

A Method for Generating Rectangle Area Light Source Using Water Surface Lighting

Liu Feipeng

College of Computer Science, Beihang University, Beijing, China

Email address:

liufeipengkk1@163.com

To cite this article:

Liu Feipeng. A Method for Generating Rectangle Area Light Source Using Water Surface Lighting. *Science Discovery*.

Vol. 6, No. 2, 2018, pp. 116-123. doi: 10.11648/j.sd.20180602.17

Received: April 29, 2018; **Accepted:** May 23, 2018; **Published:** June 22, 2018

Abstract: Light reconstruction refers to the calculation of light source information based on the actual lighting. In order to ensure the consistency of light illumination, the method based on light source modeling has been widely applied in real scenes, but the research on generating virtual light sources using real lighting results under virtual scenes is relatively scarce. In this paper, a method for generating a rectangle surface light for a virtual scene using water surface light images is proposed. Firstly, light shadow in water surface is extracted by the seed superpixel propagation method after superpixel segmentation. Secondly, the boundary of each light shadow region is extracted using the superpixel label. Finally, important sampling, based on the physical rendering of the BRDF model, is used on the points in the light shadow boundary, and a rectangular area surface light source corresponding to each light shadow is calculated by a ray-traced rectangular area surface light source generation algorithm. It is proven that rectangular surface light source can be generated precisely in a virtual scene by comparing with the actual rectangular area light source.

Keywords: Area Light, Super Pixel, Ray Tracing, Light Reconstruction

利用夜间水面光影生成矩形面光源的方法研究

刘飞鹏

计算机学院, 北京航空航天大学, 北京, 中国

邮箱

liufeipengkk1@163.com

摘要: 光源重建是指根据真实光照计算场景光源信息。为了保证光照一致性, 基于光源建模的方法在真实场景应用研究十分广泛, 但在虚拟场景下利用真实的光照信息生成虚拟场景的光源的研究相对稀缺。本文提出一种利用水面光影图像生成虚拟场景的面光源的方法。首先将图片分割成超像素, 通过seed超像素蔓延融合的方法提取光影, 然后利用超像素标签提取出每条光影区域的边界, 最后基于物理渲染的BRDF模型, 对光影边界上的点进行重要性采样, 通过基于光线跟踪的矩形区域面光源生成算法计算每条光影对应的矩形区域面光源。利用本文描述的方法, 通过跟真实的光源数据对比, 能够较好生成矩形面光源。

关键词: 区域光源, 超像素, 光线跟踪, 光源重建

1. 引言

光源建模是指根据真实光照结果提取场景光源信息。在传统的光源建模方式中,光源建模可以被分为根据标志物、借助设备和不需要标识物和辅助设备的方法,比如M Haller[1]提出基于标志物阴影的方法对点光源的位置进行求解,Karsch[2]使用一种场景结构分析的方式以及手动标定场景光源的方式来进行场景区域光源分析,Pessoa[15]利用光照探头将周围环境利用球面谐波函数制作成辐射度立方体贴图,并用该贴图表示环境光照,Kanbara[16]提出在标志点中心利用黑色反光球探测光照,Zhou[17]设计了一套通用框架模型估计不同类型光照,其核心思想也是用了反射球来探测光照, Lee[18]提出了借助静态反射球来研究动态光照的方法。对于提取真实场景下的光源信息的研究已经十分广泛,在利用真实的光照结果生成虚拟场景光源信息的方法研究却十分欠缺,目前主流的光源建模主要以点光源、平行光以及辐射度贴图等形式表示,在不借助辅助设备的情况下,真实场景下对于面光源的建模几乎是不可能的。相对于真实场景的光源重建,虚拟场景下的光源重建有着更多的优势,由于已知光照结果对应表面的参数以及已知场景的相机参数,所以在不借助人工辅助的情况下,虚拟场景的面光源定位成为可能。本文提出一种利用夜晚水面光影图像在虚拟场景中生成具有光照一致性的矩形区域光源的方法。方法主要分为两个步骤:(1)水面光影条带边界提取;(2)基于BRDF[13]光线跟踪的矩形区域光源确定。

水面光影边界提取方案基于超像素分割以及合并的算法,在超像素分割与合并方面,目前主流的超像素分割算法有Marker-watershed[4]、SLIC[7]、Grap-Based[8]等方法,在超像素合并方面Haris K[6]提出基于超像素相似性的方式快速合并方法以及Li L[5]在此基础上加入边界约束条件[5]的合并方法。在超像素分割方面,本文采用的是Achanta[7]提出的slc[7]超像素分割方法,该方法使用简单的聚类算法高效生成超像素,相比较其它方法,不仅具备良好的分割效果,还具有更快的速度和更高的内存效率。分割之后进行基于超像素的光影提取,对于超像素合并而言,经典的算法利用超像素之间的相似性以及边界约束条件进行合并,Haris K[6]和Li L[5]各自提出的超像素合并算法在前后景比较明显的图片已经能够取得比较好的效果。但该类算法并不适用于水面条带的提取,其原因在于水面光影不具备前后景的特点并且也不具备比较明显的边界信息,导致非光影区域过度融合,从而无法较好的提取光源在水面所形成的条带。本文利用基于超像素直方图相似性以及Seed超像素蔓延算法相结合的方式提取光影条带,能够较好地提取计算所需的光影条带区域,并且通过超像素标签提取出光影边界。

使用基于物理的BRDF[13]光线跟踪算法[9]计算矩形区域光源参数,参数的计算主要包含光源的辐射度、位置、以及尺寸。对于真实感场景渲染中,基于物理的BRDF[13]光线跟踪算法[9]能够真实表示水面的反射情况。本文使用基于Blin-phong模型来描述水面反射。通过重要性采样[12],该模型光线采样向视线的反射线聚拢并呈现圆锥形(如图

