VR Project: Adaptive Rendering

Zihao Ye
Department of Computer Science and Engineering
Shanghai Jiao Tong University

Introduction

Ray Tracing is a technique for rendering realistic scenes under global illumination with high computational cost. The main idea of ray tracing algorithm is to solve the rendering equation:

$$\begin{aligned} L_o(\mathbf{x}, \omega_o) &= \int_\Omega f_r(\mathbf{x}, \omega_i, \omega_o) L_i(\mathbf{x}, \omega_i)(\omega_i \cdot \mathbf{n}) d\omega_i \\ &+ L_e(\mathbf{x}, \omega_e) \end{aligned}$$

where L_o is the total spectral radiance, L_e is emitted spectral radiance, f is the reflectance distribution function.

Monte Carlo Path Tracing is one of the implementations of *Ray Tracing algorithm*Kajiya (1986). During recursion, rather than integrate over all the illuminance reaching a point at a time, it randomly selects a direction from here and keeps going, which avoids branches while computing the illuminance. By sampling many times, the expectation of illuminance will converge to the integration value. Algorithms based on sampling like MCPT inevitably introduces noises since one sample only takes a particular path instead of averaging illuminance come from all directions together. The quality of the images rendered by MCPT depends on the number of samplings. Empirically, thousands of samples will suffice to make a photo-realistic image with acceptable noise.

Related Work

To balance between computational cost and noise introduced by an inadequate amount of samplings, adaptive rendering algorithms have been proposed in recent years: Overbeck, Donner, and Ramamoorthi (2009), Moon et al. (2015), Rousselle, Knaus, and Zwicker (2012), Moon et al. (2016). The main idea of adaptive rendering is to locally fit the image rendered by MCPT algorithm by linear/nonlinear functions while disregarding the noise. However, since the image is not a highly continuous function in 2D space, the function we used to fit need to be flexible enough to deal with the discontinuity in some areas.

Proposed Work in Assigned Research Paper

Adaptive Polynomial Rendering (Moon et al. (2016) SIGGRAPH), is a paper that proposes an adaptive rendering architecture with flexibility to treat different areas adaptively. Mathematically, it's reasonable to fit image function locally with Taylor polynomial by taking 8D vector as parameter(2D position in image, corresponding 3D position in space with its 3D texture, 1D distance, and 1D visibility). By adopting different polynomial order in different areas with minimum loss, it obtains certain ability to deal with discontinuity.

Limitations of Proposed Work

Critical Review In this paper, the authors use Taylor Polynomial to locally fit the image function and use Least Squares Method to get best parameters. To select the best polynomial order k, the author proposed an alternative form of *Bias and Variance Expression* and *Multi-Stage Error Estimation*. They also proposed an algorithm called *Energy-Preserving Outlier Removal* to reduce spike noise while preserving total energy.

strengths Reasonable use of Taylor Polynomial and Least Squares Method, acceptable running time. The idea of *Adaptive Sampling* is cool and promising.

weakness Most of the techniques are proposed to select k, which does not fundamentally determines the quality of generated image. Compared to the complicated form of local functions, fine-grained features regarding 3D scenes are more useful in this task.

Proposed Research Work

Reconstruction Framework

The author use the following statistical model:

$$y(i) = \mu(i) + \epsilon(i)$$

where y is the generated image function, μ is the ground truth intensity, ϵ models MC error.

More specifically, they decompose the ground truth into two functions: $\mu(i) \equiv g(f_i) + p(i)$, the first function focuses on features related to 3D position while the second function focuses on pixel position.

Locally, they approximate the unknown μ function by using Taylor Polynomials:

$$\begin{aligned} \mu(i) &\approx \nabla g(f\_c)(f\_i - f\_c)^T + p(c) \\ &+ \sum_{1 \le a \le k} \frac{\nabla^a p(c)}{a!}((i-c)^a)^T \end{aligned}$$

The coefficients could be optimized by minimizing the following function (using least-square method):

$$\sum_{i \in \Omega_c} \left(y(i) - \alpha(f_i - f_c)^T - \beta_0 - \sum_{1 \le a \le k} \beta_a((i-c)^a) \right)^2 K_h(i)$$

After reconstruction, the output at pixel i would be:

$$\hat{y}(i) = \sum_{j \in \Omega_i} K_h^j(i)\hat{y}_k^j(i) / \sum_{j \in \Omega_i} K_h^j(i)$$

Adaptive Polynomial Reconstruction

To locally optimize the polynomial order k, we have to optimize the function:

$$\xi_c(k) \equiv \frac{1}{\sum K_h(i)} \sum_{i \in \Omega_c} K_h(i)(\hat{y}_k(i) - \mu(i))^2$$

where μ is intractable, thus we need to find alternatives to optimize.

