Reflective Shadow Maps.(反射阴影图。 )

Carsten Dachsbacher[[1]](#footnote-1) Marc Stamminger†.(Carsten Dachsbachermarc Stamminger？ )

University of Erlangen-Nuremberg University of Erlangen-Nuremberg.(埃朗根-纽伦堡大学埃朗根-纽伦堡大学。 )

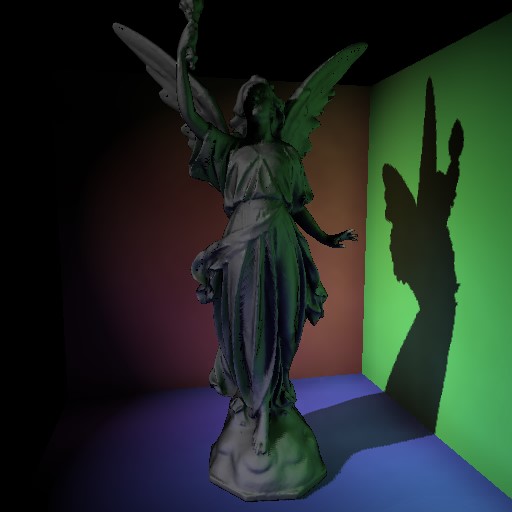
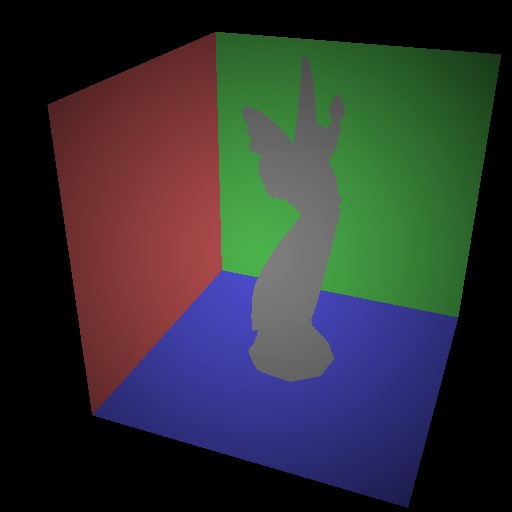
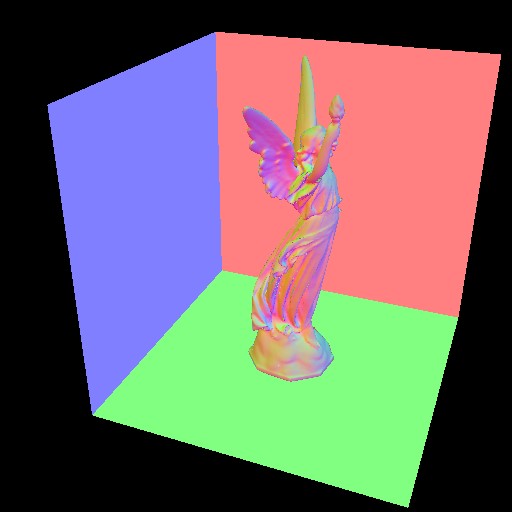
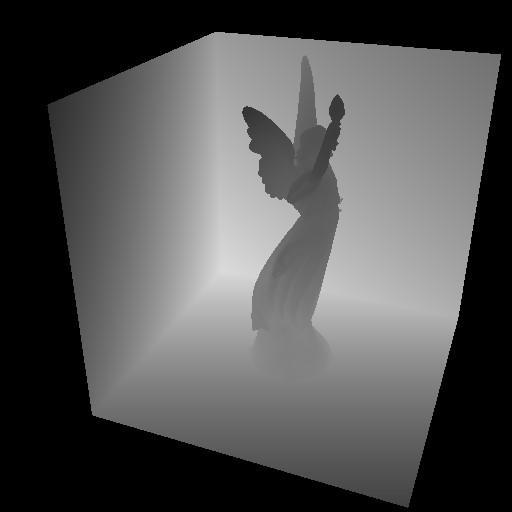


Figure 1: This figure shows the components of the reflective shadow map (depth, world space coordinates, normal, flux) and the resulting image rendered with indirect illumination from the RSM.(图1:该图显示了反射阴影图的组成部分（深度、世界空间坐标、法线、通量），以及使用RSM的间接照明呈现的结果图像。 ) Note that the angular decrease of flux is shown exaggerated for visualization.(图1:该图显示了反射阴影图的组成部分（深度、世界空间坐标、法线、通量），以及使用RSM的间接照明呈现的结果图像。 注意，通量的角度减小被夸大以用于可视化。 )

# Abstract.(摘要。 )

In this paper we present ”reflective shadow maps”, an algorithm for interactive rendering of plausible indirect illumination.(本文提出了一种交互式真实感间接光照渲染算法“反射阴影图”。 ) A reflective shadow map is an extension to a standard shadow map, where every pixel is considered as an indirect light source.(本文提出了一种交互式真实感间接光照渲染算法“反射阴影图”。 反射阴影图是标准阴影图的扩展，其中每个像素都被视为间接光源。 ) The illumination due to these indirect lights is evaluated on-the-fly using adaptive sampling in a fragment shader.(使用碎片着色器中的自适应采样来动态评估这些间接光引起的照明。 ) By using screen-space interpolation of the indirect lighting, we achieve interactive rates, even for complex scenes.(ED作为间接光源。 使用碎片着色器中的自适应采样来动态评估这些间接光引起的照明。 通过使用间接照明的屏幕空间插值，我们实现了交互速率，即使对于复杂场景也是如此。 ) Since we mainly work in screen space, the additional effort is largely independent of scene complexity.(由于我们主要在屏幕空间中工作，因此额外的工作在很大程度上与场景复杂性无关。 ) The resulting indirect light is approximate, but leads to plausible results and is suited for dynamic scenes.(由此产生的间接光是近似的，但会产生可信的结果，并且适合动态场景。 ) We describe an implementation on current graphics hardware and show results achieved with our approach.(场景。 由于我们主要在屏幕空间中工作，因此额外的工作在很大程度上与场景复杂性无关。 由此产生的间接光是近似的，但会产生可信的结果，并且适合动态场景。 我们描述了一个在当前图形硬件上的实现，并展示了用我们的方法实现的结果。 )

CR Categories: I.(CR类别:I。 )3.(3。 )3 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and textureI.([计算机图形学]:三维图形和真实感-颜色，阴影，阴影和纹理。 )3.(3。 )3 [Computer Graphics]: Hardware Architecture—Graphics processors.(I.3.3[计算机图形学]:三维图形和真实感-颜色、阴影、阴影和纹理I.3.3[计算机图形学]:硬件体系结构-图形处理器。 )

Keywords: indirect illumination, hardware-assisted rendering.(关键词:间接照明，硬件辅助渲染。 )