3)。设定阈值 t ,保证采样所有光线都在该锥形内部,该圆锥体与场景光源所在的平面相交并产生横截面,当横截面在光源内部时,光影上的点较亮,提取光影超像素中最亮的超像素,使用公式(7)计算光源的辐射强度;当横截面一部分在光源内部的时候,其对应的点为光影边界上的点,光源包裹横截面的程度导致光影边界由亮到暗的渐变,这是由于光源包裹程度越大,圆锥体内与光源相交的光线越多,利用这种特性,通过调整光源边界值,更新圆锥体内采样光线的相交状态 $G(w_i)$,光线 w_i 与光源相交, $G(w_i)$ 置为1,否则置为0,利用公式(11)计算当前的反射系数 f ,当 f 与真实反射系数一致时,光源的边界计算完成。

2. 准备工作

在本文中,光影边缘提取主要利用超像素分割以及基于seed超像素蔓延算法。图像超像素分割使用SLIC[7]算法,光影提取使用seed超像素蔓延算法,通过比较seed超像素和非seed超像素的相似性,利用seed超像素蔓延,提取图像光影区域。基于光线跟踪的矩形区域光源生成算法主要通过光影边缘上的点做重要性采样,每个点对应的采样光线集都向该点视线的反射线聚拢,该聚拢的形状呈现锥形,通过研究采样光线与光源的相交情况,分析该点实际辐射度与理论辐射度的关系,计算矩形区域光源的边界以及位置信息。

超像素分割的目的在于将图像局部聚类,目前主流的超像素分割算法有SLIC[7]、Marker-watershed[6]、Grap-Based[8]都能取得较优秀的效果,本文采用SLIC超像素分割,SLIC算法基于聚类算法,具有内存利用率高计算速度快等特性。对于超像素合并,旨在融合相似性较高的超像素,Haris K[6]和Li L[5]等人的工作对于超像素合并有着较为优秀的效果,但不适用于光影,这是由于光影边界呈现渐变的特性,不具备明显边界导致超像素过度融合从而影响光影提取。本文针对场景特殊性,利用seed超像素进行蔓延提取光影区域。

光线跟踪算法是通过跟踪光线在场景中行为来模拟真实的光照情况,双向反射分布函数能够较好的描述材质表面的反射能力,Blin-phong模型能够较好描述镜面效果,本文使用基于物理Blin-phong模型的BRDF[13]来对水面实现真实的渲染,在已知水面渲染结果未知矩形光源的情况下,通过研究真实水面上的点的采样光线与虚拟场景相交特性来计算矩形区域光源的辐射度以及位置。同时本文需要光线跟踪算法做真实水面光影渲染,用于实验对比。

3. 光影边缘提取

光影的边缘提取首先利用图像超像素分割算法进行超像素分割,由于水面光影条带不具备前后景和明确边界的特性,Haris K[7]提出的快速区域合并算法并不能提取出比较理想的光影区域,由于没有明显的边界, Li L[5]提出的基于边界的超像素分割合并算法并不适用。本文提出基于seed超像素蔓延算法进行水面光影的提取,以及利用超像素标签提取光影边缘。

3.1. 图像超像素分割算法

超像素的概念由Malik J[3]和Xiaofeng Ren[3]提出,是指具有相似特征的相邻像素构成的局部不规则区域块,SLIC[7]超像素分割能够对图像进行局部像素聚类,具备相似纹理、亮度、颜色等特征能够形成像素区域块。超像

素分割算法能够较好的对水面光影条带进行聚类,从而使像素级别的水面光影提取转化为超像素级别的提取。SLIC[7]超像素分割需要设定参数K, K为预分割图像超像素个数,假设待分割的图像像素总数为N,则预计每个超像素大小为N/K。

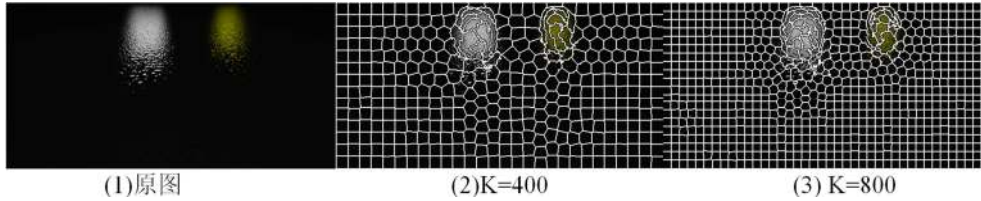


图1 SLIC超像素分割图。

3.2. 光影边缘提取

由于水面光影表现出没有前后背景的特性,不具备明确的边缘属性,直接利用超像素合并的方式提取出来的光影效果很差。本文提出利用seed超像素和蔓延的方式,并结合超像素标签来提取水面光影的边缘。光影边缘的提取分为三个步骤:(1)提取seed超像素;(2)提取光影;(3)提取光影上下左右边缘像素。

3.2.1. Seed超像素提取

seed超像素是指光影中较亮区域的超像素,seed超像素的提取方法步骤为首先将RGB图像模糊去燥,然后转化到HSV空间,设定间隔k,每隔k行,对这些行的所在的像素V分量做均值滤波去除噪点,然后提取该行的波峰,设定阈值t,若波峰处的明度V大于阈值t,则称该点为种子点,若种子点的明度V小于所在超像素的平均明度V_s且小于阈值t,则去除该种子,种子所在的超像素称之为seed超像素。

3.2.2. 光影条带提取

光影的提取主要利用seed超像素的蔓延算法。光影在水面的表现具备中间亮边缘暗的特点,这是由于处于光影中心的点采样的光线集大多数与光源相交,而处于光影边缘的点采样的光线集一部分与光源相交,利用光影超像素

