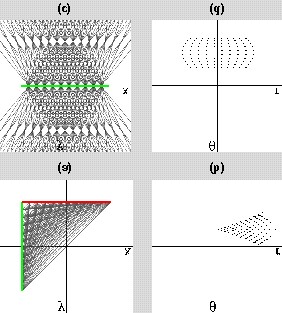
**图2：**用于可视化光线集的线空间的定义。笛卡尔空间（左侧）中的每条定向直线在直线空间（右侧）中由一个点表示。为了简化可视化，我们只显示二维的线条；三维的扩展很简单。



**图3：**使用线空间可视化光线覆盖。（a） 显示了一个光板。光线（以灰色绘制）连接两条定义线（以红色和绿色绘制）上的点。（c） 显示四个副本的排列（旋转）。（b） 和（d）显示相应的线空间可视化。对于笛卡尔空间中的任何一组线，由线空间中相应点形成的包络线表示我们对位置和方向的覆盖；理想情况下，覆盖范围应在r内尽可能宽。如图所示，（a）中的单个板不会提供完全覆盖，但（c）中的四块板的排列方式确实如此。（c） 然而，在r中是窄的。这样的排列适合于放置在原点的小物体向内看的视图。它用于生成图14d中的狮子光场。



这种表示法的一大优点是几何计算的效率。从（u，v）到平面上的点的映射是一个投影映射，只涉及线性代数（乘以3x3矩阵）。更重要的是，如将在第5节中讨论的，从图像像素（x，y）到（u，v，s，t）的逆映射也是投影映射。使用球坐标或柱坐标的方法需要大量的计算。



**图4：**利用线空间可视化采样均匀性。（a） 显示由两条直角线定义的轻型板。（c） 显示了一个轻型板，其中一条定义线位于无穷远处。这种排列产生的光线以-45°和+45°的角度穿过另一条定义线。（b） 和（d）显示相应的线空间可视化。用（r，）参数化线空间具有线空间面积相等对应于笛卡尔空间中位置和方向的等密度采样的性质，理想情况下线空间中点的密度应该是均匀的。如这些图所示，（a）中角点处的奇异性导致高度不均匀，因此采样模式效率低下，由（b）中0和−/2角度处的暗区表示。（c） 生成一组更统一的线。虽然（c）没有提供完整的覆盖范围，但有四个轮换副本。这种安排适合站在原点附近的观测者向外看。它用于生成图14c中的走廊灯光场。



光场的许多特性在线空间中更容易理解（图2到图4）。在线空间中，每一条定向线用一个点表示，每一组线用一个区域表示。特别地，由一个轻板表示的线集和与一个对象的凸壳相交的线集都是线空间中的区域。一个物体的所有视图可以从一个光板生成，如果它的一组线包括所有与物体的凸面相交的线。不幸的是，这是不可能的。因此，需要多个光板来表示对象的所有可能视图。因此，我们使用一组轻板来平铺线条空间，如图3所示。

与参数化相关的一个重要问题是采样模式。假设生成所有视图的可能性相等，则需要任何一行。因此，线空间的所有区域都应该具有相同密度的样本。图4显示了不同板坯排列的行空间中的样本密度。请注意，没有一个板的排列是完美的：具有奇点的排列（例如在一个角（4a）连接的两个多边形）是不好的，应该避免，而由平行平面（3a）形成的板生成相当均匀的图案。另外，一个平面在无穷远（4c）处的布置比具有两个有限平面（3a）的布置更好。最后，由于对称性，uv中采样的间距通常应与st相同。但是，如果观察者可能站在uv平面附近，则可以接受比st频率更低的uv采样频率。

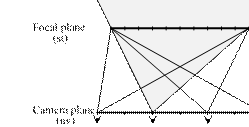
## 三。光场的创建

在本节中，我们将讨论虚拟光场（从渲染图像）和真实光场（从数字化图像）的创建。创建光场的一种方法是选择4D采样模式，并为每个线采样找到辐射度。这很容易通过光线跟踪器直接在虚拟环境中完成。这也可以用spot辐射计在真实环境中完成，但这将是非常乏味的。生成光场的一种更实际的方法是组装一组图像。

### 3.1条。从渲染图像

对于虚拟环境，只需渲染二维图像阵列就可以轻松生成光板。每个图像代表一个固定uv值的4D光板的一部分，并通过将虚拟摄影机的投影中心放置在样本上而形成

视野



**图5：**用于从透视图像阵列创建灯光板的查看几何体。

|  |
| --- |
| **图6：**光场的可视化。（a） 阵列中的每个图像表示从st平面上的所有点到达uv平面上一个点的光线，如左图所示。（b） 每个图像表示在st平面上保留一个点以绑定uv平面上所有点的光线。（a）中的图像是场景的离轴（即剪切）透视图，而（b）中的图像看起来像反射贴图。后一种情况发生是因为对象被跨在焦平面上，使得在焦平面上留下点的光线集与在对象上留下点的光线集的特性相似。 |

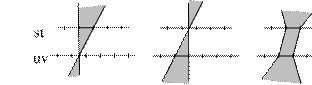
在uv平面上的位置。唯一的问题是每个图像的xy采样必须与st个采样完全对应。这很容易通过执行剪切透视投影（图5）来实现，该投影与生成立体图像对的方法类似。图6显示了生成的4D光场，可以将其可视化为st图像的uv数组或uv图像的st数组。

