Master in Intelligent Interactive Systems

Universitat Pompeu Fabra

Light Transport as an Optimal Control Problem

Mariano Zarza Pujol

Supervisor: Vicenç Gómez

Co-Supervisor: Gergely Neu

Jan 2021

upf. Universitat
Pompeu Fabra
Barcelona

Contents

Acknowledgement

I want to express my gratitude specially to my supervisor and my grandparents. They were the greatest help at the most tense moment. Furthermore, I am really grateful for the support that my close people gave me.

Abstract

Recently, reinforcement learning (RL) has attracted some attention in the research field of physically-based rendering. This thesis explores a deeper connection between both fields. In particular, we identify the light transport equations (LTE) used in physically-based rendering with the Bellman expectation equations used in optimal control. With this identification, we formulate an optimal control problem that can be used for scene rendering based on the framework of linearly-solvable Markov decision processes. This formulation provides a novel perspective on the use of RL techniques for rendering.

Keywords: Computer Graphics, Optimal Control, Light transport, Bellman equation

Chapter 1

Introduction

Recently, machine learning is attracting a lot of attention in the research field of physically based rendering. The goal of this Thesis is to explore the use of machine learning for light transport simulation, with the objective, if possible, to provide an advance to some technique. Reinforcement learning is one of the fields in ML that is really popular nowadays, Alexander Keller proposed an adaptation for computing light transport using reinforcement learning [1], uing RL strategies like Sarsa for the light transport problem. This implies several assumptions related with Optimal Control theory and Bellman Optimality. In this thesis, our main purpose is to explore different strategies of dynamic programming and control theory models like "Markov Decission Processes" .

1.1 Structure of the Thesis

The next chapter introduces the required background to related to light transport and optimal control theory. Chapter 3 describes the identification between the Bellman equations and LTE. Experimental results on a simple scene are presented in Chapter 4. Chapter 5 conludes this thesis.

Chapter 2

Background

This chapter provides a review of the necessary concepts from both lights transport and Markov Decision Processes.

2.1 Light Transport

The synthesis of photorealistic images by a computer requires the application of light transport simulation. The light transport (LTE) equation, a Fredholm integral equation of the second kind, models the distribution of light on a 3D environment. For computing the pixel colors, the LTE is usually approximated using Monte Carlo estimators. These estimators are used to sample light transport paths. For that, the contributions of light paths that begin on a camera sensor are averaged after interacting with the scene end up finding a light source. This could be compared with the trajectory of photons in real world when we see an object. Computers are only able to process a really small part of all the amount of interactions that light does in reality. So, we try to efficiently compute contributive light paths to have the best possible estimation of the light transport. Lot of research in light transport simulation have been focusing in importance sampling. IS improves the sampling process of light paths.

2.1.1 Radiometric quantities

Radiometry studies the physical measurement of the electromagnetic radiation, the basis for understanding the Global Illumination problem. The Light Transport problem is specified using radiometric quantities, we will show the basic ones.

Flux or Radiant Power (Φ)

This fundamental quantity expresses the total energy that flows through (from/to) a surface. It is the power of an electromagnetic radiation measured in Watts (**W**) (*joules/second*).

$$\Phi = \lim_{\Delta t \mapsto 0} \frac{\Delta Q}{\Delta t} = \frac{\partial Q}{\partial t}, \tag{2.1}$$

where Q refers to the radiant energy in joules (J). This energy Q, corresponds to the amount of energy that carry the photons emitted by a light source. (See subsection A.1.1)

Irradiance (E) and Radiosity (B)

If we measure the flux density over an area we get a new quantity, the flux per unit area $\partial\Phi/\partial A$. When the flux density is arriving to a surface, we call it *irradiance (E)*. Or *radiosity (B)/radiant exitance (M)* when the flux is leaving the surface. The SI unit of this quantity is watts per unit area (W/m^2). It can be generalized as,

$$E = \frac{\partial \Phi}{\partial A}. \tag{2.2}$$

Radiance (L)

Radiance captures the appearance of real world objects and is the quantity which our eyes are sensitive. We aim to compute the radiance arriving to an observer in a photo-realistic environment. We define it as the irradiance or radiant exitance arriving/leaving a surface dA over a solid angle dw. L is the flux per unit projected area $dA^{\perp}$ and per unit solid angle dw:

$$L = \frac{\partial^2 \Phi}{\partial w \partial A \cos\theta} = \frac{\partial^2 \Phi}{\partial w \partial A^{\perp}}, \tag{2.3}$$

Where $\partial A^{\perp}$ is the area of the surface or object projected in the direction perpendicular to $\partial\omega$. Solid angles dw can be thought as planar angles in 3D (See subsection A.1.2 for detailed explanation).

2.1.2 The light transport equation (LTE)

BRDF (Bidirectional Reflection Distribution function f_r)

We assume that the light entering a surface at point p with incident direction Ψ, leaves the surface at the **same point** p with direction Θ. Given this assumption, the light reflection is described by the *bidirectional reflectance distribution function* (BRDF) [2]. The BRDF is defined as the differential of outgoing radiance in a direction Θ, with respect to a differential of irradiance along incident direction Ψ. The BRDF is denoted as $f_r(x, \Psi \to \Theta)$:

$$f_r(x, \Psi \to \Theta) = \frac{dL(x \to \Theta)}{dE(x \leftarrow \Psi)}, \tag{2.4}$$

$$\frac{dL(x \to \Theta)}{L(x \leftarrow \Psi)\cos(N_x, \Psi) dw_{\Psi}}. \tag{2.5}$$

Hemispherical formulation

This formulation it is based on the *energy conservation* and can be derived directly from the BRDF equation. To get to this derivation, let us define the emitted radiance as $L_e(x \to \Theta)$ and the reflected radiance $L_r(x \to \Theta)$. Using the principle of conservation of energy, we state that the exitant radiance at a surface point x in a direction Θ, is the sum of emitted radiance $L_e(x \to \Theta)$ and reflected radiance

