Bayesian Monte Carlo approach In Computer Graphics

K. Bouatouch
Contributors: J. Brouillat, C. Bouville, R. Marques, B. Loos, C. Hansen,

University of Rennes 1, IRISA, FRVSense team

September 23, 2011

References

- J. Brouillat, C. Bouville, B. Loos, C. Hansen, K. Bouatouch
  A Bayesian Monte Carlo Approach to Global Illumination
  Computer Graphics Forum, Volume 28, Number 8, December 2009 , pp. 2315-2329(15)
- Jonathan Brouillat,
  Bayesian Monte Carlo approach to global illumination and photon-driven photon mapping
  PhD thesis, http://hal.inria.fr/tel-00474571
- J. Brouillat, C. Bouville, K. Bouatouch
  Bayesian Monte Carlo Integration Applied to Global Illumination, Technical Report IRISA, 2011

Motivations (1)

$$I = \int f(x)p(x)dx$$

- The Monte Carlo estimator depends on the arbitrary choice of the sampling density.
- Hence, the same set of observed integrand sample values will lead to different estimates depending on the chosen sampling density.
- This violates a principle of Bayesian statistics: the Likelihood Principle.

Motivations (2)

- Monte Carlo ignores sample locations and use only the value of integrand samples.
- Two samples falling on the same or close location will have equal importance, whereas the second sample brings no extra information.
- Stratified sampling and/or (deterministic) quasi-Monte Carlo reduce the occurrence of theses cases
- Classical Monte Carlo wastes important information.

The Bayesian Approach

- The Bayesian approach turns the problem of evaluating the integral into a Bayesian inference problem.
- For a given x, the integrand $f(x)$ is considered as random because it is unknown (and thus uncertain) before its evaluation.
- Bayesian Monte Carlo relies on an *a priori* knowledge of a probabilistic model of the integrand (e.g. gaussian process model).

The Bayesian Approach

In classical Monte Carlo, we want to evaluate:

$$I = \int f(x)p(x)dx$$

where $p(x)$ is a *pdf*.

Recall that Classical Monte Carlo gives:

$$\hat{I} = \frac{1}{T}\sum_{t=1}^{T} f(X_t)$$

where X_t are random samples drawn from $p(x)$.

The Bayesian Approach

Bayesian view is that all forms of uncertainty are represented by probabilities: we think of the unknown desired quantity as being random.

- $\hat{I}$ and $f(x)$ are unknown until we evaluate them.
- How do we model the uncertainty on $\hat{I}$ and $f(x)$?

The Bayesian Approach

- Put a prior on f (gaussian process model),
- Combine with a vector of observations D,
- We obtain a posterior over f, (also a gaussian process)
- This posterior gives a conditional distribution $p(I|D)$, (gaussian)
- The expected value of the distribution gives us $\hat{I}$ (maximum likelihood estimation).

Gaussian Process

- Collection of random variables, any finite number of which have a joint gaussian distribution,
- Defined by a mean function $\bar{f}(x)$ and a covariance function: $Cov[f(x_1), f(x_2)] = k(x_1, x_2)$
- Notation : $\mathcal{GP}[\bar{f}(x), k(x, x')]$
- the $\mathcal{GP}$ is stationnary if $f(x)$ is constant and $k(x, x') = k(x - x')$. If $k(x - x') = k(|x - x'|)$, $k()$ is a radial basis function (RBF).
- $k(x_1, x_2)$ must be semi definite positive (SDP)
- With this function close samples are highly correlated wheras $k(x_1, x_2) \approx 0$ for distant samples, which means that the function values are almost independent.

The Bayesian Monte Carlo problem formulation

- The gaussian process model $\mathcal{GP}[\bar{f}(x), k(x, x')]$ is the prior
- Assume an independent gaussian additive noise $\mathcal{N}(0, \sigma^2)$ with samples ϵ_i. The observations y_i are:

$$y_i = f(x_i) + \epsilon_i$$

- The covariance of the observed data is then: $\mathrm{cov}(y_p, y_q) = k(y_p, y_q) + \sigma^2 \delta_{pq}$
- $X = [x_0, x_1, \ldots, x_n]$ is a set of samples.
- $D = [(x_1, y_1), \ldots, (x_n, y_n)]$ is the set of corresponding observations.
- Problem: find the best estimate of I given D.

