The Title of Your Paper Goes Here

⟨ **TODO** pick title ⟩

Abstract

When rendering using Monte Carlo methods, either a large amount of samples are necessary or noise will be present in the image. A lot of methods have already tried to tackle this problem including adaptive sampling, reconstruction techniques and advanced image filtering techniques. In this paper I will recapitulate the work I have done for my thesis so far. The focus will be on the RPF algorithm called random parameter filtering (RPF) as proposed in [Sen and Darabi 2011]. As this algorithm is based on a cross bilateral filter, this concept will be explained and discussed as well. To conclude, some other methods that handle the noise caused by Monte Carlo rendering at low sampling rates were investigated as well. ⟨ **TODO** verwijs naar meerdere methodes die dit hebben proberen aan te pakken? ⟩ [Dutré et al. 2006]

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing; ⟨ **TODO** CRcatlist ⟩

Keywords: monte carlo rendering, filter, distribution effects, global illumination

1 Introduction

Computer Graphics has evolved from 2D to almost realistic 3D during the last decennia. To reach this realism in graphics, Global Illumination algorithms became very important, these incorporate the indirect light caused by the surroundings. These algorithms require large numbers of calculations and thus a large rendering time, especially when results without noise are expected. To lower this rendering time, Monte Carlo methods are often used because they can be used to evaluate multidimensional functions in an unbiased way. Complex and photorealistic images can be created using these Monte Carlo methods. In the case of (Monte Carlo) rendering, the multidimensional integrals that needs to be solved at every pixel of the image are: integration of the radiance over the aperture of the camera (u,v), over the time the shutter is open ($t_0 \to t_1$) and over the area of the pixel (($(i-1/2, j-1/2) \to (i+1/2, j+1/2)$).

$$I(i,j) = \int_{i-\frac{1}{2}}^{i+\frac{1}{2}} \int_{j-\frac{1}{2}}^{j+\frac{1}{2}} \cdots \int_{-1}^{1} \int_{-1}^{1} \int_{t_0}^{t_1} f(x, y, \cdots, u, v, t)\, dt\, dv\, du\, dy\, dx.$$

As Monte Carlo methods take random samples to evaluate a function, a lot of samples are usually necessary to evaluate the function precisely and thus a lot of noise will be present when using a low number of samples. One obvious solution to this problem is to use more samples, but as the rendering time increases dramatically with the amount of samples this is not always an option. Adaptive sampling algorithms can then be used to distribute these (sometimes large) numbers of samples in the best possible way across the image. Reconstruction techniques can also be used to suppress the number of samples, as these try to use the available samples as much as possible, across pixels and even across frames. Another technique that can be used is the focus of this paper and is an advanced image filter, it uses not only the color computed at each sample of the image, but also other parameters that are available during the rendering process, to filter the image.

In the related work section I will describe all the different related papers I researched before deciding to research the RPF algorithm in more detail. The technical sections describe the concept of bilateral filter after which the RPF algorithm is explained in detail. To conclude the results describe my current results followed by a discussion of these results and a concise conclusion.

2 Related Work

⟨ **TODO** plak de verschillende paper uitleggen aan elkaar, sorteer volgens jaar ??? ⟩ More information about Monte Carlo methods can be found here [Kalos and Whitlock 2009], while more information about advanced rendering can be found here [Dutré et al. 2006] and here [Pharr and Humphreys 2010]. Some recent work that try to tackle the problem of noise at low sampling rates when using Monte Carlo rendering are discussed below.

Compressive estimation for signal integration in rendering: While Monte Carlo methods are used to evaluate functions, other methods like the one explained in [Sen and Darabi 2010] can also be used to compute the integral of an unknown function. This is possible because of the theory of compressed sensing, which allows to reconstruct a signal from a few linear samples if it is sparse in a transform domain. This method can also be used to compute computer graphics distribution effects like anti-aliasing and motion blur. The advantage of this function over Monte Carlo methods is that is needs only a few samples to estimate the function.

