

Lightcuts: A Scalable Approach to Illumination

Bruce Walter Sebastian Fernandez Adam Arbree Kavita Bala Michael Donikian Donald P. Greenberg
Program of Computer Graphics, Cornell University*

摘要

Lightcuts是一个可扩展的框架，用于计算真实的光照效果。它处理任意几何形状、非漫反射材料和来自多种光源的照明，包括点光源、区域光源、高动态范围环境映射、日/天空模型和间接照明。其核心是一种新算法，用于以远低于线性成本的方式准确估计来自许多点光源的照明效果。我们展示了如何用较为廉价的成本来近似一组光源，同时限制最大近似误差。然后使用二进制光树和感知度量来适应性地划分光源组，以控制误差与成本的权衡。

我们还引入了重构切片的概念，利用空间连贯性来加速生成具有复杂照明的反锯齿图像。我们展示了五个复杂场景的结果，并表明Lightcuts可以使用仅几百个阴影光线准确近似数十万个点光源。重构切片可以将阴影光线的数量进一步减少到几十个。

CR 分类： I.3.7 [计算机图形学]: 三维图形与真实性 —— 颜色阴影、投影和纹理；

关键词： 多光源、光线追踪、投影

1 引言

尽管有大量研究集中于渲染具有复杂几何形状和材料的场景，但对于有效处理大量光源的方法研究相对较少。在典型的系统中，渲染成本与光源数量成线性增长。现实世界的场景通常包含许多光源，研究表明，人们通常更喜欢具有丰富和真实照明效果的图像。然而，在计算图形学中，我们往往被迫使用更少的光源或禁用一些重要的照明效果，如阴影，以避免过高的渲染成本。

Lightcuts框架提出了一种新的可扩展算法，用于计算来自许多点光源的照明。它的渲染成本与点光源数量强烈的亚线性关系，并且同时保持了精确解决方案的感知保真度。我们提供了一种快速近似一组光源照明的方法，更重要的是，提供了一种廉价且合理紧密的方法，用于对这样做所产生的最大误差进行限定。我们还提出了一种自动且局部自适应的方法，用于将光源划分为组，以控制成本和误差之间的权衡。为了高效找到分区，光源被组织成一个光树。Lightcuts能够处理非漫反射材料和任何可以进行光线追踪的几何形状。

拥有一个可扩展的算法使我们能够处理极大数量的光源。这特别有用，因为许多其他难以处理的照明问题可以通过使用足够多的点光源（例如，图1）来模拟照明。我们



图 1：大屏幕模型：由两个头顶区域灯、两个 HDR 平板显示器和间接照明组成的办公室。我们的可扩展框架使用 639,528 个点光源快速准确地计算出照明。监视器上的图像也是使用我们的方法计算的：光切割和重建切割。

展示了三个例子：来自区域光源的照明、高动态范围（HDR）环境映射或日/天空模型，以及间接照明。在lightcuts框架内统一不同类型的照明有额外的好处。例如，一个光源的强照明可以掩盖在近似其他照明时产生的误差，我们的系统自动利用了这种效应。

一种称为切割（reconstruction cuts）的相关技术利用空间连贯性来进一步降低渲染成本。它允许在图像上稀疏地计算lightcuts，并在它们之间智能插值。与大多数插值技术不同，重建割保留了高频细节，如阴影边界和光泽高光。Lightcuts可以使用仅几百个阴影光线来计算来自成千上万个光源的照明。重建割可以将这个数字进一步减少到只有十几个左右。

本文其余部分的组织如下。我们在第2节讨论预备工作。我们在第3节介绍了基本的lightcuts算法，第4节详细讨论了我们的实现，第5节讨论了不同的照明应用，第6节展示了lightcut的结果。然后我们在第7节描述重建切割，并在第8节演示它们的结果。第9节为结论。附录A给出了球形灯的优化。

2 Previous Work

在计算光源的照明和阴影方面，已经有大量的工作（例如，参见[Woo et al. 1990; Hasenfratz et al. 2003]的综述）。大多数技术加速了单个光源的处理，但仍与光源数量成线性比例。

几种技术专门解决了大量光源的问题。[Ward 1994]按最大贡献对光源进行排序，然后按降序评估它们的可见性，直到满足误差界限。

*email: {bjw,spf,arbree,kb,mike,dpg}@graphics.cornell.edu

我们将在第6节中将lightcuts与他的技术进行比较。[Shirley et al. 1996]将场景划分为单元格，并将每个单元格中的光源分为重要和不重要的列表，后者被非常稀疏地采样。这种或类似的蒙特卡罗技术在光源的采样概率函数足够好时表现良好，但为任意场景稳健且高效地计算这些函数仍是一个开放性问题。[Paquette et al. 1998]提出了一种使用与我们类似的光树的层次化方法。他们提供了能够保证的误差界限和良好的可伸缩性，但无法处理阴影，这限制了其适用性。[Fernandez et al. 2002]通过在场景中缓存每个光源的可见性和阻挡信息来加速许多光源的处理，但如果光源数量非常大，这会导致过度的内存需求。[Wald et al. 2003]在假设场景高度遮挡且只有一小部分光源对每个图像有贡献的情况下，能有效处理许多光源。这个子集是通过粒子追踪预处理确定的。

来自HDR环境映射的照明（通常来自照片[Debevec 1998]）正变得流行起来，用于更加真实的照明。智能采样技术可以将这些转换为用于渲染的定向点光源（例如，[Agarwal et al. 2003; Kollig and Keller 2003]），但通常仍然需要许多光源才能获得高质量的结果。

即时辐射度[Instant Radiosity]（Keller 1997）是基于从光源出发的随机粒子追踪的众多全局照明算法之一。它使用许多虚拟点光源来近似间接照明。可分辨的细节直接与虚拟光源的数量相关。这使得它与lightcuts完美契合，而以前它主要被限制在快速粗略的间接近似中。[Wald et al. 2002]在他们的交互式系统中使用了它，并添加了一些巧妙的技术来提高其分辨率。

光子映射是另一种流行的基于粒子的间接照明解决方案。它通常需要进行半球形最终聚集以获得良好的结果，每次聚集通常需要200到5000条光线[Jensen 2001 p.140]。在复杂场景中，lightcuts使用的光线比标准半球形聚集的光线更少来计算直接和间接照明。

层次化和聚类技术在许多领域广泛使用。在图形学中众所周知的例子包括辐射度技术（例如，[Hanrahan et al. 1991; Smits et al. 1994; Sillion and Puech 1994]）。与lightcuts不同，这些方法计算与视角无关的解决方案，如果详细地进行处理的话，可能会非常耗费计算和存储资源。此外，由于它们使用网格来存储数据，因此在处理一些常见的几何缺陷（如重叠或相交的多边形）时会遇到困难。最终聚集阶段通常用于提高图像质量。[Kok and Jansen 1992]和[Scheel et al. 2001; Scheel et al. 2002]的方法通过尽可能进行插值并在必要时才发射阴影光线来加速最终聚集。虽然这些方法的目标与重构切割非常相似，但它们使用基于辐射度链接结构的数据的启发式方法。

许多以前的技术使用连贯性和插值来减少渲染成本。重构切割的新颖性和力量来自于使用lightcuts框架。与之前的几种方法一样，重构切割使用定向光来廉价近似复杂光源的照明。[Walter et al. 1997]使用它们进行硬件加速的预计算全局照明解决方案的漫游。[Zaninetti et al. 1999]将它们称为光向量，并使用它们来近似来自各种光源（包括区域光源、天空穹顶和间接光源）的照明。