其他观察几何图形也很有用。光板可以由二维正交视图阵列形成。这可以通过将uv平面放置在无穷远的位置来建模，如图4c所示。在这种情况下，每个uv样本对应于一个平行投影的方向。同样，唯一的问题是将图像的xy和st样本与st四边形对齐。另一个有用的几何体包括一个具有固定视野的向外（非剪切）透视图的二维阵列。在这种情况下，每个图像都是一个光板的切片，其中st平面在无穷远处。事实上，所有这些情况都同样容易处理轻板证明了射影几何的优雅。使用每种布置的光场如第6节所示，如图14所示。

与任何采样过程一样，对光场进行采样可能会导致混叠，因为典型的光场包含高频率。幸运的是，混叠的影响可以通过在采样前进行过滤来减轻。在光场的情况下，必须在行空间中使用4D滤波器（见图7）。假设一个盒子过滤器，必须计算uv和st平面中连接样本方块的所有线上辐射的加权平均值。如果将相机放置在uv平面上并聚焦在st平面上，则过滤过程对应于在对应于st样本的像素上和与uv样本大小相等的光圈上对两者进行积分，如图8所示。哈雷[Halle94]在全息立体图的背景下讨论了这种滤波过程背后的理论。

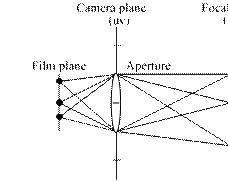
请注意，尽管预过滤具有对光场进行抗锯齿的理想效果，但它最初似乎有一个不受欢迎的副作用-由于景深而导致模糊。然而，这种模糊性恰恰适合这种情况。

回想一下从uv平面上的两个相邻摄影机位置创建一对图像时发生的情况：给定的对象点将投影到这两个图像中的不同位置，可能相距数个像素。两个投影位置之间的距离称为立体视差。将这一思想扩展到多个相机位置会产生一系列图像，其中对象看起来跳跃的距离等于视差。这种跳跃是假动作。现在回想一下，用有限光圈拍摄图像会导致失焦点在胶片平面上因一圈混乱而变得模糊。将光圈直径设置为相机位置之间的间距会导致每个对象点的混淆圆大小与立体视差相等。这将用一系列模糊图像代替跳跃。因此，我们使用有限景深来消除混叠。



像素过滤器+光圈过滤器=光线过滤器

**图7：**预先过滤光场。为了避免混叠，必须对radiance函数应用4D低通滤波器。



**图8：**使用孔径进行预过滤。此图显示聚焦在st平面上的摄影机，其uv平面上的光圈大小等于uv采样间隔。在光圈后面画一个假设的胶片平面。暂时忽略光圈（考虑一个针孔相机，它可以精确地将st平面成像到胶片平面上）。然后在胶片平面上的像素上积分相当于在由像素限定的st区域上积分。现在考虑在胶片平面上固定一个点，同时使用有限大小的光圈（回想一下，从胶片上的一个点通过光圈的所有光线都聚焦在焦平面上的一个点上）。然后在光圈上积分相当于将所有光线通过光圈限定的uv区域进行积分。因此，通过同时对像素和光圈进行积分，可以计算出适当的4D积分。

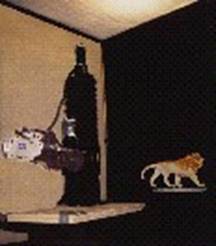


预过滤的必要性也可以在行空间中理解。回想一下我们之前的讨论，光场的样本对应于线空间中的点。有一个有限的景深，孔径大小等于紫外采样间距，可以确保每个采样充分覆盖这些线空间点之间的间隔。过小或过大的光圈会在线空间覆盖范围内产生间隙或重叠，分别导致视图出现锯齿或过度模糊。

### 3.2条。从数字化图像

将建立物理场景光场所需的图像数字化是一个艰巨的工程问题。所需的图像数量很大（成百上千），因此这个过程必须是自动化的，或者至少是计算机辅助的。此外，必须控制照明，以确保静态光场，但要足够灵活，以适当照亮场景，同时始终远离摄影机以避免不必要的阴影。最后，实际的光学系统会对视角、焦距、景深和光圈施加限制，所有这些都必须加以管理。在灯具近场光度测量装置的制造中也遇到了类似的问题[Ashdown93]。在下面的段落中，我们列举了我们在这项工作中面临的主要设计决策以及我们采用的解决方案。

**内向与外向。**要做的第一个决定是在一个小物体绕飞和一个大尺度场景的飞越之间。我们认为绕飞区比较简单，所以我们先攻击它们。



**图9：**我们的龙门相机原型。修改过的网络软件

MS运动平台配有来自LinTech和Parker的额外步进电机，提供四个自由度：水平和垂直平移、平移和倾斜。这款相机是一款松下WV-F300 3-CCD摄像机，采用佳能f/1.7 10-120mm变焦镜头。我们保持它锁定在最宽的设置（10毫米）和安装，使俯仰和偏航轴通过投影中心。数字化时，相机保持指向焦平面的中心。校准和校准通过法罗数字化臂进行验证，精确到0.3 mm。

**人工控制与计算机控制。**数字化光场的一种廉价的方法是在场景中移动手持摄像机，从生成的图像填充光场[Gortler96]。这种方法需要估计相机在每一帧的姿态和从散射数据插值光场-这是两个具有挑战性的问题。为了简化这种情况，我们选择建立一个计算机控制的相机机架，并在一个常规网格上数字化图像。