Temporal Light Field Reconstruction for Rendering Distribution Effects:
A general reconstruction technique that tries to maximize the image quality based on a given set of samples is given in [Lehtinen et al. 2011]. This technique exploits the dependencies in the temporal light field and allows efficient reuse of samples between different pixels. The effective sampling rate is multiplied by a large factor when using this technique. The paper makes the following four contributions:

1. ***The reconstruction algorithm***
   The input of the algorithm contains a set of samples in the form:
   $$s = \{(x, y, u, v, t) \to (z, v, L)\} \tag{1}$$
   With xy the screen coordinates, uv the lens coordinates, t the time, z the depth, v the 3D motion vector and L the radiance associated with the input sample.
   The reconstruction algorithm then goes as follows:
   (a) The screen coordinates of the input samples are reprojected to the (u,v,t) coordinates of the reconstruction location. Samples that are too far away from the reconstruction location in xy are discarded.
   (b) The returned clusters of samples are grouped into apparent surfaces.
   (c) If multiple apparent surfaces are found, they are sorted front-to-back. Afterwards, it is determined which one covers the reconstruction location.

   (d) The output color is computed by filtering the samples that belong to the covering surface using a circular tent filter.

2. ***Determining visibility consistency***
   By using the key observation that the relative ordering of the sreen positions of samples from a non-overlapping surface never changes under reprojections, they derive a formal criterion named SAMESURFACE to detect when a set of reprojected samples should be filtered together.

3. ***Resolve visibility without explicit surface reconstruction***
   Which surface is visible at the reconstruction location is determined. The challenge is to distinguish between small holes in the geometry and apparent holes caused by stochastic sampling. To do this a radius R is precomputed, R is the radius of the largest empty circle that is expected to be visible on the xy plane after reprojection. Holes that are smaller than R should be filled because it is beyond the resolution of the input sampling. The visibility is then determined using the following rule. A reconstruction location is covered if it is possible to find three reprojected input samples that form a triangle that covers the reconstruction location and fits inside a circle of radius R.

4. ***Hierarchical query structure***
   The reconstruction algorithm needs to retrieve the input samples that reproject to the vicinity of reconstruction locations (x, y), given (u, v, t) quickly. The input samples are organized into a bounding volume hierarchy, where the extents of the nodes xy are parameterized using u, v and t. When executing a query, the parameterized bounding volume is used to test if the reconstruction location is inside the bounds.

Reconstruction the Indirect Light Field for Global Illumination:
The algorithm that is described in [Lehtinen et al. 2012] is similar to the former algorithm and is published by almost the same authors. The paper also describes a general reconstruction technique that exploits anisotropu in the light field and permits efficient reuse of input samples between pixels or world-space locations, multiplying the effective sampling rate by a large factor. The main difference between the two algorithms is that this algorithm tries to reconstruct the indirect light field at scene point instead of at points on the lens like the algorithm in [Lehtinen et al. 2011] does. That is also why reconstructing an image using this algorithm takes three to four times longer than the former algorithm. The former algorithm uses a 2D hierarchy, but this algorithm needs to reconstruct the incident light field at arbitrary points in the scene, so a true 3D algorithm is needed.

Axis-Aligned Filtering for Interactive Sampled Soft Shadows:
The post-processing step needed in Light Field Reconstruction techniques is very expensive. Other algorithms like the one proposed in [Mehta et al. 2012] has a very simple filtering step by using axis-aligned filters. Because of this extremely simple step, this algorithm can be integrated in real-time raytracers. Adaptive filtering is used because the parameters of the used filters are adjusted depending on the input samples. This algorithm is basically an image filter for noise that is fixed on the soft shadows effect.

Adaptive Rendering with Non-Local Means Filtering:
The following algorithm found in [Rousselle et al. 2012] describes an adaptive sampling algorithm for Monte Carlo rendering. An adaptive sampling algorithm tries to determine the optimal sample distribution across the image. This can be done by allocating more samples to regions with difficult light effects. The first step of the algorithm distributes a given budget of samples over the image after which the image is filtered in the second step of the algorithm with a variant of the NL-means filter which is a generalisation of the bilateral filter. This filter considers distances between pairs of pixel values to compute filter weights, the term non-local is caused by the fact that the set of samples that contribute to one output pixel can come from a large region in the input image. In the third step of the algorithm the remaining error in the filtered image is estimated to drive the adaptive sampling in the next iteration step.

Multidimensional Adaptive Sampling and Reconstruction for Ray Tracing:
A combination of adaptive sampling and reconstruction is proposed in [Hachisuka et al. 2008]. This paper introduces a new sampling strategy for ray tracing. The samples on which the strategy operates are generated from the rendering equation directly and are thus not generated through Monte Carlo sampling. Additionally this algorithm uses all previously generated samples to generate a new sample. After sampling this high-dimensional function, a reconstruction is made by integrating the function over all but the image dimensions.

