

# Publicações Selecionadas

Luís Paulo Peixoto dos Santos

28 de Julho, 2023

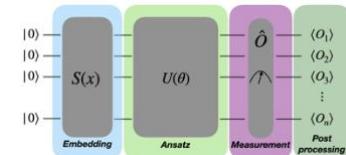
*Luís Paulo Peixoto dos Santos*

---

Este documento identifica cinco publicações seleccionadas pelo candidato como as mais representativas da qualidade e impacto do seu trabalho, conforme requerido no ponto II, 4.2, b) do Edital 1038/2023 que suporta este concurso.

---

1. André Sequeira, Luís Paulo Santos, Luís Soares Barbosa;  
“Policy Gradients using Variational Quantum Circuits”;  
Quantum Machine Intelligence, Volume 5(18), April, 2023  
DOI: <https://doi.org/10.1007/s42484-023-00101-8>  
Ficheiro local: [2023-QuantumMachineIntelligence \(pdf\)](#)

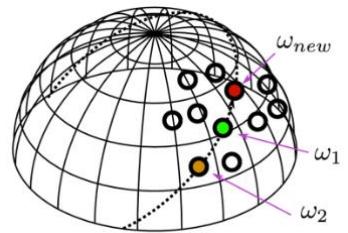


Corresponde ao primeiro artigo publicado pelo candidato numa revista científica cuja área de especialidade é a Computação Quântica – neste caso em particular, a Aprendizagem Máquina Quântica. Neste sentido este artigo constitui-se como um marco, que assinala contribuições originais reconhecidas pela comunidade; antecipa-se que seja apenas o primeiro de muitos artigos, como se verificou noutras áreas. Nota: o candidato publicou previamente artigos que versam a Computação Quântica, mas não em revistas desta área do conhecimento.

Este artigo aborda o problema relativamente pouco estudado do *Quantum Reinforcement Learning*, em particular a utilização de gradientes de estratégias (*policies*), por oposição a usar a função *Q-Value*.

Os autores demonstram que uma aproximação  $\epsilon$  do gradiente pode ser obtida usando um número de amostras logarítmico com o número de parâmetros. Verificam ainda empiricamente que o comportamento do modelo quântico é semelhante ao do modelo clássico equivalente em ambientes de teste *standard*, mas usando apenas uma fração do número total de parâmetros.

2. Thomas Bashford-Rogers, Luís Paulo Santos, Demetris Marnerides, Kurt Debattista;  
 “Ensemble Metropolis Light Transport”;  
 ACM Transactions on Graphics, Volume 41(1), February, 2022  
 Invited presentation at SIGGRAPH’2022  
 DOI: <https://doi.org/10.1145/3472294>  
 Ficheiro local: [2022-TransactionsOnGraphics](#) (pdf)



Este artigo, publicado em 2022, corresponde ao culminar (actual) das publicações do candidato no contexto da área de conhecimento de Computação Gráfica. A revista científica “ACM Transactions on Graphics” é a publicação mais conceituada nesta área. Adicionalmente, os autores foram convidados a apresentar este trabalho na ACM SIGGRAPH’2022. A série de conferências SIGGRAPH (classificadas como CORE A\*) corresponde ao evento de excelência da área, contando com milhares de participantes e com a presença dos mais reputados especialistas da área.

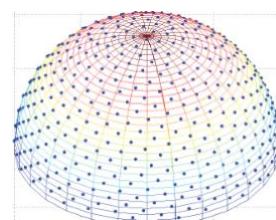
Este trabalho representa a alta qualidade atingida com a colaboração mantida desde há cerca de 30 anos com o grupo dos Professores Alan Chalmers, Kurt Debattista e Thomas Bashford-Rogers, originalmente na Universidade de Bristol, posteriormente na Universidade de Warwick, ambas no Reino Unido.

Neste artigo é apresentado um novo algoritmo baseado em conjuntos (*ensembles*) de Markov Chains que proporciona ganhos significativos de variância num conjunto alargado de cenas e é facilmente paralelizável.

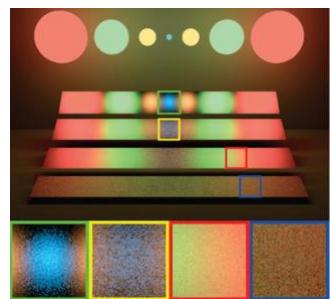
3. R. Marques, C. Bouville, M. Ribardière, L.P. Santos, K. Bouatouch;  
 “Spherical Fibonacci Point Sets for Illumination Integrals”;  
 Computer Graphics Forum, Volume 32(8), November, 2013  
 DOI: <http://dx.doi.org/10.1111/cgf.12190>  
 Ficheiro local: [2013-ComputerGraphicsForum](#) (pdf)

Este artigo (juntamente com o próximo) constitui o culminar da colaboração com o grupo do Professor Kadi Bouatouch, Université de Rennes I. O (agora) Doutor Ricardo Marques, na altura co-supervisionado pelo candidato e pelo Professor Bouatouch, propõe um novo padrão de amostragem de espaços semi-esféricos que, demonstravelmente, resulta numa melhor taxa de convergência em processos estocásticos de integração numérica, tal como a integração de Monte Carlo.

Este padrão de amostragem veio a ser adoptado por vários autores e foram, posteriormente, propostas extensões significativas, nomeadamente a possibilidade de gerar amostras sequencialmente sem conhecimento *a priori* do número total de amostras desejado.



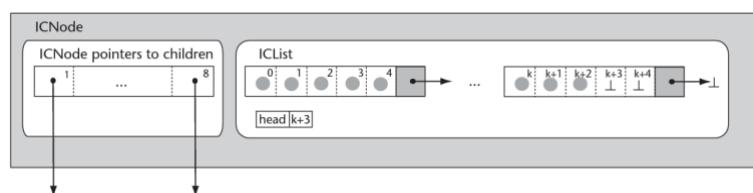
4. Marques, Ricardo; Bouville, Christian; Ribardiére, Mickael; Santos, Luís Paulo; Bouatouch, Kadi; “A Spherical Gaussian Framework for Bayesian Monte Carlo Rendering of Glossy Surfaces”; IEEE Transactions on Visualization and Computer Graphics, Volume 19(10), October, 2013  
DOI: <http://dx.doi.org/10.1109/TVCG.2013.79>  
Ficheiro local: [2013-IEEE-TVCG \(pdf\)](#)



Este artigo (juntamente com o anterior) constitui o culminar da colaboração com o grupo do Professor Kadi Bouatouch, Université de Rennes I. O (agora) Doutor Ricardo Marques, na altura co-supervisionado pelo candidato e pelo Professor Bouatouch, explorou um paradigma alternativo de integração numérica estocástica, proposto em 2003 e conhecido como Bayesian Monte Carlo. Os desafios são vários, tratando-se de um trabalho altamente exploratório, que demonstra ainda assim o potencial da abordagem. Publicado na IEEE Transactions on Visualization and Computer Graphics, tratava-se, à data, do primeiro trabalho do candidato publicado num fórum com tal notoriedade.

5. Debattista, Kurt and Dubla, Piotr and Santos, Luís Paulo and Chalmers, Alan; “Wait-Free Shared-Memory Irradiance Caching”; IEEE Computer Graphics and Applications, Volume 31(5), 2011  
DOI: <http://dx.doi.org/10.1109/MCG.2010.80>  
Ficheiro local: [2011-IEEE-ComputerGraphicsAndApplications \(pdf\)](#)

Tratando-se de um artigo de 2011 com um número reduzido de citações a selecção deste artigo pode parecer menos bem conseguida. No entanto, trata-se de uma excelente combinação de duas áreas do saber relevantes para o candidato: Computação Gráfica e Computação Paralela. É proposto um mecanismo de controlo de acesso a uma estrutura de dados partilhada entre vários fios de execução (*threads*) que é simultaneamente seguro e eficiente, reduzindo quer os custos inevitavelmente associados à actualização de dados partilhados, quer a replicação de computações que a não partilha de dados implicaria. A concepção da abordagem proposta nesta aplicação implicou quer o conhecimento de estruturas de dados *read/write* usadas na Computação Gráfica, quer o conhecimento de mecanismos de acesso de controlo e teoria do consenso. A abordagem proposta é simultaneamente segura, elegante e mais eficiente do que as alternativas mais comuns.





# Policy gradients using variational quantum circuits

André Sequeira<sup>1,2,3</sup> · Luis Paulo Santos<sup>1,2,3</sup> · Luis Soares Barbosa<sup>1,2,3</sup>

Received: 17 March 2022 / Accepted: 26 January 2023 / Published online: 11 April 2023  
© The Author(s) 2023

## Abstract

Variational quantum circuits are being used as versatile quantum machine learning models. Some empirical results exhibit an advantage in supervised and generative learning tasks. However, when applied to reinforcement learning, less is known. In this work, we considered a variational quantum circuit composed of a low-depth hardware-efficient ansatz as the parameterized policy of a reinforcement learning agent. We show that an  $\epsilon$ -approximation of the policy gradient can be obtained using a logarithmic number of samples concerning the total number of parameters. We empirically verify that such quantum models behave similarly to typical classical neural networks used in standard benchmarking environments and quantum control, using only a fraction of the parameters. Moreover, we study the barren plateau phenomenon in quantum policy gradients using the Fisher information matrix spectrum.

**Keywords** Quantum machine learning · Reinforcement learning · Policy gradients · Quantum control

## 1 Introduction

Reinforcement learning (RL) is responsible for many relevant developments in artificial intelligence (AI). Successes such as beating the world champion of Go (Silver et al. 2016) and solving numerous complex games without any human intervention (Schrittwieser et al. 2020) were relevant milestones in AI, providing optimal planning without supervision. RL is paramount in complex real-world problems such as self-driving vehicles (Kiran et al. 2021), automated trading (Liu et al. 2020; Mosavi et al. 2020), recommender

systems (Afsar et al. 2021), and quantum physics (Dalgaard et al. 2020). Recent advancements in RL are strongly associated with advances in deep learning (Goodfellow et al. 2016) since scaling to large state/action space environments is possible, as opposed to tabular RL (Sutton and Barto 2018).

Previous results suggest that RL agents obeying the rules of quantum mechanics can outperform classical RL agents (Dunjko et al. 2016, 2017; Paparo et al. 2014; Sequeira et al. 2021; Dunjko and Briegel 2018; Saggio et al. 2021). However, these suffer from the same scaling problem as classical tabular RL: they do not scale easily to real-world problems with large state-action spaces. Additionally, the lack of fault-tolerant quantum computers (Preskill 1997) further compromises the ability to handle problems of significant size.

Variational quantum circuits (VQCs) are a viable alternative since state-action pairs can be parameterized, enabling, at least in theory, a reduction in the circuit's complexity. Moreover, VQCs could enable shallow enough circuits to be confidently executed on current NISQ (*Noisy Intermediate Scale Quantum*) hardware (Preskill 2018) without resorting to typical brute force search over the state/action space as in the quantum tabular setting (Sequeira et al. 2021; Dunjko et al. 2016). Variational models are also referred to as approximately universal quantum neural networks (Farhi and Neven 2018; Schuld et al. 2021). Nevertheless, fundamental questions on

Luis Paulo Santos and Luis Soares Barbosa contributed equally to this work.

✉ André Sequeira  
andre.sequeira@inl.int

Luis Paulo Santos  
psantos@di.uminho.pt

Luis Soares Barbosa  
lsb@di.uminho.pt

<sup>1</sup> Department of Informatics, University of Minho, Braga, Portugal

<sup>2</sup> HASLab, INESC TEC, Braga, Portugal

<sup>3</sup> International Nanotechnology Laboratory (INL), Braga, Portugal

the expressivity and trainability of VQCs remain to be answered, especially from a perspective relevant to RL.

This paper proposes an RL agent's policy resorting to a shallow VQC and studies its effectiveness when embedded in the Monte-Carlo-based policy gradient algorithm REINFORCE (Williams 2004) throughout standard benchmarking environments. However, benchmarking variational algorithms for classical environments exhibit a trade of information between a quantum and a classical channel that incurs an overhead from encoding classical information into the quantum processor. Efficient encoding of real-world data constitutes a real bottleneck for NISQ devices, with the consequence of neglecting any potential quantum advantage (LaRose and Coyle 2020). In the case of a quantum agent-environment interface, the cost of data encoding can often be neglected, and there is room for potential quantum advantages from quantum data (Huang et al. 2021). In optimal quantum control, gate fidelity is improved by exploiting the full knowledge of the system's Hamiltonian (James 2021). However, such methods are only viable when the system's dynamics are known. Thus, applying variational quantum methods may indeed be relevant (Martín-Guerrero and Lamata 2021). In this setting, we considered a quantum RL agent that optimizes the gate fidelity in a model-free setting, learning directly from the interface with the noisy environment.

The main contributions of this paper are:

- Design of a variational softmax-policy using a shallow VQC similar to or outperforming long-term cumulative reward compared to a restricted class of classical neural networks used in a set of standard benchmarking environments and the problem of quantum state preparation, using a fraction of the number of trainable parameters.
- Demonstration of a logarithmic sample complexity concerning the number of parameters in gradient estimation.
- Empirical verification of different parameter initialization strategies for variational policy gradients.
- Study of the barren plateau phenomenon in quantum policy gradient optimization using the Fisher information matrix spectrum.

The rest of the paper is organized as follows. Section 2 reviews quantum variational RL's state-of-the-art. Section 3 summarizes the theory behind the classical policy gradient algorithm used in this work. Section 4 details each block of the proposed VQC and the associated quantum policy gradient algorithm. Section 4.5 explores trainability under gradient-based optimization using quantum hardware and its corresponding sample complexity. Section 5 presents the performance of the quantum variational algorithm in

simulated benchmarking environments. Section 6 analyzes the number of parameters trained and the Fisher information spectrum associated with the classical/quantum policy gradient. Section 7 closes the paper with some concluding remarks and suggestions for future work.

## 2 Related work

Despite numerous publications focusing on quantum machine learning (QML), the literature on variational methods applied to RL remains scarce. Most results to date focus on value-based function approximation rather than policy-based. Chen et al. (2020) use VQCs as quantum value function approximators for discrete state spaces, and, in Lockwood and Si (2020), the authors generalize the former result to continuous state spaces. Lockwood and Si (2021) show that simple VQC-inspired Q-networks (i.e., state-action value approximators) based on double deep Q-learning are not adequate for the Atari games, Pong and Breakout. Sanches et al. (2021) proposed a hybrid quantum-classical policy-based algorithm to solve real-world problems like vehicle routing. In Wu et al. (2021), the authors proposed a variational actor-critic agent, which is the only work so far operating on the quantum-quantum context of QML (Aïmeur et al. 2006), i.e., a quantum agent acting upon a quantum environment. The authors suggest that the variational method could solve quantum control problems. Jerbi et al. (2021) propose a novel quantum variational policy-based algorithm achieving better performance than previous value-based methods in a set of standard benchmarking environments. Their architecture consists of repeated angle-encoding to increase the expressivity of the variational model, i.e., increasing the number of functions of the input state that the model can represent (Schuld et al. 2021). Compared with Jerbi et al. (2021), our work shows that a simpler variational architecture composed of a shallow ansatz, consisting of a two-qubit entangling gate and two single-qubit gates (Bharti et al. 2022) with a single encoding layer can be considered for standard benchmarking environments. Variational policies can be devised with decreased depth and fewer trainable parameters. The type of functions our circuit can represent is substantially smaller when compared to Jerbi et al. (2021). However, simpler classes of policies may be beneficial in the language of generalization and overfitting. Furthermore, compared to Jerbi et al. (2021), this work considers a more trivial set of observables for the measurement of the quantum circuit, leading to fewer shots necessary to estimate the agent's policy and respective policy gradient.

### 3 Policy gradients

Policy gradient methods try to learn a parameterized policy  $\pi(a|s, \theta) = \mathbb{P}\{a_t = a|s_t = s, \theta_t = \theta\}$ , where  $\theta \in \mathbb{R}^k$  is the parameter vector of size  $k$ ,  $s$  and  $a$  are the state and action, respectively, and  $t$  is the time instant that can optimally select actions without resorting to a value function. These methods try to maximize a performance measure  $J(\theta)$ , performing gradient ascent on  $J(\theta)$

$$\theta_{i+1} = \theta_i + \eta \nabla_{\theta_i} J(\theta_i) \quad (1)$$

where  $\eta$  is the learning rate. Provided that the action space is discrete and relatively small, then the most prominent way of balancing exploration and exploitation is by sampling an action from a *Softmax-Policy*, also known as Neural Policy (Agarwal et al. 2019):

$$\pi(a|s, \theta) = \frac{e^{h(s, a, \theta)}}{\sum_{b \in A} e^{h(s, b, \theta)}} \quad (2)$$

where  $h(s, a, \theta) \in \mathbb{R}$  is a numerical preference for each state-action pair and  $A$  is the action set. For legibility,  $A$  will be omitted whenever a policy similar to Eq. 2 is presented. The policy gradient theorem (Sutton et al. 1999) states that the gradient of the objective function can be written as a function of the policy itself. In general, the Monte-Carlo policy gradient known as REINFORCE (Williams 2004) computes the gradient of samples obtained from  $N$  trajectories of length  $T$ , also known as the *horizon* under the parameterized policy, as in Eq. 3.

$$\nabla_{\theta} J(\theta) = \frac{1}{N} \sum_{i=0}^{N-1} \sum_{t=0}^{T-1} G_t(\tau_i) \nabla_{\theta} \log \pi(a_{t_i}|s_{t_i}, \theta) \quad (3)$$

where  $G_t(\tau)$  is the  $\gamma$ -discounted cumulative reward per time step, known as the *return* (see Eq. 5) derived from trajectory's return  $G(\tau)$  (see Eq. 4).

$$G(\tau) = \sum_{t=0}^{T-1} \gamma^t r_{t+1} \quad (4)$$

$$G_t(\tau) = \sum_{t'=0}^{T-t-1} \gamma^{t'} r_{t'+t} \quad (5)$$

A known limitation of the REINFORCE algorithm is due to Monte Carlo estimates. Stochastically sampling the trajectories results in gradient estimators with high variance, which deteriorate the performance as the environment's complexity increases (Greensmith et al. 2004). The REINFORCE estimator can be improved by leveraging a control variate known as *baseline*  $b(s_t)$ , without increasing the number of samples  $N$ . Baselines are subtracted from the return such that the optimization landscape

becomes smooth. The REINFORCE with baseline gradient estimator is represented in Eq. 6, and the complete algorithm is presented in Algorithm 1.

$$\nabla_{\theta} J(\theta) = \frac{1}{N} \sum_{i=0}^{N-1} \sum_{t=0}^{T-1} (G_t(\tau_i) - b(s_{t_i})) \nabla_{\theta} \log \pi(a_{t_i}|s_{t_i}, \theta) \quad (6)$$

For the benchmarking environments in Section 5, the average return was used as a baseline, calculated as in Eq. 7.

$$b(s_t) = \frac{1}{N} \sum_{i=0}^{N-1} G_t(\tau_i) \quad (7)$$

---

**Require:**  $\theta \in \mathbb{R}^k$ , learning rate  $\eta$ , horizon  $T$   
**while** True **do**  
  **for**  $i = 0 \dots N - 1$  **do**  
    Following  $\pi_{\theta}$ , generate trajectory of the form  
     $\tau_i = \{(s_0, a_0, r_0), \dots, (s_{T-1}, a_{T-1}, r_{T-1})\}$   
  **end for**  
  Compute gradient with baseline as in Equation (6)  
  update parameters via gradient ascent  $\theta = \theta + \eta \nabla_{\theta} J(\theta)$   
**end while**

---

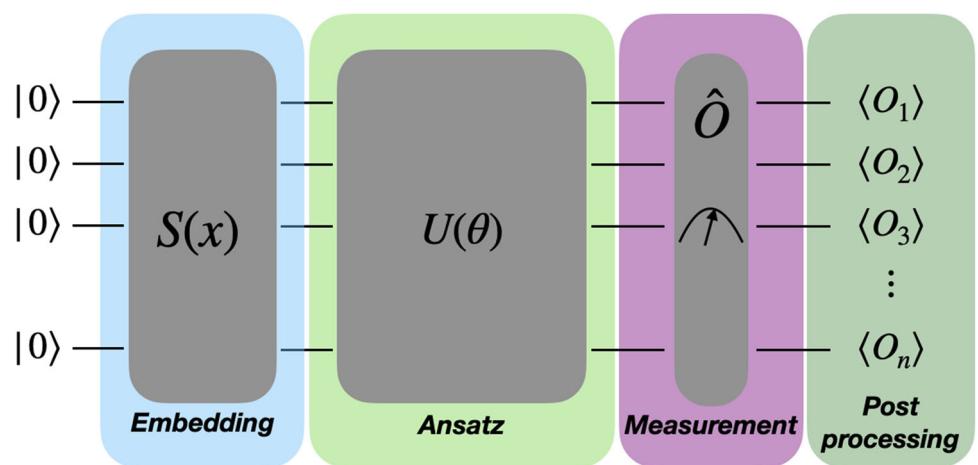
**Algorithm 1** REINFORCE with baseline.

### 4 Quantum policy gradients

This section details the proposed VQC-based policy gradient. Numerical preferences  $h(s, a, \theta) \in \mathbb{R}$  are the output of measurements in a given parameterized quantum circuit. The result can be represented as the expectation value of a given observable or the probability of measuring a basis state. We resort to the former since it allows for more compact representations of objective functions (Bharti et al. 2021). Additionally, the type of ansatz used by the proposed VQC implies that  $\theta \in \mathbb{R}^k$  is a high dimensional vector corresponding to the angles of arbitrary single-qubit rotations.

VQCs are composed of four main building blocks, as represented in Fig. 1. Initially, a state preparation routine or *embedding*,  $S$ , encodes data points into the quantum system. Next, a unitary  $U(\theta)$  maps the data into higher dimensions of the Hilbert space. Such a parameterized model corresponds to linear methods in quantum feature spaces. Expectation values returned from a measurement

**Fig. 1** Building blocks of variational quantum circuits



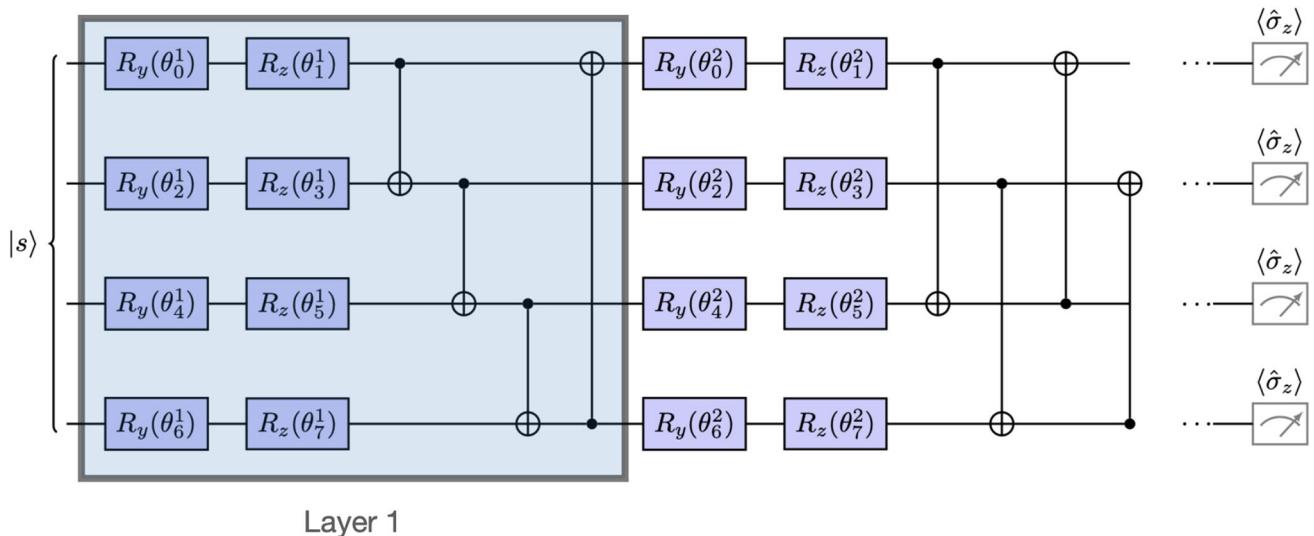
scheme are finally post-processed into the quantum neural policy. A careful analysis of each block of Fig. 1 follows. Moreover, the sample complexity of estimating the quantum policy gradient is analyzed in Section 4.5.

#### 4.1 Embedding

Unlike classical algorithms, the state-preparation routine is a crucial step for any variational quantum algorithm. There are numerous ways of encoding classical data into a quantum processor (Schuld and Petruccione 2018). Angle encoding (LaRose and Coyle 2020) is used to allow continuous-state spaces. Arbitrary Pauli rotations  $\sigma \in \{\sigma_x, \sigma_y, \sigma_z\}$  can encode a single feature per qubit. Hereby, given an agent's state  $s$  with  $n$  features,  $s = \{s_0, s_2, \dots, s_{n-1}\}$ ,  $\sigma_x$  rotations are used, requiring  $n$  qubits to encode  $|s\rangle$ , as indicated by Eq. 8.

$$|s\rangle = \bigotimes_{i=0}^{n-1} e^{-j\sigma_x s_i} |b_i\rangle \quad (8)$$

where  $|b_i\rangle$  refers to the  $i$ th qubit of an  $n$ -qubit register initially in state  $|0^n\rangle$  (represented w.l.g as  $|0\rangle$  from now on). Each feature needs to be normalized such that  $s_i \in [-\pi, \pi]$ . Since the range of each feature is usually unknown, this work resorts to normalization based on the  $L_\infty$  norm. The main advantage of angle encoding lies in the simplicity of generating the encoding, given the composition of solely  $n$  single-qubit gates, thus giving rise to a circuit of depth 1. In contrast, the main disadvantage is the linear dependence



**Fig. 2** Hardware-efficient ansatz for RL based on single-qubit  $R_y$ ,  $R_z$  rotation gates

between the number of qubits and the number of features characterizing the agent's state and the poor representational power, at least in principle (Schuld 2021).

## 4.2 Parameterized model

To the best of the authors' knowledge, no *problem-inspired* ansatz exploiting the physics behind the problem is known in RL applications. This can be explained by the difficulty of expressing and training RL agent's policies as Hamiltonian-based evolution models (Bharti et al. 2021). Moreover, since the goal is to design a NISQ ansatz to capture the agent's optimal policy in different environments, this work uses a parameterized model from the family commonly referred to as *hardware-efficient* ansatz (Bharti et al. 2021). Such models behave similarly to a classical feed-forward neural network. The main advantage of this family of ansatze is its versatility, accommodating encoding symmetries and bringing correlated qubits closer for depth reduction (Cerezo et al. 2021). The ansatz consists of an alternating-layered architecture composed of single-qubit gates followed by a cascade of entangling gates as pictured in Fig. 2.

A single layer is composed of two single-qubit  $\sigma_y, \sigma_z$  rotation gates per qubit, followed by a cascade of entangling gates, such that features are correlated in a highly entangled state. The ansatz includes  $2n$  single-qubit rotation gates per layer, each gate parameterized by a given angle. Therefore, there are  $2nL$  trainable parameters for  $L$  layers. The entangling gates follow a pattern that changes over the number of layers, inspired by the circuit-centric classifier design (Schuld et al. 2021). The pattern follows a modular arithmetic CNOT[i, (i + l) mod n] where  $i \in [1, \dots, n]$  and  $l \in [1, \dots, L]$  indexes the layers. Increasing the number of layers increases the correlation between features and expressivity.

## 4.3 Measurement

An arbitrary state  $|\psi\rangle \in \mathbb{C}^{2^n}$  is represented by an arbitrary superposition over the basis states, as in Eq. 9.

$$|\psi\rangle = \sum_{i=0}^{2^n-1} c_i |\psi_i\rangle \quad (9)$$

Measuring the state  $|\psi\rangle$  in the computational basis ( $\sigma_z$  basis) collapses the superposition into one of the basis states  $|\psi_i\rangle$  with probability  $|c_i|^2$ , as given by the Born rule (Nielsen and Chuang 2011). In general, the expectation value of some observable  $\hat{O}$  is given by the summation of

each possible outcome, i.e., the eigenvalue  $\lambda_i$  weighted by its respective probability  $p_i = |c_i|^2$  as in Eq. 10.

$$\langle \hat{O} \rangle = \langle \psi | \hat{O} | \psi \rangle = \sum_{i=0}^{2^n-1} \lambda_i p_i \quad (10)$$

Let  $\hat{O}$  be the single-qubit  $\sigma_z^i$  measurement, applied to the  $i$ th-qubit. Given that the  $\sigma_z$  eigenvalues are  $\{-1, 1\}$ , the expectation value  $\langle \sigma_z^i \rangle$  can be obtained by the probability  $p_0$  of the qubit being in the state  $|0\rangle$  as  $\langle \sigma_z^i \rangle = 2p_0 - 1$ . Notice that in practice,  $p_0$  needs to be estimated from several circuit repetitions to obtain an accurate estimate of the expectation value.

Let the state  $|\psi\rangle$  be the quantum state obtained from the encoding of an agent's state via  $S(s)$ , and the parameterized block  $U(\theta)$ , as in Sections 4.1 and 4.2 respectively. Let  $\langle \sigma_z^i \rangle$  be the quantum analogue of the numerical preference for action  $i$ , which we represent by  $\langle a_i \rangle$  for clarity. Its expectation can be formally described by Eq. 11.

$$\langle a_i \rangle_\theta = \langle 0 | S(s)^\dagger U(\theta)^\dagger \sigma_z^i U(\theta) S(s) | 0 \rangle \quad (11)$$

For a policy with  $|A|$  possible actions, each  $\sigma_z$  measurement corresponds to the numerical preference of each action. Thus,  $|A|$  single-qubit estimated expectation values are needed. If the number of features in the agent's state is larger than the number of actions, the single-qubit measurements occur only on a subset of qubits. Such measurement scheme is qubit-efficient (Schuld and Petruccione 2018). Figure 3 represents the full VQC for an environment with four feature states and four actions with three parameterized layers.

## 4.4 Classical post-processing

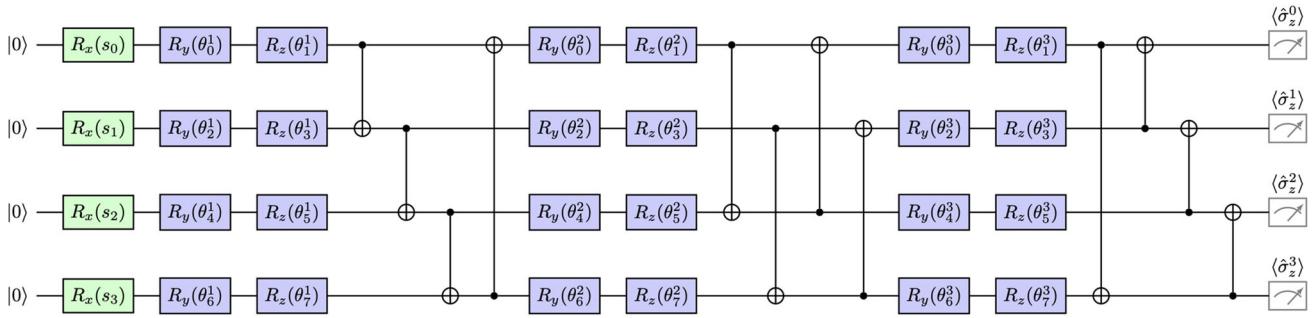
Measurement outcomes representing numerical preferences  $h(s, a, \theta) = \langle a \rangle_\theta$  are classically post-processed to convert the estimated expectation values to the final quantum neural policy, as given by Eq. 12.

$$\pi(a | s, \theta) = \frac{e^{\langle a \rangle_\theta}}{\sum_b e^{\langle b \rangle_\theta}} \quad (12)$$

Equation 12 imposes an upper bound on the greediness of  $\pi$ . It will always allow for exploratory behavior, which can negatively impact the performance of RL agents, especially in deterministic environments. As an example, consider a 2-action environment with

$$\pi = [\pi(a_0 | s, \theta), \pi(a_1 | s, \theta)]$$

The entries of  $\pi$  are given by Eq. 12 and the actions' estimated expectation values  $[\langle a_0 \rangle_\theta, \langle a_1 \rangle_\theta]$ . As these are bounded as  $\langle \sigma_z \rangle \in [-1, 1]$ , the maximum difference



**Fig. 3** Variational quantum circuit for policy-based RL with three parameterized layers

between action preferences occurs when the estimated vector is [ $\langle a_0 \rangle_\theta = -1, \langle a_1 \rangle_\theta = 1$ ]. The corresponding softmax normalized vector is:

$$\pi_a = [\pi(a_0 | s, \theta), \pi(a_1 | s, \theta)] = [0.88, 0.12]$$

In this case, the policy always has a  $\sim 0.1$  probability of selecting the worst action; the same *rationale* applies to larger action sets. Thus, a trainable parameter  $\beta$  is added to the quantum neural policy as in Eq. 13:

$$\pi(a|s, \theta) = \frac{e^{\beta \langle a \rangle_\theta}}{\sum_b e^{\beta \langle b \rangle_\theta}} \quad (13)$$

$\beta$  has the effect of scaling the output values from the quantum circuit measurements, resembling an energy-based model. Instead of decreasing  $\beta$  over time, we treat it as a hyperparameter to be tuned along with  $\theta$ . The optimization sets  $\beta$ , assuring convergence towards the optimal policy.

#### 4.5 Gradient estimation

This section develops upper bounds on both the number of samples and the number of circuit evaluations necessary to obtain an  $\epsilon$ -approximation of the policy gradient, as given by Eq. 3, restated here for completion:

$$\nabla_\theta J(\theta) = \frac{1}{N} \sum_{i=0}^{N-1} \sum_{t=0}^{T-1} G_t(\tau_i) \nabla_\theta \log \pi(a_{t_i} | s_{t_i}, \theta)$$

The gradient  $\nabla_\theta J(\theta)$  can be estimated using the same quantum device that computes expectations  $\langle a_i \rangle_\theta$ , via parameter-shift rules (Schuld et al. 2019). These rules require the policy gradient to be framed as a function of gradients of observables, as given by Eq. 14.

$$\nabla_\theta \log \pi(a|s, \theta) = \beta \left( \nabla_\theta \langle a \rangle_\theta - \sum_b \pi(b|s, \theta) \nabla_\theta \langle b \rangle_\theta \right) \quad (14)$$

By combining Eqs. 3 and 14, the quantum policy gradient estimator is given by Eq. 15:

$$\begin{aligned} \nabla_\theta J(\theta) = & \frac{1}{N} \sum_{i=0}^{N-1} \sum_{t=0}^{T-1} G_t(\tau_i) \beta \left( \nabla_\theta \langle a_{t_i} \rangle_\theta \right. \\ & \left. - \sum_{b_{t_i}} \pi(b_{t_i} | s_{t_i}, \theta) \nabla_\theta \langle b_{t_i} \rangle_\theta \right) \end{aligned} \quad (15)$$

The number of samples associated with Eq. 15 is defined as the number of visited states. Since there are  $N$  trajectories (sequences of actions,  $\tau_i$ ), each visiting  $T$  states, the total number of samples is  $\mathcal{O}(NT)$ .

Lemma 4.1 provides an upper bound for  $N$  such that the policy gradient is  $\epsilon_V$ -approximated with probability  $1 - \delta_V$ .

**Lemma 4.1** ( $\epsilon_V$ -approximation of the policy-gradient) *lemmasample Let  $\theta \in \mathbb{R}^k$ ,  $k$  being the number of parameters,  $R_{\max}$  be the maximum possible reward in any time step,  $T$  the horizon, and  $\nabla_\theta J(\theta)$  the expected policy gradient. The policy gradient,  $\hat{\nabla}_\theta J(\theta)$ , can be  $\epsilon_V$ -approximated, with probability  $1 - \delta_V$*

$$|\hat{\nabla}_\theta J(\theta) - \nabla_\theta J(\theta)| \leq \epsilon_V \quad (16)$$

using a number of samples given by

$$NT \approx \mathcal{O} \left( \frac{8\beta^2 R_{\max}^2 T^3}{\epsilon_V^2 (\gamma - 1)^4} \log \left( \frac{2k}{\delta_V} \right) \right) \quad (17)$$

The most relevant insight drawn from Lemma 4.1 is that it establishes that for obtaining an  $\epsilon_V$ -approximated policy gradient, the algorithm needs a number of samples that grows logarithmically with the total number of parameters. The proof of Lemma 4.1 is presented in detail in Appendix A.1.

Gradient-based optimization can be performed using the same quantum device that computes expectations  $\langle a_i \rangle_\theta$ , via parameter-shift rules (Sweke et al. 2020; Schuld et al. 2019), which compute the gradient of an observable w.r.t a

single variational parameter concerning rotation angles of quantum gates. Parameter-shift rules are given by Eq. 18:

$$\nabla_{\theta_i} \langle a_i \rangle_{\theta} = \frac{1}{2} \left[ \langle a_i \rangle_{\theta+\frac{\pi}{2}} - \langle a_i \rangle_{\theta-\frac{\pi}{2}} \right] \quad (18)$$

The gradient's accuracy depends on the expectation values,  $\langle a \rangle_{\theta}$ . These are estimated for each sample and action using several repetitions of the quantum circuit or shots. Lemma 4.2 establishes an upper bound on the total number of shots required to reach an  $\epsilon_{\langle \rangle}$ -approximated policy gradient, with probability  $1 - \delta_{\langle \rangle}$ .

**Lemma 4.2** (Total number of quantum circuit evaluations) *lemmaquery Let  $\theta \in \mathbb{R}^k$ ,  $\mathcal{O}(NT)$  be the sample complexity given by Lemma 4.1, and  $|A|$  the number of available actions. With probability  $1 - \delta_{\langle \rangle}$  and approximation error  $\epsilon_{\langle \rangle}$ , the quantum policy gradient algorithm requires a number of shots given by*

$$\mathcal{O} \left( \frac{|A|NT}{\epsilon_{\langle \rangle}^2} \log \left( \frac{2k}{\delta_{\langle \rangle}} \right) \right) \quad (19)$$

Similarly to Lemma 4.1, it is shown that the accuracy of the policy gradient, as a function on the total number of shots, grows logarithmically with the total number of parameters. The proof of Lemma 4.2 is presented in detail in Appendix A.2.

## 5 Performance in simulated environments

This section examines the performance of the proposed quantum policy gradient through standard benchmarking environments from the OpenAI Gym library (Brockman et al. 2016). Moreover, the quantum policy gradient was also tested in a handcrafted quantum control environment. In this setting, a quantum agent was designed to learn to prepare the state  $|1\rangle$  with high fidelity, starting from the ground state  $|0\rangle$ . The empirical reward over the number of episodes was used to discern the performance of both classical and quantum models. The best-performing classical neural network was selected from a restrictive set of networks composed of at most two hidden linear layers. All quantum circuits were built using the Pennylane library (Bergholm et al. 2020) and trained using the PyTorch automatic differentiation backend (Paszke et al. 2017) to be directly compared with classical models built with the same library. All training instances used the most common classical optimizer, ADAM (Kingma and Ba 2017).

### 5.1 Numerical experiments

The CartPole-v0 and Acrobot-v1 environments were selected as classic benchmarks. They have a continuous state space with a relatively small feature space (2 to 6 features) and discrete action space (2 to 3 possible actions). The reward function is similar to each environment. In Cartpole, the agent receives a reward of +1 at every time step. The more time the agent keeps the pole from falling, the more reward it gets. In Acrobot, the agent receives a -1 reward at every time step and reward 0 once it gets to the goal state. Thus, Acrobot will be harder to master since, for the Cartpole, every action has an immediate effect as opposed to Acrobot.

In the quantum control environment of state preparation, which we refer to as QControl on this point onward, for simplicity, the mapping  $|0\rangle \mapsto |1\rangle$  can be characterized by a time-dependent Hamiltonian  $H(t)$  of the form of Eq. 20 describing the quantum environment as in Zhang et al. (2019).

$$H(t) = 4J(t)\sigma_z + h\sigma_x \quad (20)$$

where  $h$  represents the single-qubit energy gap between tunable control fields, considered a constant energy unit.  $J(t)$  represents the dynamical pulses controlled by the RL quantum agent in a model-free setting. The learning procedure defines a fixed number of steps  $N = 10$ , from which the RL agent must be able to create the desired quantum state. The quantum environment prepares the state associated with the time step  $t + 1$ , given the gate-based Hamiltonian at time step  $t$ ,  $U(t)$ :

$$|\psi_{t+1}\rangle = U(t)|\psi\rangle \quad (21)$$

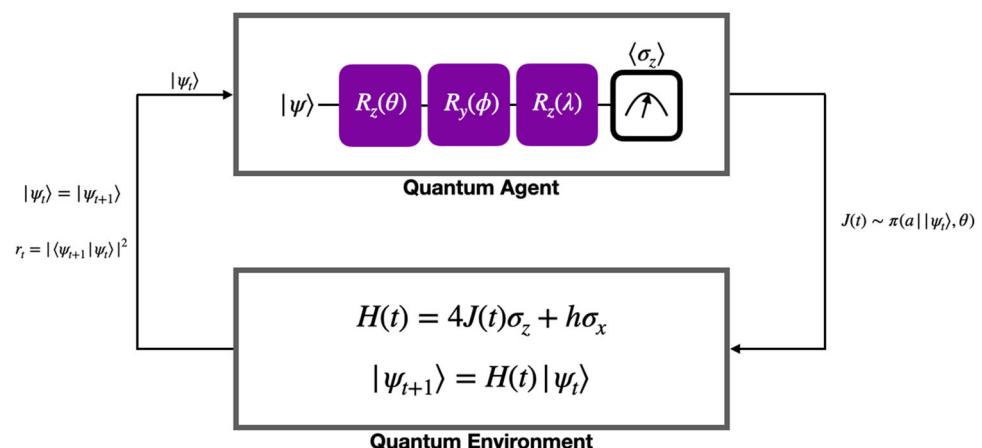
The reward function is naturally represented as the fidelity between the target state  $|\psi_T\rangle = |1\rangle$  and the prepared state  $|\psi_t\rangle$  naturally serves as the reward  $r_t$  for the agent at time step  $t$ , as in Eq. 22.

$$r_t = |\langle \psi_t | \psi_T \rangle|^2 \quad (22)$$

Using the policy gradient algorithm of Section 4, the goal is to learn how to maximize fidelity. Figure 4 depicts the agent-environment interface.

Each sequence of  $N$  pulses corresponds to an episode. The quantum agent should learn the optimal pulse sequence that maps to the state with maximum fidelity as the number of episodes increases. The quantum variational architecture selected was the same as described in Section 4. In this setting, the main difference is the lack of encoding. The quantum agent receives the quantum state from the corresponding time-step Hamiltonian applied at each time

**Fig. 4** Agent-environment interface for quantum control



step. However, since the environment is simulated, the qubit is prepared in the state of time step  $t$  and then fed to the variational quantum policy. In this setting, it is considered the binary action-space  $A = [0, 1]$  (apply pulse  $A = 1$  or not,  $A = 0$ ). A sequence of  $N$  actions corresponds to  $N$  pulses. A performance comparison is made relative to classical policy gradients. In this case, the corresponding state vector associated with the qubit was explicitly encoded at each time step, considering both real and imaginary components. All environment specifications are presented in Table 1.

Several neural network architectures were tested for the CartPole-v0 and Acrobot-v1 environments. However, the structure is the same. Every neural network is composed of fully connected layers using a rectified linear unit (ReLU) activation function in every neuron. The output layer is the only layer that does not have ReLU activation. The depth, the total number of trainable parameters, and the existence of dropout differs from network to network. All the networks using dropout have a probability equal to 0.2. Every network was trained with an ADAM optimizer with an experimentally fine-tuned learning rate of 0.01. Figure 5a and b illustrate the average reward for different classical network configurations for the benchmarking environments. The results show that a fully connected neural network with a single layer of 128 and 32 neurons performs reasonably

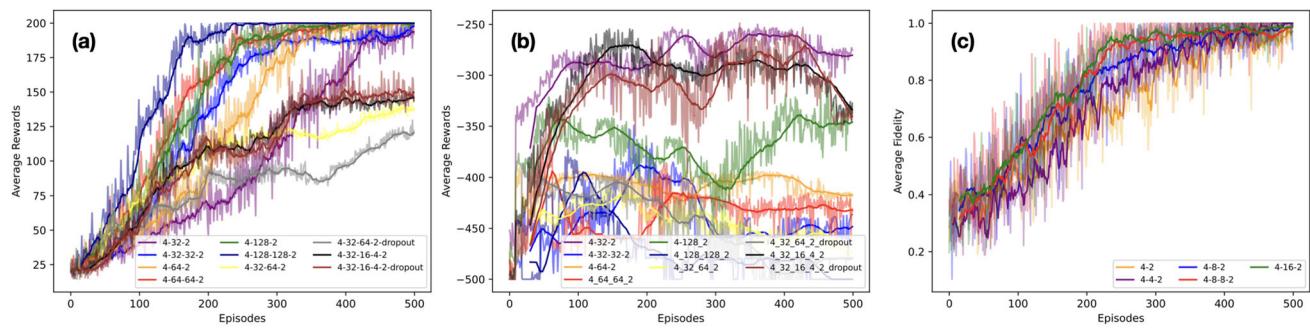
better than similar architectures for the CartPole-v0 and Acrobot-v1 environments, respectively.