Bayesian Monte Carlo Estimator

The posterior Gaussian process is

$$\begin{aligned} E(f(x)|D) &= \bar{f}(x) + k(x)^t Q^{-1}[Y - \bar{F}] \\ Cov(f(x), f(x')|D) &= k(x, x') - k(x)^t Q^{-1} k(x') \end{aligned}$$

where

$$\begin{aligned} k(x) &= (k(X1, x), ..., k(X_n, x))^t \\ Y &= (Y_1, ..., Y_n) \\ Q &= K(X, X) + \sigma^2 I_n \\ \bar{F} &= (\bar{f}(X_1), ..., \bar{f}(X_n)) \end{aligned}$$

I_n is the $n \times n$ identity matrix

Bayesian Monte Carlo Estimator

The posterior distribution of the integral I is

$$\begin{aligned} \hat{I} &= E(I|D) = \bar{I} + Z^t Q^{-1}[Y - \bar{F}] \\ Var(I|D) &= \bar{V} - Z^t Q^{-1} Z \end{aligned}$$

where

$$\begin{aligned} \bar{V} &= \int\int k(x,x')p(x)p(x')dxdx' \\ Z &= \int k(X,x)p(x)dx \\ \bar{I} &= \int \bar{f}(x)p(x)dx \\ Q &= K(X,X) + \sigma^2 I_n \\ \bar{F} &= (\bar{f}(X_1), ..., \bar{f}(X_n)) \end{aligned}$$

I_n is the $n \times n$ identity matrix

Bayesian Monte Carlo Estimator

The previous equation giving the estimate of the integral I can be rewritten as:

$$\begin{aligned} \hat{I} &= E(I|D) = \bar{I}_0 + c^t Y \\ \bar{I}_0 &= \bar{I} - c^t \bar{F} \\ c &= Q^{-1} z \end{aligned}$$

This is a quadrature rule in which c is the vector of quadrature coefficients.
These coefficients could be precomputed for predefined sample sets X_i and hyperparameter values.

Bayesian Monte Carlo Estimator

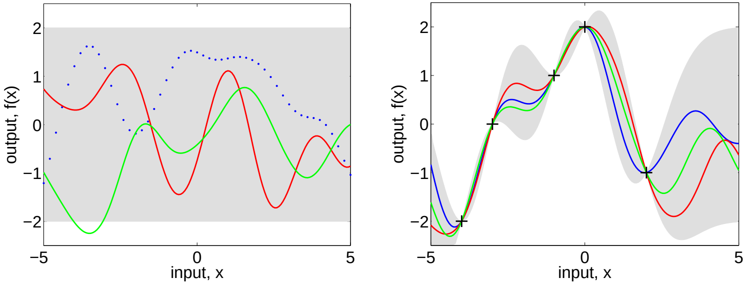
$$
\begin{aligned}
E(I|D) &= \int \int f(x)p(x)dx \; p(f/D)df \\
&= \int [\int f(x)p(f/D)df]p(x)dx = \int \bar{f}_D(x)p(x)dx
\end{aligned}
$$

where $\bar{f}_D(x)$ is the posterior mean function.

- In SMC if two samples happen to fall close to each other the function value there will be counted with double weight.
- That means that large numbers of samples are needed to adequately represent p(x).
- BMC circumvents this problem by choosing samples that are not close to each other, since p(x) is not a pdf but a known function.

Bayesian Regression

$\hat{I}$ estimator uses $E[f(x)|D]$ as an interpolant for f (bayesian regression). Examples from the Rasmussen-Williams' book: "GP for machine learning".



(a), prior (b), posterior

- a) No observations, only $\mathcal{GP}[\bar{f}(x), k(x, x')]$ is known,
- b) The a posteriori estimate of $f(x)$.

Bayesian Monte Carlo Estimator

Bayesian Monte Carlo can significantly outperform classical Monte Carlo if the prior is appropriate. But:

- How to choose the prior i.e. the GP $\mathcal{GP}[\bar{f}(x), k(x, x')]$?
- How to compute the Z vector coefficients and $\bar{I}$?
- How to deal with the matrix inversion Q^{-1} ?

Application to Global Illumination

Can Bayesian Monte Carlo approach be used for Global Illumination and Environment Map Sampling?

1. Can we obtain better rendering quality for the same number of samples?
2. Is it practical? (better rendering quality for the same computation time)

Irradiance incoming at a given point

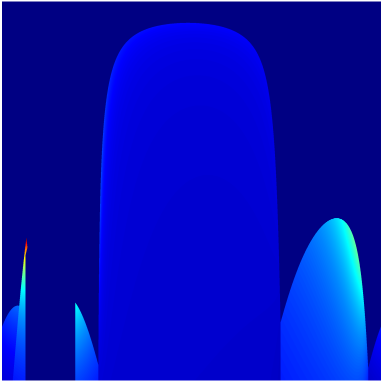
We have first applied Bayesian Monte Carlo to compute the irradiance at a given point x.

$$E = \int_{\Omega} L(x, \omega) \cos(\theta) d\omega.$$

We need a covariance function k (*luminance values incoming from closed directions are likely to be the same*). $L(x, \omega)$ could also stem from an Environment Map.