# Introduction.(导言。 )

Interactive computer graphics has developed enormously over the last years, mainly driven by the advance of graphics acceleration hardware.(交互式计算机图形学在过去的几年中得到了长足的发展，这主要得益于图形加速硬件的进步。 ) Scenes of millions of polygons can be rendered in realtime on consumer-level PC cards nowadays.(交互式计算机图形学在过去的几年中得到了长足的发展，这主要得益于图形加速硬件的进步。 现在，数百万个多边形的场景可以在消费级PC卡上实时呈现。 ) Programmability allows the inclusion of sophisticated lighting effects.(可编程性允许包含复杂的照明效果。 ) However, these effects are only simple subcases of global illumination, e.(然而，这些效应仅仅是全局照明的简单子情况，例如。 )g.(例如: ) reflections of distant objects or shadows of point lights.(现已在消费级PC卡上实时注册。 可编程性允许包含复杂的照明效果。 然而，这些效果只是全局照明的简单子情况，例如远处物体的反射或点光源的阴影。 ) Real global illumination, however, generates subtle, but also important effects that are mandatory to achieve realism.(然而，真正的全球照明，产生微妙的，但也是重要的影响，是强制性的，以实现现实主义。 )

Unfortunately, due to their global nature, full global illumination and interactivity are usually incompatible.(不幸的是，由于它们的全球性质，完全的全球照明和交互性通常是不兼容的。 ) Ray Tracing and Radiosity—just to mention the two main classes of global illumination algorithms—require minutes or hours to generate a single image with full global illumination.(不幸的是，由于它们的全球性质，完全的全球照明和交互性通常是不兼容的。 射线跟踪和辐射度——仅提及两类主要的全局照明算法——需要几分钟或几小时来生成具有完全全局照明的单个图像。 ) Recently, there has been remarkable effort to make ray tracing interactive (e.(最近，在使射线追踪交互式（例如， )g.(例如: ) [Wald et al.(辐射度——仅提及两类主要的全局光照算法——需要几分钟或几小时来生成具有完全全局光照的单个图像。 最近，在使射线追踪交互式（例如[Wald et al。 ) 2003]).(2003年])。 ) Compute clusters are necessary to achieve interactivity at good image resolution and dynamic scenes are difficult to handle, because they require to update the ray casting acceleration structures for every frame.(计算集群是实现高分辨率图像交互的必要条件，而动态场景的处理是非常困难的，因为它们需要更新每帧的光线投射加速度结构。 ) Radiosity computation times are even further from interactive.(为了实现高分辨率的交互，计算集群是必要的，而动态场景的处理是困难的，因为它们需要为每一帧更新射线投射加速度结构。辐射度计算时间距离交互甚至更远。 ) Anyhow, a once computed radiosity solution can be rendered from arbitrary view points quickly, but, as soons as objects move, the update of the solution becomes very expensive again.(不管怎样，一次计算出的辐射度解可以从任意视点快速呈现，但是，随着对象移动，解决方案的更新再次变得非常昂贵。 )

It has been observed that for many purposes, global illumination solutions do not need to be precise, but only plausible.(已经观察到，对于许多目的，全局照明解决方案不需要是精确的，而仅仅是看似合理的。 ) In this paper, we describe a method to compute a rough approximation for the one-bounce indirect light in a scene.(本文描述了一种计算场景中一次反射光近似的方法。 ) Our method is based on the idea of the shadow map.(已经观察到，对于许多目的，全局照明解决方案不需要是精确的，而仅仅是看似合理的。 本文描述了一种计算场景中一次反射光近似的方法。 我们的方法是基于阴影映射的思想。 ) In a first pass, we render the scene from the view of the light source (for now, we assume that we have only one spot or parallel light source in our scene).(在第一遍中，我们从光源的角度渲染场景（现在，我们假设在我们的场景中只有一个点光源或平行光源）。 ) The resulting depth buffer is called *shadow map*, and can be used to generate shadows.(关于影子地图的想法。 在第一遍中，我们从光源的角度渲染场景（现在，我们假设在我们的场景中只有一个点光源或平行光源）。 生成的深度缓冲区称为阴影映射，可用于生成阴影。 ) In a *reflective shadow map*, with every pixel, we additionally store the light reflected off the hit surface.(o生成阴影。 在反射阴影图中，每一个像素，我们都额外存储从击中的表面反射回来的光。 ) We interpret each of the pixels as a small area light source that illuminates the scene.(我们将每个像素解释为照亮场景的小区域光源。 ) In this paper, we describe how the illumination due to this large set of light sources can be computed efficiently and coherently, resulting in approximate, yet plausible and coherent indirect light.(，每增加一个像素，我们就会存储从击中表面反射回来的光。 我们将每个像素解释为照亮场景的小区域光源。 在本文中，我们描述了如何能够有效地和相干地计算由这一大组光源引起的照明，从而产生近似的、但合理的和相干的间接光。 )

# Previous Work.(以前的工作。 )

Shadow maps [Williams 1978; Reeves et al.(阴影图[Williams1978；Reeves et al。 ) 1987] and shadow volumes [Crow 1977] are the standard shadowing algorithms for interactive applications.(阴影图[Williams，1978；Reeves等，1987]和阴影体积[Crow，1977]是交互式应用程序的标准阴影算法。 ) Recently, there have been extensions of both approaches to area lights [Assarsson and Akenine-Moller 2003;¨ Chan and Durand 2003; Wyman and Hansen 2003].(最近，这两种方法都扩展到了区域照明[Assarsson和Akenine-Moller，2003年；échan和Durand，2003年；Wyman和Hansen，2003年]。 ) Sometimes, such soft shadows are already referred to as ‘global illumination’.(re交互式应用程序的标准阴影算法。 最近，这两种方法都扩展到了区域照明[Assarsson和Akenine-Moller，2003年；échan和Durand，2003年；Wyman和Hansen，2003年]。 有时，这种柔和的阴影已经被称为“全局光照”。 ) In this paper, we concentrate on indirect illumination from point lights, but our approach can easily be combined with any of these soft shadow techniques.(在这篇论文中，我们主要研究点光源的间接照明，但是我们的方法可以很容易地与这些软阴影技术相结合。 )

Others generate global illumination images at interactive rates, but they rely on costly precomputations and are thus not suited for dynamic scenes [Walter et al.(另一些以交互速率生成全局照明图像，但是它们依赖于昂贵的预先计算，因此不适合动态场景[Walter等人。 ) 1997; Sloan et al.(1997年； Sloan等人 ) 2002; Bala et al.(2002年； Bala等人 ) 2003].(2003年]。 ) In Instant Radiosity [Keller 1997], the indirect light is represented by a set of point lights, the contributions of which are gathered using many rendering passes.(e比率，但它们依赖于昂贵的预先计算，因此不适合动态场景[Walter等人，1997年；Sloan等人，2002年；Bala等人，2003年]。 在瞬时辐射度[Keller1997]中，间接光由一组点光源表示，其贡献通过许多渲染通道收集。 ) With Instant Radiosity, dynamic objects and lights are possible, but many rendering passes are required.(有了瞬时的辐射度，动态对象和光是可能的，但是需要许多渲染过程。 )

