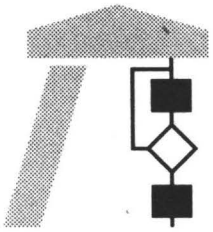
Interner Bericht

Hierarchical Monte Carlo Image Synthesis

Alexander Keller

298/99



FACHBEREICH
INFORMATIK

UNIVERSITÄT
KAISERSLAUTERN

Postfach 3049 · D-67653 Kaiserslautern

Hierarchical Monte Carlo Image Synthesis

Alexander Keller

298/99

Universität Kaiserslautern
AG Numerische Algorithmen
Postfach 30 49
67653 Kaiserslautern
Germany

Januar 1999

Herausgeber: AG Numerische Algorithmen
Leiter: Professor Dr. S. Heinrich

Hierarchical Monte Carlo Image Synthesis

Alexander Keller*

Universität Kaiserslautern

Abstract

A fundamental variance reduction technique for Monte Carlo integration in the framework of integro-approximation problems is presented. Using the method of dependent tests a successive hierarchical function approximation algorithm is developed, which captures discontinuities and exploits smoothness in the target function. The general mathematical scheme and its highly efficient implementation are illustrated for image generation by ray tracing, yielding new and much faster image synthesis algorithms.

CR Categories: I.3.3 [Computer Graphics]: Picture/ Image Generation—Antialiasingl Display algorithms; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

Keywords: Antialiasing, complexity analysis, frame buffer techniques, Monte Carlo techniques, numerical analysis, ray tracing, shading.

1 Introduction

Image synthesis usually is performed on a pixel by pixel basis [CPC84, CCC87, HA90] using variance reduced Monte Carlo integration for antialiasing. In [Mit96] the convergence rate of the Monte Carlo method with N samples per pixel was shown to be of the order $\mathcal{O}(N^{-\frac{1}{2}-\frac{1}{2d}})$ where d is the dimension of the integration domain. This is also the rate obtained by quasi-Monte Carlo integration as found by arguments in [PTVF92]. Looking at neighboring pixels, the computational cost can be further reduced by using coherence heuristics (see e.g. [Gla95] for a survey), however most of these algorithms are biased, or do violate the necessary assumptions of the central limit theorem.

In our new approach, we consider the array of pixels forming an image as the samples of a function taken at the positions of the regular rectangular pixel grid. Instead of computing individual pixel values, image synthesis now consists in determining that function, i.e. to calculate all pixel functionals at once. In this context image synthesis belongs to the class of integro-approximation problems, where Monte Carlo integration generalizes to the method of dependent tests [FC62, Sob62].

*Fachbereich Informatik, AG Numerische Algorithmen, keller@informatik.uni-kl.de, http://www.uni-kl.de/AG-Heinrich/Alex.html

In [Hei98] a multilevel version of the method of dependent tests is developed, and it is shown that for smooth model problems up to $\mathcal{O}(n^{\frac{2r}{d}})$ sampling work can be saved, where n is the number of grid points and r the degree of smoothness of the target function. The main idea of the algorithm is similar to principles in image analysis for compression algorithms: If there is correlation, i.e. coherence or smoothness, in an image, e.g. wavelet or Fourier coefficients become small and can be neglected. Transferring this to image synthesis means that coefficients providing only small contribution can be computed at a much smaller accuracy, allowing to dramatically improve convergence.

Based on these ideas we develop a very simple hierarchical variance reduction scheme for computer graphics, which is able to handle the difficult problem of discontinuities and which synthesizes images much faster than contemporary algorithms. The superior performance is obtained by calculating coefficients of a hierarchical function representation, where each level only has a *decreasing contribution* with a *decreasing variance*. Reconstructing the image function from that representation has a considerably reduced variance as compared to a pixel by pixel approach, enormously increasing the efficiency of the Monte Carlo algorithm. The resulting hierarchical Monte Carlo image synthesis procedure can be used as a front end suitable for any shader based ray tracing kernel. However the general underlying mathematical procedure indicates many other applications.

In the next section we will briefly introduce the mathematical concepts needed for the hierarchical Monte Carlo image synthesis algorithm presented in section 3. Then the superior practical performance is illustrated by a numerical reality check in section 4. The new scheme gives raise to many extensions and applications which are pointed out in section 5. The final section draws the conclusions and indicates directions of future research.

2 Image Synthesis

An image is a rectangular matrix of color values. In the sequel we consider a single scanline $(L_k)_{k=0}^{n_m-1}$ containing $n_m = 2^m + 1$ pixels with $m \in \mathbb{N}_0$. The color L_k of the k-th pixel is determined by the mean value integral

$$L_k := \frac{1}{|D|} \int_D L_S(x_k, t)\, dt,$$

where $x_k := \frac{k}{n_m - 1}$. The non-negative, bounded function $L_S : [0,1] \times D \to [0,1]$ is the radiance[1] returned by a shader routine for the ray starting from the eye through the location t of the pixel k anchored at x_k. In the simplest case the integration domain D is the two dimensional area P of a pixel. Including depth of field and motion blur, we get $\dim D = 5$ by additionally integrating over lens area and aperture time. Further adding direct illumination by area light sources or daylight simulation, the dimension becomes 7 [CPC84]. The dimension even can get infinite when using random walk approaches to simulate global illumination like bidirectional path tracing or the Metropolis light transport algorithm [VG97], where D becomes the infinite dimensional path space.

[1] For the sake of simplicity we neglect filtering and tone mapping.