色调的相似性,利用公式(1)计算两个超像素色调的直方图相似性来判断两个超像素的相似程度:

$$d(H_1, H_2) = \frac{\sum_i (H_1(i) - \bar{H}_1)(H_2(i) - \bar{H}_2)}{\sqrt{(\sum_i (H_1(i) - \bar{H}_1)^2)(\sum_i (H_2(i) - \bar{H}_2)^2)}} \quad (1)$$

其中H₁,H₂分别代表超像素S₁和S₂的直方图, H₁(i)和H₂(i)代表超像素S₁和超像素S₂直方图第i个筒的值, \bar{H}_1 和 \bar{H}_2 分别代表超像素S₁和超像素S₂直方图的平均值。

seed超像素蔓延算法主要利用seed超像素与周围超像素的相似性进行蔓延来提取光影,具体操作步骤为:

(1)Seed超像素集S={S₁, S₂, S₃... S_n},每个超像素对应的初始状态T[i]=0

(2)访问超像素集S,若S_i的状态T[i]=0,则S_i形成光带C[i],此时初始状态C[i]={S_i},将S_i状态T[i]置为1,获取S_i的邻接超像素Neighbor(S_i)={S_{i1}, S_{i2}, S_{i3}... S_{in}},按序判断超像素S_i与S_{ij}的相似程度, S_{ij}属于Neighbor(S_i),若两个超像素相似,则令T[i_j]=1,将超像素S_{ij}加入光带C[i]中,获取S_{ij}的邻接超像素,将该邻接超像素加入到Neighbor(S_i)的集合中去,当遍历完Neighbor(S_i)时,条带C[i]提取完成。

(3)按序访问超像素集S,重复步骤(2),直到S访问完成,形成超像素集合C={C₁,C₂,.....C_m},m为提取出来的光影条带个数。

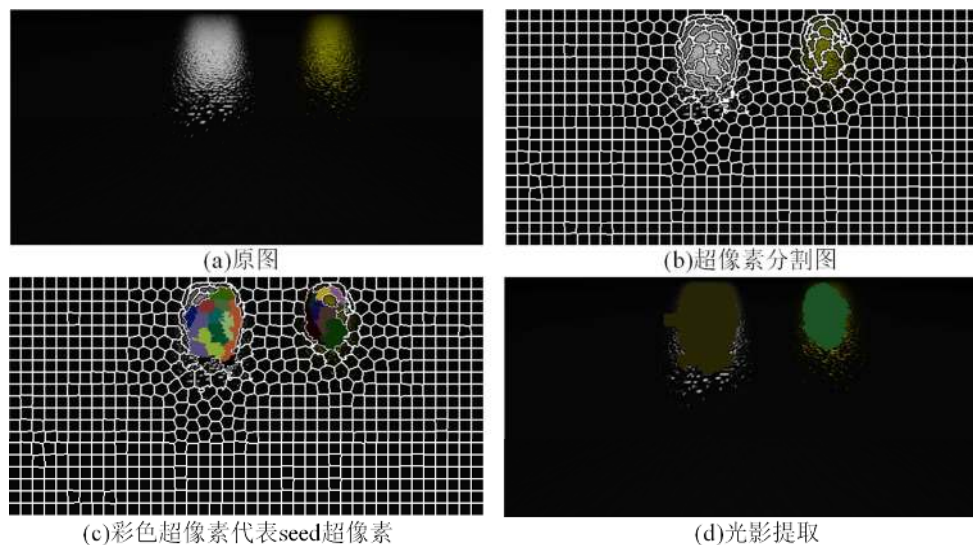


图2 通过seed超像素蔓延算法光影提取过程。

3.2.3. 光影边缘提取

光影边缘提取主要利用超像素的标签进行提取。边缘提取分别提取光影的上下左右四个方向的边缘，分别对应矩形光源的下边界、上边界、左边界与右边界。超像素分割结果分配超像素唯一标识符，以左边界提取为例， $p[i][j]$ 表示*i*行*j*列属于某一光影条带 C_m 上的一点，则点 $p[i][j]$ 的标识符为*m*，扫描以*p*为中心的3*3个像素点，若以*p*为中心的左边三个点的标识符都不为*m*，则*p*可以被认为是左边界上的一个点。

4. 基于光线跟踪的矩形光源的确定

光线跟踪算法(ray tracing)[9]是一种高质量的三维场景真实感渲染方法。其核心原理为从视点处投射光线，通过跟踪光线在场景中的行为，来模拟真实全局光照情况。水面光影的渲染，主要追踪反射光线与光源的相交信息，获取发光源的辐射度来完成的渲染，在渲染过程中利用基于物理双向反射分布函数(BRDF)[13]，该函数用于描述物体表面的辐射强度与某一方向反射出去的辐射度的比值，用于描述材质的反射能力。本文对基于物理的双向发射分布函数(BRDF)[13]光线跟踪算法。利用光影区域，使用重要性采样发出采样光线集，探测光影对应的光源在场景中的位置。

4.1. 基于重要性采样的水面辐射度

利用基于物理双向反射分布函数(BRDF)[13]光线跟踪方法，通过重要性采样与蒙特卡洛积分法，可以计算出光源辐射度*L*与光影上一点的辐射度 L_0 之间的关系。

蒙特卡洛积分法[11]利用概率统计的方法来估计数值积分，通过对某概率分布随机抽样获取大量样本，将数值积分求解转化为对概率分布上的期望，估算函数为公式(2)：

$$FN = \frac{1}{N} \sum_{i=0}^N \frac{f(X_i)}{p(X_i)} \quad (2)$$

其中， $f(X_i)$ 为被积函数，*X*为根据概率密度函数为 $p(X_i)$ 所抽取的样本，对于Blin-phong模型而言，规范化的Blin微面元模型[10]光线分布的概率密度公式为(3)：

$$D(\mathbf{w}_h) = \frac{e+2}{2\pi} (\mathbf{w}_h \cdot \mathbf{n})^e \quad (3)$$