3 The Lightcuts Approach

给定一组点光源 S ，它们对表面点 x 在观察方向 ω 造成的直接照明的辐射度 L 是

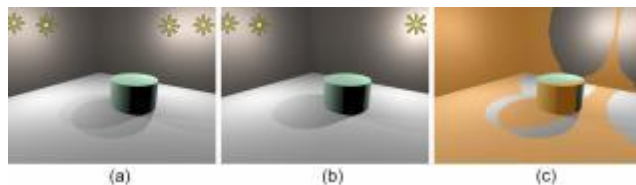


图 2: 具有 4 个点光源的简单场景。(a) 精确解。(b) 将右侧两盏灯聚集形成的近似解。(c) 橙色区域显示精确解和聚类解无法区分的地方。误差通常在灯光附近以及可见度不同的地方最大。

每个光源的材料、几何、可见性和强度项的乘积，且总和在所有光源上：

$$L_S(x; w) = \sum_{i \in S} \overbrace{M_i(x; w)}^{\text{material}} \overbrace{G_i(x)}^{\text{geometric}} \overbrace{V_i(x)}^{\text{visibility}} \overbrace{I_i}^{\text{intensity}} \quad (1)$$

精确解决方案的成本与光源数量成线性关系，因为这些项必须对每个光源进行评估。为了创建一个可扩展的亚线性方法，我们需要一种方法来近似一组光源的贡献，而无需单独评估每个光源。

我们定义一个簇 $C \subseteq S$ 为一组点光源和一个代表光源 $j \in C$ 。簇的直接照明可以通过使用代表光源的材料、几何和可见性项来近似所有光源，得到：

$$L_C(x; w) = \sum_{i \in C} M_i(x; w) G_i(x) V_i(x) I_i \\ M_j(x; w) G_j(x) V_j(x) \sum_{i \in C} I_i \quad (2)$$

簇强度 ($I_C = \sum I_i$) 可以预先计算并存储在簇中，使得簇近似的成本等同于评估单个光源的成本（即我们用一个更亮的光源替换了簇）。簇误差的数量将取决于簇内的材料、几何和可见性项的相似程度。图2展示了一个简单的例子。

Light Tree. 没有单一的光源分区可能适用于整个图像，但动态地为每个点找到新的簇分区可能过于昂贵。我们使用全局光树来快速计算局部自适应的簇分区。光树是一个二叉树，其中叶子是个别光源，内部节点是包含树下方的光源的光簇。穿过树的切割是一组节点，使得从树的根到叶子的每条路径都恰好包含一个来自切割组的节点。因此，每个切割对应于光源到簇的有效分区。图3展示了一个光树和三个不同的切割。

虽然每个切割都对应于有效的簇分区，但它们在成本和近似照明质量方面有很大差异。我们需要一种健壮和自动的方式来选择局部使用的合适切割。由于切割在图像中的变化，某些点或像素可能使用特定的簇来降低成本，而其他则用其子节点替换它们以提高精度。这样的过渡可能会导致显眼的图像伪影。为了防止这种情况，我们只在可以保证簇近似引入的误差低于感知可见性阈值时使用簇。韦伯定律[Blackwell 1972]是一种标准的众所周知的感知结果，它指出视觉信号中最小可感知的变化大致等于基础信号的固定百分比。在最坏的情况下，人类可以检测到不到1%的变化，

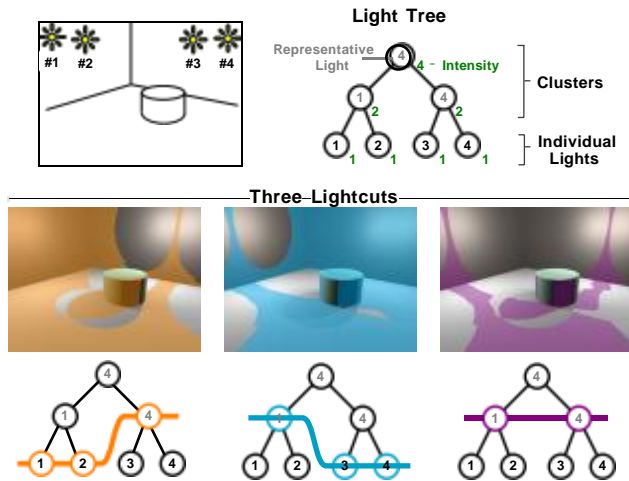


图 3: 一棵灯树和三个切割示例。该树显示在顶部, 其中包含每个节点的代表性灯光和簇强度。叶子是单独的灯, 而上面的节点是逐渐变大的灯簇。每个切割都是将灯光不同地划分为簇(橙色切割与图 2 相同)。在每次切割上方, 都会突出显示误差较小的区域。

尽管在实践中阈值通常更高。根据我们的经验, 2% 的误差比率在各种场景中不会导致可见的伪影, 并且用于我们所有的结果。改变这个值可以用来改变性能和准确性之间的权衡。

Choosing Lightcuts. 使用相对误差标准需要在我们可以决定特定簇是否可用之前对总辐射度进行估计。为了解决这个困难, 我们从一个非常粗略的切割开始(例如, 光树的根), 然后逐步细化它, 直到满足我们的误差标准。对于切割中的每个节点, 我们计算其簇估计(方程2)和误差上限(第4.1节)。每次细化步骤考虑当前切割中误差界限最大的节点。如果其误差界限大于我们的误差比率乘以当前的总照明估计, 我们就从切割中移除它, 用其在光树中的两个子节点替换它, 计算它们的簇估计和误差界限, 并更新我们对总辐射度的估计。否则, 切割遵循我们的误差标准, 我们就完成了。我们称这样的切割为lightcut。

为了使这个过程更有效, 我们要求簇的代表光源与其两个子节点之一相同。这允许我们在计算该子节点时重用代表光源的材料、几何和可见性项。我们使用堆数据结构来有效地找到剪辑中误差界限最高的簇节点。如果存在于切割中, 个别光源(即光树叶节点)被精确计算, 因此误差为零。

我们的相对误差标准在非常暗的区域高估了误差的可见性。例如, 完全遮挡的点将被允许有零误差, 但即使在黑色像素上, 足够小的误差也是不可见的。因此, 我们还设置了最大切割大小, 如果切割上的节点总数达到这个限制, 就停止进一步细化。我们选择的最大剪辑大小为1000, 足够大以至于在我们的结果中很少被达到, 然后只在额外的误差不可见的黑暗区域被达到过。

4 Implementing Lightcuts

我们的实现支持三种类型的点光源: 全向、定向和方向性。全向光源从单个点向所有方向均等地发光。

定向光源以它们的最大发射方向和指向被着色点 \mathbf{x} 的方向之间的角度 ϕ_i 定义的余弦加权半球模式发射。方向性光源模拟了一个无限远的源, 仅在单一方向发射。所有光源都有一个强度 I_i 。

构建the Light Tree. 光树将点光源组合成簇。理想情况下, 我们希望最大化它创建的簇的质量(即, 将材料、几何和可见性项最相似的光源组合在一起)。我们通过基于空间接近性和相似方向对光源进行分组来近似实现这一点。

我们将点光源按类型分为全向、定向和方向性列表, 并为每个列表构建一棵树。从概念上讲, 尽管我们将它们视为单个更大树的一部分。每个簇记录了它的两个子节点、代表光源、总强度 I_c 、轴对齐的边界框和方向边界锥。锥体仅对定向光源需要。虽然方向性光源实际上是无限远, 但在计算它们的边界框时被视为单位球面上的点。这使得方向性光源在构建光树时和其他点光源使用相同的技术, 更重要的是, 后来用于限定它们的材料项 M_i 。