Interactive ray tracing can generate global illumination effects at interactive frame rates [Wald et al.(交互式光线追踪可以以交互式帧速率产生全局照明效果[Wald等人。 ) 2001b; Wald et al.(2001年b； Wald等人 ) 2001a; Wald et al.(2001年a； Wald等人 ) 2002; Wald et al.(2002年； Wald等人 ) 2003].(2003年]。 ) These approaches still require clusters with several PCs to achieve interactivity at high resolution.(这些方法仍然需要具有多个PC的集群来实现高分辨率的交互性。 )

Our approach combines ideas from two previous publications.(我们的方法结合了以前两个出版物中的思想。 ) In [Tabellion and Lamorlette 2004], a global illumination method for offline film production rendering is presented.(我们的方法结合了以前两个出版物中的思想。 在[Tabellion and Lamorlette2004]中，提出了一种用于离线胶片生产渲染的全局光照方法。 ) The authors demonstrate that one-bounce indirect illumination is sufficient in many cases.(作者证明，在许多情况下，一次反弹的间接照明是足够的。 ) They generate a texture atlas containing the direct light, and then gather the first bounce indirect light from this texture.(Lamorlette2004]提出了一种离线胶片生产渲染的全局光照方法，证明了在许多情况下，一次反弹的间接光照是足够的，它们生成包含直接光线的纹理图谱，然后从该纹理中收集第一次反弹的间接光线。 ) The second method we build upon are Translucent Shadow Maps [Dachsbacher and Stamminger 2003].(第二种方法是半透明的阴影地图[Dachsbacher和Stamminger2003]。 ) In this paper, a shadow map is extended such that all pixels in a shadow map are considered as subsurface light sources.(从这个纹理收集第一道反弹的间接光线。 第二种方法是半透明的阴影地图[Dachsbacher和Stamminger2003]。 本文对阴影图进行了扩展，使得阴影图中的所有像素都被看作是次表面光源。 ) By gathering their contribution, translucent lighting can be approximated.(通过收集他们的贡献，半透明的照明可以近似。 )

# Reflective Shadow Maps.(反射阴影图。 )

Reflective Shadow Maps (RSMs) combine the ideas of [Tabellion and Lamorlette 2004] and [Dachsbacher and Stamminger 2003].(反射阴影图结合了[Tabellion and Lamorlette2004]和[Dachsbacher and Stamminger2003]的思想。 ) The idea is that we consider all pixels of a shadow map as indirect light sources that generate the one-bounce indirect illumination in a scene.(反射阴影图结合了[Tabellion and Lamorlette2004]和[Dachsbacher and Stamminger2003]的思想。 这个想法是，我们把阴影地图的所有像素看作是在场景中产生一次反弹的间接照明的间接光源。 ) This idea is based on the observation, that if we have a single point light source, all one-bounce indirect illumination is caused by surfaces visible in its shadow map.(t我们将阴影图的所有像素视为在场景中产生一次反弹间接照明的间接光源。 这种想法是基于这样的观察，即如果我们有一个单点光源，所有一次反弹的间接照明都是由其阴影图中可见的表面引起的。 ) So in this case, the shadow map contains all information about the indirect lighting, and no radiosity texture atlas as in [Tabellion and Lamorlette 2004] is needed.(因此，在这种情况下，阴影地图包含所有关于间接照明的信息，并且不需要像[Tabellion and Lamorlette2004]中那样的辐射度纹理地图集。 ) Thus Reflective Shadow Maps are more similar to Translucent Shadow Maps [Dachsbacher and Stamminger 2003], where the pixels of a.(用于阴影图中可见的表面。 因此，在这种情况下，阴影地图包含所有关于间接照明的信息，并且不需要像[Tabellion and Lamorlette2004]中那样的辐射度纹理地图集。 因此，反射阴影图更类似于半透明阴影图[Dachsbacher和Stamminger2003]，其中a。 )

shadow map are also considered as point lights.(阴影地图也被认为是点光源。 )

In the following, we describe what we exactly store in a reflective shadow map (Sect.(在下面，我们将描述我们确切地存储在反射阴影图中的内容（Sect。 ) 3.(3。 )1), how a reflective shadow map is generated (Sect.(在下面，我们将描述我们在反射阴影图中到底存储了什么（第3.1节），反射阴影图是如何生成的（第3.1节）。 ) 3.(3。 )2), and how the indirect illumination can be evaluated from it (Sect.(2），以及如何从它来评估间接照明（第二节）。 ) 3.(3。 )3).(3）。 ) Since the indirect light evaluation is expensive, we introduce in Sect.(由于间接光评价是昂贵的，我们在第三节介绍。 ) 4 a screen-space interpolation method that reduces the number of evaluations and leads to interactive display rates.(Nerated（第3.2节），以及如何评估间接照明（第3.3节）。 由于间接光评价是昂贵的，我们在第三节介绍。 4。一种屏幕空间内插方法，其减少了评估的数量并导致交互显示速率。 )

## Data.(数据。 )

We assume that all surfaces in the scene are diffuse reflectors.(我们假设场景中的所有表面都是漫反射器。 ) An RSM stores with every pixel *p* the depth value *dp*, the world space position *xp*, the normal *np*, and the reflected radiant flux Φ*p* of the visible surface point (see Fig.(我们假设场景中的所有表面都是漫反射器。 RSM与每个像素P一起存储可见表面点的深度值Dp、世界空间位置Xp、法线Np和反射辐射通量Φp（见图1）。 )1).(1）。 ) Every pixel is interpreted as a *pixel light* that illuminates the scene indirectly.(可见表面点（见图1）。 每个像素都被解释为间接照亮场景的像素光。 ) The world space position could be recomputed from the pixel coordinates and the depth value, however we can save valuable pixel shader instructions by having the world space positions directly available.(间接地照亮了场景。 世界空间位置可以从像素坐标和深度值重新计算，但是我们可以通过直接获得世界空间位置来保存有价值的像素着色器指令。 ) The flux Φ*p* defines its brightness, and the normal *np* its spatial emission characteristics (see Fig.(DER指令由世界空间位置直接可用。 通量Φp定义了它的亮度，法向Np定义了它的空间发射特性（见图。 ) 2).(2）。 ) If we assume that the light source is infinitely small, we can describe the radiant intensity emitted into direction *ω* as.(其空间发射特性（见图2）。 如果我们假设光源是无限小的，我们可以描述发射到方向ωas的辐射强度。 )

*Ip**,* where h|i is the dot product.(IP，其中HI是点积。 )