**球面与平面摄像机运动。**对于小物体的绕飞，一个明显的机架设计包括两个同心半圆形，类似于陀螺仪安装。这种机架中的摄像机沿球面移动，始终指向球体的中心。苹果电脑公司（Apple Computer）建造了这样一个机架，用于快速虚拟现实飞行（Quick Time VR FlyAround）获取图像[Chen95]。不幸的是，他们的系统中的照明是附加在移动摄像机上的，所以不适合获取静态光场。一般来说，球形机架比平面机架有三个优点：（a）更容易覆盖整个观察方向范围；（b）方向空间中的采样率更均匀；（c）相机和物体之间的距离是固定的，在整个相机运动范围内提供更清晰的焦点。与球形门架相比，平面门架有两个优点：（A）它更容易建造；整个结构可以从线性运动阶段组装而成；（b）它更接近于我们的轻型平板表示法。对于我们的第一个机架原型，我们选择构建一个平面机架，如图9所示。

**视野。**我们的目标是建立一个可以360度全方位观看的光场。要实现这一点，使用一个平面龙门架意味着获得四块板，每个板提供90块

#### 灯

**图10：**物体和照明支持。物体安装在一个博根流体头三脚架上，我们手动旋转到四个方向，间隔90度。照明由两个600W的Lowell Omni聚光灯提供，该聚光灯安装在天花板上，与三脚架的旋转轴对齐。一个固定的6&apos;x 6&apos;扩散器面板悬挂在聚光灯和机架之间，整个设备用黑色天鹅绒封闭，以消除杂散光。

度。这可以通过使用广角镜头进行平移但不平移或倾斜的相机来实现。此解决方案有两个缺点：（a）广角镜头显示出明显的失真，必须在采集后进行校正；（b）此解决方案在视角与传感器分辨率之间进行权衡。另一个解决方案是使用一个视图摄像机，其中传感器和光学系统在平行平面上平移，前者的移动速度比后者快。水平视差全息立体图是用这样一个相机来制作的[Halle94]。将此解决方案整合到水平和垂直移动的机架中是困难的。取而代之的是我们的相机配备了摇摄和倾斜马达，使我们能够使用窄角镜头。旋转相机的使用意味着，为了将采集到的图像转换为光板表示，必须将其重新投影到一个公共平面上。这种重投影相当于建筑摄影中的keystone校正。

**防区外距离。**平面门架的一个缺点是，随着摄影机在平面上的平移，从摄影机到对象的距离会发生变化，因此很难保持对象的焦点。上面描述的视图摄影机不存在此问题，因为当摄影机平移时，对象距离与图像距离的比率保持不变。对于旋转相机来说，伺服控制聚焦是一种选择，但是改变相机的焦距会改变它的投影中心并改变图像放大率，使采集变得复杂。相反，我们通过使用强光和小光圈来最大化景深来缓解这个问题。

**传感器旋转。**光板中的每个样本都应该理想地代表一个像素上的积分，这些像素应该位于一个共同的焦平面上。它的约束满足摄影机在平面上的平移。我们使用旋转相机意味着焦平面也会旋转。假设我们在重投影过程中仔细地重新取样，旋转焦平面的存在不会给光场带来额外的误差。实际上，我们还没有看到由于这个重采样过程而产生的工件。

**光圈尺寸。**光板中的每个样本也应表示与uv样本大小相等的光圈上的积分。我们使用一个小光圈产生的光场很少或没有紫外线抗锯齿。即使完全打开，商业摄像机的光圈也很小。我们可以通过平均一些相邻视图的数量来近似所需的抗锯齿，从而创建一个合成光圈。然而，这种技术需要非常密集的视图间距，而这又需要快速获取。我们目前没有这样做。

**对象支持。**为了使用平面机架在四个90度段中获取360度光场，机架或对象必须旋转到间隔90度的四个方向中的每一个。考虑到我们的机架庞大，后者显然更容易。对于这些实验，我们把我们的物体安装在一个三脚架上，我们手动旋转到四个位置，如图10所示。

**照明。**如果我们决定旋转对象，满足固定照明的要求意味着照明必须具有四倍对称性，或者必须随对象旋转。我们选择了后一种解决方案，将照明系统连接到旋转的轮毂上，如图10所示。设计一个既能远离机架，又能提供足够的光线均匀照亮物体的照明系统，是一个具有挑战性的问题。

使用这个机架，我们获得光场的过程如下。对于四个方向中的每一个，摄影机都通过摄影机位置的规则栅格进行平移。在每一个位置，相机被平移并倾斜到对象的中心，该中心位于三脚架的旋转轴上。然后我们获得一个图像，并使用标准纹理映射算法，将其重新投影到前面所述的公共平面上。表二给出了一组典型的采集参数。请注意，相机位置之间的距离（3.125厘米）超过了光圈的直径（1.25毫米），强调需要更密集的间距和合成孔径。

## 4压缩

光场阵列很大-本文中最大的例子是1.6gb。为了使光场的创建、传输和显示实用化，必须对其进行压缩。在众多可用压缩技术中进行选择时，我们遵循光场的几个独特特征：

**数据冗余。**一种好的压缩技术可以在不影响信号内容的情况下消除信号中的冗余。光场在所有四个维度上都表现出冗余。例如，图6a中的平滑区域告诉我们，该光场在s和t中包含冗余，而图6b中的平滑区域告诉我们光场在u和v中包含冗余。前者对应于透视图中通常的像素间一致性概念。后者可以解释为运动序列中的帧间一致性，也可以解释为漫反射或中等镜面反射表面的双向反射分布函数（BRDF）中的平滑度。当然，在这两种情况下，咬合都会产生不连续性。