相似性度量. 每棵树使用贪婪的自下而上方法构建, 通过逐步组合光源对和/或簇。在每一步, 我们选择将创建最小簇的对, 根据我们的簇大小度量 $I_c(\alpha c^2 + c^2(1 - \cos \beta c)^2)$, 其中, αc 是簇边界框的对角线长度, βc 是其边界锥的半角。常数 c 控制空间和方向相似性之间的相对缩放。对于定向光源, 它被设置为场景边界框的对角线长度, 对于全向和方向性光源则为零。

簇的代表光源始终与其一个子节点相同, 并根据子节点相对强度进行随机选择。每个单独的光源是其自己的代表。因此, 一个光源成为簇代表的概率与其强度成正比。这使得方程2中的簇近似在蒙特卡罗意义上是无偏的。然而, 一旦这么选择, 同一个代表光源将在整个图像中用于该簇。树的构建就其本质而言不可能是光源数量的亚线性, 但通常不是一个很大的成本, 因为它只需要在每个图像上执行一次(如果灯光是静态的, 则执行次数更少)。

4.1 边界簇误差

为了使用lightcuts方法, 我们需要计算相对低成本和紧密的簇误差上界(即方程2的精确和近似版本之间的差异)。通过计算簇的材料、几何和可见性项的上界, 我们可以将这些界限与簇强度相乘, 得到精确和近似簇结果的上界。由于两者都是正数, 这也是簇误差(即它们的绝对差异)的上界。

可见性项. 点光源的可见性通常为零或一, 但在某些情况下可能是分数(例如, 如果允许半透明表面)。在任意场景中保守地限定可见性是一个难题, 因此我们将使用一的平凡上界作为可见性项(即, 所有光源都可能可见)。

几何项. 我们三种点光源类型的几何项如下所列, 其中 \mathbf{y}_i 是光源的位置, ϕ_i 是定向光源的最大发射方向与指向被着色点 \mathbf{x} 的方向之间的角度。

Light Type	Omni	Oriented	Directional
$G_i(\mathbf{x}) =$	$\frac{1}{\Pi \mathbf{y}_i - \mathbf{x} \Pi^2}$	$\frac{\max(\cos \phi_i; 0)}{\Pi \mathbf{y}_i - \mathbf{x} \Pi^2}$	1

(3)

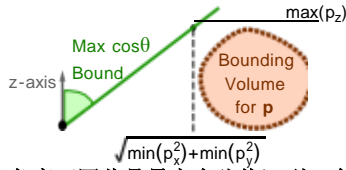


图4：界定最小角度（因此是最大余弦值）到一个边界体积。见方程4。

对于定向光源的上界很简单，因为它们的几何因子总是1。全向光源也很简单。我们只需计算点x和簇的边界体积之间的最小距离。定向光源更复杂，因为还有额外的余弦因子。我们可以使用余弦的简单上界1，但我们更倾向于一个更紧的上界。

让我们从图4所示的更简单的问题开始。对于任何点 $\mathbf{p} = [p_x; p_y; p_z]$ ，使得 θ 为从原点到点 \mathbf{p} 的向量与 z -axis 之间的角度。我们想要找到 $\cos\theta$ 的一个上界，以便在某个包围体积内对所有点 \mathbf{p} 进行累积。对于任何点 \mathbf{p} ，我们有 $\cos\theta = \frac{p_z}{\sqrt{p_x^2 + p_y^2 + p_z^2}}$ 。为了找到上界，我们首先用边界体积内 p_z 的最大值替换分子，然后选择 p_x 和 p_y 的值以使得分母取得最小或者最大值（取决于分子的符号）¹：

$$\cos\theta \leq \begin{cases} \frac{\max(p_z)}{4\min(p_x^2 + \min(p_y^2) + (\max(p_z))^2)} & \text{if } \max(p_z) \geq 0 \\ \frac{\max(p_z)}{4\max(p_x^2 + \max(p_y^2) + (\max(p_z))^2)} & \text{otherwise} \end{cases} \quad (4)$$

要将此应用于定向光源的 $\cos\phi_i$ ，我们将问题转化为如图 5 所示。首先考虑簇中的每个点对 $[\mathbf{x}, \mathbf{y}]$ 并将这两个点平移 $-\mathbf{y}_i$ 。这会将所有光源平移到原点，但将点 \mathbf{x} 分散到与簇的包围体大小相同的体积上（包围体的形状相同但是是颠倒的）。然后应用坐标变换，将 z 轴旋转以匹配簇的方向边界锥。现在我们可以使用方程4来计算簇的最小边界锥内的余弦值。如果这个角度位于边界锥内，那么我们只能使用 1 的平凡上限，但如果它位于外部，则 ϕ_i 必须至少与该角度减去圆锥体的半角一样大。

材料项。 材料项 M_i 等于 BRDF (双向反射分布函数) 乘以向量 $\mathbf{y}_i - \mathbf{x}$ 和 \mathbf{x} 处表面法线之间的角度余弦。我们已经描述了如何仅限定 BRDF 角度到一个边界 (图4)，所以这里只涉及将 BRDF 限定到边界。

我们的当前系统支持几种类型的BRDF组件：漫反射Lambertian，Phong [Phong 1975]，各向异性Ward [Larson 1992]，创建 BRDF 时可以对多个组件求和。漫反射 BRDF 分量只是一个常数（即不依赖于观察或照明方向），因此很容易受到限制。Phong 分量随光线矢量与镜面反射方向之间角度的余弦变化，并上升到某个指数。我们重用已经描述过的余弦边界机制来边界任何 Phong 组件。这种相同的方法可以适用于任何类似的关于轴对称的 BRDF 组件。

各向同性Ward BRDF不是关于任何轴对称的，因为它基于半角向量（即，表面法线和观察向量之间的角度）。

¹Note if p_x ranges from -2 to 1 then $\max(p_x) = 1$ and $(\max(p_x))^2 = 1$ but $\min(p_x^2) = 0$ and $\max(p_x^2) = 4$.

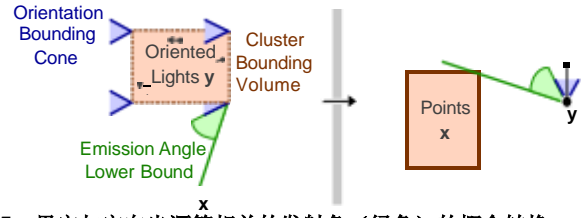


图5：界定与定向光源簇相关的发射角（绿色）的概念转换。

我们还开发了一种相对开销低且紧密的方法，用于界定到簇的最小半角。细节在[Walter 2005]中有说明。我们通过数值方法测试了所有的界限以确认它们的有效性。

原则上，只要有一种好的方法可以界定其在簇上的最大值，lightcuts可以与任何BRDF一起工作。Delta组件（例如，镜子）通过标准光线追踪方法处理，该方法递归地追踪反射和/或折射光线。

5 更多照明应用

一旦我们拥有了一个可扩展的解决方案，能够可靠地近似大量点光源的照明，我们就可以将其应用到其他难题上。这里包括的示例有：区域光源、高动态范围环境映射和间接照明。在lightcuts框架中整合不同类型的照明具有许多优势。例如，当这个误差将要被另一个组成部分的强照明所掩盖时，lightcuts会自动减少一个组成部分的准确性。

