Image Space Adaptive Polynomial Rendering

Jonathan Bosson, jonbo665@student.liu.se
TNCG13 Linköping University

Tuesday 20th December, 2016

Abstract

This report discusses a new novel approach by Moon et al. [2] to make the raycasting rendering algorithm more effective by reducing noise and computation cost. This is done by introducing an adaptive sampling rate and a unique reconstruction method through local Taylor polynomial functions. Since no analytical solution can be used, the image reconstruction uses estimations and statistical analysis to find the minimal error around bias, variance and polynomial order. An energy-preserving outlier removal technique is also introduced in order to address glossy surfaces, a wellknown issue the usual Monte-Carlo method handles poorly. Moon et al. [2] further demonstrates that this approach outmatches state-of-the-art methods in both efficiency and image quality.

1 Introduction

The Monte-Carlo raycasting algorithm (MC) has been a wellknown and potent method to render photorealistic 3D scenery since 1986 when Kajiya proposed the rendering equation [1]. The method is popular since it is capable to render any desired effect. Its biggest downfall however is the computation cost, as the method can require more than 10000 casted rays per pixel to render a converged image. This is especially apparent in high dimensional applications to simulate global illumination.

There are two distinct approaches to solve the dimensionality issue: high-dimensional adaptive methods and image space adaptive methods, both of which are built upon dynamic sampling in place of a uniform one. Image space adaptive methods, such as the one proposed by Overbeck et al. [3], finds the variance of the MC in 2D image space and then minimizes the error noise with the help of an image filter and estimated errors. Even though different filters can be used, the common behaviour is to minimize the numerical error by controlling filtering bandwidths at each pixel. The method presented by Overbeck et al. has since quickly gained attention due to its ease in implementation and generality compared to its high-dimensional adaptive counterpart.

Moon et al. [2] extends this method by changing the polynomial function locally in order to control the optimal filtering bandwithds per pixel, which minimizes the mean squared error (MSE). This can essentially be seen as a variance vs. bias tradeoff, under- or over-blurring the image. Using this approach instead of the previously mentioned bandwidth adaptation filtering step it surpasses all other currently available state-of-the-art rendering methods.

2 Adaptive Polynomial Reconstruction

The dynamic sampling of rays per pixel utilizes the fact that rendered images have a heterogeneous noise to guide areas with low sampling density. The adaptive reconstruction is to control the smoothing in an area by examining MC noise to preserve high-frequency edges.

The reconstruction can be seen as an optimization problem where the goal is to compute the true intensity $\mu(i)$ of a pixel. With an input image function $y(i)$, where i is a position in a 2D image, consider the following statistical model:

$$y(i) = \mu(i) + \epsilon(i) \tag{1}$$

As mentioned $\mu(i)$ is the true intensity of the pixel, a value that can only be achieved with an analytical solution or an infinite amount of samples. $\epsilon(i)$ in turns is the MC noise seen as variance.

Since its a statistical model can $\mu(i)$ be broken down into two separate parts, $\mu(i) \equiv g(f_i) + p(i)$. $p(i)$ is a 2D function that takes the pixel position as input and $g(f_i)$ is a linear function that takes a high-dimensional vector f_i containing information such as normals (3D), textures (3D), depths (1D), and visibility (1D). All of which can handily be calculated at the intersection points during rendering. The use of this decomposition is that the unknown $\mu(i)$ can be approximated in glossy areas by the residual $\mu(i) - g(f_i)$ using $p(i)$ and in other areas using the correlation between rendering-specific features f_i and $\mu(i)$. Moon et al. [2] locally approximates $\mu(i)$ using Taylor polynomials:

$$\mu(i) \approx \nabla g(f_c)(f_i - f_c)^T + p(c) + \textstyle\sum_{1 \leq a \leq k} \frac{\nabla^a p(c)}{a!}((i-c)^a)^T \tag{2}$$

∇ is the notation for differentation and k the Taylor polynomial order that is used. After a few interstages, Moon et al. [2] arrives at equation 3 to describe the least-squares optimization estimated within a local window Ω_c.

$$\hat{y}(i) = \sum_{j \in \Omega_i} K_h^j(i)\hat{y}_k^j(i) / \sum_{j \in \Omega_i} K_h^j(i) \tag{3}$$

K_h^j is a gaussian kernel function that weights pixel i over the filtering bandwidth h, which controls the bias-variance tradeoff of the reconstruction. $\hat{y}_k^j(i)$ is the reconstruction result of pixel i. Both of these are computed from the center pixel j. This chunk wise reconstruction is particularly favorable as Taylor polynomials can accomodate to large regions Ω_c well by simply increasing the polynomial order k.