$L_r(x \to \Theta)$ at this point x in direction Θ. From Equation 2.4 of the BRDF we get:

$$f_r(x, \Psi \to \Theta) = \frac{dL(x \to \Theta)}{dE(x \leftarrow \Psi)},$$

$$L_r(x \to \Theta) = \int_{\Omega} f_r(x, \Psi \to \Theta) L(x \leftarrow \Psi) \cos(N_x, \Psi) dw_{\Psi},$$

for a concrete incident (Ψ) and outgoing (Θ) directions. Then the outgoing radiance $L(x \to \Theta)$ is formulated as the emitted radiance L_e plus the reflected radiance L_r. The hemispherical equation is expressed as:

$$L(x \to \Theta) = L_e(x \to \Theta) + L_r(x \to \Theta),$$

$$L(x \to \Theta) = L_e(x \to \Theta) + \int_{\Omega} f_r(x, \Psi \to \Theta) L(x \leftarrow \Psi) \cos(N_x, \Psi) dw_{\Psi}. \qquad (2.6)$$

As we can observe at Equation 2.6, the rendering equation is formed by different parts. The emitted radiance, is the outgoing radiance from the own material a part from the reflected light, for example, area light sources or fluorescence materials. The reflected radiance, derived from the BRDF, is the integration over the hemisphere of the incident radiance with respect to the BRDF function. In the integrand we find this BRDF function that defines how the incident radiance is going to be reflected. Then, we find the cosine of the angle between the incident direction and the surface normal. The cosine term compensates the angle between the direction of the incident radiance and the surface where it is arriving, and allows to express the power per unit projected area $dA^{\perp}$, one of the derivatives of radiance. Finally, the term $L(x \leftarrow \Psi)$ is the incident radiance arriving to the point from all the scene. $L(x \leftarrow \Psi)$ is the term of the equation being evaluated. and the unknown quantity of the rendering equation.

Our starting point is thus the radiance equation. For clarity, we will use the following equivalent notation to refer to the LTE equation

$$L(x,\omega) = L_e(x,\omega) + \int_{\mathcal{S}^+(x)} L(h(x,\omega_i), -\omega_i) f_r(\omega_i, x, \omega) \cos\theta_i d\omega_i. \tag{2.7}$$

2.2 Optimal Control and Reinforcement Learning

Optimal control theory is a field in mathematical optimization with a long history and is concerned with the problem of designing a controller to minimize a measure of a dynamical system's behavior over time [3]. Reinforcement learning (RL) appeared more recently as a paradigm of machine learning at the intersection of psychology of animal learning and optimal control [4]. Optimal control and RL face very similar challenges and both disciplines make emphasis on different aspects of sequential decision making problems. For example, while optimal control traditionally considers methods that require complete knowledge of the system to be controlled, RL puts more emphasis on the idea of trial-and-error learning from environment interactions. Despite these differences, both fields share many theories and solution methods, in particular dynamic programming and Markov decision processes (MDPs), which we briefly present next.

2.2.1 Markov Decision Processes

Markov decision processes (MDP) are useful mathematical models that formalize sequential decision problems in the presence of uncertainty [5, 6]. The standard MDP model obeys the following steps: a state is observed, a control is applied to that observed state, a price is paid for exerting such control, and the state is updated probabilistically. This process is repeated. We define:

- A set of states $\mathcal{X}$.
- A set of admissible controls $\mathcal{U}(x)$ at each state.
- The system dynamics, a transition probability $p(x'|x,u)$, for $u \in \mathcal{U}(x)$.

- An instantaneous cost function $g(x, u)$.
- And (optionally) a final cost function $g_f(x)$ evaluated at the final state(s).

A control law or policy, denoted by $\pi(u|x)$ maps states to controls. Policies can be stochastic if the applied control at state x is chosen from $\mathcal{U}(x)$ according to some probability distribution, or deterministic if one action is applied to a given state with probability one (Dirac-delta policies). For a fixed policy π, either stochastic or deterministic, the dynamics becomes autonomous, and the MDP can be described as a Markov process with transition probability

$$p_\pi(x'|x) = \sum_{u \in \mathcal{U}(x)} p(x'|x, u)\pi(u|x). \tag{2.8}$$

From such a process, one can sample sequences of states and controls, also called trajectories $\tau = (x_0, u_0, x_1, u_1 \ldots, x_f)$. In the so-called *first-exit* formulation, the states are either non-terminal $\mathcal{N}$ or terminal $\mathcal{T}$ states. A trajectory finishes when one of a set of terminal states $x_f \in \mathcal{T}$ is reached for the first time and has probability

$$p(\tau|x_0) = \prod_{t=0}^{t_f - 1} p_\pi(x_{t+1}|x_t). \tag{2.9}$$

Associated to a trajectory τ there is an additive cost, which accumulates through all state-control pairs in the trajectory

$$C(\tau|x_0) = g_f(x_f) + \sum_{t=0}^{t_{f-1}} g(x_t, u_t). \tag{2.10}$$

Policy Evaluation: we first define the *policy evaluation* (a.k.a. prediction) problem, which corresponds to estimate the expectation of Equation (2.10), where expectation is taken over trajectories generated according to (2.9)

$$J_\pi(x) = \mathrm{E}_{\substack{x' \sim p(\cdot|x,u) \\ u' \sim \pi(\cdot|x')}} \left[C(\tau|x) \right], \text{ for } x \in \mathcal{N}, \tag{2.11}$$

and $J_\pi(x) = g_\mathcal{T}(x)$, for terminal states $x \in \mathcal{T}$.