**随机存取。**大多数压缩技术对数据的随机访问有一定的限制。例如，可变比特率编码器可能要求扫描线、分幅或帧一次解码。这类中的例子是可变比特率矢量量化和JPEG或MPEG中使用的Huffman或算术编码器。由于像素依赖于先前解码的像素、扫描线或帧，预测编码方案进一步使随机存取复杂化。这给光场带来了一个问题，因为从光场提取图像时引用的样本集分散在内存中。随着观察者的移动，访问模式以复杂的方式变化。因此，我们寻求一种支持对单个样本进行低成本随机访问的压缩技术。

**不对称。**根据编码和解码所花费的相对时间，压缩应用可分为对称或非对称。我们假设光场被提前组装和压缩，使得这是一个不对称的应用。

**计算费用。**我们寻求一种无需硬件辅助就能解码的压缩方案。尽管软件解码器已经被证明适用于JPEG和MPEG等标准，但这些实现消耗了现代微处理器的全部功能。除解压缩外，显示算法还需要执行额外的工作，如第5节所述。因此，我们寻求一种能够快速解码的压缩方案。

我们选择的压缩方案是一个两级流水线，由固定速率矢量量化和熵编码（Lempel-Ziv）组成，如图11所示。根据相似的动机，比尔斯等人。使用矢量量化来压缩纹理以用于渲染管道[Beers96]。

### 4.1条。矢量量化

我们的压缩管道的第一阶段是矢量量化（VQ）[Gersho92]，这是一种有损压缩技术，其中样本向量被量化为多个预定再现向量中的一个。复制向量称为码字，可用于编码源的码字集称为码本，码书是在训练阶段构造的，在训练阶段，量化器被要求找到一组最接近一组样本向量的码字，称为训练集。码字的质量是典型的特征

码本LZ

（0.8 MB）

光场比特流

（402 MB）（3.4 MB）

                                             指数LZ

（16.7 MB）

**图11**Tw o级压缩管道。光场被分割成块，然后使用矢量量化对其进行编码以形成码书索引阵列。使用Lempel-Ziv编码进一步压缩码本和索引数组。解压也分为两个阶段：当文件加载到内存中时进行熵解码，以及在交互式观看时按需去量化。典型的文件大小显示在每个阶段旁边。

使用均方误差（MSE），即源样本和码字样本之间平方差向量中所有样本的和。一旦一个码本被构造出来，编码就包括将源划分成向量，并为每个向量从码本中找到最接近的码字。解码包括在码本中查找索引并输出在那里找到的码字-这是一个非常快速的操作。实际上，解码速度是矢量量化的主要优点之一。

在我们的应用中，我们通常使用二维或四维光场，分别产生12维或48维向量。前者只利用了s和t的一致性，而后者则利用了所有四个维度的一致性。为了最大限度地提高图像质量，我们训练每个光场的代表性子集，然后将得到的码本与码字索引数组一起传输。由于光传输后的额外场比压缩后的光场要小，所以通常要少20%。我们在一个子集而不是整个光场上训练，以减少训练费用。

矢量量化的输出是一系列固定的码本索引。每个索引是logn位，其中N是码本中的码字数，因此量化器的压缩率为（kl）/（logn），其中k是每个向量的元素数（即维数），l是每个元素的比特数，通常为8。在我们的应用程序中，我们通常使用16384个字的码本，这一阶段的流水线压缩率为（48x8）/（log16384）=384位/14位=27:1。为了简化解码，我们使用整数字节表示每个索引，在我们的例子中是2，这将压缩比稍微降低到24:1。

### 4.2条。熵编码

我们的压缩管道的第二阶段是一个熵编码器，旨在降低表示高概率代码索引的成本。由于我们的对象通常是在恒定的颜色背景下进行渲染或拍照的，因此数组中包含了许多以高概率出现的分片。对于本文中的例子，我们使用了gzip，一种Lempel-Ziv编码的实现[Ziv77]。在该算法中，输入流被分割成不重叠的块，同时构造一个迄今为止看到的块的字典。将gzip应用到我们的代码索引数组中通常会给我们额外的5:1压缩。哈夫曼编码可能会产生稍微高一点的压缩，但编码和解码将更昂贵。因此，我们的总压缩比为24×5=120:1。有关压缩结果的更多详细信息，请参见第6节和表三。

### 4.3。减压

减压分为两个阶段。第一阶段-gzip解码-在文件加载到内存中时执行。这个阶段的输出是一个码本和一个压缩在16位字中的代码索引数组。虽然这种解码降低了一些效率，但是光场仍然是24:1压缩的，并且它现在以支持随机访问的方式表示。

第二阶段-去量化-进行如下。当观察者在场景中移动时，显示引擎请求光场的采样。每个请求由一个（u，v，s，t）坐标元组组成。对于每个请求，执行下标计算以确定要寻址的样本块。每个分片对应于一个量化向量，因此在索引数组中由单个条目表示。在代码本中查找这个索引，我们找到一个样本值的向量。然后执行第二个下标计算，给出所请求样本在向量内的偏移量。借助预先计算的下标表，可以非常有效地实现去量化。在我们的测试中，解压消耗大约25%的CPU周期。