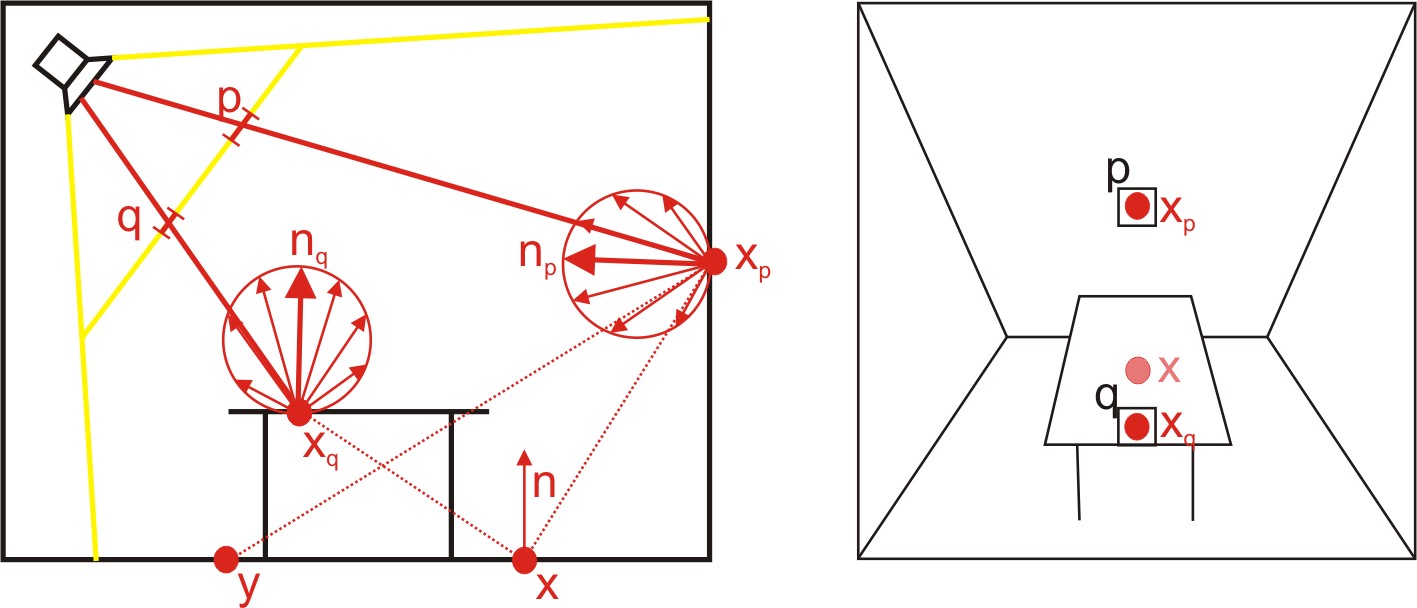
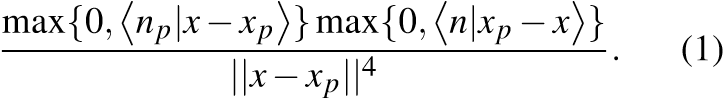


Figure 2: Two indirect pixel lights *xp* and *xq* corresponding to two RSM pixels *p* and *q.(图2:对应于两个RSM像素P和Q的两个间接像素光XP和XQ。 )*

The irradiance at a surface point *x* with normal *n* due to pixel light *p* is thus:.(因此，像素光P引起的具有法线n的表面点x处的辐照度为:。 )

*Ep*(*x,n*)=Φ*p.(Ep（x，n）=Φp。 )*

Note that we intentionally decided to store radiant flux instead of radiosity or radiance.(注意，我们故意决定存储辐射通量而不是辐射度或辐射度。 ) By this, we don’t have to care about the representative area of the light, which makes the generation and the evaluation simpler.(这样，我们就不必关心光的代表区域，这使得生成和评估更加简单。 )

## Generation.(一代人。 )

An RSM is generated just like a standard shadow map, but with multiple render targets: additionally to the depth buffer storing the depth values *dp*, we generate a normal buffer and a world space position buffer with the *np* and *xp*, and a flux buffer storing Φ*p*.(一个RSM就像一个标准的阴影地图，但是有多个渲染目标:除了存储深度值DP的深度缓冲区外，我们还生成一个正常缓冲区和一个带有NP和XP的世界空间位置缓冲区，以及一个存储ΦP的通量缓冲区。 ) Since Φ*p* stores the reflected flux, its computation is simple.(由于Φp存储了反射通量，因此计算简单。 ) First, we have to compute the flux emitted through every pixel.(首先，我们必须计算通过每个像素发射的通量。 ) For a uniform parallel light, this is a constant value.(对于均匀平行光，这是一个常数值。 ) For a uniform spot light, this flux decreases with the cosine to the spot direction due to the decreasing solid angle.(存储反射通量，计算简单。 首先，我们必须计算通过每个像素发射的通量。 对于均匀平行光，这是一个常数值。 对于均匀的聚光灯，由于立体角的减小，该通量随着光斑方向的余弦而减小。 ) The reflected flux is then the flux through the pixel times the reflection coefficient of the surface.(然后，反射通量是通过像素的通量乘以表面的反射系数。 ) No distance attenuation or receiver cosine must be computed.(不得计算距离衰减或接收器余弦。 ) As a result, the flux buffer looks like an unshaded image (compare Fig.(由于立体角减小，光斑方向。 然后，反射通量是通过像素的通量乘以表面的反射系数。 不得计算距离衰减或接收器余弦。 结果，通量缓冲器看起来像未着色的图像（比较图1）。 ) 1, 4th column).(1，第4栏）。 )

As it is typical for many global illumination algorithms, problems appear along the common boundary of two walls.(由于它是许多全局光照算法的典型，问题沿着两个墙的公共边界出现。 ) In this case, the illumination integral has a singularity, which is difficult to integrate numerically.(由于它是许多全局光照算法的典型，问题沿着两个墙的公共边界出现。 在这种情况下，光照积分具有奇异性，这是难以数值积分。 ) We found that these problems can be largely reduced, if we move the pixel lights in negative normal direction by some constant offset.(我们发现，如果将像素光沿负法线方向移动一定的恒定偏移量，这些问题就可以大大减少。 ) This is possible, because we do not consider occlusion for indirect illumination (see below).(这是可能的，因为我们不考虑遮挡间接照明（见下文）。 )

## Evaluation.(评估。 )

The indirect irradiance at a surface point *x* with normal *n* can be approximated by summing up the illumination due to all pixel lights:.(法线为n的表面点x处的间接辐照度可以通过求和所有像素光引起的照度来近似:。 )

*E*(*x,n*)= ∑ *Ep*(*x,n*) (2).(E（x，n)=∑Ep(x，n)(2)。 )

pixels*p.(像素p。 )*

Consider as example Fig.(考虑作为示例图。 ) 2.(2。 ) A spot light illuminates a room with a table from the upper left.(聚光灯从左上角用一张桌子照亮房间。 ) For a particular light view pixel *p*, we have a pixel light source at position *xp*, illuminating the scene.(考虑作为示例图。 聚光灯从左上角用一张桌子照亮房间。 对于特定的光视图像素p，我们在位置xp处有一个像素光源，照亮场景。 )

