

7-15-2024

## Real-time Non-photorealistic Rendering Method for Black and White Comic Style in Games and Animation

Yan Hu

*College of Information, North China University of Technology, Beijing 100144, China,  
huyan0413@126.com*

Lizhe Chen

*College of Information, North China University of Technology, Beijing 100144, China,  
chenlizheme@outlook.com*

Hanna Xie

*College of Information, North China University of Technology, Beijing 100144, China*

Yuyao Ge

*College of Information, North China University of Technology, Beijing 100144, China*

*See next page for additional authors*

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research, Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

---

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation. For more information, please contact [xtfzxb@126.com](mailto:xtfzxb@126.com).

---

# Real-time Non-photorealistic Rendering Method for Black and White Comic Style in Games and Animation

## Abstract

**Abstract:** To address the issues of high resource consumption and lengthy workflow in general nonphotorealistic, this paper proposes a real-time non-photorealistic rendering method for black and white comic style in games and animation. A specialized lighting model is designed to highlight the main environmental light and the grayscale grading of diffuse reflection based on the analysis of the lighting model effect. The pre-processing of the scene is achieved by merging the various components of the lighting model. A screen space three-phase edge detection method is proposed to sequentially perform depth edge detection, normal edge detection, and color edge detection on the pre-processed scene lighting results, and the edge detection results are combined. The scene lighting results are rendered by screen space partitioning, utilizing dot shading and line shading respectively based on the grayscale value of the lighting results, with interpolation applied to shading boundaries. The results are merged by designing a segmented function based on the grayscale value of the lighting model results, combining the screen space three-phase edge detection results with the screen space partition rendering results to obtain the final rendered output. Experimental results demonstrate that the proposed method can achieve nonphotorealistic rendering of black and white comic style with relatively low resource consumption and reduced error rate, showcasing a distinctive black and white comic style.

## Keywords

black and white comic style, non-photorealistic rendering, real-time rendering, specialized lighting model, edge detection

## Authors

Yan Hu, Lizhe Chen, Hanna Xie, Yuyao Ge, Shun Zhou, and Xingquan Cai

## Recommended Citation

Hu Yan, Chen Lizhe, Xie Hanna, et al. Real-time Non-photorealistic Rendering Method for Black and White Comic Style in Games and Animation[J]. Journal of System Simulation, 2024, 36(7): 1699-1712.

# 面向游戏与动漫的黑白漫画风格非真实感实时渲染方法

胡燕, 陈立哲\*, 谢含纳, 葛钰晓, 周顺, 蔡兴泉

(北方工业大学 信息学院, 北京 100144)

**摘要:** 针对一般黑白漫画风格的非真实感渲染方法资源开销高和实时性差的问题, 提出一种面向游戏与动漫的黑白漫画风格非真实感实时渲染方法。设计特化光照模型, 通过分析光照模型效果, 设计主体突出的环境光项和灰度分级的漫反射项, 合并光照模型各项, 实现基于特化光照模型的场景预处理; 提出屏幕空间三相边缘检测方法, 将预处理得到的场景光照结果依次进行深度边缘检测、法线边缘检测、颜色边缘检测, 合并边缘检测结果; 对场景光照结果进行屏幕空间分区域渲染, 依据光照结果灰度值大小, 分别采用网点着色和线条着色方式处理, 并采用插值方法处理着色交界; 合并结果, 依据光照模型结果灰度值设计分段函数, 合并屏幕空间三相边缘检测结果与屏幕空间分区域渲染结果, 得到最终渲染结果。实验表明, 该方法能够以较低的资源开销和较快的渲染速度实现黑白漫画风格非真实感渲染, 在有限的资源下提高渲染的效果, 具有显著的黑白漫画风格。

**关键词:** 黑白漫画风格; 非真实感渲染; 实时渲染; 特化光照模型; 边缘检测

中图分类号: TP391.9 文献标志码: A 文章编号: 1004-731X(2024)07-1699-14

DOI: 10.16182/j.issn1004731x.joss.23-0458

**引用格式:** 胡燕, 陈立哲, 谢含纳, 等. 面向游戏与动漫的黑白漫画风格非真实感实时渲染方法[J]. 系统仿真学报, 2024, 36(7): 1699-1712.

**Reference format:** Hu Yan, Chen Lizhe, Xie Hanna, et al. Real-time Non-photorealistic Rendering Method for Black and White Comic Style in Games and Animation[J]. Journal of System Simulation, 2024, 36(7): 1699-1712.

## Real-time Non-photorealistic Rendering Method for Black and White Comic Style in Games and Animation

Hu Yan, Chen Lizhe\*, Xie Hanna, Ge Yuyao, Zhou Shun, Cai Xingquan

(College of Information, North China University of Technology, Beijing 100144, China)

**Abstract:** To address the issues of high resource consumption and lengthy workflow in general non-photorealistic, this paper proposes a real-time non-photorealistic rendering method for black and white comic style in games and animation. A specialized lighting model is designed to highlight the main environmental light and the grayscale grading of diffuse reflection based on the analysis of the lighting model effect. The pre-processing of the scene is achieved by merging the various components of the lighting model. A screen space three-phase edge detection method is proposed to sequentially perform depth edge detection, normal edge detection, and color edge detection on the pre-processed scene lighting results, and the edge detection results are combined. The scene lighting results are rendered by screen space partitioning, utilizing dot shading and line shading respectively based on the grayscale value of the lighting results, with interpolation applied to shading boundaries. The results are merged by

收稿日期: 2023-04-18 修回日期: 2023-06-26

基金项目: 教育部人文社会科学基金(22YJAZH002)

第一作者: 胡燕(1983-), 女, 讲师, 硕士, 研究方向为计算机图形和新媒体艺术。E-mail: huyan0413@126.com

通讯作者: 陈立哲(2002-), 男, 本科生, 研究方向为计算机图形渲染。E-mail: chenlizheme@outlook.com

designing a segmented function based on the grayscale value of the lighting model results, combining the screen space three-phase edge detection results with the screen space partition rendering results to obtain the final rendered output. Experimental results demonstrate that the proposed method can achieve non-photorealistic rendering of black and white comic style with relatively low resource consumption and reduced error rate, showcasing a distinctive black and white comic style.

**Keywords:** black and white comic style; non-photorealistic rendering; real-time rendering; specialized lighting model; edge detection

## 0 引言

随着游戏与动漫行业的快速发展,业界对非真实感渲染的需求日益增加,非真实感渲染一直是游戏与动漫领域的研究热点和难点<sup>[1]</sup>。怎样高效且优质地对场景进行非真实感渲染、并尽可能地降低人工成本,是互联网游戏与动漫行业的重要课题<sup>[2]</sup>。

传统的黑白漫画风格非真实感渲染主要依赖于艺术工作者手绘纹理贴图。该方法人工成本高,渲染资源开销大,且主观性较强。因此,本文主要研究一种面向游戏与动漫的黑白漫画风格非真实感实时渲染方法,在保证渲染精度的基础上,提升渲染速度、降低部署成本,为黑白漫画风格非真实感渲染提供快速高效的实现方法。

## 1 相关工作

随着非真实感渲染技术的不断发展和突破,该领域已经取得了较大进步。然而,当前的非真实感渲染方法并不能通过特定数值标准或特定的图像特征进行分类,因此难以形成针对性的特化方法<sup>[3]</sup>。

早期的非真实感渲染往往致力于实现多种非真实感渲染的统一模型。文献[4]基于 Phong 着色模型提出了色调偏移着色模型,使用亮度与色调两种方式指示模型表面方向与渲染区域,达到可调卡通着色的目的,但在风格差异化上表现不足。