SURE-based Optimization for Adaptive Sampling and Reconstruction:
A similar goal is aimed for by [Li et al. 2012] allthough samples are distributed over the image based on an estimator of the Mean Squared Error (MSE) of the image. The estimator that is used is called Stein's Unbiased Risk Estimator (SURE), it allows to estimate the error of an estimator without knowing the true value that is estimated. Another difference is that a filterbank is used instead of a single filter. The usage of a filterbank makes this algorithm different from algorithms like RPF because any kind of filter can be added to this filterbank. The authors of the paper have experimented with isotropic Gaussian, cross bilateral and a modified non-local means filter. A small number of initial samples are rendered first, followed by filtering each pixel with all filters in the filterbank. Next the filtered color with the lowest SURE error is chosen for each pixel and is used in the reconstruction. When more samples are available, these are used for the pixels with the largest SURE errors after which the process goes back to filtering each pixel with all the filters from the filterbank.

Adaptive sampling and reconstruction using greedy error minimization:
Another algorithm that does not fix which filter is used for different pixels is described in [Rousselle et al. 2011]. As this algorithm is greedy, it minimizes the function at each stage hoping to reach a global optimum. Given a current sample distribution, a filter that minimizes the pixel error is selected from a discrete set of filters that can be different for each pixel. Given the filter selection, addition samples are distributed to further reduce the MSE. Because the MSE cannot be calculated exactly the change in MSE is calculated instead. This whole process can be repeated until any chosen termination criterion is met.

High-Quality Spatio-Temporal Rendering using Semi-Analytical Visibility:
A visibility technique with the only purpose to render motion blur with per-pixel anti-aliasing is described in [Gribel et al. 2011]. A number of line samples are used over a rectangular group of pixels that form a 2D spatio-temporal visibility problem together with the time dimension. This problem needs to be solved per line sample. Each group of pixels in the image is rendered separately. For each

group a Bounding Volume Hierarchy is used to receive only the geometry that overlaps with the tile. Furthermore each line sample is processed one at a time by calculating what triangles intersect with the sample followed by resolving the depth visibility. When all the triangles have been processed, the final visibility is resolved and for each pixel the contribution of the line samples overlapping the pixel is accumulated to the color of that pixel.

A Theory of Monte Carlo Visibility Sampling: [Ramamoorthi et al. 2012] tries to lower the amount of noise introduced by Monte Carlo sampling of the visibility term in the rendering equation. By analysing the effectiveness of different non-adaptive Monte Carlo sampling patterns for rendering soft shadows, they search for the pattern with the lowest expected variance for a certain visibility function. These results can lead to a reduction in the needed number of shadow samples without losing precision by using the best sampling pattern.

3 Bilateral Filter

The Bilateral filter is introduced in [Tomasi and Manduchi 1998], I have used some extra information from [Devarajan and Nyikal 2013] as well.
As the name already suggests, a bilateral filter filters an image using two different inputs. To understand the working of a bilateral filter, the concept of a Gaussian filter is explained first.

3.1 Gaussian filter

As an image has two dimensions, the 2D version of the gaussian filter will be explained, in fact this is simply the product of two one dimensional gaussians, one per direction. The equation of a 2D Gaussian filter is defined as:

$$g(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad (2)$$

with x and y being the distance from the origin and σ being the standard deviation of the Gaussian distribution. The shape of a Gaussian filter being a 2D bell shaped function.
When an image is filtered using a Gaussian filter, this will lead to blurred images as the values of each pixel will fade into the values of the surrounding pixels. ⟨ **TODO** Add a Gaussian filtered image!! ⟩

3.2 Bilateral filter as two Gaussian filters

A bilateral filter can be explained as being two Gaussian filter that filter respectively in the spatial (the domain filter D) and in the intensity domain (the range filter R).

$$\begin{aligned} D(x,y) &= e^{-\frac{x^2+y^2}{2\sigma^2}} \\ R(p_i,p_j) &= e^{-\frac{f(p_i)^2+f(p_j)^2}{2\sigma^2}} \end{aligned} \qquad (3)$$

The domain filter filters using the difference x and y between the positions of the two pixels and the range filter uses the intensity value (obtained by the image function f) of both pixels.