区域光源。 区域光源的照明和柔和阴影的计算是困难的，标准技术通常在区域光源的大小和数量上伸缩性较差。一个常见的方法是使用多个点光源来近似每个区域光源，但所需数量根据局部配置而大不相同。靠近区域光源或在其半影区域的位置需要许多点光源，而其他位置则需要较少。人们已经提出了许多启发式方法（例如，细分到某个固定的立体角），但这些方法可能并不适用于所有地方，并且通常需要手动调整参数。我们的系统中，每个光源可以被转换成大量的点。lightcut算法将自动和自适应地选择实际使用的样本数量。任何散射发光的区域光源都可以通过其表面上的定向光源来近似。对于球形光源，我们在附录A中描述了一种使用全向光源的更好方法。

HDR 环境映射。 高动态范围环境映射是捕捉真实世界环境照明并将其应用到合成场景中的一种流行方式。然而计算准确的照明，特别是阴影，可能非常昂贵。

最常见的方法是将环境映射转换成一组离散的定向光源。一个关键问题是要使用多少定向光源。使用太少会导致图像伪影，如块状阴影，而使用太多则会显著增加渲染成本。例如，[Agarwal et al. 2003] 建议使用300个定向光源通常是足够的。在我们的经验中，对于孤立对象，300个通常是足够的，但渲染具有显著遮挡和/或窄光泽BRDF的完整场景可能需要3000个或更多。lightcuts方法可以比以前的方法更有效地处理如此大量的光源。

间接照明。 间接照明（也称为全局照明）因其图像质量和真实感而受到青睐，但对于许多应用而言被认为过于昂贵。大量研究致力于提高它们的物理和数学准确性，



Point Lights	Avg Cut Size	Avg Shadow Rays	Time	Reference
4608	264	259	128s	1096s

图 6: 来自 72 个区域光源的直射光的厨房场景。

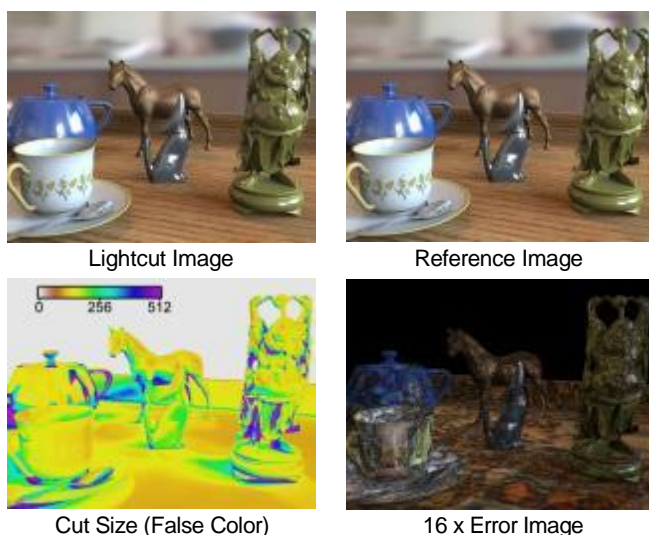
但同时也对成本较低的近似解感兴趣，只要它们不引入令人反感的图像伪影即可（例如，[Tabellion and Lamorlette 2004]）。

即时辐射度（Instant Radiosity）是这样一种方法。它首先跟踪光粒子在场景中的概率散射。然后在这些散射位置创建虚拟点光源，使它们的总照明模拟间接照明的效果。在选择虚拟光源的数量时，存在成本与质量的权衡。使用更多的光源可以再现更多间接照明的细节，但也增加了评估如此多光源的成本。以前的结果大多限于数十个或最多数百个虚拟光源（例如，[Keller 1997; Wald et al. 2002]）。使用我们的可扩展算法，我们可以使用数千甚至数百万个虚拟光源来再现更多的间接照明细节。

以前的即时辐射度结果仅限于使用少量虚拟间接光源的漫反射间接效果。摆脱了这一限制，我们的系统还可以包括一些光泽间接效果。在将粒子碰撞转换为虚拟定向点光源时，仅使用BRDF的漫反射成分，但与以前的系统不同，我们在它们照明的点上使用完整的BRDF。

使用即时辐射度时，需要注意其固有的限制。它无法再现某些类型的间接照明（例如焦散光效应），如果需要这些效果，必须使用其他方法。它还在短距离和/或光泽间接效果方面存在困难，尽管使用更多的虚拟光源有所帮助。粒子追踪是一个随机过程，因此虚拟光源的位置存在相当大的随机性。任何特定光源的贡献主要是干扰，只有通过结合许多光源的结果，我们才能获得统计上可靠的信息。没有干扰抑制技术，这种干扰将在照明容易被一个或几个光源主导的位置（例如凹角或尖锐的光泽反射）中非常明显。

干扰抑制通常通过限制虚拟光源的最大贡献来实现。这会偏向结果，但比令人反感的干扰伪影更可取。[Keller 1997] 使用的硬件将值限制在最大255，而[Wald et al. 2002] 让用户指定在计算间接光的几何因子时使用的最小距离。后者不适用于非漫反射BRDF，因此我们对个别间接光的总贡献MiGiVili应用了一个夹紧阈值。



Point Lights	Avg Cut Size	Avg Shadow Rays	Time	Reference
3000	(187) 132	(168) 118	54s	424s

图 7: HDR 环境贴图照亮的 Tableau 场景。括号中是仅包含几何图形的像素的平均值。

我们不会限制簇的贡献，因为如果太亮，它们将通过正常的减光细化来处理。该阈值可以由用户指定或自动确定，如下所示。

我们的自动夹紧限制每个间接光的贡献不超过总结果的固定百分比。由于这个夹紧阈值是从相同的干扰粒子数据中计算出来的，我们必须注意不要强调干扰。我们使用lightcut选择/精炼过程中的误差比率参数的一半（例如，如果误差比率为2%，则使用1%）。在lightcut精炼过程中，我们跟踪切割中可能需要夹紧的任何间接节点。根据我们的经验，自动夹紧几乎不需要额外开销，至少和手动调整的夹紧阈值一样有效，且为用户减少了一个参数的担忧。

6 Lightcut Results

在这一部分中，我们展示了将lightcuts实现应用于五个不同照明场景的结果。所有结果都使用2%的错误比率和最大1000个节点的剪裁大小。本节中的所有图像分辨率为640x480，每个像素使用一个眼睛光线，但由于反射和透明度，这有时需要对多个点进行着色。时间是在一台装有两个3 GHz Xeon处理器的单个工作站上测量的，尽管我们的系统也能有效地使用多台机器。

误差。 Lightcuts保守地限制了每个簇的最大误差，以确保单个簇的细化过渡不可见。理论上，许多不同簇的误差仍然可能累加成大的、可见的误差。但实际上我们发现这并不是问题，原因有二：每个簇的误差界限是最坏情况的界限；实际的平均簇误差远低于我们的2%界限。另外，不相关的误差积累比相关误差要慢得多（i.e. $O(pN)$ vs. $O(N)$ ）。在构建树的过程中，我们随机化选择代表性光源，以确保簇误差是不相关的。

$$2 \frac{1}{1000} \text{ of the image white point is a reasonable starting point.}$$



Model	Polygons	Number of Point Lights				Per Pixel Averages		Tree Build	Image Time
		Direct	Env Map	Indirect	Total	(d+e+i) Lightcut Size	Shadow Rays		
Kitchen	388552	(72) 4608	5064	50000	59672	(54+244+345) 643	(0.8%) 478	3.4s	290s
Tableau	630843	0	3000	10000	13000	(0+112+104) 216	(1.1%) 145	0.9s	79s
Grand Central	1468407	(820) 38400	5064	100000	143464	(73+136+480) 689	(0.33%) 475	9.7s	409s
Temple	2124003	0	5064	500000	505064	(0+185+437) 622	(0.07%) 373	44s	225s
Bigscreen	628046	(4) 614528	0	25000	639528	(83+0+222) 305	(0.04%) 228	46s	98s