In the QControl environment, eight different neural networks were tested with a single hidden layer. Since the optimal neural network for this problem is still an open question, to the best of the author's knowledge, it was decided to successively increase the size of the network until it solves the task of comparing the minimum viable network with the VQC. For this set of classical architectures, the neural network with a single layer of 16 neurons was chosen since it achieves the best average fidelity as the minimum viable network solving the problem, as illustrated in Fig. 5c.

The second step compares the performance of the quantum neural policy of Section 4 against the aforementioned classical architecture. Increasing the number of layers in the parameterized quantum model would perhaps increase the expressivity of the model (Schuld 2021). At the same time, increasing the number of layers leads to more complex optimization tasks, given that more parameters need to be optimized. For some variational architectures, there is a threshold for expressivity in terms of the number of layers (Sim et al. 2019). We encountered precisely this in practice. For Cartpole, the expressivity of the quantum neural policy saturates after three layers, and for the Acrobot, after four layers. From there on, the agent's performance deteriorated rather than improved. For the QControl environment, the

**Table 1** Description of the environments (#F, number of features; #A, number of actions; Max #s, maximum steps)

Environment	#F	#A	Reward (per step)	Max #s (per episode)	Terminal states
CartPole-v0	4	2	1	200	Out of bounds or reward 200 or below horizontal line
Acrobot-v1	6	3	-1	500	500 steps
QControl	4	2	$ ⟨ψ_t  ψ⟩ ^2$	10	$ ⟨ψ_t  ψ⟩ ^2 \leq 10^{-4}$ or 10 steps



**Fig. 5** Different classical neural network architectures used in the three simulated environments. Panels **a**, **b**, and **c** represent different architectures for the Cartpole, Acrobot, and QControl environments, respectively. Each label indicates the respective network structure and

if it uses dropout. Each label represents the total number of neurons in each input, hidden, and output layer. For example, 4 – 4 – 4 has input, hidden, and output layers with four neurons each

classical NN was compared with a simplified version of the variational softmax policy. In this case, it was considered a VQC with the most general gate with three parameters that can approximately prepare every single-qubit state. The observables for the numerical action preference are the opposite sign computational basis measurement, i.e.,  $[\langle \sigma_z \rangle, -\langle \sigma_z \rangle]$ . In every environment, the model's learning rate was fine-tuned by trial and error as opposed to  $\beta$ , which was randomly initialized. The optimal configuration for the learning rate, number of layers, and batch size used to compare are presented in Table 2.

Figure 6a, b, and c compare the average cumulative reward through several episodes for quantum and classical neural policies for the Cartpole, Acrobot, and QControl environments, respectively. A running mean was plotted to smooth the reward curves since the policy and environments are noisy. Figure 6c also plots the respective control trajectory obtained by the variational quantum policy.

One can conclude that the quantum and classical neural policies perform similarly in every environment. In the QControl environment, the classical policy achieves a slightly greater cumulative reward. Nonetheless, there is clear evidence that the quantum-inspired policy needs fewer interactions with the environment to converge to near-optimal behavior. Moreover, the total number of

trainable parameters for the quantum and classical models is summarized in the Table 3. The input layer of a classical neural network is related to the number of qubits in a quantum circuit. Furthermore, we take the number of layers in the VQC as the number of hidden layers in a classical neural network. Given that the quantum circuit is unitary, the number of neurons in a quantum neural network is constant, i.e., equal to the system's number of qubits. Thus, one can conclude that the quantum policy has similar or even outperforming behavior compared to the classical policy with an extremely reduced total number of trainable parameters.

## 5.2 The effect of initialization

The parameters' initialization strategy can dramatically improve the convergence of a machine learning algorithm. Random initialization is often used to break the symmetry between different neurons (Goodfellow et al. 2016). However, if the parameters are arbitrarily large, the activation function may saturate, difficulting the learning task. Therefore, parameters are often drawn from specific distributions. For instance, the Glorot (Glorot and Bengio 2010) initialization strategy is among the most commonly used to balance initialization and regularization (Goodfellow et al. 2016).

In quantum machine learning models, the problem persists. However, it was verified experimentally that the Glorot initialization has a slight advantage compared to other strategies. The empirical results reported in Section 5.1 were obtained using such a strategy.

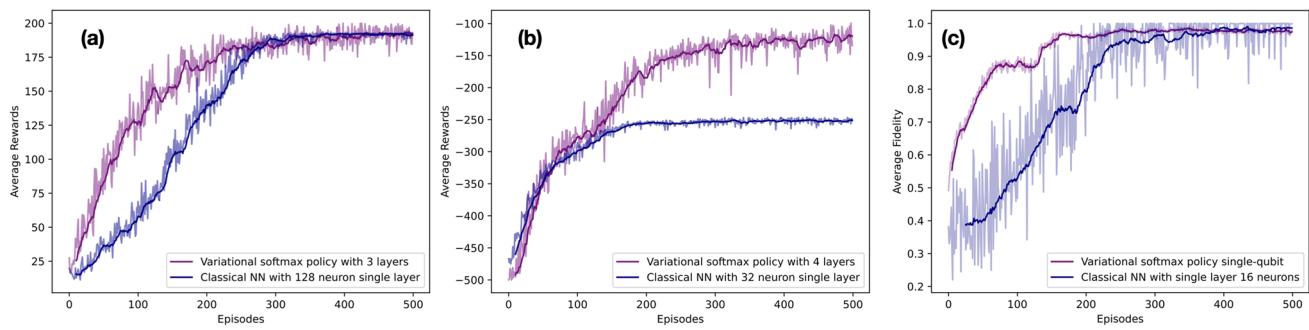
The Glorot strategy samples the parameters of the network from a normal distribution  $\mathcal{N}(0, std^2)$  with standard deviation given by Eq. 23:

$$std = \text{gain} * \sqrt{\frac{6}{n_{in} + n_{out}}} \quad (23)$$

where gain is a constant multiplicative factor.  $n_{in}$  and  $n_{out}$  are the number of inputs and outputs of a layer, respectively.

**Table 2** Specification for hyperparameter, number of layers, and batch size used for the classical and quantum neural policies in the three simulated environments

Environment	Policy	Learning rate	#Layers	Batch size
CartPole-v0	Quantum	0.1	3	10
CartPole-v0	Classical	0.01	—	10
Acrobot-v1	Quantum	0.1	4	10
Acrobot-v1	Classical	0.01	—	10
QControl	Quantum	0.01	1	10
QControl	Classical	0.01	—	10



**Fig. 6** Average cumulative reward. Comparison between the variational softmax policy and the respective classical NN. Panels **a**, **b**, and **c** represent the average reward comparison for the Cartpole, Acrobot, and QControl environments, respectively

It was devised to initialize all layers with approximately the same activation and gradient variance, assuming that the neural network does not have nonlinear activations, being thus reducible to a chain of matrix multiplications. The latter assumption motivates this strategy in quantum learning models since they are composed of unitary layers without nonlinearities. The only nonlinearity is introduced by the measurement (Nielsen and Chuang 2011).

Figure 7a, b, and c plot the average reward obtained by the quantum agent in the CartPole, Acrobot, and QControl environments, respectively, following the most common initialization strategies. Glorot initialization has a slightly better performance and stability. Moreover, it is verified empirically that for policy gradients, initialization from normal distributions generates better results for the classic environments compared to uniform distributions, as reported in Zhang et al. (2022) for standard machine learning cost functions. However, in the QControl task was not observed the same behavior since uniform sampling  $U(-1, 1)$  achieves similar performance than  $N(0, 1)$ .

## 6 Quantum enhancements

In this section, further steps are taken toward studying the possible advantages of quantum RL agents following two different strategies:

**Table 3** Number of parameters trained for both environments (*Env*, environment; *I*, input layer; *O*, output layer; #*N*, neurons; #*R*, rotations per qubit; #*P*, parameters)

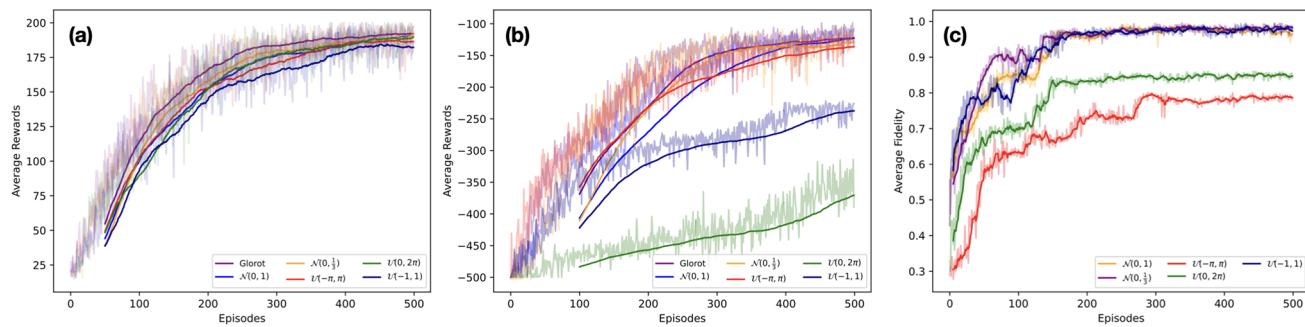
Env	Policy	I	O	#N	#R	$\beta$	#P
CartPole-v0	Quantum	4	2	—	2	Yes	25
CartPole-v0	Classical	4	2	128	—	No	768
Acrobot-v1	Quantum	6	3	—	2	Yes	33
Acrobot-v1	Classical	6	3	32	—	No	288
QControl	Quantum	1	1	—	3	Yes	3
QControl	Classical	4	2	16	—	No	96

- **Parameter count** — Comparison between quantum and classical agents regarding the number of parameters trained. It is unclear whether this is a robust approach to quantify advantage, given that the number of parameters alone can be misleading. For example, the function  $\sin(\theta)$  has a single parameter and is more complex than polynomial  $ax^3 + bx^2 + cx + d$ . However, having smaller networks could enable solutions for more significant problems at a smaller cost. Even though only parameter-shift rules are allowed on real quantum hardware, it enables a lower cost on memory than backpropagation. Perhaps the training difference may be negligible from a tradeoff between memory and time consumption for large enough problems. As reported in Table 3, a massive reduction in the number of parameters in the quantum neural network compared with the classical counterpart for all three simulated environments.
- **Fisher information** — The Fisher information matrix spectrum is related to the effect of barren plateaus in the optimization surface itself. Studying the properties of the matrix eigenvalues should help to explain the hardness of training.

The Fisher information (Ly et al. 2017) is crucial both in computation and statistics as a measure of the amount of information in a random variable  $X$  in a statistical model parameterized by  $\theta$ . Its most general form amounts to the negative Hessian of the log-likelihood. Suppose a datapoint  $x$  sampled i.i.d from  $p(x|\theta)$  where  $\theta \in \mathbb{R}^k$ . Since the Hessian reveals information about the curvature of a function, the Fisher information matrix (see Eq. 24) captures the sensitivity concerning changes in the parameter space, i.e., changes in the curvature of the loss function.

$$F(\theta) = \mathbb{E}_{x \sim p} \left[ \nabla_{\theta} \log p(x|\theta) \nabla_{\theta} \log p(x|\theta)^{\top} \right] \in \mathbb{R}^{k \times k} \quad (24)$$

The Fisher information matrix is computationally demanding to obtain. Thus, the empirical Fisher



**Fig. 7** Normal and uniform distributions used to initialize the parameters of the variational softmax policy. Panels **a**, **b**, and **c** represent the average reward comparison for the Cartpole, Acrobot, and QControl environments, respectively

information matrix is usually used in practice and can be computed as in Eq. 25:

$$F(\theta) = \frac{1}{T} \sum_{i=1}^T \nabla_\theta \log p(x_i|\theta) \nabla_\theta \log p(x_i|\theta)^\top \quad (25)$$

Equation 25 captures the curvature of the score function at all parameter combinations. That is, it can be used as a measure for studying barren plateaus in maximum likelihood estimators (Karakida et al. 2019), given that all the matrix entries will approach zero with the flatness of the model's landscape. This effect is captured by looking at the spectrum of the matrix. If the model is in a barren plateau, then the eigenvalues of the matrix will approach zero (Abbas et al. 2021).

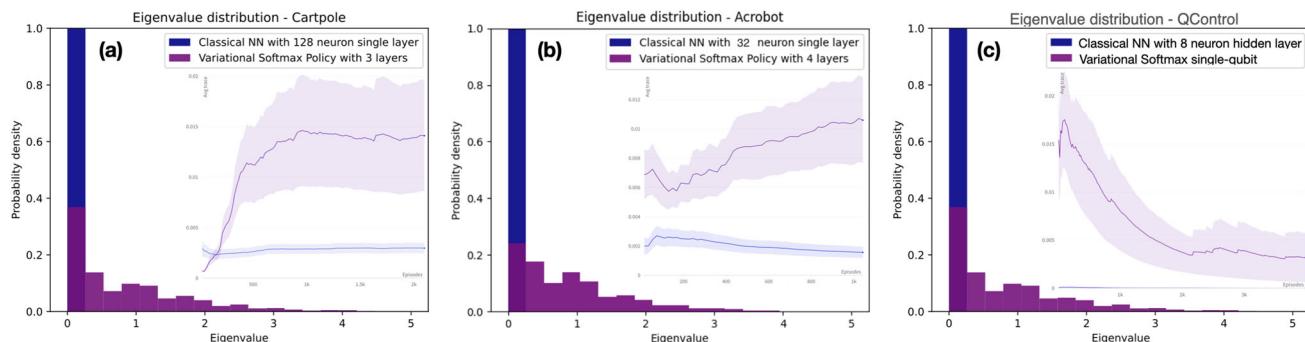
In the context of policy gradients, the empirical Fisher information matrix (Kakade 2001) is obtained by multiplying the vector resultant of the gradient of the log-policy with its transpose as in Eq. 26:

$$F(\theta) = \frac{1}{T} \sum_{t=1}^T \nabla_\theta \log \pi(a_t|s_t, \theta) \nabla_\theta \log \pi(a_t|s_t, \theta)^\top \quad (26)$$

Inspecting the spectrum of the matrix in Eq. 26 reveals the flatness of the loss landscape. Thus, it can harness the

hardness of the model's trainability for both RL agents based on classical neural networks and VQCs (Abbas et al. 2021). This work considers the trace and the eigenvalues' probability density of the Fisher information matrix. The trace will approach zero if the model is closer to a barren plateau and the eigenvalues' probability density unveils the magnitude of the associated eigenvalues.

Figure 8a, b, and c plot the average Fisher information matrix eigenvalue distribution for training episodes during the entire training for the CartPole, Acrobot, and QControl environments, respectively. Subpanels in every plot indicate the associated information matrix trace. On average, the Fisher information matrix of the quantum model exhibits significantly larger density in eigenvalues different from zero compared to the classical model during the entire training. The same behavior is observed for every environment, explaining the improvement of the training performance for quantum agents (Section 5) compared to classical ones. Although it is not visible from the eigenvalue distribution, the classical model has larger eigenvalues than the quantum model. However, their density is extremely small, thus making it negligible in a distribution plot. Further analysis is required to understand the behavior of both classical and quantum agents thoroughly.



**Fig. 8** Probability density for the Fisher information matrix eigenvalues and average trace. Panels **a**, **b**, and **c** represent the eigenvalue distribution and trace of the Fisher information matrix for the Cartpole, Acrobot, and QControl environments, respectively

## 7 Conclusion

In this work, a VQC was embedded into the decision-making process of an RL agent, following the policy gradient algorithm, solving a set of standard benchmarking environments efficiently. Empirical results demonstrate that such variational quantum models behave similarly or even outperform several typically used classical neural networks. The quantum-inspired policy needs fewer interactions to converge to an optimal behavior, benefiting from a reduction in the total number of trainable parameters.

Parameter-shift rules were used to perform gradient-based optimization resorting to the same quantum model used to compute the policy. It was proved that the sample complexity for gradient estimation via parameter-shift rules grows only logarithmically with the number of parameters.

The Fisher information spectrum was used to study the effect of barren plateaus in quantum policy gradients. The spectrum indicates that the quantum model comprises larger eigenvalues than its classical counterpart, suggesting that the optimization surface is less prone to plateaus.

Finally, it was verified that the quantum model could prepare a single-qubit state with high fidelity in fewer episodes than the classical counterpart with a single layer.

Concerning future work, it would be interesting to apply such RL-based variational quantum models to quantum control problems of larger dimensions. Specifically, their application to noisy environments would be of general interest. Moreover, studying the expectation value of policy gradients given a specific initialization strategy to support empirical claims is crucial. At last, the quantum Fisher information (Meyer 2021) should be addressed to analyze the information behind quantum states. Moreover, it would be interesting to embed the quantum Fisher information in a natural gradient optimization (Stokes et al. 2020) to derive quantum natural policy gradients. Advanced RL models such as actor-critic or deep deterministic policy gradients (DDPG) could benefit from quantum-aware optimization.

## Appendix A. Upper bounds on gradient estimation

This appendix develops the proofs for Lemmas 4.1 and 4.2, as presented in Section 4.5.

### A.1: $\epsilon_{\nabla}$ -approximation of the policy-gradient

Lemma 4.1 establishes an upper bound on the number of samples required to  $\epsilon_{\nabla}$ -estimate the policy gradient  $\hat{\nabla}_{\theta} J(\theta)$ .

**Lemma A.1** ( $\epsilon_{\nabla}$ -approximation of the policy-gradient) *Let  $\theta \in \mathbb{R}^k$ ,  $k$  being the number of*

*parameters,  $R_{\max}$  be the maximum possible reward in any time step,  $T$  the horizon, and  $\nabla_{\theta} J(\theta)$  the expected policy gradient. The policy gradient,  $\hat{\nabla}_{\theta} J(\theta)$ , can be  $\epsilon_{\nabla}$ -approximated, with probability  $1 - \delta_{\nabla}$*

$$|\hat{\nabla}_{\theta} J(\theta) - \nabla_{\theta} J(\theta)| \leq \epsilon_{\nabla} \quad (27)$$

*using a number of samples given by*

$$NT \approx \mathcal{O}\left(\frac{8\beta^2 R_{\max}^2 T^3}{\epsilon_{\nabla}^2 (\gamma - 1)^4} \log\left(\frac{2k}{\delta_{\nabla}}\right)\right) \quad (28)$$

*Proof* The policy gradient is estimated by resorting to Monte Carlo techniques, as described by Eq. 15, restated here for completion.

$$\begin{aligned} \nabla_{\theta} J(\theta) = \frac{1}{N} \sum_{i=0}^{N-1} \sum_{t=0}^{T-1} G_t(\tau_i) \beta & \left( \nabla_{\theta} \langle a_{t_i} \rangle_{\theta} \right. \\ & \left. - \sum_{b_{t_i}} \pi(b_{t_i} | s_{t_i}, \theta) \nabla_{\theta} \langle b_{t_i} \rangle_{\theta} \right) \end{aligned}$$

Recall that the number of samples is defined as the number of visited states. Since there are  $N$  trajectories (sequences of actions,  $\tau_i$ ), each visiting  $T$  states, the total number of samples is equal to  $NT$ .

Since the expectation value of a single qubit observable is bounded as  $\langle \sigma_z \rangle \in [-1, 1]$  and since the gradient of an action's expected value is given by Eq. 18, then  $\nabla_{\theta} \langle a \rangle_{\theta} \in [-1, 1]$ . Therefore, the following holds:

$$\beta \left( \nabla_{\theta} \langle a_{t_i} \rangle_{\theta} - \sum_{b_{t_i}} \pi(b_{t_i} | s_{t_i}, \theta) \nabla_{\theta} \langle b_{t_i} \rangle_{\theta} \right) \in [-2\beta, 2\beta] \quad (29)$$

By defining  $R_{\max}$  as the maximum possible reward at any time step and by recalling Eq. 4, then

$$G(\tau) = \sum_{t=0}^{T-1} \gamma^t r_{t+1} \leq R_{\max} \sum_{t=0}^{T-1} \gamma^t = R_{\max} \frac{\gamma^T - 1}{\gamma - 1} \quad (30)$$

where the expression for the sum of  $T$  terms of a geometric progression was used. Using this upper bound on  $G(\tau)$  enables the following result

$$\sum_{t=0}^{T-1} G_t(\tau) \leq R_{\max} \sum_{t=0}^{T-1} \frac{\gamma^{T-t} - 1}{(\gamma - 1)} \leq R_{\max} \frac{T}{(\gamma - 1)^2} \quad (31)$$

where the last inequality can be obtained by algebraic development and by resorting again to the sum of terms of a geometric progression.

Combining results Eqs. 29 and 31, gives

$$\sum_{t=0}^{T-1} G_t(\tau_i) \beta \left( \nabla_\theta \langle a_{\tau_i} \rangle_\theta - \sum_{b_{\tau_i}} \pi(b_{\tau_i} | s_{\tau_i}, \theta) \nabla_\theta \langle b_{\tau_i} \rangle_\theta \right) \\ \in \left[ -\frac{2\beta R_{\max} T}{(\gamma - 1)^2}, \frac{2\beta R_{\max} T}{(\gamma - 1)^2} \right] \quad (32)$$

From Hoeffding's inequality (Hoeffding 1963), the probability of the average over  $N$  estimates of the policy gradient random variable being  $\epsilon_V$ -inaccurate is given by

$$\mathbb{P}[|\nabla_\theta^* J(\theta) - \nabla_\theta J(\theta)| \geq \epsilon_V] \leq 2 \exp\left(-\frac{2N\epsilon_V^2(\gamma - 1)^4}{16\beta^2 R_{\max}^2 T^2}\right) \quad (33)$$

From the union bound, for all  $k$  parameters, the probability is less than

$$\mathbb{P}\left[\bigcup_k 2 \exp\left(-\frac{N\epsilon_V^2(\gamma - 1)^4}{8\beta^2 R_{\max}^2 T^2}\right)\right] \leq 2k \exp\left(-\frac{N\epsilon_V^2(\gamma - 1)^4}{8\beta^2 R_{\max}^2 T^2}\right) \quad (34)$$

Let  $\delta_V = \mathbb{P}[|\nabla_\theta^* J(\theta) - \nabla_\theta J(\theta)| \geq \epsilon_V]$ . Then

$$1 - \delta_V = \mathbb{P}[|\nabla_\theta^* J(\theta) - \nabla_\theta J(\theta)| \leq \epsilon_V] \\ 1 - \delta_V \geq 1 - 2k \exp\left(-\frac{N\epsilon_V^2(\gamma - 1)^4}{8\beta^2 R_{\max}^2 T^2}\right) \\ \delta_V \leq 2k \exp\left(-\frac{N\epsilon_V^2(\gamma - 1)^4}{8\beta^2 R_{\max}^2 T^2}\right) \quad (35)$$

Thus, an upper bound on  $N$  can be obtained

$$N \leq \frac{8\beta^2 R_{\max}^2 T^2}{\epsilon_V^2(\gamma - 1)^4} \log\left(\frac{2k}{\delta_V}\right) \quad (36)$$

Considering  $NT$  samples completes the proof.  $\square$

## A.2: Total number of quantum circuit evaluations

Lemma 4.2 establishes an upper bound on the number of quantum circuit evaluations (or shots) required to  $\epsilon_{\langle \cdot \rangle}$ -estimate the policy gradient  $\hat{\nabla}_\theta J(\theta)$  with probability  $1 - \delta_{\langle \cdot \rangle}$ . This result builds on Lemma 4.1 and the same approach is used to demonstrate it.

*Proof* An action preference observable  $\langle a \rangle_\theta$  is given by a single-qubit observable  $\langle \sigma_z \rangle$ , as described in Section 4.3. The number of shots,  $n'$ , required to estimate the observable expectation with additive error  $\epsilon_{\langle \cdot \rangle}$  with probability  $1 - \delta_{\langle \cdot \rangle}$  is akin to the estimation of the probability of a Bernoulli distribution using Hoeffding inequality. Since  $\langle a \rangle_\theta \in [-1, 1]$ , then, by resorting to Hoeffding inequality and the union bound, we have

$$\mathbb{P}[|\langle a \rangle_\theta^* - \langle a \rangle_\theta| \geq \epsilon_{\langle \cdot \rangle}] \leq 2k \exp\left(-\frac{n'\epsilon_{\langle \cdot \rangle}^2}{2}\right) \quad (37)$$

Following the same reasoning as described in the proof of Lemma 4.1,  $n'$  is given by

$$n' \leq \frac{2}{\epsilon_{\langle \cdot \rangle}^2} \log\left(\frac{2k}{\delta_{\langle \cdot \rangle}}\right) \quad (38)$$

Since the observable's gradient  $\nabla_\theta \langle a \rangle_\theta$  is estimated via parameter shift rules, as stated in Eq. 18, it requires the estimation of each action preference observable twice, i.e., both  $\langle a \rangle_{\theta+\frac{\pi}{2}}$  and  $\langle a \rangle_{\theta-\frac{\pi}{2}}$ . Therefore, the number of shots,  $n$ , required to estimate  $\nabla_\theta \langle a \rangle_\theta$  is given by

$$n = 2n' \leq \frac{4}{\epsilon_{\langle \cdot \rangle}^2} \log\left(\frac{2k}{\delta_{\langle \cdot \rangle}}\right) \approx \mathcal{O}\left(\frac{1}{\epsilon_{\langle \cdot \rangle}^2} \log\left(\frac{2k}{\delta_{\langle \cdot \rangle}}\right)\right) \quad (39)$$

Recalling that  $\mathcal{O}(NT)$  samples are needed as in Lemma 4.1 and that each sample incurs  $|A|$  estimates, completes the proof.  $\square$

**Funding** Open access funding provided by FCT—FCCN (b-on). This work is financed by National Funds through the Portuguese funding agency, FCT — Fundação para a Ciência e a Tecnologia, within grants LA/P/0063/2020, UI/BD/152698/2022 and project IBEX, with reference PTDC/CC1-COM/4280/2021.

**Data availability** Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

## Declarations

**Conflict of interest** The authors declare no competing interests.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>

## References

- Abbas A, Sutter D, Zoufal C, Lucchi A, Figalli A, Woerner S (2021) The power of quantum neural networks. Nat. Comput. Sci. 1(6):403–409. <https://doi.org/10.1038/s43588-021-00084-1>
- Afsar MM, Crump T, Far B (2021) Reinforcement learning based recommender systems: a survey
- Agarwal A, Jiang N, Kakade S (2019) Reinforcement learning: theory and algorithms
- Aïmeur E., Brassard G, Gambs S (2006) Machine learning in a quantum world. In: Lamontagne L, Marchand M (eds) Advances in artificial intelligence. Springer, Berlin, pp 431–442

- Bergholm V, Izaac J, Schuld M, Gogolin C, Alam MS, Ahmed S, Arrazola JM, Blank C, Delgado A, Jahangiri S, McKiernan K, Meyer JJ, Niu Z, Száva A, Killoran N (2020) PennyLane: automatic differentiation of hybrid quantum-classical computations
- Bharti K, Cervera-Lierta A, Kyaw TH, Haug T, Alperin-Lea S, Anand A, Degroote M, Heimonen H, Kottmann JS, Menke T, Mok W-K, Sim S, Kwek L-C (2021) Aspuru-guzik A.: noisy intermediate-scale quantum (NISQ) algorithms
- Bharti K, Cervera-Lierta A, Kyaw TH, Haug T, Alperin-Lea S, Anand A, Degroote M, Heimonen H, Kottmann JS, Menke T, Mok W-K, Sim S, Kwek L-C, Aspuru-Guzik A (2022) Noisy intermediate-scale quantum algorithms. *Rev Mod Phys* 94:015004. <https://doi.org/10.1103/RevModPhys.94.015004>
- Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J, Zaremba W (2016) Open AI Gym
- Cerezo M, Arrasmith A, Babbush R, Benjamin SC, Endo S, Fujii K, McClean JR, Mitarai K, Yuan X, Cincio L, Coles PJ (2021) Variational quantum algorithms. *Nature Reviews Physics* 3(9):625–644. <https://doi.org/10.1038/s42254-021-00348-9>
- Chen SYC, Yang CHH, Qi J, Chen PY, Ma X, Goan HS (2020) Variational quantum circuits for deep reinforcement learning. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2020.3010470>. arXiv: 1907.00397
- Dalgaard M, Motzoi F, Sørensen JJ, Sherson J (2020) Global optimization of quantum dynamics with alphazero deep exploration npj. *Quantum Information* 6(1). <https://doi.org/10.1038/s41534-019-0241-0>
- Dunjko V, Briegel HJ (2018) Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Rep Prog Phys* 81(7):074001. <https://doi.org/10.1088/1361-6633/aab406>
- Dunjko V, Liu Y-K, Wu X, Taylor JM (2017) Exponential improvements for quantum-accessible reinforcement learning. arXiv: 1710.11160
- Dunjko V, Taylor JM, Briegel HJ (2016) Quantum-enhanced machine learning. *Phys Rev Lett* 117(13):1–19. <https://doi.org/10.1103/PhysRevLett.117.130501>. arXiv: 1610.08251
- Farhi E, Neven H (2018) Classification with quantum neural networks on near term processors
- Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: AISTATS
- Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge
- Greensmith E, Bartlett PL, Baxter J (2004) Variance reduction techniques for gradient estimates in reinforcement learning. *J Mach Learn Res* 5:1471–1530
- Hoeffding W (1963) Probability inequalities for sums of bounded random variables. *J Am Stat Assoc* 58(301):13–30. <https://doi.org/10.1080/01621459.1963.10500830>
- Huang H-Y, Broughton M, Mohseni M, Babbush R, Boixo S, Neven H, McClean J (2021) Power of data in quantum machine learning. *Nat Commun* 12. <https://doi.org/10.1038/s41467-021-22539-9>
- James MR (2021) Optimal quantum control theory. *Annu Rev Control Robot Auton Syst* 4(1):343–367. <https://doi.org/10.1146/annurev-control-061520-010444>
- Jerbi S, Gyurik C, Marshall S, Briegel HJ, Dunjko V (2021) Variational quantum policies for reinforcement learning
- Kakade S (2001) A natural policy gradient. In: Proceedings of the 14th international conference on neural information processing systems: natural and synthetic. NIPS'01. MIT Press, pp 1531–1538
- Karakida R, Akaho S, Amari S-I (2019) Universal statistics of fisher information in deep neural networks: mean field approach
- Kingma DP, Ba J. (2017) Adam: a method for stochastic optimization
- Kiran BR, Sobh I, Talpaert V, Mannion P, Sallab AAA, Yogamani S, Pérez P (2021) Deep reinforcement learning for autonomous driving: A survey
- LaRose R, Coyle B (2020) Robust data encodings for quantum classifiers. ArXiv: 2003.01695
- Liu X-Y, Yang H, Chen Q, Zhang R, Yang L, Xiao B, Wang CD (2020) FinRL: a deep reinforcement learning library for automated stock trading in quantitative finance
- Lockwood O, Si M (2020) Reinforcement learning with quantum variational circuits. In: Proceedings of the 16th AAAI conference on artificial intelligence and interactive digital entertainment, AIIDE 2020
- Lockwood O, Si M (2021) Playing atari with hybrid quantum-classical reinforcement learning
- Ly A, Marsman M, Verhagen J, Grasman R, Wagenmakers E-J (2017) A tutorial on fisher information
- Martín-Guerrero J, Lamata L (2021) Reinforcement learning and physics. *Appl Sci* 11:8589. <https://doi.org/10.3390/app11188589>
- Meyer JJ (2021) Fisher information in noisy intermediate-scale quantum applications. *Quantum* 5:539. <https://doi.org/10.22331/q-2021-09-09-539>
- Mosavi A, Ghamisi P, Faghan Y, Duan P, Shamshirband S (2020) Comprehensive review of deep reinforcement learning methods and applications in economics. <https://doi.org/10.20944/preprints202003.0309.v1>
- Nielsen MA, Chuang IL (2011) Quantum computation and quantum information: 10th anniversary edition, 10th edn. Cambridge University Press, Cambridge
- Paparo GD, Dunjko V, Makmal A, Martin-Delgado MA, Briegel HJ (2014) Quantum speedup for active learning agents. *Phys Rev X* 4(3):1–14. <https://doi.org/10.1103/PhysRevX.4.031002>. ArXiv:1401.4997
- Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L, Lerer A (2017) Automatic differentiation in pytorch
- Preskill J (1997) Fault-tolerant quantum computation. arXiv: Quantum Physics
- Preskill J (2018) Quantum computing in the nisq era and beyond. *Quantum* 2:79. <https://doi.org/10.22331/q-2018-08-06-79>
- Saggio V, Asenbeck BE, Hamann A, Strömborg T, Schiansky P, Dunjko V, Friis N, Harris NC, Hochberg M, Englund D et al (2021) Experimental quantum speed-up in reinforcement learning agents. *Nature* 591(7849):229–233. <https://doi.org/10.1038/s41586-021-03242-7>
- Sanches F, Weinberg S, Ide T, Kamiya K (2021) Short quantum circuits in reinforcement learning policies for the vehicle routing problem
- Schrittwieser J, Antonoglou I, Hubert T, Simonyan K, Sifre L, Schmitt S, Guez A, Lockhart E, Hassabis D, Graepel T et al (2020) Mastering atari, go, chess and shogi by planning with a learned model. *Nature* 588(7839):604–609. <https://doi.org/10.1038/s41586-020-03051-4>
- Schuld M (2021) Quantum machine learning models are kernel methods
- Schuld M, Bergholm V, Gogolin C, Izaac JA, Killoran N (2019) Evaluating analytic gradients on quantum hardware. *Phys Rev A* 99:032331
- Schuld M, Petruccione F (2018) Supervised learning with quantum computers, 1st edn. Springer, Berlin
- Schuld M, Sweke R, Meyer JJ (2021) Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Phys Rev A* 103(3). <https://doi.org/10.1103/physreva.103.032430>

- Sequeira A, Santos LP, Barbosa LS (2021) Quantum tree-based planning. *IEEE Access* 9:125416–125427. <https://doi.org/10.1109/ACCESS.2021.3110652>
- Silver D, Huang A, Maddison CJ, Guez A, Sifre L, van den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, Dieleman S, Grewe D, Nham J, Kalchbrenner N, Sutskever I, Lillicrap TP, Leach M, Kavukcuoglu K, Graepel T, Hassabis D (2016) Mastering the game of go with deep neural networks and tree search. *Nature* 529:484–489
- Sim S, Johnson PD, Aspuru-Guzik A (2019) Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Adv Quantum Technol* 2(12):1900070. <https://doi.org/10.1002/qute.201900070>
- Stokes J, Izaac J, Killoran N, Carleo G (2020) Quantum natural gradient. *Quantum* 4:269. <https://doi.org/10.22331/q-2020-05-25-269>
- Sutton RS, Barto AG (2018) Reinforcement learning: an introduction. A bradford book, Cambridge
- Sutton R, McAllester DA, Singh S, Mansour Y (1999) Policy gradient methods for reinforcement learning with function approximation. In: NIPS
- Sweke R, Wilde F, Meyer J, Schuld M, Faehrmann PK, Meynard-Piganeau B, Eisert J (2020) Stochastic gradient descent for hybrid quantum-classical optimization. *Quantum* 4:314. <https://doi.org/10.22331/q-2020-08-31-314>
- Williams RJ (2004) Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach Learn* 8:229–256
- Wu S, Jin S, Wen D, Wang X (2021) Quantum reinforcement learning in continuous action space
- Zhang K, Hsieh M-H, Liu L, Tao D (2022) Gaussian initializations help deep variational quantum circuits escape from the barren plateau arXiv. <https://doi.org/10.48550/ARXIV.2203.09376>
- Zhang X-M, Wei Z, Asad R, Yang X-C, Wang X (2019) When does reinforcement learning stand out in quantum control? a comparative study on state preparation. *npj Quantum Information* 5:1–7. <https://doi.org/10.1038/s41534-019-0201-8>

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

# Ensemble Metropolis Light Transport

THOMAS BASHFORD-ROGERS, University of the West of England, UK

LUÍS PAULO SANTOS, Universidade do Minho/INESC TEC, Portugal

DEMETRIS MARNERIDES and KURT DEBATTISTA, University of Warwick, UK



Fig. 1. The DOOR AJAR scene showing the reference on the left, Metropolis Light Transport in the centre, and Ensemble Metropolis Light Transport on the right, rendered at an equal sample count. The use of the ensemble to propose anisotropic transition kernels allows sampling to be adapted to the scene geometry and lighting information, leading to variance reduction as can be seen by lower MSE values shown at the top of each image.

This article proposes a **Markov Chain Monte Carlo (MCMC)** rendering algorithm based on a family of guided transition kernels. The kernels exploit properties of ensembles of light transport paths, which are distributed according to the lighting in the scene, and utilize this information to make informed decisions for guiding local path sampling. Critically, our approach does not require caching distributions in world space, saving time and memory, yet it is able to make guided sampling decisions based on whole paths. We show how this can be implemented efficiently by organizing the paths in each ensemble and designing transition kernels for MCMC rendering based on a carefully chosen subset of paths from the ensemble. This algorithm is easy to parallelize and leads to improvements in variance when rendering a variety of scenes.

CCS Concepts: • Computing methodologies → Ray tracing;

Additional Key Words and Phrases: Light transport, MCMC, ensemble

## ACM Reference format:

Thomas Bashford-Rogers, Luis Paulo Santos, Demetris Marnerides, and Kurt Debattista. 2021. Ensemble Metropolis Light Transport. *ACM Trans. Graph.* 41, 1, Article 5 (December 2021), 15 pages.

<https://doi.org/10.1145/3472294>

5

## 1 INTRODUCTION

Accurately rendering photorealistic imagery requires computing extremely large numbers of light paths in a virtual environment. While most research has been applied to traditional Monte Carlo estimators for rendering, **Markov Chain Monte Carlo (MCMC)** methods, such as **Metropolis Light Transport (MLT)** [Veach and Guibas 1997], have shown impressive capabilities for computing light transport efficiently, even in complicated scenes.

MCMC algorithms such as MLT generate a chain of paths that follow the distribution of the lighting in the scene. Each new path is generated by applying a transition kernel to the previous path, and probabilistically replacing the previous path with the new path. In the original application of MCMC to graphics [Veach and Guibas 1997], transition kernels were designed to either locally explore regions around an existing path, or to globally explore path space. While these transition kernels have been improved to consider local geometric or lighting information [Li et al. 2015; Otsu et al. 2018], the use of non-local information capturing a wider range of lighting can also be used to guide transition kernels. Such non-local information can be captured by path guiding methods, for example [Müller et al. 2017]; however, this comes at a precomputation and memory cost, and the cached distributions of lighting may not clearly map to transition kernels. Another approach is to exploit the fact that multiple light paths generated by MCMC

---

Authors' addresses: T. B.-Rogers, University of the West of England, Coldharbour Ln, Bristol BS16 1QY, UK; L. P. Santos, Universidade do Minho/INESC TEC, Campus de Gualtar, Braga 4710-057, Portugal; D. Marnerides and K. Debattista, University of Warwick, Gibbet Hill Road, Coventry CV4 7AL, UK.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM. 0730-0301/2021/12-ART5 \$15.00  
<https://doi.org/10.1145/3472294>

algorithms will be distributed proportional to the lighting in the scene, and as such can be used to generate guided transition kernels.

This article proposes such a method that uses ensembles of light paths to guide mutations of existing paths. We name this approach **Ensemble Metropolis Light Transport (EMLT)**. Crucially, these guided transition kernels do not need to be based on caching distributions in world space; they only require moderately sized ensembles in the low tens of thousands of paths, and can be combined to build families of mutation strategies. To summarize, the main contributions of this work are as follows:

- The introduction of EMLT, a method that guides sampling based on a complementary ensemble of transport paths.
- A family of adaptive and anisotropic proposal distributions for path mutations based on ensemble sampling.
- The use of a carefully chosen subset of paths from the ensemble to create guided transition kernels.
- Results showing improvement of EMLT over traditional approaches in a range of real-world scenes.

## 2 BACKGROUND AND RELATED WORK

This section introduces the relevant background theory of light transport, path guiding methods which exploit information about the radiance or importance distribution in the scene to reduce variance, and MCMC methods which can efficiently compute images in challenging scenes.

### 2.1 Light Transport

The path integral form of the rendering equation [Kajiya 1986] is given by [Hachisuka et al. 2014; Veach and Guibas 1997]

$$I_j = \int_{\mathcal{P}} h_j(\bar{x}) f(\bar{x}) d\mu(\bar{x}), \quad (1)$$

and states that the intensity  $I_j$  at a pixel  $j$  consists of the contribution  $f(\bar{x})$  of light paths  $\bar{x}$  weighted by a pixel filter  $h_j(\bar{x})$ . The domain of integration is the union of all possible path lengths  $\mathcal{P} = \bigcup_{k=2}^{\infty} \mathcal{P}(k)$  where  $\mathcal{P}(k)$  are all paths of length  $k$ . In this work, path vertices  $x_0..x_k$  lie on the scene manifold  $\mathcal{M}$ , i.e., integration is with respect to the product area measure  $\mu$ , and are indexed starting from the light source  $x_0$ . The contribution of a path of length  $k$  is defined as  $f(\bar{x}) = Le(\bar{x}_0)G(\bar{x}_0 \leftrightarrow \bar{x}_1) \prod_{j=1}^{k-1} fr(\bar{x}_{j-1} \rightarrow \bar{x}_j \rightarrow \bar{x}_{j+1})G(\bar{x}_j \leftrightarrow \bar{x}_{j+1})$ , where  $Le$  is the emitted radiance,  $G$  is the geometry term, and  $fr$  is the **Bidirectional Reflectance Distribution Function (BRDF)**.

There are multiple ways of solving Equation (1), almost all relying on Monte Carlo estimation:

$$I_j \approx \frac{1}{N} \sum_{i=1}^N \frac{h_j(\bar{x}(i)) f(\bar{x}(i))}{p(\bar{x}(i))}, \quad (2)$$

where  $p(\bar{x}(i))$  denotes the **probability density function (pdf)** of sampling the  $i$ th path and is a product of probability densities for sampling each vertex to build up the path. This typically consists of sampling the sensor, lens, BRDFs and light sources. Ideally  $p(\bar{x}(i)) \propto h_j(\bar{x}(i)) f(\bar{x}(i))$ ; however, this is typically not possible

in practice. Therefore, distributions which approximate some components of  $h_j(\bar{x}(i)) f(\bar{x}(i))$  are used (see Christensen and Jarosz [2016] for a survey of these methods).

### 2.2 Path Guiding

Most rendering techniques generate light paths incrementally by sampling the next vertex in a path given the previous vertex. Path guiding approaches build on traditional BRDF and cosine sampling to include information about incoming illumination or importance when generating samples. Most techniques cache a distribution that represents the incoming radiance or importance at a sparse set of locations in a scene, and query locations at runtime using a spatial data structure. Examples include 5D spatio-directional Trees [Lafontaine and Willems 1995; Müller et al. 2017], 7D distributions [Pantaleoni 2020], use of various basis functions to store radiance at discrete points in the scene [Bashford-Rogers et al. 2012; Diolatzis et al. 2020; Herholz et al. 2016; Hey and Purgathofer 2002; Jensen 1995; Ruppert et al. 2020; Vorba et al. 2014], or using machine learning methods [Bako et al. 2019; Dahm and Keller 2017].

Path guiding has also been applied to sample partial or complete light paths. Approaches such as neural importance sampling [Guo et al. 2018; Müller et al. 2018; Zheng and Zwicker 2019] have learned a warping in **Primary Sample Space (PSS)** [Kelemen et al. 2002] which encodes the illumination distribution in PSS based on a small set of paths traced before rendering. However, as these approaches are designed to generate full paths, they face the curse of dimensionality and are more effective in lower dimensional scenarios such as importance sampling one bounce indirect lighting. A related approach to whole path importance sampling was proposed by Reibold et al. [2018] that selectively stores and samples distributions for high contribution paths which were unlikely to be sampled through BRDF sampling.

EMLT exploits information about the lighting distribution in the scene, and can use any of the distributions commonly used for path guiding to generate samples. However, our method does not require a spatial cache, and builds distributions on the fly from a small set of paths from a complementary ensemble (see Section 3).