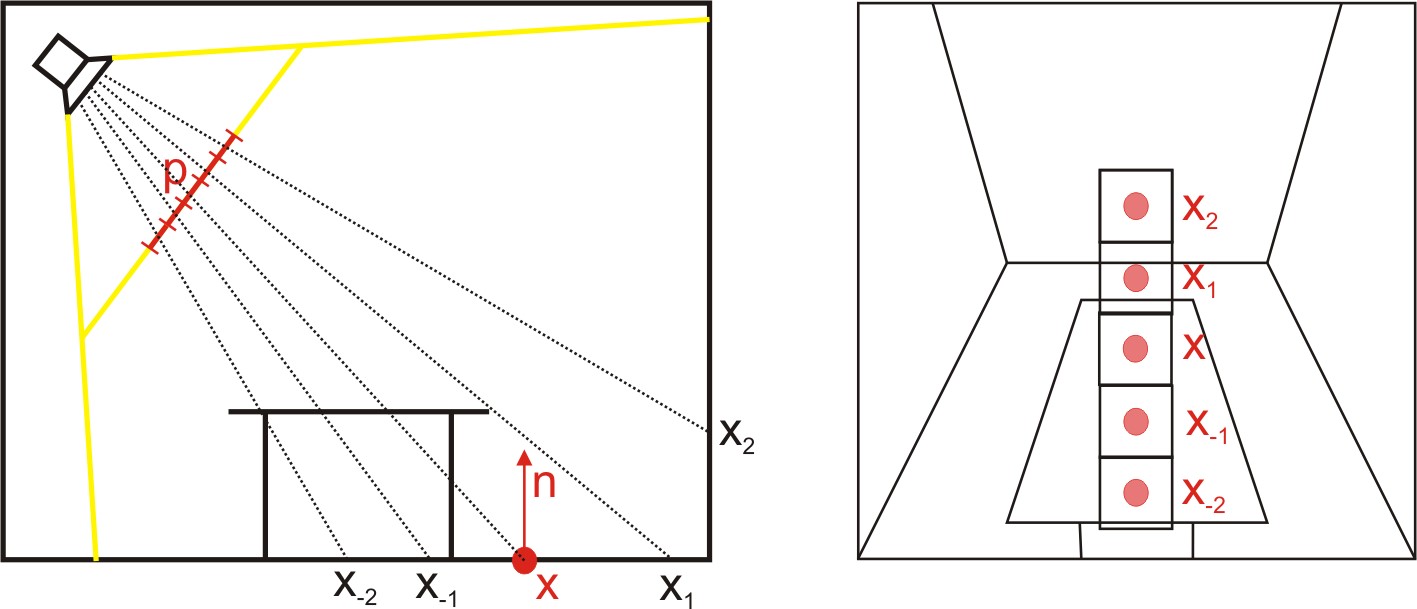


Figure 3: RSM sampling.(图3:RSM采样。 )

along its normal *np*.(沿着正常的NP。 ) The point *x* on the floor receives light from *xp* according to Equ.(。地板上的X点根据EQU接收来自XP的光。 ) 2.(2。 ) For pixel *q*, the pixel light lies on the table and thus does not illuminate *x*.(根据EQU。 2。对于像素Q，像素光位于桌子上，因此不照亮X。 )

Note that we do not consider occlusion for the indirect light sources, so that in Fig.(注意，对于间接光源，我们不考虑遮挡，因此在图1中。 ) 2 *y* is indirectly illuminated by *xp*, although *xp* is not visible from *y*.(注意，对于间接光源，我们不考虑遮挡，因此在图1中。 2Y被XP间接照亮，尽管XP在Y上不可见。 ) This is a severe approximation, and can lead to very wrong results.(。这是一个严重的近似值，可能导致非常错误的结果。 ) However, in many cases it suffices to generate the subtle indirect lighting effects; unprecise results are often acceptable, as long as the indirect lighting effect is visible.(然而，在许多情况下，只要产生微妙的间接照明效果就足够了； 只要间接照明效果是可见的，不精确的结果通常是可以接受的。 )

For a typical shadow map, the number of pixels is large (512x512), so the evaluation of the above sum is very expensive and not practical in a realtime context.(对于典型的阴影图，像素的数量很大(512×512)，因此上面的和的评估非常昂贵，并且在实时上下文中不实用。（ ) Instead, we have to reduce the sum to a restricted number of light sources, e.(相反，我们必须把总数减少到有限的光源数目，例如。 )g.(例如: ) 400.(400美元。 ) We do this using an importance-driven approach, where we try to concentrate the sampling to the relevant pixel lights.(对于典型的阴影图，像素的数量很大(512×512)，因此上面的和的评估非常昂贵，并且在实时上下文中不实用。（ 相反，我们必须把总数减少到有限的光源数目，例如400个。 我们使用一种重要性驱动的方法来实现这一点，在这种方法中，我们尝试将采样集中到相关的像素光。 ) The idea can be best described for the example in Fig.(对于图1中的示例，可以最好地描述该想法。 ) 3.(3。 ) *x* is not directly illuminated, so it is not visible in the shadow map.(S使用一种重要性驱动的方法，我们尝试将采样集中到相关的像素光。 对于图1中的示例，可以最好地描述该想法。 3。X没有被直接照亮，因此在阴影图中不可见。 ) If we project *x* into the shadow map, the pixel lights that are closest in world space are also close in the shadow map.(没有直接照明，因此在阴影图中不可见。 如果我们把x投射到阴影图中，那么在世界空间中最近的像素光在阴影图中也是最近的。 ) *x*−2 and *x*−1 are relatively close, but since their normal points away from *x*, they do not contribute indirect illumination.(n对于阴影图，在世界空间中最近的像素光也在阴影图中接近。 x−2和x−1相对较近，但由于它们的法线点远离x，因此不会产生间接照明。 ) *x*1 is very close, but lies on the same plane (floor) as *x*, so it also does not contribute.(，它们不促进间接照明。 X1非常接近，但与X位于同一平面（地板），因此它也不起作用。 ) The most relevant pixel light is *x*2.(，所以它也不起作用。 最相关的像素光是x2。 )

In general, we can say that the distance between *x* and a pixel light *xp* in the shadow map is a reasonable approximation for their distance in world space.(一般来说，我们可以说阴影图中x和像素光xp之间的距离是它们在世界空间中距离的合理近似值。 ) If the depth values with respect to the light source differ significantly, the world space distance is much bigger and we thus overestimate the influence.(在阴影图中是对它们在世界空间中距离的合理近似。 如果相对于光源的深度值显著不同，则世界空间距离更大，因此我们高估了影响。 ) However, the important indirect lights will always be close, and these must also be close in the shadow map.(相对于光源的深度值差异很大，世界空间距离大得多，因此我们高估了这种影响。 然而，重要的间接光线将永远是近的，这些也必须是近的阴影地图。 )