We can now write down the equation to compute final value of a pixel filtered using a bilateral filter.

$$\begin{aligned} b(p_i) &= k^{-1}\sum_{m=0}^{N-1} f(p_m)D(x,y)R(p_i,p_m) \\ \text{with} & \\ k &= \sum_{m=0}^{N-1} D(x,y)R(p_i,p_m) \end{aligned} \qquad (4)$$

x and y are again the difference in the positions of the pixels p_i and p_m and N is the size of the neighboorhood taken into account for each pixel.

4 Random Parameter Filtering algorithm

⟨ **TODO** cross bilateral filter explanation ⟩ ⟨ **TODO** Mutual information ⟩ ⟨ **TODO** different parts of algorithm ⟩ blabla

5 Results

⟨ **TODO** vergeet niet PBRT2 te vermelden ⟩

6 Discussion

⟨ **TODO** stel resultaten voor en beschrijf en interpreteer ze, welke zijn de sterke-zwakke punten, welke problemen zijn nog niet helemaal opgelost, eventueel vergelijkingen met gelijkaardige systemen, enz. ⟩

7 Conclusion

⟨ **TODO** typische een zeer korte summary + conclusie van je werk ⟩

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Acknowledgements

⟨ **TODO** KULEUVEn en de makers van RPF enzo.. ⟩

References

DEVARAJAN, H., AND NYIKAL, H., 2013. Image scaling and bilateral filtering. `http://scien.stanford.edu/pages/labsite/2006/psych221/projects/06/imagescaling/index2.html`.

[Online, accessed 31 jan 2013].

DUTRÉ, P., BALA, K., AND BEKAERT, P. 2006. *Advanced global illumination*. Ak Peters Series. AK Peters.

GRIBEL, C. J., BARRINGER, R., AND AKENINE-MÖLLER, T. 2011. High-Quality Spatio-Temporal Rendering using Semi-Analytical Visibility. *ACM Transactions on Graphics, 30* (August), 54:1–54:11.

HACHISUKA, T., JAROSZ, W., WEISTROFFER, R. P., DALE, K., HUMPHREYS, G., ZWICKER, M., AND JENSEN, H. W. 2008. Multidimensional adaptive sampling and reconstruction for ray tracing. *ACM Trans. Graph. 27*, 3 (Aug.), 33:1–33:10.

KALOS, M., AND WHITLOCK, P. 2009. *Monte Carlo Methods*. Wiley.

LEHTINEN, J., AILA, T., CHEN, J., LAINE, S., AND DURAND, F. 2011. Temporal light field reconstruction for rendering distribution effects. *ACM Trans. Graph. 30*, 4.

LEHTINEN, J., AILA, T., LAINE, S., AND DURAND, F. 2012. Reconstructing the indirect light field for global illumination. *ACM Trans. Graph. 31*, 4 (July), 51:1–51:10.

LI, T.-M., WU, Y.-T., AND CHUANG, Y.-Y. 2012. Sure-based optimization for adaptive sampling and reconstruction. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2012) 31*, 6 (November), 186:1–186:9.

MEHTA, S., WANG, B., AND RAMAMOORTHI, R. 2012. Axis-aligned filtering for interactive sampled soft shadows. *ACM Trans. Graph. 31*, 6 (Nov), 163:1–163:10.

PHARR, M., AND HUMPHREYS, G. 2010. *Physically Based Rendering: From Theory to Implementation*. The Morgan Kaufmann series in interactive 3D technology. Elsevier Science.

RAMAMOORTHI, R., ANDERSON, J., MEYER, M., AND NOWROUZEZAHRAI, D. 2012. A theory of monte carlo visibility sampling. *ACM Transactions on Graphics*.

ROUSSELLE, F., KNAUS, C., AND ZWICKER, M. 2011. Adaptive sampling and reconstruction using greedy error minimization. *ACM Trans. Graph. 30*, 6 (Dec.), 159:1–159:12.

ROUSSELLE, F., KNAUS, C., AND ZWICKER, M. 2012. Adaptive rendering with non-local means filtering. *ACM Trans. Graph. 31*, 6 (Nov.), 195:1–195:11.

SEN, P., AND DARABI, S. 2010. Compressive estimation for signal integration in rendering. *Computer Graphics Forum 29*, 4.

SEN, P., AND DARABI, S. 2011. On Filtering the Noise from the Random Parameters in Monte Carlo Rendering. *ACM Transactions on Graphics (TOG) to appear*.

TOMASI, C., AND MANDUCHI, R. 1998. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, 839 –846.