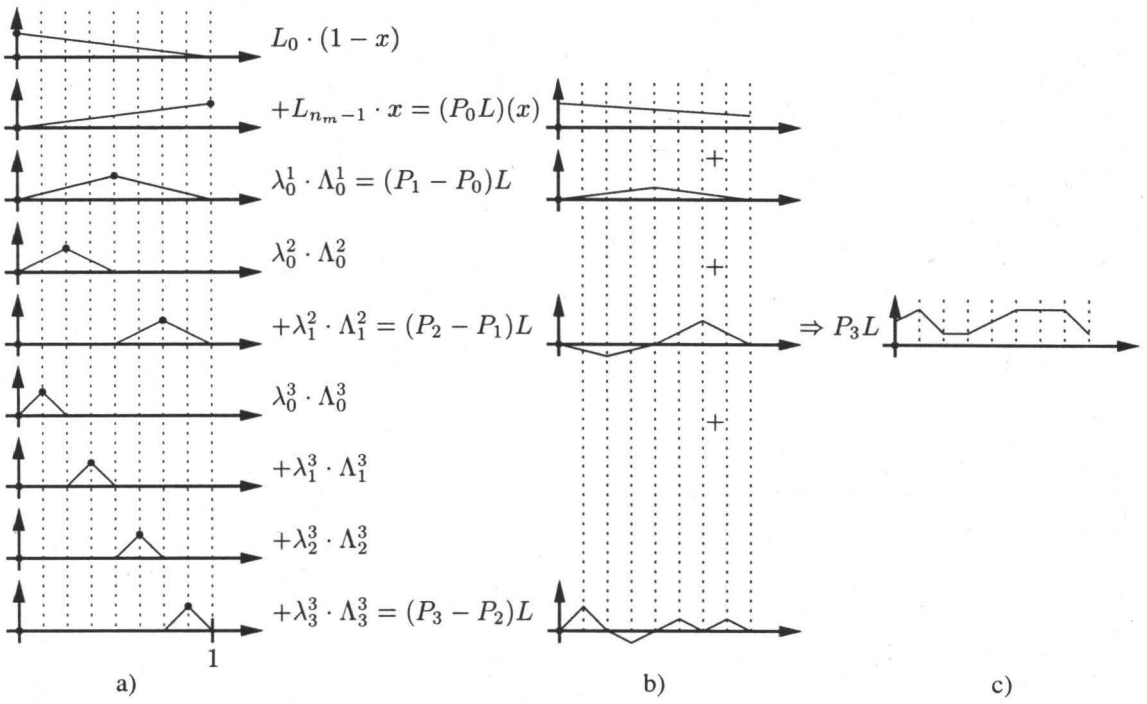


Figure 1: Hierarchical synthesis of a scanline: a) The basis vectors Λ_k^l of the hierarchical hat basis, which are scaled by the coefficients λ_k^l yield the b) difference levels $(P_l - P_{l-1})L$, resulting in the c) reconstructed scanline where $(P_m L)(x_k) = L_k$.

2.1 Integro-Approximation

The pixels $(L_k)_{k=0}^{n_m-1}$ of a scanline also can be considered as samples of a continuous function

$$L(x) := \frac{1}{|D|} \int_D L_S(x,t)\, dt \in C([0,1]),$$

where L is an integral dependent on the parameter x and fulfils the interpolation constraint

$$L(x_k) = L_k.$$

Thus image synthesis can be interpreted as integro-approximation problem which can be solved by constructing an interpolation operator P_m of the kind that $L \approx P_m L$ in the sense that

$$(P_m L)(x_k) = L_k.$$

2.2 The Method of Dependent Tests

In the sequel we will apply the method of dependent tests [FC62, Sob62] which is a generalization of the Monte Carlo method: Instead of estimating a single function value (i.e. a pixel), the whole function (i.e. the whole scanline) is estimated by

$$L(x) \approx \frac{1}{N} \sum_{i=0}^{N-1} L_S(x, t_i). \tag{1}$$

The very important thing to note is that *only one* set of i.i.d. random samples $(t_i)_{i=0}^{N-1} \subset D$ is chosen and then used for *all* values of $x \in [0,1]$. This is similar to approaches as in [HA90, Kel97], where in the case of $D = P$ in every pixel k we find the same supersampling pattern. So estimates of L for different values of x are not statistically independent, allowing to reduce variance by exploiting correlation similar to [Hei98].

2.3 Multilevel Function Representation

For an arbitrary sequence $(P_l)_{l=0}^m$ of interpolation operators, we can write

$$P_m L = P_0 L + \sum_{l=1}^{m} [P_l - P_{l-1}]L$$

yielding a multilevel representation of P_m which is the sum of the base interpolation P_0 and the difference levels $P_l - P_{l-1}$.

2.3.1 Piecewise linear hierarchical Interpolation

We now select a dyadic hierarchy of $m+1$ grids each using $n_l := 2^l + 1$ grid points in the level l, where $0 \le l \le m$ and choose the piecewise linear Lagrange interpolation

$$(P_l L)(x) := \sum_{k=0}^{n_l - 1} L_{k 2^{m-l}} \Lambda(2^l x - k),$$

where $\Lambda(x) := \max\{0, 1 - |x|\}$ is the hat function. Using the dilated and translated hat function

$$\Lambda_k^l(x) := \Lambda(2^l x - 2k - 1),$$

in level $l = 0$ we get

$$\begin{aligned} (P_0 L)(x) &= L_0 \cdot (1-x) + L_{n_m-1} \cdot x, \text{ and} \\ ([P_l - P_{l-1}]L)(x) &= \sum_{k=0}^{2^{l-1}-1} \lambda_k^l \Lambda_k^l(x) \end{aligned}$$

for the remaining difference levels where $l > 0$ with the $n_l - n_{l-1} = 2^{l-1}$ coefficients

$$\lambda_k^l = L_{(2k+1)2^{m-l}} - \frac{L_{k2^{m-(l-1)}} + L_{(k+1)2^{m-(l-1)}}}{2}, \tag{2}$$