So we decided to obtain the pixel light samples as follows: first, we project *x* into the shadow map (→ (*s,t*)).(因此，我们决定如下获得像素光样本:首先，我们将X投影到阴影映射(→(s，t））中。 ) We then select pixel lights around (*s,t*), where the sample density decrease with the squared distance to (*s,t*).(然后，我们选择（s，t）周围的像素光，其中样本密度随着到（s，t）的平方距离而减小。） ) This can easily be achieved by selecting the samples in polar coordinates relative to (*s,t*), i.(这可以很容易地通过选择相对于（s，t）的极坐标中的样本来实现，即。 )e.(e。 ) if *ξ*1 and *ξ*2 are uniformly distributed random numbers, we select the pixel light at position.(即，如果ξ1和ξ2是均匀分布的随机数，则选择位置处的像素光。 )

(*s*+*r*max*ξ*1 sin(2*πξ*2)*,t* +*r*max*ξ*1 cos(2*πξ*2))*.(（S rmaxξ1sin(2πξ2)，T rmaxξ1cos(2πξ2))。 )* (3).(。（3）。 )

We then have to compensate the varying sampling density by weighting the achieved samples with *ξ*12 (and a final normalization).(然后，我们必须通过用ξ12加权所获得的样本（和最终的归一化）来补偿变化的采样密度。 ) An example sampling pattern is shown in Fig.(示例性采样模式示于图1中。 ) 4.(4。 )

In our implementation, we precompute such a sampling pattern and reuse it for all indirect light computations, which gives us coherency.(在我们的实现中，我们预先计算这样的采样模式，并将其重用到所有间接光计算中，这给我们带来了相干性。 ) This temporal coherency reduces flickering in dynamic scenes, however the spatial coherence can result in banding artifacts if the number of samples is not large enough.(这种时间相干性减少了动态场景中的闪烁，然而，如果样本的数量不够大，则空间相干性可能导致条带伪影。 ) In our example scenes, 400 samples were sufficient.(l间接光计算，这给我们提供了相干性。 这种时间相干性减少了动态场景中的闪烁，然而，如果样本的数量不够大，则空间相干性可能导致条带伪影。 在我们的示例场景中，400个样本就足够了。 ) Because we use a once computed sampling pattern, we could use Poisson sampling to obtain a more even sample distribution.(因为我们使用一次计算的采样模式，所以我们可以使用泊松采样来获得更均匀的样本分布。 )

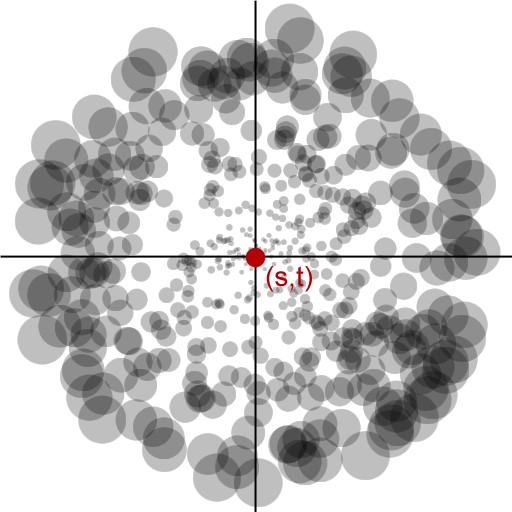


Figure 4: Sampling pattern example.(图4:采样模式示例。 ) The sample density decreases and the sample weights (visualized by the disk radius) increases with the distance to the center.(图4:采样模式示例。 随着到中心的距离，样本密度减小，样本重量（通过盘半径可视化）增大。 )

# Screen-Space Interpolation.(屏幕空间插值 )

We compute the direct illumination using per-pixel lighting, where the RSM is bound as standard shadow map.(我们使用每个像素的光照来计算直接照明，其中RSM被绑定为标准阴影图。 ) The indirect lighting computation as described above is still too expensive to be performed for every pixel in an interactive application.(我们使用每个像素的光照来计算直接照明，其中RSM被绑定为标准阴影图。 如上所述的间接照明计算对于交互应用中的每个像素来说仍然太昂贵。 ) However, with a simple interpolation scheme we can drastically reduce the number of evaluations and use cheap interpolation for the majority of the pixels.(对交互式应用程序中的每个像素执行Xpensive。 然而，通过简单的插值方案，我们可以大幅减少求值的次数，并对大多数像素使用廉价的插值。 )

In a first pass, we compute the indirect illumination for a lowresolution image of the camera view.(在第一步中，我们计算相机视图的低分辨率图像的间接照度。 ) We then render the full resolution camera view and check for every pixel, whether the indirect light can be interpolated from the four surrounding low-res samples.(然后，我们渲染全分辨率相机视图，并检查每个像素，间接光是否可以从周围的四个低分辨率采样插值。 ) Such a low-res sample is regarded as suitable for interpolation if the sample’s normal is similar to the pixel’s normal, and if its world space location is close to the pixel’s location.(相机视图的低分辨率图像的间接照明。 然后，我们渲染全分辨率相机视图，并检查每个像素，间接光是否可以从周围的四个低分辨率采样插值。 如果样本的法线与像素的法线相似，并且如果其世界空间位置接近像素的位置，则认为这样的低分辨率样本适合于插值。 ) Each sample’s contribution is weighted by the factors used for bi-linear interpolation, including a normalization if not all four samples are used.(如果样本的法线与像素的法线相似，并且其世界空间位置接近像素的位置，则E被认为适合于插值。 每个样本的贡献由用于双线性插值的因子加权，如果不使用所有四个样本，则包括归一化。 ) If three or four samples are considered as suitable, we interpolate the indirect illumination.(如果认为三个或四个样本是合适的，我们内插间接照明。 ) Otherwise, we discard the pixel in this render pass and rather compute the indirect illumination with the complete gathering step in a final pass.(如果不是所有四个样本都使用，则包括归一化。 如果认为三个或四个样本是合适的，我们内插间接照明。 否则，我们在这个渲染过程中丢弃像素，而是在最后一个过程中使用完整的收集步骤来计算间接照明。 )

Fig.(无花果。 ) 5 shows the effectiveness of our solution.(5显示了我们解决方案的有效性。 ) Only for the red pixels, interpolation was not sufficient to compute the indirect light.(仅对于红色像素，插值不足以计算间接光。 ) Of course, this effectiveness depends on the scene.(当然，这种效果取决于场景。 ) On smooth surfaces, interpolation works very well, whereas on unconnected, complex geometry, such as a tree, the interpolation scheme will not be applicable, so that the method falls back to full evaluation.(无花果。 5显示了我们解决方案的有效性。 仅对于红色像素，插值不足以计算间接光。 当然，这种效果取决于场景。 在光滑曲面上，插值效果很好，而在非连通的复杂几何体（如树）上，插值方案将不适用，因此该方法返回到完全求值。 )