图9：启用所有照明组件的五个场景的 640x480 图像的 Lightcut 结果。平均切割尺寸包括来自三个组成部分的数量：直接、环境贴图 and 间接。每个像素的平均阴影光线也显示为灯光总数的百分比。混叠（例如，中央车站的窗户）是由于这里每个像素仅使用一束光线造成的。

本质上，lightcut 提供了一种随机误差范围，而不是绝对误差范围。大的总误差还是可能的，但可能性很小。

图6所示的厨房模型基于实际房屋的一部分。它包含72个区域光源，每个光源使用64个点光源进行近似，总共有4608个点光源。即使仔细检查，也无法在lightcut结果和精确评估每个点光源的参考图像之间看出可见差异。误差图像几乎是黑色的。将误差放大16倍显示，与预期一致的是，误差通常在较亮的区域更大，并由使用簇与细化簇之间的过渡引起的许多不连续性构成。Lightcut图像的生成时间为128秒，平均剪裁大小为264，每像素影子光线为259，而参考图像的生成时间要长得多，为1096秒，平均每像素影子光线为3198。不会对材料或几何项为零的光源发射影子光线。

图7所示的台面模型有几个具有不同光泽材料的物体，放在一个木托盘上，并由捕获的HDR环境映射照明（从[Debevec 1998]的厨房映射，与我们的厨房模型无关）。我们使用了[Agarwal et al. 2003]的技术将这个映射转换为3000个定向光源进行渲染。同样，参考图像与lightcut图像之间没有可见差异。可视化显示了每像素的剪裁大小（即lightcut上的节点数量）。剪裁大小与渲染成本密切相关。

可扩展性。随着点光源数量的增加，lightcuts在本质上更具可扩展性（即亚线性而不是线性）。为了证明这一点，我们通过改变每个区域光源和环境映射分别创建的点光源和定向光源的数量，

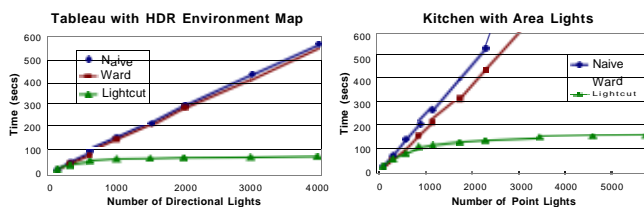


图8：随着点光源数量的增加，Lightcut 性能从根本上得到更好的扩展（即亚线性）。

来变化厨房和台面示例中的点光源数量。图8展示了lightcuts和简单（参考）解决方案的图像时间与点光源数量的对比。我们还与[Ward 1994]进行了比较，我们认为在通用性、稳健性和处理数千个光源的能力方面，这是最好的替代方法之一。

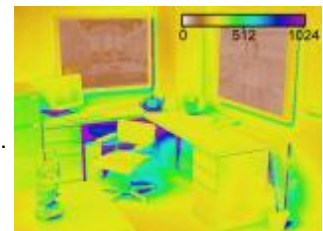
Ward的技术计算所有光源的潜在贡献，假设完全可见，按降序排序，然后逐步评估它们的可见性，直到剩余潜在贡献降至评估光源总量的一小部分以下（在我们的比较中为10%）。由于阴影光线（即可见性测试）通常是主要成本，减少它们通常会补偿排序的成本（对于厨房是真实的，对于台面刚好）。然而，如图所示，成本仍然呈线性表现。假设我们有1000个光源，其未遮挡的贡献大致相等。我们将必须检查至少800个光源的可见性，以满足10%的总误差界限，而这个数字与光源数量成线性增长。这是Ward技术或任何提供绝对误差界限的技术的致命弱点。Lightcuts的卓越可扩展性意味着随着光源数量的增加，其优势迅速增长。

混合照明。我们可以利用这种可扩展性在场景中计算更丰富和更复杂的照明效果，如图9所示。Grand Central是纽约市著名地标的模型，包含220个全向点光源，分布在主大厅的天花板附近，以及侧廊中的600个球形灯。真实的建筑实际上包含更多的灯光。Temple是一个埃及寺庙的模型，是我们最具几何复杂性的模型，有210万个三角形。此类几何复杂性

³Using the optimal value of 0.5 for the visibility of the untested lights.



Bigscreen Model
Figure 10: Bigscreen model.



引起了走样问题，因为我们每个像素只使用了一个眼睛光线。下一节将介绍重构切割，它允许我们以降低的成本进行抗锯齿。**Bigscreen**模型（见图10）显示了一个办公室，由两个悬挂的区域光源（每个64点）和两个大型HDR监视器照明，显示我们其他场景的图像。监视器分辨率为640x480，通过将每个像素转换成一个点光源来模拟。

我们已经在厨房、**Grand Central**和**temple**模型中添加了[Preetham et al. 1999]的太阳/天空模型，它像HDR环境映射一样起作用。太阳被转换成64个方向光源，天空则被转换成均匀分布在方向球上的5000个方向光源。我们还使用即时辐射度方法向所有模型添加了间接照明。在厨房和**Grand Central**模型中，只有一小部分来自太阳/天空的粒子通过窗户进入内部；许多间接光源最终在建筑物的外部。使用我们的可扩展算法，我们可以通过生成更多间接光源来补偿。

使用lightcuts时，光源数量并不强烈相关图像成本。相反，图像成本的最强预测因素是光源的遮挡程度。因此，厨房和特别是**Grand Central**是最昂贵的，因为它们的光源，特别是太阳/天空的遮挡程度很高。尽管**Temple**和**Bigscreen**有更多的点光源和几何体，但由于它们的可见性更高，因此成本较低。

在lightcuts中，阴影光线是主要成本，占据了总时间的大约50%，尽管这是我们代码中最优化的部分（用C编写，而其他部分是Java）。计算误差界限占了另外20%，着色占了10%。包括树遍历、维护最大堆、更新估计等各种较小的操作占据了剩余部分。树构建对于最大数量的光源开始变得显著，但有比我们简单的贪婪方法更快的方法。

界限紧密性。能够廉价地限制簇的最大贡献，从而限制误差的能力，对于我们的可扩展性至关重要。使用更紧的（可能更昂贵的）界限可能会减少平均切割大小，从而降低总成本。在图11中，我们展示了如果我们对几何和材料项 $G_i M_i$ 有精确界限，将会产生的切割大小。这是我们在没有可见性界限的情况下所能希望达到的最好结果。总的来说，我们的界限表现得非常好。间接光源由于分散更广，更难界定，但我们的界限仍然表现良好。显著的进一步提高可能需要有效的保守可见性界限。这在许多具体情况下是可能的（例如，参见[Cohen-Or et al. 2003]），但通常是一个未解决的问题。

Model	Illumination	Point Lights	Avg Cut Size	
			Lightcut	Exact $G_i M_i$ Bound
Kitchen	Direct	4608	264	261
	+ Indirect	54608	643	497
Tableau	Env Map	3000	132	120
	+ Indirect	13000	216	153
Temple	Sun/Sky	5064	294	287

图 11：更严格的边界对光切尺寸的影响程度。

7 重构切割

Reconstruction cuts 是一种新技术，利用空间连贯性来降低平均着色成本。这个想法是在图像上稀疏地计算lightcuts（例如，在图像块的角落），然后适当地插值它们的照明信息来着色图像的其余部分。尽管lightcuts允许

可扩展地计算数千甚至数百万个点光源的精确照明，但它们通常仍然需要为每个着色点发射数百个阴影光线。通过采取略微不那么保守的方法，重构切割能够利用照明的平滑性来大幅度降低着色成本。