Since the reconstruction is defined chunk wise is the same required for the optimization goal. Equation 4 defines the reconstruction error of a polynomial at center pixel c as the L2 error.

$$\epsilon_c(k) \equiv \frac{1}{\sum_{i \in \Omega_c} K_h(i)} \sum_{i \in \Omega_c} K_h(i)(\hat{y}_k(i) - \mu(i))^2 \tag{4}$$

The optimal pylonomial order k_{opt} giving the minimal error $\epsilon_c(k_{opt})$ can thus be computed, although not directly as the L2 error uses the unknown $\mu(i)$ term. Instead, the recontruction bias-variance is mathematically expressed in section 2.1, an estimation process for the error terms presented in section 2.2 and removal of outliers while preserving energy explained in section 2.3.

2.1 Bias and Variance Expression

To estimate the actual error $(\hat{y}_k(i) - \mu(i))^2$ in equation 4, it is first broken down into bias and variance segments by taking the mathematical expectation E. The bias part is approximated as:

$$E(\hat{y}_k(i) - \mu(i)) \approx \sum_{i \in \Omega_c} (H_{ij}(k))\mu(j) - \mu(i) \quad (5)$$

where $H_{ij}(k)$ is the element at position (i, j) in a matrix that determines a projection from the input values y to projected values $\hat{y}_k$. This term is further explained by Moon et al. in a previous publication [4]. After the bias term has been applied is the input image seen as an unbiased rendering result. The variance can be defined with $\sigma^2(y(j))$ as the variance with pixel intensity $y(j)$ as:

$$\sigma^2(\hat{y}_k(i)) \approx \sum_{i \in \Omega_c} (H_{ij}(k))^2 \sigma^2(y(j)) \quad (6)$$

k_{opt} can thus be defined as equation 7 using the new decomposition, however it still contains the unknown terms $\mu(i)$ in the bias and σ^2in the variance computations.

$$k_{opt} = \underset{k}{argmin} \sum_{i \in \Omega_c} K_h(i)((E(\hat{y}_k(i) - \mu(i)))^2 + \sigma^2(\hat{y}_k(i))) \quad (7)$$

2.2 Multi-Stage Error Estimation

To solve this equation 7 is an estimation process used to estimate the unknown terms. One approach to solve equation 5 and 6 is to use the MC input intensities $y(i)$ and $y(j)$ and sample variance $s^2(y(j))$, however this method leads to the polynomial order selection to be ambiguous. Instead multi-state error estimation is proposed that iteratively guesses $\mu(i)$, $\mu(j)$ and $\sigma^2(y(j))$. The bias part is computed as follows:

$$E_t(\hat{y}_k(i) - \mu(i)) \approx \sum_{j \in \Omega_c} H_{ij}(k)\hat{y}_{t-1}(j) - \hat{y}_t(i) \quad (8)$$

In the first iteration $t = 1$, $\hat{y}_0(i)$ and $\hat{y}_0(j)$ is set as the previously suggested MC input values $y(i)$ and $y(j)$. In the second, and future, iterations the result from the reconstruction $\hat{y}_1(i)$ and $\hat{y}_1(j)$ are used in place of the unknown $\mu(i)$ and $\mu(j)$. This gives an error much lower than the MC input intensities and hence also the estimation errors $|\mu(j) - \hat{y}_1(j)|$ and $|\mu(i) - \hat{y}_1(i)|$ is reduced for each repetition. The variance part is computed in a very similar way, shown in equation 9.

$$\sigma^2(\hat{y}_k(i)) \approx \sum_{j \in \Omega_c} (H_{ij}(k))^2 \hat{\sigma}^2_{t-1}(y(j)) \quad (9)$$

For the first iteration the sample variance $s^2(y(j))$ is used and later iterations use the guessed variance $\sigma^2_{t-1}(y(j)$ from previous iteration.

Moon et al. [2] shows that the error, and thus noise in the rendered image, is reduced with each iteration after the first one. However, since each iteration requires a significant recalculation for the optimal polynomial order for the variance term and the improvement between two and three iterations is negligible, is two recommended for a balance between efficiency and quality.

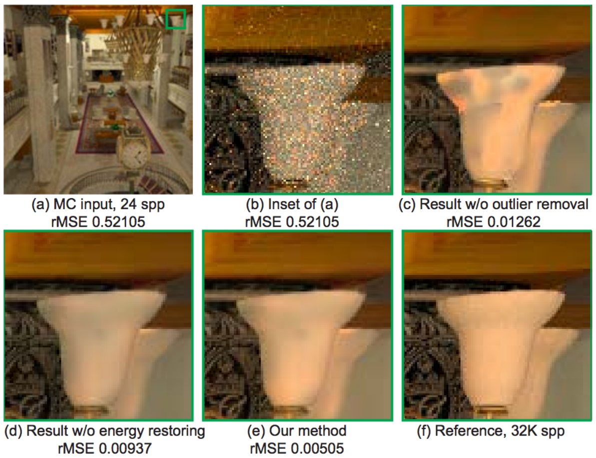

Figure 1: Image taken from Moon et al. [2] displaying the energy-preservation method. (a) is a uniformly generated image and (b) its inset. The result without the outlier removal method (c) exhibit artifacts due to the difficulty to distribute the excessive energy. (d) shows removes outliers but is not energy preserving carry a visible energy loss. The presented energy-preserving outlier removal technique (e) resolves both problems (c) and (d) had.

