Non-photorealistic Radiance Remapping

Kohei Doi
Kyushu University
doi.kohei.682@s.kyush-u.ac.jp

Yuki Morimoto
Kyushu University
morimoto@design.kyushu-u.ac.jp

ABSTRACT

We present a color mapping method that corresponds to intensity and considers the interaction of light among objects in a 3-dimensional scene. Previous methods that map color according to geometric specifications or a rendered result do not reproduce the interreflection of surrounding objects. Using a path tracer, the proposed method replaces the radiance with an arbitrary one-dimensional texture in the sampled light transport path. The texture appearing on the object is calculated by a weighted average of the modified estimations. The proposed method can generate color variation in physically based intensity. It hence can reproduce physical events such as caustics or color bleeding while retaining the flexibility of rendering within a path tracing framework.

CCS CONCEPTS

• **Computing methodologies** → Non-photorealistic rendering.

KEYWORDS

Stylized rendering, Path tracing

ACM Reference Format:
Kohei Doi and Yuki Morimoto. 2019. Non-photorealistic Radiance Remapping. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Posters (SIGGRAPH '20 Posters), August 17, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3388770.3407395

1 METHOD

Cartoon shading [Lake et al. 2000] applies a one-dimensional texture to simplify shading using surface normal and light direction. Image processing can also realize stylized rendering. [Magdics et al. 2013] implemented color quantization and shadow recoloring in the shading stages of the graphics pipeline. However, these methods only partially consider events caused by global illumination.

Our method uses path tracing (PT) to modify the appearance of materials by applying an arbitrary one-dimensional texture to the visual intensity of a scene (Figure 1). At every evaluation of the sampled light path, texture is applied with a weight depending on the path. The texture coordinates and weights are computed from the intensity.

SIGGRAPH '20 Posters, August 17, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7973-1/20/08.
https://doi.org/10.1145/3388770.3407395

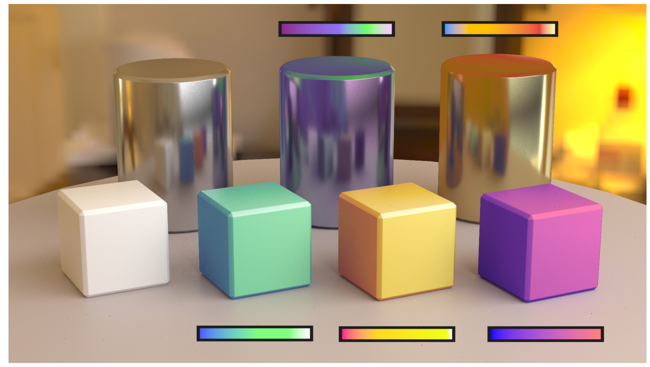

Figure 1: Our result with 1-dimensional textures applied to all objects except for the leftmost.

1.1 Path Tracing Using Non-photorealistic Color

Our aim is to derive the non-photorealistic outgoing radiance in direction $\boldsymbol{\omega_o}$ at a point $\boldsymbol{x}$ from L_P, which is the rendering equation [Kajiya 1986]

$$L_P = L_e + \int_{\mathcal{S}^2} f(\boldsymbol{\omega_i},\ \boldsymbol{\omega_o})\, L_i(\boldsymbol{\omega_i})\, |\mathbf{n}\cdot\boldsymbol{\omega_i}|\, d\boldsymbol{\omega_i}, \tag{1}$$

where $\mathcal{S}^2$ is the sphere over $\boldsymbol{x}$, $\mathbf{n}$ is the surface normal, $\boldsymbol{\omega_i}$ is the incoming direction, L_e and L_i respectively represent the emission and incoming radiance, and f is any bidirectional scattering distribution function (BSDF). To estimate L_P, a Monte-Carlo integrator samples $\boldsymbol{\omega_i}$ from an arbitrary probability density p to obtain

$$\hat{L}_P = L_e + \frac{f(\boldsymbol{\omega_i},\ \boldsymbol{\omega_o})\, L_i(\boldsymbol{\omega_i})\, |\mathbf{n}\cdot\boldsymbol{\omega_i}|}{p(\boldsymbol{\omega_i})}. \tag{2}$$

With physically based radiance $\hat{L}_P$, we derive non-photorealistic radiance L_{NP}. We refer to a one-dimensional texture $T(u)$ defined on $[0, 1]$, where u is mapped from $\hat{L}_P$. Because we average the modified samples to obtain L_{NP}, the texture value needs to be weighted with W according to the contribution of $\hat{L}_P$ to L_P. By averaging the weighted texture value, the non-photorealistic appearance is expressed as

$$L_{NP} = \int_{\mathcal{S}^2} W\left(\hat{L}_P\right) T\left(u\left(\hat{L}_P\right)\right) d\boldsymbol{\omega_i} \tag{3}$$

$$\cong \frac{W\left(\hat{L}_P\right)}{p(\boldsymbol{\omega_i})} T\left(u\left(\hat{L}_P\right)\right). \tag{4}$$

1.2 Weighting and Mapping of a Texture

In this section, we describe how to compute weight $W(\hat{L}_P)$ to combine the samples and texture coordinate $u(\hat{L}_P)$. To define these functions, we use intensity I, which we obtain with a tonemap function.

$$I = tonemap\left(\hat{L}_P\right) \tag{5}$$

We define weight W as follows:

$$\frac{W\left(\hat{L}_P\right)}{p(\boldsymbol{\omega_i})} = (1-\alpha)\max\{I,\ w_{min}\} + \alpha\hat{L}_P. \tag{6}$$

The first term is the weight according to the intensity, which is basically proportional to I. We use minimum weight w_{min} to prevent $\hat{L}_{NP}$ from becoming black with small I. In addition, a user can control the weight of the incoming color. The weight according to the intensity and $\hat{L}_P$ are interpolated with $\alpha \in [0, 1]$ to compute the final W.