Irradiance at a Given Point

Luminance incoming at x from all the hemisphere



The gaussian process model

We take a Square Exponential (SE) function to model $k()$:

$$k(x_1, x_2) = k(|x_1 - x_2|) = w_0 e^{\frac{-|x_1 - x_2|}{2\ell^2}}$$

- x_i are direction vectors i.e. points on the unit sphere and $|x_1 - x_2|$ is a 3D cartesian distance
- w_0 is the variance of $f()$
- ℓ (the lenghtscale) characterizes the *strength* of the correlation between samples
- The mean function $\bar{f}$ is assumed constant for final gathering application
- $\{w_0, \ell, \bar{f}, \sigma\}$ are the hyperparameters of the model.

But how to choose these hyperparameters ?

Hyperparameters Determination

The covariance function of the observations y_i:

$$k(x_p, x_q) = k(|x_p - x_q|) = w_0 e^{\frac{-|x_p - x_q|}{2\ell^2}} + \sigma^2 \delta_{pq}$$

First, we measure the actual covariance of the signal, then fit it to the model.

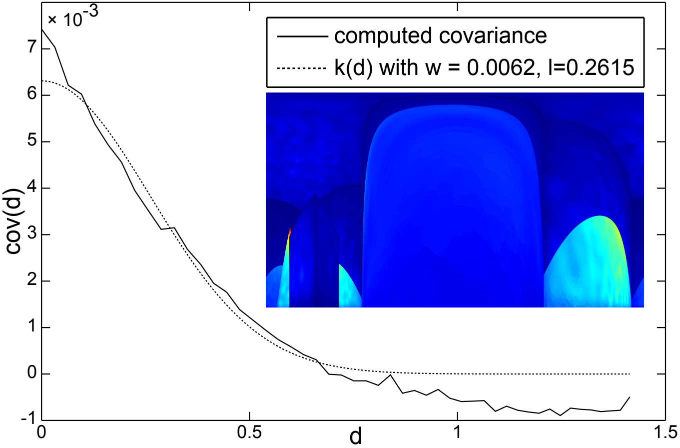
$$k(d) = E[(L(x_1) - \bar{L})(L(x_2) - \bar{L})] \quad with\ d = |x_1 - x_2|$$

Measured covariance of the incoming luminance (25k couples):

$$w_0 = 6.2 \cdot 10^{-3} \quad \ell = 0.2615 \quad \sigma^2 = 0.24$$

Hyperparameters Determination

Covariance function fitting



Hyperparameters Determination

Optimizating the likelihood function

- Let $D_t = \{(x_1, f(x_1)), \ldots, (x_n, f(x_n))\}$ be a set of collected training data
- Let the covariance matrix $Q(D_t, \vartheta)$ be given by

$$Q(\vartheta) = \begin{bmatrix} k(x_1, x_1, \vartheta) & k(x_1, x_2, \vartheta) & k(x_1, x_n, \vartheta) \\ \vdots & \ddots_{\sigma} & \vdots \\ k(x_n, x_1, \vartheta) & \cdots & k(x_n, x_n, \vartheta) \end{bmatrix} + \sigma^2 \, Id_n$$

where Id_n is the identity matrix, and σ is eventual noise of observation contained in $f(x)$

- Find the ϑ which maximizes the likelihood (optimization)

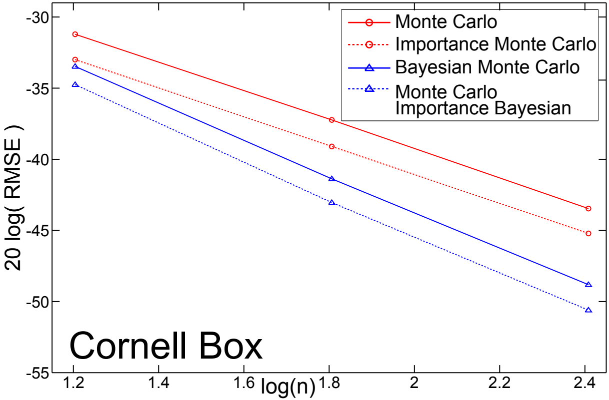
$$\log[p(D_t|\vartheta)] = -\frac{1}{2}(Y - \bar{F})^t Q^{-1}(\vartheta)(Y - \bar{F}) - \frac{1}{2}\log|Q(\vartheta)| - \frac{n}{2}\log(2\pi)$$

Comparison with Classical Monte Carlo

Perform several integral estimations then compute the variance of the results.
Compare:

- Classical Monte Carlo
- Monte Carlo with Importance Sampling
- Bayesian Monte Carlo
- Bayesian Monte Carlo with Importance Sampling

RMSE Comparison



Making BMC Rendering Practical

Still holds the problem of computing $\bar{I}$, Z and Q^{-1}...