2.3 Energy-preserving Outlier Removal

Glossy surfaces causing outliers with exaggerated intensities in MC renderers is a wellknown issue. However, if the outliers are removed before rendering it causes noticable energy loss in the resulting image.

Moon et al. [2] proposes an extension to the removal of outliers in the pre-process with an energy restoration method in the post-process. During the removal of the outliers the energy loss, ie. difference e_o between the outliers intensity and the median intensity in that area (that the outlier pixel o is replaced with), is saved. After the reconstruction can the lost energy be

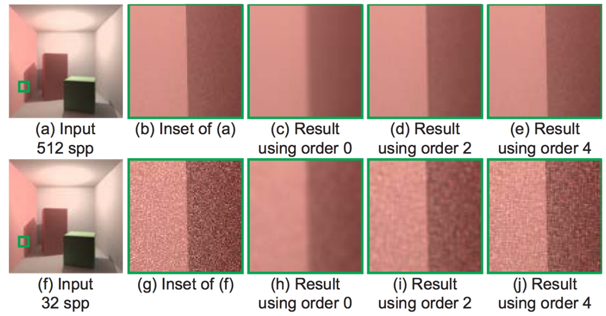

Figure 2: Image taken from Moon et al. [2] showing results of different polynomials orders under two sample counts. Polynomials of higher order ($k \geq 4$) are more noisy but preserve the discontinuous edge better. This tradeoff drives the discussed adaptive control between the bias and variance of the reconstruction result.

restored to the output $\hat{y}(i)$ but over the area Ω_o around outlier pixel o as:

$$\hat{y}(i) = \hat{y}(i) + \rho_o \hat{y}(i) \qquad (10)$$

where $\rho_o > 0$ is a compensation parameter calculated from equation 11 and responsible for controlling the distribution of the energy loss from pixel o.

$$e_o = \sum_{i \in \Omega_o} \rho_o \hat{y}(i) \qquad (11)$$

3 Adaptive Sampling Rate

To send in a dynamic set of rays per pixel can an iterative adaptive method be used that was proposed by Overbeck et al. [3], where the goal is to add additional rays on pixels with high errors. In this method, a uniform pass is first computed with a very low number of samples (4 or 8). The reconstruction error $\hat{e}_c(\hat{k}_{opt})$ using the presented bias and variance estimations. Since the the reconstruction equation (3) uses a local average with the K-function, should the error $\hat{e}_i$ of the output $\hat{y}(i)$ at pixel i be combined with the kernel funcion K as well. With the estimated error of each pixel known, additional samples are added on pixels with the maximum MSE reduction rate $\Delta(\hat{e}_i)$ until a defined sample budget has been met.

4 Discussion and Future Work

Moon et al. [2] takes one step further from previous image space adaptive methods by adding taylor polynomials in the reconstruction and estimation stages. It also introduces a blockwise optimization process which improves the optimal order selection to become more robust. This effectively means its more capable of properly denoising very noisy input images, mostly visible in glossy scenes with high numerical errors where other methods would over-blurr the image.

However, in higher dimensions, such as including time and second bounces, could adaptively controlling other spaces be important. Image space methods, this one included, uses a random sampling which can cause issues in for example moving highlights. The estimation process is limited to the quality of the input image. Since even extremely highly sampled images can render noisy edges is it a significant challenge to reconstruct the input image appropriately when it does not come with enough information. Moon et al. [2] states that future work revolves around increasing the sampling dimension as well as minimizing accounting for the dimensionality issue.

References

[1] Kajiya, J. T. 1986. **The rendering equation.** *In ACM SIGGRAPH, 86, 143150.*

[2] Moon, B., McDonagh, S., Mitchell, K. and Gross, M. 2016. **Adaptive polynomial rendering.** *ACM Transactions on Graphics (TOG) 35.4 40.*

[3] Overbeck, R. S., Donner, C., and Ramamoorthi, R. 2009. **Adaptive wavelet rendering.** *ACM Trans. Graph. 28, 5, 140:1140:12.*

[4] Moon, B., Iglesis-Guitian, J. A., Yoon, S.-E., and Mitchell, K. 2015. **Adaptive rendering with linear predictions.** *ACM Trans. Graph. 34, 4, 121:1121:11.*