This function is known as the cost-to-go starting at state x and applying the policy π thereafter. The Bellman expectation equations are a set of self-consistent equations that allow to compute J_π recursively. In the first-exit formulation of an MDP, these equations (one per each non-terminal state) become

$$J_\pi(x) = \mathrm{E}_{\substack{u\sim\pi(\cdot|x) \\ x'\sim p(\cdot|x,u)}} \Big[g(x,u) + J_\pi(x') \Big], \text{ for } x \in \mathcal{N}, \tag{2.12}$$

and $J_\pi(x) = g_\mathcal{T}(x)$, for terminal states $x \in \mathcal{T}$.

Equivalently, one can consider cost-to-go defined for state-control pairs (also known as Q-functions)

$$J_\pi(x,u) = g(x,u) + \mathrm{E}_{\substack{x'\sim p(\cdot|x,u) \\ u'\sim\pi(\cdot|x')}} \left[J_\pi(x',u') \right]. \tag{2.13}$$

The prediction problem can be solved iterating the above linear equations, which is a form of dynamic programming. We can use linear algebra and express them using vector-matrix notation. We define

- A vector $\mathbf{j}$ stacking all J_π of non-terminal states.
- A cost matrix G for each $g(x,u)$ at non-terminal states and end-cost vector $\mathbf{g} = g(x_f)$ for terminal states.
- Two matrices $P_\mathcal{N} = p_\pi(x'|x)$ for $x' \in \mathcal{N}$, and $P_\mathcal{T} = p_\pi(x'|x)$ for $x' \in \mathcal{T}$.

The cost-to-go for a fixed policy can be solved analytically

$$\mathbf{j} = G + P_\mathcal{N}\mathbf{j} + P_\mathcal{T}\mathbf{g} \tag{2.14}$$

$$= (I - P_\mathcal{N})^{-1}(G + P_\mathcal{T}\mathbf{g}). \tag{2.15}$$

Policy Optimization: The control problem is to find the optimal policy, the one that minimizes the expected cost-to-go

$$J^*(x) = \min J_\pi(x), \tag{2.16}$$

$$\pi^*(x) = \arg\min_\pi J_\pi(x), \tag{2.17}$$

that can again be written recursively as a set of equations, known as the Bellman optimality equations

$$J^*(x) = \min_u \mathrm{E}_{\substack{u\sim\pi(\cdot|x)\\ x'\sim p(\cdot|x,u)}} \Big[g(x,u) + J^*(x')\Big]. \tag{2.18}$$

In this case, the presence of the min function makes the system non-linear, rendering the control problem fundamentally harder than the prediction problem.

Value iteration is a dynamic programming algorithm that computes the optimal cost-to-go by iterating the Bellman optimality equations until a fixed point is reached.

An alternative procedure is policy iteration, which alternates a policy evaluation step with a policy improvement step. One can perform policy evaluation using the formulation of Equation (2.14) to solve a system of the form $A_\pi \mathbf{j} = \mathbf{b}_\pi$; then, the policy has to be improved and the process repeated.

In the next section, we introduce a subclass of problems that makes certain assumptions on the underlying MDP that greatly simplify the policy optimization problem.

2.2.2 Linearly-Solvable MDPs

Linearly-Solvable MDPs (or LMDPs) were introduced by Todorov [7] in the discrete-time case, and Kappen [8] in the continuous time formulation. The main idea is to consider an MDP where

- The system dynamics is defined as a probability distribution over next states (the passive dynamics), a Markov chain that characterizes how the system evolves in the absence of control

$$p(x'|x) = p(x'|x, u = \mathbf{0}). \tag{2.19}$$

- The controls are also another Markov chain, i.e., probability distributions over next states that are constrained by the passive dynamics in the sense that transitions not allowed by p are also forbidden for u

$$u(x'|x) = p_\pi(x'|x), \qquad u(x'|x) = 0, \quad \text{for } p(x'|x) = 0. \tag{2.20}$$

- The instantaneous cost consists of two terms: a state-dependent term that penalizes “bad” states $g(x)$, and an control term that penalizes controls u that deviate from p in an entropic/information sense.

$$g(x, u) = g(x) + \lambda \sum_{x'} \log \frac{u(x'|x)}{p(x'|x)}, \tag{2.21}$$

where the term λ balances both terms. For first-exit problems, we set $g(x) = g_f(x)$. The controller u is thus free to reshape p in any way, but pays a price for deviating from it while at the same time should avoid visiting “bad” states.

This class of optimal control problems is also known under other names: Kullback-Leibler control (as derived for probabilistic graphical models [9]), path-integral control (as derived in the continuous-time case [10, 11]), and is closely related to maximum entropy RL [12].

LMDPs have the interesting property that the Bellman optimality equations are linear in the cost-to-go, reducing significantly the complexity of the control problem to the same as the prediction problem. Effectively, the max operator can be removed after an adequate (log) transformation of the state cost, and because all the stochasticity (noise) is controllable.

We show the formulation for first-exit problems only, for more details see [13]. We have

- A vector $\mathbf{z}$ with one element per state, representing the log-transformed cost-to-go, or desirability function $z(x) = \exp(-J(x))$.
- A stochastic matrix P encoding the passive dynamics $p(x'|x)$, where the row index corresponds to x and the column index to x'
- A diagonal matrix Q with elements $\exp(-\lambda g(x))$ along its main diagonal.

The optimal cost-to-go (or desirability function) can be obtained starting from a randomly initialized vector $\mathbf{z}$ and iterating the following fixed point equation

$$\mathbf{z} \leftarrow QP\mathbf{z}, \tag{2.22}$$

which transform $\mathbf{z}$ to the leading eigenvector of QP. From $\mathbf{z}$, the optimal control and cost-to-go can be computed as [13]

$$u^*(x'|x) = \frac{p(x'|x)z(x')}{\mathrm{E}_{x' \sim p(\cdot|x)}\Big[z(x')\Big]}, \tag{2.23}$$

$$J^*(x) = -\lambda \log z(x). \tag{2.24}$$

In addition to leading to a linear Bellman equation, LMDPs enjoy several computational advantages and theoretical properties, such as compositionality of optimal control laws [14, 15, 16], convexity of the inverse optimal control problem [17], or fast rates in the online learning setting [18]. LMDPs have been applied in numerous settings for robotics [19, 20, 21], multi-agent systems [22, 23, 24], or for finding policies in other complex scenarios such as power grids [25], online forums [26], crowd-sourcing [27], or consequential rankings [28].