其中： \mathbf{w}_h 为半角向量，*n*为法线，*e*为光泽常数

重要性采样(Important Sampling)[14]是光线追踪中比较常用的一种采样方式。该采样方式是一种方差缩减技术，当采样源自于被积函数具有相似的概率分布函数时，蒙特卡洛估算方式能够快速收敛，即概率密度函数 $p(\mathbf{w}_h) = c * D(\mathbf{w}_h)$ 。其基本理念可以描述为光线采集不再是半球均匀采样，而是服从概率密度函数 $p(\mathbf{w}_h)$ 的采样，对于Blin-phong模型而言，将该概率密度函数 $p(\mathbf{w}_h)$ 转化到极坐标系下可以表示为公式(4)中的第一个公式。通过对概率分布函数求逆将服从均匀分布的样本转化为所需样本 (θ, φ) ，采样公式为公式(5)

$$\begin{cases} p(\theta, \varphi) = \frac{e+2}{2\pi} \cos^e \theta \sin \theta \\ p(\theta) = \int_0^{2\pi} p(\theta, \varphi) d\varphi \\ p(\varphi|\theta) = \frac{p(\theta, \varphi)}{p(\theta)} \end{cases} \quad (4)$$

$$\begin{cases} \theta = \arccos(e+1\sqrt{\varepsilon_1}) \\ \varphi = 2\pi\varepsilon_2 \end{cases} \quad (5)$$

其中： ε_1 和 ε_2 均为[0,1]之间的均匀随机样本。利用蒙特卡洛积分法以及采样数据，对于水面上的一点*p*，入射光线为 \mathbf{w}_0 ，法线为*n*，利用公式(2)，以及Walter B[10]提出的微表面模型，点*p*辐射度 L_0 有：

$$L_0(p, \mathbf{w}, \mathbf{n}) = \frac{1}{N} \sum_{i=1}^N \frac{L * D(\mathbf{w}_h) fr(\mathbf{w}_0) G(\mathbf{w}_i)}{4p(\mathbf{w}_h) \cos \theta_0 \cos \theta_i} \quad (6)$$

其中： θ_0 为视线 \mathbf{w}_0 与法线的夹角， θ_i 为采样光线 \mathbf{w}_i 与法线的夹角， \mathbf{w}_i 的方向通过公式(8)计算， $fr(\mathbf{w}_0)$ 为菲涅尔项， $G(\mathbf{w}_i)$ 为 \mathbf{w}_i 光线与光源的相交情况，若不相交， $G(\mathbf{w}_i) = 0$ ，否则 $G(\mathbf{w}_i) = 1$ 。考虑到gamma矫正，光源出射辐射度*L*与光影上一点的辐射度 L_0 之间的关系为：

$$L = \frac{\text{gamma}^{-1}(L_0(p, \mathbf{w}, \mathbf{n}))}{\frac{1}{N} \sum_{i=1}^N L * D(\mathbf{w}_h) fr(\mathbf{w}_0) G(\mathbf{w}_i) / (4p(\mathbf{w}_h) \cos \theta_0 \cos \theta_i)} \quad (7)$$

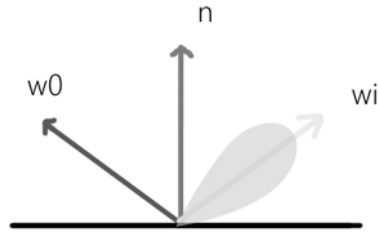


图3 重要性采样下Blin-phong光线采样表现出圆锥形，采样光线 \mathbf{w}_i 在锥形内部。

4.2. 光源辐射度

人眼观察到的水面上的颜色实际上是光源的出射辐射度对水面的作用效果，水面对光源的出射辐射度进行反射和吸收，反射出来的辐射度形成水面的光影，对于同一条带而言，其对应的光源的出射辐射度值一样。利用光源辐射度*L*与光影上一点的辐射度 L_0 之间的关系，求解光源辐射度。对于求解光影*C_i*对应的矩形光源辐射度值，具体的操作步骤为：

- (1) 选取光影*C_i*中的亮度值最高超像素*S_i*，*S_i*中获取亮度值最高的像素点*E*，像素值为 L_p 。
- (2) 利用公式(5)对点*E*所在的空间位置*p*进行采样，采样集为 $\mathbf{w} = \{ \mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_m \}$ 。
- (3) 将光束集 \mathbf{w} 对虚拟场景做相交性判断，与光线集相交最多的面片并且通过深度测试的面片为光源所在的面片，记录该面片面片深度*t_i*为光源平面深度*P=t_i*，该平面为矩形区域光源所在平面。判断采样光线集 \mathbf{w} 与该对象的相交性，若相交 $G(\mathbf{w}_i) = 1$ ，否则 $G(\mathbf{w}_i) = 0$ ；
- (4) 利用公式(7)计算光源的辐射度*L*。

4.3. 矩形光源位置

水面光影上的一点,对采样的光线在三维空间上的分布呈现出圆锥形状(采样光线以视线的反射线为中心聚拢,这是由于Blin-phong模型的重要性采样所决定的,以反射线为中心,越往外所采集的光线越少),从该点出沿着反射线延长圆锥,该圆锥与矩形区域光源相交,圆锥的横截面在矩形区域光源所在的平面与矩形区域光源所在的位置关系有三种:(1)横截面在光源外(2)横截面在光源内(3)横截面与光源边界相交。对于第一种情况,该点所采样的光线在光源之外,该光源对该点辐射度的贡献为0,若没有其它光源干预,该点的亮度比较低。对于第二种情况。横截面在光源内部,该点的辐射度受立体角与法线夹角影响,对于光影内部较亮的超像素,属于这种情况,光源辐射强度的计算正是利用了这种特性。对于第三种情况,由于横截面与光源边界相交,随着横截面慢慢移出矩形光源,具体表现的形式为光影边界亮度呈现渐变。由于这种特性的存在,使得矩形光源边界的计算成为可能,从而能确定虚拟场景中矩形光源位置。