随后,学者们更偏向于对单一风格做特化渲染,以强化渲染的特征性。文献[5]提出了水彩模拟的非真实感渲染方法,沿用特化光照模型、修改着色方式、构造着色系统的思路进行三维场

景的风格化渲染,重点模拟水彩画的颜料颗粒感,通过对边缘笔触、透明质感等细节的针对性模拟来加强水彩绘制风格特征。文献[6]也在粉笔画艺术风格的研究上采取了类似思路。

针对机械产品立体图的渲染重点和产品工程信息表述等,文献[7]利用计算机模拟出机械零件的手绘轴测图笔触效果。在基于图像的油画风格化绘制方面,文献[8]提出了基于多尺度笔刷的分层图像油画风格化绘制,来模拟艺术家的油画绘制过程,采用不同尺度的笔刷按照从粗到细的顺序逐层绘制。

随着激光扫描技术的发展,文献[9]提出面向高分辨率激光扫描点云的三维建筑物风格化渲染方法,快速且高效地将可视化风格信息从三维网格向三维点云进行传递。

在性能优化需求较高的应用情景下,非真实感渲染会尝试完全以屏幕空间所采集的数据绘制非真实感效果。此时,该非真实感渲染可以被称为风格化滤镜,常应用于视频和图像领域。文献[10]提出了一种基于边缘特征和超像素分割的图像和视频的多边形渲染方法。

黑白漫画风格的非真实感渲染,则长期依赖于艺术工作者来完成,即人工绘制符合画面着色规律的纹理贴图,并基于主观对画面逐帧添加光照效果。这一方法所绘制的黑白漫画非真实感画面人工成本高,且由于需要逐帧绘制多种纹理贴图,并调整光照效果,其渲染速度较慢、内存开销较高。由于模拟和绘制不同艺术风格时,非真实感渲染方法差别较大,实现方法与评估体系不尽相同<sup>[11]</sup>,其他非真实感渲染方法难以直接应用

于黑白漫画风格非真实感渲染上。受上述文献和黑白漫画风格非真实感渲染技术现状的启发, 本文主要研究面向游戏与动漫的黑白漫画风格非真实感实时渲染方法。

## 2 方法描述

本文方法主要通过修改光栅化着色器、并添加特殊后处理效果, 实现黑白漫画风格的非真实感渲染。设计特化光照模型, 对待渲染场景进行预处理; 进行屏幕空间三相边缘检测, 对光照模型结果依次进行深度边缘检测、法线边缘检测、颜色边缘检测, 并合并所有边缘检测结果; 进行屏幕空间分区域渲染, 设计网点着色与线条着色方法, 计算三维场景在屏幕空间的着色结果; 设计分段函数, 合并屏幕空间三相边缘检测结果与屏幕空间分区域渲染结果, 得到最终渲染结果, 实现黑白漫画风格的非真实感渲染。

### 2.1 设计特化光照模型

为了更精确地进行屏幕空间的渲染着色, 需要通过设计特化光照模型来进行预处理工作<sup>[12-13]</sup>。由于黑白漫画风格非真实感渲染在全局角度上遵循基本光照规律, 在细节及特定场景上与之相异, 故本文方法仍然沿用Blinn-Phong模型的思路, 将任意一点的光照分为三部分, 即环境光、高光反射与漫反射光来考虑。

#### 2.1.1 分析光照模型效果

在传统的黑白漫画美术风格中, 艺术工作者往往通过控制画面的明暗来加强对画面中重要部分的表现效果, 并弱化其他部分的表现效果。基于此, 该非真实感渲染方法中的光照模型须根据场景环境光照效果改变而突出画面中主要需要表现的角色, 并削弱画面中重要性较低的物体的表现效果。为在随后对物体屏幕空间分区域渲染时取得较好的效果, 物体所受到的光照应以梯度的形式突变, 形成灰度值区分较大的不同渲染区域。

为了表现黑白漫画效果中距离视角较近时详

细描绘、距离视角较远时略缩描绘的特点, 近处所具有的光照层次数应当多于远处。此外, 在对高光反射项处理时, 基于黑白漫画风格中对高光区域进行强调性处理, 本文直接沿用Blinn-Phong模型的高光反射项, 并将其数值增大一定倍数, 即可得到较好的效果。

#### 2.1.2 设计主体突出的环境光项

在黑白漫画场景中, 一般画面着重表现的物体定义为主体物, 与之相反的则是非主体物。主体物往往会采用区别于场景中其他物体的色调, 用以强调其存在。对于非主体物, 环境光对其产生正常影响; 对于主体物, 光照效果一定程度上因环境明暗变化而产生相反的变化, 同时对环境光变化产生一定的正向反馈。整合以上特性, 考虑到光照变化的平滑性质, 参考一般的Lerp插值计算方法, 即可构造出任意物体在环境光照为 $I$ 时所受到的光照度:

$$L_a = p_1(1 - I) + p_2 I \quad (1)$$

式中:  $L_a$  为物体所受环境光;  $I$  为环境光强度, 由光照强度的定义可知,  $0 \leq I \leq 1$ ;  $p_1$ 、 $p_2$  分别为主体物与非主体物的表面材质区分度。与一般插值方法类似, 式(1)的构造使其具有良好的平滑性质。考虑到修正后的光照度 $L_a$  仍应控制在 $0 \sim 1$ 之间, 且 $L_a$  在任意环境光强度下应均匀地变化, 故 $p_1$ 、 $p_2$  满足 $p_1 + p_2 = 1$ , 且 $p_1, p_2 \geq 0$ 。由此可以得到对于任意物体的环境光光照度:

$$L_a = p_1 + I - 2p_1 I \quad (2)$$

根据式(1), 可以得出系数 $p_1$ 的取值范围为区间 $[0, 1]$ 。当 $I = p_1 = 0.5$ 时, 函数取得对称形式。当 $p_1 \neq I$ 时, 式(2)中将会混合2个系数, 并根据 $I$ 的不同取值进行加权平均。通过图1展示的环境光变化函数图像可以观察到, 在 $p_1$ 较小时, 物体所受环境光与实际环境光呈正相关变化; 随着 $p_1$ 的逐渐增大, 物体所受环境光与实际环境光将逐渐变为负相关。在这一调节过程中, 物体所受环境光始终保持在 $0 \sim 1$ 之间。