## 5显示

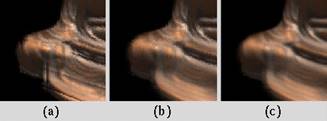
该系统的最后一部分是一个实时查看器，它根据给定的成像几何结构构造和显示来自光板的图像。观察者必须从4D光场中重新采样一段2D线；每一条线代表一条穿过眼点和像素中心的光线，如图12所示。该过程分为两个步骤：第一步包括计算每个图像光线的（u，v，s，t）线参数，第二步是对这些线参数处的辐射率进行重新采样。

As mentioned previously, a big advantage of the light slab representation is the efficiency of the inverse calculation of the line parameters. Conceptually the (u, v) and (s, t) parameters may be calculated by determining the point of intersection of an image ray with each plane. Thus, any ray tracer could easily be adapted to use light slabs. However, a polygonal rendering system also may be used to view a light slab. The transformation from image coordinates (x, y) to both the (u, v) and the (s, t) coordinates is a projective map. Therefore, computing the line coordinates can be done using texture mapping. The uv quadrilateral is drawn using the current viewing transformation, and during scan conversion the (uw, vw, w) coordinates at the corners of the quadrilateral are interpolated. The resulting u = /w and v = /w coordinates at each pixel represent the ray intersection with the uv quadrilateral. A similar procedure can be used to generate the (s, t) coordinates by drawing the st quadrilateral. Thus, the inverse transformation from (x, y) to (u, v, s, t) reduces essentially to two texture coordinate calculations per ray. This is cheap and can be done in real time, and is supported in many rendering systems, both hardware and software.*uwvw*

Only lines with (u, v) and (s, t) coordinates inside both quadrilaterals are represented in the light slab. Thus, if the texture coordinates for each plane are computed by drawing each quadrilaterial one after the other, then only those pixels that have both valid uv and st coordinates should be looked up in the light slab array. Alternatively, the two quadrilaterals may be simultaneously scan converted in their region of overlap to cut down on unnecessary calculations; this is the technique that we use in our software implementation.



**Figure 12:** The process of resampling a light slab during display.



**Figure 13:** The effects of interpolation during slice extraction. (a) No interpolation. (b) Linear interpolation in uv only. (c) Quadralinear interpolation in uvst.

To draw an image of a collection of light slabs, we draw them sequentially. If the sets of lines in the collection of light slabs do not overlap, then each pixel is drawn only once and so this is quite efficient. To further increase efficiency, "back-facing" light slabs may be culled.

The second step involves resampling the radiance. The ideal resampling process first reconstructs the function from the original samples, and then applies a bandpass filter to the reconstructed function to remove high frequencies that may cause aliasing. In our system, we approximate the resampling process by simply interpolating the 4D function from the nearest samples. This is correct only if the new sampling rate is greater than the original sampling rate, which is usually the case when displaying light fields. However, if the image of the light field is very small, then some form of prefiltering should be applied. This could easily be done with a 4D variation of the standard mipmapping algorithm [Williams83].

Figure 13 shows the effect of nearest neighbor versus bilinear interpolation on the uv plane versus quadrilinear interpolation of the full 4D function. Quadralinear interpolation coupled with the proper prefiltering generates images with few aliasing artifacts. The improvement is particularly dramatic when the object or camera is moving. However, quadralinear filtering is more expensive and can sometimes be avoided. For example, if the sampling rates in the uv and st planes are different, and then the benefits of filtering one plane may be greater than the other plane.

## 6. Results

Figure 14 shows images extracted from four light fields. The first is a buddha constructed from rendered images. The model is an irregular polygon mesh constructed from range data. The input images were generated using RenderMan, which also provided the machinery for computing pixel and aperture

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **buddha** | **kidney** | **hallway** | **lion** |
| Number of slabs | 1 | 1 | 4 | 4 |
| Images per slab | 16x16 | 64x64 | 64x32 | 32x16 |
| Total images | 256 | 4096 | 8192 | 2048 |
| Pixels per image | 2562 | 1282 | 2562 | 2562 |
| Raw size (MB) | 50 | 201 | 1608 | 402 |
| Prefiltering | uvst | st only | uvst | st only |

**Table I:** Statistics of the light fields shown in figure 14.

antialiasing. The light field configuration was a single slab similar to that shown in figure 3a.

Our second light field is a human abdomen constructed from volume renderings. The two tan-colored organs on either side of the spine are the kidneys. In this case, the input images were orthographic views, so we employed a slab with one plane at infinity as shown in figure 4c. Because an orthographic image contains rays of constant direction, we generated more input images than in the first example in order to provide the angular range needed for creating perspective views. The images include pixel antialiasing but no aperture antialiasing. However, the dense spacing of input images reduces aperture aliasing artifacts to a minimum.

Our third example is an outward-looking light field depicting a hallway in Berkeley&apos;s Soda Hall, rendered using a radiosity program. To allow a full range of observer motion while optimizing sampling uniformity, we used four slabs with one plane at infinity, a four-slab version of figure 4c. The input images were rendered on an SGI RealityEngine, using the accumulation buffer to provide both pixel and aperture antialiasing.

Our last example is a light field constructed from digitized images. The scene is of a toy lion, and the light field consists of four slabs as shown in figure 3c, allowing the observer to walk completely around the object. The sensor and optical system provide pixel antialiasing, but the aperture diameter was too small to provide correct aperture antialiasing. As a result, the light field exhibits some aliasing, which appears as double images. These artifacts are worst near the head and tail of the lion because of their greater distance from the axis around which the camera rotated.