- How do we choose $\bar{I}$ ($\bar{f}$)?
- How do we compute the integrals associated with Z?
- How do we manage the cost of inverting Q^{-1} ($n \times n$ matrix)?

For each computation...

Determining $\bar{I}$ and $\bar{f}$

We need to compute $\bar{I}$ value and Z vector.

$$\bar{I} = \int \bar{f}(x)p(x)dx$$

$\bar{f} = I_{MC}$ (in case of constant mean function) the classical Monte Carlo estimator value I.

$$\bar{I} = \pi \bar{f}$$

If ℓ value is too low or is equal to 0, BMC estimator provides the same value as MC in worst cases (e.g. low ℓ value).

Computing z

Z depends only on the samples positions:

$$Z = \int k(X, x)p(x)dx \qquad z_i = \int k(x_i, x)p(x)dx$$

- z_i is thus a function of ℓ and the sampling direction x_i (actually depends on θ_i only).
- As the function $z_i(\ell, \theta_i)$ is very smooth, we precompute a lookup table and interpolate.

Precomputing distributions

Z and the covaraince matrix Q^{-1} depend only on the relative position of the samples to each other. For a given distribution of directions, we can precompute Z and Q^{-1}.

- draws M random distributions of N samples, with $M << nbPixels$
- precompute Z and Q^{-1} and the vector of quadrature coefficients $C_y = Q^{-1}Z$

Precomputing distributions

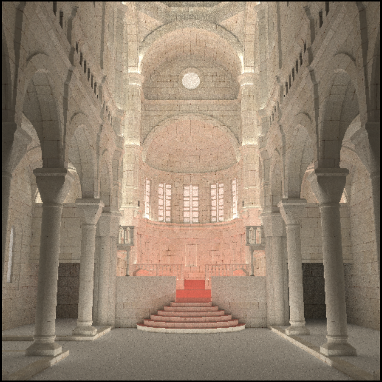
During each the rendering, for each integration:

- randomly pick a distribution $\mathcal{D}$ and the corresponding precomputed C_y vector
- rotate it around the normal axis (same coefficients C_y)
- evaluate samples and compute monte carlo estimation of the integral ($\bar{f}$)
- use C_y to compute the bayesian estimation of the integral with:

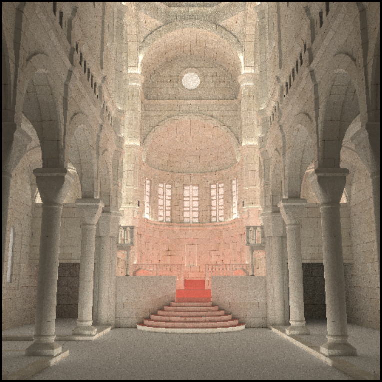
$$\hat{I} = M_0 + C_y^T (Y - \bar{f(X)})$$