Bias and Variance Expression According to the definition of variance: $\sigma(X) = EX^2 - (EX)^2$, we derive:

$$E(\hat{y}_k(i) - \mu(i))^2 = bias^2(\hat{y}_k(i)) + \sigma^2(\hat{y}_k(i))$$

where the bias and variance could be computed by the hat matrix:

$$E(\hat{y}_k(i)) \approx \sum_{j \in \Omega_c} H_{ij}(k)\mu(j) - \mu(i)$$

$$\sigma^2(\hat{y}_k(i)) \approx \sum_{j \in \Omega^c} (H_{ij}(k))^2 \sigma^2(y(j))$$

then the optimal k could be chosen via

$$k_{opt} = \arg\min_k \sum_{i \in \Omega_c} K_h(i)((E(\hat{y}_k(i) - \mu(i)))^2 + \sigma^2(\hat{y}_k(i)))$$

in which the normalization term has been dropped out.

Multi-Stage Error Estimation Directly use y and sample variance s to replace mu and σ will make them too noisy resulting in a noisy selection of polynomial order.

The author proposed a multi-stage way to deal with this problem,

Energy-Preserving Outlier Removal When MC renderer renders a scene includes glossy materials, the outlier (spike noise) pixels exhibit an excessive intensity. The outliers should be removed as a pre-process, making the optimization process for reconstruction more robustly.

The side effect of outlier removal is the energy loss in the reconstructed image. The author proposed an energy-preserving outlier removal technique.

If the difference between the intensity of the center pixel and the average intensity of all neighboring pixels is three times higher than the computed standard deviation, it is replaced with a pixel that has the median intensity and its variance.

Energy is preserved by

$$\hat{y}(i) = \hat{y}(i) + \rho\_o\hat{y}(i)$$

where ρ is computed via=

$$e_o = \sum \_i \in \Omega\_o\rho\_o\hat{y}(i)$$

Adaptive Polynomial Sampling

At last, the author proposed Adaptive Sampling method which allocates ray samples adaptively in each area by maintaining a weight function K.

Methodology

We could regard the procedure of reconstructing image as regressing given training data with noise. Similar to the idea of Cubic spline interpolation, the paper aims to reconstruct the image by locally fitting the image using Taylor Polynomial(order less or equal to 3) on variables considering both 2D and 3D properties. Several advanced approximation methods are adopted to derive unbiased estimation. The idea of Adaptive Sampling method is similar to that of *Active Learning*, in which the learner has the ability or need to influence or select its own training data Cohn, Ghahramani, and Jordan (1996).

Implementation

My experiment is based on my MCPT ray tracer. Since acceleration method like SAH-KD tree has been adopted, it usually takes about 1300 secs to render most scenes in default setting(32 samples per pixel) on a 4-core CPU machine.

The reconstruction part of my implementation uses the adaptive polynomial rendering algorithm. To cope with complex matrix operations efficiently in C++, I use the highly-optimized linear-algebra library armadillo.

To save time, *Adaptive Sampling* and *Outlier Removal* mentioned in the paper has not been performed yet.

Different window-size selection will result in totally different performance. In my current setting, the window size is 15×15, and the central point is selected if and only if the row number and column number are both multiples of 15.

Results

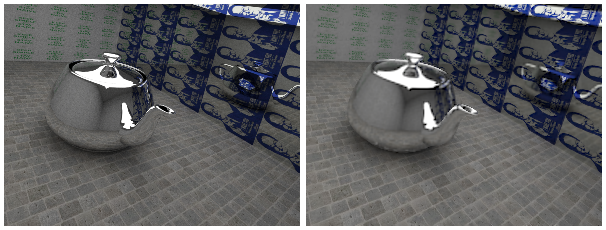
Teapot with diffusive face before/after reconstruction:

Diffusive

(a) Before reconstruction (b) After reconstruction

Specular

Teapot with specular face before/after reconstruction.



(a) Before reconstruction (b) After reconstruction

Due to the lack of ground truth image(which takes thousands of samples), we are unable to compare the statistical criterion like *rmse* between the images before and after reconstruction.

Conclusions & Future Directions

The experiment result demonstrates that Adaptive Polynomial Rendering is good at reconstructing faces that are diffusive with a complex texture. However, it struggles when dealing with the specular or reflexive faces.

The details in reflexive or specular faces could not be precisely captured since they could not be featured only by the 8D vector(mentioned above) that focuses on the ray's first encountered face. To improve its performance, information considering the ray's second encountered face could by involved.

References

Cohn, D. A.; Ghahramani, Z.; and Jordan, M. I. 1996. Active learning with statistical models. *Journal of artificial intelligence research* 4(1):129–145.

Kajiya, J. T. 1986. The rendering equation. *SIGGRAPH Comput. Graph.* 20(4):143–150.

Moon, B.; Iglesias-Guitian, J. A.; Yoon, S.-E.; and Mitchell, K. 2015. Adaptive rendering with linear predictions. *ACM Transactions on Graphics (TOG)* 34(4):121.

Moon, B.; McDonagh, S.; Mitchell, K.; and Gross, M. 2016. Adaptive polynomial rendering. *ACM Transactions on Graphics (TOG)* 35(4):40.

Overbeck, R. S.; Donner, C.; and Ramamoorthi, R. 2009. Adaptive wavelet rendering. *ACM Trans. Graph.* 28(5):140–1.

Rousselle, F.; Knaus, C.; and Zwicker, M. 2012. Adaptive rendering with non-local means filtering. *ACM Transactions on Graphics (TOG)* 31(6):195.