In the next chapter the connection between the light transport equation and optimal control will become clear.

Chapter 3

Light Transport and Stochastic Optimal Control

This chapter presents a formulation of the Light Transport as an optimal control problem. We can identify the light transport equations defined in (2.7) and the *Bellman expectation equations* (2.13) defined for general MDPs that we repeat here for clarity

$$\begin{aligned}
\textbf{LTE}: \quad & L(x,\omega) = L_e(x,\omega) + \int_{\mathcal{S}^+(x)} L(h(x,\omega_i), -\omega_i) f_r(\omega_i, x, \omega) \cos\theta_i d\omega_i. \\
\textbf{Bellman}: \quad & J_\pi(x,u) = g(x,u) + \mathrm{E}_{\substack{x' \sim p(\cdot|x,u) \\ u' \sim \pi(\cdot|x')}} \left[J_\pi(x', u') \right].
\end{aligned}$$

We see that the (state-action) cost-to-go $J_\pi(x,u)$ corresponds to the exiting radiance $L(x,\omega)$ on a surface point x following ongoing direction ω. The expectation over trajectories appears as an integral. More precisely:

- The state space $\mathcal{X}$ is continuous and corresponds to 3D coordinates.

- Controls are also continuous and correspond to directions ω.

- The instantaneous cost function $g(x,\omega)$ corresponds to the emitted radiance $L_e(x,\omega)$.

- The dynamics is deterministic $p(x'|x, \omega) = \delta(x', h(x, \omega_i))$ and corresponds to the ray-casting function h.
- Thanks to the previous knowledge about the light transport problem, the policy could be different depending on the component. In our case, the policy we want to optimize is stochastic with density function defined as the Bidirectional Reflection Distribution function (BDRDF) $f_r(x, \omega, \omega_i)$.
- The set of terminal states correspond to the light sources in the scene.

Strictly speaking this identification only takes place after proper discretization of the LTE integral. We postpone this issue and elaborate on this later.

This result has an important implication: the radiance at a point in the scene can be computed by solving a linear system of equations. This is a novel result, to the best of our knowledge.

3.1 Linearly-Solvable MDP formulation

Since the dynamics is deterministic, the MDP considered above satisfies the conditions to be written as a LMDP. This has additional implications in terms of solution methods for rendering. We define the new state as the concatenation of coordinates x and direction ω, and denote it as $\mathrm{x} = (x, \omega)$. Let us define the two formulation (optimal control and light transport) together.

3.1.1 State space representation

In computer graphics, there are multiple data structures to represent radiance and other terms of the LTE. For our purpose, we base our structure in a variation of the irradiance volumes [29] and the representation that is used by Keller for its RL approach [1]. For us, the discrete state space is placed into the scene as shown in Figure 1: a number of points are sampled for each surface proportionally to its area. For each sampled point x, we stratify an hemisphere centered at x. The point x paired with a stratum k is a discrete state $\mathrm{x} = (x, \omega)$.

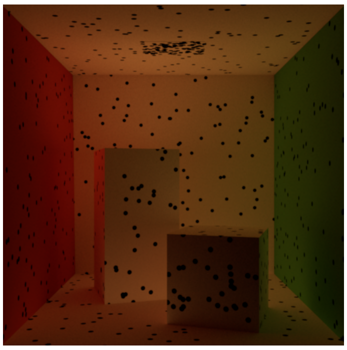


Figure 1: Happy Smiley

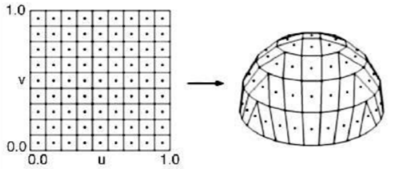


Figure 2: Unit square grid mapped to hemispherical coordinates

Each stratum captures the passive dynamics from a set of possible incident directions $S^2(\theta, \phi)$ in its region. The hemisphere region of the stratum will define a constant reflectance value over its directions. This results in a piece-wise distribution of the dynamics over S^+ at this point. We decided to uniformly stratify the hemisphere into same area regions for each state. This is done by mapping a uniform grid from the unit square, like Figure 2. If the states at a point x held different area regions, the dynamics discretization would be affected by the different da of the stratum's. The commonly used latitude-longitude mapping produce different areas for the same bunch of directions, induced by the nature of the solid angle (subsection A.1.2) and the hemispherical coordinates subsection A.2.1. To avoid this distortion when representing the passive dynamics, we use the hemispherical mapping proposed by [30]. This mapping allows to preserve the uniform areas of a unit square grid into the hemisphere.

The resolution of the discretised state space is tuned by two components. The number of points at each surface, and the number of stratum's at each point. Changing each of the two components varies linearly the size of the discretization. If we fix to k either of the two components of the resolution, the size increase from the other component with the order of $O(kn)$, linearly proportional to a value k. If we take into the account the two components, the size of the data increases quadratically to the order of $O(mn)$, where m is the number of points and n the states per point. This is one of the main points that justifies our data structure. The state space size increases linearly with the number of points or stratum's. Usually, the data structures used for representing radiance or other quantities, increase exponentially to the number of points in the scene.