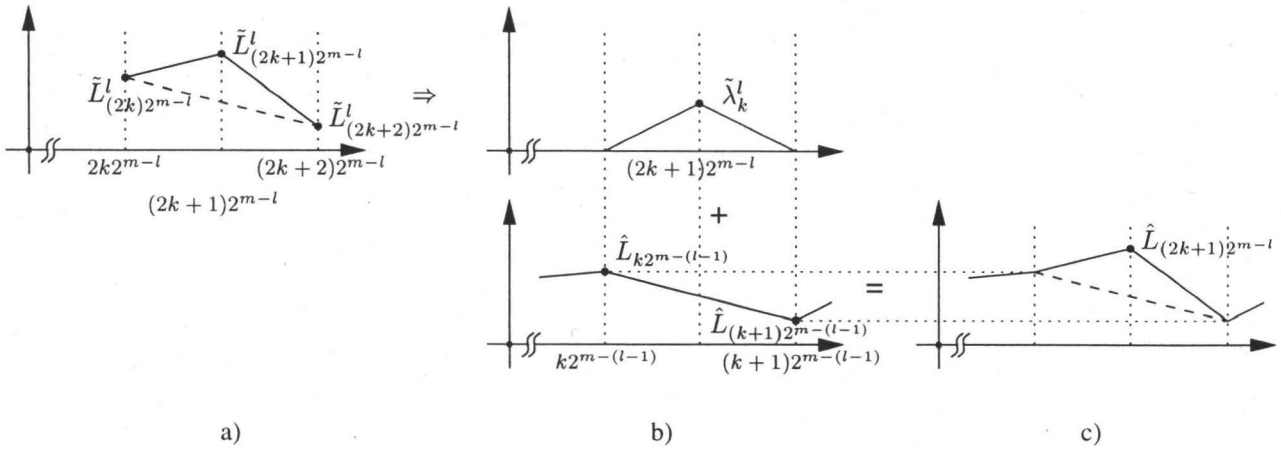


Figure 2: Hierarchical refinement: a) Estimation of $\tilde{L}^l_{(2k+1)2^{m-l}}$, $\tilde{L}^l_{(2k)2^{m-l}}$, and $\tilde{L}^l_{(2k+2)2^{m-l}}$ in level l with N_l samples, then b) computation of the corrector $\tilde{\lambda}^l_k$ and prediction by linear interpolation of $\hat{L}_{k2^{m-(l-1)}}$, and $\hat{L}_{(k+1)2^{m-(l-1)}}$, which were already computed on the coarser level $l-1$, finally c) yielding the new pixel value $\hat{L}_{(2k+1)2^{m-l}}$ on the finer level l.

where the current integer step width at level l is 2^{m-l}. Seen in the context of the lifting scheme [Swe96], λ^l_k corresponds to the correction term (or detail coefficient) for linear prediction, however our function decomposition misses the lifting step! As illustrated in figure 1, a scanline $(L_k)_{k=0}^{n_m-1}$ can be represented in the hierarchical hat basis by the two outmost color values L_0, L_{n_m-1}, and the coefficients $(\lambda^l_k)_{k=0}^{2^{l-1}-1}$ for $0 < l \leq m$.

3 The new Algorithm

The hierarchical Monte Carlo image synthesis algorithm is a hierarchical variance reduction scheme, where a coarser function approximation is used to separate the main part of a finer approximation level. Since correlation in an image causes the finer levels to have less contribution to the reconstructed function (see [Swe96]), more work is spent to compute the predictor levels than for the corrector (i.e. the detail) levels. However the prediction may be unreliable due to discontinuities in the integrand and hence it may be necessary to avoid large errors by localization.

3.1 Hierarchical Sampling

The large support of the basis function in the coarser levels requires very exact computation, while the coefficients in the finer levels have small support and small contribution, and as such need not to be of that high accuracy to obtain an overall reduced variance. We select the sampling rate

$$N_l := N \cdot \frac{n_m}{n_l} \cdot 2^{\alpha l} \cdot \frac{2^\alpha - 1}{2^{\alpha(m+1)} - 1}$$

for the integrals at level l, where $\alpha \in \mathbb{R}^+$ is a positive parameter controlling the geometric decay of the number of samples in the finer levels for the dyadic hierarchy from section 2.3.1. Larger values of α concentrate more samples in the finer levels. The total sampling cost then is the sum over the $m+1$ levels of the samples used per grid point:

$$\begin{aligned}
\sum_{l=0}^{m} N_l \cdot n_l &= \sum_{l=0}^{m} N \cdot \frac{n_m}{n_l} \cdot 2^{\alpha l} \cdot \frac{2^\alpha - 1}{2^{\alpha(m+1)} - 1} \cdot n_l \\
&= N \cdot n_m \cdot \frac{2^\alpha - 1}{2^{\alpha(m+1)} - 1} \cdot \sum_{l=0}^{m} (2^\alpha)^l \\
&= N \cdot n_m \cdot \frac{2^\alpha - 1}{2^{\alpha(m+1)} - 1} \cdot \frac{(2^\alpha)^{(m+1)} - 1}{2^\alpha - 1} \\
&= N \cdot n_m
\end{aligned}$$

Obviously N_l has been chosen the way that the total cost is equal to the cost of the usual approach, where we would use N samples for each pixel of a scanline with resolution n_m. For the real sampling routine the truncated integer values $\lfloor N_l \rfloor$ are used, whereas N can be chosen from $\mathbb{R}^+$.