### 2.3 MCMC

MCMC [Hastings 1970; Metropolis et al. 1953] techniques provide another approach to sample a space and were initially applied to rendering as MLT [Veach and Guibas 1997]. In MLT, sampling starts from an initial path  $\bar{x}$ , and then proposes a new path  $\bar{x}'$  from a transition kernel  $T(\bar{x} \rightarrow \bar{x}')$ . The transition kernels have to satisfy certain properties in order for the chain to explore the state space: ergodicity meaning all states will be visited by the chain in a finite time, and aperiodicity meaning states will not get stuck in a loop. At the limit, these states are distributed according to a target distribution,  $\frac{f}{b}$ , where  $b = \int_{\mathcal{P}} h_j(\bar{x}) f(\bar{x}) d\mu(\bar{x})$  is a normalization constant. A scalar contribution function,  $f^* : \mathbb{R}^S \mapsto \mathbb{R}$ , is defined where  $S$  are the spectra or color channels associated with evaluating  $f(\bar{x})$ . Then, based on the detailed balance condition  $f^*(\bar{x}) T(\bar{x} \rightarrow \bar{x}') a(\bar{x} \rightarrow \bar{x}') = f^*(\bar{x}') T(\bar{x}' \rightarrow \bar{x}) a(\bar{x}' \rightarrow \bar{x})$ , the new state  $\bar{x}'$  is probabilistically chosen to replace the previous

state based on calculating an acceptance probability:

$$a(\bar{x} \rightarrow \bar{x}') = \min\left(1, \frac{f^*(\bar{x}')T(\bar{x}' \rightarrow \bar{x})}{f^*(\bar{x})T(\bar{x} \rightarrow \bar{x}')} \right). \quad (3)$$

This leads to a chain of light paths, each dependent only on the previously sampled light path, which explore the path space. In rendering, the scalar contribution function is typically chosen to be the luminance of the contribution of the path, but other functions can be chosen (see Gruson et al. [2016]; Hoberock and Hart [2010]).  $b$  is typically estimated by a separate Monte Carlo estimator, and the initial state of the chain is generated through resampling a path from a small set of paths computed at startup. Then, the resulting Monte Carlo estimator is given by

$$I_j \approx \frac{b}{N} \sum_{i=1}^N \frac{h_j(\bar{x}(i))f(\bar{x}(i))}{f(\bar{x}(i))} = \frac{b}{N} \sum_{i=1}^N h_j(\bar{x}(i)), \quad (4)$$

meaning that samples will be distributed according to the integrand. Equation (4) is typically evaluated over the image plane which allows for information about light transport to be shared between pixels, resulting in a significantly more efficient estimator. Many strategies can be designed such that many terms in the numerator and denominator in Equation (3) cancel, something which is especially important to remove the weak singularity in the geometry term. The variance reduction properties of this method also depends on the ability of the transition kernel to explore the state space.

Veach and Guibas [1997] proposed a series of transition kernels which were chosen to reduce variance for different types of light transport. Bidirectional mutations were designed to ensure ergodicity through deleting a randomly chosen series of vertices from  $\bar{x}$ , and replacing them with vertices generated through sampling the same pdfs used in a standard Monte Carlo estimator, i.e., BRDF and light source sampling. The remaining strategies, known as perturbations, were designed to explore sub spaces of path space given the state of the path. Lens perturbations explored image space by perturbing the position of the path vertex on the image plane, tracing a path through any specular interactions until a non-specular vertex is reached, then deterministically connecting to the unchanged light subpath. This connection leaves a geometry term associated with the deterministic connection when evaluating Equation (3). The caustic perturbation was designed to explore caustics through perturbing the outgoing direction for the first vertex on the caustic subpath, following the chain of specular interactions, and then deterministically connecting to the camera. The multi-chain perturbation explores specular-diffuse-specular paths through combining a perturbation on the lens with directional perturbations at each non-specular surface before deterministically connecting to the remaining light subpath.

There have been several extensions to the original MLT algorithm which have added or improved mutation strategies such as perturbations in participating media [Pauly et al. 2000], improved sampling of specular chains [Jakob and Marschner 2012], and perturbations in half vector space [Kaplanyan et al. 2014]. Kelemen et al. [2002] introduced mutations to light paths in PSS (PSSMLT). These mutations were improved by Hachisuka et al. [2014] who

combined PSSMLT with Multiple Importance Sampling [Veach and Guibas 1995], Bitterli and Jarosz [2019] who detected and perturbed high variance paths in PSS, the use of delayed rejection by Rioux-Lavoie et al. [2020], the use of Hamiltonian Monte Carlo applied to rendering by Li et al. [2015] who used anisotropic Gaussian kernels generated from a path gradient, and Luan et al. [2020] who used the Metropolis-adjusted Langevin algorithm also based on the gradients of the path. Integration in both path space and PSS have been proposed [Bitterli et al. 2018; Otsu et al. 2017; Pantaleoni 2017] which allows path space mutations to be combined with PSS mutations. For further information, Šík and Křivánek [2018] provide a detailed survey of MCMC methods in rendering.

Closer to our work, adaptive perturbation sizes based on scene geometry were proposed by Otsu et al. [2018], which used cone tracing to estimate how large a perturbation could be based on the surrounding geometry of a path. This was applied starting at the camera, followed specular bounces if any, then traced one extra path vertex to form a perturbed path.

Other methods mutate a set of paths but do not directly use these to adapt transition kernels. Energy Redistribution Path Tracing [Cline et al. 2005] combined Path Tracing and MLT by creating many short chains whenever a path would be better explored by MCMC methods than standard Monte Carlo. Segovia et al. [2007] used Multiple-Try MCMC to generate paths for Instant Radiosity [Keller 1997], and Nimier-David et al. [2019] also proposed a Multiple-Try MCMC method suitable for vectorized instructions.

## 2.4 Ensemble MCMC Methods

The use of multiple paths have been used in rendering to reduce variance and better explore path space. These methods have largely focused on variants of parallel tempering, also known as Replica Exchange Monte Carlo [Swendsen and Wang 1986]. This uses multiple Markov Chains to explore different spaces, and uses a detailed balance preserving transition to swap chains between spaces. This was introduced to graphics by Kitaoka et al. [2009], and improved by Šík and Křivánek [2016] and Otsu et al. [2013]. These approaches have also been applied to progressive photon mapping [Hachisuka and Jensen 2011], the combination VCM/UPS with MCMC [Šík et al. 2016], and with stratified MCMC on the image plane [Gruson et al. 2020]. Hachisuka et al. [2014] also used a pool of chains of different lengths to sample path lengths proportional to their contribution.

Another related approach is to use **Population Monte Carlo (PMC)** [Cappé et al. 2004; Fan et al. 2007; Lai et al. 2007]. This iteratively and adaptively samples and resamples a population of paths proportional to their contribution and guides future samples, typically by adapting the parameters of distributions or kernels used to generate samples. While this is related to our approach, it is not trivial to combine PMC with MCMC methods without biasing the result, and it is also not clear how this approach can be applied when computing high dimensional integrals.

One approach outside of the graphics literature which is closely related to our work is **Affine Invariant Sampling (AIS)** [Goodman and Weare 2010]. This work considered a pool or ensemble of walkers  $\in \mathbb{R}^N$ , and used the states of all other walkers to guide

perturbations for each walker. The authors proposed three perturbations: stretch moves which shift a walker’s position toward or away from a randomly sampled walker in the ensemble, a walk move which samples a subset of walkers and builds a Gaussian transition kernel, and a replacement move which aims to reconstruct the whole space and sample from the reconstruction. These techniques were shown to efficiently guide the sampling, especially in the case of complicated distributions. This was further extended by Foreman-Mackey et al. [2013] which proposed a parallel approach to using the ensemble of walkers. We also use a similar approach of partitioning the walkers into two pools, and using one pool to guide sampling in the complementary pool.

### 3 ENSEMBLE METROPOLIS LIGHT TRANSPORT

Our approach is based on an ensemble of paths which capture global information of the distribution of lighting in a scene to guide sampling for each path in the ensemble. We first define an ensemble of chains containing  $O$  paths:

$$\mathbf{X} = \{\bar{x}^1, \bar{x}^2, \dots, \bar{x}^O\}. \quad (5)$$

This ensemble can be considered to be in  $\mathcal{P}^O$ . Similar to the argument in Goodman and Weare [2010], if we consider a product density using the ensemble  $F(\mathbf{X}) = f(\bar{x}^1)f(\bar{x}^2)\dots f(\bar{x}^O)$ , then any MCMC algorithm which preserves this density is valid. Such a strategy is to update each path in the ensemble conditioned on the other paths in the ensemble, i.e., following partial resampling [Liu 2008], if when updating path  $\bar{x}^i$  the remaining paths in the ensemble  $\{\bar{x}^1, \dots, \bar{x}^{i-1}, \bar{x}^{i+1}, \dots, \bar{x}^O\}$  remain fixed, then the update of the  $i$ ’th path of the ensemble preserves the joint distribution  $F$ . This also allows the other paths in the ensemble to guide the sampling of each path of the ensemble.

This implies updating each path in series, as each update relies on fixing the states of all other paths in the ensemble. However, paths can be updated in parallel for all  $\bar{x}^i \in \mathbf{X}$  by defining a complementary ensemble,  $\mathbf{Y} = \{\bar{y}^1, \bar{y}^2, \dots, \bar{y}^O\}$ , to guide sampling for each path in  $\mathbf{X}$  [Foreman-Mackey et al. 2013]. Therefore, each path in  $\mathbf{X}$  can be processed in parallel using  $\mathbf{Y}$  as guidance for sampling, i.e., the transition kernel takes the form  $T(\bar{x}^i \rightarrow \bar{x}'^i | \mathbf{Y})$ . This transition kernel can be written as the product of multiple sampling events; in the case of light transport this corresponds to progressively sampling a subpath:

$$T(\bar{x}^i \rightarrow \bar{x}'^i | \mathbf{Y}) = \prod_{j=1}^k K(\bar{x}_j^i \rightarrow \bar{x}'_j^i | \mathbf{Y}), \quad (6)$$

where  $K(\bar{x}_j^i \rightarrow \bar{x}'_j^i | \mathbf{Y})$  is a transition kernel for the  $j$ ’th sampling event of  $k$  events. Specifically, this is the transition kernel associated with perturbing the direction of a path vertex conditioned on the set of paths from the complementary ensemble. This transition kernel can be applied to one or more path vertices, producing a perturbation to a light path. The acceptance probability for updating each path in the ensemble is therefore computed as

$$a(\bar{x}^i \rightarrow \bar{x}'^i) = \min\left(1, \frac{f^*(\bar{x}'^i)T(\bar{x}'^i \rightarrow \bar{x}^i | \mathbf{Y})}{f^*(\bar{x}^i)T(\bar{x}^i \rightarrow \bar{x}'^i | \mathbf{Y})}\right). \quad (7)$$

---

**ALGORITHM 1:** The EMLT algorithm. Two ensembles of paths  $\mathbf{X}$  and  $\mathbf{Y}$  are input, and during rendering, paths from the ensemble  $\mathbf{X}$  are processed in parallel  $it$  times, and the guided transitions kernels based on  $\mathbf{Y}$  are used to propose new paths. After all paths in  $\mathbf{X}$  are processed,  $\mathbf{X}$  and  $\mathbf{Y}$  are swapped.

---

```

Input:  $\mathbf{X}$  and  $\mathbf{Y}$ 
1 while rendering do
2   ParFor  $\bar{x}^i \in \mathbf{X}$ 
3     for  $it$  iterations do                                ▷ See Section 3.4
4        $\bar{x}'^i \sim T(\bar{x}^i \rightarrow \bar{x}'^i | \mathbf{Y})$           ▷ See Section 3.3
5        $a \leftarrow a(\bar{x}^i \rightarrow \bar{x}'^i | \mathbf{Y})$            ▷ Equation (7)
6       Accumulate to Image
7       if  $\xi < a$  then
8          $\bar{x}^i \leftarrow \bar{x}'^i$ 
9       end
10      end
11    end
12  Swap  $\mathbf{X}$  and  $\mathbf{Y}$ 
13 end

```

---

Once all the paths of the ensemble  $\mathbf{X}$  have been updated, this is referred to as an iteration, the ensembles are swapped  $\mathbf{X} \leftrightarrow \mathbf{Y}$  and paths of  $\mathbf{Y}$  are updated based on using  $\mathbf{X}$  as path guidance:  $T(\bar{y}^i \rightarrow \bar{y}'^i | \mathbf{X})$ . However, without loss of generality we refer to  $\mathbf{Y}$  as the complementary ensemble in the remainder of the text. Algorithm 1 summarizes the EMLT algorithm. Firstly, the two ensembles  $\mathbf{X}$  and  $\mathbf{Y}$  are initialized, then during rendering each path is processed in parallel  $it$  times (lines 2 and 3) using the proposed guided transition kernels (lines 4–9). When all paths in an ensemble are processed, then ensembles are swapped (line 12), and the process repeats.

The use of ensembles for guiding sampling of paths could be applied to either path space or PSS. One possibility is to apply the use of ensembles to PSS through a strategy which directly perturbs a point in PSS based on other points in the ensemble, similar to AIS. However, due to the difference in the number of random numbers required to sample paths, it is not clear how walkers of different dimensionalities could be used to create any of the transition kernels proposed by Goodman and Weare [2010]. Secondly, interpolating between points in high dimensions, which is the result of applying AIS to PSS, is unlikely to lead to usable paths, especially if there are small regions in PSS containing valid light transport paths. However, the alternative of applying this to path space is also not trivial as samples are no longer in  $\mathbb{R}^N$ , and the strategies outlined in Goodman and Weare [2010] are not immediately applicable. Our proposed transition kernels are designed to be suitable for path space, but are also constructed to inherit the advantages of using an ensemble to guide sampling.

This then allows scope for a wide range of new guided transition kernels which are conditioned on the complementary ensemble. While the entire complementary ensemble could be used to create transition kernels, this would be prohibitively expensive when the complementary ensemble is large. An alternative, and significantly faster, approach that we propose in this article is to use a carefully chosen subset of the paths in the complementary ensemble. These paths should be similar, both in interaction types and spatial proximity, such that they can still produce valid guided transition

kernels. Section 3.1 describes how to efficiently find and weight the subset of paths from the complementary ensemble, then Section 3.2 describes how guided transition kernels can be constructed from this subset of paths. Finally, Section 3.3 describes how these guided transition kernels can be combined into path perturbations.

### 3.1 Complementary Ensemble

Before describing the transition kernels, we first describe two aspects of using the complementary ensemble for sampling. The first, explained in Section 3.1.1, is how to select paths from the complementary ensemble for sampling. This is important as many of the guided perturbations require paths to be sampled that maintain the same number of path vertices with the same interaction types as the original path.

The second aspect deals with the similarity of paths sampled from the complementary ensemble to the original path. This is required as although paths with the same length and interaction types may be sampled from the ensemble, paths which are similar to the original path are likely to lead to better proposal distributions than those further away. Effective use of this similarity between paths is what allows our approach to avoid a spatial cache. Section 3.1.2 describes an approach for measuring similarity between paths.

**3.1.1 Finding Paths.** As discussed previously, the proposed guided perturbations rely on a subset of  $M$  paths from  $\Upsilon$ :  $\Upsilon = \{\bar{v}^1 .. \bar{v}^M\}$ . These  $M$  paths are located in  $\Upsilon$  based on similar properties to a base path  $\bar{x}^i$ , such as identical length or the same Heckbert notation interaction types [Heckbert 1990]. This is motivated by two observations: (i) perturbations guided by similar paths, rather than all paths, are likely to explore similar regions of path space leading to higher acceptance probabilities and (ii) perturbations are more likely to succeed since they rely on preserving interaction types.

Therefore, the set of paths in  $\Upsilon$  is deterministically selected from the paths in  $\Upsilon$  with similar properties to  $\bar{x}^i$ . This is facilitated by a tree data structure over path lengths and interaction types which can be queried in  $O(1)$  time to find a subset of  $\Upsilon$  which matches the desired properties. This is built at the start of the rendering process, or at the end of each iteration, and please see the supplementary material for more details about the construction and traversal of this data structure. If the selection of paths forming  $\Upsilon$  was probabilistic and dependent on  $\bar{x}^i$ , then the probability of sampling the set  $\Upsilon$  given  $\bar{x}^i$  would have to be computed taking into account all paths with similar properties in  $\Upsilon$  which would be prohibitively slow. By performing a deterministic selection, in our case based on a counter which is stored with the ensemble and updated each step, this has the effect of having a minimal impact on performance with the additional benefit that the computation of the acceptance probabilities is significantly simplified as the probability of sampling  $\Upsilon$  is not required.

**3.1.2 Measuring Similarity.** The set of paths returned from querying the ensemble,  $\Upsilon$ , may have similar properties to the current light path  $\bar{x}^i$ . However, while some of the vertices in the paths returned may have similar positions in world space to  $\bar{x}^i$ , others

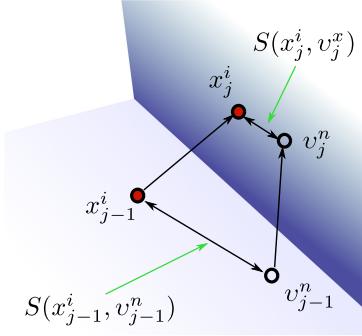


Fig. 2. Measuring similarity between a base path (red circles) and a path from  $v^n \in \Upsilon$  (empty circles). This is a product of the similarity between pairs of path vertices with the same index  $j$ :  $S(x_j^i, v_j^n)$ .

may not. When developing guided transition kernels, it is useful to have a measure of how similar light paths, or vertices within light paths, are to each other. For instance, some transition kernels can benefit from calculating weights for each vertex from  $\Upsilon$  as this is likely to provide a good estimate of nearby lighting.

The use of entire light paths in MCMC methods widens the range of methods to measure similarity. While Chaitanya et al. [2018] proposed an effective heuristic of total path length, i.e., the sum of distances between path vertices, we typically do not need to consider the whole path. We develop a heuristic based on the world space position of a set of vertices from  $\Upsilon$ , and vertices in the current light path  $\bar{x}^i$ . Other attributes, such as normals, albedo, or surface roughness could be considered, but we found that using the world space position was effective for computing similarity.

Specifically, given the  $j$ 'th vertex from  $\bar{x}^i$ ,  $x_j^i$ , and the previous vertex,  $x_{j-1}^i$ , the similarity value can be computed for all paths in  $\Upsilon$  by computing the distance to  $v_j^n$  and  $v_{j-1}^n$ ,  $n \in [1..M]$ . We define the difference between the world position of two vertices as  $d(x_j^i, v_j^n) = \max(|x_j^i - v_j^n|^2, \epsilon)^{-1}$  where  $\epsilon$  is a small positive constant (we use  $\epsilon = 0.0001$ ). From this, we define a normalized similarity value as

$$S(x_j^i, v_j^n) = \frac{2}{1 + e^{-d(x_j^i, v_j^n)}} - 1. \quad (8)$$

This scaled sigmoid leads to a larger value when vertices are similar, and smaller the further apart they become. See Figure 2 for an illustration of similarity computation. The similarity of multiple vertices starting at the  $j$ 'th position in the path to the  $k$ 'th position can be computed as

$$S(x^i, v^n, j, k) = \prod_{l=j}^k S(x_l^i, v_l^n). \quad (9)$$

### 3.2 Guided Transition Kernels

We first describe guided transition kernels for a single vertex, and then describe how full perturbation strategies can be built from these individual strategies in the following section. All of these methods require information gathered from the set returned from querying the tree structure  $\Upsilon$ .

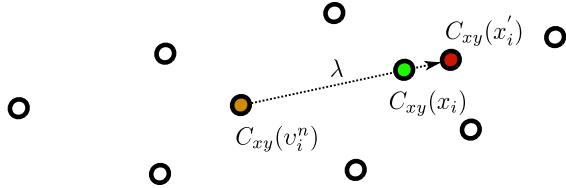


Fig. 3. Given some 2D domain, vertices in  $\Upsilon$  can be projected onto that domain (empty circles). One of these is selected (the orange circle) and used by Linear Transition Kernels to form a ray from that point to a projection of a vertex from the current path (the green circle). A distance along this ray is sampled  $\lambda$  which generates a new point in this domain (the red circle).

**Linear Transition Kernel.** The simplest form of guided transition kernels are the linear transition kernels, which is suitable for guided sampling on the lens. These operate in  $\mathbb{R}^2$  in our implementation. Given coordinates  $C_{xy}(x_i) \in \mathbb{R}^2$  of the current path vertex and the coordinates of a path vertex from  $\Upsilon$ :  $C_{xy}(v_i^n) \in \mathbb{R}^2$ , this generates a proposal along a ray in  $\mathbb{R}^2$ :  $C_{xy}(x'_i) = C_{xy}(v_i^n) + \lambda \cdot (C_{xy}(x_i) - C_{xy}(v_i^n))$ . The distance along the ray,  $\lambda$  is sampled from a distribution centered on  $C_{xy}(x_i)$ . This is illustrated in Figure 3.

Goodman and Weare [2010] proposed the stretch move which samples  $\lambda$  from a distribution  $\lambda \sim g(c) = \frac{1}{\sqrt{c}}$ , where  $c \in [\frac{1}{1+\alpha}, 1+\alpha]$ , where  $\alpha \in \mathbb{R}^+$  is a scaling term. This density is symmetric  $g(c) = cg(\frac{1}{c})$  (see Goodman and Weare [2010]), and this leads to the ratio of  $\frac{K(\bar{x}'_j \rightarrow \bar{x}_j | \Upsilon)}{K(\bar{x}_j \rightarrow \bar{x}'_j | \Upsilon)} = \lambda$ , simplifying the acceptance probability. However, other 1D distributions can be sampled to generate  $\lambda$ . For example, a uniform  $\lambda \sim [1-\beta, 1+\beta]$ ,  $\beta \in \mathbb{R}^+$ , or truncated Gaussian can be used, and so long as these are symmetric they simplify in the computation of the acceptance probability.

**Linear Hemispherical Transition Kernels.** While linear transition kernels are defined in  $\mathbb{R}^2$ , many transition kernels are required to be defined over the (hemi)sphere  $\mathbb{S}^2$ . Therefore, we extend the linear transition kernels to the (hemi)sphere. This starts by sampling  $\lambda$  from one of the linear distributions, then mapping this to a perturbation of the original direction on the sphere. Given two directions in the sphere  $\omega_1$  and  $\omega_2$ , these directions may correspond to a direction on the original path, and the other on a path from  $\Upsilon$ , a new direction  $\omega_{new}$  can be sampled along the great arc connecting these two directions:  $\omega_{new} = \frac{\omega_1 \sin((\omega_1 \cdot \omega_2) \lambda)}{\sin((\omega_1 \cdot \omega_2))} + \frac{\omega_2 \sin((\omega_1 \cdot \omega_2)(1-\lambda))}{\sin((\omega_1 \cdot \omega_2))}$ , i.e., a slerp between  $\omega_1$  and  $\omega_2$  with parameter  $1-\lambda$  (see Figure 4). This leads to

$$\frac{K(\bar{x}'_j \rightarrow \bar{x}_j | \Upsilon)}{K(\bar{x}_j \rightarrow \bar{x}'_j | \Upsilon)} = \frac{\sin \cos^{-1}(\omega_1 \cdot \omega_2)}{\sin \cos^{-1}(\omega_{new} \cdot \omega_2)} = \frac{\sqrt{1 - (\omega_1 \cdot \omega_2)^2}}{\sqrt{1 - (\omega_{new} \cdot \omega_2)^2}}. \quad (10)$$

**Guided Anisotropic Transition Kernels.** The linear and hemispherical transition kernels rely on a single path from the complementary ensemble, and domains in  $\mathbb{R}^2$  or  $\mathbb{S}^2$ . However, more

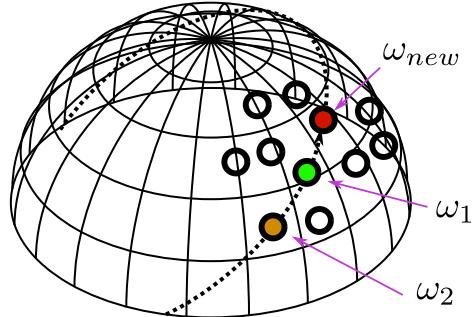


Fig. 4. The Linear Hemispherical Transition Kernel uses the outgoing direction of a path from  $\Upsilon$  (the orange circle), and the current path (the green circle) to propose a new direction (the red circle) along the great arc denoted by the dashed line. Other directions from  $\Upsilon$  which are not considered are shown as empty circles.

information can be gained from utilizing all  $M$  paths in  $\Upsilon$ . For example, a distribution in world space can be fit to the vertices at a certain point along the path, recentered at the current path vertex, and this distribution can be used for sampling. This allows lighting information from multiple paths to inform sampling of the current path, similar to Reibold et al. [2018].

There are multiple methods to achieve this; we describe one such approach. We start with the  $j$ 'th vertex in a path  $x_j$  and another vertex in the scene  $x'_{j-1}$ , and then retrieve the set of path vertices from  $\Upsilon$  which match the index:  $v_j^n, v_{j-1}^n \in \Upsilon$ . For each sub-path, we assign a weight:

$$w(n) = \frac{S(x^i, v^n, j, j+1)}{\sum_{k=1}^M S(x_j, v^k, j, j+1)}. \quad (11)$$

Then each of these points is projected onto the plane defined by  $x_j$  and the normal at  $x_j$ :  $N(x_j)$ . Next, an anisotropic Gaussian  $\mathcal{N}(\mu, \Sigma; \Upsilon, x_j)$  is fitted to these points via weighted maximum likelihood estimation where the weight of each point is that assigned to each path:  $\mu = \frac{1}{M} \sum_{n=1}^M w(n)v_j^n$  and  $\Sigma = \frac{1}{M-1} \sum_{n=1}^M w(n)(v_j^n - \mu)^2$ . This is recentered such that  $\mu = x_j$ , leading to  $\mathcal{N}(x_j, \Sigma; \Upsilon, \bar{x}_j)$ . This recentering is required such that the sampled point is close to the original, and to ensure that the evaluation of the reverse transition kernel returns a value similar to the proposed transition kernel in the computation of the acceptance probability. The steps of this algorithm are shown in Figure 5. Occasionally, all weights can be zero, or a degenerate covariance matrix can be computed. We detect these cases, and revert to a von Mises–Fisher distribution aligned in the direction  $x'_{j-1} \rightarrow \bar{x}_j$  with a high concentration parameter for sampling. Another approach could be to convolve with an isotropic Gaussian similar to Li et al. [2015]; however, the value to use for the variance of the isotropic Gaussian is unclear in our case.

Once the anisotropic Gaussian is defined in world space, it is sampled producing a point  $z'_j$ . This point may not be aligned to the scene geometry, so a ray is traced from  $x_{j-1}$  in the direction  $x_{j-1} \rightarrow z'_j$ , producing a new point on the scene manifold  $x'_j$ . The

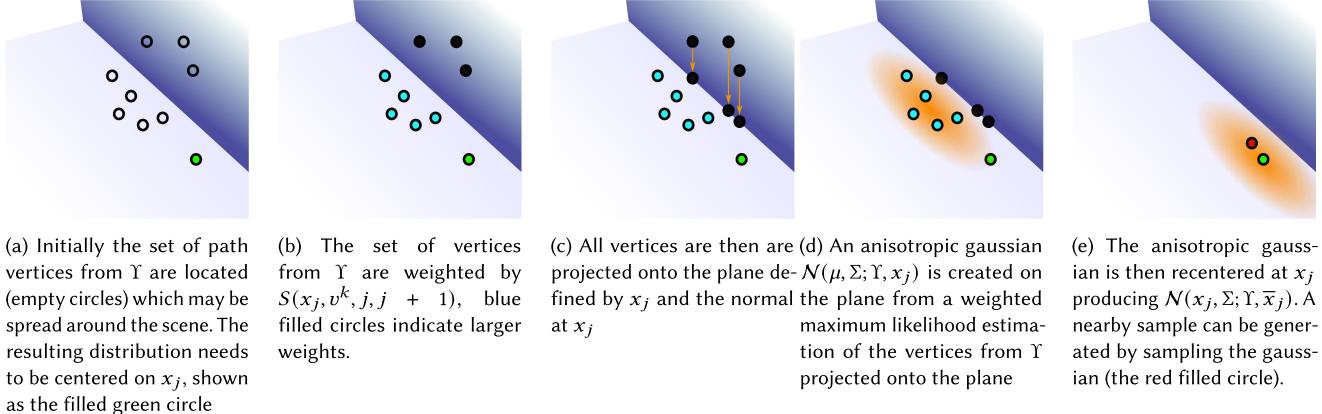


Fig. 5. The procedure to build and sample anisotropic Gaussian transition kernels.

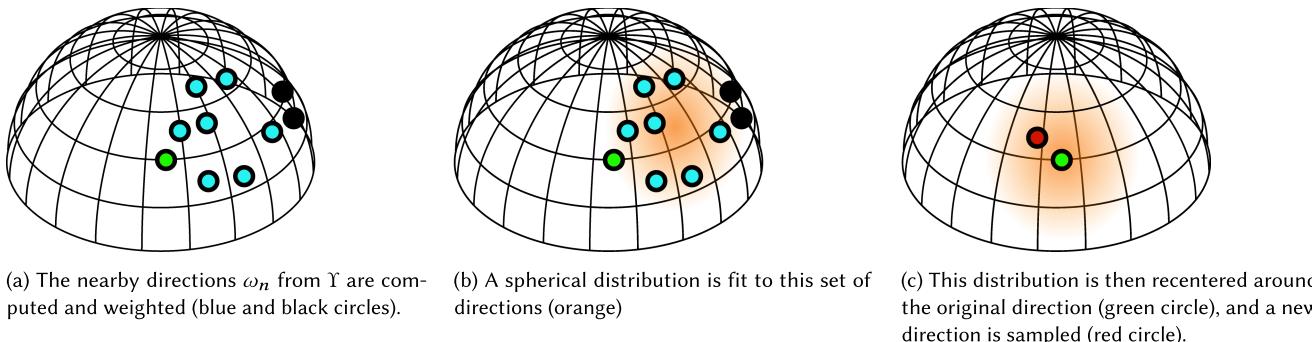


Fig. 6. Generating and sampling using the guided directional transition kernels.

density w.r.t. area of computing this point is  $K(\bar{x}_j^i \rightarrow \bar{x}'_j | \Upsilon) = N(z'_j | x_j, \Sigma; \Upsilon, x_j) \frac{G(x'_j \leftrightarrow x_{j-1})}{G(z'_j \leftrightarrow x_{j-1})}$ , where the final term stems from the ratio of geometry terms resulting from the Jacobian from sampling a point on the plane over which  $N(z'_j | x_j, \Sigma; \Upsilon, x_j)$  is defined, to the scene manifold.

To compute the acceptance probability, this process has to be computed in reverse; the distribution  $N(x'_j, \Sigma; \Upsilon, x'_j)$  is first computed, then the vertex  $x_j$  is projected onto the plane defined by  $x'_j$  and  $N(\bar{x}'_j)$  leading to a point  $z_j$ . This leads to a resulting ratio:

$$\frac{K(\bar{x}'_j \rightarrow \bar{x}_j^i | \Upsilon)}{K(\bar{x}_j^i \rightarrow \bar{x}'_j | \Upsilon)} = \frac{N(z_j | x'_j, \Sigma; \Upsilon, x'_j) G(z'_j \leftrightarrow x_{j-1}) G(x_j \leftrightarrow x_{j-1})}{N(z'_j | x_j, \Sigma; \Upsilon, x_j) G(z_j \leftrightarrow x_{j-1}) G(x'_j \leftrightarrow x_{j-1})}. \quad (12)$$

A simpler version of this approach can be used on the image plane. In this case, an anisotropic Gaussian can be fit to the image plane coordinates of the paths in  $\Upsilon$ , each weighted by the similarity measure. Again, this can be centered at the image plane coordinates of the current path, and a new point on the image plane for the proposed path can be sampled from this distribution.

**Guided Directional Transition Kernels.** Guided anisotropic transition kernels form an anisotropic distribution in world space. However, sometimes it is useful to sample perturbations over solid

angle. The linear hemispherical transition kernel performs this, but restricted along a great arc. Another approach is to fit a distribution on  $\mathbb{S}^2$ . Various approaches for this exist, for example tabulated, spherical Gaussian or a von Mises–Fisher distribution. Any distribution on the sphere whose parameters can be estimated from a set of directions can be used. Given a set of normalized directions from some base vertex  $x_j$  to each member of  $\Upsilon$ ,  $\omega_n = x_j \rightarrow v_j^n$ , and weights computed in the same manner as Equation (11), the parameters of a distribution can be estimated. Similar to the guided anisotropic perturbations, this distribution is recentered around the original direction from the vertex. This can then be sampled generating directions which are guided by nearby paths. Figure 6 shows this process.

### 3.3 Guided Perturbation Strategies

The previous section defined a range of guided transition kernels which are designed to update individual path vertices guided by global information from the ensemble. When perturbing a path, these guided transition kernels can be combined into a wide range of guided perturbation strategies designed to explore different lighting effects. Note that these can be combined with the original mutation and perturbation strategies; this simply adds to the strategies available. We always include the bidirectional mutation strategy from Veach and Guibas [1997] as this ensures ergodicity, thereby guaranteeing that the whole space will be explored.

The following lists the strategies we have implemented, but many more can be built using combinations of the kernels defined in Section 3.2.

#### Linear Lens Perturbation.

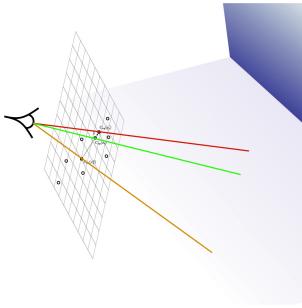


Fig. 7. Linear Lens Perturbation.

interaction types and also typically moves paths toward higher contribution regions for that path type. This strategy samples a path for the perturbation from  $\Upsilon$ , where the weight for the  $k$ 'th path is given by  $\frac{S(x^i, v^k, j, k)}{\sum_{p=1}^N S(x^i, v^p, j, k)}$ , where  $j$  is the index of a vertex on the camera, and  $k$  is the index of the first non-specular vertex in the path. Figure 7 illustrates this strategy.

#### Linear Caustic Perturbation.

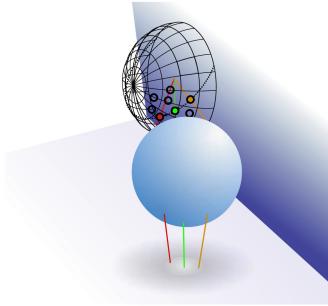


Fig. 8. Linear Caustic Perturbation.

of the scene. This first finds the best path from  $\Upsilon$  which closest matches the starting point and first specular vertex of the original caustic subpath, and sets the directions  $\omega_1 = x_c^i \rightarrow x_{c-1}^i$ , and  $\omega_2 = x_c^i \rightarrow v_{c-1}^n$ . This then perturbs the direction on the hemisphere, and traces the specular subpath to the first diffuse vertex, and connects to the camera. This is visualized in Figure 8.

#### Linear Multi-Chain Perturbation.

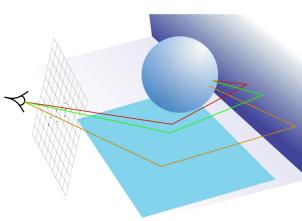


Fig. 9. Linear Multi-chain Perturbation.

The Linear Lens Perturbation uses the linear transition kernel on the image plane, then similar to Veach and Guibas [1997] traces a subpath over any specular vertices, then connects to the original path. As  $\Upsilon$  contains paths with similar interaction types and lengths to the current path, this strategy aims to explore the image plane around similar

is generated. A deterministic connection to the next specular subpath is then made and this process repeats until the path can be reconnected to the light subpath of the original path (see Figure 9).

#### Anisotropic Path Perturbation.

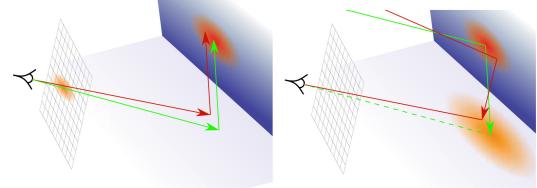


Fig. 10. The Anisotropic Path Perturbation uses the guided anisotropic perturbations for the first and second non-specular interactions from the camera. This can either start from the camera (left image), or toward the camera (right image).

This perturbation strategy comprises using the guided anisotropic transition kernels to perturb the current path. This can be applied to any number of path vertices, either from the light source or the eye. As the process of fitting an anisotropic Gaussian is relatively expensive, we restrict this perturbation to the first two vertices from the camera, and randomly select whether to sample from or toward the camera. If sampling from the camera is selected, an anisotropic Gaussian is created on the image plane and sampled, and for all other non-specular interactions the guided anisotropic transition kernels in world space are used, then reconnected to the original path. Likewise, if sampling toward the camera is selected, the guided anisotropic transition kernel is used to generate path vertices which are deterministically connected to the camera. This perturbation strategy helps to explore the local region around the path, and sampling more than one vertex from the camera helps to minimize the impact of the weak singularity in the geometry term near edges visible from the camera (see Figure 10). Another option is to sample with respect to solid angle similar to Otsu et al. [2018] or using the guided hemispherical perturbation which would cancel geometry terms in the calculation of the acceptance probability.

#### Environment Perturbation.

This perturbation is designed to explore environment lighting based on nearby paths. This perturbation uses the linear hemispherical transition kernel or the guided directional transition kernel applied toward the environment map to perturb the direction to the environment map, assuming that the first path vertex from the light is non-specular. This is illustrated in Figure 11. Variants of this strategy can also be applied to other area light sources, or for light source selection.

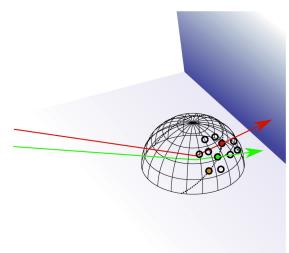


Fig. 11. Environment Perturbation. This figure illustrates the Environment Perturbation strategy. It shows a camera (lens) on the left, a light source on the right, and a scene in the middle. A path is shown starting from the light source, reflecting off a surface, and then being perturbed by a guided directional transition kernel applied toward the environment map to explore environment lighting. Variants of this strategy can also be applied to other area light sources, or for light source selection.

### 3.4 Implementation Details

This method is initialized similar to the resampling approach described in MLT [Veach and Guibas 1997]. A large set of paths is computed with bidirectional path tracing, the contributions of these paths stored and used to scale the result, then these paths are resampled into two subsets: one to generate the ensemble  $X$ , and the other used to generate the complementary ensemble  $Y$ .

There is no limit on the size of  $X$  and  $Y$ ; however, for convenience we choose them to be the same size of  $|X| = |Y| = 16,384$  (please see the supplementary material for further analysis). There is also much flexibility about when to update and swap ensembles (see Section 3 for details). Although updating the data structures for finding paths sampling is relatively inexpensive, this does come with some computational overhead of clearing out previous values and reinserting new values. Therefore, each path in the ensemble is perturbed or mutated  $it$  times (see Algorithm 1) before the ensembles are swapped, which amortizes the overhead of updating the data structure compared to swapping after every mutation or perturbation. We choose  $it$  to take a value of  $\lfloor \frac{W \times H}{|X| + |Y|} \rfloor$ , where  $W$  and  $H$  are the width and height of the image plane, respectively, which balances the computation cost of rebuilding the data structures and runtime performance.

There is significant freedom to choose the value of  $\alpha$  used for the linear transition kernels described in Section 3.2. However, the method does not work well if this is set to a constant, as if the paths in  $Y$  are clustered in a small region of space, e.g., in a small area on the image plane, then  $\alpha$  should be large to facilitate exploration of a small space. Conversely, when paths are spread over a large area  $\alpha$  should be small such that the proposed path is able to explore a similar region of the space. We solve this issue by adapting  $\alpha$  based on  $Y$ . For a linear perturbation on the lens, we first compute a ratio of the bounding box of the image plane coordinates of each path in  $Y$  to the image plane resolution:  $b_{res}$ .  $\alpha$  is then computed by linearly interpolating between two bounds  $\alpha_l$  and  $\alpha_s$  based on a weight  $w_\alpha = (1 + e^{-\frac{b_{res}-c}{\sigma^2}})^{-1}$ , where  $c \in [0..1]$ . This uses a generalized sigmoid as a weighting function as it gives control over where and how fast the weights transition from 0 to 1. We use the parameters  $\alpha_l = 0.5$ ,  $\alpha_s = 0.05$ ,  $c = 0.1$ , and  $\sigma^2 = 0.02$ , although the algorithm is quite robust to these values. The supplementary material provides further details on the impact of  $\alpha$ .

For the linear lens, caustic, and multi-chain perturbations, if the number of paths in  $Y$  is less than two, then the original perturbation strategies are used. This is to handle two situations: one is if no nearby paths are found, then the path can still be perturbed, and secondly, if only one path is found then there is too little information about nearby paths to create a useful sampling distribution. We also set the probabilities of sampling each proposed mutation type to be equal.

## 4 RESULTS

EMLT, MLT [Veach and Guibas 1997], and Geometry Aware MLT (GAMLT) [Otsu et al. 2018] were implemented into the same rendering framework for comparison. We implemented the Bidirectional, Lens, Caustic, and Multichain perturbations in MLT, and compare to GAMLT as it is the closest method to ours in terms

of using adaptive sized perturbations in path space. We tested the methods in a variety of scenes, from those which exhibit challenging light transport where MCMC methods are expected to perform well, to simpler scenes which represent more common use cases for rendering. All results were computed on a laptop with an i7-8750H and 16GB RAM. Computation was spread over 12 threads using a thread pool to process paths in parallel and the ensemble was the same size per scene (see Section 3.4 for more information). We set a constant probability for the bidirectional mutation of  $\frac{1}{3}$ . All results were rendered at an average of 64 mutations per pixel to allow equal comparison between methods.

### 4.1 Indirect Lighting

Our method is primarily focused on efficiently computing global illumination. Therefore, we first investigate the performance of EMLT in scenes with indirect lighting only, as direct lighting can be efficiently handled by other techniques in these scenes when compared to MLT. We show results for six scenes which exhibit different types of lighting effects. The DOOR AJAR scene in Figure 1 is a challenging scenario where light propagates through the ajar door. Similarly, the BEDROOM scene in Figure 15 has thick, diffuse curtains with a light source on the other side leading to a very challenging lighting scenario. The CLASSROOM scene in Figure 15 is lit by an environment map with light entering through the windows. The KITCHEN scene in Figure 15 shows strong indirect lighting on the back wall and glossy reflections. The CORNELL BOX scene in Figure 16 is representative of many real-world scenes with simple lighting configurations. Finally, the STAIRCASE scene exhibits simple indirect lighting above the stairs, and more complicated indirect lighting under the stairs. Insets in the images show details, and the values printed on the top left of full resolution images correspond to **Mean Squared Error (MSE)** for the whole image.

In Figure 17, we show loglog convergence plots for MSE versus average mutations per pixel for the scenes used in this article to show how error decreases. This shows that there is an improvement in convergence using EMLT (blue line) compared to MLT (green line) and GAMLT (red line; see below for further discussion). This can be seen in the rendered images as a reduction in noise compared to MLT. Diffuse and low glossy surfaces, such as the walls in the DOOR AJAR, CLASSROOM, and CORNELL BOX scenes, or behind the cooker in the KITCHEN scene, exhibit significantly reduced variance with EMLT. This is due to the transition kernels adapting to both the illumination and scene as encoded in the paths in the ensemble. However, EMLT also captures higher frequency lighting effects, and can adapt to higher glossy materials, as can be seen in the BEDROOM scene above the curtains, the metal on the chairs in the CLASSROOM scene, and the strong indirect lighting on the back wall in the KITCHEN scene. EMLT leads to improvements in MSE for all tested scenes compared to MLT: 2.23× for DOOR AJAR, 1.38× for the BEDROOM scene, 2.11× for the CLASSROOM scene, 1.44× for KITCHEN, 1.71× for the STAIRCASE scene, and 2.93× for the CORNELL BOX scene.



Fig. 12. Visualization of perturbations on the image plane for the KITCHEN scene. Green colors mean perturbations were predominantly vertical, red means predominantly horizontal, and yellow means perturbations were predominantly isotropic. The left image shows perturbations from MLT and the right shows our method.

Figure 12 visualizes the anisotropic nature of the perturbations proposed in this article for the KITCHEN scene; other scenes are shown in the supplementary material. The colors visualize the predominant direction of the perturbations on the image plane, green represents vertical, red shows horizontal, while yellow are isotropic. The perturbations for MLT are predominantly isotropic, as the lens and multi-chain perturbations use an isotropic transition kernel. In MLT, caustic perturbations are isotropic over the hemisphere, but do lead to an anisotropic distribution on the image plane. Our method in contrast adapts to both geometry and lighting information as can be seen in the right image in Figure 12. This shows that EMLT proposes perturbations which are predominantly horizontal in the strong horizontal indirect lighting on the rear wall, whereas on the rest of the wall the perturbations are predominantly vertical.