# Implementation.(执行情况。 )

We implemented our approach using Direct3D9 with Microsoft’s High Level Shader Language and contemporary graphics hardware supporting floating point render targets and programmable vertex and fragment processing.(我们使用Direct3D9和Microsoft的高级着色器语言以及支持浮点渲染目标和可编程顶点及片段处理的当代图形硬件实现了我们的方法。 ) GPUs providing Pixel Shaders 2.(提供像素着色器的图形处理器2。 )0 are sufficient for an implementation of our method, although we used Pixel Shader 3.(OFT的高级着色语言和现代图形硬件支持浮点渲染，目标和可编程顶点和片段处理。 尽管我们使用了像素着色器3，但是提供像素着色器2.0的GPU对于我们的方法的实现是足够的。 )0 hardware (supporting floating point blending natively) for measuring our results.(0用于测量结果的硬件（本机支持浮点混合）。 )

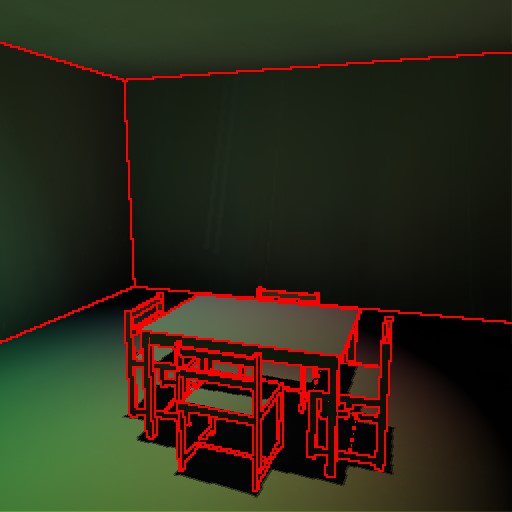


Figure 5: Efficiency of screen space interpolation: only for the red pixels no interpolation was reasonable.(图5:屏幕空间插值的效率:只有红色像素没有插值是合理的。 )

The generation of the final image consists of multiple render passes, most of which are required for the indirect lighting computation.(最终图像的生成由多个渲染通道组成，其中大部分是间接光照计算所需的。 ) In an initial pass, the scene is rendered from the light’s view.(在最初的过程中，场景是从光线的视角渲染出来的。 ) But instead of storing only the depth value to the first visible surface through each pixel—like normal shadow maps—we also store the surface normals, its world space location and the energy flux.(最终图像的生成由多个渲染通道组成，其中大部分是间接光照计算所需的。 在最初的过程中，场景是从光线的视角渲染出来的。 但是，我们也存储了表面法线、它的世界空间位置和能量流，而不是只存储通过每个像素的法线阴影图到第一个可见表面的深度值。 ) Although the world space location could be recomputed from the depth value, we chose to store it in the render targets (three in total) to save per-pixel instructions in subsequent render passes.(通过每个像素点的法线阴影图，我们还存储了曲面的法线、它的世界空间位置和能量流。 虽然可以从深度值重新计算世界空间位置，但我们选择将其存储在呈现目标中（总共三个），以便在后续呈现过程中保存每个像素的指令。 )

In order to decouple scene complexity from rendering performance as far as possible and to prevent the execution of complex shaders for hidden surfaces, we apply the direct and indirect illumination as a deferred shading process.(为了尽可能地将场景复杂度与渲染性能解耦，避免对隐藏表面执行复杂的着色器，我们采用直接和间接光照作为延迟着色过程。 ) We generate textures storing the required information per pixel, namely world space location and normal, its corresponding coordinates in light space and the material parameters.(我们生成纹理存储每个像素所需的信息，即世界空间位置和法线，它在光空间的对应坐标和材料参数。 ) These textures are view-dependent and need to be updated when the camera moves.(间接照明作为一种延迟的遮光过程。 我们生成纹理存储每个像素所需的信息，即世界空间位置和法线，它在光空间的对应坐标和材料参数。 这些纹理是视图相关的，需要在相机移动时更新。 )

The next step is the gathering of the indirect illumination.(下一步是收集间接照明。 ) This requires sampling of the RSM at many different locations.(这需要在许多不同位置对RSM进行采样。 ) For each sample location the illumination is computed as in Equ.(对于每个样本位置，照度按等式计算。 ) 1 and the contributions are accumulated.(下一步是收集间接照明。 这需要在许多不同位置对RSM进行采样。 对于每个样本位置，照度按等式计算。 1，并累计贡献。 ) When using Pixel Shader 2.(使用像素着色器2时。 )0 GPUs, instructions and texture lookups per shader are limited and the computation has to be split up to distinct render passes and intermediate results need to be accumulated.(e累计缴款。 当使用像素着色器2.0GPU时，每个着色器的指令和纹理查找受到限制，并且必须将计算分割成不同的渲染过程，并且需要累积中间结果。 ) In this case, the sampling positions may be provided by the main application via shader constants.(需要积累中期结果。 在这种情况下，采样位置可以由主应用经由着色器常数提供。 ) Applying GPUs supporting PS3.(应用支持PS3的GPU。 )0, all sampling can be done in a single loop, where the sample positions are provided in a lookup texture.(0中，所有采样都可以在单个循环中完成，其中，采样位置以查找纹理的形式提供。 ) Both options achieve comparable results, but the latter is faster.(采样位置可由主应用经由着色器常数提供。 应用支持PS3.0的GPU，所有采样都可以在单个循环中完成，其中样本位置在查找纹理中提供。 这两种选择都取得了可比的结果，但后者速度更快。 )

We also implemented the screen-space interpolation scheme as described above.(我们还实现了如上所述的屏幕空间插值方案。 ) First, a low-resolution image is computed with full evaluation of the indirect lighting.(首先，在充分评价间接照明的基础上，计算出低分辨率的图像。 ) Then, a full resolution image is rendered, where a pixel shader evaluates the number of low-ressamples suitable for interpolation and performs the interpolation if three or four samples are suitable.(我们还实现了如上所述的屏幕空间插值方案。 首先，在充分评价间接照明的基础上，计算出低分辨率的图像。 然后，呈现全分辨率图像，其中像素着色器评估适合于内插的低重采样的数目，并且如果三个或四个采样合适，则执行内插。 ) After this, our image is largely finished, but contains some pixels for which we still have to compute the indirect illumination.(R评估适合于插值的低重采样数，并且如果三个或四个采样合适，则执行插值。 在此之后，我们的图像基本上完成，但包含一些像素，我们仍然需要计算间接照明。 ) This is done with the same Pixel Shaders as used for the low-res image.(这是使用与低分辨率图像相同的像素着色器完成的。 ) To avoid the recomputation of all pixels, early z-culling techniques could be used where available.(为了避免重新计算所有像素，可以在可用的情况下使用早期的Z剔除技术。 ) We decided to render the two final render passes as a grid of quadrilaterals on the screen and to use an occlusion query for each quad.(照明。 这是使用与低分辨率图像相同的像素着色器完成的。 为了避免重新计算所有像素，可以在可用的情况下使用早期的Z剔除技术。 我们决定在屏幕上以四边形网格的形式呈现这两个最终呈现过程，并对每个四边形使用遮挡查询。 ) This tells us, which quadrilaterals are complete reconstructed (all pixels rendered) in the first step and which contain discarded pixels which need accurate computation of the indirect illumination.(并对每个四元组使用遮挡查询。 这告诉我们，哪些四边形在第一步中被完全重建（所有像素被渲染），哪些包含被丢弃的像素，这些像素需要间接照明的精确计算。 )