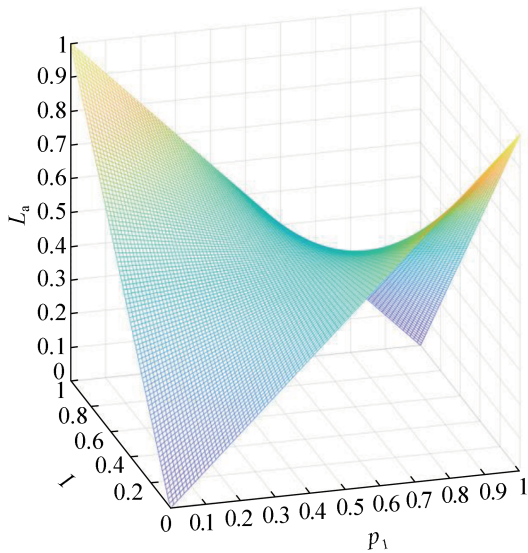


图1 环境光变化的函数图像  
Fig. 1 Function image of ambient light change

### 2.1.3 设计灰度分级的漫反射项

在进行漫反射时，由于黑白漫画美术风格中通常灰度数量较少，不同灰度之间表现方式突变，故需将漫反射所得到的光照结果分离为多个梯度，以方便后续着色。同时，考虑到该美术效果中物体的描绘复杂度随距离视点远近而产生变化，在色调分离时应当根据与屏幕的视点距离对分离梯度做出调整，在距离较近时增加光照度的层次分级数，距离较远时减少层次分级数。基于上述分析，本文设计了新的特化漫反射计算方法，主体物所受漫反射光为

$$L_d = \frac{p_d(k_d I_p \max(0, N \cdot L))}{p_{\max}} \quad (3)$$

式中： $p_d$ 为灰度层次度； $p_{\max}$ 为 $p_d$ 可取的最大值，通常来讲 $p_{\max} \in (0, 100]$ ，而 $p_d \in (0, p_{\max}]$ ； $k_d$ 为漫反射系数，用于表示物体对漫反射光的吸收率； $I_p$ 为光源的光照强度； $N$ 为物体表面上受光照点的法向量； $L$ 为光线在该点的入射方向。本文采用向下取整的方式确保漫反射计算出的光照结果能够被分离为多个梯度，以便后续着色。如果使用一般的四舍五入方式取整，可能会导致灰度层次变化不均匀，从而影响着色结果。灰度层次度主要用于度量模型着色受屏幕空间深度影响的变化幅

度。当该值增加时，光照分离产生的梯度也随之增加，其数值由不同物体的材质决定。深度变化范围为0~1之间，灰度层次度变化范围为0~10之间，此时设定原漫反射光照值为恒定值0.68。在使用该公式特化漫反射项后，当物体灰度层次度值增加时，物体随深度变化将产生更多的颜色层次，而当灰度层次度较小时，物体产生的颜色层次也相应减小。在极端情况下，将灰度层次度设置为0时，物体将仅存在一个颜色层次，且其固定色值为黑色。

### 2.1.4 合并光照模型各项

由于该光照模型具有线性性质，因此直接将环境光、漫反射光、高光项合并即可得到光照模型整体项<sup>[14]</sup>。

## 2.2 屏幕空间三相边缘检测

在利用光照模型结果进行分区域着色前，应当先在屏幕空间的深度、法线、颜色三个相位进行边缘检测，并得到光照模型的边缘检测图像，以防止着色结果对边缘检测结果产生干扰。

### 2.2.1 对光照模型结果进行深度边缘检测

由于深度值突变区域仅存在于物体的外层边缘处，区域面积较小，因此先对深度值进行边缘检测可以得到对光照模型输出干扰最小的检测结果。

相比较而言，Sobel算子处理方式简单、快速，可以满足本文的实时性与自动化要求。因此，本文采用Sobel算子进行边缘检测。

Sobel算子由横向、纵向两个 $3 \times 3$ 的矩阵构成，分别记作 $G_x$ 与 $G_y$ ，如式(4)所示。在图像上对两矩阵做卷积，得到的值即为图像的一阶梯度。

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (4)$$

随后，将该算子与屏幕深度空间结果进行卷积，即可得到图像梯度：

$$G_1 = \sqrt{G_x^2 + G_y^2} \quad (5)$$

以梯度大小将场景像素点与RGB值均为0的三维零向量进行插值, 即可得到描边后的图像:

$$f_{\text{depth}} = (1 - G_1)(0, 0, 0) + E \quad (6)$$

式中:  $E$  为原图像的色值。由式(5)和图像色值定义可知,  $f_{\text{depth}} \geq 0$ 。

### 2.2.2 对深度边缘检测结果进行法线边缘检测

得到深度边缘检测结果后, 为得到场景中具有结构性突变的区域边缘, 将进行对法线空间的边缘检测。对法线空间的边缘检测将会在物体表面具有结构性突变的区域检测出边缘。

与2.2.1中所描述的深度空间边缘检测类似, 在进行法线空间边缘检测时, 为了确保整体检测的实时性, 仍然对法线空间图像使用Sobel算子进行卷积。使用卷积结果  $G_2$  作为参量对检测结果与深度边缘检测后所得色值进行插值, 即可得到法线空间边缘检测结果:

$$f_{\text{normal}} = (1 - G_2)(0, 0, 0) + f_{\text{depth}} \quad (7)$$

同理可知,  $f_{\text{normal}} \geq 0$ 。

### 2.2.3 对法线边缘检测结果进行颜色边缘检测

在进行法线边缘检测后, 为强化上述屏幕深度空间、法线空间的边缘检测效果, 并去除杂色, 对法线空间边缘检测结果进行颜色空间边缘检测。同时, 对颜色空间的边缘检测也将获取上两轮边缘检测中遗漏的法线与深度都近似、且颜色变化突兀的边缘。为达到上述目的, 与法线空间边缘检测结果不同, 颜色空间的边缘检测直接在法线空间检测结果的基础上进行, 而颜色空间的边缘检测结果  $G_3$  同样也将作为参量, 对法线空间边缘检测的结果进行插值以得到颜色空间处理结果:

$$f_{\text{color}} = (1 - G_3)(0, 0, 0) + f_{\text{normal}} \quad (8)$$

同理可知,  $f_{\text{color}} \geq 0$ 。

### 2.2.4 线性插值合并边缘检测结果

通过以上计算, 即可分别得到深度空间、法线空间、颜色空间的边缘检测结果。为了进一步

强化边缘检测结果, 并排除不合理的多余边缘, 对深度、法线及颜色空间的边缘检测结果进行线性插值:

$$f = a_1 f_{\text{depth}} + a_2 f_{\text{normal}} + a_3 f_{\text{color}} \quad (9)$$

根据  $f_{\text{depth}}$ ,  $f_{\text{normal}}$ ,  $f_{\text{color}}$  的定义可知,  $a_1$ 、 $a_2$ 、 $a_3$  满足  $a_1 + a_2 + a_3 = 1$ 。同时, 根据  $f$  值的定义, 还应满足  $a_1, a_2, a_3 \geq 0$ 。

如图2所示, 在传统图像处理方法中, 往往仅通过屏幕的颜色空间进行边缘检测, 这一做法将会丢失部分结构信息<sup>[5]</sup>。同时, 画面在仅通过深度、法线与颜色三者之一的空间进行检测后, 无法完整描述屏幕空间的结构性边缘信息。而在采用本文方法对深度、颜色与法线三个空间边缘检测进行边缘检测并完成一系列插值处理后, 最终结果记录了明显的结构信息。

## 2.3 设计屏幕空间分区域渲染

在得到使用边缘检测后的特化光照模型区域划分处理结果后, 本文将黑白漫画的亮度拆分成多个灰度区, 并在不同灰度区中应用不同的着色方式, 就可以得到屏幕空间分区域渲染图像。