3.1.2 The Passive Dynamics

In the light transport equation, we defined as the passive dynamics a uniform density over the (discretized) volume weighted by the BDRDF (the $\cos\theta$ at the hemispherical formulation)

$$p(\mathrm{x}' = (x', \omega') \mid \mathrm{x} = (x, \omega)) = \begin{cases} \epsilon & \text{if } \mathrm{x} = \mathrm{x}', \text{ or } G(x, x') = 0 \\ \frac{1}{Z(\omega)}(\frac{\cos\theta}{\pi} f_r(\omega', x, \omega)) & \text{for } \mathrm{x}' = (h(x, \omega'), -\omega') \\ 0 & \text{otherwise,} \end{cases} \tag{3.1}$$

where $Z(\omega) = \sum_{\omega_i} \frac{\cos\theta}{\pi} f_r(\omega', x, \omega)$ sums over all discretized directions and ensures that the passive dynamics is properly normalized. $\epsilon = 0.00005$ and $G(x, x')$ is the geometry term from the LTE area formulation, at appendix.

3.1.3 The State-Cost function

For simplicity we assume that the emitted radiance is constant for all surfaces. This means that we can set the state-cost term to a constant $g(\mathrm{x}) = 1$. In our environment, only the light sources emit radiance and will correspond to zero state cost. The immediate cost of a state is being determined by the emitted radiance to the whole scene from the light sources. Our optimal control formulation, tries to maximize the radiance being reflected from a point x to x'.

Similarly to an application of [13] for shortest path finding, the diagonal matrix Q will have $\exp(-\lambda \cdot 0) = 1$ for terminal states and $\exp(-\lambda \cdot 1) > 0$ for non-terminal ones. Although we will experiment with different values of λ, we choose initially to be the sum of the power per unit area dA from all the scene light sources. Then, the total power sum is scaled with exponents of 10: $\{0.001, 0.01, 0.1, 1...\}$. We compute the solution for different values of λ to obtain different densities of $\mathbf{z}$.

3.1.4 Combining passive dynamics and state-costs and the power method

Following the iterative method subsection 2.2.2, for n states, we represent the desirability $z(\mathrm{x})$ and the immediate cost $g(\mathrm{x})$ with the $(n, 1)$ column vectors $\mathbf{z}$ and Q . The passive dynamics $p\,(\mathrm{x'}|\mathrm{x})$ is stored as a $\mathbf{n}$-by-$\mathbf{n}$ matrix P, where the row index correspond to x and the column to the next state x'. The vector $\mathbf{z}$ is initialized with random values between $[0, 1)$ before the iteration begins.

The construction of the matrix P is the challenging part. To represent the formulated passive dynamics in P, we compute the product $\frac{\cos\theta}{\pi} f_r(\omega', x, \omega)$ for each pair of states x and x'. Notice that many transitions x → x' are forbidden, resulting in a very sparse matrix. This sparsity comes from the fact that the stratum is a bunch of incoming directions. Since the stratum determines each state, lots of state tuples are non valid due to the nature of reflections in light transport. It was already of our interest to produce a sparse matrix, so only contributing directions will be taken into account. The values of matrix P are assigned following equation 3.1. For a self-transition of a state with itself (x = x'), a small enough probability value is given to keep the Markov chain non-periodic, a condition required by the theory. Due to the state space data structure, the visibility term $V(\mathrm{x}, \mathrm{x}')$ between points of the scene geometry is included the matrix P. We evaluate the geometry term G for each pair of states (x, x') to determine if going from state x to x' is possible. See the LTE area formulation for more details about the geometry term. subsection A.1.1.

3.1.5 From the Optimal Controls to Light-Tracing

The optimal control $u^*(\mathrm{x}'|\mathrm{x})$ is given in closed form

$$u^*(\mathrm{x}'|\mathrm{x}) = \frac{p(\mathrm{x}'|\mathrm{x})z(\mathrm{x}')}{\mathrm{E}_{\mathrm{x}'\sim p(\cdot|\mathrm{x})}\Big[z(x')\Big]}, \tag{3.2}$$

once it is computed from the iterative method. An estimate of $L(x, \omega)$ can be then computed as $L(x, \omega) = -\lambda \log(z(\mathrm{x}))$.

Each row vector $u^*(\cdot|x)$ of $\mathbf{u}*$ is a discrete probability distribution for reaching x' from x under the optimal dynamics (the one that provides the best tradeoff between maximizing the distance to the light source and deviating from the passive dynamics). Then, each state x stores a discrete pdf. The pdf stores a probability for each outgoing stratum for a point and an incident direction. For only using u* to compute the radiance, the integral from the LTE is approximated like:

$$\int_{\mathcal{S}^+(x)} L(h(x,\omega_i), -\omega_i) f_r(\omega_i, x, \omega) \cos\theta_i d\omega_i \cong \frac{\pi}{n} \sum_{k=0}^{n} \frac{L(y, -\omega_k) f_r(\omega_k, x, \omega) \cos\theta_k}{u(x'_k|x)}, \tag{3.3}$$

where $y = h(x, \omega_i)$ .Direction ω_k is obtained by sampling the pdf stored at the point x of state x. For doing this we use the inversion method, see chapter B. We sample N times the inverse CDF and we obtain the next state and its probability. ω_k is sampled according to the direction between x and x'.

Chapter 4

Experimental Results

This chapter presents an integration and implementation of the previously described models in a path tracing algorithm. Optimal control models, usually have an state space representing all the possible states of the problem. This part will be the one more challenging to implement. It is important to define an appropriate structure to represent the light distribution over all the locations. The irradiance volume [29], widely used on light transport, is one of the best candidates. For doing this, we setup different scenes environments to be rendered under the different formulations proposed.