最简单的方法是简单地插值稀疏lightcuts的辐射度（例如，Gouraud着色），但这将导致高频图像特征（如阴影边界和光泽高光）的不可接受模糊。许多扩展这一基本想法的方法已被提出，以保留特定类型的特征（例如，[Ward and Heckbert 1992] 插值辐射度以保持漫反射纹理，[Křivanek et al. 2005] 使用多个方向系数来保持低频光泽），但我们希望保留所有特征，包括尖锐的阴影边界。

要在一个点计算重构切割，我们首先需要一组附近的样本（即已经计算并处理过的lightcuts的位置，以便于轻松插值）。然后，我们从顶向下遍历全局光树。如果所有样本都同意一个节点被遮挡，则我们立即丢弃它。如果一个节点的照明在样本间非常相似，那么我们将使用冒牌灯进行廉价插值。这些特殊的定向灯旨在模仿样本中记录的簇的聚合行为。否则，我们将默认采用lightcut风格的行为：沿树向下细化，直到错误足够小，并使用方程式2中的簇近似，包括发射阴影光线。

插值或丢弃树上较高位置的节点可以节省大量成本。然而，当样本跨越阴影边界或其他尖锐特征时，我们将转而采用更为稳健但成本更高的方法处理受影响的节点。因为重构切割有效地包括了可见性剔除，它们不像lightcuts那样受高遮挡场景的影响。

Samples. 首先通过为点 \mathbf{x}_k 和观察方向 \mathbf{w}^k 计算一个lightcut来创建样本 k 。这将包括在lightcut上的每个光树节点 n 处使用方程2计算辐射度估计 \tilde{I}_n^k 。对于剪辑上方的每个节点，我们定义 \tilde{I}_n^k 为其在剪辑上的所有后代的辐射度估计之和。类似地，总辐射度估计 I_n^k 是剪辑中所有节点的总和。

要将一个lightcut转换为一个样本，我们为剪辑上或以上的每个节点创建冒牌的 *impostor directional point lights*（具有方向 \mathbf{d}_n^k 和强度 γ_n^k ）。它们的方向模仿相应簇或光源的入射光的平均方向，它们的照明精确地再现了样本点的辐射度估计 \tilde{I}_n^k 。冒牌光源从不被遮挡，因此相对便宜（即可见性和几何项都等于一）。

对于剪辑上的节点，冒牌方向 \mathbf{d}_n^k 与其代表光源的方向相同。对于剪辑上方的节点， \mathbf{d}_n^k 是其在剪辑上的后代的方向的平均值，按它们各自的辐射度估计加权。使用这个方向，我们可以评估冒牌光源的材料项 M_n^k 和其强度 γ_n^k 。我们还根据总辐射度估计计算第二个强度 Γ_n^k ，并用于计算相对阈值（即节点与总照明的相对大小）。

$$\gamma_n^k = \tilde{I}_n^k(\mathbf{x}^k; \mathbf{w}^k) / M_n^k(\mathbf{x}^k; \mathbf{w}^k) \quad (5)$$

$$\Gamma_n^k = \tilde{I}_n^k(\mathbf{x}^k; \mathbf{w}^k) / M_n^k(\mathbf{x}^k; \mathbf{w}^k) \quad (6)$$

偶尔我们需要为样本的lightcut下方的节点创建一个冒牌的定向光源。这些在需要时被创建，并使用与它们在剪辑上的祖先相同的方向，但模拟强度由节点的簇强度 I_c 与其祖先的比例降低。

确切的样本大小取决于lightcut，但24KB是典型的。样本为剪辑上或以上的每个节点存储32字节，其中包括7个浮点数（1 for Γ_n^k and 3 each for d_n^k and y_n^k ），如果存在，则包含指向其子节点数据的偏移量。我们将图像分解成块，这样任何时候只保留一小部分样本。

计算重构切割。 给定一组样本，我们希望使用它们快速估计方程1。重构切割使用全局光树的自上而下遍历。在访问的每个节点，我们计算样本的最小和最大模拟强度 y_n^k 并设置阈值 τ_n ，为lightcut误差比例（例如2%）乘以样本中 Γ_n^k 的最小值。然后我们选择以下适用的规则之一：

1. **丢弃。** 如果 $\max(y_n^k) = 0$ ，则该节点在所有样本上的辐射度均为零，假定在这里也是零。
2. **插值。** 如果 $\max(y_n^k) - \min(y_n^k) < \tau_n$ 且 $\min(y_n^k) > 0$ ，则我们计算 d_n^k 和 y_n^k 的加权平均值，创建一个插值的冒牌定向光源。然后，该节点的估计值等于插值方向的材料项乘以插值强度。
3. **簇评估。** 如果 $\max(y_n^k) < \tau_n$ 或我们处于叶节点（即个别光源），则我们使用方程2估计辐射度。这包括向代表光源发射阴影光线（如果材料和几何项不为零）。
4. **细化。** 否则我们递归下树，对每个子节点执行相同的测试。

虽然上述规则涵盖了大多数情况，但为了防止偶尔的伪影（例如，在光泽材料上），需要进行一些细化。首先，我们不允许在光泽高光内部进行插值。漫反射BRDF的最大可能值为 $1/\pi$ 。当为插值的冒牌光源计算材料项时，如果其BRDF值大于 $1/\pi$ ，则该方向必须位于材料的光泽叶片内部，我们不允许该节点的插值。其次，只有在或低于至少一个样本lightcut的节点才允许进行簇评估。低于所有样本lightcut的节点不使用插值，因为它们没有好的插值信息。第三，如果簇评估的结果比预期的要大得多，我们递归地评估其子节点。

图像块。 首先将图像分割成 16×16 像素的块，这些块不共享样本信息。这样做可以保持存储需求较低，并且方便并行处理。为了处理每个块，我们最初将其分割为 4×4 像素的小块，但可能会根据以下测试进一步细分。

为了计算一个块，我们在其角落处计算样本，通过其像素发射眼睛光线，并测试结果点是否与角落样本匹配。如果使用抗锯齿，每个像素将有多条眼睛光线。一组眼睛光线被认为是匹配的，如果它们都击中了具有相同材料类型的表面，并且表面法线相差不超过某个角度（例如30度）。我们还使用圆锥测试来检测可能需要块细分的局部阴影条件（例如，见图12）。对于每个眼睛光线，我们在其交点处构造一个圆锥，以局部表面法线为中心。如果任何其他眼睛光线的交点位于这个圆锥内，则该块不通过圆锥测试。圆锥测试使用真实的几何法线，并且不受凹凸贴图或插值法线等着色效果的影响。

如果眼睛光线不匹配并且块大于一个像素，那么我们将块分成四个更小的块并再次尝试。如果块是像素大小的，我们放松要求，省略圆锥测试，只要求每个光线至少与两个附近样本匹配。如果仍然没有足够的匹配样本，那么我们在该光线处计算一个新样本。

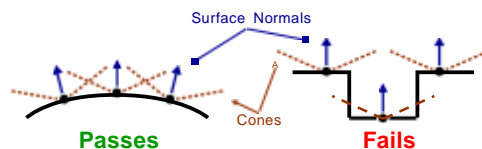


图12：块圆锥测试。（右侧）失败，因为上面两个点位于下面一个点的圆锥内。（左侧）通过，因为没有点位于其他点定义的圆锥内。

在块细分之后，我们使用重构切割为每个剩余的眼睛光线计算颜色。对于大于一个像素的块，我们使用四个角作为附近样本的集合，并在插值冒牌光源时使用图像空间双线性插值权重。像素大小块内的眼睛光线使用它们的匹配样本集合，并使用与表面点和样本点之间的世界空间倒数距离平方成正比的插值权重。