### 2.3.1 设计网点着色方法

在对灰度值较高部分进行网点着色时, 利用屏幕空间uv均匀变换的性质, 对于屏幕空间上任意一点  $P$ , 取其坐标的  $r$  倍以及小数部分, 计算这个小数部分与点(0.5,0.5)的距离。通过利用二阶范数的连续变化性质, 可以计算任意位置的色值, 点  $P$  的色值为

$$C_1 = \|(rP - [rP]) - (0.5, 0.5)\|_2 \quad (10)$$

此时可以得到灰度渐变的大量圆点状图案。为控制圆点图案的大小, 并消除渐变效果, 应对该图案进行二值化处理, 以得到仿网点式的贴图效果。

图3为仅采用网点对目标模型进行着色的效果。

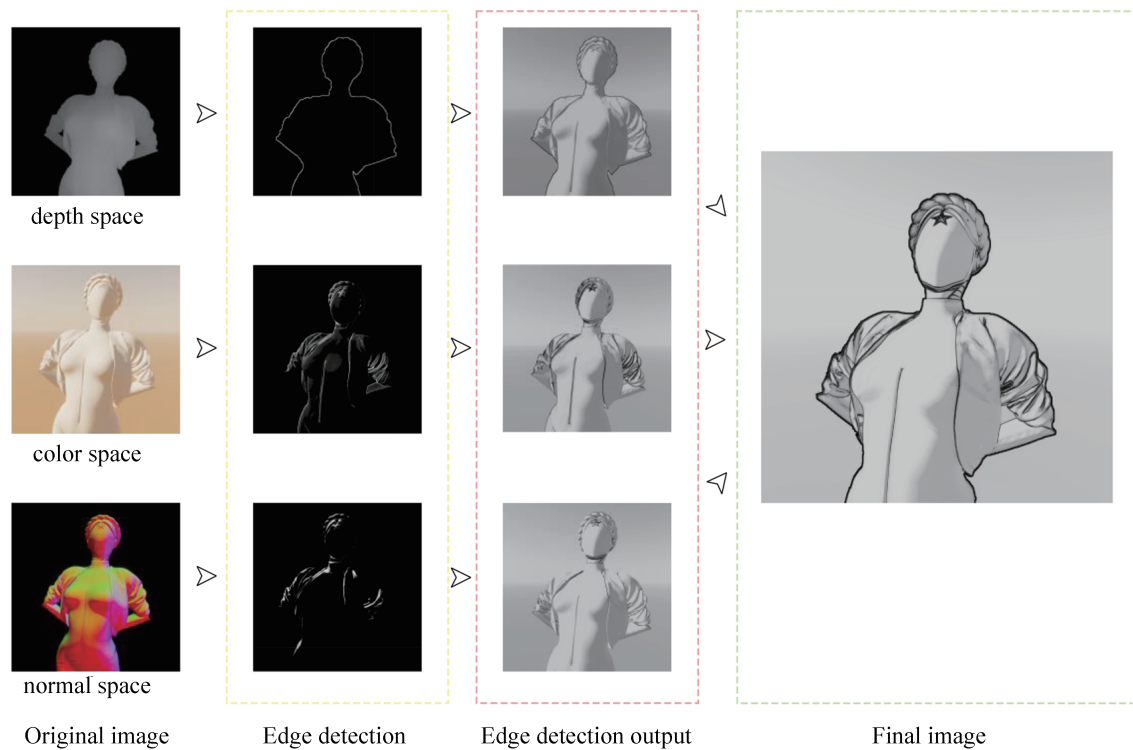


图2 边缘检测后的合并结果  
Fig. 2 Merge result of edge detection

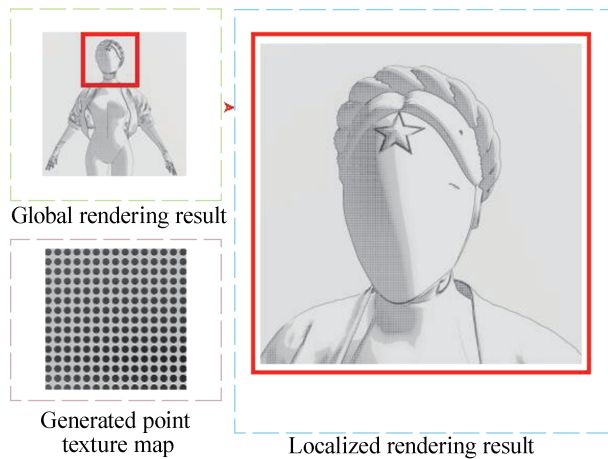


图3 使用生成的网点纹理对目标进行着色  
Fig. 3 Target shade with generated dot texture

### 2.3.2 设计线条着色方法

为形成平行的排线，本文定义  $a$  作为所形成排线的斜率，其中两条相邻直线之间的距离为  $d$ 。由于要形成数条斜率相等、距离相等的直线，因此需保证各直线之间的距离  $d$  相等。要创造满足上述条件的线条，可以将两直线距离转化为其在坐标系  $y$  轴上的截距，再对此截距取模，即可得

到特定间距的平行线组：

$$C_2 = \lfloor w |y - ax| \rfloor \bmod [k^2 + d^2 + 1] \quad (11)$$

式中： $w$  为非负整数，用于控制直线的宽度。该公式用于具体控制屏幕空间某一点的色值。此外，本文采用了利用光照度对贴图生成进行干涉的方法，替换了渲染区的背景颜色，以此削减图像的摩尔纹效应。考虑场景立体感，不同面的线条方向应该在一定程度上不一致，因此可以使用物体法线来控制线条方向，如式(12)所示。

$$C_2 = \lfloor w |y - N \cdot ax| \rfloor \bmod [k^2 + d^2 + 1] \quad (12)$$

图4展示了使用线条对目标屏幕空间进行着色后得到的着色效果。

### 2.3.3 使用插值方法处理着色边界

为了保证结果图像纹理效果的连续性，应对分区渲染的边界位置使用线性插值的方式修改其灰度。对于在屏幕空间上光照模型结果灰度值为  $I$  的某一区域，其插值灰度值为

$$C_m = C_1 I + C_2 (1 - I) \quad (13)$$



式中:  $I$ 为插值系数, 且满足  $t \leq 1$ 。在其值为0.5时, 两种渲染方法将平均地作用在像素点上。由于插值本身的特性, 该方法将会使渲染区域交界处产生精度损失, 然而这一损失将在可接受范围内。

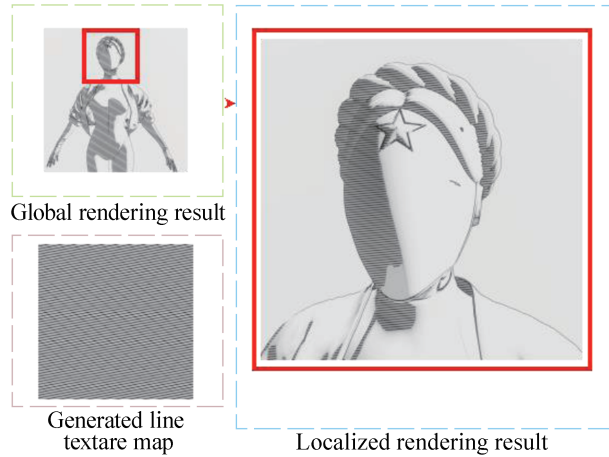


图4 使用生成的线条纹理对目标进行着色  
Fig. 4 Target shade with generated line texture

### 2.3.4 合并屏幕空间分区域渲染结果

在合并屏幕空间分区域渲染在各个区域的渲染结果时, 以光照模型结果灰度值为分段依据, 将网点着色区、两着色区的插值处理区以及线条着色区合并至一张图像上, 如式(14)所示。

$$C = \begin{cases} C_1, I > a_1 \\ C_m, a_1 \geq I \geq a_2 \\ C_2, I < a_2 \end{cases} \quad (14)$$

式中:  $I$ 为光照模型的灰度值;  $a_1$ 、 $a_2$ 为渲染方法的分界位置。当原屏幕颜色空间灰度值大于 $a_1$ 时, 将会采用网点着色方式; 当灰度值小于 $a_2$ 时, 将会采用线条着色方式; 当灰度值处于 $a_1$ 、 $a_2$ 之间时, 将会采用插值方式处理着色边界。