4.1 Optimal control computation and results

4.1.1 Convergence of the power method

We first evaluate empirically the speed of convergence of $\mathbf{z}$ to the largest eigenvector that determines the solution of the equation. We check the convergence of $\mathbf{z}$ at each iteration by computing the norm of the difference between two consecutive iterations $\mathbf{z}_i$ and the previous $\mathbf{z}_{(i-1)}$. We compute $\mathbf{z}$ by iterating the power method until a maximum number of iterations or a distance tolerance is reached. The tolerance used for the results is $\rho = 1x10^{-40}$. To see the convergence we visualize how the distance varies respect to the iteration number. The way the convergence changes in function of δ, is a hint for knowing if the power method is converging to a proper solution. The uniformity of the solutions with respect the matrix P should depend on delta.

Figures 3 and 4 are the convergences for a state space with 32 points sampled per surface and 8 states per point. We can observe that for high deltas the method converges really fast, and the opposite for small ones. This results have been tested for different number of states, but changing the resolution has its interest at the render task. At the log scale, for each delta we can better see the number of iterations it takes to reach the tolerance.

However, we need to set the adequate delta to obtain the desired uniformity for the distributions of $u*$. To find the range of deltas suitable for converging our problem we look around the range $\lambda = [0.1, 1.5]$. Fig.3 and fig.4 show that the method converges fast to set of distributions, then keeps iterating doing until the tolerance is reached. However an enough small tolerance is achieved at $50 - 100$ iterations. Small λ show a slow convergence and uniform distributions with respect to the passive dynamics. As the delta keeps growing, converges faster and the

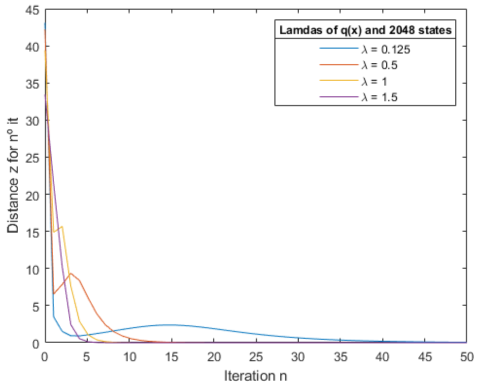


Figure 3: Convergence of z.

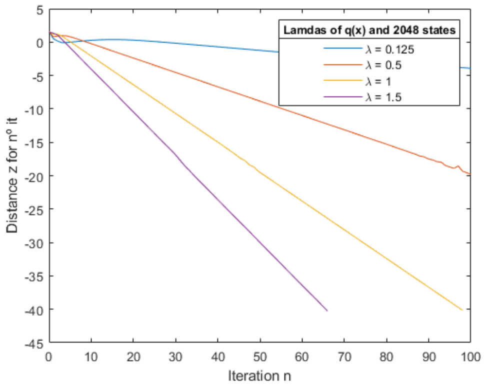


Figure 4: z distance in log scale.

4.2 u* render results

The results show that estimating the radiance using the u* for sampling the directions converges to a visual solution. However, artifacts can be detected.If we look at Figure 6, we can see how low resolutions of the state space strongly affects the artifacts. The artifacts are generated by the discretisation of the problem, the state space data structure, and the power method. Due to $\lambda = 1$ and the scene geometry, the points more at the back part of the scene, have more possible next states. As we increase the resolution, the artifacts start disappearing from the back to the front. For example, let us take a look to the points pointing to the camera at the nearest box. They have a lot of possible directions that will not be contributive. The nature of the power method and the chosen λ, results in a certain distribution for these points. This distribution has strong probabilities for just a few next states and 0 for a lot of possible directions.

At Figure 7, we can see how increasing the resolution increases the convergence, this fact gives us the intuition that with precise modifications u* could be used for slightly increase the convergence of usual MC methods. To obtain MC noisy images, which are better for human eye. We relax the discretization effects by mixing the optimal control with stochastic Monte Carlo. For adapting the method to the typical rendering tasks, we setup an Octree data structure for saving the points and its respective optimal distributions. Then, a routine for constructing the CDF's for each point state x is performed. Relaxing the distribution as desired for obtaining more or less uniform distributions.

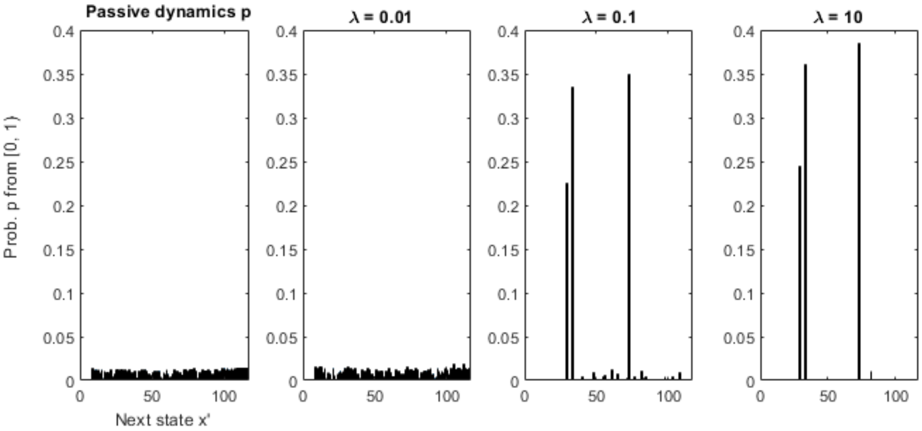
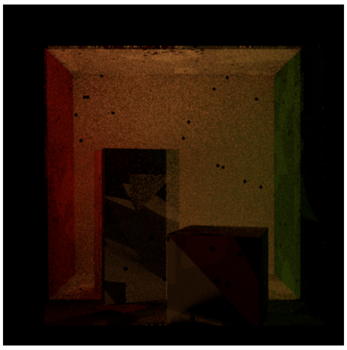
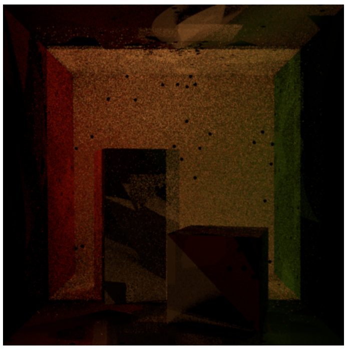


Figure 5: For seeing the distributions of u^* and its variation through different deltas, we look for a concrete state s and check the distribution for going to all x', u^* (x, $\cdot$) . This results in a discrete PDF for each state represented by a n-row vector $\mathbf{u}_{\mathrm{x}}$. See in this figure an example.