Texture coordinate u is defined as the normalization of I to $[0, 1]$ from the range $[I_{min},\ I_{max}]$, which is provided by the user.

$$u = \begin{cases} 1 & if\ \ I_{max} \le I \\ \frac{I - I_{min}}{I_{max} - I_{min}} & if\ \ I_{min} < I < I_{max} \\ 0 & if\ \ I \le I_{min} \end{cases} \tag{7}$$

2 RESULTS AND CONCLUSION

We implemented our method as a material in an RGB path tracer that combines BSDF-oriented sampling and explicit light sampling using next-event estimation. For the tonemap function (Eq. 5), we transformed $\hat{L}_P$ into XYZ color space and gamma corrected the Y value to obtain I.

We compared our method with extended implementations of cartoon shading for 3D objects [Lake et al. 2000] and color replacement for 2D images [Magdics et al. 2013]. We implemented three extensions A1, A2, and A3 using geometric specifications that map one-dimensional texture onto the angle between the normal and incident direction. The texture mapping of A1 computes $u = \max\{\mathbf{n} \cdot \boldsymbol{\omega_i},\ 0\}$ with sampled $\boldsymbol{\omega_i}$ in the direction toward the lights. A2 maps a texture based on direct illumination. It first computes direct diffuse component $D = f(\ \boldsymbol{\omega_i},\ \boldsymbol{\omega_o})L_i(\boldsymbol{\omega_i})(\mathbf{n} \cdot \boldsymbol{\omega_i})$with $\boldsymbol{\omega_i}$ sampled toward the lights. We then used Eq. 7 to compute the texture coordinate. A3 considers indirect illumination. The object color is the texture value at $u = \max\{\mathbf{n} \cdot \boldsymbol{\omega_i},\ 0\}$ with $\boldsymbol{\omega_i}$ sampled along the BSDF, and it is shaded by path-traced incoming radiance.

The image processing implementation B uses two inputs: a simple path-traced image and mask image that contains direct diffuse and interreflection components of the sphere. Each pixel in the mask region is replaced with the texture color.

For each implementation, we used a texture that returns a different hue to visualize the map from intensity to texture space (Figure 2 bottom).

Figure 2 shows a scene containing a sphere with ideal diffusion, a floor with a diffuse and glossy component, and two area lights that shade the sphere. For methods A1 and A2, a darker (lower) value appears in the lower part of the sphere because these methods do not consider the reflection from the floor to the sphere.In the green circled areas, there are red parts mapped from $u = 1$ because the method A1 and A3 depend only on the incoming direction whereas simple PT yields a different intensity. In addition, A3 presents a photorealistic shading by path traced illumination. However, the

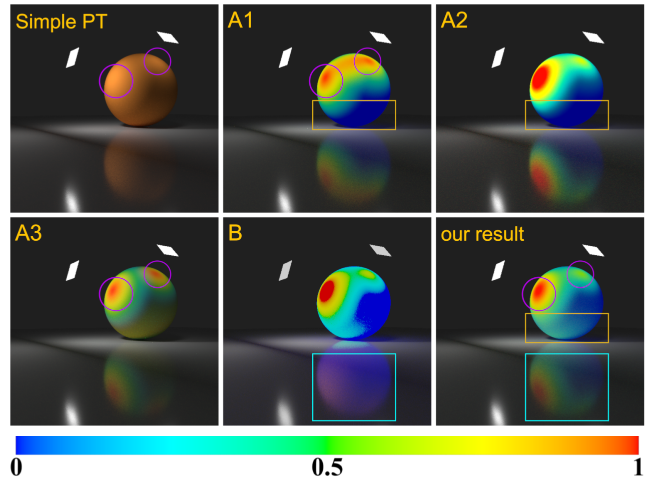

Figure 2: Comparison of global illumination methods. Effects of indirect incoming (orange) and outgoing (blue) light on the sphere are shown. Purple circles show the dependencies of the texture value on the lighting.

darkest shade is not blue. These problems occur because the texture coordinates are independently computed of the shading.

Method B yields colors that incorporate the reflection from the floor of the sphere. However, the floor does not correctly reflect the altered color of the sphere.

In the results of the proposed method, the color corresponds to the indirect light from the floor and the intensity of the shading. Furthermore, texture is incorporated in the reflection in the floor.

In this paper, we proposed the first computing method for non-photorealistic expression in a path tracing framework. In future, practical features for users will make it possible to illuminate fantastic and complex scenes.

REFERENCES

James T. Kajiya. 1986. The rendering equation. In Proceedings of the 13th annual conference on Computer graphics and interactive techniques (SIGGRAPH '86). Association for Computing Machinery, New York, NY, USA, 143–150. DOI:https://doi.org/10.1145/15922.15902

Adam Lake, Carl Marshall, Mark Harris, and Marc Blackstein. 2000. Stylized rendering techniques for scalable real-time 3D animation. In Proceedings of the 1st international symposium on Non-photorealistic animation and rendering (NPAR '00). Association for Computing Machinery, New York, NY, USA, 13–20. DOI:https://doi.org/10.1145/340916.340918

Milán Magdics, Catherine Sauvaget, Rubén J. García, and Mateu Sbert. 2013. Post-processing NPR effects for video games. In Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry (VRCAI '13). Association for Computing Machinery, New York, NY, USA, 147–156. DOI:https://doi.org/10.1145/2534329.2534348