**4.1.1 Comparison to GAMLT.** Our implementation of GAMLT for indirect lighting used the geometry aware multi-chain perturbations with the same parameters as used for the results in Otsu et al. [2018], and we also extended this to use geometry aware lens and caustic perturbations following the same approach as described in Otsu et al. [2018]. We found this significantly improved the results in GAMLT and these extra perturbation strategies were implemented to facilitate a fair comparison between EMLT, MLT, and GAMLT. The convergence plots in Figure 17 show that EMLT outperforms GAMLT in several scenes, although both EMLT and GAMLT have similar variance reduction properties in the KITCHEN and CLASSROOM scenes. This is due to more small scale details in these scenes which GAMLT can adapt to, leading to approximately equal performance to EMLT. However, EMLT also adapts to these details, while also adapting to incident illumination. GAMLT is also significantly more computationally expensive than EMLT. In our implementation, we observed a 4 to 10 times increase in time to generate the same number of samples. Figure 18 shows results comparing EMLT to GAMLT for the STAIRCASE scene. Both EMLT and GAMLT are able to achieve variance reduction by considering scene geometry in the transition kernels, but EMLT is also able to better adapt to illumination, as can be seen in the inset images.

## 4.2 Ablation Study

We performed an ablation study to assess the impact of the proposed strategies. We rendered the scenes using either the Anisotropic Path Perturbation, the Linear Guided Perturbations, or the combination of these two strategies. Figure 13 shows zoomed

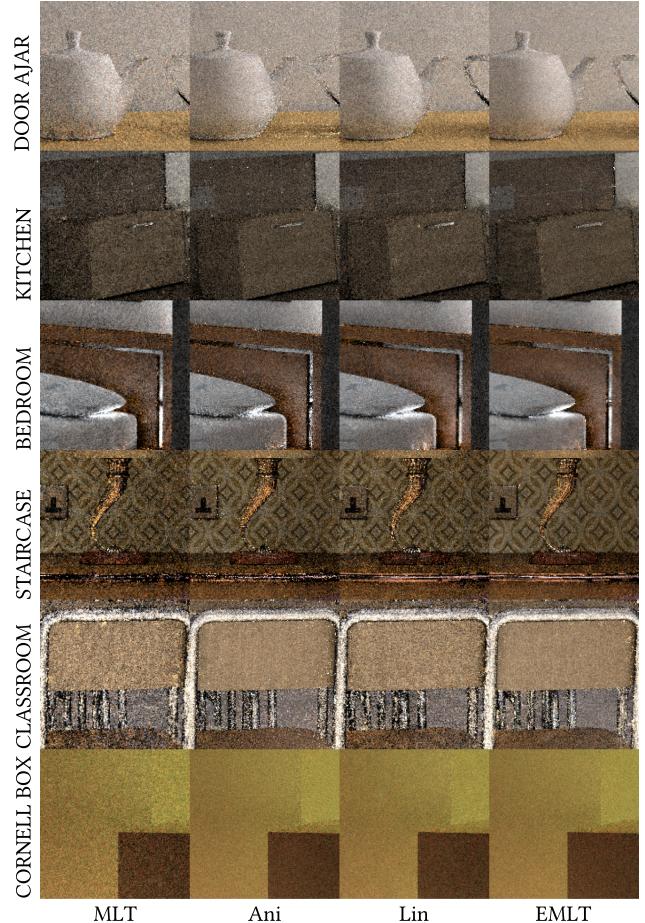


Fig. 13. Ablation study for the perturbation types presented in this article, where Ani refers to the Anisotropic Path Perturbation, Lin refers to the Linear Guided Perturbations, and EMLT refers to the combination. Combination refers to using all strategies presented in this article. The images show insets of the scenes used in this article, and MSE values are shown in Table 1.

Table 1. MSE Values for the Different Strategies Used in the Ablation Study (for the Scenes Used in this Article, see Figure 13 for Accompanying Images)

Scene	MLT	Ani	Lin	EMLT
DOOR AJAR	7.32e-3	4.14e-3	3.86e-3	3.28e-3
KITCHEN	2.19e-2	1.75e-2	1.56e-2	1.52e-2
BEDROOM	1.15e-4	9.28e-5	9.24e-5	8.34e-5
STAIRCASE	7.42e-5	5.07e-5	4.92e-5	4.34e-5
CLASSROOM	2.45e-2	1.26e-2	1.32e-2	1.16e-2
CORNELL BOX	9.62e-3	4.72e-3	4.34e-3	3.28e-3

in regions of the scenes used in this article highlighting the differences between strategies visually, and Table 1 provides MSE values for these strategies across all scenes. This shows that the Anisotropic Path Perturbation is effective at reducing noise in regions with low frequency variations in lighting and is responsible for an average of 63% improvement over MLT in the scenes

Table 2. MSE Values for Direct Lighting

Scene	MIS	MLT	EMLT
BREAKFAST ROOM	0.1082	0.0213	0.0177
SPONZA	0.0193	0.0038	0.0031

in this article. However, this exhibits noise when sampling higher frequency lighting as the subset of paths contained in  $\Upsilon$  are less likely to be able to capture this type of lighting effect. Conversely, the Linear Lens, Caustic, and Multichain Perturbations are efficient at capturing these higher frequency effects and are responsible for an average of 68% improvement over MLT, but exhibit more noise in more uniform regions of the scene. The combination of these strategies is able to reduce noise in both low and high frequency variations in lighting.

#### 4.3 Direct Lighting

To more clearly demonstrate the performance of the environment perturbation, we apply our method to direct lighting. This uses a combination of the Guided Lens Perturbation to perturb path position on the image plane, and the Environment Perturbation to guide sampling on the environment map. We compare with a traditional approach of BRDF and environment sampling combined with MIS, MLT applied to direct lighting, and EMLT. Please note, this is not meant to compete with specialized direct lighting approaches, but to illustrate the proposed perturbation strategies. In Figure 19, we show results for the BREAKFAST ROOM and SPONZA scene, only showing direct illumination from the environment map. Table 2 shows MSE for the scenes and sampling techniques. This shows that our perturbations outperform MLT, and significantly improve on using MIS for direct lighting.

#### 4.4 Performance

Our method has some computational overhead compared to MLT. On average, we observed an 18% overhead with our method due to (a) rebuilding the pools of paths (see Section 3.1.1), although this is amortized by infrequently updating the pools, and (b) slightly more complicated procedures for sampling perturbations. The majority of the overhead in our implementation comes from computing the similarity measure (Section 3.1.2). The impact of this overhead is illustrated in Figure 14, where we show the results for equal time versus equal quality for the scenes where our method performs best (DOOR AJAR) and worst (KITCHEN). The images show the same insets corresponding to Figures 1 and 15, respectively. These results show that the overhead of our method is outweighed by the variance reduction of guided transition kernels in EMLT.

EMLT also proposes perturbations which are more likely to be accepted than with MLT. For the scenes in this article, we observed a 14% increase in the acceptance probability averaged over all the scenes.

## 5 DISCUSSION AND FUTURE WORK

In this section, we briefly discuss our method, and propose directions to extend this work.

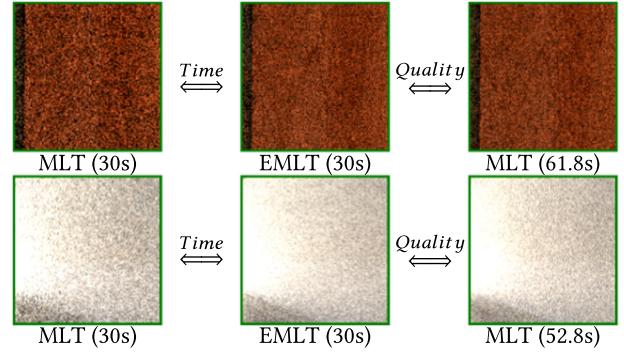


Fig. 14. Equal time versus equal quality for the DOOR AJAR and KITCHEN scenes. The images correspond to the same insets as the main results. The left column shows MLT rendered for 30 s, the middle is EMLT rendered for 30 s, and the right images show MLT rendered to the same MSE for the whole image.

**Derivative-Based Approaches.** Our method guides sampling based on creating distributions from an ensemble of paths which capture lighting information in the region near the current path. The size of this region depends on the number of paths selected from the pool ( $\Upsilon$ ), and the size of the pool. Approaches such as Li et al. [2015] and Luan et al. [2020] use gradient information associated with the path to create proposal distributions which allow perturbations to be proportional to the local gradient. Our work is complementary to these approaches as we target perturbations guided over a wider region, and as such can take into account larger scale geometric and lighting details, whereas these approaches allow for more optimal local perturbations but do not consider lighting from nearby paths, or geometrical detail. Combining both approaches would be an interesting avenue for future work.

**Combination with Primary Sample Space.** Our approach works in world space as discussed in Section 3. However, using the approaches which fuse world space Metropolis Light Transport and PSSMLT [Bitterli et al. 2018; Otsu et al. 2017; Pantaleoni 2017] would allow our approach to be combined with PSS approaches. Another approach would be to adapt our method to work in PSS; however, this is not trivial as discussed in Section 3.

**Combination with Other Mutation Strategies.** Perturbation strategies such as Manifold Perturbations [Jakob and Marschner 2012], Multiple Try Metropolis [Nimier-David et al. 2019; Segovia et al. 2007], selectively choosing paths to perturb [Bitterli and Jarosz 2019], and Delayed Rejection MLT [Rioux-Lavoie et al. 2020] could all be combined with our approach into a larger set of possible strategies. Manifold perturbations are especially effective at locally perturbing specular paths, thus complementary to our approach which perturbs paths in a wider region. Delayed Rejection MLT would help balance between when to use the different strategies, leading to a more efficient method.



Fig. 15. Results for the BEDROOM, KITCHEN, and CLASSROOM scenes. The left images are the reference, middle is MLT, and the right images are EMLT. Insets highlight reduced variance with our method, for example showing where the sampling has adapted to scene geometry or illumination.

**Parameters.** Our method requires several parameters such as pool size, update frequency, the number of paths in  $\Upsilon$ , and the parameters used to compute  $\alpha$ . We discuss these parameters in Section 3.4; however, we do not claim these parameters are optimal. Theoretically finding optimal values of these parameters would be useful as it would further increase the efficiency of our method. One possibility is to use the scene acceleration structure to estimate maximum values for the parameters used in the same manner as the approach taken by Otsu et al. [2018].

**Limitations.** While our method achieves variance reduction for scenes with both complicated and simple lighting, there are some situations or sets of parameters where our method is outperformed

by MLT. An example of this is when the size of the ensemble becomes very small. In this case, there is not enough information in the complementary ensemble to guide sampling, and our method falls back to MLT, albeit with the computational overhead of maintaining a pool. We found this was not a problem using the range of parameters outlined in Section 3.4, but there may be scenes which require the ensemble to represent more paths.

Finally, the approach of deterministically selecting a subset of paths from an ensemble of paths could be used for path guiding in non-MCMC methods such as path tracing. This would have the advantage of no longer requiring an additional spatial data structure as is needed by the approaches in Section 2.2, and would likely require different probability distributions than those used

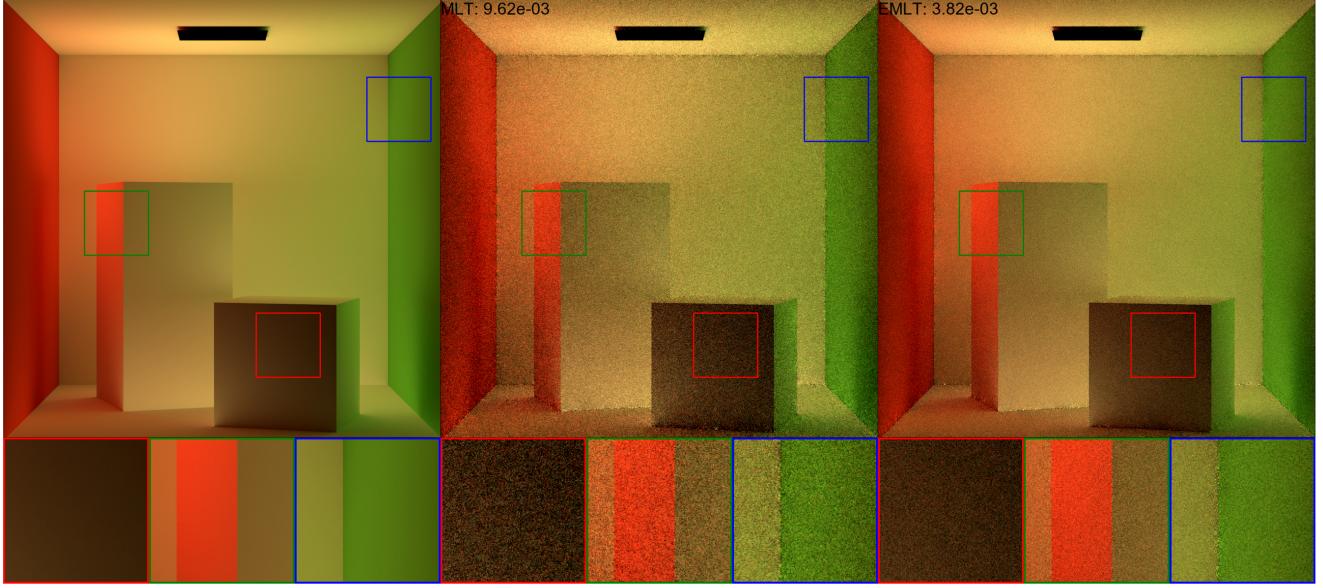


Fig. 16. Results for the CORNELL BOX scene. This illustrates that EMLT provides an advantage over MLT in simple scenes.

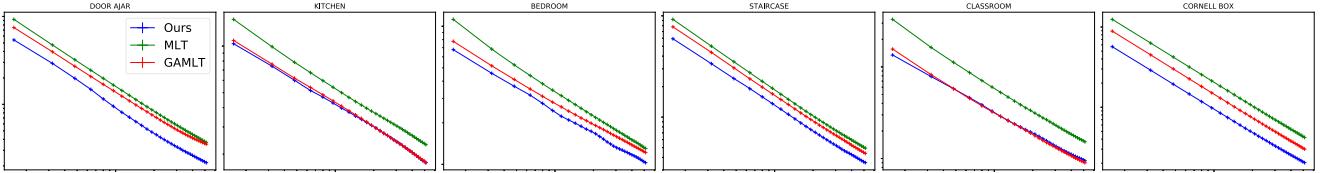


Fig. 17. Convergence graphs showing MSE versus average mutations per pixel on a logarithmic scale for the scenes used in this article. The blue line is EMLT, green is MLT, and red is GAMLT.



Fig. 18. Results for the STAIRCASE scene showing the reference on the left, GAMLT in the middle, and EMLT on the right. This shows that both EMLT and GAMLT can adapt transition kernels to the scene geometry, but EMLT can also sample illumination, leading to variance reduction.

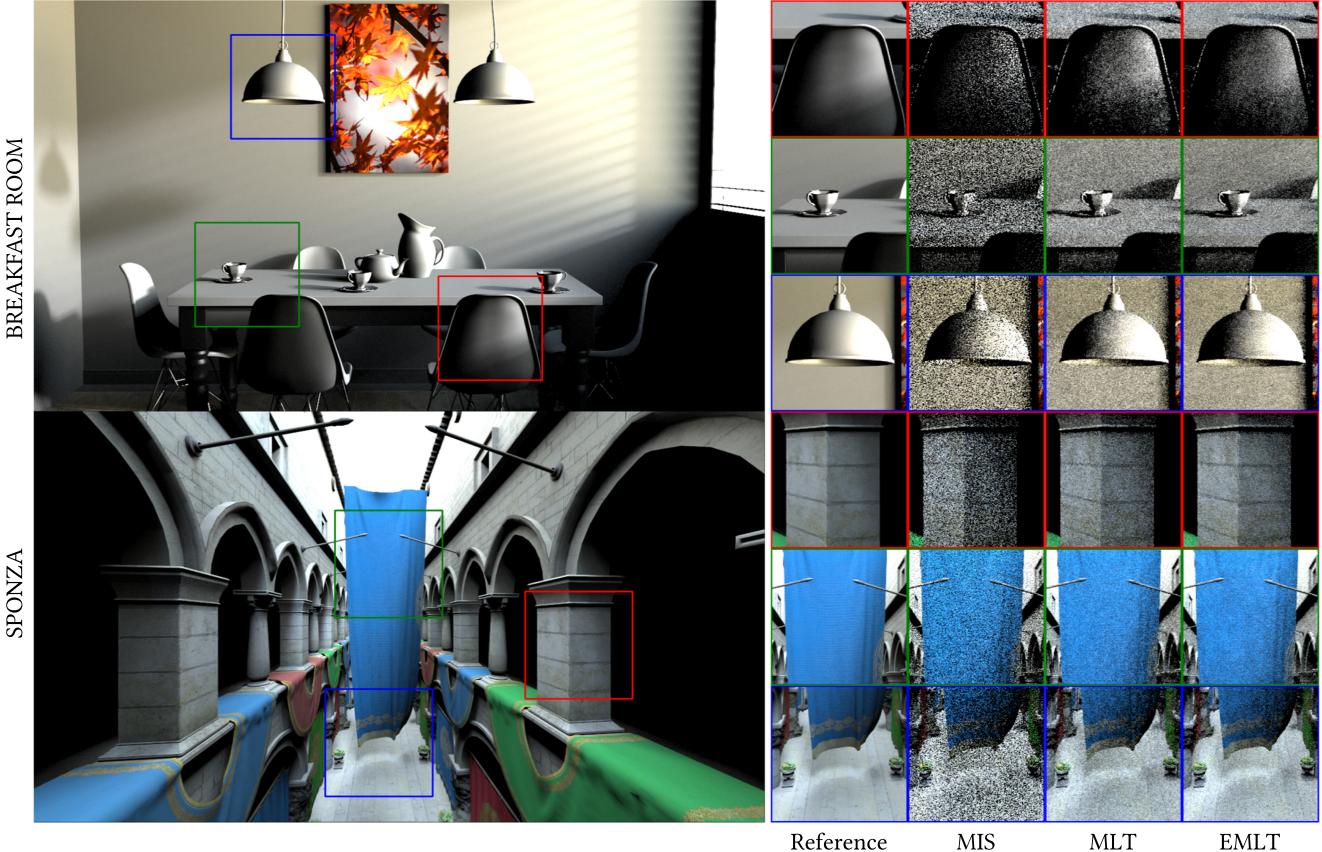


Fig. 19. Environment Lighting in the BREAKFAST ROOM scene (top row) and SPONZA (bottom row). The inset images show (left to right): Reference, MIS, MLT, and EMLT. Our guided perturbations reduce variance compared to MLT, and significantly compared to MIS.

in this article. However, we believe this could lead to conceptually simpler and easier to implement path guiding with lower memory overheads.

## 6 CONCLUSION

This article has presented a new family of transition kernels for MCMC rendering algorithms. These are based on efficiently sampling ensembles of transport paths, and utilizing these ensembles to guide path mutations. This approach does not require spatial caching of radiance or importance distributions, nor the associated spatial data structures, yet is efficient and reduces variance in scenes of different complexity and light transport effects. We believe that many more transport kernels of the type presented in this article are possible, and we hope this work opens up new possibilities for further variance reduction strategies for MCMC methods in the future.

## ACKNOWLEDGMENTS

We would like to thank the reviewers for their helpful comments and feedback. We also thank *Benedikt Bitterli* for the DOOR AJAR and CORNELL BOX scenes, *SlykDrako* for the BEDROOM, *Jay-Artist* for KITCHEN, *NovaZeeke* for CLASSROOM, *Wig42*

for STAIRCASE and BREAKFAST ROOM, *Frank Meinl* for the SPONZA scene, and *Benedikt Bitterli* for making these scenes available [Bitterli 2016].

## REFERENCES

- Steve Bakó, Mark Meyer, Tony DeRose, and Pradeep Sen. 2019. Offline deep importance sampling for Monte Carlo path tracing. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 527–542.
- Thomas Bashford-Rogers, Kurt Debattista, and Alan Chalmers. 2012. A significance cache for accelerating global illumination. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 1837–1851.
- Benedikt Bitterli. 2016. Rendering resources. Retrieved on 14 March, 2020 from <https://benedikt-bitterli.me/resources/>.
- Benedikt Bitterli, Wenzel Jakob, Jan Novák, and Wojciech Jarosz. 2018. Reversible jump metropolis light transport using inverse mappings. *ACM Transactions on Graphics (TOG)* 37, 1 (2018), 1.
- Benedikt Bitterli and Wojciech Jarosz. 2019. Selectively metropolised Monte Carlo light transport simulation. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–10.
- Olivier Cappé, Arnaud Guillin, Jean-Michel Marin, and Christian P. Robert. 2004. Population Monte Carlo. *Journal of Computational and Graphical Statistics* 13, 4 (2004), 907–929.
- Chakravarthy R. Alla Chaitanya, Laurent Belcour, Toshiya Hachisuka, Simon Premoze, Jacopo Pantaleoni, and Derek Nowrouzezahrai. 2018. Matrix bidirectional path tracing. In *Proceedings of the Eurographics Symposium on Rendering: Experimental Ideas & Implementations*. Eurographics Association, 23–32.
- Per H. Christensen and Wojciech Jarosz. 2016. The path to path-traced movies. *Foundations and Trends® in Computer Graphics and Vision* 10, 2 (2016), 103–175.

- David Cline, Justin Talbot, and Parris Egbert. 2005. Energy redistribution path tracing. In *ACM Transactions on Graphics (TOG)*, Vol. 24. ACM, 1186–1195.
- Ken Dahm and Alexander Keller. 2017. Learning light transport the reinforced way. arXiv:1701.07403. <https://arxiv.org/abs/1701.07403>.
- Stavros Diolatzis, Adrien Gruson, Wenzel Jakob, Derek Nowrouzezahrai, and George Drettakis. 2020. Practical product path guiding using linearly transformed cosines. In *Computer Graphics Forum*, Vol. 39.
- ShaoHua Fan, Yu-Chi Lai, Stephen Chenney, and Charles Dyer. 2007. *Population Monte Carlo Samplers for Rendering*. Technical Report. University of Wisconsin-Madison, Department of Computer Sciences.
- Daniel Foreman-Mackey, David W. Hogg, Dustin Lang, and Jonathan Goodman. 2013. emcee: The MCMC hammer. *Publications of the Astronomical Society of the Pacific* 125, 925 (2013), 306.
- Jonathan Goodman and Jonathan Weare. 2010. Ensemble samplers with affine invariance. *Communications in Applied Mathematics and Computational Science* 5, 1 (2010), 65–80.
- Adrien Gruson, Mickaël Ribardière, Martin Šík, Jiří Vorba, Rémi Cozot, Kadi Bouatouch, and Jaroslav Krivánek. 2016. A spatial target function for metropolis photon tracing. *ACM Transactions on Graphics (TOG)* 36, 4 (2016), 1.
- Adrian Gruson, Rex West, and Toshiya Hachisuka. 2020. Stratified Markov chain Monte Carlo light transport. In *Eurographics*. The Eurographics Association.
- Jerry Guo, Pablo Bauszat, Jacco Bikker, and Elmar Eisemann. 2018. Primary sample space path guiding. In *Eurographics Symposium on Rendering*, Vol. 2018. The Eurographics Association, 73–82.
- Toshiya Hachisuka and Henrik Wann Jensen. 2011. Robust adaptive photon tracing using photon path visibility. *ACM Transactions on Graphics (TOG)* 30, 5 (2011), 114.
- Toshiya Hachisuka, Anton S. Kaplanyan, and Carsten Dachsbacher. 2014. Multiplexed metropolis light transport. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 100.
- W. Keith Hastings. 1970. Monte Carlo sampling methods using Markov chains and their applications.
- Paul S. Heckbert. 1990. Adaptive radiosity textures for bidirectional ray tracing. *ACM SIGGRAPH Computer Graphics* 24, 4 (1990), 145–154.
- Sebastian Herholz, Oskar Elek, Jiří Vorba, Hendrik Lensch, and Jaroslav Krivánek. 2016. Product importance sampling for light transport path guiding. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 67–77.
- Heinrich Hey and Werner Purgathofer. 2002. Importance sampling with hemispherical particle footprints. In *Proceedings of the 18th Spring Conference on Computer Graphics*. ACM, 107–114.
- Jared Hoberock and John C. Hart. 2010. Arbitrary importance functions for metropolis light transport. In *Computer Graphics Forum*, Vol. 29. Wiley Online Library, 1993–2003.
- Wenzel Jakob and Steve Marschner. 2012. Manifold exploration: A Markov chain Monte Carlo technique for rendering scenes with difficult specular transport. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 58.
- Henrik Wann Jensen. 1995. Importance driven path tracing using the photon map. In *Rendering Techniques '95*. Springer, 326–335.
- James T. Kajiya. 1986. The rendering equation. In *ACM SIGGRAPH Computer Graphics*, Vol. 20. ACM, 143–150.
- Anton S. Kaplanyan, Johannes Hanika, and Carsten Dachsbaucher. 2014. The natural-constraint representation of the path space for efficient light transport simulation. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 102.
- Csaba Kelemen, László Szirmay-Kalos, György Antal, and Ferenc Csonka. 2002. A simple and robust mutation strategy for the metropolis light transport algorithm. In *Computer Graphics Forum*, Vol. 21. Wiley Online Library, 531–540.
- Alexander Keller. 1997. Instant radiosity. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'97)*. ACM Press/Addison-Wesley Publishing Co., 49–56. <https://doi.org/10.1145/258734.258769>
- Shinya Kitaoka, Yoshifumi Kitamura, and Fumio Kishino. 2009. Replica exchange light transport. In *Computer Graphics Forum*, Vol. 28. Wiley Online Library, 2330–2342.
- Eric P. Lafontaine and Yves D. Willems. 1995. A 5D tree to reduce the variance of Monte Carlo ray tracing. In *Rendering Techniques '95*. Springer, 11–20.
- Yu-Chi Lai, Shao Hua Fan, Stephen Chenney, and Charles Dyer. 2007. Photorealistic image rendering with population monte carlo energy redistribution. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*. Eurographics Association, 287–295.
- Tzu-Mao Li, Jaakkko Lehtinen, Ravi Ramamoorthi, Wenzel Jakob, and Frédo Durand. 2015. Anisotropic Gaussian mutations for metropolis light transport through Hessian-Hamiltonian dynamics. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 209.
- Jun S. Liu. 2008. *Monte Carlo Strategies in Scientific Computing*. Springer Science & Business Media.
- Fujun Luan, Shuang Zhao, Kavita Bala, and Ioannis Gkioulekas. 2020. Langevin Monte Carlo rendering with gradient-based adaptation. *ACM Transactions on Graphics* 39, 4 (2020).
- Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. 1953. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* 21, 6 (1953), 1087–1092.
- Thomas Müller, Markus Gross, and Jan Novák. 2017. Practical path guiding for efficient light-transport simulation. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 91–100.
- Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. 2018. Neural importance sampling. arXiv:1808.03856. <https://arxiv.org/abs/1808.03856>
- Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. 2019. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 203.
- Hisanari Otsu, Johannes Hanika, Toshiya Hachisuka, and Carsten Dachsbaucher. 2018. Geometry-aware metropolis light transport. In *SIGGRAPH Asia 2018 Technical Papers*. ACM, 278.
- Hisanari Otsu, Anton S. Kaplanyan, Johannes Hanika, Carsten Dachsbaucher, and Toshiya Hachisuka. 2017. Fusing state spaces for Markov chain Monte Carlo rendering. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 74.
- Hisanari Otsu, Yonghao Yue, Qiming Hou, Kei Iwasaki, Yoshinori Dobashi, and Tomoyuki Nishita. 2013. Replica exchange light transport on relaxed distributions. In *ACM SIGGRAPH 2013 Posters*. ACM, 106.
- Jacopo Pantaleoni. 2017. Charted Metropolis light transport. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 75.
- Jacopo Pantaleoni. 2020. Online path sampling control with progressive spatio-temporal filtering. *SN Computer Science* 1, 5 (Aug. 2020), 279. <https://doi.org/10.1007/s42979-020-00291-z>
- Mark Pauly, Thomas Kollig, and Alexander Keller. 2000. Metropolis light transport for participating media. In *Rendering Techniques 2000*. Springer, 11–22.
- Florian Reibold, Johannes Hanika, Alisa Jung, and Carsten Dachsbaucher. 2018. Selective guided sampling with complete light transport paths. In *SIGGRAPH Asia 2018 Technical Papers*. ACM, 223.
- Damien Rioux-Lavoie, Joey Litthalen, Adrien Gruson, Toshiya Hachisuka, and Derek Nowrouzezahrai. 2020. Delayed rejection metropolis light transport. *ACM Transactions on Graphics* 39, 3, Article 26 (April 2020). <https://doi.org/10.1145/3388538>
- Lukas Ruppert, Sebastian Herholz, and Hendrik P. A. Lensch. 2020. Robust fitting of parallax-aware mixtures for path guiding. *ACM Transactions on Graphics* 39, 4 (July 2020), Article 147, 15 pages. <https://doi.org/10.1145/3386569.3392421>
- Benjamin Segovia, Jean Claude Iehl, and Bernard Péroche. 2007. Metropolis instant radiosity. In *Computer Graphics Forum*, Vol. 26. Wiley Online Library, 425–434.
- Martin Šík and Jaroslav Krivánek. 2016. Improving global exploration of MCMC light transport simulation. In *ACM SIGGRAPH 2016 Posters*. ACM, 50.
- Martin Šík and Jaroslav Krivánek. 2018. Survey of Markov chain Monte Carlo methods in light transport simulation. *IEEE Transactions on Visualization and Computer Graphics* 26, 4 (2018), 1821–1840.
- Martin Šík, Hisanari Otsu, Toshiya Hachisuka, and Jaroslav Krivánek. 2016. Robust light transport simulation via metropolised bidirectional estimators. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 245.
- Robert H. Swendsen and Jian-Sheng Wang. 1986. Replica Monte Carlo simulation of spin-glasses. *Physical Review Letters* 57, 21 (1986), 2607.
- Eric Veach and Leonidas J. Guibas. 1995. Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*. ACM, 419–428.
- Eric Veach and Leonidas J. Guibas. 1997. Metropolis light transport. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press/Addison-Wesley Publishing Co., 65–76.
- Jiří Vorba, Ondřej Karlík, Martin Šík, Tobias Ritschel, and Jaroslav Krivánek. 2014. Online learning of parametric mixture models for light transport simulation. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 101.
- Quan Zheng and Matthias Zwicker. 2019. Learning to importance sample in primary sample space. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 169–179.

Received September 2020; revised April 2021; accepted June 2021

# Spherical Fibonacci Point Sets for Illumination Integrals

R. Marques<sup>1</sup>, C. Bouville<sup>2</sup>, M. Ribardière<sup>2</sup>, L. P. Santos<sup>3</sup> and K. Bouatouch<sup>2</sup>

<sup>1</sup>INRIA Rennes, France

ricardo.marques@inria.fr

<sup>2</sup>IRISA Rennes, France

{christian.bouville, mickael.ribardiere, kadi.bouatouch}@irisa.fr

<sup>3</sup>Universidade do Minho, Braga, Portugal

psantos@di.uminho.pt

## Abstract

Quasi-Monte Carlo (QMC) methods exhibit a faster convergence rate than that of classic Monte Carlo methods. This feature has made QMC prevalent in image synthesis, where it is frequently used for approximating the value of spherical integrals (e.g. illumination integral). The common approach for generating QMC sampling patterns for spherical integration is to resort to unit square low-discrepancy sequences and map them to the hemisphere. However such an approach is suboptimal as these sequences do not account for the spherical topology and their discrepancy properties on the unit square are impaired by the spherical projection. In this paper we present a strategy for producing high-quality QMC sampling patterns for spherical integration by resorting to spherical Fibonacci point sets. We show that these patterns, when applied to illumination integrals, are very simple to generate and consistently outperform existing approaches, both in terms of root mean square error (RMSE) and image quality. Furthermore, only a single pattern is required to produce an image, thanks to a scrambling scheme performed directly in the spherical domain.

**Keywords:** Monte Carlo techniques, spherical integration, global illumination, rendering, ray tracing

**ACM CCS:** I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism-Raytracing

## 1. Introduction

Among all the methods which have been proposed to speed up Monte Carlo integration for rendering, Quasi-Monte Carlo (QMC) methods play an important role as they allow improving the convergence rate as well as controlling the error noise perception. The principle is to use more regularly distributed sample sets (i.e. with some determinism) than the crude random sample sets associated with Monte Carlo integration.

QMC integration is now extensively used in computer graphics (see e.g. [SEB08]). Keller has shown in [Kel12] that QMC techniques can be applied in a consistent way to deal with a wide range of problems (anti-aliasing, depth of field, motion blur, spectral rendering, . . .). However, few applications have been reported in the literature specifically addressing hemispherical sampling with a view of computing the illumination integral. Unlike the unit square sampling case, no explicit construction of optimal point sets for spherical sampling is known and generally the spherical point sets are generated by lifting point sets from the unit square to the unit sphere through an equal-area transform. Although such point con-

structions are not proved to be optimal, recent results from the numerical analysis literature suggest that both (0, 2)-sequences and Fibonacci lattices lifted to the sphere are quite close to optimality in terms of discrepancy [ABD12], [BD11]. Nevertheless their performance is not exactly equivalent: several authors have shown that spherical Fibonacci (SF) lattices are particularly well-suited to sphere sampling compared to other low-discrepancy point sets. Furthermore, similar point structures arise spontaneously in nature so as to implement a best packing strategy on the sphere (e.g. packing of seeds in the sunflowers head [Vog79]), a clear indication that these structures have intrinsically good spherical uniformity properties.

In this paper, we introduce theoretical aspects on QMC spherical integration that, to the authors knowledge, have never been used in the graphics community. In concrete terms, we define worst case integration error (w.c.e.), spherical cap discrepancy (s.c.d.) and an inter-samples distance-based energy metric  $E_N$ , which allows to assess the quality of a spherical samples set for spherical integration.

The second and major contribution of this work is the introduction of the Fibonacci point sets for spherical quadrature, based on

previous works [HN04], [SJP06]. We compare the quality of Fibonacci point sets for estimating the illumination integral with that of state-of-the-art QMC-compliant point set distributions such as blue noise [dGBOD12], Larcher–Pillichshammer point sets [LP01] and the popular Sobol (0,2)-sequence [Sob67]. We show that the Fibonacci point sets consistently outperform those methods and that the improvement is, in general, remarkable in terms of root mean square error (RMSE) value and percentage of rays saved for the same RMSE quality. The noise perception in the resulting images is also reduced. Furthermore, the generation of the Fibonacci point sets is much simpler than the other tested methods, and a single sequence is needed to synthesize an image.

The rest of this paper is structured as follows. In the next section, we introduce theoretical concepts regarding spherical integration using QMC and present the related work. It is followed by a detailed description of the Fibonacci spherical point sets. In Section 4, we specify how we have implemented bidirectional reflectance distribution function (BRDF) sampling in the context of QMC integration and make explicit the interest in generating high-quality spherical distributions for this particular case. Sections 5 and 6 present the benefits of using Fibonacci point sets compared to a Sobol sequence, blue noise point sets and the Larcher–Pillichshammer points. We finish with a conclusion and future work.

## 2. Background

### 2.1. QMC spherical integration

The goal of QMC integration is to find sampling patterns that yield a better order of convergence than the  $\mathcal{O}(N^{-1/2})$  rate obtained with purely random distributions. In the case of QMC integration over the unit square  $[0, 1]^2$ , it is well known that the best theoretical rate of convergence of the worst case error is  $\mathcal{O}(N^{-1}\sqrt{\log N})$  (see e.g. [BD11]). To find point set constructions that approximate this optimal rate of convergence, the star discrepancy is often used as a criterion to characterize the uniformity of the point distribution (the connection between this criterion and the worst case error is given by the Koksma–Hlawka inequality [Nie88], [BD11]). Moreover, a point set construction is called a low-discrepancy sequence when its unit square discrepancy convergence rate towards 0 is of order  $\mathcal{O}(N^{-1}(\log N)^2)$ .

Unlike the unit square case, QMC rules for numerical integration over the unit sphere  $\mathbb{S}^2$  in  $\mathbb{R}^3$  are less known to the graphics community. Therefore, a brief presentation of important results on this subject will be given in the following of this section.

A set of sampling directions  $\{\omega_{1,N}, \dots, \omega_{N,N}\}$  defined as points on the unit sphere  $\mathbb{S}^2$  is appropriate for Monte Carlo integration if it is *asymptotically uniformly distributed*, that is, if

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{j=1}^N f(\omega_{j,N}) = \frac{1}{4\pi} \int_{\mathbb{S}^2} f(\omega) d\Omega(\omega) \quad (1)$$

is true for every function  $f(\omega)$  on the sphere  $\mathbb{S}^2$ ,  $\Omega$  being the surface measure on  $\mathbb{S}^2$ . Similarly to the unit square case, this property is equivalent to:

$$\lim_{N \rightarrow \infty} \frac{\text{Card}\{j : \omega_{j,N} \in \mathcal{C}\}}{N} = \frac{\Omega(\mathcal{C})}{4\pi}, \quad (2)$$

for every spherical cap  $\mathcal{C}$  with area  $\Omega(\mathcal{C})$  [KN06]. Informally speaking, Equation (2) means that a spherical cap of any area has its fair share of points as  $N \rightarrow \infty$ . Among all sampling patterns complying with this definition, we are interested in point sets  $P_s = \{\omega_{1,N}, \dots, \omega_{N,N}\} \subseteq \mathbb{S}^2$  such that the w.c.e.

$$\text{w.c.e.} := e(P_s) = \sup_f \left| \frac{1}{N} \sum_{j=1}^N f(\omega_{j,N}) - \frac{1}{4\pi} \int_{\mathbb{S}^2} f(\omega) d\Omega(\omega) \right|$$

achieves the best rate of convergence as  $N \rightarrow \infty$ . This is equivalent to finding the point sets  $P_s$  which minimize the s.c.d. defined as follows:

$$\text{s.c.d.} := D(P_s; \mathcal{C}) = \sup_{\mathcal{C} \subseteq \mathbb{S}^2} \left| \frac{\text{Card}\{j : \omega_{j,N} \in \mathcal{C}\}}{N} - \frac{\Omega(\mathcal{C})}{4\pi} \right|,$$

where the supremum is extended over all spherical caps  $\mathcal{C} \in \mathbb{S}^2$ . The mathematical relationship between w.c.e. and s.c.d. is more complex than in the unit square case as explained in [BD11], [BSSW12]. Minimizing the s.c.d. is still equivalent to minimizing the w.c.e. However, both criteria only follow the same  $\mathcal{O}(N^{-3/4})$  rate of convergence towards 0 if  $f$  fulfills a specific smoothness criterion. Roughly speaking, it must be at least a  $C_0$  continuous function. In such a case, in application of the Stolarsky's invariance principle, the w.c.e. is proportional to the distance-based energy metric  $E_N$  [BD11], [BSSW12] given by

$$E_N(P_s) = \left( \frac{4}{3} - \frac{1}{N^2} \sum_{j=1}^N \sum_{i=1}^N |\omega_i - \omega_j| \right)^{\frac{1}{2}}, \quad (3)$$

which means that minimizing the w.c.e. is equivalent to maximizing the sum of distances term  $\sum_{j=1}^N \sum_{i=1}^N |\omega_i - \omega_j|$  while keeping the property of asymptotically uniform distribution.  $E_N$  can also be interpreted as an optimal spherical packing criterion [SK97].

The order of convergence of the w.c.e. can be higher than  $\mathcal{O}(N^{-3/4})$  if the order of continuity of the integrand is higher than  $C_0$ , but this depends on the points construction algorithm since some are more capable of taking advantage of smooth functions than others as explained in [BSSW12].

The s.c.d. order of convergence cannot be better than  $\mathcal{O}(N^{-3/4})$  but there surely exist point sets for which the order of convergence is better than  $\mathcal{O}(N^{-3/4}\sqrt{\log N})$  [Bec84], in which case these configurations are said to be *low-discrepancy sequences*. Note that this order of convergence is lower than the  $\mathcal{O}(N^{-1}(\log N)^2)$  rate of low-discrepancy sequences in the  $[0, 1]^2$  unit square.

In contrast with the unit square case, no explicit direct construction of low-discrepancy sequences on the unit sphere is known. That is why QMC sequences on  $\mathbb{S}^2$  are generally produced by lifting a  $[0, 1]^2$  low discrepancy point set to  $\mathbb{S}^2$  through an equal-area transform. An alternative to this approach consists in generating the patterns directly on the sphere according to an extremal energy criterion [SK97]. Among the patterns with good  $E_N$  properties, SF point sets (or equivalently generalized spiral points) are particularly well suited to QMC integration over the sphere as shown in [HN04], hence our interest in applying them for illumination integral computation.

## 2.2. Related work

The use of low-discrepancy sequences is widespread in computer graphics [KPR12]. Their goal is to improve the convergence rate of the integral estimate by using sample sets which minimize a discrepancy criterion. Among the most popular low-discrepancy sequences is the Sobol's (0,2)-sequence [Sob67] which guarantees both minimum distance and stratification criteria in each successive set of  $b^m$  samples, where  $b$  is the base of the sequence. Lower unit square discrepancy values can be obtained using the Larcher–Pillichshammer point sets [LP01], however these points cannot be generated by an (infinite) sequence. Moreover, Kollig *et al.* [KK02] showed that both sequences can be easily scrambled to decorrelate directions for neighbouring pixels, thus avoiding artefacts without sacrificing the discrepancy and stratification properties.

An alternative approach for producing uniform point set distributions on a unit square is to use a blue noise generator [LD08]. This class of point set generators produces high-quality uniform (yet unstructured) distributions which try to approach the spectral characteristics of Poisson disk distributions. The goal is to concentrate the noise in high frequencies where it is less visible. The resulting distributions exhibit better uniformity properties when compared to (0,2)-sequences, but this is achieved at a higher computational cost. Recent works have focused on efficient generation of high-quality blue noise patterns [CYC\*12], [dGBOD12], [EPM\*11], [Fat11], among which the state-of-the-art is currently given in [dGBOD12].

The unit square-based distributions generated by the methods described above must be lifted to the  $\mathbb{S}^2$  sphere using an equal-area projection so as to be used for (hemi)spherical integration. Such projections preserve the property of asymptotic distribution uniformity, but not the samples distance. As discrepancy and w.c.e. directly depend on the distance between samples (see Equation (3)), the resulting sets become suboptimal for (hemi)spherical sampling.

An explicit spherical construction of point sets with small s.c.d. has been proposed in [LPS86], but recently a better order of convergence has been reported by lifting (0,2)-sequences and Fibonacci lattices from the unit square  $[0, 1]^2$  to the  $\mathbb{S}^2$  sphere [ABD12], [BD11]. Both resulting sampling patterns exhibit good discrepancy properties in the spherical domain. Nevertheless their performance is not exactly equivalent as shown by other authors [Nye03], [Gon10] which conclude that Fibonacci lattices are more efficient.

Throughout this paper, we will use a SF lattice implementation based on [SJP06]. We will show that this algorithm is simpler and more efficient than the Sobol (0,2)-sequence [Sob67], the state-of-the-art blue noise [dGBOD12] and the Larcher–Pillichshammer point sets [LP01] when the goal is spherical sampling.

## 3. Spherical Fibonacci Point Sets

Our goal in this section is to explain how Fibonacci lattices are generated and why such point constructions are well suited to spherical sampling. In the following, we introduce SF point sets through a lifting procedure from the unit square to the unit sphere. We have chosen this procedure since it allows establishing a connection with traditional QMC point constructions defined over the unit square.

However the same point sets can be equivalently derived on the sphere or on a disk as shown in [SK97] or [Vog79], respectively.