3.2 Hierarchical Refinement

We now use the method of dependent tests (1) to estimate coefficients of the hierarchical function representation of section 2.3.1. Therefore a sample set $(t_i)_{i=0}^{N_m-1}$ is fixed once in advance for *all* computations and we denote

$$\tilde{L}^l_k := \frac{1}{N_l} \sum_{i=0}^{N_l-1} L_S(x_k, t_i). \tag{3}$$

The algorithm now is recursively defined by starting in level P_0, where we approximate $L_0 \approx \hat{L}_0 := \tilde{L}^0_0$ and $L_{n_m-1} \approx \hat{L}_{n_m-1} := \tilde{L}^0_{n_m-1}$. Then the subsequent refinement is derived from equation (2):

$$\begin{aligned}
&L_{(2k+1)2^{m-l}} \\
&= \underbrace{\frac{L_{k2^{m-(l-1)}} + L_{(k+1)2^{m-(l-1)}}}{2}}_{\text{Predictor}} + \underbrace{\lambda^l_k}_{\text{Corrector}} \\
&\approx \hat{L}_{(2k+1)2^{m-l}} \\
&:= \frac{\hat{L}_{k2^{m-(l-1)}} + \hat{L}_{(k+1)2^{m-(l-1)}}}{2} \\
&\quad + \underbrace{\tilde{L}^l_{(2k+1)2^{m-l}} - \frac{\tilde{L}^l_{(2k)2^{m-l}} + \tilde{L}^l_{(2k+2)2^{m-l}}}{2}}_{=:\tilde{\lambda}^l_k},
\end{aligned} \tag{4}$$

where the predictor is the function linearly interpolated at the coarser level $l-1$ and the corrector $\tilde{\lambda}_k^l$ is computed at the finer level l to yield the approximate pixel value $\hat{L}_{(2k+1)2^{m-l}}$. This hierarchical refinement procedure, which is illustrated in figure 2, in fact is a hierarchical separation of the main part: Functionals with large support are calculated at high precision and used as control variate for the next finer level, where the functionals with smaller support are estimated at reduced precision. If correlation is present, it will be detected using the method of dependent tests resulting in small contributions $\tilde{\lambda}_k^l$.

3.3 To localize or not to localize...

Implementing the algorithm from the previous section yields superior results in smooth areas of the image while introducing disturbing interpolation artifacts as can be seen in figure 5a. These artifacts raise from locally high variance in the estimator (3) on D: Looking at direct illumination calculations, the predictor is calculated at sufficient accuracy. The corrector then is calculated at a lower sampling rate and it can happen, that the estimator dramatically over- or underestimates the true contribution, since e.g. by chance all rays happen to see or do not happen to see the light source, although it is partially occluded. Then the linear interpolation in the prediction step propagates this large error over all successive levels, resulting in the striping artifacts.

This artifact is easily prevented by localization, i.e. by replacing the predictor-corrector value $\hat{L}_{(2k+1)2^{m-l}}$ by the direct estimate $\tilde{L}^l_{(2k+1)2^{m-l}}$. A robust indicator for localization must depend only on the set of samples $(L_S(x_k,t_i))_{i=0}^{N_l-1}$ in order to be general, and cannot rely on statistical tests, since N_l might be too small to fulfil the assumptions of the law of large numbers. Localization is indicated[2]

1. if the radiance in (4) exits the valid range, i.e.

$$\hat{L}_k \notin [0,1].$$

2. if the contrast between the predictor-corrector value and the direct estimate in (4) is too large, i.e.

$$C(\hat{L}_k, \tilde{L}_k^l) > T_1,$$

where T_1 is the contrast threshold and

$$C(\hat{L}_k, \tilde{L}_k^l) := \frac{|\tilde{L}_k^l - \hat{L}_k|}{\tilde{L}_k^l + \hat{L}_k} \in [0,1] \text{ for } \hat{L}_k, \tilde{L}_k^l \geq 0.$$

3. if any estimator $\tilde{L}_k^l$ in (4) has inherent high contrast, i.e.

$$C(\min \tilde{L}_k^l, \max \tilde{L}_k^l) > T_2,$$

where $\min \tilde{L}_k^l := \min_{0 \leq i < N_l} L_S(x_k,t_i)$ and $\max \tilde{L}_k^l$ is defined analogously.

Although L_S has been assumed to be non-negative and bounded, the first case can occur due to estimation errors in the corrector. Compared to empirical variance, which is not very reliable at low sampling rates, contrast is a much sharper indicator, is easily determined while computing (4), and senses similar to a human eye. The second case clearly indicates striping, since either the predictor or the corrector cause a large deviation from the direct pixel estimate.

[2]The different criterions have to be applied to each wavelength, e.g. red, green, blue, of the color vectors L. This is omitted here for the sake of simplicity.

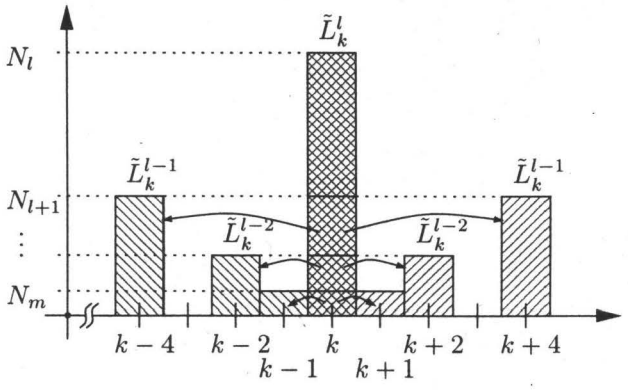


Figure 3: On the fly addition of intermediate $\tilde{L}_k^{l'}$ for $m \geq l' > l$ while computing $\tilde{L}_k^l$.