## 2.4 合并结果

在合并屏幕空间分区域渲染结果与屏幕空间三边边缘检测结果时, 采用正片叠底的合并方式可以最大限度地保留两结果的特征。图5为应用这一方法后所得到的着色结果。

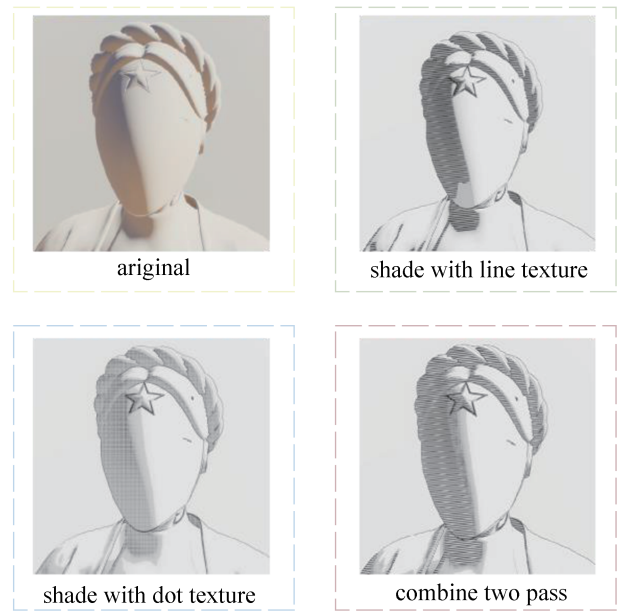


图5 结合两种纹理得到的最终渲染效果  
Fig. 5 Final rendering results obtained by combining two textures

## 3 实验与结果评估

为了验证本文方法的可行性与质量, 本文设计了多个性能评估实验与视觉效果评估实验。实验所用计算机系统的硬件环境为AMD Ryzen 9 5900HS with Radeon Graphics处理器, 32 GB内存, NVIDIA GeForce RTX 3080 Laptop显卡, 操作系统为Windows11, 开发语言为C++。

### 3.1 实验准备

在进行实验前, 生成了100个由长方体、圆柱体等常见几何物体组合而成的三维场景作为测试场景。在生成这些随机场景时, 进行了一定条件约束: 在所有场景中, 顶点数限定在100~10 000 000之间, 三角面数限定在33~10 000 000之间; 场景中最多存在500个光源, 在最少情况下则不存在光源。以上提到的所有场景信息均保证同时出现在屏幕空间中。

除随机生成的场景外, 在进行视觉效果评估实验时, 本文对一系列不同的三维模型进行了渲染, 并展示了该方法相比于传统渲染方法在性能

和视觉效果上的优势。这些三维模型的具体信息如表1所示。

表1 实验用三维模型详细信息  
Table 1 Detailed information of 3D models used in experiments

模型名称	模型顶点数	模型面数	类型
Cornell Box	56	42	场景
Utah Teapot	3 644	6 320	物体
Chess Set	26 464	53 072	物体
Stanford Bunny	35 947	69 451	动物角色
Zuo Ling	46 548	109 169	人物角色
You Ling	61 823	128 472	人物角色
Stanford Dragon	871 414	1 742 368	动物角色

### 3.2 评价指标

本文使用三项指标评价渲染结果，分别是渲染时间、内存占用与图片视觉分数。

在性能评估实验中，主要应用渲染时间和内存占用两项评价指标。渲染时间指渲染器从加载场景开始至渲染结束所使用的时间开销，分别取使用CPU与GPU的渲染时间，并与常规Blinn-Phong模型渲染方法所用时间进行对比。内存占用指除场景加载、映射与光栅化之外的额外内存开销。

在视觉效果评估实验中，主要使用了图片视觉分数这一指标。该指标的获取方法是邀请若干位志愿者观看参考图像和目标图像，假定参考图像的得分为3分，要求他们对目标图像相对于参考图像的视觉感受给出0~5分的评价。若干位志愿者对某一图像的打分平均值即为该图像的主观评分。通过对特定方法生成的一系列渲染图片序列进行评分，计算其评分平均值，即可得到该方法的图片视觉分数。

### 3.3 性能评估实验

渲染方法的可行性与其资源开销密切相关，因此，本文设计了两种性能评估实验，即时间复杂度分析和空间复杂度分析。

#### 3.3.1 时间复杂度分析实验

通过使用本文方法逐一渲染所有测试场景，得出在场景顶点、三角面与光源数量变化时，渲染平均时间开销变化的图像。

如表2~4所示，本文方法的渲染时间开销与场景中顶点和三角面数量关系性较弱，同时与场景中光源数量基本呈线性相关。

表2 顶点数量与时间开销的关系  
Table 2 Relationship between vertex count and time cost

顶点数量	平均CPU时间 开销/ms	平均GPU时间 开销/ms
10	21.47	6.26
100	10.16	17.85
1 000	6.60	2.32
10 000	49.64	17.81
50 000	33.57	14.92
100 000	25.83	13.72
1 000 000	43.27	27.55
10 000 000	23.41	8.83

表3 三角面数量与时间开销的关系  
Table 3 Relationship between triangle count and time cost

三角面数量	平均CPU时间 开销/ms	平均GPU时间 开销/ms
10	12.56	2.32
100	24.42	5.54
1 000	24.91	4.94
10 000	38.91	9.98
50 000	42.33	14.06
100 000	32.14	8.64
1 000 000	33.65	11.58
10 000 000	42.53	22.23

表4 光源数量与时间开销的关系  
Table 4 Relationship between number of light sources and time cost

光源数量	平均CPU时间 开销/ms	平均GPU时间 开销/ms
10	1.94	0.63
50	5.39	2.42
100	13.94	6.18
200	19.38	8.56
300	29.98	13.77
400	43.63	19.84
500	59.13	29.38

分析可知, 本文方法在渲染时, 仅通过光照模型向屏幕空间给出的标注进行着色, 该着色阶段仅会对屏幕空间大小范围内的所有像素点进行着色, 而像素点数量仅与屏幕本身相关。当进行光照时, 由于光强度的可加性, 本文方法将逐一遍历所有光源, 将其应用于屏幕空间中顶点。综上所述, 光源数量与时间开销呈线性关系。

### 3.3.2 空间复杂度分析实验

与时间复杂度的测试相同, 使用本文方法逐一渲染所有测试场景, 可以得出随场景顶点、三角面与光源数量变化时, 渲染平均内存开销的变化。

如图6所示, 当分别按顶点数、三角面数与光源数对样本场景排序并依次渲染后, 可以发现在场景中顶点、三角面和光源增多的情况下, 渲染内存开销都出现了上升的情况, 且上升比例相近。分析可知, 渲染内存开销取决于场景中存储的信息量和屏幕空间大小。当屏幕空间本身不改变时, 场景中存储的信息量将线性地改变内存开销状况。

### 3.3.3 与纹理贴图方法性能对比实验

业界通常对模型应用黑白纹理贴图, 并对其进行渲染以模拟黑白漫画美术风格的光照效果与画面风格<sup>[16]</sup>。这类方法往往可以得到较好的单帧渲染效果, 但由于贴图本身的适用性问题, 在该特殊情景中, 往往需要工作者对贴图UV进行手动或自动化的复杂调整, 以此适配变化的光照与场景<sup>[17]</sup>。因此该方法更多应用于单帧渲染<sup>[18]</sup>。