A Fibonacci lattice in the unit square is a set  $\mathcal{Q}_m$  of  $F_m$  points  $(x, y)$  defined as follows [NH94]:

$$\left. \begin{aligned} x_j &= \left\{ j \frac{F_{m-1}}{F_m} \right\} \\ y_j &= \frac{j}{F_m} \end{aligned} \right\} 0 \leq j < F_m,$$

where  $F_{m-1}$  and  $F_m$  are the two last numbers of a sequence of  $m + 1$  Fibonacci numbers [GKP94] given by the recurrence equation  $F_m = F_{m-1} + F_{m-2}$  for  $m > 1$ .  $F_0 = 0$ ,  $F_1 = 1$  and  $\{x\} = x - \lfloor x \rfloor$  denotes the fractional part for non-negative real numbers  $x$ . By directly lifting this lattice to the unit sphere with the cylindrical Lambert map, we obtain the following point set [Sve94], [HN04]:

$$\begin{aligned} \theta_j &= \arccos(1 - 2j/F_m) \\ \phi_j &= 2\pi \left\{ j \frac{F_{m-1}}{F_m} \right\}, \end{aligned}$$

where  $\theta_j$  and  $\phi_j$  are the polar and azimuthal angles, respectively, of a lattice node  $\omega_j$ . As the Fibonacci ratio  $F_m/F_{m-1}$  quickly approaches the golden ratio  $\Phi = (1 + \sqrt{5})/2$  as  $m$  increases [GKP94], we can write:

$$\lim_{m \rightarrow \infty} \phi_j = 2j\pi\Phi^{-1}$$

due to the periodicity of the spherical coordinates. Hence, setting  $F_m = N$ , the coordinates of an  $N$ -point SF set are given by:

$$\left. \begin{aligned} \theta_j &= \arccos(1 - 2j/N) \\ \phi_j &= 2j\pi\Phi^{-1} \end{aligned} \right\} 0 \leq j < N.$$

Note that in this case,  $N$  need not be a Fibonacci number anymore, which allows generating point sets with an arbitrary number of points. The resulting point sets are no longer lattices when projected back in the original  $(x, y)$  plane since  $\Phi$  is irrational. Therefore, from now on, these point sets will be called SF *point sets*. Letting  $z_j$  denote the  $z$  coordinate of point  $j$ , we have:

$$z_j = \cos \theta_j = 1 - 2j/N,$$

which means that the  $z$  coordinates of the lattice nodes are evenly spaced. Such an arrangement divides the sphere into equal-area spherical ‘rings’ due to the area-preserving property of the Lambert map [Gon10], each ‘ring’ containing a single lattice node. Swinbank *et al.* [SJP06] slightly improved the point set used in [HN04] by introducing an offset of  $1/N$  to the  $z_j$  coordinates (i.e. half the  $z$  coordinate spacing) to achieve a more uniform distribution near the poles. Then we have:

$$\left. \begin{aligned} \theta_j &= \arccos \left( 1 - \frac{2j+1}{N} \right) \\ \phi_j &= 2j\pi\Phi^{-1} \end{aligned} \right\} 0 \leq j < N. \quad (4)$$

As observed in [Gon10], the same point set can be produced using  $\Phi^{-2} = (3 - \sqrt{5})/2$  instead of  $\Phi^{-1}$ . The  $\phi_j$  angles will then be multiples of the golden angle  $\pi(3 - \sqrt{5})$ . More details on the properties of the SF point set can be found in [SJP06], [Gon10]. In particular,

this point set can also be generated by projecting a Fermat spiral on a sphere, also known as the cyclotron spiral. This arrangement can also be found in nature (e.g. the packing of seeds on the sunflowers head [Vog79]), a clear indication of its near-optimality w.r.t. the distance-based energy metric  $E_N$  (Equation (3)). Other theoretical approaches proposed in the literature lead to similar arrangements (e.g. [SK97]).

In the case of illumination integrals (see Equation (5)), the integration domain is not the sphere, but the hemisphere  $\Omega_{2\pi}$ , where the vertical axis  $z$  is aligned with the surface normal. By modifying Equation (4), an  $N$ -point hemispherical SF point set will then be defined as follows:

$$\left. \begin{array}{l} \theta_j = \arccos(z_j) \\ \phi_j = 2j\pi\Phi^{-1} \end{array} \right\} 0 \leq j < N,$$

where the  $z_j = (1 - \frac{2j+1}{2N})$  are the  $z$ -coordinates of the points on the hemisphere. Such a point set can be very easily generated using the pseudo-code presented in Algorithm 1.

---

Algorithm 1 The spherical Fibonacci point set algorithm.

---

```

1:  $\Delta\phi \leftarrow \pi(3 - \sqrt{5})$            (Golden angle (step on  $\phi$ )
2:  $\phi \leftarrow 0$                          (Initialize  $\phi$ )
3:  $\Delta z \leftarrow 1/n$                    (Compute the step on  $z$ )
4:  $z \leftarrow 1 - \Delta z/2$              (Initialize  $z$  with offset
5: for all  $j \leftarrow [1 : n]$  do
6:    $z_j \leftarrow z$ 
7:    $\theta_j \leftarrow \arccos(z_j)$ 
8:    $\phi_j \leftarrow \text{mod}(\phi, 2\pi)$       (Modulo of  $\phi$ )
9:    $z \leftarrow z - \Delta z$                 (Give a step on  $z$ 
10:   $\phi \leftarrow \phi + \Delta\phi$             (Give a step on  $\phi$ 
11: end for
```

---

Image synthesis involves the computation of many illumination integrals. Using the same point set for computing all illumination integrals results in visible patterns in the rendered images. To avoid this problem the sample sets must be scrambled at each illumination integral evaluation. We used a scrambling strategy of the SF sampling pattern which is made directly in the spherical domain by rotating them about the  $z$ -axis with a random angle uniformly distributed over  $[0, 2\pi]$ . This method has proved to be efficient as will be seen in the results, as no low-frequency patterns can be seen. This method has the advantage of preserving the inter-samples distances and thus the energy  $E_N$ . When using the Lambert cylindrical projection, a rotation about the  $z$ -axis on the sphere is equivalent to a Cranley–Patterson [CP76] rotation along the  $x$ -axis in unit square.

#### 4. QMC for Illumination Integrals

To render an image of a scene, the illumination integral must be computed at each point of visible surfaces. This integral gives the reflected radiance  $L_o(\omega_o)$  at a given visible point and can be ex-

pressed as follows:

$$L_o(\omega_o) = \int_{\Omega_{2\pi}} L_i(\omega_i) \rho(\omega_i, \omega_o)(\omega_i \cdot \mathbf{n}) d\Omega(\omega_i), \quad (5)$$

where  $\rho(\omega_i, \omega_o)$  is the BRDF,  $\mathbf{n}$  is the surface normal at the shading point and  $\Omega_{2\pi}$  is the hemisphere of unit radius, the main axis of which is aligned with  $\mathbf{n}$ . The incident direction  $\omega_i$  and the direction of observation  $\omega_o$  are considered as points on the unit hemisphere  $\Omega_{2\pi}$ . A straightforward application of Equation (1) would consist in computing an estimate of  $L_o(\omega_o)$  by averaging samples of the integrand of Equation (5) with uniformly distributed sampling points on  $\Omega_{2\pi}$ . Such an approach would be quite inefficient since the product  $\rho(\omega_i, \omega_o)(\omega_i \cdot \mathbf{n})$  is generally close to zero in a large part of the integration domain. In classic Monte Carlo method, a common solution is to distribute the samples according to a pdf proportional to  $\rho(\omega_i, \omega_o)(\omega_i \cdot \mathbf{n})$ . In the QMC deterministic context, as probabilistic distributions cannot be used, instead this function is moved into the integration variables through an appropriate variable substitution. In the following, we will show how to reformulate the problem of optimally sampling  $\rho(\omega_i, \omega_o)(\omega_i \cdot \mathbf{n})$  in the context of QMC integration, starting from a uniform point set distribution.

Equation (5) can be developed as follows:

$$L_o = \int_0^{2\pi} d\phi \int_0^{\pi/2} \rho(\omega_i, \omega_o) L_i(\theta, \phi) \cos \theta \sin \theta d\theta, \quad (6)$$

where  $\theta$  and  $\phi$  are the spherical coordinates of the incident direction  $\omega_i$  w.r.t. the  $z$ -axis.

In the case of Phong glossy BRDF:

$$\rho(\omega_i, \omega_o) = k \frac{(\max[0, (\omega_i \cdot \omega_r)])^n}{\omega_i \cdot \mathbf{n}},$$

where  $\omega_r = 2(\omega_o \cdot \mathbf{n}) - \omega_o$  is the perfect mirror incident direction. A diffuse BRDF can be seen as a special case for which  $\omega_r = \mathbf{n}$  and  $n = 1$  (its albedo is then  $\pi k$ ).

Considering that the incident radiance function is zero for incident directions below the tangent plane (i.e.  $L_i(\omega_i) = 0$  if  $(\omega_i \cdot \mathbf{n}) < 0$ ), we can take the hemisphere  $\Omega_{2\pi}^{(r)}$  centred about  $\omega_r$  as the integration domain. Our coordinate frame will then be rotated such that its  $z$  is axis aligned with  $\omega_r$  and therefore, the polar angle  $\theta$  of a point  $\omega$  on  $\Omega_{2\pi}^{(r)}$  will be defined by  $\theta = \arccos(z)$  with  $z = (\omega \cdot \omega_r)$ .

Consequently, by making the variable substitution  $z = \cos \theta$ , Equation (6) can be written as follows:

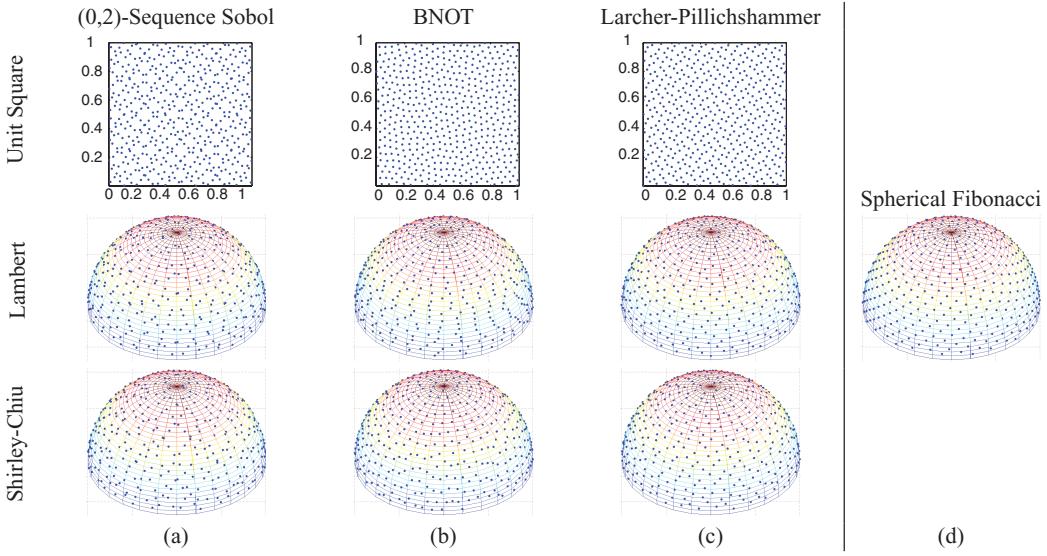
$$L_o(\omega_o) = k \int_0^{2\pi} d\phi \int_0^1 L_i(z, \phi) z^n dz.$$

Making the substitution  $z' = z^{n+1}$ , we have:

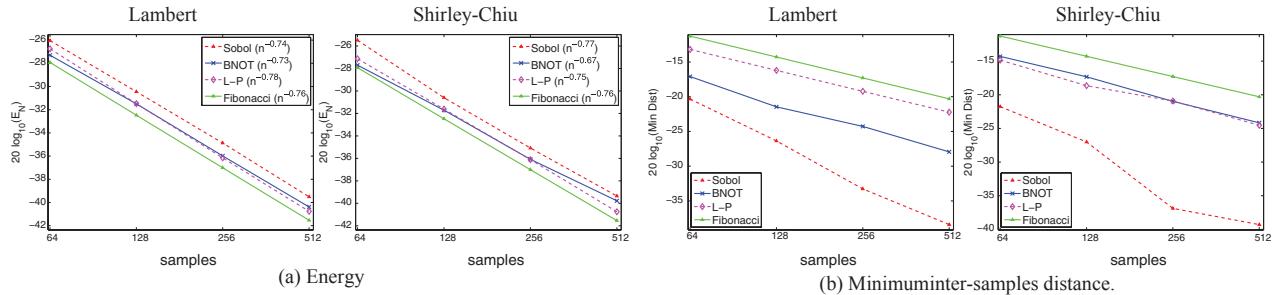
$$L_o(\omega_o) = \frac{k}{n+1} \int_0^{2\pi} d\phi \int_0^1 L'_i(z', \phi) dz',$$

where  $L'_i(z', \phi) = L_i(z'^{1/(n+1)}, \phi)$ . As the integral bounds still define an hemispherical integration domain, an estimate of  $L_o(\omega_o)$  is obtained using Equation (1):

$$\tilde{L}_o(\omega_o) = \frac{2\pi k}{N(n+1)} \sum_{j=1}^N L_i(z_j^{1/(n+1)}, \phi_j), \quad (7)$$



**Figure 1:** Examples of point sets of size 512 produced by different algorithms. Top row: unit square projection. Second row: Lambert cylindrical projection. Third row: Shirley–Chiu concentric maps projection. At the right is the Fibonacci point set generated directly in the spherical domain.



**Figure 2:** Properties of the tested point sets. For each metric (energy and minimum distance), the same point set was projected using the Lambert (left) and Shirley–Chiu (right) projections, except for the Fibonacci point set which is generated directly in the spherical domain.

where  $(z_j, \phi_j)$  are the coordinates of a uniformly distributed samples set  $P_N$  on  $\Omega_{2\pi}^{(r)}$ . Equation (7) means that incident radiance function  $L_i()$  is sampled with a sampling pattern obtained by morphing the  $z$  coordinates of the samples of the uniformly distributed set  $P_N$  with the function  $f(z) = z^{1/(n+1)}$ .

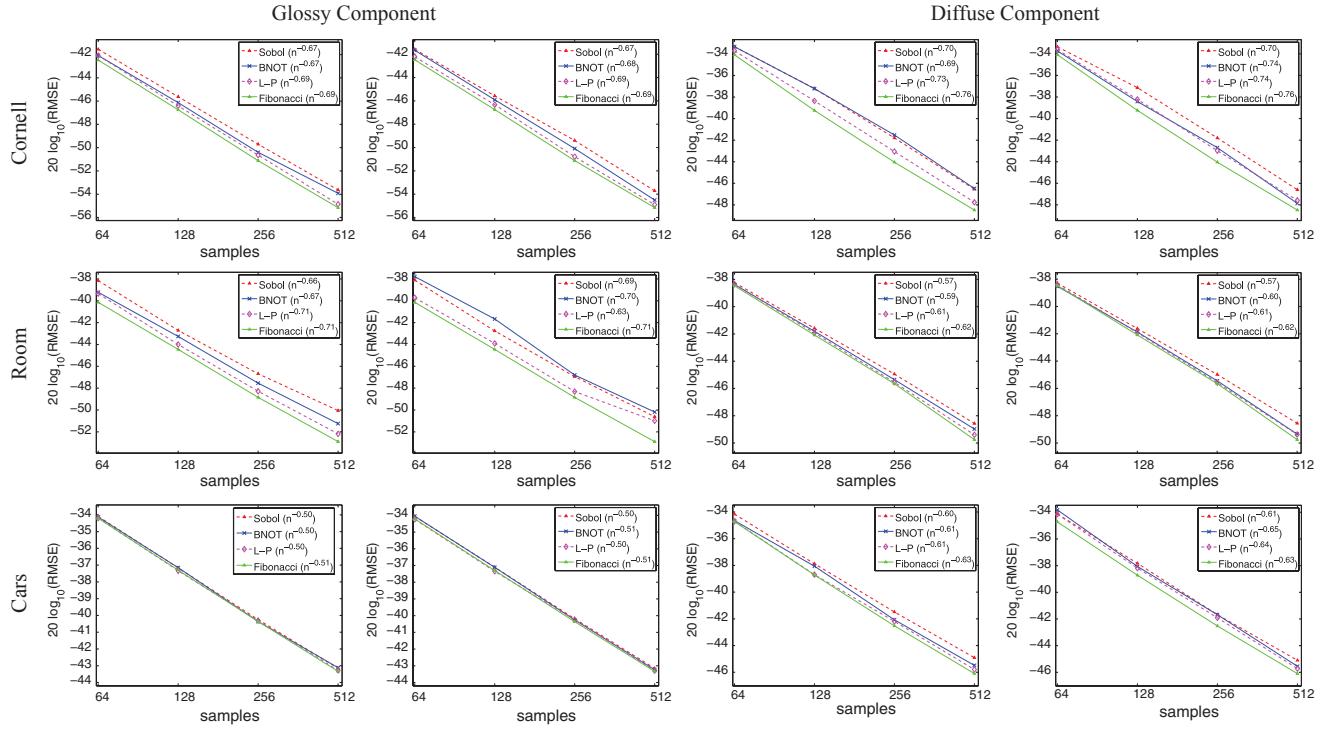
To sum up, the above derivations show how to use a spherical uniform point set to compute an approximation of the illumination integral while taking into account the BRDF shape. Although the original sample set  $P_N$  undergoes a morphing operation, the w.c.e. of the estimate given by Equation (7) is still strongly dependent on the characteristics of  $P_N$  (and in particular on the energy  $E_N$ ), as will be seen in the following sections.

## 5. Tested Point Sets

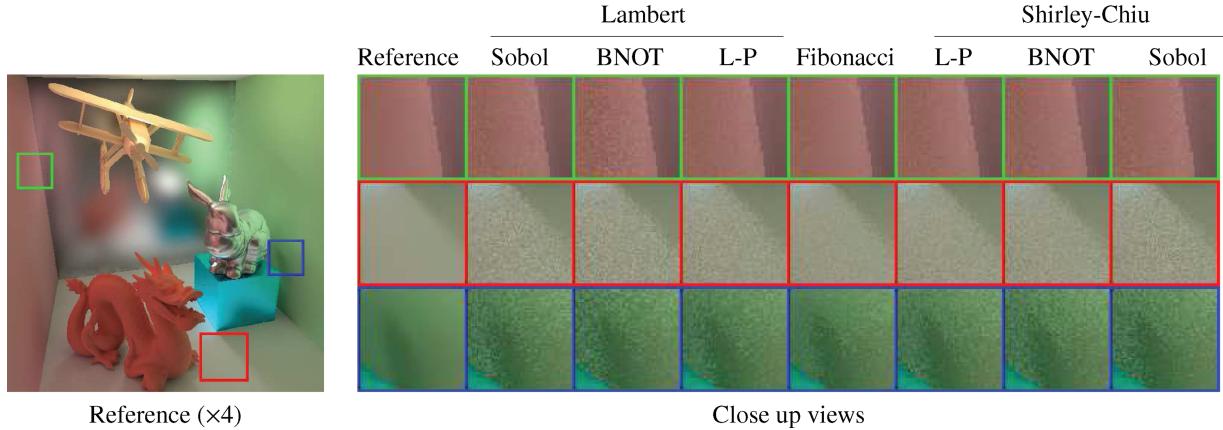
In this section, our goal is to compare the properties of the presented SF point sets with those of the sample sets produced by the following algorithms:

- Sobol (0, 2)-sequence with random digit scrambling as described in [KK02];
- Periodic blue noise, generated with the state-of-the-art algorithm of de Goes *et al.* [dGBOD12];
- Larcher–Pillichshammer points [LP01] with random digit scrambling as described in [KK02].

Henceforth, we will refer to these three algorithms as Sobol, BNOT and L–P, respectively. Figure 1 shows different projections of sets of 512 samples generated using Sobol, BNOT and L–P, as well as an example of an SF point set. We used two different techniques for projecting the unit square point sets to the unit hemisphere: the well-known Lambert cylindrical projection (e.g. see [ABD12]) and the concentric map of Shirley and Chiu [SC97]. Note that these projections do not apply to Fibonacci point sets since they are generated directly in the sphere. The pattern generated by the Sobol sequence is apparently non-optimal in terms of discrepancy, since the distance of a sample to its closest neighbour is quite variable.



**Figure 3:** RMSE plots for the three test scenes as a function of the number of samples. The slopes of the RMSE line fits are displayed in the legend in-between brackets.



**Figure 4:** Cornell Box scene (indirect radiance component only). The rabbit, the blue box and the back wall material contain a glossy BRDF, while the rest of the objects have a perfectly diffuse BRDF. Left: reference image multiplied by a factor of 4. Right: close up views for all the used methods with 128 and 256 sample rays for the glossy and diffuse components, respectively.

This can be observed both on the unit square and on the unit hemisphere projections. On the other hand, the BNOT and L-P sampling patterns (Figures 1b and c, respectively), seem to be more uniformly distributed than the Sobol sequence. As for the SF point set (Figure 1d), it exhibits superior uniformity properties when compared to all the other point sets projected on the hemisphere.

A quantitative analysis of these visual impressions can be made by comparing the different sampling patterns in terms of the en-

ergy metric defined in Equation (3). Let us recall that, as stated in Section 2.1, the w.c.e. is proportional to energy under a  $C_0$  continuity assumption for the integrand. Figure 2(a) clearly illustrates that the Fibonacci point set exhibits a lower energy (Equation (3)) than the other tested algorithms and is thus expected to yield a lower w.c.e. value. In the same line of results, Figure 2(b) shows that the minimum inter-sample distance is consistently larger for the Fibonacci point sets, which is an indication of better uniformity properties. All the tested point sets (except for BNOT using the

**Table 1:** Comparison of the results obtained using a Sobol sequence, blue noise and the Larcher–Pillichshammer points, relative to those obtained using spherical Fibonacci (SF) point sets. The glossy and diffuse components are presented separately, as well as the used projection. For each projection, the first column states the relative RMSE w.r.t. that of SF, using 512 sample rays for all methods. The second column shows the number of rays required to achieve the same RMSE as SF with 512 rays. In-between brackets is the corresponding percentage.

Scene	Point set	Glossy component				Diffuse component			
		Lambert		Shirley-Chiu		Lambert		Shirley-Chiu	
		RMSE	Same quality rays needed	RMSE	Same quality rays needed	RMSE	Same quality rays needed	RMSE	Same quality rays needed
Cornell Box	Sobol	+19.2%	658(+28.5%)	+18.1%	667(+30.2%)	+24.9%	762(+48.8%)	+24.2%	760(+48.5%)
	BNOT	+15.2%	610(+19.2%)	+7.8%	601(+17.4%)	+25.9%	785(+53.4%)	+7.4%	605(+18.1%)
	L-P	+3.4%	544(+6.3%)	+3.0%	541(+5.6%)	+8.7%	603(+17.8%)	+10.9%	614(+19.9%)
	Sobol	+38.7%	799(+56.0%)	+30.0%	718(+40.3%)	+14.6%	661(+29.1%)	+14.8%	662(+29.2%)
	BNOT	+20.9%	665(+29.9%)	+36.6%	763(+49.1%)	+9.4%	601(+17.5%)	+4.8%	569(+11.2%)
	L-P	+8.5%	557(+8.8%)	+24.6%	665(+29.8%)	+4.1%	555(+8.5%)	+4.5%	558(+9.0%)
Room	Sobol	+2.2%	528(+3.1%)	+2.1%	531(+3.7%)	+14.7%	634(+23.9%)	+12.2%	607(+18.5%)
	BNOT	+2.5%	526(+2.7%)	+1.2%	520(+1.6%)	+7.3%	569(+11.0%)	+6.4%	559(+9.3%)
	L-P	+0.6%	514(+0.4%)	+1.1%	518(+0.5%)	+3.5%	535(+4.5%)	+4.4%	544(+6.3%)
Cars									

Shirley–Chiu projection) yield approximately the same  $\mathcal{O}(N^{-3/4})$  rate of decay for  $E_N$ , which corresponds to the optimum rate of convergence for the w.c.e., as explained in Section 2.1. Recall that this convergence rate is obtained under a  $C_0$  continuity assumption of the integrand, which is in general not fulfilled for illumination integrals. Nevertheless, as will be seen in the next section, these inconsistencies have marginal effects. In particular, we will show experimentally that the accuracy of the estimates given by Equation (7) strongly depends on the energy  $E_N$  of the uniformly distributed samples set  $P_N$ .

## 6. Results

### 6.1. General considerations

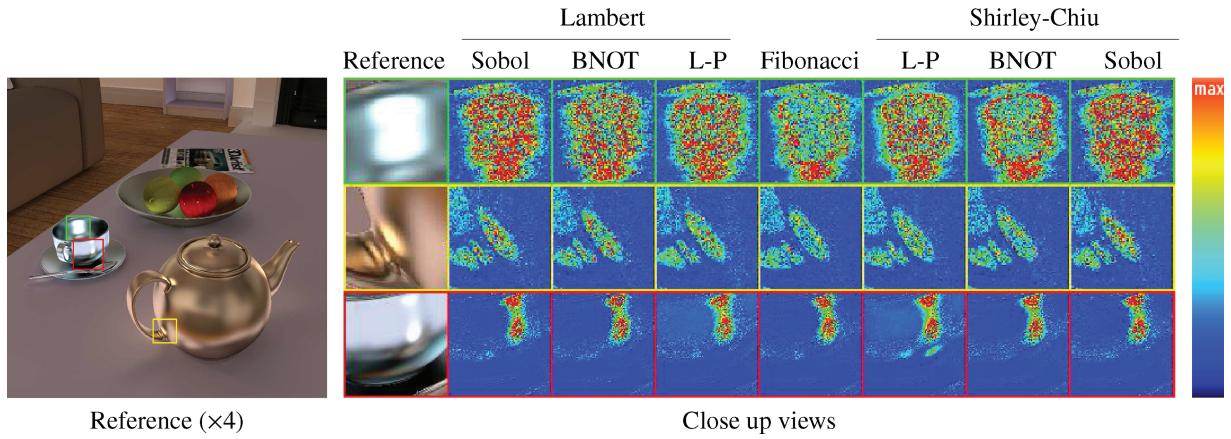
The results presented in this section have been generated with the Mitsuba raytracer [Jak10] on a 64-bit machine equipped with a 2 GHz Intel Core i7 processor and a 8-Gb RAM. Three different scenes have been used: Cornell Box (185 K triangles), Room (540 K triangles) and Cars (1500 K triangles). The illumination integral computation has been performed in the context of final gathering for photon mapping, using the estimators given in Section 4. We have compared the results produced using the different point set construction strategies which have been presented in Section 5. A reference image has been computed using a sampling pattern produced by a Sobol sequence and a large number of samples until convergence was achieved. This reference image was then used to evaluate the RMSE of the images produced with the different point sets. For SF and BNOT, scrambling is performed on the sphere (as described in Section 3), but for L-P and Sobol sampling patterns, it is made on the plane according to the random digit scrambling method proposed in [KK02]. We generate two distinct fixed-size sample sets for the diffuse and glossy components of the BRDF (see Section 4 for details on samples set generation).

### 6.2. RMSE analysis and convergence slope

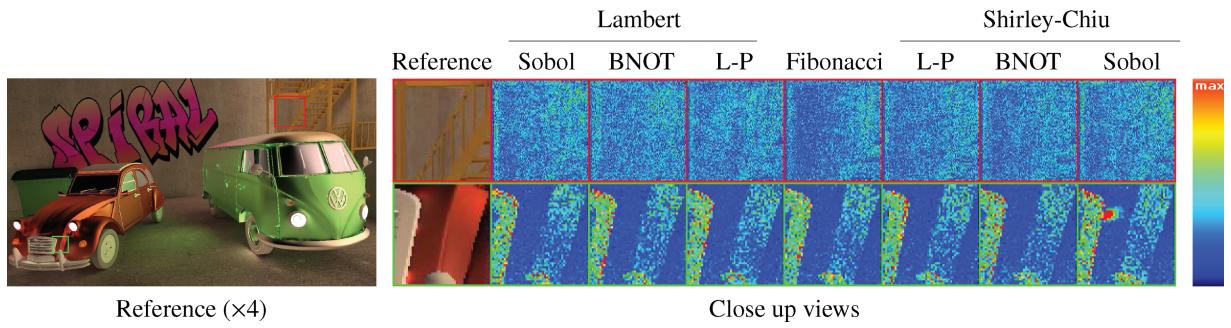
Figure 3 shows that for the same number of samples, the SF point sets yield consistently smaller RMSE values than the other tested methods. Indeed, we have not registered any case where the Fibonacci lattices have been outperformed in terms of RMSE value.

The convergence slope of QMC methods depends on the smoothness properties of the integrands. Therefore, it is not guaranteed that the theoretical convergence rate for the w.c.e. ( $\mathcal{O}(N^{-3/4})$  for  $C_0$  continuous functions) can be obtained for highly discontinuous integrands, such as those commonly met in illumination integrals. Nevertheless, in the Cornell Box scene and in the glossy component of the Room scene, it was possible to report convergence rates close to the theoretical  $\mathcal{O}(N^{-3/4})$ , which means that the integrand for that scene fulfills the  $C_0$  smoothness condition most of the times.

A comparison between the convergence rates in Figure 3 shows that the convergence slope of the SF point set is in general as good or better than those of the other tested point sets. Note that when the convergence slope is steeper for all methods (e.g. diffuse component of the Cornell Box scene) SF point sets clearly outperform the other tested point sets. This can be explained by the fact that SF point sets are more able to take advantage of smooth integrands. According



**Figure 5:** Room scene (indirect radiance component only). The teapot, the teacup and the fruit-dish materials contain a glossy BRDF, while the rest of the objects have a perfectly diffuse BRDF. Left: reference image multiplied by a factor of 4. Right: close-up views comparison of the error images for all the used methods using 32 and 128 sample rays for the glossy and diffuse components, respectively. The colour encodes the error magnitude.



**Figure 6:** Cars scene. The materials associated with the glasses and the bodyworks of both cars contain a glossy BRDF, while the rest of the objects have a perfectly diffuse BRDF. Left: reference image multiplied by a factor of 4. Right: close-up views comparison of the error images for all the used methods with 512 sample rays. The colour encodes the error magnitude.

to [BD11], a convergence rate as high as  $\mathcal{O}(N^{-2})$  is possible with SF point sets in the case of very smooth integrands. On the other hand, when the rate of decay is close to  $\mathcal{O}(N^{-1/2})$  (e.g. the glossy component of the Cars scene in Figure 3), all the point sets yield similar performances since QMC in general is inefficient for very discontinuous integrands.

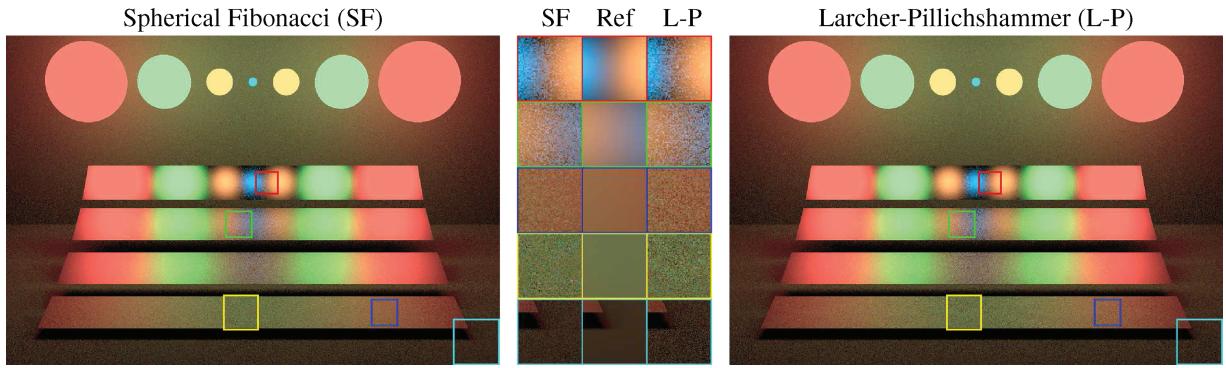
### 6.3. Efficiency and image quality

The benefit of using SF point sets is thoroughly assessed in Table 1. The results show that for 512 samples per shading point, the RMSE of L-P, BNOT and Sobol point sets w.r.t. to that of SF can be up to +8.7%, +36.6% and +38.7%, respectively. Note that this results in an even higher percentage of saved rays. As an example, for the same cases pointed out, L-P needs +17.8% sample rays, BNOT +49.1% and Sobol +56% to achieve the same RMSE as SF with 512 sample rays.

The number of rays needed to close the gap between the RMSE of SF with 512 rays and that of the other methods depends on the

rate of convergence for the given configuration: scene, sampling method, spherical projection and radiance component. This can be clearly seen in Table 1 for the Room scene using the L-P points and a Lambert projection. In this case, the relative RMSE of the glossy component (+8.5%) is more than twice that of the diffuse component (+4.1%), using 512 samples. However, both components require approximately the same number of samples (around 556) to achieve the same RMSE than SF with 512 samples. The reason for this is that the glossy component converges faster than the diffuse component (see legend Figure 3).

The improvement brought by the use of SF point sets can be appreciated on the close-up views of Figure 4 which show that SF yields less visual noise compared to the other methods. As for the Room scene in Figure 5, the error images indicate that SF performs better in critical areas such as the specular highlights. In the Cars scene (Figure 6) on the other hand, the high discontinuity of the incident radiance makes the performance of all methods be roughly similar (as seen in Table 1). Nevertheless, it is still possible to identify image regions where the incident radiance is smoother,



**Figure 7:** This figure shows how spherical Fibonacci (SF) point sets behave for an incident radiance function covering a wide range of frequencies and materials of different glossinesses. Direct lighting of different plates with light sources of varying size using SF (left) and Larcher–Pillichshammer (L–P, right) point sets. The L–P point sets have been projected using the Lambert cylindrical projection. The Phong shininess coefficient  $n$  of each of the plates is 10, 50, 80 and 200 from bottom to top, respectively, while the background is perfectly diffuse. The RMSE of the image rendered using the L–P points is 7.55% higher. The maximum quadratic error per pixel is 0.39 for SF and 0.71 for L–P.

which favours SF point sets as shown on the top row of the close up views of Figure 6.

Figure 7 shows images computed with the SF point sets and L–P with a Lambert projection. We have compared SF with L–P since they both provide the smallest RMSE. The scene is made up of four plates, each one having a different shininess coefficient. It contains seven light sources of variable size and variable radiance producing a direct incident radiance along the plates of variable frequency. With this scene, our objective is to show how SF behaves compared to L–P point sets when the incident light contains structured circular patterns and/or high frequencies that could interfere with the regular sampling pattern of SF. Despite the regularity of the SF point sets no regular patterns can be seen, thanks to the used spherical scrambling method.

## 7. Conclusions

In this paper, we have presented an algorithm for efficient generation of high-quality spherical QMC sequences for approximating illumination integrals. The advantages of our approach can be summarized as follows:

*Simplicity:* The SF point sets algorithm is simpler to implement than the other tested QMC sample sets.

*Compactness:* A single sequence is needed to synthesize an image. This is achieved by exploiting the axial symmetry of the BRDF lobes, which allows scrambling the point sets directly on the spherical domain using just a random axial rotation. This feature might make SF point sets particularly well suited to graphics processing unit implementations.

*Efficiency:* SF point sets outperform L–P, Sobol and blue noise-based QMC in all the test cases, allowing to save a very significant amount of sampling rays for the same image quality.

The main reason for the improvement brought by SF point sets is that they better suit the spherical geometry. The other methods, in contrast, by focusing on the unit square distribution, introduce boundaries that do not exist on the sphere. Instead, our approach tries to obtain the best samples distribution directly on the sphere and then mask the effect of its regularity on the rendered image by an appropriate scrambling method.

## 8. Future Work

An obvious research line is to develop adaptive sampling schemes while keeping the high quality of the energy criterion exhibited by the SF point sets. As for increasing the quality of QMC BRDF-based sampling, we consider that we are already quite close to optimality and little margin for improvement exists. To go further, one could resort to non-frequentist approaches, that is, *Bayesian Monte Carlo* [BBL\*09], which allow adapting the sampling patterns according to a global covariance function of the incident radiance samples. Another research line is the reduction of the perceived error by introducing some correlation between the random rotation angles assigned to sample sets used in illumination integrals of neighbour pixels.

## Acknowledgements

This work is partially funded by National Funds through the FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project PEst-OE/EEI/UI0752/2011.

## References

- [ABD12] AISTLEITNER C., BRAUCHART J., DICK J.: Point sets on the sphere  $\mathbb{S}^2$  with small spherical cap discrepancy. *Discrete & Computational Geometry* 48, 4 (2012), 990–1024.

- [BBL\*09] BROUILLAT J., BOUVILLE C., LOOS B., HANSEN C. D., BOUATTUCH K.: A Bayesian Monte Carlo approach to global illumination. *Computer Graphics Forum* 28, 8 (2009), 2315–2329.
- [BD11] BRAUCHART J. S., DICK J.: Quasi-monte carlo rules for numerical integration over the unit sphere  $S^2$ . *ArXiv e-prints* (Jan. 2011). <http://arxiv.org/abs/1101.5450>.
- [Bec84] BECK J.: Sums of distances between points on a sphere — an application of the theory of irregularities of distribution to discrete geometry. *Mathematika* 31, 1 (June 1984), 33–41.
- [BSSW12] BRAUCHART J. S., SAFF E. B., SLOAN I. H., WOMERSLEY R. S.: QMC designs: Optimal order Quasi Monte Carlo Integration schemes on the sphere. *ArXiv e-prints* (Aug. 2012). <http://arxiv.org/abs/1208.3267>.
- [CP76] CRANLEY R., PATTERSON T.: Randomization of number theoretic methods for multiple integration. *SIAM Journal on Numerical Analysis* 13, 6 (1976), 904–914.
- [CYC\*12] CHEN Z., YUAN Z., CHOI Y.-K., LIU L., WANG W.: Variational blue noise sampling. *IEEE Transactions on Visualization and Computer Graphics* 18, 10 (2012), 1784–1796.
- [dGBOD12] DE GOES F., BREEDEN K., OSTROMOUKHOV V., DESBRUN M.: Blue noise through optimal transport. *ACM Transactions on Graphics (SIGGRAPH Asia)* 31 (2012) 171:1–171:11.
- [EPM\*11] EBEIDA M. S., PATNEY A., MITCHELL S. A., DAVIDSON A., KNUPP P. M., OWENS J. D.: Efficient maximal Poisson-disk sampling. *ACM Transactions on Graphics* 30, 4 (2011) 49:1–49:12.
- [Fat11] FATTAL R.: Blue-noise point sampling using kernel density model. *ACM SIGGRAPH 2011 Papers* 28, 3 (2011), 1–10.
- [GKP94] GRAHAM R. L., KNUTH D. E., PATASHNIK O.: *Concrete Mathematics: A Foundation for Computer Science* (2nd edition). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1994.
- [Gon10] GONZÁLEZ I.: Measurement of areas on a sphere using Fibonacci and Latitude–Longitude lattices. *Mathematical Geosciences* 42 (2010), 49–64.
- [HN04] HANNAY J. H., NYE J. F.: Fibonacci numerical integration on a sphere. *Journal of Physics A: Mathematical and General* 37, 48 (2004), 11591–11601.
- [Jak10] JAKOB W.: Mitsuba renderer, 2010. <http://www.mitsuba-renderer.org>. Accessed September 2012.
- [Kel12] KELLER A.: Quasi-Monte Carlo image synthesis in a nutshell, 2012. <https://sites.google.com/site/qmcrendering/>. Accessed September 2012.
- [KK02] KOLLIG T., KELLER A.: Efficient multidimensional sampling. *Computer Graphics Forum* 21, 3 (2002), 557–563.
- [KN06] KUIPERS L., NIEDERREITER H.: *Uniform Distribution of Sequences*. Dover Books on Mathematics. Dover Publications, New York, 2006.
- [KPR12] KELLER A., PREMOZE S., RAAB M.: Advanced (quasi) monte carlo methods for image synthesis. In *ACM SIGGRAPH 2012 Courses* (New York, NY, USA, 2012), SIGGRAPH ’12, ACM, pp. 21:1–21:46.
- [LD08] LAGAE A., DUTRÉ P.: A comparison of methods for generating Poisson disk distributions. *Computer Graphics Forum* 27, 1 (March 2008), 114–129.
- [LP01] LARCHER G., PILICHSHAMMER F.: Walsh series analysis of the  $L^2$ -discrepancy of symmetrized point sets. *Monatshefte für Mathematik* 132, 1 (2001), 1–18.
- [LPS86] LUBOTZKY A., PHILLIPS R., SARNAK P.: Hecke operators and distributing points on the sphere I. *Communications on Pure and Applied Mathematics* 39, S1 (1986), S149–S186.
- [NH94] NIEDERREITER H., H. S. I.: Integration of nonperiodic functions of two variables by Fibonacci lattice rules. *Journal of Computational and Applied Mathematics* 51, 1 (1994), 57–70.
- [Nie88] NIEDERREITER H.: Low-discrepancy and low-dispersion sequences. *Journal of Number Theory* 30, 1 (1988), 51–70.
- [Nye03] NYE J.: A simple method of spherical near-field scanning to measure the far fields of antennas or passive scatterers. *IEEE Transactions on Antennas and Propagation* 51, 8 (August 2003), 2091–2098.
- [SC97] SHIRLEY P., CHIU K.: A low distortion map between disk and square. *Journal of Graphics Tools* 2, 3 (December 1997), 45–52.
- [SEB08] SHIRLEY P., EDWARDS D., BOULOS S.: Monte Carlo and quasi-Monte Carlo methods for computer graphics. In *Monte Carlo and Quasi-Monte Carlo Methods 2006*. A. KELLER, S. HEINRICH and H. NIEDERREITER (Eds.). Springer, Berlin Heidelberg, 2008, pp. 167–177.
- [SJP06] SWINBANK R., JAMES PURSER R.: Fibonacci grids: A novel approach to global modelling. *Quarterly Journal of the Royal Meteorological Society* 132, 619 (2006), 1769–1793.
- [SK97] SAFF E., KUIJLAARS A.: Distributing many points on a sphere. *The Mathematical Intelligencer* 19 (1997), 5–11.
- [Sob67] SOBOL I.: On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics* 7 (1967), 86–112.
- [Sve94] SVERGUN D. I.: Solution scattering from biopolymers: Advanced contrast-variation data analysis. *Acta Crystallographica Section A* 50, 3 (May 1994), 391–402.
- [Vog79] VOGEL H.: A better way to construct the sunflower head. *Mathematical Biosciences* 44, 3–4 (1979), 179–189.

# A Spherical Gaussian Framework for Bayesian Monte Carlo Rendering of Glossy Surfaces

Ricardo Marques, Christian Bouville, Mickaël Ribardière, Luís Paulo Santos, and Kadi Bouatouch

**Abstract**—The Monte Carlo method has proved to be very powerful to cope with global illumination problems but it remains costly in terms of sampling operations. In various applications, previous work has shown that Bayesian Monte Carlo can significantly outperform importance sampling Monte Carlo thanks to a more effective use of the prior knowledge and of the information brought by the samples set. These good results have been confirmed in the context of global illumination but strictly limited to the perfect diffuse case. Our main goal in this paper is to propose a more general Bayesian Monte Carlo solution that allows dealing with nondiffuse BRDFs thanks to a spherical Gaussian-based framework. We also propose a fast hyperparameters determination method that avoids learning the hyperparameters for each BRDF. These contributions represent two major steps toward generalizing Bayesian Monte Carlo for global illumination rendering. We show that we achieve substantial quality improvements over importance sampling at comparable computational cost.

**Index Terms**—Bayesian Monte Carlo, Gaussian process, spherical Gaussian

## 1 INTRODUCTION

In global illumination rendering, the main issue lies in the computation of multidimensional integrals involving intensive ray-traced sampling. Although the Monte Carlo method has proved to be very powerful when coping with this problem, it remains costly in terms of sampling operations. As the computational cost of sampling is very high compared to the sheer quadrature cost, we can ask ourselves if Monte Carlo methods make an efficient use of information brought by the samples. This question has been raised by O'Hagan which led him to propose the "Bayes-Hermite quadrature" [1], a new form of quadrature which is referred to as "Bayesian Monte Carlo" (BMC) by other authors. While keeping the fundamental property of data dimension independence of Monte Carlo methods, it considerably broadens the set of theoretical tools that can be used to exploit the information produced by sampling. In particular, BMC uses the information regarding the samples location, which is ignored in the classic Monte Carlo method (CMC). Moreover, BMC offers much more flexibility in the exploitation of the prior knowledge compared to CMC which mainly relies on sampling strategies. However, all these advantages are obtained at the expense of the quadrature complexity and additional preprocessing. Brouillat et al. [2] have proposed efficient

solutions that make the overhead of computing the BMC quadrature negligible compared to CMC. Moreover, their results show that BMC can significantly outperform CMC methods, even when including the preprocessing step. Nevertheless, their work only considers diffuse reflections and their strategy for efficiently computing the quadrature coefficients does not apply to nondiffuse BRDFs. A direct application of the method to view dependent BRDFs would require massive precomputations of multidimensional tables so as to allow an acceptable rendering time. The prior model construction strategy presented in [2] would also be inappropriate to the sharpness features of glossy BRDFs. It would require performing a learning phase for each BRDF present in the scene, which would represent a high computational cost. Furthermore, the proposed samples set optimization method is not suited to the highly nonuniform samples distributions needed to efficiently compute glossy reflections. Applied to glossy BRDFs, their method converges towards local minima which are very far from the optimal solution.

In this paper, we propose a new theoretical framework that includes a novel method of quadrature computation based on spherical Gaussian functions that can be generalized to a broad class of BRDFs (any BRDF which can be approximated by a sum of one or more spherical Gaussian functions) and potentially to other rendering applications. We account for the BRDF sharpness by using a new computation method for the prior mean function and by introducing a new factor in the method of Brouillat et al. [2] for constructing optimized samples set. Finally, we propose a fast hyperparameters evaluation method that avoids the learning step.