# Results.(结果。 )

We implemented our approach using an ATI Radeon 9700 card for testing the Pixel Shader 2.(我们使用ATI Radeon9700卡实现了我们的方法，用于测试像素着色器2。 )0 version and a GeForce Quadro FX4000 for the PS3.(0版本和用于PS3的GeForce Quadro FX4000。 )0 code.(0代码。 ) Table 1 shows the result of our implementation of the Reflective Shadow Map method using a resolution of 512x512 for the reflective shadow map and the camera view.(我们使用ATI Radeon9700卡来测试Pixel Shader2.0版本，并使用GeForce Quadro FX4000来测试PS3.0代码。 表1显示了我们使用512x512分辨率的反射阴影图和相机视图实现反射阴影图方法的结果。 ) We measured the difference in performance for various screen subsamplings and number of sample taps (the amount is determined empirically).(我们测量了不同屏幕子采样和样本抽头数量（数量根据经验确定）的性能差异。 ) As it can be taken from the table, the performance does not only depend on the total number of computed indirect illumination values, but also suffers from pipeline stalls through the occlusion query.(512用于反射阴影图和照相机视图。 我们测量了不同屏幕子采样和样本抽头数量（数量根据经验确定）的性能差异。 从表中可以看出，性能不仅取决于计算的间接照明值的总数，而且还受到阻塞查询导致的管道阻塞的影响。 ) All timings were collected on a Pentium 4 processor with 2.(所有计时器都是在带有2的Pentium4处理器上收集的。 )4GHz and the before mentioned GeForce card.(4GHz和前面提到的GeForce卡。 ) Note that we recompute the light and camera view for each frame, that is, the light source and camera can be moved freely and interactively by the user.(NLY依赖于计算出的间接光照值的总数，而且还受管道阻塞的影响通过阻塞查询。 所有计时器都是在2.4GHz的奔腾4处理器和前面提到的GeForce卡上收集的。 请注意，我们重新计算每个帧的光和相机视图，也就是说，光源和相机可以由用户自由地和交互地移动。 ) The impact on the image quality for various settings is shown in Figure 7.(不同设置对图像质量的影响如图7所示。 ) Figure 8 shows the contribution of one-bounce indirect illumination to the table test scene.(重新计算每个帧的光和相机视图，即，光源和相机可以由用户自由和交互地移动。 不同设置对图像质量的影响如图7所示。 图8显示了一次反弹间接照明对表测试场景的贡献。 )

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | indirect.(间接的。 ) | low-res.(低分辨率。 ) | screen.(屏幕。 ) |
| scene.(场景。 ) | fps.(平安险。 ) | samples.(样品。 ) | computed.(算了。 ) | first pass.(第一关。 ) | quads.(四分卫。 ) |
| table.(桌子。 ) | 24.(24。 )2.(2。 ) | 112.(112。 ) | 63299.(63299。 ) | 32 × 32.(32×32。 ) | 32 × 32.(32×32。 ) |
|  | 14.(14。 )3.(3。 ) | 224.(224。 ) | 63299.(63299。 ) | 32 × 32.(32×32。 ) | 32 × 32.(32×32。 ) |
|  | 22.(22。 )0.(0。 ) | 112.(112。 ) | 66120.(66120。 ) | 64 × 64.(64×64。 ) | 32 × 32.(32×32。 ) |
|  | 13.(13。 )0.(0。 ) | 224.(224。 ) | 66120.(66120。 ) | 64 × 64.(64×64。 ) | 32 × 32.(32×32。 ) |
|  | 27.(27。 )5.(5。 ) | 112.(112。 ) | 92745.(92745。 ) | 32 × 32.(32×32。 ) | 16 × 16.(16×16。 ) |
|  | 15.(15。 )9.(9。 ) | 224.(224。 ) | 92745.(92745。 ) | 32 × 32.(32×32。 ) | 16 × 16.(16×16。 ) |
| lucy.(露西。 ) | 19.(19。 )3.(3。 ) | 112.(112。 ) | 77613.(77613。 ) | 32 × 32.(32×32。 ) | 32 × 32.(32×32。 ) |
|  | 11.(11。 )4.(4。 ) | 224.(224。 ) | 77613.(77613。 ) | 32 × 32.(32×32。 ) | 32 × 32.(32×32。 ) |
|  | 15.(15。 )8.(8。 ) | 112.(112。 ) | 82031.(第82031号。 ) | 64 × 64.(64×64。 ) | 32 × 32.(32×32。 ) |
|  | 9.(9。 )1.(1。 ) | 224.(224。 ) | 82031.(第82031号。 ) | 64 × 64.(64×64。 ) | 32 × 32.(32×32。 ) |
|  | 13.(13。 )3.(3。 ) | 112.(112。 ) | 116320.(116320。 ) | 64 × 64.(64×64。 ) | 16 × 16.(16×16。 ) |
| droids.(机器人。 ) | 15.(15。 )1.(1。 ) | 112.(112。 ) | 98048.(98048。 ) | 64 × 64.(64×64。 ) | 32 × 32.(32×32。 ) |
|  | 7.(7。 )9.(9。 ) | 224.(224。 ) | 98048.(98048。 ) | 64 × 64.(64×64。 ) | 32 × 32.(32×32。 ) |
|  | 4.(4。 )2.(2。 ) | 448.(448。 ) | 98048.(98048。 ) | 64 × 64.(64×64。 ) | 32 × 32.(32×32。 ) |
|  | 16.(16。 )4.(4。 ) | 112.(112。 ) | 95744.(95744。 ) | 32 × 32.(32×32。 ) | 32 × 32.(32×32。 ) |
|  | 9.(9。 )2.(2。 ) | 224.(224。 ) | 95744.(95744。 ) | 32 × 32.(32×32。 ) | 32 × 32.(32×32。 ) |
|  | 4.(4。 )8.(8。 ) | 448.(448。 ) | 95744.(95744。 ) | 32 × 32.(32×32。 ) | 32 × 32.(32×32。 ) |
|  | 18.(18。 )2.(2。 ) | 112.(112。 ) | 134144.(134144。 ) | 32 × 32.(32×32。 ) | 16 × 16.(16×16。 ) |
|  | 10.(10。 )3.(3。 ) | 224.(224。 ) | 134144.(134144。 ) | 32 × 32.(32×32。 ) | 16 × 16.(16×16。 ) |
|  | 5.(5。 )5.(5。 ) |

1. e-mail: dachsbacher@cs.(电子邮件:dachsbacher@cs。 )fau.(福。 )de †e-mail: stamminger@cs.(电子邮件:dachsbacher@cs.fau.de？电子邮件:stamminger@cs。 )fau.(福。 )de.(德。 ) [↑](#footnote-ref-1)