为验证本文方法的可靠性, 采用简单二值化方法获取场景渲染的灰度图片参照组, 将通过本文方法与业界成熟的方法所得到的渲染结果进行对比, 综合评估本文方法的效果。在一些简单场景中, 以划分颜色梯度的方式取得参照组, 并同时使用纹理贴图方法和本文所设计的渲染方法得到2种方法对同一场景的黑白漫画表现结果。依据场景的复杂程度, 进行不同阶段的测试。

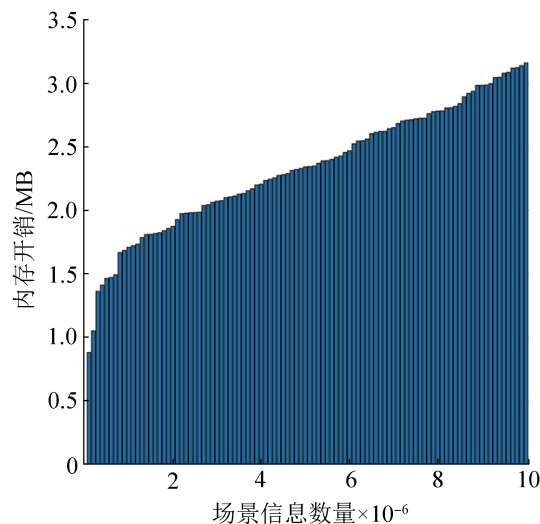


图6 渲染时内存消耗随场景信息变化

Fig. 6 Memory consumption for rendering varies with changes in scene information

在第一轮测试中, 选取了样本场景里顶点数大于1 000 000、三角面数多500 000且光源数大于250个的复杂场景进行渲染测试。结果显示, 在渲染的错误率上, 本文方法要劣于传统方法, 然而在渲染时间与渲染内存占用上, 本文方法优势明显。分析可知, 本文方法不需要对物体逐一标注纹理贴图, 而是直接根据屏幕获取的光照模型信息, 逐像素地修改屏幕空间颜色即可。表5记录了复杂场景中两种方法的各项指标对比。

在第二轮测试中, 选取了样本场景里顶点数大于10 000、三角面数大于8 000且光源数大于50的一般场景进行渲染测试。结果显示, 即使忽略了手动标注贴图的时间, 本文方法仍然在时间与空间复杂度上具有巨大的优势, 同时, 在这类场景中, 由于光照信息相比复杂场景而言更简单, 本文的渲染错误率也大幅下降。如表6所示。

在第三轮测试中, 选取了样本场景里顶点数小于100、三角面数小于100且光源数小于10的简单场景进行渲染测试。与其余场景测试结果一致, 本文方法仍然在性能开销上具有较高的优势, 同时, 由于光照信息进一步减少, 本文方法渲染结果的错误率也有所下降。如表7所示。

表 5 复杂场景测试性能开销统计

Table 5 Performance cost statistics for complex scene testing

渲染方法	手绘纹理贴图数量	平均渲染空间开销/MB	CPU 平均渲染时间开销/ms	GPU 平均渲染时间开销/ms
纹理贴图方法	4	11.23	32.14	9.13
	8	17.19	48.32	16.32
	16	30.06	78.49	28.97
本文方法	0	3.18	8.01	1.69

表 6 普通场景测试性能开销统计

Table 6 Performance cost statistics for normal scene testing

渲染方法	手绘纹理贴图数量	平均渲染空间开销/MB	CPU 平均渲染时间开销/ms	GPU 平均渲染时间开销/ms
纹理贴图方法	4	7.30	9.51	3.33
	8	9.12	28.43	5.18
	16	11.63	48.92	10.97
本文方法	0	2.54	3.54	0.52

表 7 简单场景测试性能开销统计

Table 7 Performance cost statistics for simple scene testing

渲染方法	手绘纹理贴图数量	平均渲染空间开销/MB	CPU 平均渲染时间开销/ms	GPU 平均渲染时间开销/ms
纹理贴图方法	4	2.92	2.14	6.54
	8	5.48	3.32	3.75
	16	8.65	5.40	2.32
本文方法	0	1.29	0.52	0.39

经过三轮对比测试可以得出，本文方法在性能开销上远优于传统手绘贴图方法。同时，本文方法的错误率控制在较低范围内，可以认为本文方法是传统方法在损失一定渲染精度的条件下的替代方法。

### 3.4 视觉效果评估实验

渲染方法的质量取决于其视觉效果，因此，本文设计了两个视觉效果评估实验，即消融实验和与纹理贴图方法对比实验。

#### 3.4.1 消融实验

在消融实验中，以完整的本文方法作为参照组，分别消除或替换了本文方法中特化光照模型、屏幕空间三相边缘检测，以及屏幕空间分区域渲染这三个步骤，并得到其图片视觉分数评价，以此研究本文方法各个步骤对整体渲染效果的贡献。

首先进行关于光照模型选用的消融实验。本文采用了常规光照模型参与本文方法的渲染流程作为实验组，并以完整的本文方法作为对照组，

对所有预先准备的三维模型场景分别进行了渲染，取得图片视觉分数。图 7 是两种方法在某一特定角度渲染的 Cornell Box 场景示例。在该实验中，选取场景二值化方法生成的渲染图像作为参考图像。

表 8 展示了分别应用普通光照模型和特化光照模型在各个场景取得的图片视觉分数。由表可知，在对各个场景模型的渲染中，采用特化光照模型的渲染图像具有更高的图片视觉分数，应用特化光照模型参与本文方法流程所生成的图像相比应用普通光照模型参与本文方法流程所生成的图像具有更好的视觉表现效果。同时，如表 8 所示，相比人物或动物角色，本文方法在面对一般物体和机械部件时具有更好的渲染效果。该实验证明了本文方法中应用特化光照模型的必要性。

随后进行关于屏幕空间三相边缘检测的消融实验。在该实验中，设立了三个实验组，分别消除了进行边缘检测的某一特定维度作为边缘检测结果，参与本文方法的后续流程。在该实验中，以去除边缘检测步骤的本文方法生成的图像作为

参考图像, 以完整的本文方法作为对照组。图 8 展示了屏幕空间三相边缘检测的消融实验结果。

如图 8 所示, 同时应用深度空间、颜色空间和法线空间进行边缘检测后, 渲染结果具有更好的视觉表现。同时, 在机械部件或简单场景中, 颜色空间和法线空间边缘检测对整体效果的影响相比深度空间边缘检测要大, 而在人物角色等复杂物体中, 深度空间的边缘检测则对视觉效果产生了更大的影响。该实验证明了本文方法中应用

屏幕空间三相边缘检测的必要性。

最后进行关于屏幕空间分区域渲染的消融实验。在该实验中, 设立了两个实验组, 分别消除了分区域渲染中的网点着色样式和线条着色样式, 参与本文方法的后续流程。在该实验中, 以未进行屏幕空间分区域渲染的渲染结果作为参考图像, 以完整的本文方法作为对照组。图 9 展示了屏幕空间分区域渲染的消融实验结果。