8 重构切割结果

在这一节中，我们展示了将重构切割技术应用于我们的场景的一些结果。结果图像和统计数据显示在图13中。这些结果使用自适应抗锯齿[Painter and Sloan 1989]，每个像素使用5到50个眼睛光线。稀疏样本使用与第6节相同的参数的lightcuts计算。大部分的着色是使用重构切割完成的，它在样本之间智能地进行插值。

通过利用空间连贯性，重构切割可以使用远少于lightcuts的阴影光线来着色点。虽然lightcut平均需要几百个阴影光线，但重构切割在我们的结果中使用的阴影光线不到十四个。事实上，我们的大部分阴影光线用于计算稀疏样本（lightcuts），即使重构切割的数量是其15-25倍。这使我们能够以远低于仅使用lightcuts的成本生成更高质量的图像，并使用抗锯齿。对于相同大小的图像，这些结果的质量更高，并且成本与第6节中的相似或更低。此外，对于更大的图像，渲染成本增长速度比像素数量增长速度慢。

样本在自适应大小的图像块角落（大小在 4×4 到 1×1 像素之间）处计算，并在需要时偶尔在像素内计算（以红色 1×1 显示）。如temple图像所示，所有图像中的大多数像素位于 4×4 块内。即使在需要每个像素多个眼睛光线的抗锯齿情况下，每个像素平均需要的样本数量也不到一个。我们可以允许更大的块（例如， 8×8 ），但在已经稀疏的地方使样本更稀疏收益较小，甚至可能适得其反，因为重构切割成本与附近样本的相似性密切相关。

由于重构切割的准确性依赖于覆盖局部照明条件的附近样本，存在样本之间遗漏小特征的可能性。虽然这偶尔会发生，但很少成问题。例如，在寺庙的柱子上的几个地方，插值使凹槽变得平滑，但错误相当小，我们发现人们很难注意到它们。未来对块细化规则的改进可能会修正这些错误。

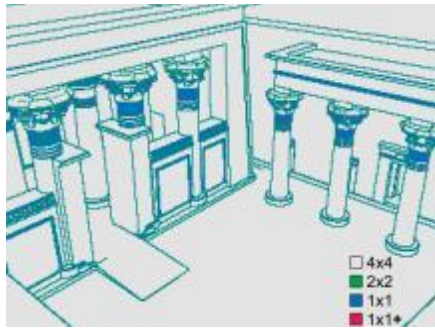
在图14中展示了Metropolis解决方案及其与我们结果的放大（5倍）差异。Metropolis [Veach and Guibas 1997] 被认为是最好和最快的通用蒙特卡罗求解器。其结果的计算时间是我们的13倍，并且仍然包含一些可见的干扰。主要区别是Metropolis解决方案中的干扰以及我们的结果的一些角落阴影。



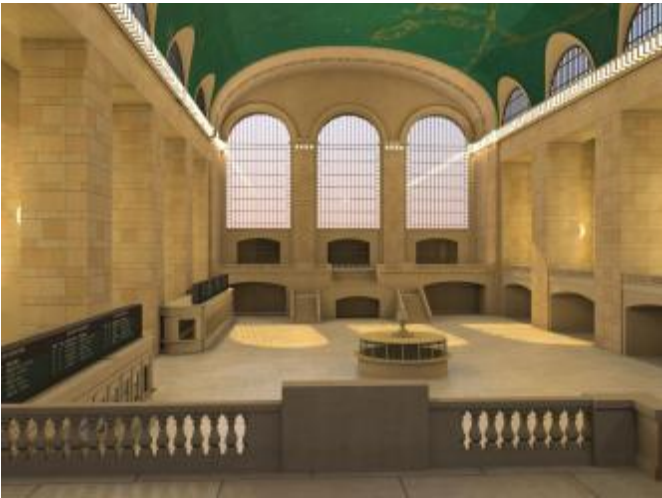
Kitchen



Tableau



Temple (Reconstruction Block Size)



Grand Central



Temple

Model	Point Lights	Per Pixel Averages		Avg Shadow Rays Per Pixel		Avg Per Reconstruction Cut		Image Time	
		Eye Rays	Samples	Samples	Recon Cuts	Shadow Rays	Interpolations	1280x960	640x480
Kitchen	59672	5.4	0.29	143	50	9.1	14.4	672s	257s
Tableau	13000	5.4	0.23	50	41	10.6	17.8	298s	111s
Grand Central	143464	6.9	0.46	225	93	13.3	11.5	1177s	454s
Temple	505064	5.5	0.25	91	52	9.4	6.0	511s	189s
Bigscreen	639528	5.3	0.25	64	24	4.6	15.0	260s	98s

图 13: 场景的 1280x960 图像 (除非另有说明) 的重建切割结果。图像自适应地使用每像素 5 至 50 条眼光光线来消除锯齿。样本 (图 9 中的光切割) 是稀疏计算的, 并且大多数着色是使用重建切割来智能插值来计算的。重建切割比光切割便宜得多, 并且需要的阴影光线也少得多。大屏图像如图1所示。

这是由于即时辐射度没有再现非常短距离的间接照明效果导致的。

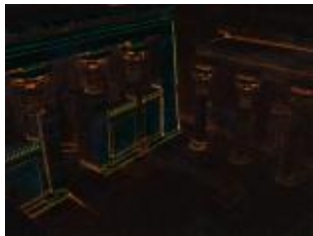
9 Conclusion

我们提出了lightcuts作为一种新的可扩展的统一照明框架。其核心组件是一种强亚线性算法, 用于计算来自数千甚至数百万个点光源的照明, 使用感知误差度量和保守的每个簇误差界限。我们展示了它可以大大减少计算照明所需的阴影光线数量, 从而降低成本, 适用于多种光源, 包括区域光源、HDR环境地图、日/天空模型和间接照明。此外, 它还可以处理非常复杂的场景, 包括具有详细几何和光泽材料的场景。我们还介绍了重构切割, 它通过利用照明的连贯性来进一步加速着色。

本工作还有许多改进和扩展的空间。



Metropolis Solution



5 x Difference

图14: 寺庙与都市形象的比较。

例如, 扩展到额外的照明类型 (例如, 即时辐射度无法处理的间接组件), 整合更多类型的光源, 包括非漫反射源和聚光灯, 开发更多BRDF类型的界限, 对lightcuts随机误差进行更正式的分析, 进一步改进重构切割规则以利用更多连贯性, 以及添加保守的可见性界限以进一步加速lightcuts。

Acknowledgments

感谢以下模型设计者: Jeremiah Fairbanks (厨房), Will Stokes (大屏幕), Moreno Piccolotto, Yasemin Kologlu, Anne Briggs, Dana Getman (大中央车站) 以及Bristol大学的Veronica Sundstedt, Patrick Ledda 和图形小组 (寺庙)。这项工作得到了NSF资助 (编号ACI-0205438) 和英特尔公司的支持。本文中表达的观点仅代表作者本人, 不代表美国空军、国防部或美国政府的官方政策或立场。