Table I summarizes the statistics of each light field. Table II gives additional information on the lion dataset. Table III gives the performance of our compression pipeline on two representative datasets. The buddha was compressed using a 2D tiling of the

|  |  |
| --- | --- |
| **Camera motion**  translation per slab | 100 cm x 50 cm |
| pan and tilt per slab | 90° x 45° |
| number of slabs | 4 slabs 90° apart |
| total pan and tilt | 360° x 45° |
| **Sampling density** distance to object | 50 cm |
| camera pan per sample | 3.6° |
| camera translation per sample | 3.125 cm |
| **Aperture**  focal distance of lens | 10mm |
| F-number | f/8 |
| aperture diameter | 1.25 mm |
| **Acquisition time** time per image | 3 seconds |
| total acquisition time | 4 hours |

**Table II:** Acquisition parameters for the lion light field. Distance to object and camera pan per sample are given at the center of the plane of camera motion. Total acquisition time includes longer gantry movements at the end of each row and manual setup time for each of the four orientations. The aperture diameter is the focal length divided by the F-number.

|  |  |  |
| --- | --- | --- |
|  | **buddha** | **lion** |
| **Vector quantization** raw size (MB) | 50.3 | 402.7 |
| fraction in training set | 5% | 3% |
| samples per tile | 2x2x1x1 | 2x2x2x2 |
| bytes per sample | 3 | 3 |
| vector dimension | 12 | 48 |
| number of codewords | 8192 | 16384 |
| codebook size (MB) | 0.1 | 0.8 |
| bytes per codeword index | 2 | 2 |
| index array size (MB) | 8.4 | 16.8 |
| total size (MB) | 8.5 | 17.6 |
| compression rate | 6:1 | 23:1 |
| **Entropy coding** gzipped codebook (MB) | 0.1 | 0.6 |
| gzipped index array (MB) | 1.0 | 2.8 |
| total size (MB) | 1.1 | 3.4 |
| compression due to gzip | 8:1 | 5:1 |
| total compression | 45:1 | 118:1 |
| **Compression performance** training time | 15 mins | 4 hrs |
| encoding time | 1 mins | 8 mins |
| original entropy (bits/pixel) | 4.2 | 2.9 |
| image quality (PSNR) | 36 | 27 |

**Table III:** Compression statistics for two light fields. The buddha was compressed using 2D tiles of RGB pixels, forming 12-dimensional vectors, and the lion was compressed using 4D tiles (2D tiles of RGB pixels from each of 2 x 2 adjacent camera positions), forming 48-dimensional vectors. Bytes per codeword index include padding as described in section 4. Peak signal-to-noise ratio (PSNR) is computed as 10log10/MSE).(2552



light field, yielding a total compression rate of 45:1. The lion was compressed using a 4D tiling, yielding a higher compression rate of 118:1. During interactive viewing, the compressed buddha is indistinguishable from the original; the compressed lion exhibits some artifacts, but only at high magnifications. Representative images are shown in figure 15. We hav e also experimented with higher rates. As a general rule, the artifacts become objectionable only above 200:1.

Finally, table IV summarizes the performance of our interactive viewer operating on the lion light field. As the table shows, we achieve interactive playback rates for reasonable image sizes. Note that the size of the light field has no effect on playback rate; only the image size matters. Memory size is not an issue because the compressed fields are small.

## 7. Discussion and future work

We hav e described a new light field representation, the light slab, for storing all the radiance values in free space. Both inserting images into the field and extracting new views from the field involve resampling, a simple and robust procedure. The resulting system is easily implemented on workstations and personal computers, requiring modest amounts of memory and cycles. Thus, this technique is useful for many applications requiring interaction with 3D scenes.

|  |  |  |  |
| --- | --- | --- | --- |
| **Display times (ms)** coordinate calculation | **no bilerp** | **uv lerp** | **uvst lerp** |
| 13 | 13 | 13 |
| sample extraction | 14 | 59 | 214 |
| overhead total | 3 | 3 | 3 |
| 30 | 75 | 230 |

**Table IV:** Display performance for the lion light field. Displayed images are 192 x 192 pixels. Sample extraction includes VQ decoding and sample interpolation. Display overhead includes reading the mouse, computing the observer position, and copying the image to the frame buffer. Timings are for a software-only implementation on a 250 MHz MIPS 4400 processor.



There are three major limitation of our method. First, the sampling density must be high to avoid excessive blurriness. This requires rendering or acquiring a large number of images, which may take a long time and consume a lot of memory. Howev er, denser sample spacing leads to greater inter-sample coherence, so the size of the light field is usually manageable after compression. Second, the observer is restricted to regions of space free of occluders. This limitation can be addressed by stitching together multiple light fields based on a partition of the scene geometry into convex regions. If we augment light fields to include Zdepth, the regions need not even be convex. Third, the illumination must be fixed. If we ignore interreflections, this limitation can be addressed by augmenting light fields to include surface normals and optical properties. To handle interreflections, we might try representing illumination as a superposition of basis functions [Nimeroff94]. This would correspond in our case to computing a sum of light fields each lit with a different illumination function.

It is useful to compare this approach with depth-based or correspondence-based view interpolation. In these systems, a 3D model is created to improve quality of the interpolation and hence decrease the number of pre-acquired images. In our approach, a much larger number of images is acquired, and at first this seems like a disadvantage. However, because of the 3D structure of the light field, simple compression schemes are able to find and exploit this same 3D structure. In our case, simple 4D block coding leads to compression rates of over 100:1. Given the success of the compression, a high density compressed light field has an advantage over other approaches because the resampling process is simpler, and no explicit 3D structure must be found or stored.