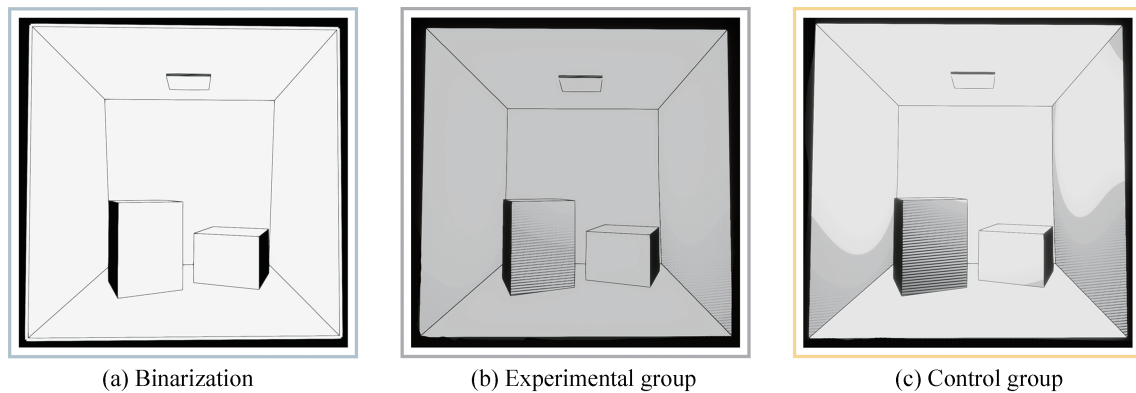


图 7 Cornell Box 场景在特定视角下的渲染效果示例  
Fig. 7 Rendering example of Cornell Box scene from a specific view angle

表 8 不同光照模型对渲染效果影响的消融实验结果  
Table 8 Ablation experiment results on influence of different lighting models on rendering effect

渲染方法	Cornell Box	Utah Teapot	Chess Set	Stanford Bunny	Zuo Ling	You Ling	Stanford Dragon
二值化处理	3.00	3.00	3.00	3.00	3.00	3.00	3.00
普通光照	4.72	4.43	4.78	4.43	4.15	4.19	3.89
特化光照	4.96	4.83	4.83	4.59	4.51	4.37	4.19

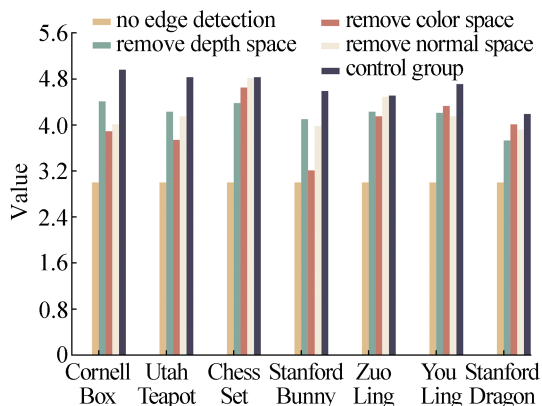


图 8 屏幕空间三相边缘检测的消融实验结果  
Fig. 8 Results of ablation experiment on screen space three-phase edge detection

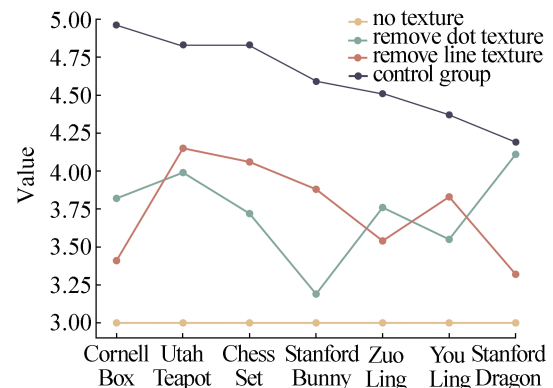


图 9 关于屏幕空间分区域渲染的消融实验结果  
Fig. 9 Results of ablation experiment on screen space partition rendering

如图 9 所示，去掉分区域渲染中网点着色样式或线条着色样式，均会对渲染结果产生较大影响；同时，两种着色样式对渲染结果的影响取决于模型的具体拓扑结构，不具有特定趋势。该实验证明了本文方法中应用屏幕空间分区域渲染的必要性。

综合各消融实验结果可知，本文方法中的特化光照模型、屏幕空间三边边缘检测和屏幕空间分区域渲染均是必要步骤；其中，特化光照模型对渲染结果的贡献最小，屏幕空间分区域渲染对渲染结果的贡献最大。

### 3.4.2 与纹理贴图方法视觉效果对比实验

为进一步证明使用本文方法得到的视觉效果优于一般方法，本文进行了与纹理贴图方法的对比实验。在该实验中，对每个实验用三维模型分别应用了两种方法，取得模型的多角度渲染结果，同时以简单二值化方法为参考图像，得到两种方法在各个模型上的图片视觉分数评价，以此研究本文方法与纹理贴图方法的视觉效果对比。图 10~11 展示了使用本文方法渲染和使用纹理贴图方法渲染得到的部分图片，及其图片视觉分数。

如图 10~11 所示，与传统纹理贴图方法相比，本文方法在机械零件、物体与场景渲染中得到的图片视觉分数更高，在这一类物体的渲染上具有较大的优越性；在动物角色或人物角色的渲染方面，本文方法所得到的图片视觉分数比纹理贴图方法略高，效果相比纹理贴图方法更好。就本文方法自身而言，当模型面数增加时，本文方法所得到的渲染结果将逐渐变差。综上所述，本文方法在大多数应用情境下，可以产生相比纹理贴图方法而言更好的渲染结果，且在渲染面数较低的简单物体时，具有更好的表现效果。

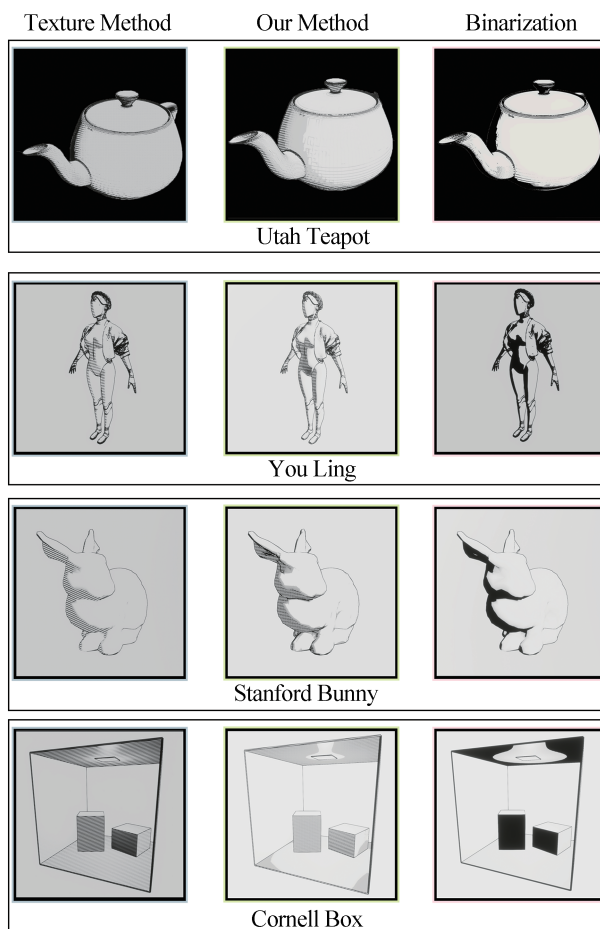


图 10 视觉效果对比实验中选取的部分渲染图片  
Fig. 10 Some rendered images for visual effect comparative experiment