References

- AGARWAL, S., RAMAMOORTHY, R., BELONGIE, S., AND JENSEN, H. W. 2003. Structured importance sampling of environment maps. *ACM Transactions on Graphics* 22, 3 (July), 605–612.
- BLACKWELL, H. R. 1972. Luminance difference thresholds. In *Handbook of Sensory Physiology*, vol. VII/4: Visual Psychophysics. Springer-Verlag, 78–101.
- COHEN-OR, D., CHRYSANTHOU, Y. L., SILVA, C. T., AND DURAND, F. 2003. A survey of visibility for walkthrough applications. *IEEE Transactions on Visualization and Computer Graphics* 9, 3, 412–431.
- DEBEVEC, P. 1998. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of SIGGRAPH 98*, Computer Graphics Proceedings, Annual Conference Series, 189–198.
- FERNANDEZ, S., BALA, K., AND GREENBERG, D. P. 2002. Local illumination environments for direct lighting acceleration. In *Rendering Techniques 2002: 13th Eurographics Workshop on Rendering*, 7–14.
- HANRAHAN, P., SALZMAN, D., AND AUPPERLE, L. 1991. A rapid hierarchical radiosity algorithm. In *Computer Graphics (Proceedings of SIGGRAPH 91)*, vol. 25, 197–206.
- HASENFRATZ, J.-M., LAPIERRE, M., HOLZSCHUCH, N., AND SILLION, F. 2003. A survey of real-time soft shadows algorithms. In *Eurographics, Eurographics*. State-of-the-Art Report.
- JENSEN, H. W. 2001. *Realistic image synthesis using photon mapping*. A. K. Peters, Ltd.
- KELLER, A. 1997. Instant radiosity. In *Proceedings of SIGGRAPH 97*, Computer Graphics Proceedings, Annual Conference Series, 49–56.
- KOK, A. J. F., AND JANSEN, F. W. 1992. Adaptive sampling of area light sources in ray tracing including diffuse interreflection. *Computer Graphics Forum (Eurographics '92)* 11, 3 (Sept.), 289–298.
- KOLLIG, T., AND KELLER, A. 2003. Efficient illumination by high dynamic range images. In *Eurographics Symposium on Rendering: 14th Eurographics Workshop on Rendering*, 45–51.
- KŮŤÁNEK, J., GAUTRON, P., PATTANAIK, S., AND BOUATOUCH, K. 2005. Radiance caching for efficient global illumination computation. *IEEE Transactions on Visualization and Computer Graphics*.
- LARSON, G. J. W. 1992. Measuring and modeling anisotropic reflection. In *Computer Graphics (Proceedings of SIGGRAPH 92)*, vol. 26, 265–272.
- PAINTER, J., AND SLOAN, K. 1989. Antialiased ray tracing by adaptive progressive refinement. In *Computer Graphics (Proceedings of SIGGRAPH 89)*, vol. 23, 281–288.
- PAQUETTE, E., POULIN, P., AND DRETTAKIS, G. 1998. A light hierarchy for fast rendering of scenes with many lights. *Computer Graphics Forum* 17, 3, 63–74.
- PHONG, B. T. 1975. Illumination for computer generated pictures. *Commun. ACM* 18, 6, 311–317.
- PREETHAM, A. J., SHIRLEY, P. S., AND SMITS, B. E. 1999. A practical analytic model for daylight. In *Proceedings of SIGGRAPH 99*, Computer Graphics Proceedings, Annual Conference Series, 91–100.
- SHEEL, A., STAMMINGER, M., AND SEIDEL, H.-P. 2001. Thrifty final gather for radiosity. In *Rendering Techniques 2001: 12th Eurographics Workshop on Rendering*, 1–12.
- SHEEL, A., STAMMINGER, M., AND SEIDEL, H. 2002. Grid based final gather for radiosity on complex clustered scenes. *Computer Graphics Forum* 21, 3, 547–556.
- SHIRLEY, P., WANG, C., AND ZIMMERMAN, K. 1996. Monte carlo techniques for direct lighting calculations. *ACM Transactions on Graphics* 15, 1 (Jan.), 1–36.
- SILLION, F. X., AND PUECH, C. 1994. *Radiosity and Global Illumination*. Morgan Kaufmann Publishers Inc.
- SMITS, B., ARVO, J., AND GREENBERG, D. 1994. A clustering algorithm for radiosity in complex environments. In *Proceedings of SIGGRAPH 94*, Annual Conference Series, 435–442.
- TABELLION, E., AND LAMORLETTE, A. 2004. An approximate global illumination system for computer generated films. *ACM Transactions on Graphics* 23, 3 (Aug.), 469–476.
- VEACH, E., AND GUIBAS, L. J. 1997. Metropolis light transport. In *Proceedings of SIGGRAPH 97*, Computer Graphics Proceedings, Annual Conference Series, 65–76.
- WALD, I., KOLLIG, T., BENTHIN, C., KELLER, A., AND SLUSALLEK, P. 2002. Interactive global illumination using fast ray tracing. In *Rendering Techniques 2002: 13th Eurographics Workshop on Rendering*, 15–24.
- WALD, I., BENTHIN, C., AND SLUSALLEK, P. 2003. Interactive global illumination in complex and highly occluded environments. In *Eurographics Symposium on Rendering: 14th Eurographics Workshop on Rendering*, 74–81.
- WALTER, B., ALPPAY, G., LAFORTUNE, E. P. F., FERNANDEZ, S., AND GREENBERG, D. P. 1997. Fitting virtual lights for non-diffuse walkthroughs. In *Proceedings of SIGGRAPH 97*, Computer Graphics Proceedings, Annual Conference Series, 45–48.
- WALTER, B. 2005. Notes on the Ward BRDF. Technical Report PCG-05-06, Cornell Program of Computer Graphics, Apr.
- WARD, G. J., AND HECKBERT, P. 1992. Irradiance gradients. In *Third Eurographics Workshop on Rendering*, 85–98.
- WARD, G. 1994. Adaptive shadow testing for ray tracing. In *Photorealistic Rendering in Computer Graphics (Proceedings of the Second Eurographics Workshop on Rendering)*, Springer-Verlag, New York, 11–20.
- WOO, A., POULIN, P., AND FOURNIER, A. 1990. A survey of shadow algorithms. *IEEE Computer Graphics and Applications* 10, 6 (Nov.), 13–32.
- ZANINETTI, J., BOY, P., AND PEROCHE, B. 1999. An adaptive method for area light sources and daylight in ray tracing. *Computer Graphics Forum* 18, 3 (Sept.), 139–150.

A Spherical Lights

Any diffuse area light can be simulated using oriented lights on its surface. For spherical lights, we have developed a better technique that simulates them using omni lights (which are a better match for far field emission). Placing omni lights on the surface of the sphere would incorrectly make it appear too bright near its silhouette as compared to its center. Similarly a uniform distribution inside the volume of the sphere would exhibit the reverse problem. However, by choosing the right volume distribution inside the sphere, we can correctly match the emission of a spherical light.

$$d(\mathbf{x}) = \frac{1}{\pi^2 R^2} \frac{1}{R^2 - r^2(\mathbf{x})} \quad (7)$$

The normalized point distribution $d(\mathbf{x})$ is defined inside a sphere of radius R where $r^2(\mathbf{x})$ is the squared distance from the sphere's center. The beauty of this distribution is that it projects to a uniform distribution across the apparent solid angle of the sphere when viewed from any position outside the sphere, which is exactly the property we need. We also need a way to generate random points according to this distribution. Given three uniformly distributed random numbers, ξ_1, ξ_2, ξ_3 , in the range $[0; 1]$ we can compute points with the right distribution for a sphere centered at the origin using:

$$\begin{aligned} x &= R \sqrt{\xi_1} \cos(2\pi\xi_2) \\ y &= R \sqrt{\xi_1} \sin(2\pi\xi_2) \\ z &= R^2 - x^2 - y^2 \sin(\pi(\xi_3 - 1/2)) \end{aligned} \quad (8)$$

The omni lights generated from spherical lights behave exactly like normal omni lights except that when computing their visibility factors, they can only be occluded by geometry outside the sphere (i.e. their shadow rays terminate at the surface of the sphere).