4.3.1. 采样圆锥横截面曲线表示

对于Blin-phong重要性采样,大部分光线在圆锥内部,有较小概率会将光线采样到圆锥外部,为了防止这种情况,设定阈值 t ,使 $t = \cos^e + 1 \theta_0$,因此阈值 t 决定圆锥的大小,相应公式(5)中,对应的均匀随机样本 ϵ_1 的取值范围限定在 $[t, 1]$ 之间。对于光影边缘的一点 p ,其视线反射的向量为 r ,以 r 为 z 轴建立笛卡尔坐标系 $u-v-r$,根据采样公式(5),则 p 点采样光线 w_i 的方向为:

$$d_i = (u \sin \theta \cos \varphi + v \sin \theta \sin \varphi) + r \cos \theta \quad (8)$$

当 $\cos \theta = t$ 时, w_i 为圆锥边缘上的采样的光线。 w_i 与光源所在平面 $P=ti$ (ti 为矩形光源平面在场景中的深度,光源辐射强度确定时从虚拟场景中获取)的交点为 $p'(x', y', ti)$,对于该横截面的 x' 与 y' 有:

$$\begin{cases} x' = p.x + l * (\cos \varphi * u.x + \sin \varphi * v.x + tz * r.x) \\ y' = p.y + l * (\cos \varphi * u.y + \sin \varphi * v.y + tz * r.y) \end{cases} \quad (9)$$

其中

$$\begin{cases} l = \frac{ti - p.z}{\cos \varphi * u.z + \sin \varphi * v.z + tz * r.z} \\ tz = \cos \theta_0 / \sin \theta_0 \end{cases} \quad (10)$$

公式(9)表示为点 p 的采样光线所形成的锥形在矩形光源处的横截面曲线,该曲线从形状上呈现为椭圆,通过对公式(9)一阶求导,获取 x' 的最大值 x'_{max} 和最小值 x'_{min} 与 y' 的最大值 y'_{max} 和最小值 y'_{min} ,则光源左边界left-edge和右边界right-edge在 $[x'_{min}, x'_{max}]$ 之间,上边界up-edge和下边界down-edge在 $[y'_{min}, y'_{max}]$ 之间。

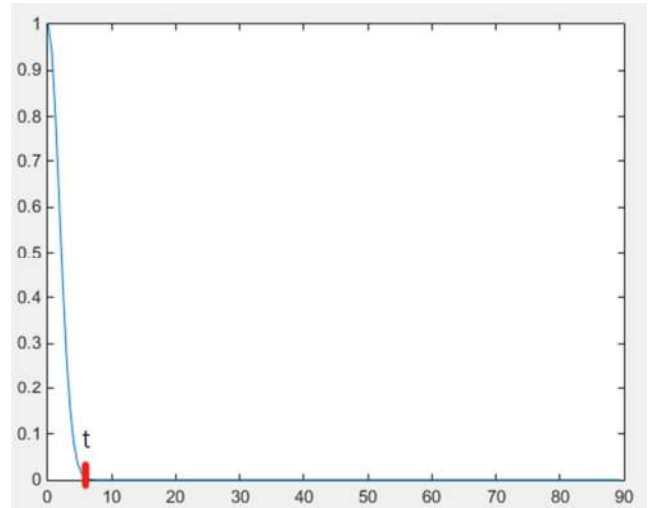


图4 Blin-phong模型,对于公式(3), $e=800$ 对应光线采样的概率分布图,横轴表示采样光线与视线反射线之间的角度 θ ,纵轴表示光线采样概率 $P(X \leq \theta)$,由该图可以看出,阈值 t 对应的角度 θ_0 ,当 $\theta > \theta_0$ 时,光线被采样的概率值接近于0,通过设定阈值 t ,计算出光线所在的圆锥。

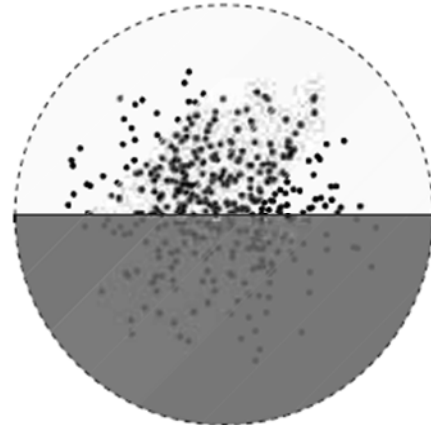


图5 光影下边界点对应的圆锥截面与光源上边界相交,采样光线只有部分在光源内部。

4.3.2. 矩形光源位置确定

光影边缘点 p ,该点的辐射度为 L_0 ,对应矩形光源出射辐射度为 L ,真实反射系数为 $\text{coef} = \gamma^{-1}(L_0)/L$,设定采样光线数量为 N ,根据公式(8)生成采样光线集合 $w = \{w_1, w_2, \dots, w_n\}$,根据公式(6)有反射系数:

$$f = \frac{1}{N} \sum_{i=1}^N \frac{D(w_h) f_r(w_0) G(w_i)}{4p(w_h) \cos \theta_0 \cos \theta_i} \quad (11)$$