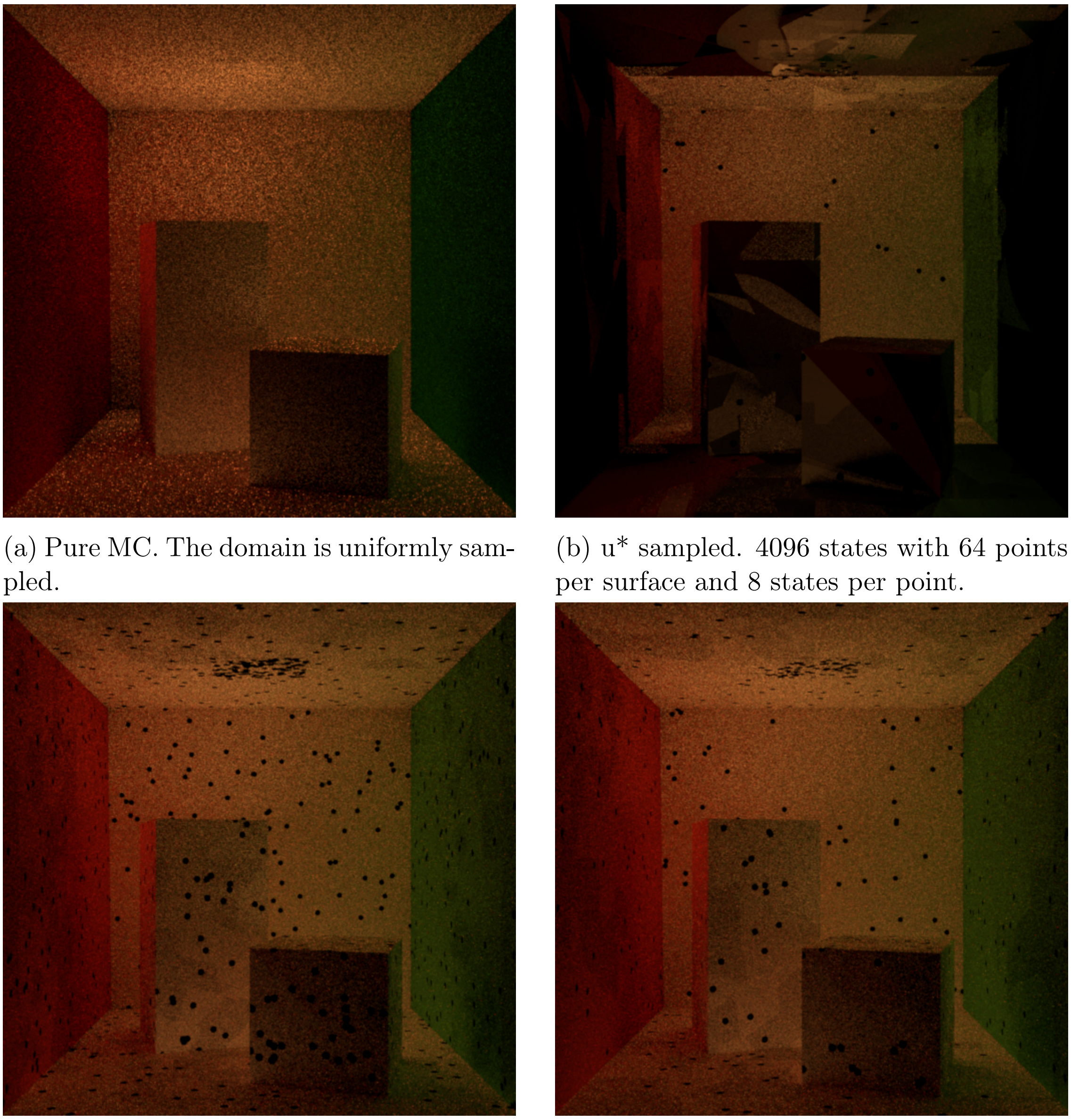


(a) $\lambda = 0.01$



(b) $\lambda = 0.1$

Figure 6: Small state space with only 2048 states. The image is computed with 16 samples per pixel. For such small number of states, the u* gets too much affected by the visibility between states. Notice that the part of the scene from where less points are observable, is bad estimated. The positioning of the light source at the back part of the ceiling also affects.



(a) Pure MC. The domain is uniformly sampled.

(b) u* sampled. 4096 states with 64 points per surface and 8 states per point.

(c) u* sampled. 8192 states with 128 points per surface and 8 states per point.

(d) u* sampled. 8192 states with 64 points per surface and 16 states per point.

Figure 7: The renders are done with 16 samples per pixel. Only Indirect illumination. In this case, the λ and resolutions are precisely chosen to obtain better visual results. MC noisy images and avoid fast converged pixel values.