Bayesian Monte Carlo Rendering

Uniform MC - 144 samples

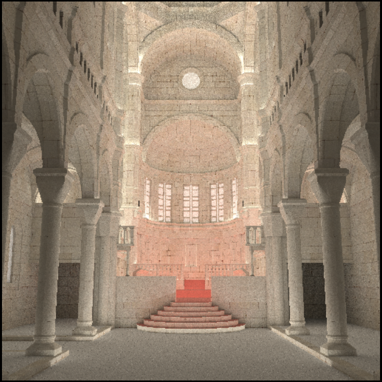


Uniform MC - 144 samples

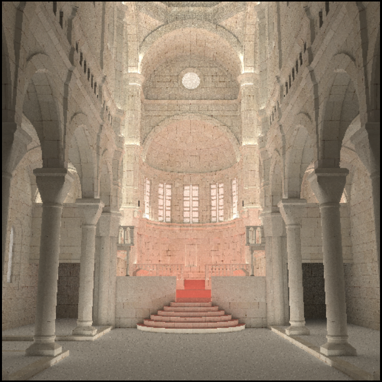


Bayesian Monte Carlo Rendering

Uniform MC - 144 samples

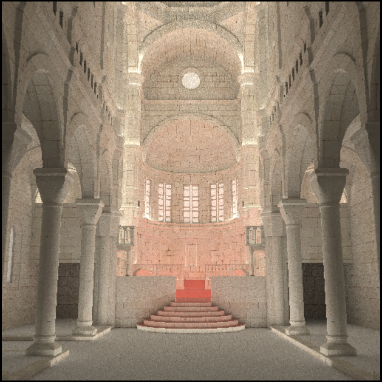


Uniform BMC - 144 samples

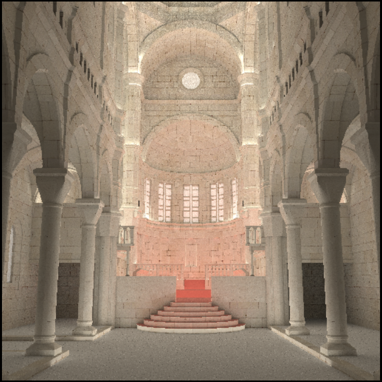


Bayesian Monte Carlo Rendering

Uniform MC - 144 samples

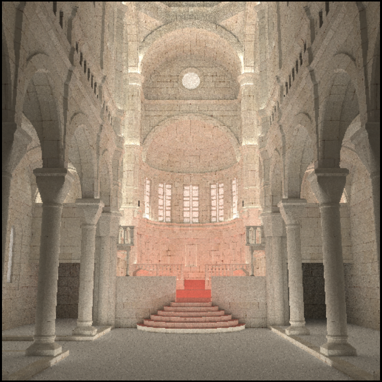


Stratified MC - 144 samples

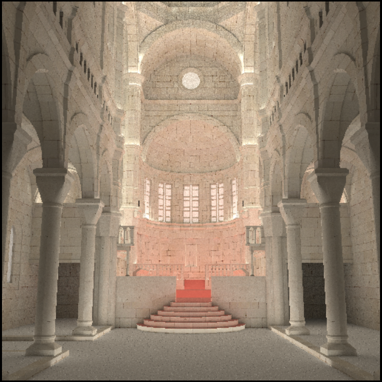


Bayesian Monte Carlo Rendering

Uniform MC - 144 samples



Stratified BMC - 144 samples



Bayesian Monte Carlo Rendering

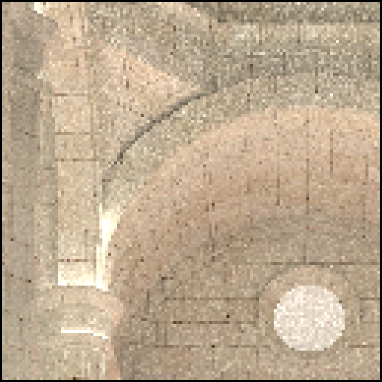
Uniform MC - 144 samples

Uniform MC - 144 samples

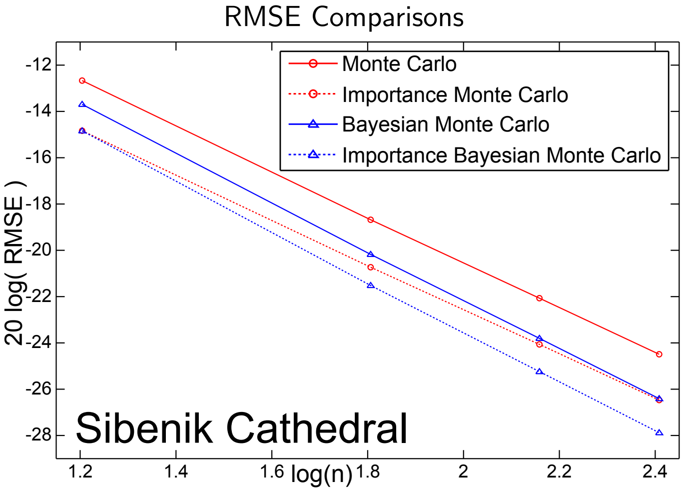
Bayesian Monte Carlo Rendering

Uniform BMC - 144 samples

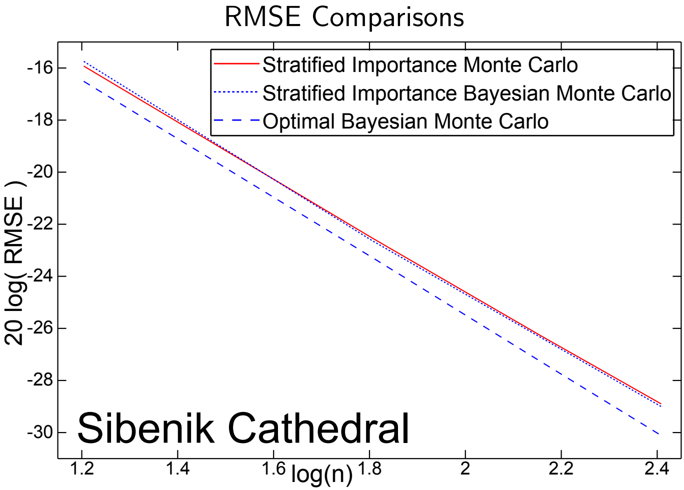
Uniform BMC - 144 samples



Results


RMSE Comparisons
Monte Carlo
Importance Monte Carlo
Bayesian Monte Carlo
Importance Bayesian Monte Carlo
20 log(RMSE)
-12
-14
-16
-18
-20
-22
-24
-26
-28
log(n)
1.2
1.4
1.6
1.8
2
2.2
2.4
Sibenik Cathedral