By the third condition we prevent prediction errors in the levels of small sample size, where a large contrast in a large sample set indicates a potentially large error caused by taking only a fraction of that sample set at a finer level. Figure 5b shows the effect of localization, and figure 5c illustrates that localization rarely occurs, so that we can take almost full advantage of the new sampling scheme.

3.4 Implementation

Hierarchical Monte Carlo image synthesis is implemented as an in-place reconstruction algorithm, requiring only the storage for the scanline. It initializes the scanline to black and computes the outmost pixels $\tilde{L}_0^0$ and $\tilde{L}_{n_m-1}^0$ by (3). Looking at the recursion (4), we see that all estimates $\tilde{L}_k^{l'}$ for $m \geq l' > l$, implicitely computed during the calculation of $\tilde{L}_k^l$, are needed for the predictor calculation in the finer levels. Therefore these values are added on the fly to the corresponding pixels in the scanline as illustrated in figure 3. The set of samples $(t_i)_{i=0}^{N_m-1}$ must be incremental, so that every subset of the first N_l $(0 < l \leq m)$ samples is a quadrature, too.

The subsequent refinement first estimates the central value $\tilde{L}^l_{(2k+1)2^{m-l}}$ using the nested integration scheme. The sum of the left and right value $\tilde{L}^l_{(2k)2^{m-l}}$ and $\tilde{L}^l_{(2k+2)2^{m-l}}$ is already stored in the pixel $(2k+1)2^{m-l}$ by the nested integration scheme, so that $\tilde{\lambda}_k^l$ and finally $\hat{L}_{(2k+1)2^{m-l}}$ can be computed by (4).

Before storing $\hat{L}_{(2k+1)2^{m-l}}$, we check the conditions from section 3.3. In case of localization $\hat{L}_{(2k+1)2^{m-l}}$ is replaced by $\tilde{L}^l_{(2k+1)2^{m-l}}$. If the sample rate N_l is below some minimum rate $N_{fail} < N_0$, at least N_{fail} samples are drawn to estimate $\tilde{L}$. The localized pixels are predicted to be integrals of high variance, so N_{fail} should be chosen large as compared to the average pixel sampling rate N. Since localization usually is a rare event, this does not influence the performance of the algorithm very much.

The complete algorithm consists of two nested `for`-loops, one controlling the level l, the other performing the reconstruction (4) with integer stepwidth 2^{m-l} and the nested integration in figure 3.

3.4.1 Arbitrary Image Size

The algorithm derived so far works on scanlines with $n_m = 2^m + 1$ pixels. By choosing $L_k = 0$ (i.e. black) for $n \leq k < n_m$, it is easily extended to an arbitrary scanline resolution n, where n_m is chosen minimal. In a more elaborate approach, we could apply extrapolation in the same way as it is used in the lifting scheme

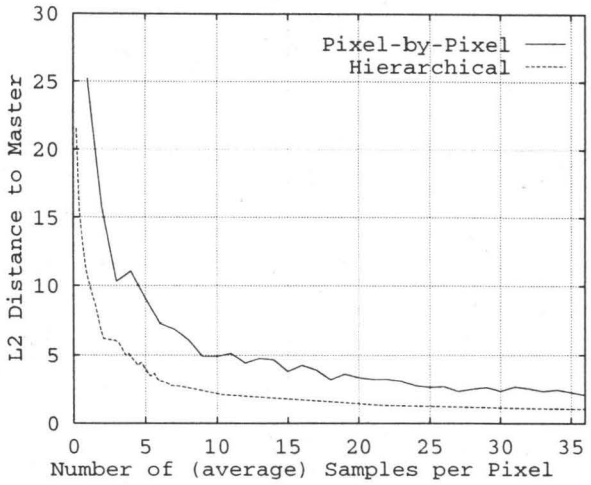


Figure 4: Convergence Graph: Comparison of the usual pixel per pixel approach to the new hierarchical image synthesis algorithm.

[Swe97] to treat the image border. For the scope of this article however, we chose the first approach in order to clearly present the basic algorithm. For parallel computation it is useful to divide the image in small blocks and to apply the algorithm to each of the subimages.

4 Numerical Results and Discussion

The set of parameters is robust and easily chosen. A good choice for the parameter of decay is $\alpha \approx 0.3$. The two contrast thresholds T_1 and T_2 are multiples of the usual threshold $T = (T_{red}, T_{green}, T_{blue}) = (0.4, 0.3, 0.6)$, where we chose $T_1 = 0.8T$ and $T_2 = 2T$. Then we fix the average sampling rate N and the failsafe sampling rate N_{fail}. For the measurements we fixed $N = \frac{N_{fail}}{2}$. If N_{fail} is small, so is N, and it might happen that some levels become interpolated in the regions where no localization event prevents hierarchical reconstruction. That way even sampling rates of less than 1 sample per pixel are achieved. The performance of hierarchical Monte Carlo image synthesis is illustrated in the graph in figure 4. A master solution of the test scene in figure 5 has been calculated using the pixel by pixel approach with 2500 samples per pixel. One sample consists in shooting a primary ray from the eye through a sample position in the pixel under consideration and sending one shadow ray to a sample point on the extended light source. Then the $\|\cdot\|_2$-distance of the pixel by pixel approach and the new algorithm with the same average sample number per pixel (i.e. the same computational effort) to that master were compared. The average sampling rate per pixel has been determined by counting all samples used by the new algorithm and dividing this figure by the number of pixels in the image. It is strikingly obvious, that the new scanline integro-approximation scheme is much more efficient than a simple pixel by pixel integration; so almost up to a factor 5 of work can be saved. The main part of the error is caused by the small number of direct estimators enforced by the localization criterion (see the red and yellow areas in the figures 5, 6, and 7). This justifies to select a high failsafe supersampling rate N_{fail}, nevertheless yielding a much higher efficiency as compared to a pixel by pixel sampling scheme.