以上边界边界为例,点 p 为光影下边缘上的一点,采样光线与矩形光源平面的交点都在圆锥横截曲线内,按照光线是否在光源内部,将采样光线进行划分,如图6所示,交点不在光源内部中有 $G(w_i)=0$,在光源内部有 $G(w_i)=1$ 。根据公式(9)可以求出圆锥横截面在 y 轴上的最小值 y'_{min} 和最大值 y'_{max} , y 在点 p 的圆锥横截曲线的 y 最小值 y'_{min} 和最大值 y'_{max} 之间,假设当光源上边界up-edg= y 时,有 k 根光线在光源中, m 根光线在光源外,其中 $k+m=N$,通过调整 y 的值, k 与 m 的数值发生变化,从而导致 f 数值发生变化,当 coef 与 f 的差值小于阈值 tc 时,此时根据点 p 求出

光源的上边界为 y 符合边界上真实的光照情况。对于光影下边界点集 $P=\{p_0, p_1, \dots, p_m\}$ ，求解出对应的上边界集合 $Y'=\{y_0, y_1, \dots, y_m\}$ ，则上边界 $up-edge = \sum(Y')/m$ 。

5. 实验结果分析

本文所有实验结果都在如下平台下进行：CPU为Inter(R) Core(TM) i5-6300HQ 2.30GHZ，内存为12GB,显卡为NVIDIA Geforce GTX 960M，显存为4GB,水面的渲染利用NVIDIA Optix Ray Tracing Engine[19]完成。实验结果分为两部分，一部分为光影提取结论分析，另一部分为矩形区域光源数据与真实数据(Ground-truth)之间的对比，以此来验证本文虚拟场景矩形区域光源方案的可行性。

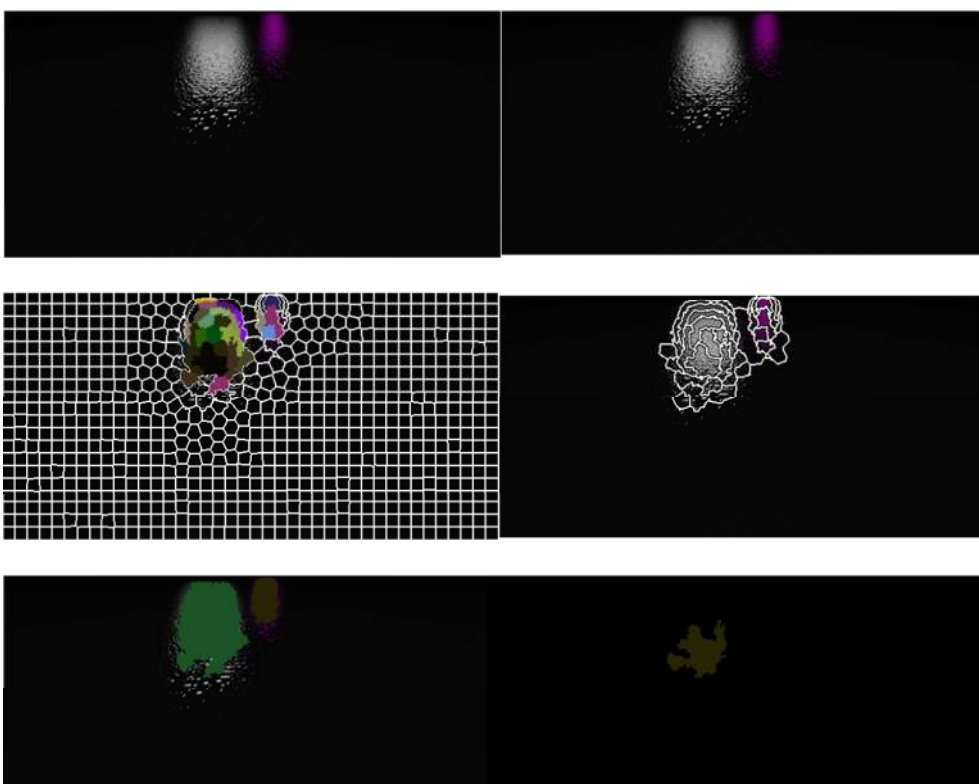


图6 左侧一列为本文方法提取的光影，右侧一列为通过Haris K[6]的Fast Region Merge的方式提取的光影，其中右侧第二幅图超像素完成合并，右侧第三幅图为基于合并图超像素做相似性判断提取光影结果。

5.2. 矩形光源实验结果分析

本文设计了三组对比实验，其中真实光源数据以及水面光影图利用光线跟踪和基于物理渲染方式完成。渲染中使用的相机位置为(0,0.2,10)，相机fov大小为60度，宽高比为1:1，渲染图像大小为600*600，水面区域在图像中大小为600*300，水面为真实的高度数据，矩形区域光源提取算法中所用到的相机以及水面高度图均与渲染时用到的数据保持一致。在所有表中，Light表示光源名称，Light-depth表示矩形区域光源深度，通过调整该参数，用于观察同一光影在不同场景深度情况下产生不同参数的矩形区域光源， e 表示Blin-phong光泽度常数，Left-Bottom表示矩形区域光源左下角在空间中的位置，width和height

5.1. 光影提取实验结果分析

本文将基于seed超像素蔓延的光影提取方法与Haris K[6]快速区域合并提取方法进行对比，可以发现传统的快速区域合并提取方案出现单光影条带提取质量差，多光影条带漏提取的现象，这是由于光影图像缺乏边界，边缘过于模糊从而引起超像素过度地融合。从图(7)中可以看出基于seed超像素蔓延的光影提取方法并不能较为完美的将整条光影条带提出，对于光影边界过于暗的区域，由于与seed超像素色调上的直方图差异过大，导致该超像素无法融合，对于此类情况并不影响后续矩形光源边界的计算，一是由于过暗的超像素内部只有少数像素点能用于边界计算，二是由于光源边界对应的是一段渐变的光影边缘，只要提取的光影包含实际光影渐变边缘的一部分，提取到的光影边界对于后续计算仍然是有效的。