There are many representations for light used in computer graphics and computer vision, for example, images, shadow and environment maps, light sources, radiosity and radiance basis functions, and ray tracing procedures. However, abstract light representations have not been systematically studied in the same way as modeling and display primitives. A fruitful line of future research would be to reexamine these representations from first principles. Such reexaminations may in turn lead to new methods for the central problems in these fields.

Another area of future research is the design of instrumentation for acquisition. A large parallel array of cameras connected to a parallel computer could be built to acquire and compress a light field in real time. In the short term, there are many interesting engineering issues in designing and building gantries to move a small number of cameras and lights to sequentially acquire both inward- and outward-looking light fields. This same instrumentation could lead to breakthroughs in both 3D shape acquisition and reflection measurements. In fact, the interaction of light with any object can be represented as a higher-dimensional interaction matrix; acquiring, compressing, and manipulating such representations are a fruitful area for investigation.

## 8. Acknowledgements

We would like to thank David Addleman and George Dabrowski of Cyberware for helping us design and build the camera gantry, Craig Kolb and James Davis for helping us calibrate it, Brian Curless for scanning the buddha, Julie Dorsey for shading it and allowing us to use it, Carlo Sequin for the Soda Hall model, Seth Teller, Celeste Fowler, and Thomas Funkhauser for its radiosity solution, Lucas Pereira for rendering it, Benjamin Zhu for reimplementing our hardware-accelerated viewer in software, and Navin Chaddha for his vector quantization code. We also wish to thank Eric Chen and Michael Chen for allowing us to examine the Apple ObjectMaker, and Alain Fournier and Bob Lewis for showing us their wav elet light field work. Finally, we wish to thank Nina Amenta for sparking our interest in two-plane parameterizations of lines, Michael Cohen for reinforcing our interest in image-based representations, and Gavin Miller for inspiring us with his grail of volumetric hyperreality. This work was supported by the NSF under contracts CCR-9157767 and CCR-9508579.

## 9. References

[Adelson91] Adelson, E.H., Bergen, J.R., &apos;&apos;The Plenoptic Function and the Elements of Early Vision,&apos;&apos; In Computation Models of Visual Processing, M. Landy and J.A. Movshon, eds., MIT Press, Cambridge, 1991.

[Ashdown93] Ashdown, I., &apos;&apos;Near-Field Photometry: A New Approach,&apos;&apos; Journal of the Illuminating Engineering Society, Vol. 22, No. 1, Winter, 1993, pp. 163-180.

[Beers96] Beers, A., Agrawala, M., Chaddha, N., &apos;&apos;Rendering from Compressed Textures.&apos;&apos; In these proceedings.

[Benton83] Benton, S., &apos;&apos;Survey of Holographic Stereograms,&apos;&apos; Processing and Display of Three-Dimensional Data, Proc. SPIE, Vol. 367, 1983.

[Blinn76] Blinn, J.F., Newell, M.E., &apos;&apos;Texture and Reflection in Computer

Generated Images,&apos;&apos; CACM, Vol. 19, No. 10, October, 1976, pp.

542-547.

[Bolles87] Bolles, R., Baker, H., Marimont, D., &apos;&apos;Epipolar-Plane Image

Analysis: An Approach to Determining Structure from Motion,&apos;&apos; International Journal of Computer Vision, Vol. 1, No. 1, 1987, pp.

7-55.

[Chen93] Chen, S.E., Williams, L., &apos;&apos;View Interpolation for Image Synthesis,&apos;&apos; Proc. SIGGRAPH &apos;93 (Anaheim, California, August 1-6, 1993). In Computer Graphics Proceedings, Annual Conference Series, 1993, ACM SIGGRAPH, pp. 279-288.

[Chen95] Chen, S.E., &apos;&apos;QuickTime VR — An Image-Based Approach to Virtual Environment Navigation,&apos;&apos; Proc. SIGGRAPH &apos;95 (Los Angeles, CA, August 6-11, 1995). In Computer Graphics Proceedings, Annual Conference Series, 1995, ACM SIGGRAPH, pp. 29-38.

[Fuchs94] Fuchs, H., Bishop, G., Arthur, K., McMillan, L., Bajcsy, R.,

Lee, S.W., Farid, H., Kanade, T., &apos;&apos;Virtual Space Teleconferencing Using a Sea of Cameras,&apos;&apos; Proc. First International Conference on Medical Robotics and Computer Assisted Surgery, 1994, pp. 161-167.

[Gersho92] Gersho, A., Gray, R.M., Vector Quantization and Signal Compression, Kluwer Academic Publishers, 1992.

[Gershun36] Gershun, A., &apos;&apos;The Light Field,&apos;&apos; Moscow, 1936. Translated by P. Moon and G. Timoshenko in Journal of Mathematics and Physics, Vol. XVIII, MIT, 1939, pp. 51-151.

[Gortler96] Gortler, S.J., Grzeszczuk, R., Szeliski, R., Cohen, M., &apos;&apos;The Lumigraph.&apos;&apos; In these proceedings.

[Greene86] Greene, N., &apos;&apos;Environment Mapping and Other Applications of World Projections,&apos;&apos; IEEE Computer Graphics and Applications, Vol. 6, No. 11, November, 1986, pp. 21-29.