4.1 Antialiasing

The method of dependent tests is used to detect correlation. Using the same sampling pattern per pixel as in [HA90], may cause objectionable aliasing artifacts. However the hierarchical approach uses a much higher sampling rate in the lower levels of interpolation, whereas higher levels sampled at lower rate only have small contribution. In consequence e.g. area light sources do not appear discretized as point light sources, but present the expected smooth transitions between dark and lit parts of the image (see figures 5, 6, and 7). For an optimal sampling, a hierarchical Poisson Disk pattern (see the sampling survey in [Gla95]) would be suited best.

4.2 Splitting

The localization criterion is sensitive to the overall contrast of the samples returned by the shader function. In order to reduce the localization events, it is useful to apply splitting [AK90] in the shaders. Considering an extended light source, this means to span multiple stratified shadow rays for one primary ray. This prefiltering of the light incident in the point seen by the primary ray reduces the overall contrast of the shader return values, which in consequence causes less localization events and increases the efficiency of the algorithm. Splitting should be applied for all dimensions of the integration domain D which are prone to high variance.

Daylight simulation incorporates an expensive hemispherical integration. Splitting is illustrated for the overcast sky model for a relatively smooth scene in figure 6 and a very complex model in figure 7. For each primary ray the shaders performed a prefiltering by sending 49 stratified shadow rays into the sky. The average pixel supersampling rate was 4.9, respectively 5.1. The new algorithm detects the areas where the target function is highly correlated, as e.g. the ground or the background sky map, and dramatically reduces the number of samples, as can be seen in the false color images, where green indicates areas of hierarchical reconstruction. As compared to a pixel by pixel approach, image generation was accelerated by a factor 2 for the same quality.

5 Applications and Extensions

Hierarchical Monte Carlo image synthesis is a fundamental procedure which can be applied whenever the shader function L_S can be provided. In consequence it directly applies to distribution ray tracing [CPC84], the photon map [JC98] with all its extensions, and bidirectional path tracing [Laf96].

In order to take advantage of the speedup for multiple light sources, the new algorithm is applied separately for each light source by separating the integrals. Otherwise artifacts may appear due to the high variance in pixels with low sampling rate that randomly include only a fraction of the light sources.

Sky illumination computations are similar to expensive final gather operations over the cosine weighted hemisphere as they are required for smooth radiosity or photon map images. Here the new algorithm can be applied to first calculate the irradiance image function, which usually is a very correlated function [WH92], and then convolve it with the textured scene sampled by primary rays. This way the concept of irradiance caching is captured by a much faster and more precise algorithm. However the separation of irradiance computation and texture filtering results in a biased algorithm.

The prediction in (4) also can be used for fast incremental display of the image while it is being computed or transferred by a render server.

5.1 Rendering Animations

If enough storage is available, it is straightforward to extend the algorithm from the scanline to the two-dimensional image, or even to an animation over the additional dimension of time. This way it is possible to exploit further correlation, bearing a dramatic potential for increasing rendering efficiency for animations as already indicated by the convergence graph in figure 4.

5.2 Quasi-Monte Carlo Integration

The core sampling routine (3) is nothing else than pixel sampling. So the random sampling pattern $(t_i)_{i=0}^{N_m-1}$ simply can be replaced by an incremental low discrepancy sampling pattern, like e.g. the Halton sequence, or the even more elaborate jittered low discrepancy sampling pattern, making the method save about another 30% of work [HK94, Kel97], which is only possible, because the localization criterion is not based on statistical items, like the e.g. empirical variance, which are not available for deterministic sampling.

5.3 Graphics Hardware

Implementing a two-dimensional version of the new algorithm is straightforward. It could be implemented using a hierarchical Z-buffer [GK93] with hierarchical occlusion culling [ZMHH97] ontop of an accumulation buffer [HA90]. The necessary operations like the hierarchical distribution of sample values and the decision whether or not to interpolate are easily coded into a graphics controller. The hierarchical Monte Carlo image synthesis so unifies the concepts of a hierarchical Z-buffer, hierarchical occlusion culling, and a hierarchical accumulation buffer. Finally, progressive display at varying resolutions is possible as in the Infinite Reality graphics hardware [MBDM97]: If rendering time is limited, intermediate levels of reconstruction can be interpolated to the full resolution of the display by the hardware.

6 Conclusion and Future Research

We presented a very fast and efficient hierarchical Monte Carlo image synthesis algorithm. Due to its generality the new variance reduction scheme is easily adapted to any shader based ray tracer saving up to half an order of magnitude in rendering time, and has much potential for further extensions and even more speedup. The new algorithm is based on the concept of integro-approximation, exploiting correlation exposed by the target function, which is a typical property of image data.

After comparing our approach to other high speed rendering algorithms as e.g. [Guo98], we will investigate the application for directly synthesizing quantized and compressed images as presented in e.g. [BM95] in the context of [CDSB97].