分别为矩形区域光源的宽和高，LightIntensity为光源的辐射强度。在所有表中，Ground-truth-light0和Ground-truth-light1是真实的光源，light0和light1是利用Ground-truth-light0和Ground-truth-light1的水面光影生成的虚拟矩形区域光源。

表1中，通过对比Ground-truth-light0和Light0，在虚拟光源与真实光源深度相同的情况下，常数 e 会影响光源的大小， e 越小生成矩形区域光源尺寸越小。这与Blin-phong模型光泽度的特性保持一致，当 e 和真实的数据保持一致时，生成的矩形区域光源较为准确，可以验证本文所描述的矩形区域光源生成算法具有可靠性和准确性的。表2中，当常数 e 保持一致时，虚拟场景通过变动light-depth的值来

模拟生成的虚拟光源light0所在虚拟场景中的深度，实验结果表明生成的矩形区域光源距离相机越远，生成的光源尺寸越大，距离相机越近，生成的光源尺寸越小表3中，水面包含两组矩形光源 Ground-truth-light0 和 Ground-truth-light0的光影，对应生成虚拟场景中的light-0和light1，该组实验表示多光影能对应生成多个矩形区域光源。

通过表1、表2、表3中的数据对比发现，利用本文论述的方法，水面光影图片能够较好的在虚拟场景中生成矩形区域光源。在位置计算上，当光源深度与光泽系数一致时，能够得到较高的精确度。当光线不一致的时候，本文阐述的方法能够产生较好质量的光源，保证水面光照的一致性。

表1 同一光影受常数e的影响对应生成的光源属性。

Light	Ligth-depth	e	Left-Bottom (x, y, z)	width	height	Light-intensity (r, g, b)
Ground-truth-light0	0	500	(-4,2,0)	3	3	(20,20,20)
Light0	0	500	(-3.9687, 1.99092, 0)	2.95216	3.04463	(21.4182, 21.4182, 21.4182)
Light0	0	200	(-3.5585, 2.4205, 0)	2.15409	2.41417	(22.9963, 22.9963, 22.9963)

表2 光影受矩形区域光源深度影响生成的光源属性。

Light	Ligth-depth	e	Left-Bottom (x, y, z)	width	height	Light-intensity (r, g, b)
Ground-truth-light0	0	500	(-4,2,0)	3	3	(20,20,20)
Light0	0	500	(-3.9687, 1.99092, 0)	2.95216	3.04463	(21.4182, 21.4182, 21.4182)
Light0	-3	500	(5.0552, 2.65204, -3)	3.67095	3.81414	(22.3919, 22.3919, 22.3919)
Light0	3	500	(-2.8414, 1.39481, 3)	2.20807	2.20612	(20.8895, 20.8895, 20.8895)

表3 多光影生对应成多个矩形局域光源。

Light	Ligth-depth	e	Left-Bottom (x, y, z)	width	height	Light-intensity (r, g, b)
Ground-truth-light0	0	500	(-4,2,0)	3	3	(20,20,0)
Ground-truth-light1	0	500	(5,2,0)	2	2	(10,10,0)
Light0	0	500	(-3.9687, 1.99092, 0)	2.95216	3.04463	(21.4182, 21.4182, 21.4182)
Light1	0	500	(5.0145, 1.99256, 0)	1.99571	2.01065	(7.41385, 7.41385, 0.037866)
Light0	-3	500	(-5.0552, 2.65204, -3)	3.67095	3.81414	(22.3919, 22.3919, 22.3919)
Light1	-3	500	(6.63608, 2.53988, -3)	2.43089	2.72177	(6.96159, 6.96159, 0.0355561)

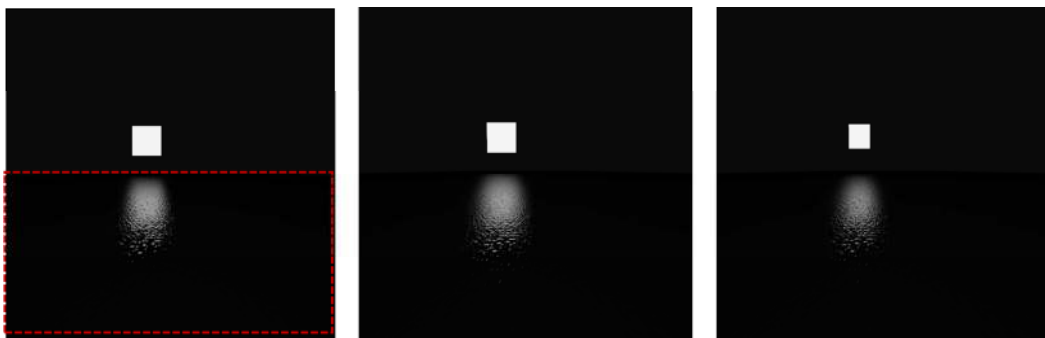


图7 表1中矩形区域光源的渲染图。

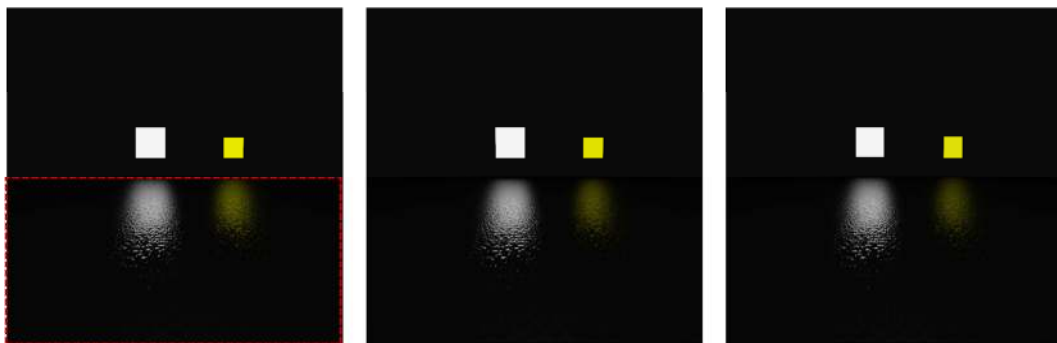


图8 表3中矩形区域光源的渲染图。

其中：图7左图为ground-truth-light0光源的渲染结果(对应表1第1行数据)；中间图利用ground-truth-light0生成光影图(左图红色选框部分)生成的虚拟光源渲染图(对应表1第2行数据)，其中e=500,light depth = 0；右图利用