Results


RMSE Comparisons
Stratified Importance Monte Carlo
Stratified Importance Bayesian Monte Carlo
Optimal Bayesian Monte Carlo
20 log(RMSE)
-16
-18
-20
-22
-24
-26
-28
-30
1.2
1.4
1.6
1.8
2
2.2
2.4
log(n)
Sibenik Cathedral

Optimized Distributions

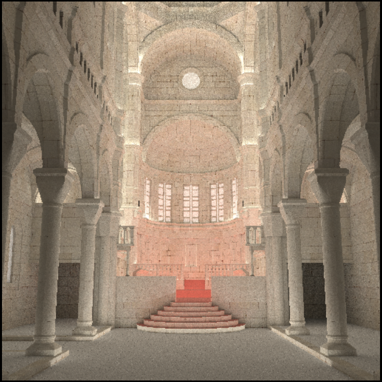
Given a covariance function k we can compute a *theoretical* expression of the variance of the BMC estimate:

$$Var[I|f(D)] = V_0 - Z^t Q^{-1} z \quad (1)$$

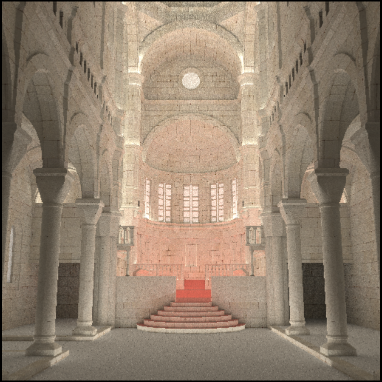
For a signal following our $\mathcal{GP}$ prior, the variance of the BMC estimate depends on the choice of the samples. By an optimization process, we can find a distribution which minimize $Var[I|D]$.