Future research will concentrate on developing an efficient hierarchical algorithm for the lifting scheme, i.e. basis functions with overlapping support and higher order predictors, and to implement obvious applications as e.g. hierarchical form factor calculation in wavelet radiosity, hierarchical Monte Carlo radiosity, bidirectional path tracing, or the Metropolis light transport algorithm.

7 Acknowledgements

The author would like to thank Stefan Heinrich for the profound discussions, Gerald Maitschke from Digital Equipment GmbH for providing a DEC alpha station and Rolf Herken from mental images. Special thanks go to the team of architects Christa Marx, Sandra Mozew, Marc Wilke, and Martina Zang for providing their nice models of the city of Kaiserslautern. The remaining models are courtesy of Greg Larson and Peter Shirley.

References

[AK90] J. Arvo and D. Kirk. Particle Transport and Image Synthesis. In *Computer Graphics (SIGGRAPH 90 Conference Proceedings)*, pages 63 – 66, 1990.

[BM95] M. Bolin and G. Meyer. A frequency based ray tracer. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 409–418. ACM SIGGRAPH, Addison Wesley, August 1995.

[CCC87] R. Cook, L. Carpenter, and E. Catmull. The Reyes image rendering architecture. In Maureen C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, pages 95–102, July 1987.

[CDSB97] R. Claypoole, G. Davis, W. Sweldens, and R. Baraniuk. Nonlinear wavelet transform for image coding. In *Proceedings of the 31st Asilomar Conference on Signals, Systems, and Computers, Volume 1*, pages 662–667, 1997.

[CPC84] R. Cook, T. Porter, and L. Carpenter. Distributed Ray Tracing. In *Computer Graphics (SIGGRAPH 84 Conference Proceedings)*, pages 137–145, 1984.

[FC62] A. Frolov and N. Chentsov. On the calculation of certain integrals dependent on a parameter by the Monte Carlo method. *Zh. Vychisl. Mat. Mat. Fiz.*, 2(4):714 – 717, 1962. (in Russian).

[GK93] N. Greene and M. Kass. Hierarchical Z-buffer visibility. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 231–240, 1993.

[Gla95] A. Glassner. *Principles of Digital Image Synthesis.* Morgan Kaufmann, 1995.

[Guo98] B. Guo. Progressive radiance evaluation using directional coherence maps. In Michael Cohen, editor, *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, pages 255–266. ACM SIGGRAPH, Addison Wesley, July 1998.

[HA90] P. Haeberli and K. Akeley. The Accumulation Buffer: Hardware Support for High-Quality Rendering. In *Computer Graphics (SIGGRAPH 90 Conference Proceedings)*, pages 309–318, 1990.

[Hei98] S. Heinrich. Monte Carlo Complexity of Global Solution of Integral Equations. *J. Complexity*, 14:151–175, 1998.

[HK94] S. Heinrich and A. Keller. Quasi-Monte Carlo Methods in Computer Graphics, Part I: The QMC-Buffer. Interner Bericht 242/94, University of Kaiserslautern, 1994.

[JC98] H. Jensen and P. Christensen. Efficient Simulation of Light Transport in Scenes with Participating Media using Photon Maps. In Michael Cohen, editor, *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, pages 311–320. ACM SIGGRAPH, Addison Wesley, July 1998.

[Kel97] A. Keller. Instant Radiosity. In *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 49–56, 1997.

[Laf96] E. Lafortune. *Mathematical Models and Monte Carlo Algorithms for Physically Based Rendering*. PhD thesis, Katholieke Universitiet Leuven, Belgium, 1996.

[MBDM97] J. Montrym, D. Baum, D. Dignam, and C. Migdal. InfiniteReality: A Real-Time Graphics System. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 293–302. ACM SIGGRAPH, Addison Wesley, August 1997.

[Mit96] D. Mitchell. Consequences of Stratified Sampling in Graphics. In *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 277–280, 1996.

[PTVF92] H. Press, S. Teukolsky, T. Vetterling, and B. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1992.

[Sob62] I. Sobol. The use of w^2-distribution for error estimation in the calculation of integrals by the Monte Carlo method. *U.S.S.R. Comput. Math. and Math. Phys.*, 2:717 – 723, 1962.

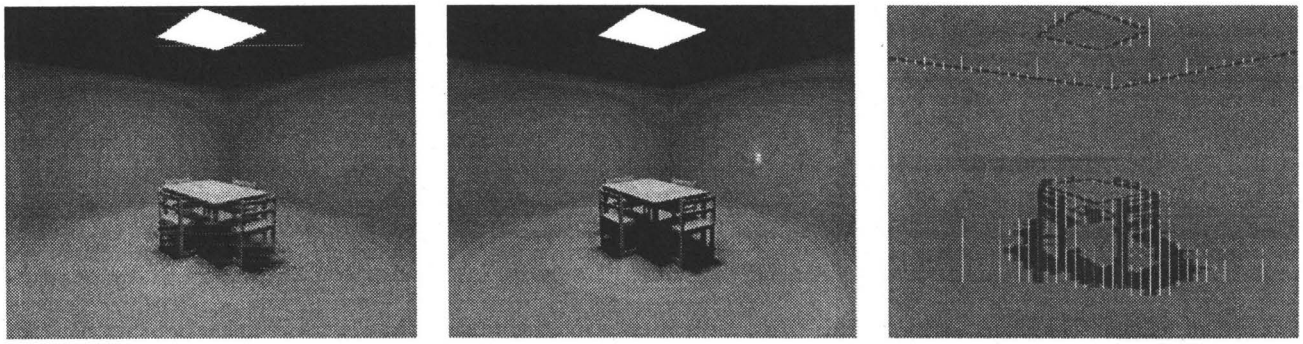
[Swe96] W. Sweldens. Wavelets and the lifting scheme: A 5 minute tour. *Z. Angew. Math. Mech.*, 76 (Suppl. 2):41–44, 1996.