ground-truth-light0生成光影图(左图红色选框部分)生成的虚拟光源渲染图(对应表1第3行数据)，其中e=200,light depth = 0。

图8左图为ground-truth-light0和ground-truth-light1光源的渲染结果(对应表1第1行数据);中间图图为利用ground-truth-light0和ground-truth-light1光源生成的光影图(左图红色选框部分)生成的虚拟光源渲染图(对应表3第3行和第4行数据数据),其中 $e=500$,light depth = 0;右图为利用ground-truth-light0和ground-truth-light1光源生成的光影图(左一图红色选框部分)生成的虚拟光源渲染图(对应表3第3行和第4行数据数据),其中 $e=500$,light depth=-3。

5.3. 局限性

本文方法的前提在于夜间的水面光影不能重叠,这是由于重叠的水面光影会使边界消失以及重叠部分的辐射度大于主要矩形光源贡献的辐射度,从而导致无法正确计算光影对应的矩形区域光源,同时多个光影的重叠代表采样的复杂性,说明光影上的一点的采样光线与多个光源进行相交,本文所阐述的方法并不能处理这种情况。

6. 结论与未来工作

本文提出一种利用夜景水面光影图像生成虚拟场景光源的方法,用于生成虚拟场景中的矩形光源。此方法利用夜晚中的水面图片,利用其光影信息推测虚拟场景中的矩形光源。首先通过超像素分割以及seed超像素蔓延方法提取出水面光影数据,然后利用超像素标签提取出水面光影边界,最后通过研究BRDF[13]的Blin-phong模型的采样光线分布特性,研究光影边界点在渲染过程中采样光线与矩形区域光源的相交特性,来计算光影对应的虚拟场景中光源的位置以及虚拟场景中光源的辐射度。本文通过利用真实的水面数据生成的光源与真实的光源数据对比,验证本文所阐述的虚拟场景中矩形区域光源生成算法具备较高可靠性。

为了生成更多种类的光源,比如聚光灯,球形光源,需要研究光影形状特性以及采样光线与对应种类光源的相交特性。同时为了使适用性更广,比如解决光影重叠情况下的光源生成,可以参考增强现实的方法,从真实的场景中提炼出一些有效的光源信息作为约束条件。

参考文献

- [1] Haller, Michael, S. Drab, and W. Hartmann. "A real-time shadow approach for an augmented reality application using shadow volumes [C]." *ACM Symposium on Virtual Reality Software and Technology* ACM, 2003:56-65.
- [2] Karsch, Kevin, et al. "Rendering synthetic objects into legacy photographs [C]." ACM, 2011:1-12.
- [3] Malik J. Learning a classification model for segmentation[J]. Proc Iccv, 2003, 1:10-17 vol.1.
- [4] Hu Z, Zou Q, Li Q. Watershed superpixel[J]. 2015.
- [5] Li L, Yao J, Tu J, et al. Edge-based split-and-merge superpixel segmentation [C]. IEEE International Conference on Information and Automation. IEEE, 2015:970-975.
- [6] Haris K, Efstratiadis S N, Maglaveras N, et al. Hybrid image segmentation using watersheds and fast region merging[J]. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 1998, 7(12):1684.
- [7] Achanta R, Shaji A, Smith K, et al. SLIC superpixels[J]. Eplf, 2010.
- [8] Felzenszwalb P F, Huttenlocher D P. Efficient Graph-Based Image Segmentation [J]. International Journal of Computer Vision, 2004, 59(2):167-181.
- [9] Jensen H W. Monte Carlo Ray Tracing [J]. 2003.
- [10] Walter B, Marschner S R, Li H, et al. Microfacet Models for Refraction through Rough Surfaces. [C]. Eurographics Symposium on Rendering Techniques, Grenoble, France. DBLP, 2007:195-206.
- [11] Rubinstein R Y. Simulation and the Monte Carlo Method[M]. Simulation and the Monte Carlo method. John Wiley & Sons, 2008:167-168.
- [12] Art Owen, Yi Zhou Associate. Safe and Effective Importance Sampling[J]. Journal of the American Statistical Association, 2000, 95(449):135-143.
- [13] Dornbach P. Implementation of Bidirectional Ray Tracing Algorithm [J]. Cescg, 1998.
- [14] Neal R M. Annealed Importance Sampling[M]. Kluwer Academic Publishers, 2001.
- [15] Pessoa S A, Moura G d S, Lima J P S d M, et al. RPR-SORS: real-time photorealistic rendering of synthetic objects into real scenes [J]. Computers & Graphics, 2012, 36(2): 50-69.
- [16] Kanbara M, Yokoya N. Real-time estimation of light source environment for photorealistic augmented reality [C] //Proceedings of the 17th International Conference on Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2004, 2:911-914.
- [17] Zhou W, Kambhmettu C. A unified framework for scene illuminant estimation[J]. Image and Vision Computing, 2008, 26(3): 415-429.
- [18] Lee S, Jung S K. Estimation of illuminants for plausible lighting in augmented reality [C] //Proceedings of the International Symposium on Ubiquitous Virtual Reality. Los Alamitos: IEEE Computer Society Press, 2011: 17-20.
- [19] NVIDIA. 2016. Nvidia optix ray tracing engine. <http://developer.nvidia.com/optix>.