In the following, after a presentation of related work, we introduce the theoretical aspects of BMC. Then we present the application of BMC to the illumination integral and develop our theoretical framework. It is followed by a description of the full rendering algorithm and a presentation of comparative results. Then, we suggest various research directions in a discussion section. We conclude by summarizing what was learned from our implementation and mention new perspectives of application of BMC.

- R. Marques is with the Institut National de Recherche en Informatique et Automatique, Campus Universitaire de Beaulieu, 35042 Rennes Cedex, France. E-mail: ricardo.marques@inria.fr.
- C. Bouville, M. Ribardière, and K. Bouatouch are with the Institut de Recherche en Informatique et Systèmes Aleatoires, Campus Universitaire de Beaulieu, 3042 Rennes Cedex, France.  
E-mail: (mickael.ribardiere, christian.bouville, kadi.bouatouch)@irisar.fr.
- L.P. Santos is with the Department of Informatica, University of Minho, Campus de Gualtar, Braga 4710-057, Portugal.  
E-mail: psantos@di.uminho.pt.

Manuscript received 6 Sept. 2012; revised 25 Jan. 2013; accepted 27 Apr. 2013; published online 7 May 2013.

Recommended for acceptance by J. Kautz.

For information on obtaining reprints of this article, please send e-mail to: [tvcg@computer.org](mailto:tvcg@computer.org), and reference IEEECS Log Number TVCG-2012-09-0188. Digital Object Identifier no. 10.1109/TVCG.2013.79.

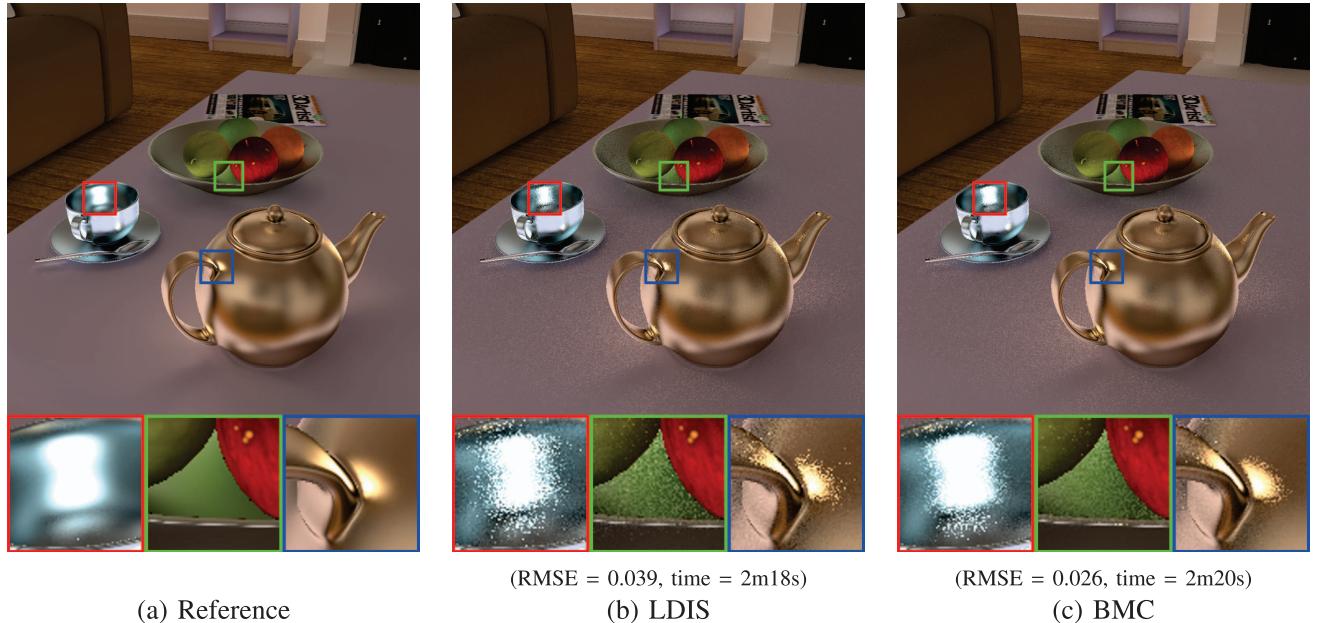


Fig. 1. Indirect radiance component for the room scene rendered with low discrepancy Monte Carlo importance sampling (LDIS, (b)) and BMC (c). The shown images have been multiplied by a factor of 4. The RMSE and the time are computed by considering only the glossy component. Sixteen ray samples per visible point were used for the materials with a glossy BRDF (teapot, tea cup and fruit-dish), while 64 samples were used for the diffuse BRDFs.

## 2 RELATED WORK

The fundamental difference between BMC and CMC lies in the use of a prior stochastic model of the function  $f(\mathbf{x})$  to be integrated. Note that in the Bayesian framework the argument  $\mathbf{x}$  is considered deterministic, as opposed to CMC methods. Therefore, the randomness is not introduced by random sampling as in CMC but by the uncertainty (in a Bayesian sense) we put on the function to be integrated before any samples are drawn. Several approaches have been proposed in the literature regarding how to model this uncertainty. Our approach is based on the Bayes-Hermite quadrature of O'Hagan [1] which uses a Gaussian process (GP) for the prior model. We have found this method more appropriate for the computation of global illumination integrals because it easily leads to closed-form solutions. Rasmussen and Ghahramani [3] have shown that GP-based BMC can significantly outperform Monte Carlo importance sampling (MCIS). Several applications of the Bayes-Hermite quadrature have then been proposed in the literature and among them, the work of Pfingsten et al. [4] is particularly interesting in the way they derive a closed-form solution. However, it is not suited to the particularities of spherical functions we are faced with in global illumination problems. Broullat et al. [2] have proposed solutions in the context of final gathering for photon mapping and have clearly shown the benefit of BMC over MCIS. However, this method can only deal with diffuse reflections. The main goal of this paper is thus to propose a BMC framework which can efficiently deal with the problems raised by glossy reflections.

## 3 BACKGROUND

In this section, we give a brief introduction to the theoretical basis of BMC, and make a short description of the previous application of BMC to the diffuse BRDF case. Detailed

information can be found in [1], [2], and [3]. The notations used throughout the paper are described in Table 1.

### 3.1 The Gaussian Process Prior

As in any Bayesian method, we need to state our knowledge prior to performing observations. This is usually defined as a probability distribution often simply called the prior. In BMC, the prior is modeled by a Gaussian process as explained in the following.

Let us consider the computation of the integral:

$$I = \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x}, \text{ with } \mathbf{x} \in \mathbb{R}^D, \quad (1)$$

where  $p(\mathbf{x})$  is analytically known and  $f(\mathbf{x})$  can only be determined through physical observations or, as in our case, through numerical evaluations. Given the high cost of these evaluations, it is infeasible to have a detailed knowledge about  $f(\mathbf{x})$  in all its domain. Our knowledge about  $f(\mathbf{x})$  is restricted to a limited set of samples while anywhere else in the domain, we are not sure about the value of  $f(\mathbf{x})$ . This uncertainty about  $f(\mathbf{x})$  leads us to interpret it as a random quantity. The Bayesian reasoning states that all forms of uncertainty can be modeled by probability. Consequently,  $f(\mathbf{x})$  can be considered as random because its value is unknown and thus uncertain. To model our uncertainty about  $f(\mathbf{x})$ , we will use a stochastic model called Gaussian process. For a more practical knowledge about GP, the reader may refer to [5]. Formally, a GP is a collection of random variables, any finite number of which has a joint Gaussian distribution. A GP is completely defined by its mean function  $\bar{f}(\mathbf{x})$  and its covariance function  $k(\mathbf{x}, \mathbf{x}')$ , which must be positive definite [6]:

$$\bar{f}(\mathbf{x}) = E[f(\mathbf{x})], \quad (2)$$

$$k(\mathbf{x}, \mathbf{x}') = E[(f(\mathbf{x}) - \bar{f}(\mathbf{x}))(f(\mathbf{x}') - \bar{f}(\mathbf{x}'))], \quad (3)$$

TABLE 1  
Notations Description

$\mathbf{x}$	Point in $\mathbb{R}^D$
$f(\mathbf{x})$	Analytically known function
$\bar{f}(\mathbf{x})$	Unknown function interpreted as a random quantity and modeled through a Gaussian Process (GP)
$\tilde{\mathbf{F}}$	Mean function $E[\bar{f}(\mathbf{x})]$
$k(\mathbf{x}, \mathbf{x}')$	Vector of mean function values
$k(\mathbf{x} - \mathbf{x}')$	General non-stationary covariance function
$k'(\mathbf{x} - \mathbf{x})$	Stationary covariance function
$\mathbf{k}(\mathbf{x})$	Stationary correlation function
$K$	Vector of covariance values between a position $\mathbf{x}$ and a set of sample locations $\mathbf{x}_1, \dots, \mathbf{x}_n$
$Q$	Noise-free covariance matrix
$Y_i$	Covariance matrix
$\mathbf{Y}$	Noisy observation of $f$ at location $\mathbf{x}_i$ ( $f(\mathbf{x}_i) + \epsilon_i$ )
$\epsilon_i$	Vector of observations $(Y_1, \dots, Y_n)$
$\sigma_n^2$	Sample of an i.i.d. Gaussian noise ( $E[\epsilon_i] = 0$ )
$\sigma_f^2$	Variance of $\epsilon_i$
$\sigma_y^2$	Variance of the GP
$L_i(\omega)$	Variance of the observations ( $\sigma_n^2 + \sigma_f^2$ )
$l$	Incident radiance from direction $\omega$
$G()$	Lengthscale of the GP which models $L_i$
$\rho()$	Spherical Gaussian function
$m$	BRDF
$w$	Shininess coefficient of the BRDF
	Gaussian BRDF lobe width $w = 1/\sqrt{m}$

and will be denoted as

$$f(\mathbf{x}) \sim \mathcal{GP}[\bar{f}(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')].$$

We use the GP formalized by (2) and (3) as our prior model.  $\bar{f}(\mathbf{x})$  is our expectation of  $f(\mathbf{x})$  before any observation is made. The covariance function characterizes the belief we have on the smoothness of  $f(\mathbf{x})$ , i.e., how correlated are nearby samples. A strong correlation between the samples implies that  $f(\mathbf{x})$  is very smooth. The covariance function of the prior GP is often assumed stationary. A covariance function  $k$  is stationary when  $k(\mathbf{x}, \mathbf{x}') := k(\mathbf{x} - \mathbf{x}')$ ,  $\forall (\mathbf{x}, \mathbf{x}')$ , in which case the variance of  $f(\mathbf{x})$ , i.e.,  $k(\mathbf{x}, \mathbf{x}) = k(\mathbf{0})$ , is constant. The prior covariance function  $k$  is parametrized by a set of *hyperparameters* which need to be determined in a preprocessing step. This problem will be addressed in Sections 4.2.1 and 5.2. For now, we assume that the hyperparameters are known.

### 3.2 The Posterior Gaussian Process

Once the prior is defined, we can collect observations of  $f(\mathbf{x})$  so as to refine our model by leveraging the prior GP. The resulting process is also a GP and is called posterior GP. In this section, we describe how to compute the posterior GP given the prior and a set of observations.

Let us suppose that we are provided with a set  $\mathcal{D}$  of noisy samples of  $f(\mathbf{x})$ :

$$\mathcal{D} = \{(\mathbf{x}_i, Y_i) \mid i = 1, \dots, n\} \quad \text{with} \quad Y_i = f(\mathbf{x}_i) + \epsilon_i,$$

the  $\epsilon_i$  being samples of an independent, identically distributed Gaussian noise with zero mean and variance  $\sigma_n^2$ ,  $\mathbf{x}_i$  being a sample location (the input) and  $Y_i$  its corresponding sample value (the output). Given this additive noise assumption, the covariance of the observations becomes:

$$\text{cov}(Y_p, Y_q) = k(\mathbf{x}_p, \mathbf{x}_q) + \sigma_n^2 \delta_{pq}, \quad (4)$$

where  $\delta_{pq}$  is the Kronecker symbol.  $\sigma_n^2$  is thus a hyperparameter of the prior model and represents the variance which is unexplained by the GP. It is important to understand that this noise component is not only due to measurement errors and computation inaccuracies. In a very broad sense, the noise term allows to accommodate highly discontinuous data that do not fit the smoothness assumption implied by the prior GP. Indeed, illumination functions are generally discontinuous and cannot be modeled accurately with smooth basis functions. The noise term provides the necessary flexibility so that when building the posterior process, the most plausible smooth model can be fitted to discontinuous data. This data fitting has the same effect as the prefiltering method proposed by Křivánek and Colbert [7] which aims at low-pass filtering the observed samples to reduce aliasing. However, GP model fitting does not assume any band-limited functions hypothesis. Moreover, smoothness has to be understood in terms of order of continuity of the integrand, not in terms of spectral bandwidth. In conclusion, BMC is applicable whatever the smoothness or the bandwidth of the integrand but of course, it performs better when the noise level  $\sigma_n^2$  is low, that is when the prior fits well the data.

The posterior process results from applying the Bayes' rule to incorporate the information brought by the samples. It can be shown that the obtained posterior process is also a GP with mean and covariance functions given by [6]:

$$\begin{aligned} E[f(\mathbf{x})|\mathcal{D}] &= \bar{f}(\mathbf{x}) + \mathbf{k}(\mathbf{x})^t Q^{-1}(\mathbf{Y} - \bar{\mathbf{F}}), \\ \text{cov}[f(\mathbf{x}), f(\mathbf{x}')|\mathcal{D}] &= k(\mathbf{x}, \mathbf{x}') - \mathbf{k}(\mathbf{x})^t Q^{-1} \mathbf{k}(\mathbf{x}'), \end{aligned} \quad (5)$$

with

$$\begin{aligned} \mathbf{k}(\mathbf{x}) &= (k(\mathbf{x}_1, \mathbf{x}), \dots, k(\mathbf{x}_n, \mathbf{x}))^t, \\ K_{i,j} &= k(\mathbf{x}_i, \mathbf{x}_j) \quad \text{with } (i, j) \in [1, n]^2, \\ Q &= (K + \sigma_n^2 I_n), \\ \mathbf{Y} &= (Y_1, \dots, Y_n)^t, \\ \bar{\mathbf{F}} &= (\bar{f}(\mathbf{x}_1), \dots, \bar{f}(\mathbf{x}_n))^t, \end{aligned} \quad (6)$$

and  $I_n$  being the  $n \times n$  identity matrix.  $\mathbf{Y}$  is the vector of observed samples value while  $\bar{\mathbf{F}}$  is the vector of prior GP mean values at the sampling locations.  $Q$  contains the covariance between the samples and is called covariance matrix. Equation (5) gives an estimate of  $f(\mathbf{x})$  for an unobserved input  $\mathbf{x}$  given the observed data set  $\mathcal{D}$ . This particular form of regression is called *Bayesian regression*.

### 3.3 Bayesian Quadrature

Equation (5) gives a posterior estimate of  $f(\mathbf{x})$  given the observed samples. Then, from (1), a posterior estimate of  $I$ , that is  $\hat{I} = E(I|\mathcal{D})$ , is obtained by integrating both terms of (5):

$$\hat{I} = \bar{I} + \mathbf{z}^t Q^{-1}(\mathbf{Y} - \bar{\mathbf{F}}), \quad (7)$$

with

$$\bar{I} = \int \bar{f}(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}, \quad (8)$$

$$\mathbf{z} = \int \mathbf{k}(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}. \quad (9)$$

It can be seen that the posterior estimate  $\hat{I}$  results from adding to the prior expectation  $\bar{I}$  a corrective term that represents the effect of the observed samples  $\mathcal{D}$ . This term includes three factors:  $(\mathbf{Y} - \bar{\mathbf{F}})$ , which measures how wrong was our prior expectation for each observed sample value,  $Q^{-1}$ , the inverse covariance matrix which accounts for the relative positions of the observed samples, and the  $\mathbf{z}$  vector, which captures the influence of each sample on the deterministic part of the integrand given the covariance function.

### 3.4 Variance Analysis

#### 3.4.1 Variance of the Integral Estimate and Optimal Sampling Pattern

The posterior estimate of  $I$  given by (7) has a Gaussian distribution of mean  $\hat{I}$  and a variance given by

$$\text{Var}(I|\mathcal{D}) = \bar{V} - \mathbf{z}^t Q^{-1} \mathbf{z}, \quad (10)$$

with

$$\bar{V} = \iint k(\mathbf{x}, \mathbf{x}') p(\mathbf{x}) p(\mathbf{x}') d\mathbf{x} d\mathbf{x}'. \quad (11)$$

Note that the estimate of the posterior variance in (10) does not depend on the observed values  $\mathbf{Y}$ . It only depends on the location  $\mathbf{x}$  of the samples. This might seem not plausible at first sight but actually, the covariance function of our prior model (see (4)) already comprises the statistical information necessary to estimate  $\text{Var}(I|\mathcal{D})$ . This variance is due to the implicit variability of our observations given our prior. It can be interpreted as a measure of the confidence we may attribute to our integral estimate. Note also that BMC does not require drawing the samples randomly according to a PDF. However, (10) shows that the  $\text{Var}(I|\mathcal{D})$  strongly depends on the choice of the samples set position  $\{\mathbf{x}_i\}$ . An optimal choice consists in selecting the  $\{\mathbf{x}_i\}$  that minimizes  $\text{Var}(I|\mathcal{D})$ . This optimal set will reflect both the influence of the deterministic function  $p(\mathbf{x})$  (through the  $\mathbf{z}$  vector) and that of the prior knowledge on  $f(\mathbf{x})$  (through the covariance function). The resulting effect can be compared to product sampling or bidirectional sampling techniques that try to obtain information on  $f(\mathbf{x})$  to improve the efficiency of importance sampling.

#### 3.4.2 Bias Considerations

As opposed to CMC, in BMC the observed data are considered as known and thus deterministic whereas the function  $f(\mathbf{x})$  (see (1)) is considered as uncertain. The BMC integral estimate is obtained by computing the integral of the most probable function among all possible realizations of the posterior GP. The estimator of (7) is thus unbiased (in the Bayesian sense) as its value coincides with the expected value of the posterior probability distribution. However, this unbiasedness has nothing to do with the usual interpretation of this term in a classic Monte Carlo framework in which sampling is produced by a random process. That is why unbiasedness considerations can be misleading when comparing classic and Bayesian Monte Carlo, since the randomness in each of the methods is of a very different nature.

Specifically, in a frequentist approach such as CMC, the estimate can be refined by averaging estimates with respect to multiple sample sets whereas a Bayesian method uses averaging with respect to the posterior distribution instead. In [8, ch. 3, Sect. 2], Bishop applies a frequentist bias-

variance analysis to regression and he shows on a parametric regression example how the regularization parameter (which, in BMC, corresponds to the noise ratio introduced in the next section) can be used to control the frequentist bias. However, Bishop points out that this is of limited practical interest since averaging w.r.t. multiple sample sets would affect the efficiency of a Bayesian approach. Indeed, as noted by Bishop, to fully benefit from the Bayesian approach, a direct application of BMC would combine multiple sample sets into one large set rather than considering each set individually. Therefore, in our BMC method, we will consider only one single samples set and apply a full-fledged Bayesian method. Furthermore, the Bayesian framework offers much more efficient methods to implement progressive refinement than averaging with respect to multiple sample sets.

### 3.5 The Case of Stationary Covariance Functions and the Noise Ratio Hyperparameter

In the absence of accurate information on  $f(\mathbf{x})$  variations, it is sensible to assume a stationary covariance function for the prior GP. Note, however, that this does not mean that the prior GP is second-order stationary. For this to be true, the mean function  $\bar{f}(\mathbf{x})$  must be constant, which is not the case in general.

As mentioned in Section 3.1, the variance of the prior GP is constant in the case of stationary covariance functions and we have

$$k(\mathbf{x} - \mathbf{x}') = \sigma_f^2 k'(\mathbf{x} - \mathbf{x}'), \quad (12)$$

where  $\sigma_f^2 = k(0)$  is the prior GP variance and  $k'(\mathbf{x} - \mathbf{x}')$  is its correlation function.  $\sigma_f$  is thus a hyperparameter of our GP model. It represents the strength of the correlation induced by the GP. As the additive noise  $\varepsilon$  is assumed independent from the prior GP, the variance of the observed samples value  $\sigma_y^2$  can be expressed as follows:

$$\sigma_y^2 = \sigma_f^2 + \sigma_n^2.$$

The above equations reveal the  $\sigma_f$  and  $\sigma_n$  hyperparameters as totally independent variables. However, in global illumination problems, many integrals of the same type but with different data have to be computed to render a full image. The observed variance  $\sigma_y^2$  can strongly vary from one integral to the other, but in practice, we can expect that the values of both hyperparameters will roughly have the same order of magnitude as  $\sigma_y^2$ . Therefore, the noise contribution is better characterized by the ratio  $\sigma_n/\sigma_f$  rather than  $\sigma_n$  alone. Henceforth, we will thus use the noise ratio hyperparameter  $\sigma'_n = \sigma_n/\sigma_f$  instead of  $\sigma_n$  to represent the noise contribution.

In the following of this section, we show that once  $\sigma'_n$  has been determined, the value of  $\sigma_f$  is not required to compute the integral estimate  $\hat{I}$ .

Given (12), the  $Q$  matrix defined in (6) can be expressed as follows:

$$Q = \sigma_f^2 Q', \quad (13)$$

with  $Q' = K' + \sigma_n'^2 I_n,$

where  $K'$  is obtained by replacing  $k(\mathbf{x} - \mathbf{x}')$  by  $k'(\mathbf{x} - \mathbf{x}')$ :

$$K'_{i,j} = k'(\mathbf{x}_i - \mathbf{x}_j) \quad \text{with } (i, j) \in [1, n]^2.$$

We can now rewrite (7) with terms that depend on  $\sigma'_n$  instead of  $\sigma_f^2$  and  $\sigma_n^2$ :

$$\hat{I} = \bar{I} + \mathbf{z}'^t Q'^{-1} (\mathbf{Y} - \bar{\mathbf{F}}), \quad (14)$$

where the  $\mathbf{z}'$  vector is obtained by replacing  $k(\mathbf{x} - \mathbf{x}')$  by  $k'(\mathbf{x} - \mathbf{x}')$  in (9). Each element of this vector can then be expressed by the following convolution integral:

$$z'_i = \int k'(\mathbf{x} - \mathbf{x}_i) p(\mathbf{x}) d\mathbf{x}, \quad (15)$$

where  $\mathbf{x}_i$  is a sample location. Assuming that the  $\sigma'_n$  hyperparameter is known, the integral estimate given by (14) does not require determining  $\sigma_f^2$  and  $\sigma_n^2$ . The noise ratio  $\sigma'_n$  can be interpreted as the level of confidence we have in the GP model fitting. The higher is  $\sigma'_n$ , the lower is the proportion of the observed samples variance that can be explained by the GP.

### 3.6 BMC Algorithm Overview

To summarize, the BMC method consists of the following steps:

1. Build the GP prior by choosing a covariance function and a mean function.
2. Learn the hyperparameters associated with the covariance function.
3. Select the set of sampling positions  $\{\mathbf{x}_i\}$  which minimize the variance of the BMC estimate and compute the inverted covariance matrix  $Q'^{-1}$ .
4. Compute the  $\mathbf{z}'$  vector and the vector of quadrature coefficients  $\mathbf{c} = \mathbf{z}' Q'^{-1}$ .
5. Collect the observed samples value  $\mathbf{Y}_i$  for each sampling position  $\{\mathbf{x}_i\}$ .
6. Compute the prior mean value vector  $\bar{\mathbf{F}}$ .
7. Compute the posterior estimate with (14).

As mentioned above, global illumination problems imply the computation of many integrals of the same type with different data and of course, all these steps need not be repeated for each integral evaluation. As a matter of fact, in implementing a BMC algorithm, we will try to transfer the most computer intensive steps to a preprocessing stage. By using the same set of hyperparameters for all integral evaluations, Brouillat et al. [2] have shown that steps 1 to 4 can be preprocessed in the case of diffuse reflection, which makes the computing load of BMC comparable to CMC methods. However, their approach is not applicable to the case of glossy BRDF because of the dependence on the viewing direction. The deterministic function  $p(\mathbf{x})$  that essentially contains the BRDF will then change at each integral evaluation and steps 2 and 3 can no longer be preprocessed as we will see in the following. In the next sections, we will propose a more general approach based on a spherical Gaussian framework that can be applied to a large class of problems.

## 4 A SPHERICAL GAUSSIAN-BASED BMC FRAMEWORK

We will now consider the computation of the illumination integral at a given shading point:

$$L_o(\omega_o) = \int_{\Omega_{2\pi}} L_i(\omega_i) \rho(\omega_i, \omega_o) (\omega_i \cdot \mathbf{n}) d\Omega(\omega_i). \quad (16)$$

In this integral, the analytical part, which corresponds to  $p(\mathbf{x})$  in (1) is naturally the BRDF factor  $\rho(\omega_i, \omega_o)(\omega_i \cdot \mathbf{n})$ . The unknown function  $f(\mathbf{x})$  in (1), modeled with a Gaussian process, is the incident radiance  $L_i(\omega_i)$  at the shading point in (16). Our goal is to compute an estimate of  $L_o(\omega_o)$  from a set of samples value  $\{L_i(\omega_j), j \in [1, n]\}$ , where  $\omega_j$  are the samples location (called  $\mathbf{x}_i$  in Section 3.2).

Throughout this section, we describe our approach to the different problems involved in computing the Bayesian quadrature. We start by describing a spherical Gaussian-based framework for BMC which greatly simplifies the computation of the integrals for the  $\mathbf{z}'$  vector. Then we show how the prior GP covariance and mean functions are built, and how optimal sampling patterns are built. Finally, we elaborate on our rendering algorithm.

### 4.1 Our Theoretical Approach

In this section, we essentially address step 4 of the BMC algorithm presented in Section 3.6 and more particularly the problem of making the computation of the  $\mathbf{z}'$  vector tractable knowing that it is not possible to precompute the  $z'_j$  coefficients given their dependence on  $\omega_o$ . Our approach is to model both the BRDF and the correlation function with spherical Gaussian functions (SGF), a choice which allows us to reduce the  $\mathbf{z}'$  computation to a simple query to a scene-independent two-entry look-up table.

An interesting property of SGFs is that the product of two SGFs yields an SGF. This property will be very useful for the computation of the (15) integral. As far as  $p(\mathbf{x})$  can be modeled by an SGFs product or mixture, the method described in this section can be applied to compute the quadrature of (14) given its linearity properties.

An SGF results from the restriction of a Gaussian RBF (Radial Basis Function) to the unit sphere  $S^2$ . Consequently, for  $x, y \in S^2$ , we have

$$|x - y|^2 = 2(1 - x \cdot y), \quad (17)$$

since  $|x| = |y| = 1$ . An SGF can, thus, be expressed as follows:

$$G(x - y; \mu, \lambda) := \mu \exp\left(\frac{x \cdot y - 1}{\lambda^2}\right).$$

Since to each point  $x \in S^2$  corresponds a direction  $\omega$ , we model the correlation function  $k'$  by an SGF  $G_{k'}$  defined as follows:

$$k'(\omega - \omega') := G_{k'}(\omega - \omega'; 1, l) = \exp\left(\frac{\omega \cdot \omega' - 1}{l^2}\right), \quad (18)$$

where  $l$  is the lengthscale hyperparameter which controls the smoothness of the prior GP. We model the BRDF  $\rho$  by an SGF  $G_\rho$  defined as

$$\rho(\omega_i, \omega_o) := G_\rho(\omega_r - \omega_i; k_s, w) = k_s \exp\left(\frac{\omega_r \cdot \omega_i - 1}{w^2}\right), \quad (19)$$

where  $k_s$  is the specular coefficient.  $G_\rho$  has an axially symmetric lobe whose axis is aligned with  $\omega_r$ , which is itself function of the outgoing direction  $\omega_o$ .  $w$  characterizes the lobe sharpness, i.e., the “width” of the lobe. It is important to note that this choice does not restrict the generality of our analysis as several works have shown that most BRDFs can be approximated by a mixture of SGFs (e.g., in [9]). In particular, one single SGF is sufficient to model the glossy

term of a Phong's BRDF. In this case, the lobe axis direction  $\omega_r$  is the perfect mirror incident direction:

$$\omega_r = 2(\omega_o \cdot \mathbf{n})\mathbf{n} - \omega_o,$$

and using the following approximation:

$$\cos^m \theta = e^{m \ln(\cos \theta)} \approx e^{m(\cos \theta - 1)}, \quad (20)$$

with  $\cos \theta = (\omega_r \cdot \omega_i)$ , we have  $w = 1/\sqrt{m}$ ,  $m$  being the Phong shininess parameter. The resulting approximation RMSE is below  $10^{-3}$  when  $m > 10$ . Note also that modeling BRDFs with SGFs (which are naturally isotropic) does not restrict our framework to the isotropic BRDF case since anisotropic BRDFs can be modeled using a weighted sum of (isotropic) SGFs as mentioned above. Moreover, we have found that the Blinn-Phong BRDF can be modeled with only three SGFs with an RMSE below 0.05. More details regarding the potential application of our approach to multiple-lobe BRDFs are given in Section 6.2.

Given (15), (18), and (19), each coefficient of the  $\mathbf{z}'$  vector can be expressed as follows:

$$z'_j = \int_{\Omega_{2\pi}} G_{k'}(\omega_j - \omega'; 1, l) G_\rho(\omega_r - \omega'; k_s, w) d\Omega(\omega'),$$

where  $\omega_j$  is the direction vector of sample  $j$ . As the product of two SGFs is also an SGF, we have

$$z'_j = \int_{\Omega_{2\pi}} G(\omega_m - \omega'; c_m, l_m) d\Omega(\omega'), \quad (21)$$

with

$$\begin{aligned} \frac{1}{l_m^2} &= \left| \frac{\omega_j}{l^2} + \frac{\omega_r}{w^2} \right|^2, \\ \omega_m &= l_m^2 \left( \frac{\omega_j}{l^2} + \frac{\omega_r}{w^2} \right), \\ c_m &= k_s \exp \left( \frac{1}{l_m^2} - \frac{1}{l^2} - \frac{1}{w^2} \right). \end{aligned}$$

By developing (21), we have

$$z'_j = c_m S_g(\omega_m, l_m),$$

where  $S_g(\omega, l)$  is the spherical Gaussian integral (SGI):

$$S_g(\omega, l) = \int_{\Omega_{2\pi}} \exp \left( \frac{\omega \cdot \omega' - 1}{l^2} \right) d\Omega(\omega'). \quad (22)$$

The computation of the  $\mathbf{z}'$  coefficients is then reduced to evaluations of the SGI of (22) which can easily be tabulated for quick evaluations. Only the elevation angle  $\theta = (\omega, \mathbf{n})$  is necessary to specify the input direction  $\omega$  in (22) given the axial symmetry of the SGI about the normal  $\mathbf{n}$ . Therefore, only a single two-entry table with  $(\theta, l)$  as inputs is necessary for SGI evaluations. This table is independent of the scene and the used BRDFs. It is computed just once and used for any BMC integration within this framework.

## 4.2 Constructing the Prior GP Model

### 4.2.1 Hyperparameters Selection

In the preceding sections, we have shown that only two hyperparameters are needed to compute the integral estimate from (14): the noise ratio  $\sigma'_n$  and the lengthscale  $l$ .

Our strategy for determining appropriate hyperparameters value consists in using the same set of hyperparameters for all illumination integrals that involve the same BRDF. We shall see in Section 5 that this choice is perfectly acceptable since the hyperparameters values mainly depend on the BRDF shininess, although they can vary depending on the scene lighting conditions. This simplification allows us to learn the hyperparameters for each BRDF at a scene level, and then precompute the inverse covariance matrix  $Q^{-1}$  and the optimal sampling pattern (i.e., steps 2 and 3 of the algorithm of Section 3.6).

To learn the hyperparameters at a scene level for a given BRDF, we use the same technique as that of Brouillat et al. [2], with the difference that we sample the incoming radiance with a distribution concentrated within the BRDF lobe. In this way, the fitting of the covariance function is adapted for the range of interest of the intersamples distance, that is, a narrow range of distances for high shininess and a wide one for low shininess. But this approach has a limitation regarding scenes with several BRDFs since it implies learning the hyperparameters separately for each shininess parameter used in the scene. Although this operation takes just a few seconds per shininess parameter (typically around 1 to 4 seconds depending on the scene complexity), it would make the BMC approach inefficient for scenes with different shininess parameters. To cope with this problem, a fast method to approximate the hyperparameters is proposed in Section 5.4.

### 4.2.2 Determining the Mean Function

To fully define the GP, a mean function  $\bar{L}_i(\omega_i)$  corresponding to  $f(\mathbf{x})$  in Section 3.1 must be specified. Our approach to this question is to assume a locally constant mean function  $\bar{L}_i$  that will be determined from the observed samples value  $L_i(\omega_j)$ . A simple average of the samples value would provide a strongly biased  $\bar{L}_i$  estimate since the samples distribution is not at all uniform as explained in Section 4.3. Instead, we leverage the prior GP as described in [6, ch. 2, Sect. 7]. The method consists in inferring a mean function from the observed samples value through an explicit basis function model. In our case, we only use one basis function ( $h(\omega) = 1$ ) and its associated weight is thus the desired constant mean value, which yields:

$$\bar{L}_i = \frac{HQ'^{-1}Y}{HQ'^{-1}H^t}, \quad (23)$$

where  $H = [h(\omega_1), \dots, h(\omega_n)] = [1, \dots, 1]$  and  $Y = [L_i(\omega_1), \dots, L_i(\omega_n)]^t$  is the vector of observations. As shown in [6], this method slightly increases the estimates variance but this can be neglected in practice. Then from (8) and (19), we have

$$\bar{I} = \bar{L}_i \mu(\omega_r), \quad (24)$$

where  $\mu(\omega_r)$  is the SGI:

$$\mu(\omega_r) = \int_{\Omega_{2\pi}} G_\rho(\omega_r - \omega; k_s, w) d\Omega(\omega).$$

To evaluate this integral, the same two-entry table as the one required for evaluating the SGI of (22) can be used.

Moreover, since  $H = [1, \dots, 1]$ , (23) can be expressed as a weighted sum of the observed samples value:

$$\bar{L}_i = \frac{1}{\Gamma} \sum_j \gamma_j Y_j,$$

where the weight  $\gamma_j$  is equal to the sum of the coefficients of the  $j$ th column of the  $Q'^{-1}$  matrix and  $\Gamma = \sum_j \gamma_j$  is the sum of all coefficients of the  $Q'^{-1}$  matrix. All the  $\gamma_j/\Gamma$  weights can be precomputed once the sampling pattern and the hyperparameters have been determined.

#### 4.3 Optimal Sampling Pattern

The choice of the sample directions  $\{\omega_j\}$  is crucial to the quality of the rendering integral estimate, especially in the case of glossy BRDFs. Nevertheless, not only the BRDF but also the characteristics of the incident radiance function  $L_i$  should be taken into account for this choice. These characteristics are represented in a stochastic manner through the hyperparameters of the prior GP.

The variance estimate in (10) factors in the influence of the BRDF (through  $\mathbf{z}'$ ), the prior GP hyperparameters, and the sampling pattern defined by the  $\{\omega_j\}$  set. Moreover, since the global hyperparameters  $\sigma_n'$  and  $l$  are known at this stage (see Section 3.6), the variance only depends on the  $\{\omega_j\}$  set. This is a strong feature of the BMC framework since we can optimize the samples direction to minimize  $\text{Var}(I|\mathcal{D})$ .

Optimizing the directions for a large set of samples can become cumbersome. A naive approach would be to consider that the variance is a function of  $n$  variables which are the samples directions. But minimizing such a function would become intractable even for medium size sample sets (40 samples or more). Our solution to this problem consists in modifying the spiral points algorithm [10] as shown in Algorithm 1.

**Algorithm 1.** The modified spiral points algorithm.

```

1:  $\Delta\phi \leftarrow \pi(3 - \sqrt{5})$             $\triangleright$  Compute the step on  $\phi$ 
2:  $\phi \leftarrow 0$                            $\triangleright$  Initialize  $\phi$  for the first sample
3:  $\Delta z \leftarrow 1/n$                       $\triangleright$  Compute the step on  $z$ 
4:  $z \leftarrow 1 - \Delta z/2$                  $\triangleright$  Initialize  $z$  for the first sample
5: for all  $k \leftarrow [1 : n]$  do
6:    $z_p \leftarrow \text{polyval}(\text{coeff}, z)$        $\triangleright$  Covariance effect
7:    $z_v \leftarrow \frac{1}{m} \ln[1 + z_p(e^m - 1)]$      $\triangleright$  BRDF effect
8:    $\theta_k \leftarrow \arccos(\min(z_v, 1))$          $\triangleright$  Compute  $\theta$  angle
9:    $\phi_k \leftarrow \text{mod}(\phi, 2\pi)$              $\triangleright$  Compute  $\phi$  angle
10:   $z \leftarrow z - \Delta z$                        $\triangleright$  Give a step on  $z$ 
11:   $\phi \leftarrow \phi + \Delta\phi$                    $\triangleright$  Give a step on  $\phi$ 
12: end for
```

The original algorithm generates a discrete spiral on a sphere about the up axis  $z$  with a constant pitch ( $\Delta z$ ). In our application, the  $z$ -axis is aligned with the surface normal  $\mathbf{n}$ . The  $z$ -coordinates of the original spiral points are thus uniformly distributed. We replace the uniform distribution by a polynomial distribution (whose coefficients are called  $\text{coeff}$  in Algorithm 1, line 6) to account for the covariance effect (line 6). In addition, we change this algorithm to produce a discrete spiral on the BRDF lobe (line 7) rather than on the sphere. In Section 4.4, we show

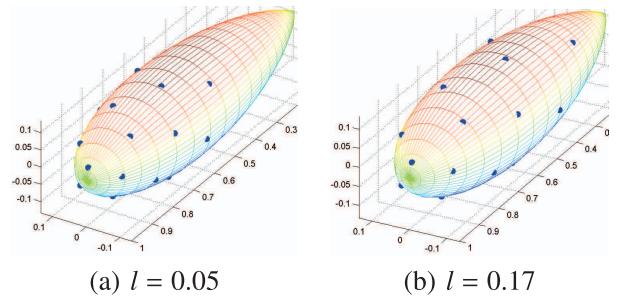


Fig. 2. Optimal sampling patterns of size 20 computed for two different lengthscale values and a purely glossy Phong BRDF with a shininess  $m = 50$ . Each blue point on the lobe corresponds to a sample direction.

that this can be obtained by warping the  $z$ -coordinates with the following function:

$$z_v = g(z) = \frac{1}{m} \ln[1 + z(e^m - 1)],$$

where  $m$  is the Phong shininess parameter.

The optimizer will then take the polynomial coefficients  $\text{coeff}$  as input arguments to minimize  $\text{Var}(I|\mathcal{D})$  instead of taking as input all the sampling directions. The number of parameters to optimize is thus reduced to the number of coefficients of the polynomial (a polynomial of degree 3 is sufficient). Actually, only the term  $-\mathbf{z}' Q^{-1} \mathbf{z}$  of (10) needs to be considered as  $\bar{V}$  does not depend on the sampling pattern. The convergence is very fast (a few seconds) with the BFGS quasi-Newton method provided by Matlab and the result is not very sensitive to the initial values even if the initial polynomial is of degree 0.

The variance of the BMC estimate strongly depends on the correlation function which is, in its turn, parametrized by the lengthscale hyperparameter. We can thus expect the optimal sample set to depend on the lengthscale value. A very smooth  $L_i$  function implies a large lengthscale value. Consequently, a high concentration of samples around the BRDF apex would be redundant as the samples are highly correlated. This explains the sparsity of the samples on the lobe (see Fig. 2b). Conversely, if  $L_i$  has a low lengthscale value the samples will tend to be concentrated around the BRDF lobe apex (see Fig. 2a). In a preprocessing step, we compute optimal sampling patterns for a discrete set of pairs  $(l, m)$ ,  $l$  and  $m$  being the lengthscale and the Phong shininess parameter, respectively. These optimal sampling patterns are independent of the scene to be processed. When rendering, we select the appropriate sampling patterns from the precomputed ones.

Note that the use of an optimized sampling pattern is beneficial to fully exploit the BMC method. It allows accounting for the prior information about the unknown function  $L_i$  on the samples distribution. This aspect is detailed in Section 5.

#### 4.4 The $z$ Warping Function for Gaussian Lobe

Our approach to this problem is to exploit the property of every function  $f$  on  $S^2$  that:

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{j=1}^n f(\omega_{j,n}) = \frac{1}{4\pi} \int_{S^2} f(\omega) d\Omega(\omega),$$

for configurations of asymptotically uniformly distributed sampling directions  $\{\omega_{1,n}, \dots, \omega_{n,n}\}$  [10]. Therefore, at the beginning of the optimization algorithm, if the initial sampling pattern is such a configuration, all samples can contribute equally to the integral estimate, which can be considered as a sensible choice for a starting point. Our goal in this section is to show how such configurations can be built with the spiral points algorithm. In Section 4.3, we have assumed that the lobe central axis is aligned with the surface normal ( $\omega_r = \mathbf{n}$ ) and consequently, using the spherical Gaussian BRDF expression given by (19), the illumination integral of (16) becomes:

$$L_o(\omega_o) = k_s \int_{\Omega_{2\pi}} L_i(\omega_i) \exp[m(\mathbf{n} \cdot \omega_i - 1)] d\Omega(\omega_i).$$

This equation can be further developed as follows:

$$L_o(\omega_o) = k_s \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\frac{\pi}{2}} L(\theta, \phi) e^{m(\cos \theta - 1)} \sin \theta d\theta d\phi,$$

where  $(\theta, \phi)$  are the polar coordinates of the  $\omega_i$ . If we make the substitution  $z = \cos \theta$ , we obtain

$$L_o(\omega_o) = k_s \int_{\phi=0}^{2\pi} \int_{z=0}^1 L(z, \phi) e^{m(z-1)} dz d\phi,$$

If we directly sample the function  $L(z, \phi) e^{m(z-1)}$ , the distribution over the hemisphere  $\Omega_{2\pi}$  must be uniform as stated above, which is precisely what the original spiral points algorithm does. But if we sample the function  $L(z, \phi)$  only, another substitution is necessary to include the BRDF factor into new integration variables. This is possible by making the substitution

$$z' = \frac{e^{mz} - 1}{e^m - 1}, \quad (25)$$

and then

$$dz' = \frac{m}{1 - e^{-m}} e^{m(z-1)} dz,$$

which leads to:

$$L_o(\omega_o) = \frac{k_s(1 - e^{-m})}{m} \int_{\phi=0}^{2\pi} \int_{z'=0}^1 L(z', \phi) dz' d\phi.$$

Note that the integration domain remains the hemisphere  $\Omega_{2\pi}$ . This domain is then uniformly sampled as for the original integral but we still need to revert to the original  $z$ -coordinate to obtain the polar coordinates of the sampling directions. By inverting (25), we have

$$z = \frac{1}{m} \ln[1 + z'(e^m - 1)],$$

which gives the warping function we need for our spiral point algorithm. Its effect can be interpreted as a warping of the distance measure so that intersamples distances are measured between points on the Gaussian lobe rather than on the unit sphere.

## 4.5 The Rendering Algorithm

The BMC integrator described in Algorithm 2 is applied to every shading point. The BMC function (line 1) has two

parameters: the outgoing direction  $\omega_o$  and the surface normal  $\mathbf{n}$ . It returns the BMC estimate of  $L_o(\omega_o)$  as defined in (16). All global data for the whole scene including the hyperparameters  $(l, \sigma_n')$ , the inverse covariance matrix  $Q'^{-1}$ , the optimal sampling pattern  $SS$  and the BRDF parameters  $(w, k_s)$  are assumed to belong to the program global scope and are thus not explicitly declared.