[Greene94] Greene, N. and Kass, M., &apos;&apos;Approximating Visibility with Environment Maps,&apos;&apos; Apple Technical Report No. 41, November, 1994.

[Halle94] Halle, M., &apos;&apos;Holographic Stereograms as Discrete Imaging Systems.&apos;&apos; Practical Holography, Proc. SPIE, Vol. 2176, February, 1994.

[Katayama95] Katayama, A., Tanaka, K., Oshino, T., Tamura, H., &apos;&apos;Viewpoint-Dependent Stereoscopic Display Using Interpolation of Multiviewpoint Images,&apos;&apos; Stereoscopic Displays and Virtual Reality Systems II, Proc. SPIE, Vol. 2409, S. Fisher, J. Merritt, B. Bolas eds.

1995, pp. 11-20.

[Laveau94] Laveau, S., Faugeras, O.D., &apos;&apos;3-D Scene Representation as a

Collection of Images and Fundamental Matrices,&apos;&apos; INRIA Technical Report No. 2205, 1994.

[Levin71] Levin, R., &apos;&apos;Photometric Characteristics of Light Controlling

Apparatus,&apos;&apos; Illuminating Engineering, Vol. 66, No. 4, 1971, pp.

205-215.

[McMillan95a] McMillan, L., Bishop, G., &apos;&apos;Head-Tracked Stereoscopic Display Using Image Warping,&apos;&apos; Stereoscopic Displays and Virtual Reality Systems II, Proc. SPIE, Vol. 2409, S. Fisher, J. Merritt, B.

Bolas eds. 1995, pp. 21-30.

[McMillan95b] McMillan, L., Bishop, G., Plenoptic Modeling: An Image-Based Rendering System, Proc. SIGGRAPH &apos;95 (Los Angeles, CA, August 6-11, 1995). In Computer Graphics Proceedings, Annual Conference Series, 1995, ACM SIGGRAPH, pp. 39-46.

[Miller95] Miller, G., &apos;&apos;Volumetric Hyper-Reality: A Computer Graphics Holy Grail for the 21st Century?,&apos;&apos; Proc. Graphics Interface &apos;95, W. Davis and P. Prusinkiewicz eds., Canadian Information Processing Society, 1995, pp. 56-64.

[Moon81] Moon, P., Spencer, D.E., The Photic Field, MIT Press, 1981.

[Narayanan95] Narayanan, P.J., &apos;&apos;Virtualized Reality: Concepts and Early Results,&apos;&apos; Proc. IEEE Workshop on the Representation of Visual Scenes, IEEE, 1995.

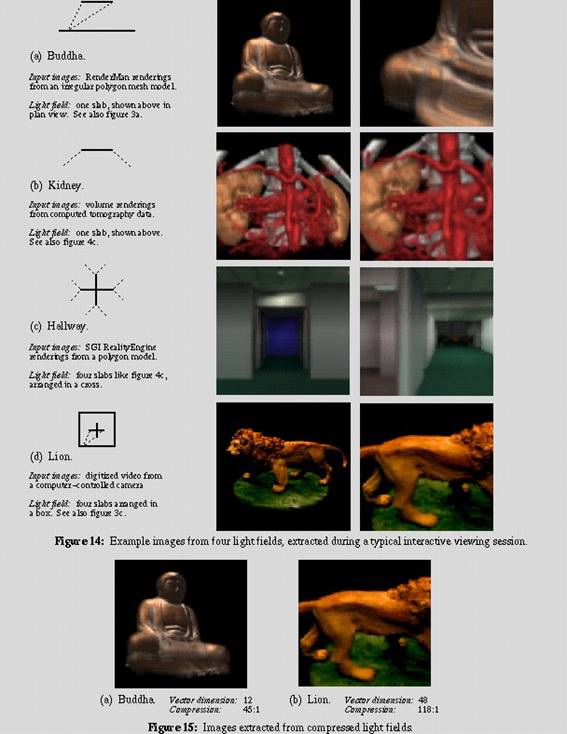
[Nimeroff94] Nimeroff, J., Simoncelli, E., Dorsey, J., &apos;&apos;Efficient Rerendering of Naturally Illuminated Scenes,&apos;&apos; Proc. Fifth Eurographics Rendering Workshop, 1994, pp. 359-373.

[Sbert93] Sbert, A.M., &apos;&apos;An Integral Geometry Based Method for FormFactor Computation,&apos;&apos; Computer Graphics Forum, Vol. 13, No. 3, 1993, pp. 409-420.

[Seitz95] Seitz, S., Dyer, C., &apos;&apos;Physically-Valid View Synthesis by Image Interpolation,&apos;&apos; Proc. IEEE Workshop on the Representation of Visual Scenes, IEEE, 1995.

[Williams83] Williams, L., &apos;&apos;Pyramidal Parametrics,&apos;&apos; Computer Graphics (Proc. Siggraph &apos;83), Vol. 17, No. 3, July, 1983, pp. 1-11.

[Ziv77] Ziv, J., Lempel, A., &apos;&apos;A universal algorithm for sequential data compression,&apos;&apos; IEEE Transactions on Information Theory, IT-23:337-343, 1977.



[[1]](" \l "_ftnref1" \o ") For those familiar with Gershun&apos;s paper, he actually uses the term light field to mean the irradiance vector as a function of position. For this reason P. Moon in a later book [Moon81] uses the term photic field to denote what we call the light field.