Bayesian Monte Carlo Rendering - Sibenik Cathedral

Uniform MC - 144 samples

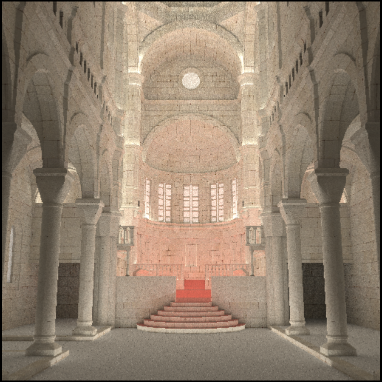


Uniform MC - 144 samples

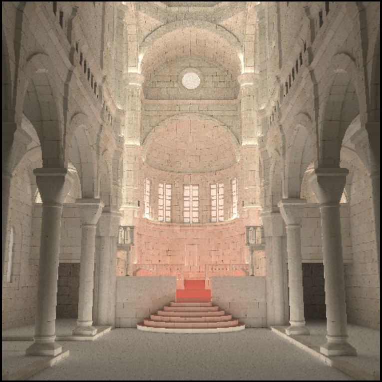


Bayesian Monte Carlo Rendering - Sibenik Cathedral

Uniform MC - 144 samples

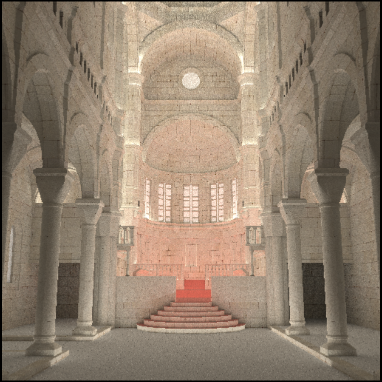


Strat. Imp. MC - 144 samples

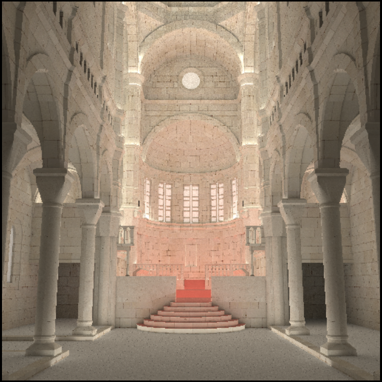


Bayesian Monte Carlo Rendering - Sibenik Cathedral

Uniform MC - 144 samples



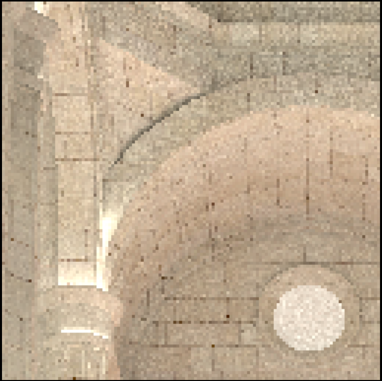
Optimized BMC - 144 samples



Bayesian Monte Carlo Rendering - Sibenik Cathedral

Strat. Imp. MC - 144 samples

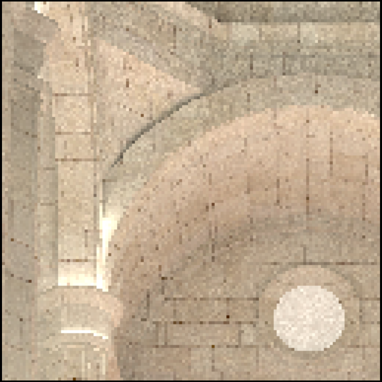
Strat. Imp. MC - 144 samples



Bayesian Monte Carlo Rendering - Sibenik Cathedral

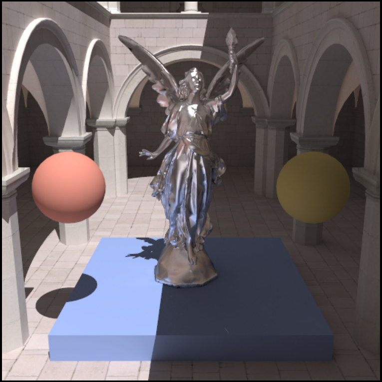
Optimized BMC - 144 samples

Optimized BMC - 144 samples

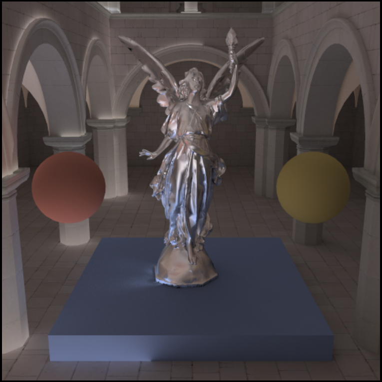


Bayesian Monte Carlo Rendering - Sponza Lucy

Reference

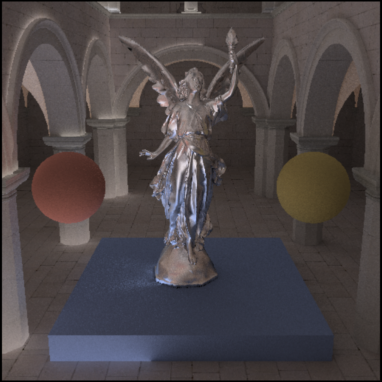


Reference - Indirect

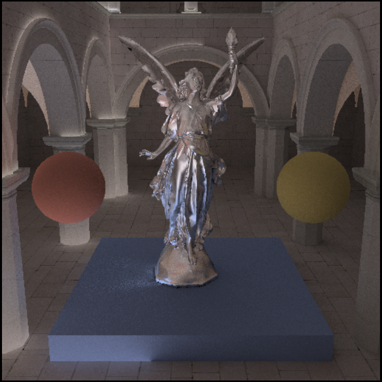


Bayesian Monte Carlo Rendering - Sponza Lucy

Imp. MC - 256 samples

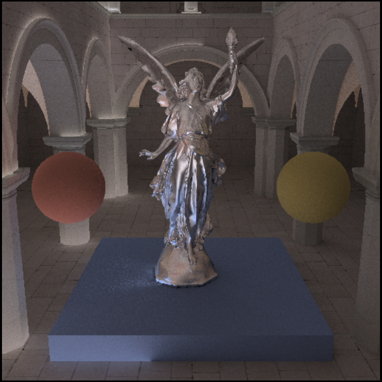


Imp. MC - 256 samples

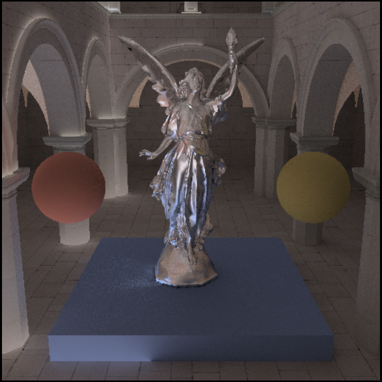


Bayesian Monte Carlo Rendering - Sponza Lucy

Imp. MC - 256 samples



Imp. BMC - 256 samples

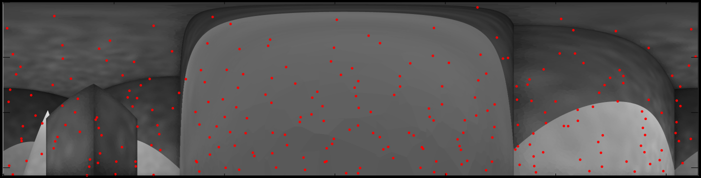


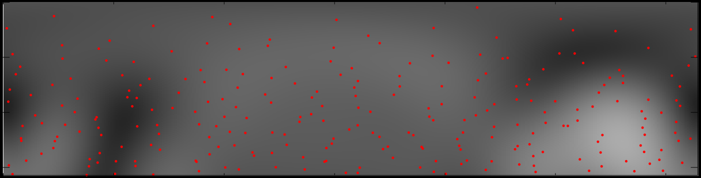
Bayesian Monte Carlo Rendering - Sponza Lucy

MC diff. (x10) - 256 samples

BMC diff. (x10) - 256 samples

Bayesian Monte Carlo Rendering - Quadrature





Bayesian Monte Carlo - Conclusion

- We proposed to apply Bayesian Monte Carlo to computer graphics.
- We showed that despite the particular nature of luminance signal, BMC can reduce the variance when computing irradiance
- We proposed a scheme to overcome the cost of classical BMC (without optimized distributions)
- We showed that BMC performs at least as good as MC, even when used in conjunction with other noise-reduction methods

Bayesian Monte Carlo - Future Works

- Local computation of ℓ and σ: practical?