**Algorithm 2.** The Bayesian Monte Carlo integrator.

```

1: Function BMC ( $\omega_o, \mathbf{n}$ )
2:    $SS_{rotated} \leftarrow \text{rotateRandom}\phi(SS)$ 
3:    $\omega_r \leftarrow \text{reflect}(\omega_o, \mathbf{n})$ 
4:    $SS_{rotated} \leftarrow \text{centerOnR}(SS_{rotated}, \omega_r)$ 
5:   for all  $\omega_j \in SS_{rotated}$  do
6:      $\mathbf{Y}[j] \leftarrow L_i(\omega_j)$ 
7:   end for
8:    $\bar{L}_i \leftarrow \text{computeWeightedMean}(\mathbf{Y})$ 
9:    $\bar{I} \leftarrow \bar{L}_i \times \mu(\omega_r)$ 
10:   $\bar{\mathbf{F}} \leftarrow \bar{L}_i \times \mathbf{H}$ 
11:   $\mathbf{z}' \leftarrow \text{computeZvector}(\omega_r, w, SS_{rotated})$ 
12:   $L_o \leftarrow \bar{I} + \mathbf{z}'^t \times Q'^{-1} \times (\mathbf{Y} - \bar{\mathbf{F}})$ 
13:  return  $L_o$ 
14: end function
```

The optimal sampling pattern, initially centered around the surface normal  $\mathbf{n}$ , is rotated around  $\mathbf{n}$  (line 2) by a random  $\phi$  value. The idea behind this random rotation is to introduce a scrambling effect that allows masking the visibility of the regular sampling pattern, hence avoiding the introduction of artifacts in the final image. Note that since the covariance function is an SGF dependent only on the intersamples distance, it is invariant to rotations of the whole sampling pattern. This property allows generating different optimal sampling patterns by randomly rotating the original optimal sampling pattern while keeping the same matrix  $Q'^{-1}$ . The mirror incident direction  $\omega_r$  is computed at line 3. An additional rotation aligns the central axis of the sampling pattern with  $\omega_r$  (line 4). The loop beginning at line 5 numerically evaluates the samples value for each sampling direction  $\omega_j$ .  $\bar{L}_i$  is computed at line 8 according to (23) and  $\bar{I}$  is computed at line 9 according to (24). At line 11,  $\mathbf{z}'$  is computed as described in Section 4.1 and (21) through a lookup in a scene-independent pre-computed table of SGI values. Finally, the Bayesian quadrature (see (14)) is applied to estimate  $L_o$ , which amounts to a vector-matrix and a vector-vector product.

As the sampling pattern  $SS$  is centered around  $\omega_r$ , a part of the sampled BRDF lobe may be located under the plane tangent to the object surface at the shading point; in other words, some of the directions of the sampling pattern may be occluded by the object surface. A straightforward solution would be to simply discard these samples. But this would imply recomputing and inverting the covariance matrix whenever one or more samples of the samples set lie outside the domain of integration. To avoid this computational overhead, we use a fixed number of samples, which raises the problem of assigning a value to the occluded samples.

The values of the occluded samples could be naively set to zero, but such a method could potentially introduce wrong information with an important effect on the

TABLE 2  
Test Profiles

Scene	triangles	$m$	$l_l$	$\sigma'_{nl}$
Dragon	100K	20	0.29	0.96
Buddha	170K	50	0.17	0.53
Horse	110K	80	0.13	0.67
VW	440K	20, 50, 80	nd	nd
Room	540K	20, 50, 80	nd	nd
Plates	20	10, 50, 80, 200	nd	nd

Columns list the scene characteristics, the Phong shininess parameter  $m$  and the learned hyperparameters  $l_l$ ,  $\sigma'_{nl}$ . Values of  $l_l$  and  $\sigma'_{nl}$  tagged as “nd” indicate that the approximate method of hyperparameters is used, as shown in Section 5.4.

estimated integral value. The reason for this is that although the considered sampling direction  $\omega_j$  is outside the integration domain, i.e.,  $\omega_j \notin \Omega_{2\pi}$ , it may still contribute to the reconstruction of the function  $L_i(\omega_i)$  within the integration domain. Setting the value of such samples to zero would introduce wrong information and force the reconstructed signal to tend toward zero when approaching the surface tangent plane. Indeed, there is no reason why an artificial discontinuity of the incident radiance should be introduced at the tangent plane since occlusion due to the surface is already accounted for through the choice of the integration domain. We have experimentally found that an efficient solution to this problem is to assign to such samples the value of the closest sample that lies in the unoccluded part of the lobe. In particular, this solution performs better than assigning the occluded samples the mean value of the samples within the integration domain.

## 5 RESULTS

### 5.1 Experimental Setup

To assess the effectiveness of the BMC method, its results are compared with those of BRDF-based low-discrepancy Monte Carlo importance sampling using a Halton sequence (LDIS), and pure random Monte Carlo importance sampling (MCIS). To evaluate the importance of the optimized sample sets in the BMC integration we also tested BMC using LD-based sample sets (LDBMC). The efficiency of each method is computed in terms of root mean square error (RMSE) with respect to reference images computed using LDIS with 10,000 samples per pixel. We have chosen to test our framework using Phong shininess coefficients  $m$  varying from 10 to 200, a range of values which we consider illustrative of the common usage of glossy BRDFs. Six different scenes have been used, the details of which are presented in Table 2. In some scenes (i.e., Dragon, Buddha, and Horse), for the sake of computation speed, incident illumination is simulated by an environment map, a simplification that does not limit the significance of our results and conclusions since our goal is to show that we can obtain better estimates with less samples and a negligible computing overhead whatever the used global illumination method. In other scenes (i.e., VW and Room), the results are generated using final gathering for photon mapping. As for the Plates scene, only direct incident illumination was computed. The results were generated at a resolution of  $1,024 \times 1,024$  pixels for all the scenes except

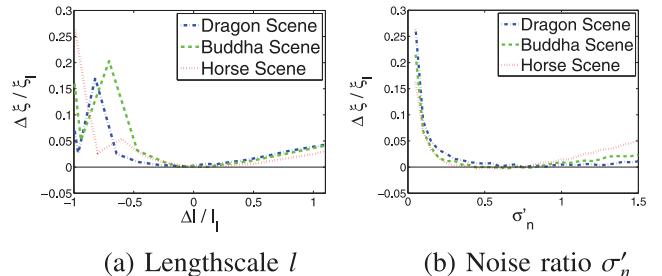


Fig. 3. Relative RMSE difference when varying one of the hyperparameters.  $\Delta\xi = \xi - \xi_l$ ,  $\xi$  being the RMSE of an image computed when varying one hyperparameter, and  $\xi_l$  being the RMSE of an image computed with the learned hyperparameters (listed in Table 2). In (a),  $\sigma'_{nl}$  is fixed to its learned value, while in (b) the lengthscale  $l$  is assigned its learned value. The abscissa axis of the lengthscale plot (a) is the relative difference between the tested lengthscale and the learned value lengthscale  $l_l$ .

the Plates scene ( $1,152 \times 768$  pixels), using a 64 bit machine equipped with a 2-GHz Intel Core i7 processor and 8 GB of RAM. The Mitsuba raytracer [11] was used to implement the methods of MCIS, LDIS, LDBMC, and BMC.

### 5.2 Hyperparameters Learning

The validity of our prior model depends on the good choice of the hyperparameters. Table 2 lists the hyperparameter values obtained by learning for the scenes Dragon, Buddha, and Horse. The learning method provides us with the hyperparameters that allow the best global adaptation of the GP model. But it is not guaranteed that these hyperparameters yield the best integral estimate. To validate our hyperparameters learning method, we have measured the RMSE for different values of the hyperparameters (see Fig. 3).

Let  $l_l$  and  $\sigma'_{nl}$  be the learned hyperparameters listed in Table 2. To generate the RMSE plots of Fig. 3a, we have fixed the value of  $\sigma'_{nl}$  to  $\sigma'_{nl}$  and computed the RMSE for different values of  $l$ . We have repeated this process for the plot of Fig. 3b by varying the value of  $\sigma'_{nl}$  while keeping  $l = l_l$ . Fig. 3a shows that the RMSE is minimum around  $\Delta l = l_l - l = 0$  for the three scenes, which means that the learned lengthscale value  $l_l$  is appropriate for integration. Furthermore, Fig. 3b shows that we can draw the same conclusion for  $\sigma'_{nl}$ . Note that the RMSE is more sensitive to negative than to positive deviations from the optimal value. A more detailed analysis of this interesting feature will be presented in Section 5.4.

### 5.3 Comparison with Importance Sampling

Fig. 4 illustrates the results obtained with LDIS and BMC for the three scenes with a varying number of samples. The rendering times of both methods are similar as BMC with a single BRDF entails a negligible overhead compared to LDIS. The close-up views clearly demonstrate that BMC achieves a reduction of high-frequency noise as compared to LDIS for the same number of samples. The RMSE is also reduced.

Fig. 5 shows the RMSE plots as a function of the number of samples for MCIS, LDIS, LDBMC, and BMC. We can observe that the RMSE of BMC is consistently lower than that of the other methods. This result confirms the advantage of BMC compared to LDIS observed in Fig. 4.

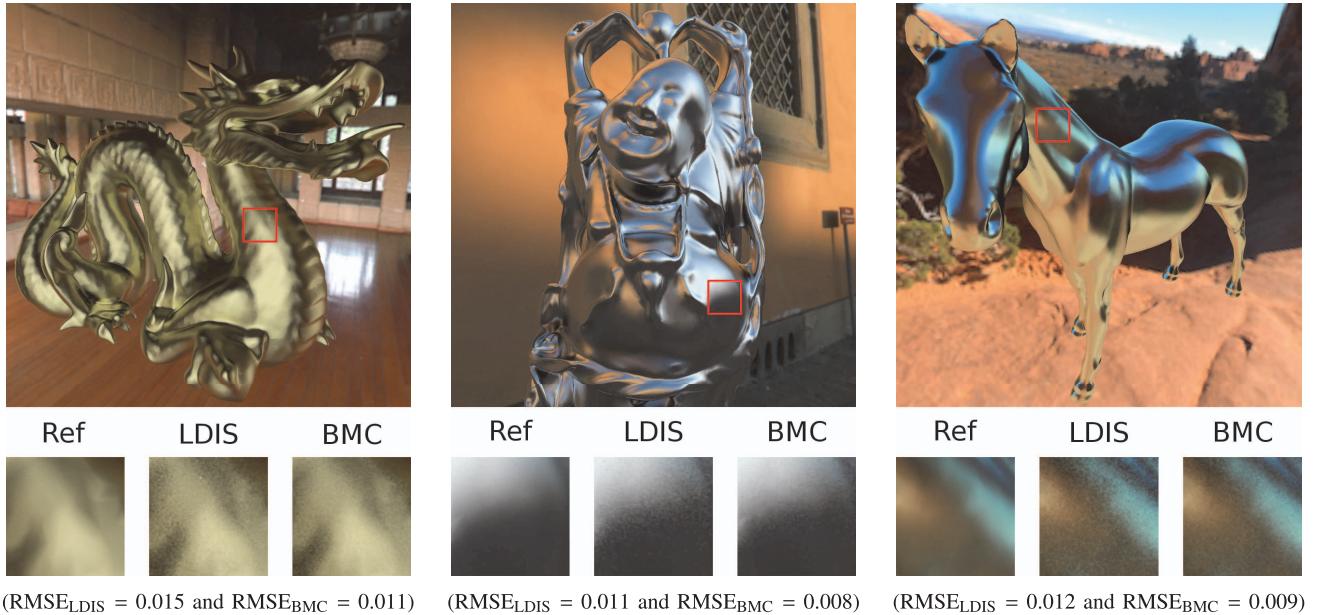


Fig. 4. Close-up views for three scenes rendered with Bayesian Monte Carlo (BMC) and low-discrepancy importance sampling (LDIS). The number of ray samples per visible point is 40 for the Dragon and the Buddha, and 20 for the Horse. Each of these objects has a purely glossy Phong BRDF with a shininess parameter of 20, 50, and 80, respectively, from left to right.

Moreover, it is interesting to note that the slope of the straight line fit of the RMSE plot as a function of the number of samples  $n$  is slightly but consistently steeper for the BMC method ( $n^{-0.72}$ ) than for LDIS ( $n^{-0.68}$ ). Recall that the theoretical optimal rate of convergence for Quasi-Monte Carlo (QMC) integration over the unit sphere in  $\mathbb{R}^3$  is of order  $n^{-0.75}$  [12] under minimal integrand smoothness assumption. LDIS is below the  $n^{-0.75}$  rate because this optimal rate assumes a hypothesis on the smoothness of the integrand which is generally not fulfilled by the incident radiance function. In BMC, the embedded Bayesian regression smooths out the discontinuities before integration, which explains its better performances with respect to the convergence rate. Numerical simulations based on the variance expression given by (10) show that this rate can even be better than  $n^{-0.75}$  for lower noise ratio, that is when the GP prior fits well the incident radiance function.

Note also in Fig. 5 that the results obtained with LDBMC, when low discrepancy sequences are used instead of optimized point sets, do not show significant improvements compared to standard importance sampling

except for small point set sizes. This result shows the importance of optimized point sets for BMC. It is only through an appropriate samples distribution that the information brought by the prior covariance function can be efficiently exploited.

#### 5.4 Skipping the Learning Step

A detailed analysis of the sensitivity of the method suggests that BMC can still achieve good performances using approximated values for  $l$  and  $\sigma'_n$ , which would allow skipping the learning step. Indeed, Fig. 3b shows that for our test cases, we can use any value of  $\sigma'_n$  in the interval  $[0.4, 1.1]$  without incurring a significant RMSE increase. However, we have observed that when increasing  $\sigma'_n$ , the error tends to increase faster in the areas of high variance of the incident radiance and is thus more conspicuous. We have found experimentally that  $\sigma'_n = 0.5$  is a good tradeoff.

Fig. 3a shows that the estimation of the integral value is not very sensitive to positive variations of  $l$ . Moreover, Fig. 6 demonstrates that the learned values of lengthscale exhibit a dependence on the shininess parameter  $m$ , such

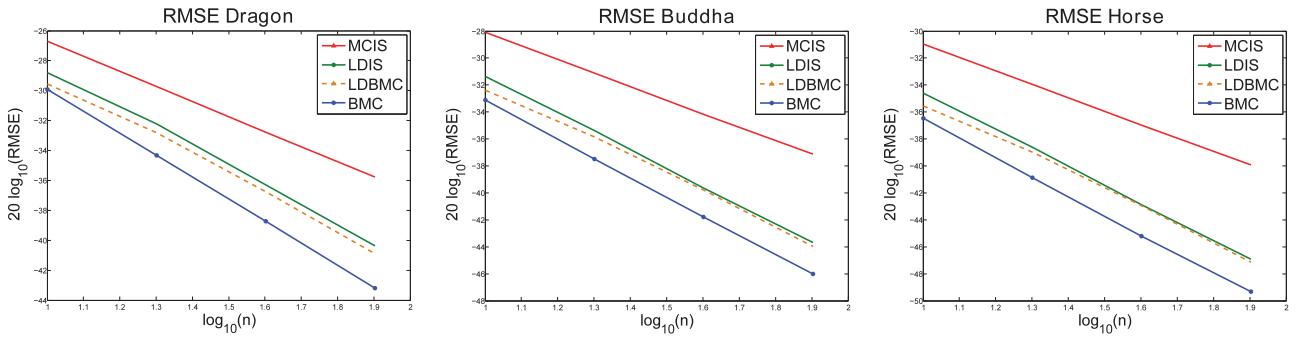


Fig. 5. RMSE plots as a function of the number of samples  $n$ . Note that the slope of the line fit for BMC ( $n^{-0.72}$ ) is steeper than for LDIS ( $n^{-0.68}$ ), LDBMC ( $n^{-0.67}$ ), and for MCIS ( $n^{-0.50}$ ).

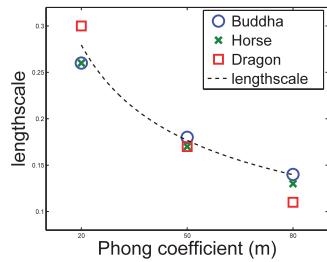


Fig. 6. Fitting of the learned lengthscales as a function of the shininess parameter  $m$ .

that  $l_l \approx \alpha/\sqrt{m} = \alpha \times w$  where  $m = 1/w^2$  as defined in Section 4.1.

A linear regression on the learned lengthscale values yields  $\alpha \approx 1.25$  (see Fig. 6). To test this fast hyperparameter derivation method, we have compared the performances of BMC without hyperparameters learning and LDIS for three different scenes (VW, Room, and Plates). The results are shown in Figs. 1, 7, and 8 and demonstrate that BMC still clearly outperforms LDIS both in terms of RMSE and visual quality. We have found that the slope of the line fit on the resulting RMSE plot is steeper for BMC, similarly to what we have obtained with learned hyperparameters in Fig. 5.

Fig. 8 shows images computed with the LDIS (see Fig. 8a) and BMC (see Fig. 8b) methods. The scene is made up of four plates, each one having a different shininess coefficient. It contains seven light sources of variable size and variable radiance producing an incident radiance along the plates of variable frequency. Note that this frequency increases as we approach the center of the plates. The objective is to show, through this scene, how our BMC method behaves for different frequencies of the incident radiance and different shininess coefficients. Both the BMC and LDIS methods exhibit the same general features: for the same illumination conditions the noise increases as  $m$

decreases, and for the same shininess coefficient  $m$  the noise is higher for sharp variations of the incident radiance. We can notice that the BMC method provides results with a lower noise when compared to the LDIS method. This can be explained by the fact that in BMC the GP acts as a low-pass filter. The color transitions also appear smoother in the BMC images than in the LDIS images.

## 6 DISCUSSION

### 6.1 Possible Improvements

The results we have obtained demonstrate the soundness of our approach. However, the implementation of BMC presented in this paper is far from exploiting the full potential of the Bayesian approach. Indeed the main advantage of BMC over other methods lies in its ability to incorporate the prior knowledge, which can be obtained in two complementary ways. First by specifying a more accurate mean function for the prior GP with, for example, a rough approximation of incoming radiance using spherical harmonics or an SGF mixture (to this regard, there are some similarities with the control covariate method [13]). Second by improving the covariance function adaptation. In particular, the hyperparameters can be adapted locally to better fit the local radiance function characteristics. For example, different classes of pixels can be distinguished based on the samples variance. In this way, different hyperparameters can be chosen for each class. Indeed, the pixels for which the samples variance is high have generally more impact on the visual quality and should be assigned proper hyperparameters value whereas the hyperparameters choice is much less critical for low variance pixels. As these latter pixels represent the great majority, this explains the low RMSE sensitivity to hyperparameters variation observed in Fig. 3. Furthermore, noise ratios are expected to decrease with local hyperparameter adaptation

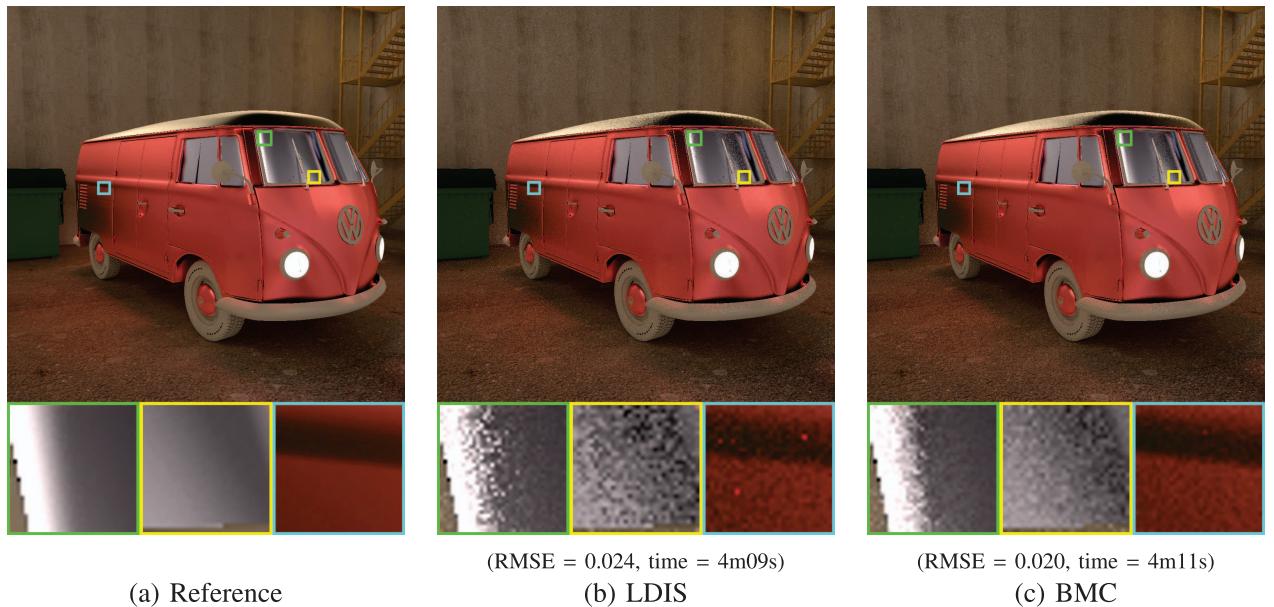


Fig. 7. Indirect radiance component for the VW scene rendered with LDIS (b) and BMC (c). The shown images have been multiplied by a factor of 4. The RMSE and the time are computed by considering only the glossy component. Sixteen ray samples per visible point were used for the materials with a glossy BRDF (VW glass, VW bodywork, and VW roof), while 64 samples were used for the diffuse BRDFs.

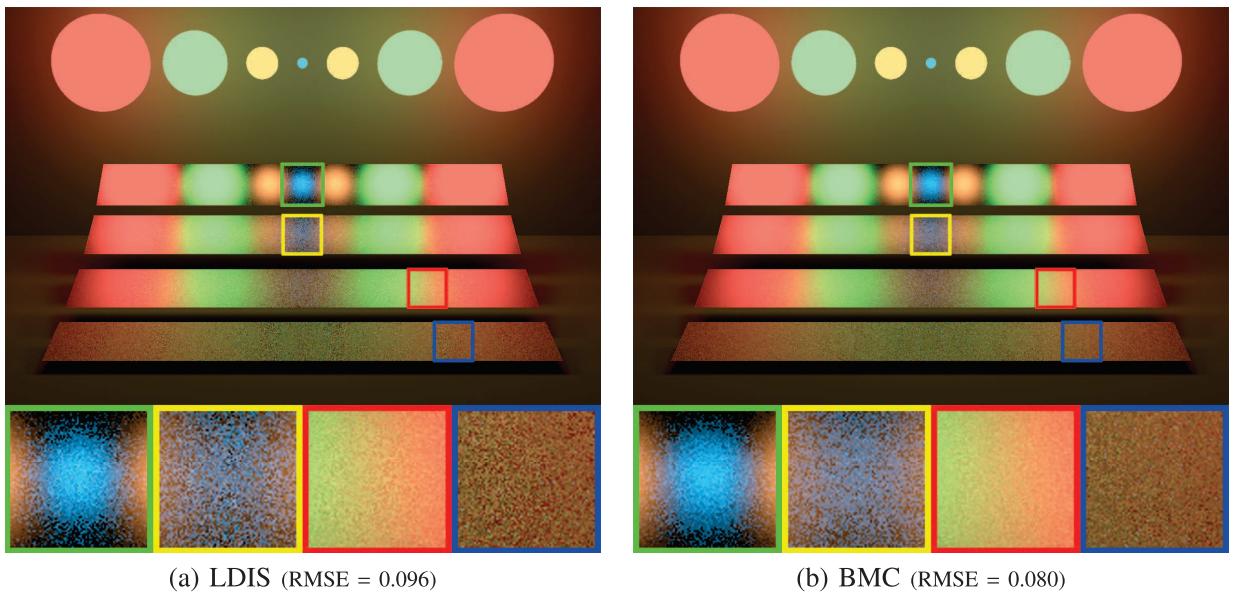


Fig. 8. Direct radiance component for the Plates scene rendered with LDIS (a) and BMC (b). The RMSE has been computed considering only objects with a glossy BRDF (i.e., the four plates), for which 32 ray samples per visible point were used. The Phong shininess coefficient  $m$  of each of the plates is 10, 50, 80, and 200 from bottom up, respectively.

and consequently higher convergence rate can be obtained as explained in Section 5.3.

## 6.2 BMC with Many Complex BRDFs

In this paper, we have focused our attention to single spherical Gaussian BRDF since it is the basic component in our approach to illumination integral computation. In Section 4.1, we have given some indications on how to generalize our approach to any type of BRDF, either analytical or measured, but further research work is necessary to develop an efficient implementation. In the following, we try to be more specific on the different possible approaches. If a BRDF is approximated by a weighted sum of  $n$  SGFs, it is easy to see that the illumination integral can be broken down into a weighted sum of  $n$  integrals of the same type as (16). Our method can then be applied separately to each term of the sum and each integral will be assigned a samples set of size proportional to its weight.

However, this straightforward method cannot be optimal in terms of sampling efficiency since each integral estimate will not benefit from the information brought by the samples used in the other integrals. An alternative to this method is then to use a single samples set for all the BRDF lobes. In this case, only one global covariance matrix is necessary for computing the integral estimate with (14). In (15), since the function  $p(\mathbf{x})$  that represents the BRDF is expressed as a weighted sum of SGFs  $\sum_j \alpha_j G_j()$ , the  $\mathbf{z}'$  vector in (14) will also be expressed as a weighted sum of vectors  $\sum_j \alpha_j \mathbf{z}'_j$  and each  $\mathbf{z}'_j$  can be computed with the method described in Section 4.1. As for samples set optimization, the morphing function can be built from an approximation of the BRDF leading to a simpler analytical expression.

As mentioned above, one of the main problems to solve in BMC lies in the computation of the  $\mathbf{z}'$  vector and the complexity of this computation depends on the choice of the deterministic function  $p(\mathbf{x})$  in (1). So far, we have considered in a rigid manner that this function corresponds

exactly to the BRDF but actually this choice can be made more flexible since  $p(\mathbf{x})$  does not need to be an accurate approximation of the BRDF. If a function  $g(\mathbf{x})$  (e.g., a weighted sum of SGFs) is a quite raw approximation of  $p(\mathbf{x})$ , it is possible to consider that the unknown function is  $f(\mathbf{x})p(\mathbf{x})/g(\mathbf{x})$  and substitute  $g(\mathbf{x})$  to  $p(\mathbf{x})$  in (15). This will not change very much the behavior of the unknown function as the factor  $p(\mathbf{x})/g(\mathbf{x})$  is generally a very smooth function but it could greatly simplify the computation of the  $\mathbf{z}'$  vector for complex BRDFs.

As regards the application of BMC for scenes with a large number of different BRDFs, the single problem that could arise is the need for precomputing an optimized samples set for each shininess coefficient. This operation is much lighter than learning the hyperparameters and is scene independent. But still, if one wants to avoid performing this operation for each different shininess coefficient, the same strategy as the one applied for the hyperparameters could be used: optimize the sample sets for a subset of shininess values (which results in a polynomial of degree 3 as described in Section 4.3) and interpolate the values of these polynomials for the intermediate values.

## 6.3 Limitations

As explained in this paper, the main difficulties that we are faced with in implementing BMC lie in the computation of the inverted covariance matrix and the  $\mathbf{z}'$  vector. Our proposal in this paper is to solve this problem by precomputing optimized points sets and using spherical Gaussian mixtures. As far as this solution is applicable, there is no reason why BMC should not perform better than CMC methods.

Directly applying our framework to the case of a very large samples set would increase the memory footprint and make costly the preprocessing step because of the inversion of the covariance matrix and the samples set optimization.

One possible solution is to resort to a progressive approach which would refine the integral estimate by using successive small sample sets. At each iteration, the prior GP for the current estimate (for the current samples set) would use the posterior GP of the previous estimate (for the previous samples set). In this way, the estimate is progressively refined.

The extension of BMC to larger dimensionality, such as illumination integral calculations requiring image plane sampling, lens sampling and multiple reflections, is not obvious. Indeed, finding an appropriate prior model (definition of the known and uncertain functions, determination of the hyperparameters and the mean function) while keeping a good tradeoff between efficiency and performance is a difficult task.

## 7 CONCLUSION

In this paper, important steps are given toward the goal of reaching a general Bayesian Monte Carlo approach for solving global illumination problems, which was until now limited to the diffuse BRDF case. Modeling the BRDF and the covariance function with the same family of spherical Gaussian functions is a key point, since it allows making the computation of the Bayesian quadrature coefficients tractable for glossy BRDFs. The results confirm that Bayesian Monte Carlo outperforms the classical Monte Carlo methods. In a context where rendering times are dominated by the cost of sampling, Bayesian Monte Carlo has a clear advantage over low discrepancy Monte Carlo importance sampling since it exhibits less noise and lower RMSE for the same number of samples, while having a negligible overhead. Finally, we demonstrated that Bayesian Monte Carlo outperforms low discrepancy Monte Carlo importance sampling even with a suboptimal hyperparameters setting that avoids the learning step. Future work will aim at improving the efficiency of Bayesian Monte Carlo as explained above and also at extending its use to other problems of larger dimensionality, for example, measured BRDFs, multiple reflections, depth blur, multi-view rendering, among others.

## ACKNOWLEDGMENTS

The authors thank the reviewers for their valuable comments which helped improving the paper. This work is partially funded by National Funds through the FCT - Fundação para a Ciencia e a Tecnologia (Portuguese Foundation for Science and Technology) within project PEst-OE/EEI/UI0752/2011.

## REFERENCES

- [1] A. O'Hagan, "Bayes-Hermite Quadrature," *J. Statistical Planning and Inference*, vol. 29, no. 3, pp. 245-260, 1991.
- [2] J. Brouillat, C. Bouville, B. Loos, C. Hansen, and K. Bouatouch, "A Bayesian Monte Carlo Approach to Global Illumination," *Computer Graphics Forum*, vol. 28, no. 8, pp. 2315-2329, <http://dx.doi.org/10.1111/j.1467-8659.2009.01537.x>, 2009.
- [3] C.E. Rasmussen and Z. Ghahramani, "Bayesian Monte Carlo," *Proc. Conf. Neural Information Processing Systems*, pp. 489-496, 2002.
- [4] T. Pfingsten, D.J.L. Herrmann, and C.E. Rasmussen, "Model-Based Design Analysis and Yield Optimization," *IEEE Trans. Semiconductor Manufacturing*, vol. 19, no. 4, pp. 475-486, Nov. 2006.
- [5] A. Geiger, "Gaussian Process for Machine Learning," <http://www.rainsoft.de/projects/gausspro.html>, 2007.
- [6] C.E. Rasmussen and C.K.I. Williams, *Gaussian Process for Machine Learning*. MIT Press, 2006.
- [7] J. Krivánek and M. Colbert, "Real-Time Shading with Filtered Importance Sampling," *Computer Graphics Forum*, vol. 27, no. 4, pp. 1147-1154, 2008.
- [8] C.M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006.
- [9] J. Wang, P. Ren, M. Gong, J. Snyder, and B. Guo, "All-Frequency Rendering of Dynamic, Spatially-Varying Reflectance," *ACM Trans. Graphics*, vol. 28, pp. 133:1-133:10, <http://doi.acm.org/10.1145/1618452.1618479>, Dec. 2009.
- [10] E. Saff and A. Kuijlaars, "Distributing Many Points on a Sphere," *Math. Intelligencer*, vol. 19, no. 1, pp. 5-11, 1997.
- [11] W. Jakob, "Mitsuba Renderer," <http://www.mitsuba-renderer.org>, 2010.
- [12] J.S. Brauchart and J. Dick, "Quasi-Monte Carlo Rules for Numerical Integration over the Unit Sphere  $\mathbb{S}^2$ ," <http://arxiv.org/abs/1101.5450>, Jan. 2011.
- [13] E. Lafontaine and Y. Willem, "A 5D Tree To Reduce the Variance of Monte Carlo Ray Tracing," *Proc. Sixth Eurographics Workshop Rendering*, pp. 11-20, 1995.



**Ricardo Marques** received the MSc degree in computer graphics and distributed parallel computation from the Universidade do Minho (fall 2009), after which he worked as a researcher in the same university. He joined the Institut National de Recherche en Informatique et Automatique (INRIA) and the FRVSense team as a PhD student in the fall 2010 under the supervision of Kadi Bouatouch. His research interests include spherical integration methods applied to light transport simulation.



**Christian Bouville** is presently invited researcher in the FRVSense team at IRISA in Rennes, France. He has been team leader, project leader and Emeritus expert at Orange Labs until 2006 and has been involved in many European and national projects. His main fields of research are now global illumination models and image-based rendering with a special interest in machine learning approaches.



**Mickaël Ribardière** received the master's degree in computer science from the University of Limoges, in 2007 and the PhD degree from the University of Rennes 1 in 2010. He is currently a research scientist in the FRVSense team at the Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA). His research interests include global illumination, lighting simulation for complex environment, and computer vision.



**Luís Paulo Santos** is an assistant professor at the Universidade do Minho, Portugal. He published several papers on both computer graphics and parallel processing on international conferences and journals. He manages two nationally funded graphics R&D projects and participates in multiple European projects with both academia and industry. His main research interests include interactive global illumination and parallel processing. He is a member of the Direction Board of the Portuguese chapter of Eurographics since 2008, was deputy director of the University of Minho's Department of Informatics from 2010 to 2012 and director of the doctoral program on informatics over the same period.



**Kadi Bouatouch** received the PhD degree in 1977 and the higher doctorate degree in computer science in the field of computer graphics in 1989. She was an electronics and automatic systems engineer (ENSEM 1974). She is working on global illumination, lighting simulation for complex environments, GPU-based rendering and computer vision. She is currently a professor at the University of Rennes 1, France and researcher at the Institut de Recherche en Informatique et Systèmes Aléatoires. She is member of Eurographics.

▷ For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).

# Wait-Free Shared-Memory Irradiance Caching

**Kurt Debattista and Piotr Dubla** ■ University of Warwick

**Luís Paulo Peixoto dos Santos** ■ Universidade do Minho

**Alan Chalmers** ■ University of Warwick

**D**esktop multicore computing calls for modifying traditional rendering algorithms to parallelize the available resources and maximize their use. For certain algorithms, such as classic ray tracing, this conversion can be relatively straightforward, but computing more complex lighting conditions requires careful consideration of thread synchronization to minimize overhead and allow computation.<sup>1</sup>

---

**Parallelizing rendering algorithms to exploit multiprocessor and multicore machines isn't straightforward. For example, the irradiance cache (IC) is an acceleration data structure that caches indirect diffuse irradiance values. In multicore systems, threads must share the IC to achieve high efficiency. A novel wait-free access mechanism significantly reduces synchronization overhead.**

We've developed a parallel solution to one method used for computing complex lighting conditions—namely, the irradiance cache (IC), an acceleration data structure that caches sparsely sampled values for indirect diffuse irradiance in the framework of a distributed ray-tracing algorithm.<sup>2</sup> However, because all rendering threads can write to and read from the IC, a multi-threaded shared-memory system must have a data-access-control mechanism to ensure that the data structure isn't corrupted.

Such control mechanisms incur their own overhead, so they must be carefully designed to not compromise performance. Traditionally, data-access-control mechanisms use lock-based mutual exclusion. However, nonblocking data structures that take the form of obstruction-free, lock-free, or wait-free data structures offer considerable performance advantages. Wait-free structures are the most powerful (see

the “Nonblocking Synchronization” sidebar), but system developers and researchers have considered them difficult to construct, and they're relatively rare in practice.<sup>3</sup>

Here, we present an efficient wait-free algorithm that lets all threads concurrently access an unbounded shared IC, without using locks or critical sections. This is an extension of our previous research,<sup>4</sup> which presented an initial version of the wait-free algorithm and tested it on an eight-core machine. The earlier algorithm could handle only fixed-size arrays in the IC. Furthermore, it discarded some data when a conflict among threads occurred. We've fixed these limitations to guarantee the successful insertion of all new irradiance values. We've also assessed this algorithm's efficiency on two highly concurrent multicore systems with up to 24 physical cores. The results demonstrate its superior performance and scalability over two other more traditional and straightforward mechanisms for sharing the IC among a shared-memory system's threads. (An audiovisual presentation that augments this article's description of the algorithm is available at <http://doi.ieeecomputersociety.org/10.1109/MCG.2010.80>.)

## Related Work in Parallel IC

Recent ray-tracing improvements have enabled interactive computation of many global effects, such as specular phenomena and correct shadows.<sup>5</sup> However, the dense sampling at each shading point required by indirect diffuse interreflections dramatically increases rendering times. Gregory Ward and his colleagues exploited a continuous smooth

# Nonblocking Synchronization

Using shared-memory multithreading to execute parallel computations requires careful design of the access to data structures that all threads can access concurrently. A data-access-control mechanism must be able to ensure that the data structure isn't corrupted.

Traditionally, access control to shared-memory data structures is maintained via mutual exclusion. When critical sections are reasonably large, threads can use blocking mechanisms to preempt the running thread. However, when frequent access to a shared data structure is required, blocking costs can be prohibitive. In such cases, a thread would typically enter a busy-wait state, usually using a spin lock, when another thread is in the critical section and maintain that state until the other thread completes. Such control mechanisms incur overheads, such as serialization of accesses to the shared data structure. Blocking entails expensive context switches, and busy-waiting of frequently accessed resources leads to contention that can drastically reduce performance as the number of threads increases.<sup>1</sup>

Alternatives that avoid mutual exclusion exist in the form of nonblocking synchronization. By carefully ordering instructions, a system developer can remove all critical sections and thereby reduce contention by allowing nonblocking algorithms to guarantee that none of the code is serialized.<sup>2</sup>

The weakest form of nonblocking data structures is *obstruction-free methods*, which guarantee that a thread can complete in finite time if it operates in isolation. When nonblocking data structures can guarantee that at least one among a set of concurrent threads will complete in finite time, they're said to be *lock free*. All lock-free algorithms are obstruction free. However, lock-free and obstruction-free methods rely on retrials and can't guarantee an upper bound on the number of executed instructions.

When an algorithm can guarantee that all threads will complete in finite time, the algorithm is said to be *wait free*. Wait-free algorithms can guarantee an upper bound on the number of instructions, thus avoiding starvation, deadlock, livelock, and priority-inversion problems. Wait-free algorithms are ideal for multiprogrammed multiprocessors—for example, when a thread holding a lock is preempted and causes all other threads to busy-wait uselessly. Clearly, all wait-free data structures are also lock free.

The construction of nonblocking algorithms requires powerful atomic primitives, which execute without interruption as a single instruction on modern architectures. We can view these algorithms as a limiting case that re-

```
1 atomic XADD(address location)
2 {
3     int value = *location;
4     *location = value + 1;
5     return value
6 }
```

Figure A. Pseudocode for the fetch and add (XADD) atomic instruction. XADD atomically adds 1 to a value and returns the previous value. This function would be implemented as a single instruction on modern processor architectures.

```
1 atomic CAS(address location, value cmpVal, value newVal)
2 {
3     if(*location == cmpVal) {
4         *location = newVal;
5         return true;
6     } else return false;
7 }
```

Figure B. Pseudocode for the compare and swap (CAS) atomic instruction. CAS compares the value at location with cmpVal and, if they are the same, updates the value pointed to by location to newVal. This function would be implemented as a single instruction on modern processor architectures.

duces the size of critical sections to the size of individual machine instructions.

Figures A and B show pseudocode for two atomic instructions, fetch and add (XADD) and compare and swap (CAS), that we use for our wait-free irradiance cache. Maurice Herlihy provided a hierarchy of such primitives' effectiveness.<sup>3</sup> The most effective are those that can be used to implement any wait-free data structure, which Herlihy described as CAS (or load-link store-conditional instruction pairs, which are an alternative to the CAS in some architectures).

## References

1. T.E. Anderson, E.D. Lazowska, and H.M. Levy, "The Performance Implications of Thread Management Alternatives for Shared-Memory Multiprocessors," *IEEE Trans. Computers*, vol. 38, no. 12, 1989, pp. 1631–1644.
2. M. Herlihy and N. Shavit, *The Art of Multiprocessor Programming*, Morgan Kaufmann, 2008.
3. M. Herlihy, "Wait-Free Synchronization," *ACM Trans. Programming Languages and Systems*, vol. 13, no. 1, 1991, pp. 124–149.

function that generally characterizes the indirect diffuse component over space, unaffected by the high-frequency changes common with the specular component.<sup>2</sup> They proposed the IC to allow sparse evaluation of indirect diffuse irradiance, storing the sparsely calculated values in the IC and

reusing them to extrapolate or interpolate values at nearby locations. By caching indirect diffuse irradiance samples in the framework of a distributed ray-tracing algorithm,<sup>2</sup> the IC allows irradiance values to be interpolated for regions in a given sample's neighborhood. This reduces rendering

## Irradiance Caching

Physically based computation of the radiance reflected at a point  $p$  along a direction  $\Theta$  is dictated by the rendering equation:<sup>1</sup>

$$L_r(p \rightarrow \Theta) = \int_{\Omega} f_r(p, \Theta \leftrightarrow \Psi) L_i(p \leftarrow \Psi) \cos(\vec{N}_p, \Psi) d\Omega,$$

where  $f_r(p, \Theta \leftrightarrow \Psi)$  is the bidirectional reflectance distribution function at  $p$  for directions  $\Theta$  and  $\Psi$ ,  $L_i(p \leftarrow \Psi)$  is the incident radiance at  $p$  along  $\Psi$ ,  $\vec{N}_p$  is the surface normal at  $p$ , and  $\Omega$ , the integration domain, is the hemisphere centered at  $p$  and oriented around  $\vec{N}_p$ .

Ray tracing approximates  $L_r(p \rightarrow \Theta)$  by shooting rays along a number of directions  $\Psi$ , thus sampling  $L_i(p \leftarrow \Psi)$  for these directions. We can usually compute certain light transport phenomena, such as specular scattering and direct illumination, with a limited number of rays. Other phenomena, because they lack directionality, require *hemisphere sampling*—that is, stochastically selecting and shooting many rays across  $\Omega$ . This stochastic integration method, called Monte Carlo integration, usually accounts for much of the computation.

One such phenomenon is *indirect diffuse reflection*, the diffusely reflected radiance at  $p$  along a given direction re-

sulting from the indirect irradiance  $E(p)$ .  $E(p)$  is the indirect incident radiant flux per unit area at  $p$ . Accurately computing  $E(p)$  requires densely sampling  $\Omega$ . In a ray-tracing context, this requires shooting hundreds or thousands of rays distributed across the hemisphere while, simultaneously, carefully avoiding directions corresponding to light sources so that we include only indirect lighting—that is, light that's been reflected by at least one object.

Indirect diffuse reflections are crucial to convey a perception of realism (see Figure C), but sampling the hemisphere at all shading points results in very long rendering times, deemed unacceptable even for most offline renderings. Gregory Ward and his colleagues realized in 1988 that indirect diffuse reflection is generally a continuous smooth function over space, not affected by the high-frequency changes common with specular reflections.<sup>2</sup> They proposed accelerating the computation of indirect diffuse reflections by densely sampling the hemisphere at only a sparse set of shading points and interpolating the remaining ones. They store the sparsely calculated, indirect irradiance values in the irradiance cache (IC) data structure and later reuse them to extrapolate or interpolate irradiance values at nearby locations.



Figure C. The contribution of indirect lighting: (1) direct only, (2) indirect only, and (3) full. Without the computation of the indirect lighting, the rendered images are largely inaccurate.

time by exploiting spatial coherence. (The “Irradiance Caching” sidebar describes the motivation and mechanism behind this data structure.)

To accelerate range searches for locating valid samples in the IC, the algorithm builds an octree incrementally every time a new sample is added. Writing to the cache requires both storing the new indirect diffuse irradiance value and updating the octree. In parallel systems, each rendering process, or thread, might evaluate new indirect diffuse irradiance values and add them to the IC. To increase efficiency, all processes must share the IC, thus avoiding replicated work in which one process evaluates an irradiance value that other processes have already calculated. Ideally, the IC becomes a shared data structure, requiring some sharing

mechanism to ensure that all processes can access the available data, that the data isn't corrupted, and that overheads don't compromise efficiency.

In distributed-memory systems, such as workstation clusters, each node has its own address space, resulting in multiple copies of the shared data structure that are regularly synchronized. The standard radiance distribution<sup>6</sup> supports a parallel renderer over a distributed system using Sun's Network File System for concurrent IC access. This approach has led to contention and can result in poor performance when using inefficient file lock managers. Roland Kohlhaas and his colleagues broadcast IC values among processors after every 50 samples calculated at each slave.<sup>7</sup> David Robertson and his colleagues presented a centralized par-

We can interpolate  $E(p)$  from a set  $S(p)$  of previously evaluated irradiance values  $E(p_i)$  at points  $p_i$ , by using a weighted average:

$$E(p) = \frac{\sum_{i \in S(p)} w_i(p) E(p_i)}{\sum_{i \in S(p)} w_i(p)},$$

where the weights

$$w_i(p) = \left( \frac{\|p - p_i\|}{R_i} + \sqrt{1 - \vec{N}_p \cdot \vec{N}_{p_i}} \right)^{-1}$$

depend on the distance between  $p$  and  $p_i$ , on the harmonic mean distance ( $R_i$ ) to objects visible from  $p_i$ , and on the relative orientation of the normals at  $p$  and  $p_i$ . We can determine  $S(p)$  by requiring  $w_i(p)$  to be larger than the reciprocal of the maximum acceptable error,  $a$ , which is a user-supplied parameter:  $S(p) = \{i : w_i(p) > 1/a\}$ .

When the renderer requires indirect irradiance at any point  $p$ , it first determines  $S(p)$  by querying the IC. If  $S(p)$  is empty,  $E(p)$  is evaluated by the Monte Carlo integration; otherwise, it's interpolated from the  $E(p_i)$  belonging to  $S(p)$ . Querying the IC to determine  $S(p)$  amounts to locating all samples  $p_i$  stored in the cache that meet the search criterion. This range search task is computationally demanding, but we can optimize it by resorting to 3D hierarchical data structures, such as octrees or  $k$ -d trees, and spatially ordering the IC.