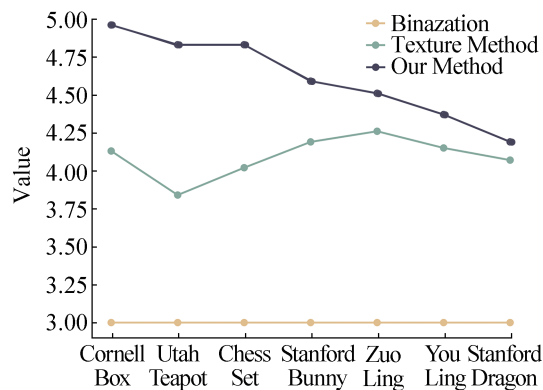


图 11 视觉效果对比实验结果  
Fig. 11 Results of visual effect comparative experiment

## 4 结论

本文提出的游戏与动漫的黑白漫画风格非真实感渲染方法在模仿黑白漫画美术风格方面取得了较好的视觉效果。通过改进的特化光照模型和屏幕空间三相边缘检测方法, 实现了结构层次的描边效果, 并通过分区域渲染的方式完成了实时渲染。与传统的纹理贴图方法相比, 该方法在渲染效率和视觉效果方面表现明显优越。

此外, 该实时渲染方法对计算机仿真领域具有重要贡献。首先, 该实时渲染方法提供了一种实现黑白漫画风格的非真实感渲染方法, 使得计算机生成的三维场景能够呈现出逼真的黑白漫画效果, 对于游戏开发、动漫制作以及虚拟现实等领域具有很大的潜力和应用价值。然后, 该实时渲染方法通过引入特化光照模型和屏幕空间深度等技术, 不仅可以用于黑白漫画风格的渲染, 还可以在艺术设计、电影特效和虚拟现实等领域轻松实现其他艺术风格的非真实感渲染。最后, 该实时渲染方法对于游戏开发者和动漫制作团队而言, 能够更加高效地生成黑白漫画风格的内容。

综上所述, 该实时渲染方法在模仿黑白漫画美术风格方面取得了显著成果。未来的研究可以进一步完善阴影和散射的渲染效果, 保证渲染方法的可微分性质, 提升渲染的灵活性。

### 参考文献:

- [1] Kumar M P P, Poornima B, Nagendraswamy H S, et al. A Comprehensive Survey on Non-photorealistic Rendering and Benchmark Developments for Image Abstraction and Stylization[J]. *Iran Journal of Computer Science*, 2019, 2(3): 131-165.
- [2] Sofiane Ouazaa, Javier Burguete, M Pilar Paniagua, et al. Simulating Water Distribution Patterns for Fixed Spray Plate Sprinkler Using the Ballistic Theory[J]. *Spanish Journal of Agricultural Research*, 2014, 12(3): 850-863.
- [3] 钱文华, 曹进德, 徐丹, 等. 非真实感绘制技术研究现状与展望[J]. *中国图象图形学报*, 2020, 25(7): 1283-1295. Qian Wenhua, Cao Jinde, Xu Dan, et al. Research Status and Progress of Non-photorealistic Rendering Technology[J]. *Journal of Image and Graphics*, 2020, 25

- (7): 1283-1295.
- [4] Gooch A, Gooch B, Shirley P, et al. A Non-photorealistic Lighting Model for Automatic Technical Illustration[C]// *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*. New York, NY, USA: Association for Computing Machinery, 1998: 447-452.
- [5] 康丽锋, 唐棣. 基于彩色形态学的水彩画模拟[J]. *计算机工程与应用*, 2010, 46(5): 157-159, 242. Kang Lifeng, Tang Di. Watercolor Simulation Based on Color Morphology[J]. *Computer Engineering and Applications*, 2010, 46(5): 157-159, 242.
- [6] 钱文华, 徐丹, 官铮, 等. 粉笔画艺术风格模拟[J]. *中国图象图形学报*, 2017, 22(5): 622-630. Qian Wenhua, Xu Dan, Guan Zheng, et al. Simulating Chalk Art Style Painting[J]. *Journal of Image and Graphics*, 2017, 22(5): 622-630.
- [7] 王浩田, 戴立玲. 机械零件的非真实感渲染研究[J]. *计算机工程与设计*, 2012, 33(11): 4377-4381. Wang Haotian, Dai Liling. Research on Rendering Mechanical Parts in Non-photorealistic Style[J]. *Computer Engineering and Design*, 2012, 33(11): 4377-4381.
- [8] 陈颖, 荆树旭, 石剑, 等. 基于多尺度笔刷的分层油画风格化[J]. *计算机辅助设计与图形学学报*, 2020, 32(4): 575-581. Chen Ying, Jing Shuxu, Shi Jian, et al. Layered Oil Painting Stylization with Multi-scale Brushes[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2020, 32(4): 575-581.
- [9] 唐翔, 王伟鹏, 罗欢, 等. 面向高分辨率激光扫描点云的三维建筑物风格化渲染方法[J]. *福州大学学报(自然科学版)*, 2021, 49(6): 761-766. Tang Xiang, Wang Weipeng, Luo Huan, et al. A Method on Stylized Rendering of 3-D Building via High-resolution Laser Scanning Point Clouds[J]. *Journal of Fuzhou University(Natural Science Edition)*, 2021, 49(6): 761-766.
- [10] 王一凡, 赵乐义, 李毅. 基于生成对抗网络的图像动漫风格化[J]. *计算机工程与应用*, 2022, 58(18): 104-110. Wang Yifan, Zhao Leyi, Li Yi. Image Animation Stylization Based on Generative Adversarial Network[J]. *Computer Engineering and Applications*, 2022, 58(18): 104-110.
- [11] 王少荣, 敖知琪, 要曙丽, 等. 自适应的线条画绘制[J]. *中国图象图形学报*, 2018, 23(5): 730-739. Wang Shaorong, Ao Zhiqi, Yao Shuli, et al. Adaptive Line Drawing[J]. *Journal of Image and Graphics*, 2018, 23(5): 730-739.

- [12] 杜玺伦, 谭家万. 真实感图形学的光照模型研究综述[J]. 电子技术与软件工程, 2014(1): 112-113.
- [13] 孟祥艳, 任郁苗, 潘海仙. 真实感体绘制中的光照与阴影模型算法研究[J]. 激光杂志, 2016, 37(8): 141-144.  
Meng Xiangyan, Ren Yumiao, Pan Haixian. Algorithm for Illumination and Shadow Model in Realistic Volume Rendering[J]. Laser Journal, 2016, 37(8): 141-144.
- [14] Zhang Yuechen, He Zexin, Xing Jinbo, et al. Ref-NPR: Reference-based Non-photorealistic Radiance Fields for Controllable Scene Stylization[C]//2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NJ, USA: IEEE, 2023: 4242-4251.
- [15] 蒋宝吉, 唐棣. 基于三维模型的风格化线条渲染方法[J]. 计算机科学, 2013, 40(10): 289-291, 308.  
Jiang Baoji, Tang Di. Research on Stylized Lines Rendering of 3D Models[J]. Computer Science, 2013, 40(10): 289-291, 308.
- [16] Lorenzo Scalera, Stefano Seriani, Alessandro Gasparetto, et al. Non-photorealistic Rendering Techniques for Artistic Robotic Painting[J]. Robotics, 2019, 8(1): 10.
- [17] Jong Hyun Kim, Jung Lee. Layered Non-photorealistic Rendering with Anisotropic Depth-of-field Filtering[J]. Multimedia Tools and Applications, 2020, 79(1): 1291-1309.
- [18] Jutta Schumann, Thomas Strothotte, Stefan Laser, et al. Assessing the Effect of Non-photorealistic Rendered Images in CAD[C]//Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. New York, NY, USA: Association for Computing Machinery, 1996: 35-41.