- Glossy reflections: z becomes 5-dimensional
- Environment mapping
- Path tracing: higher dimensional integrand
- Upsampling: spatial and temporal

Thank you for your attention!
Questions?

Splitting the integrand

Split the integral into several integrals and apply appropriate Monte Carlo optimisation on each part.

$$f(x) = f_0(x) + f_1(x) + f_2(x)$$
$$I = \int_D f_0(x)dx + \int_D f_1(x)dx + \int_D f_2(x)dx$$

- $\int_D f_0(x)dx$ will be evaluated with cosine importance sampling (e.g. phong diffuse part)
- $\int_D f_1(x)dx$ will be evaluated with power cosine importance sampling (e.g. phong specular part)
- $\int_D f_1(x)dx$ is too complex and will be evaluated with stratified sampling only

Control Variates

Sometimes the knowledge about $f(x)$ can not be used for importance sampling:

$$f(x) = g(x) + f'(x) \quad \textit{with} \quad \exists x, g(x) = 0$$

$g(x)$ can be used as an importance sampling function only if:

$$\forall x, g(x) = 0 \Rightarrow f(x) = 0$$

Use $g(x)$ as a control variate.

Control Variates

We know that $f(x)$ has a certain shape:

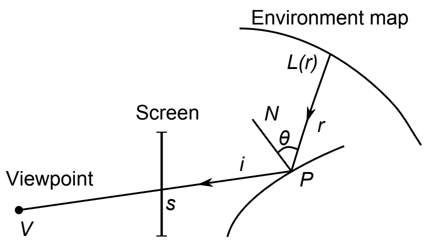
$$f(x) = g(x) + f'(x) \quad \mathit{with} \quad \exists x, g(x) = 0.$$

$g(x)$ is the control variate:

$$I = \int_D f(x)dx = G + \int_D (f(x) - g(x))dx \quad \mathit{with} \quad G = \int_D g(x)dx.$$

The variance of the estimator depends on the choice of $g(x)$.

General rendering equation with an environment map



$$I(c) = \int_{R(c)} h(|s-c|) \left[\int_{\Omega_{2\pi}} f_r[i(s), r] L(r) \cos\theta \, d\Omega \right] ds$$

- c : pixel center
- $h(s)$: anti-aliasing filter kernel
- $R(c)$: anti-aliasing filter window
- $f_r(i, r)$: BRDF

BMC for environment map rendering

- break down the integral into diffuse and specular components using:

$$f_r(i,r) = f_s(i,r) + f_d$$

- proposed covariance function for the integrand:

$$k(s,r,s',r') = w_0 \exp\left[\frac{-|s-s'|^2}{l_s^2} + \frac{-|r-r'|^2}{l_r^2}\right]$$

- closed form solution for computing the z_i coefficients when:
  1. the filter kernel is a box or gaussian
  2. the BRDF is factorized (possibly in squared-exponential functions)