By exploiting spatial coherence, the IC offers an order-of-magnitude improvement in rendering time over Monte Carlo integration. We can further improve performance when rendering animations of static scenes because the indirect diffuse irradiance remains constant, which allows reuse of the IC samples across frames.

Researchers have recently extended the IC as a stand-alone algorithm in many guises—for example, as an accel-

eration data structure for rendering glossy surfaces by storing radiance,<sup>3</sup> as a participating-media phenomenon,<sup>4</sup> for translucency,<sup>5</sup> or in conjunction with photon mapping.<sup>6</sup> Other extensions have exploited coherence in the temporal domain.<sup>7–9</sup> PDI/DreamWorks has also used similar methods to accelerate rendering.<sup>10</sup>

## References

1. J.T. Kajiya “The Rendering Equation,” *Proc. Siggraph*, ACM Press, 1986, pp. 143–150.
2. G. Ward, “A Ray Tracing Solution for Diffuse Interreflection,” *Proc. Siggraph*, ACM Press, 1988, pp. 85–92.
3. J. Krivanek et al., “Radiance Caching for Efficient Global Illumination Computation,” *IEEE Trans. Visualization and Computer Graphics*, vol. 11, no. 5, 2005, pp. 550–561.
4. D. Jarosz and J. Zwicker, “Radiance Caching for Participating Media,” *ACM Trans. Computer Graphics*, vol. 27, no. 1, 2008, article 56.
5. S.-L. Keng, W.-Y. Lee, and J.-H. Chuang, “An Efficient Caching-Based Rendering of Translucent Materials,” *The Visual Computer*, vol. 23, no. 1, 2006, pp. 59–69.
6. H.W. Jensen, *Realistic Image Synthesis Using Photon Mapping*, A K Peters, 2001.
7. M. Smyk et al., “Temporally Coherent Irradiance Caching for High Quality Animation Rendering,” *Computer Graphics Forum*, vol. 24, no. 3, 2005, pp. 401–412.
8. P. Gautron, K. Bouatouch, and S. Pattanaik, “Temporal Radiance Caching,” *IEEE Trans. Visualization and Computer Graphics*, vol. 13, no. 5, 2007, pp. 891–901.
9. K. Debattista et al., “Instant Caching for Interactive Global Illumination,” *Computer Graphics Forum*, vol. 28, no. 8, 2009, pp. 2216–2228.
10. E. Tabellion and A. Lamorlette, “An Approximate Global Illumination System for Computer Generated Films,” *ACM Trans. Graphics*, vol. 23, no. 3, 2004, pp. 469–476.

allel radiance version that sends the calculated IC values to a master process whenever a threshold is met.<sup>8</sup> Each slave then collects the values deposited at the master by the other slaves. Another proposal restricts diffuse irradiance evaluations to a subset of the available processors, synchronizing the IC among these at a higher frequency than with the remaining processors.<sup>9</sup>

We're unaware of any publication describing a data-access-control mechanism for sharing the IC among rendering threads in a shared-memory parallel system, other than our previous wait-free algorithm. The algorithm we propose here supports extendable memory for inserting an unbounded number of IC samples and for successfully inserting all new irradiance values.

## Data-Access-Control Algorithms

We now present the algorithms for the three data-access-control mechanisms we evaluated in experiments.

We begin with a traditional single-threaded IC that has no access control (see Figure 1). The `IrradianceCache` data structure represents the IC; it consists of an octree of recursive nodes. The individual node is called `ICNode`. Each `ICNode` contains pointers to another eight nodes and an `ICList` storing the list of IC samples. Figure 2 shows the `ICNode` for the wait-free method. For the other methods, the `ICList` is just a single dynamic array.

### **The Lock-Based Irradiance Cache**

The lock-based access-control algorithm (LCK)

```

1 IrradianceCache IC;
2
3 ComputeIndirectDiffuse() {
4     //get irradiance from IC if there are valid records
5     inIC = IC.getIrradiance ();
6     if (!inIC) { // no valid records found
7         // compute it by sampling the hemisphere
8         ICsample = ComputeIrradianceRT ();
9         // insert new IC sample into the octree
10        IC.insert (ICsample);
11    }
12 }
13
14 IrradianceCache::getIrradiance(Irr) {
15     Irr = {0,0,0};
16     <Traverse the octree>
17     <verify validity of sample>
18     <extrapolate irradiance; add to Irr>
19     if (found) return true;
20     else return false;
21 }
22
23 IrradianceCache::insert (ICsample) {
24     // recursively traverse the octree
25     // starting at root
26     IC.root.insert (ICsample);
27 }
28
29 ICNode::insert (ICSample) {
30     if (correct insertion node) {
31         IClist.Add (ICsample);
32     } else {
33         // go deeper in the octree
34         xyz = EvaluateOctant();
35         if (children[xyz] == NULL)
36             children[xyz] = new ICNode ();
37         children[xyz].insert (ICsample);
38     }
39 }
40
41 ICLList::Add (ICsample) {
42     // insert new record in head of list
43     IClist.records[head++] = ICsample;
44 }

```

Figure 1. A traditional sequential irradiance cache (IC). This approach has no access control. It consists of an octree of recursive nodes.

locks the IC whenever a read or write is made to it (see Figure 3, lines 4–6 and 12–14). However, the code responsible for hemisphere sampling, ComputeIrradianceRT(), isn't a critical re-

gion, so for this computation the method allows concurrent irradiance evaluation. The LCK's major disadvantage is that it serializes all accesses—both reads and writes—to the shared IC. As the number of threads increases, contention will increase, preventing performance from scaling with the degree of parallelism.

### The Local-Write Irradiance Cache

An alternative approach is to have a global IC readable by all threads and an additional local IC per thread (see Figure 4). Each thread writes only to its local IC but reads from both. At certain predefined execution points, such as the end of a frame, the local ICs sequentially merge into the global IC. This synchronization uses a frame end as a barrier, effectively constituting a blocking approach to synchronization.

This approach's major drawback is that it disallows any sharing in a single frame, thus resulting in work replication. The LW algorithm has a much higher IC sample count than the other two approaches we evaluated because each thread must locally evaluate all irradiance values required by its assigned image tiles. Additionally, memory consumption is dictated by the number of threads used and the octree's complexity.

### The Wait-Free Irradiance Cache

The wait-free (WF) algorithm doesn't rely on any critical sections to both read and write to the shared IC. The algorithm changes three methods from the traditional IC.

The first method is the ICLList::Add function (see Figure 5). The insert onto the node itself takes the form of an insertion onto an array or unbounded queue. For completeness, we demonstrate how the method works for an unbounded queue. Insertion onto a fixed-size array is just a specialized case of this algorithm. The fixed-size array version is the same as the enqueue function of the Herlihy-Wing concurrent queue.<sup>10</sup>

The structure of ICLList is an unbounded queue

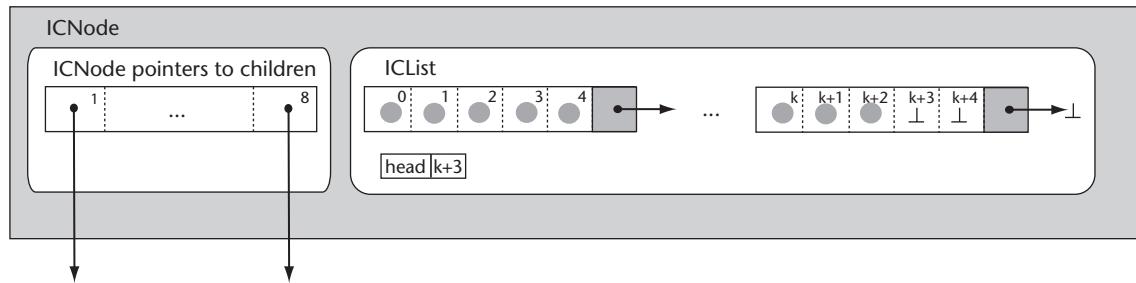


Figure 2. An ICNode structure for wait-free access control. For the other methods, ICLList is just a single dynamic array that's extended whenever required.

of arrays, used to maintain coherence and ensure that queue extensions aren't frequent (see Figure 2). In Figure 5, the array type is qNode, which contains an array of qNodeSize elements and a pointer to another qNode. Initially, the queue of arrays contains only one qNode; when the insertion requires a new qNode, the executing thread creates it and attaches it to the previous one. An array is always initialized with a list of NULL pointers (or some other symbol that the computation doesn't use) to denote that none of the threads has yet added an ICsample. When adding samples to an IC node, the algorithm uses the atomic fetch-and-add operator (XADD; see Figure 5, line 3). This returns a unique index into the list of records, ensuring that samples are never overwritten. Simultaneously, the thread increments the index to the next free position.

When the structure must be extended (see Figure 5, line 13), the algorithm creates a new qNode and uses a compare-and-swap instruction (CAS) to insert it onto the previous qNode (see Figure 5, line 19). If another thread hasn't yet extended the queue—that is, if the pointer is still NULL, the CAS completes successfully and the executing thread inserts the associated ICsample onto the structure (see Figure 5, line 26). However, if another thread extended the queue, the CAS will fail and this thread will discard the created qNode (see Figure 5, line 20). The thread will then insert the associated sample onto the qNode that some other thread created (otherwise, the CAS would have succeeded).

Figure 6 demonstrates three concurrent threads—R, G, and B—executing this method for a qNodeSize of five. This will help illustrate how ICList::Add() functions. In Figure 6a, the ICList is completely empty. In Figure 6b, R has just incremented the head but hasn't yet inserted the sample. R inserts the sample in Figure 6c. In Figure 6d, both B and G have just incremented the head but not inserted the samples. In Figure 6e, B hasn't inserted its sample and R has inserted another sample, but G has yet to insert the sample and is still on the same line of code as in Figure 6d.

Figure 6f demonstrates the scenario of the list needing to be extended and the possible conflicts that might occur. R has just filled in the first qNode, and G and B are about to insert another two samples. They have, in fact, already incremented head. Because both G and B have checked that the last qNode is full and has no successor (see Figure 5, line 16), both created a new qNode. However, the CAS at line 19 in Figure 5 means that only one will succeed in attaching it

```

1 ComputeIndirectDiffuse()
2 {
3     //get irradiance from IC if there are valid
4     //records
5     IC.lock();
6     inIC = IC.getIrradiance (Irr);
7     IC.unlock();
8
9     if (!inIC) { // no valid records found
10        // compute it by sampling the hemisphere
11        ICsample = ComputeIrradianceRT ();
12        // insert new IC sample into the octree
13        IC.lock();
14        IC.insert (ICsample);
15        IC.unlock();
16    }

```

**Figure 3.** A lock-based IC (LCK). This method uses busy-waiting for reading and extending the IC.

```

1 IrradianceCache IClocal[number threads],
2           ICglobal;
3
4 ComputeIndirectDiffuse()
5 {
6     //get irradiance from IC if there are valid
7     //records
8     inIC = ICglobal.getIrradiance (Irr);
9
10    if (!inIC)
11        inIC = IClocal[current thread].getIrradiance ();
12
13    if (!inIC) { // no valid records found
14        // compute it by sampling the hemisphere
15        ICsample = ComputeIrradianceRT ();
16        // insert new sample into the local cache
17        IClocal[current thread].insert (ICsample);
18    }
19
20 }

```

**Figure 4.** A local-write IC (LW). This method maintains a separate IC per thread and synchronizes them after a single frame is rendered.

to the previous qNode. In this case, at Figure 6g, we can see that G has succeeded and B is deleting the qNode it created. G has inserted the sample onto the new qNode. At Figure 6h, B inserts the sample onto the qNode that G had created.

The second method that's changed from the traditional IC is the insert onto the octree structure (see Figure 7). When adding a new child node to the octree, the executing thread builds the new node using a temporary pointer. Once built, the node is attached to the octree using the CAS operator (see Figure 7, line 10). The reasoning underlying this method is similar to that for ICList::Add. Either this thread creates the subtree or some other thread does; computation proceeds notwithstanding.

The final modified method is IrradianceCache::getIrradiance() (see Figure 8). The modifications simply reflect the structure

```

1 ICLList::Add (ICsample) {
2     // get index of new sample in node list
3     int index = XADD (&head);
4     int iteration = index / qNodeSize;
5     int pos = index % qNodeSize;
6     qNode * tail;
7     int count = 0;
8
9     // identify node - can be optimized with a local tail
10    for (tail = qHead; tail->next != NULL && count < iteration;
11        tail = tail->next, count++);
12
13    if (iteration > count){
14        for (int n = 0; n < iteration - count; n++) {
15            // this is where we add the new Array
16            if (tail->next == NULL) {
17                // all entries are initialized as NULL
18                qNode * newN = new qNode;
19                if(!CAS(&tail->next, NULL, newN))
20                    delete newN;
21            }
22            tail = tail->next;
23        }
24        // if this thread did not update
25        // some other thread must have updated
26        tail->records[pos] = ICsample;
27    }
28 }

```

Figure 5. The wait-free IC Add method. This wait-free method ensures that IC samples can be added concurrently by different threads without blocking or busy-waiting.

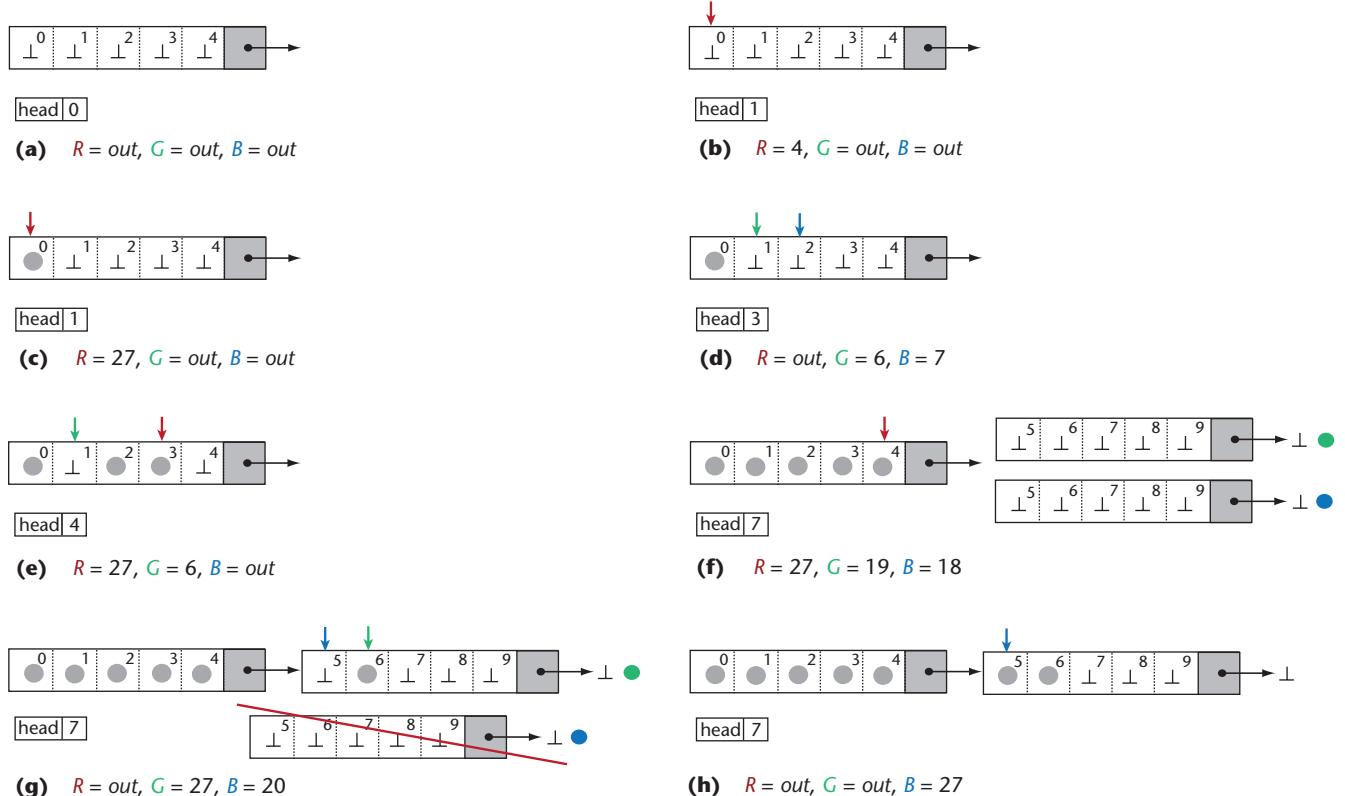


Figure 6. How three threads ( $R$ ,  $G$ , and  $B$ ) would concurrently add samples to a node using our novel wait-free method. Numbers refer to the thread's location in Figure 5.  $\text{out}$  means that the thread isn't executing this function.  $\perp$  represents the NULL pointer.

```

1 ICNode::insert (ICSample) {
2     if (correct insertion node)
3         IClist.Add (ICsample);
4     else { // go deeper in the octree
5
6         xyz = EvaluateOctant();
7         if (children[xyz]==NULL) {
8             temp = new ICNode();
9             // Update new branch into the octree
10            if (!CAS (children[xyz], NULL, temp))
11                free temp;
12        }
13        // irrelevant to whether this thread created the subtree
14        // someone must have created anyway
15        // recurse the insertion of ICsample onto the subtree
16        children[xyz].insert (ICsample);
17    }
18 }

```

**Figure 7.** The wait-free IC `insert` method. This method ensures that the octree grows dynamically without the use of busy-waiting or blocking.

```

1 Irradiance Cache::getIrradiance(Irr) {
2     Irr = {0,0,0};
3     <Traverse the octree>
4     for (qNode * tNode = qHead; tNode != NULL; tNode = tNode ->next) {
5         for (i = 0; i < qNodeSize; i++)
6             if (tNode ->records[i] != NULL) {
7                 <verify validity of sample>
8                 <extrapolate irradiance; add to Irr>
9             }
10    }
11    if (found) return true;
12    else return false;
13 }

```

**Figure 8.** The wait-free IC `getIrradiance` method. The method takes into account the dynamic nature of the `ICList` construction in the wait-free IC.

changes to `ICList`. The method uses the fact of `qNode` elements being initialized to `NULL`. This function queries all elements that aren't `NULL` and uses them to calculate the irradiance, if the valid neighborhood criterion is satisfied.

The wait-free approach ensures that all threads can access the single shared IC concurrently. Our experimental results show increased execution time both when interpolating and creating IC samples, without suffering the LW approach's larger memory requirements.

## Experimental Results

We obtained all the results presented here on two shared-memory systems.

One system was a dual-quad-core machine based on the Intel Xeon E5520 (Nehalem architecture), running at 2.26 GHz with 12 Gbytes of RAM. These processors include the Intel QuickPath Interconnect (which replaces the legacy front-side bus). They also support hyperthreading, enabling two threads per core and thus reporting a total of 16

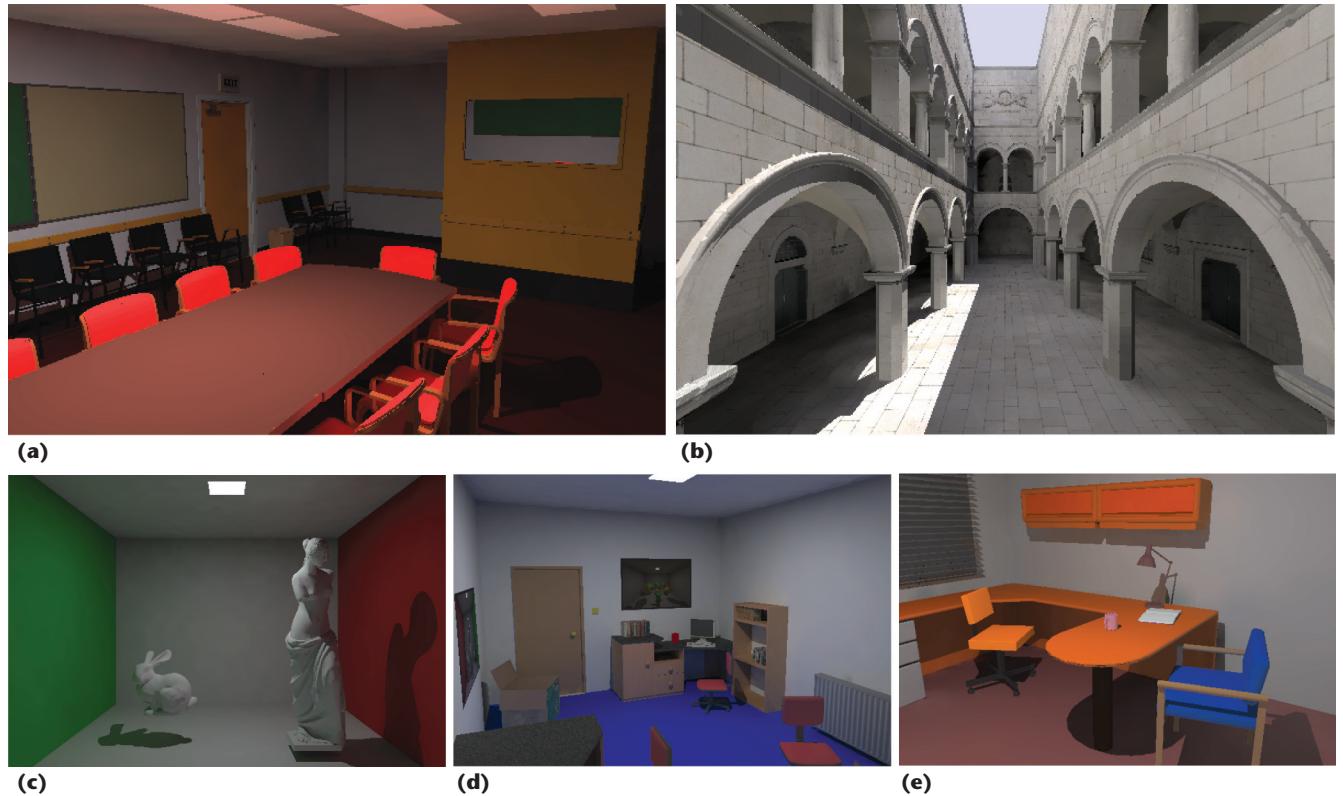
logical processors to the operating system. Hyperthreading replicates certain processor resources but not the main execution units. Intel claims up to 30 percent speed improvement, compared to otherwise identical, nonhyperthreaded processors.<sup>11</sup>

The second system was a quad hexacore machine based on the Intel Xeon E7450 (Dunnington architecture), running at 2.40 GHz with 64 Gbytes of RAM. With 24 physical cores, this system let us evaluate the scalability of our wait-free access-control mechanism.

Both systems ran CentOS 5.2 with the Intel Compiler Suite Professional v. 11.0.

For all experiments, we used our own interactive ray tracer, which doesn't employ packetization or explicit SIMD (single instruction, multiple data) operations. The only exception is the ray-bounding volume-intersection test used to traverse the acceleration data structure, which is a bounding-volume-hierarchy implementation based on Ingo Wald and his colleagues' research.<sup>12</sup>

Figure 9 shows the five scenes we used. We



**Figure 9.** The five scenes used in the experiments: (a) conference room (190k polygons), (b) Sponza (66k polygons), (c) Cornell (48k polygons), (d) desk (12k polygons), and (e) office (20k polygons). The scene provide a range of geometric complexity, physical dimensions, and lighting conditions, all rendered at  $640 \times 480$  resolution.

picked them to provide a range of geometric complexity, physical dimensions, and lighting conditions. We rendered all scenes at  $640 \times 480$  resolution. We use the following labels for the methods: traditional sequential (TRA), lock based (LCK), local write (LW), and wait free (WF). Tables detailing the results for still images and animation are at <http://doi.ieeecomputersociety.org/10.1109/MCG.2010.80>.

### Still Images

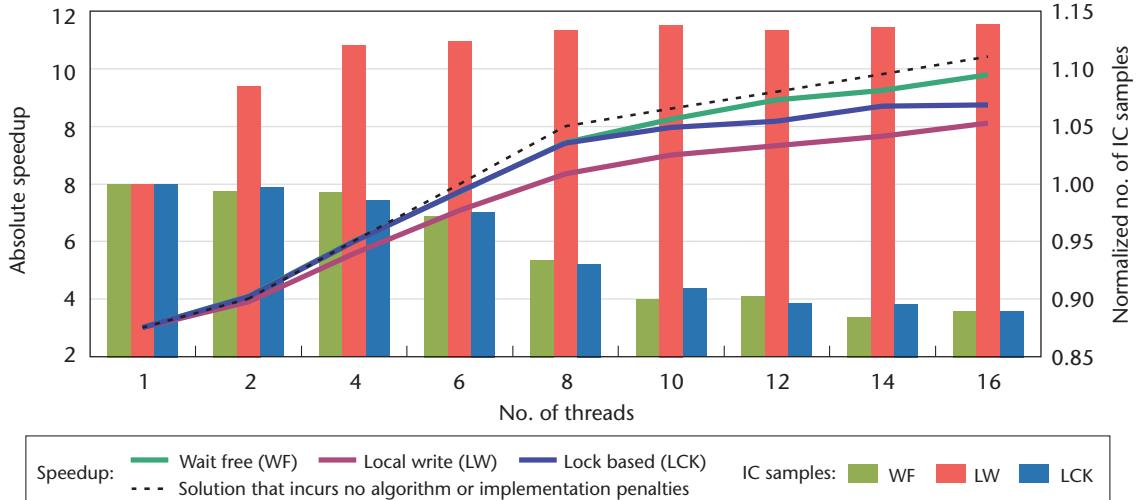
We varied the number of threads, and thus the number of used cores, up to 16 for the Nehalem architecture and 24 for the Dunnington architecture. We obtained the results for a single thread using TRA, with no data-access-control, and computed the speedup for the different techniques with respect to the sequential timings. Each image was calculated with an empty IC to show a worst-case scenario with maximal irradiance calculations.

Figures 10 and 11 present the speedup, normalized number of evaluated IC samples, and efficiency for both architectures. Each metric is the average over the five scenes. We normalized the metrics with respect to the results obtained for the same scene with one single thread and the traditional approach to the IC (no data-access-control). Absolute values particular to each scene are therefore irrelevant, as long as the behavior was similar

across the different scenes for each access-control mechanism. We measured worst-case standard deviations of 14.2 and 4.2 percent for, respectively, absolute speedup and the normalized number of generated IC samples. The low worst-case values indicate that we can use these averages as reliable statistics to analyze our results.

For absolute speedup, Figures 10a and 11a include a dashed line depicting the linear speedup possible if the parallel solution incurred no algorithmic or implementation penalties. For the Nehalem architecture with more than eight threads, linear speedup increased only 30 percent with each additional logical core, corresponding to Intel's claim about hyperthreading's maximum speed improvement.<sup>11</sup>

For all experiments with the Nehalem system and up to 14 threads in the Dunnington case, LW performed and scaled worse than the two other algorithms. This is because only one frame was rendered and the local caches merged only at the end of the frame, so no sharing actually occurred. Each thread had to evaluate all irradiance samples that projected into its assigned tiles of the image plane, leading to work replication. This is evident by the number of evaluated irradiance samples (see Figures 10a and 11a), which increased dramatically with the level of concurrency.



(a)

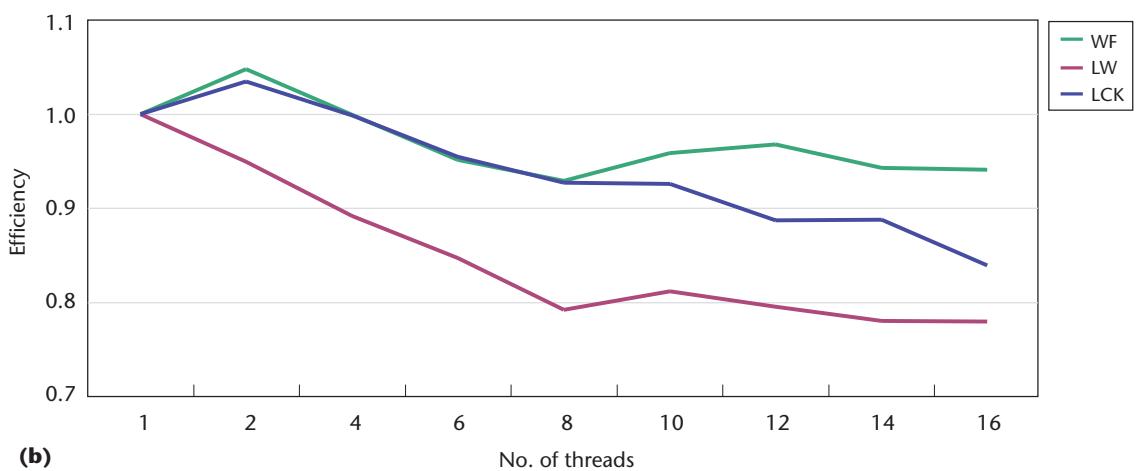


Figure 10. Performance for still images on the Nehalem architecture: (a) speedup and the normalized number of IC samples and (b) efficiency. (All values are averaged over the five scenes used in the experiments.) The wait-free IC presents an efficiency above 94 percent up to 16 logical (hyperthreaded) cores.

The performance difference between LCK and WF became evident as the number of threads increased: time waiting for locks grew, causing a major performance loss. The wait-free algorithm scaled much better. With fewer threads, LCK performed similarly to WF because the computation spent most of the time evaluating new irradiance samples, which isn't a critical region of the code. As the number of threads increased, the rendering performed more range searches. Because LCK serialized these searches, it incurred a performance penalty. Figures 10a and 11a clearly show that LCK's performance loss isn't due to work replication. In fact, the total number of IC samples evaluated by LCK and WF decreased above a certain number of threads. Success in finding valid samples to interpolate from depended on the order in which the threads requested and evaluated samples. Concurrent rendering of multiple image plane tiles quickly filled the IC with samples that

were better distributed over object space, resulting in more successful range searches than with the sequential approach. Above a significant number of threads, LCK's serialization penalty became larger than the overhead associated with work replication, and it performed even worse than LW (see Figure 11).

Parallel algorithms seldom exhibit linear speedup because of overheads, such as load imbalance, work replication, and communication and synchronization costs. Wait-free access control can minimize the last two overheads; it exhibited almost linear speedup and, consequently, a nearly constant efficiency of 0.9 on systems up to 24 cores (see Figures 10b and 11b). With around 14 cores on the Dunnington system, LCK speedup reached an inflection point and started decreasing, showing that lock-based approaches don't scale with increasing levels of concurrency. On the other hand, WF speedup grew linearly up to 24 cores, although at

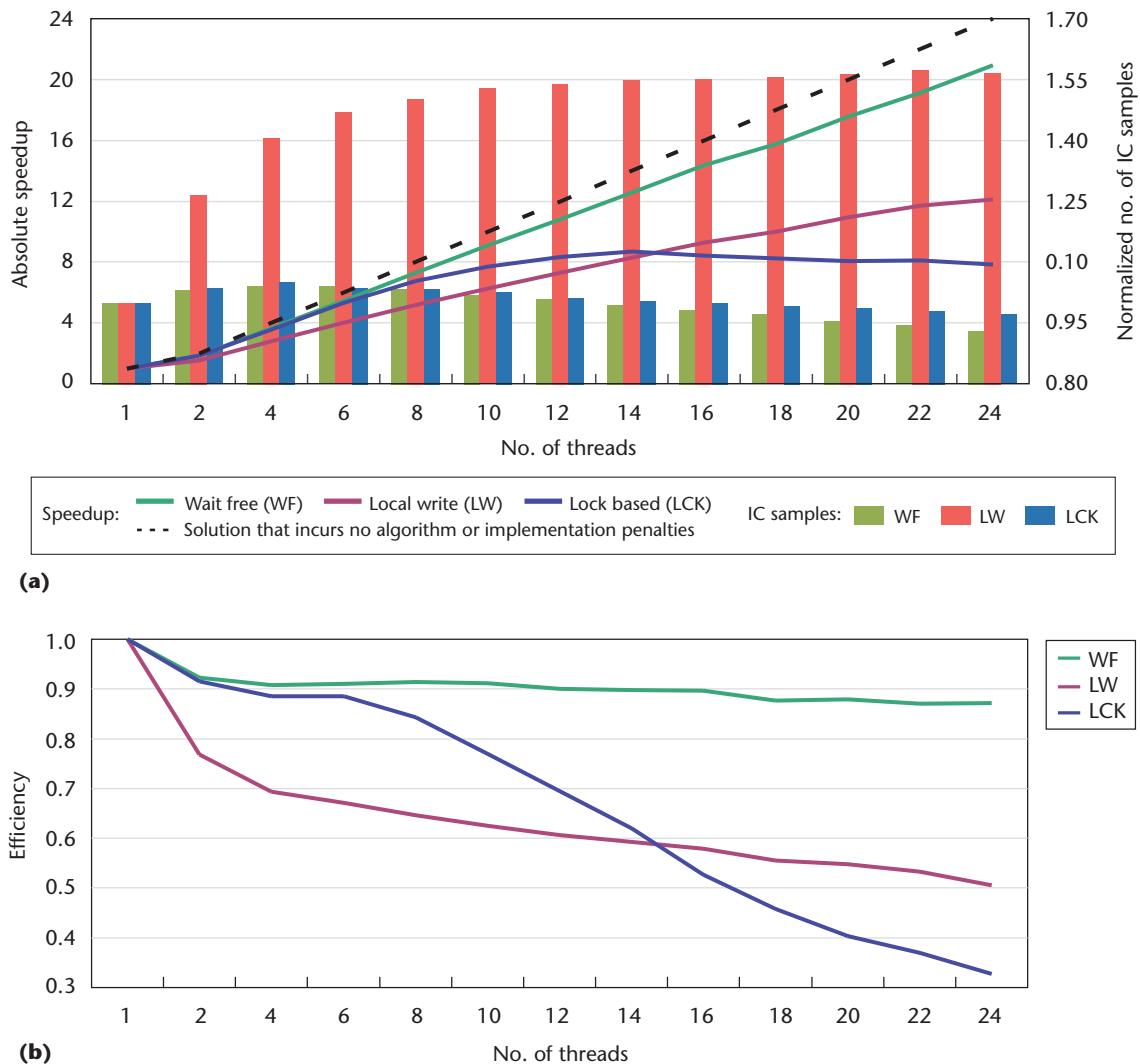


Figure 11. Performance for still images on the Dunnington architecture: (a) speedup and the normalized number of IC samples and (b) efficiency. (All values are averaged over the five scenes used in the experiments.) The wait-free IC achieves close to ideal speedup and, at 24 cores, is twice as fast as its closest rival.

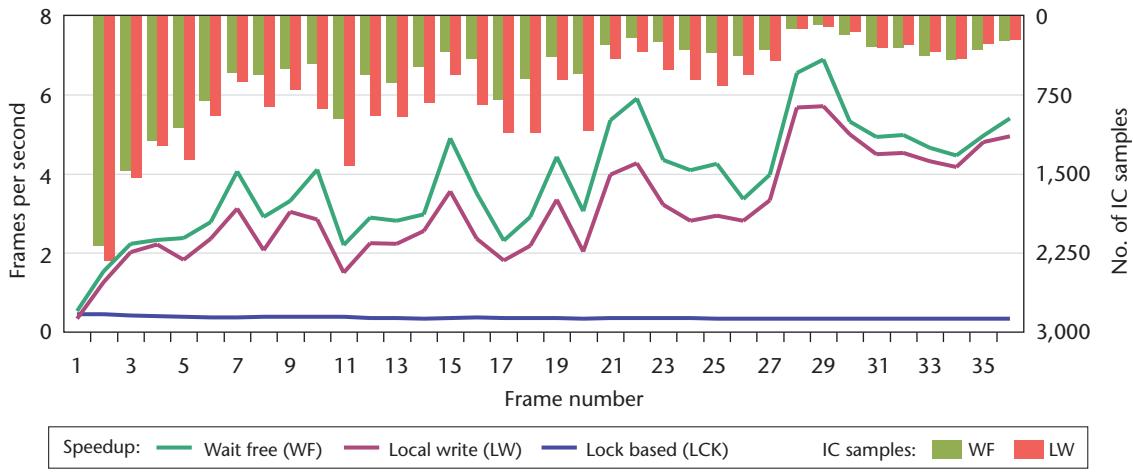
a rate slightly lower than the increase in the number of cores. The WF curve's shape suggests that any eventual inflection point is still far from being reached, which demonstrates its superior scalability potential. The shared-memory parallel ray tracer also incurs overheads such as workload distribution and results gathering, which were partially responsible for the small deviation from linear speedup with WF.

### Animations

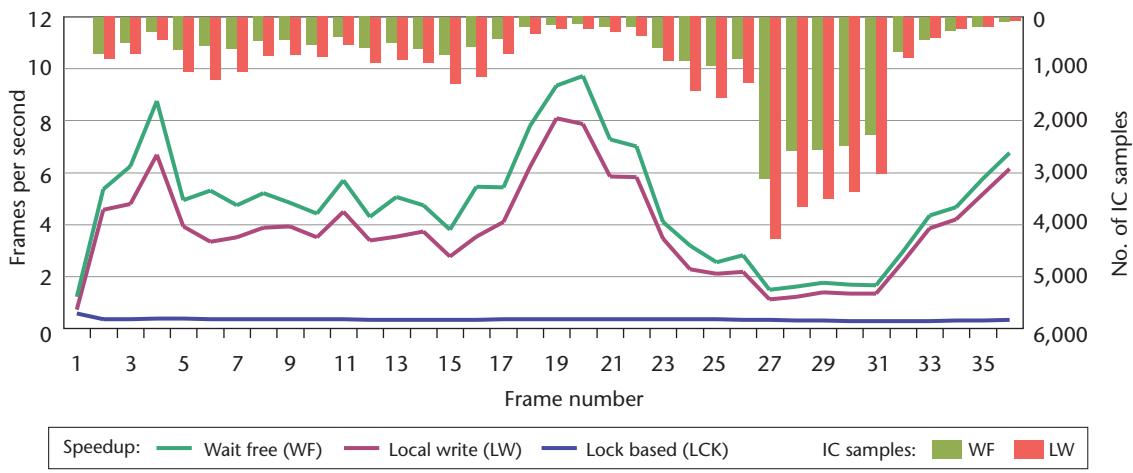
Figure 12 shows, for the Sponza and conference room scenes running on 24 cores on the Dunnington system, the frame rate and the number of IC samples evaluated per frame when running an animation of 36 frames while the camera performed a 360-degree rotation around the scene (10 degrees from frame to frame). Each frame reused previously created cache samples while si-

multaneously calculating new ones. This provides a performance overview of mixed evaluation and interpolation, unlike the case of the still images. For each scene, the first frame is the equivalent of one of the still images, where the cache is totally empty and all the samples must be generated.

Clearly, LCK performs worse than LW and WF. Because the IC won't be empty except for the first frame, the computation can reuse many irradiance samples. However, LCK serializes all range searches to locate these samples, thus severely affecting performance. LCK achieves the best rendering time for the first frame, suggesting that temporal reuse of previously calculated irradiance samples is worse than recalculating these values, which completely contradicts the rationale behind the IC.<sup>6</sup> We can thus conclude that synchronization overheads make such lock-based access mechanisms prohibitive when rendering animations of



(a)



(b)

**Figure 12.** Performance for animation on the Dunnington system with 24 cores, for the (a) Sponza and (b) conference room scenes. The wait-free IC outperforms the other methods for both scenes.

static scenes on highly concurrent shared-memory systems.

WF outperforms LW because the former shares irradiance samples immediately without any extra synchronization overhead associated with reading, whereas the latter doesn't share samples in a frame and so incurs extensive, costly evaluations of more indirect diffuse irradiance values. The bars in Figure 12 clearly show that, for WF and LW, performance variations from frame to frame correlate highly with the number of IC samples evaluated per frame. More important, these graphs show that WF's better results come from evaluating fewer irradiance samples, which is a consequence of efficient data sharing among threads.

In summary, LCK is penalized mostly by reading serialization, and LW is penalized by work replication. WF efficiently shares IC values while minimizing writing overheads and eliminating synchronization overheads associated with concurrent reads.

Multicore and multiprocessor systems now represent the standard form of desktop computing. Soon, such systems will likely have a degree of parallelism larger than what's available on current machines. So, the relevance of efficient, scalable, and reliable shared data structures for maximizing performance is continually increasing. Wait-free data structures offer an alternative to traditional locking and blocking approaches and enable traditional graphics algorithms to exploit modern hardware. The shared-memory IC we've demonstrated shows such techniques' potential. Our algorithm has let us achieve close to interactive rates for ray tracing with global illumination. We hope our solution will motivate similar parallel methods in other computer graphics areas. ■■■

### Acknowledgments

Partial funding for this work came from the Portuguese Foundation for Science and Technology's Project IGIDE

(*Interactive Global Illumination within Dynamic Environments*), grant PT-DC/EIA/65965/2006, and from the United Kingdom's Engineering and Physical Sciences Research Council, grant EP/D069874/2. Greg Ward provided the office and conference scenes from the Radiance package, and Stanford's Graphics Group provided the bunny model from the Stanford 3D Repository. Alberto Proen a granted us access to the two multicore systems.

## References

- M. Herlihy, "Technical Perspective: Highly Concurrent Data Structures," *Comm. ACM*, vol. 52, no. 5, 2009, p. 99.
- G. Ward, "A Ray Tracing Solution for Diffuse Interreflection," *Proc. Siggraph*, ACM Press, 1988, pp. 85-92.
- M. Herlihy, V. Luchangco, and M. Moir, "Obstruction-Free Synchronization: Double-Ended Queues as an Example," *Proc. 23rd Int'l Conf. Distributed Computing Systems (ICDCS 03)*, IEEE CS Press, 2003, pp. 522-529.
- P. Dubla et al., "Wait-Free Shared-Memory Irradiance Cache," *Proc. Eurographics Symp. Parallel Graphics and Visualization (ESPGV 09)*, Eurographics Assoc., 2009, pp. 57-64.
- I. Wald et al., "State of the Art in Ray Tracing Animated Scenes," *Eurographics 2007 State of the Art Reports*, Eurographics Assoc., 2007, pp. 89-116.
- G. Ward, "The Radiance Lighting Simulation and Rendering System," *Proc. Siggraph*, ACM Press, 1994, pp. 459-472.
- R. Koholka, H. Mayer, and A. Goller, "MPI-Parallelized Radiance on SGI CoW and SMP," *Proc. 4th Int'l ACPC Conf. (ParNum 99)*, Springer, 1999, pp. 549-558.
- D. Robertson et al., "Parallelization of Radiance for Real Time Interactive Lighting Visualization Walk-
- throughs," *Proc. 1999 ACM/IEEE Conf. Supercomputing*, ACM Press, 1999, article 61.
- K. Debattista, L.P. Santos, and A. Chalmers, "Accelerating the Irradiance Cache through Parallel Component-Based Rendering," *Proc. Eurographics Symp. Parallel Graphics and Visualization*, Eurographics Assoc., 2006, pp. 27-34.
- M.P. Herlihy and J.M. Wing, "Linearizability: A Correctness Condition for Concurrent Objects," *ACM Trans. Programming Languages and Systems*, vol. 12, no. 3, 1990, pp. 463-492.
- D. Marr et al., "Hyper-threading Technology Architecture and Microarchitecture," *Intel Technology J.*, vol. 6, no. 1, 2002, pp. 4-15.
- I. Wald, S. Boulos, and P. Shirley, "Ray Tracing Deformable Scenes Using Dynamic Bounding Volume Hierarchies," *ACM Trans. Graphics*, vol. 26, no. 1, 2007, article 6.

**Kurt Debattista** is an assistant professor at the University of Warwick's International Digital Laboratory, WMG. His research focuses on high-fidelity rendering, parallel computing, high-dynamic-range imaging, and serious games. Debattista has a PhD in computer science from the University of Bristol. Contact him at k.debattista@warwick.ac.uk.

**Piotr Dubla** is a graphics programmer at Rockstar North, a game studio, while completing his PhD at the University of Warwick's International Digital Laboratory, WMG. His research interests include interactive ray tracing, global illumination, offline physically based rendering, and realistic simulations of light transport as well as parallel processing and data structures for computer graphics and procedural content generation. Dubla has a BSc Hons (with distinction) in computer science from the University of Cape Town. Contact him at p.b.dubla@wariwck.ac.uk.

**Luis Paulo Peixoto dos Santos** is an auxiliary professor in Universidade do Minho's Department of Computer Science. His research interests include high-fidelity interactive rendering and parallel processing. Santos has a PhD in parallel processing from Universidade do Minho. Contact him at psantos@di.uminho.pt.

**Alan Chalmers** is a professor at the University of Warwick's International Digital Laboratory, WMG. His research interests include high-fidelity graphics, multisensory perception, virtual archaeology, and parallel rendering. Chalmers has a PhD in computer science from the University of Bristol. He is the honorary president of Afrigraph and a former vice president of ACM Siggraph. Contact him at a.g.chalmers@warwick.ac.uk.

**CN** Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.