[Swe97] W. Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2):511–546, 1997.

[VG97] E. Veach and L. Guibas. Metropolis light transport. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 65–76. ACM SIGGRAPH, Addison Wesley, August 1997.

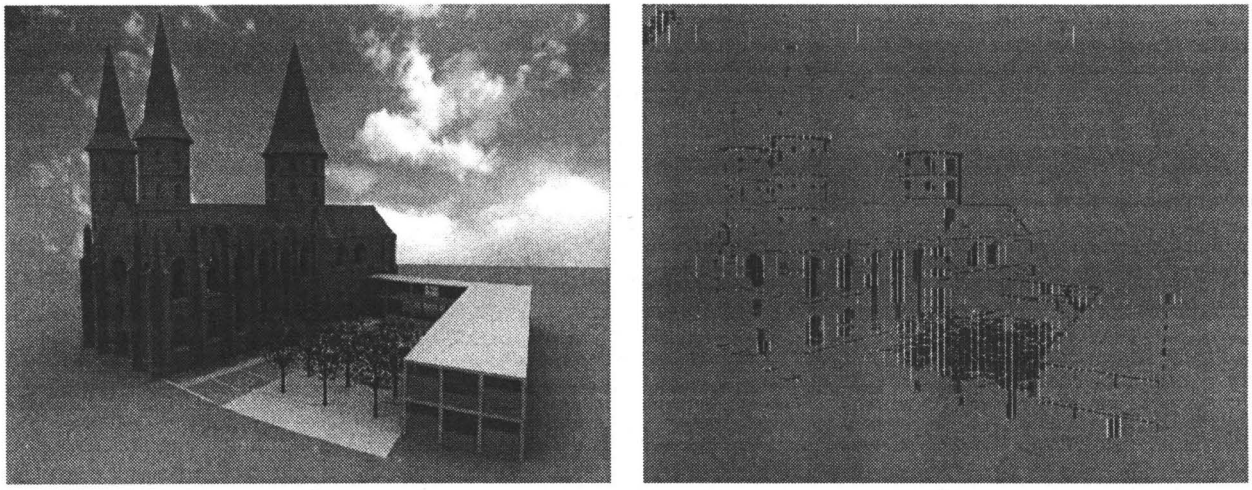
[WH92] G. J. Ward and P. Heckbert. Irradiance gradients. In *3rd Eurographics Workshop on Rendering*, Bristol, United Kingdom, May 1992.

[ZMHH97] H. Zhang, D. Manocha, T. Hudson, and K. Hoff III. Visibility culling using hierarchical occlusion maps. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 77–88. ACM SIGGRAPH, Addison Wesley, August 1997.



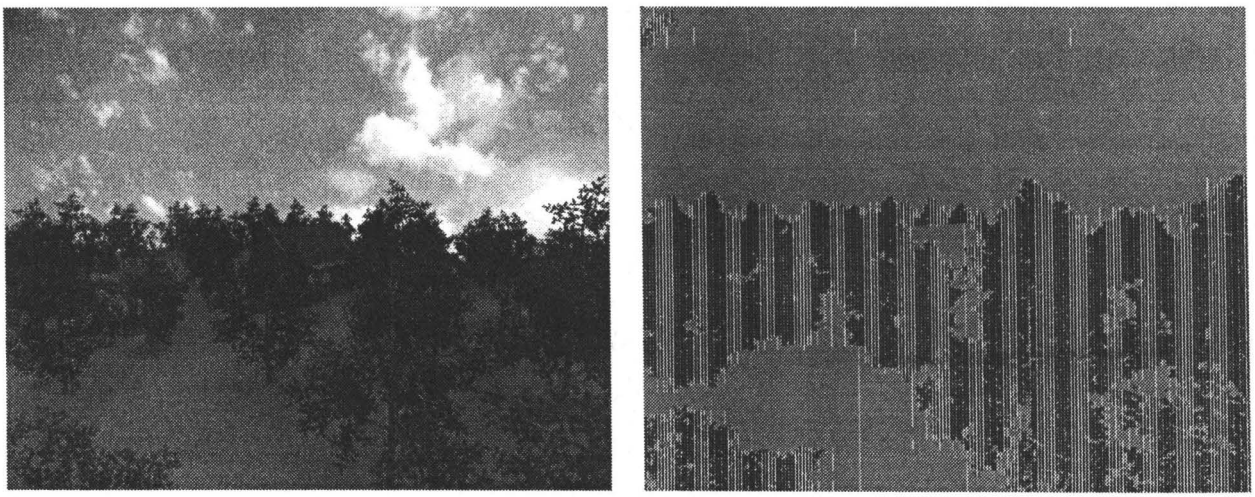
a) b) c)

Figure 5: Localization: a) Striping effects in the basic algorithm without localization due to high variance. b) Striping removed by the localization criterion and its c) impact, where green indicates the use of the predictor-corrector result $\hat{L}_{(2k+1)2^{m-l}}$, yellow indicates the use of the local estimate $\tilde{L}^l_{(2k+1)2^{m-l}}$, and red is used whenever $\tilde{L}^l_{(2k+1)2^{m-l}}$ has been upsampled due to $N_l < N_{fail}$ in the local case.



a) b)

Figure 6: a) Overcast sky daylight architectural visualization, and b) localization (see figure 5 for the explanation of the colors).



a) b)

Figure 7: a) Overcast daylight simulation for a very complex model, and b) localization (see figure 5 for the explanation of the colors).