Chapter 5

Discussion and Conclusions

In this work, we have identified an equivalence between Optimal control and light transport. We open the possibility of exploring more adequate Optimal Control methods that can end unifying both disciplines. At least, a theoretical formulation is given, which can allow to understand better the use of methods derive from Optimal Control theory. Through the results, we show that there is an underlying between estimation and control from LMDPs, that also has a counterpart in rendering. The sampling methods used for rendering such as importance sampling, etc... can be viewed as optimization problems, where a cumulative radiance is maximised with a proper Optimal control formulation. Is not common in rendering to address the problem using algebraic methods such as finding the largest eigenvector of a matrix. A similar identification has been recently introduced in the literature between the light transport equation with a particular action-value update used in reinforcement learning. This connection has been done superficially and is focused in a particular algorithm for reinforcement learning. Our identification, is rooted in the bellman equations, providing a deeper relationship between the equations. Also, we can think about Optimal Control as a possible way for solving some Fredholm integrals of the second kind. The LTE follows this integrals schema. Additionally, with some adaptations and tuning of the power method, it seems that we could obtain a way to improve pure MC with a MC boosted by the linear-MDP solution. In conclusion, We provide an Optimal control formulation of the LTE solved by a linear method. Which implies a more precise definition of the resemblance between the equations of the two fields. The method and optimal policy obtained, converge into a correct solution. This leads us to confirm our firsts hypothesis.

List of Figures

Bibliography

[1] Dahm, K. & Keller, A. Learning light transport the reinforced way. In *ACM SIGGRAPH 2017 Talks*, SIGGRAPH '17 (ACM, New York, NY, USA, 2017).

[2] Nicodemus, F. E., Richmond, J. C., Hsia, J. J., Ginsberg, I. W. & Limperis, T. Radiometry. In Wolff, L. B., Shafer, S. A. & Healey, G. (eds.) *Radiometry*, chap. Geometrical Considerations and Nomenclature for Reflectance, 94–145 (Jones and Bartlett Publishers, Inc., USA, 1992).

[3] Sussmann, H. J. & Willems, J. C. 300 years of optimal control: from the brachystochrone to the maximum principle. *IEEE Control Systems Magazine* **17**, 32–44 (1997).

[4] Sutton, R. S. & Barto, A. G. *Introduction to Reinforcement Learning* (MIT Press, Cambridge, MA, USA, 1998), 1st edn.

[5] Bertsekas, D. P. *Dynamic Programming and Optimal Control* (Athena Scientific, 2000), 2nd edn.

[6] Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming* (John Wiley & Sons, Inc., USA, 1994), 1st edn.

[7] Todorov, E. Linearly-solvable markov decision problems. In *Advances in neural information processing systems*, 1369–1376 (2007).

[8] Kappen, H. J. Linear theory for control of nonlinear stochastic systems. *Physical review letters* **95**, 200201 (2005).

[9] Kappen, H. J., Gómez, V. & Opper, M. Optimal control as a graphical model inference problem. *Machine learning* **87**, 159–182 (2012).

[10] Kappen, H. J. Path integrals and symmetry breaking for optimal control theory. *Journal of Statistical Mechanics: Theory and Experiment* **2005**, P11011–P11011 (2005).

[11] Theodorou, E. A. & Todorov, E. Relative entropy and free energy dualities: Connections to path integral and KL control. In *Proceedings of the 51th IEEE Conference on Decision and Control, CDC 2012, December 10-13, 2012, Maui, HI, USA*, 1466–1473 (IEEE, 2012).

[12] Ziebart, B. D. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. Ph.D. thesis, USA (2010).

[13] Todorov, E. Efficient computation of optimal actions. *Proceedings of the national academy of sciences* **106**, 11478–11483 (2009).

[14] Todorov, E. Compositionality of optimal control laws. In *Advances in Neural Information Processing Systems*, 1856–1864 (2009).

[15] Jonsson, A. & Gómez, V. Hierarchical linearly-solvable markov decision problems. In *26th International Conference on Automated Planning and Scheduling*, ICAPS'16, 193–201 (AAAI Press, 2016).

[16] Saxe, A. M., Earle, A. C. & Rosman, B. Hierarchy through composition with multitask LMDPs. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 3017–3026 (JMLR. org, 2017).

[17] Dvijotham, K. & Todorov, E. Inverse optimal control with linearly-solvable mdps. In *International Conference on Machine Learning*, 335–342 (2010).

[18] Neu, G. & Gómez, V. Fast rates for online learning in Linearly Solvable Markov Decision Processes. In *Proceedings of the 2017 Conference on Learning Theory*, vol. 65 of *Proceedings of Machine Learning Research*, 1567–1588 (PMLR, 2017).

[19] Williams, G., Aldrich, A. & Theodorou, E. A. Model predictive path integral control: From theory to parallel computation. *Journal of Guidance, Control, and Dynamics* **40**, 344–357 (2017).

[20] Gómez, V., Kappen, H. J., Peters, J. & Neumann, G. Policy search for path integral control. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 482–497 (Springer, 2014).

[21] Matsubara, T., Gómez, V. & Kappen, H. J. Latent Kullback Leibler control for continuous-state systems using probabilistic graphical models. *30th Conference on Uncertainty in Artificial Intelligence (UAI)* (2014).

[22] Van Den Broek, B., Wiegerinck, W. & Kappen, B. Graphical model inference in optimal control of stochastic multi-agent systems. *Journal of Artificial Intelligence Research* **32**, 95–122 (2008).

[23] Gómez, V., Thijssen, S., Symington, A. C., Hailes, S. & Kappen, H. J. Real-time stochastic optimal control for multi-agent quadrotor systems. In *26th International Conference on Automated Planning and Scheduling* (2016).

[24] Wan, N., Gahlawat, A., Hovakimyan, N., Theodorou, E. A. & Voulgaris, P. G. Cooperative path integral control for stochastic multi-agent systems. *arXiv preprint arXiv:2009.14775* (2020).

[25] Chertkov, M., Chernyak, V. Y. & Deka, D. Ensemble control of cycling energy loads: Markov decision approach. In *Energy Markets and Responsive Grids*, 363–382 (Springer, 2018).

[26] Thalmeier, D., Gómez, V. & Kappen, H. J. Action selection in growing state spaces: control of network structure growth. *Journal of Physics A: Mathematical and Theoretical* **50**, 034006 (2016).

[27] Abbasi-Yadkori, Y., Bartlett, P., Chen, X. & Malek, A. Large-scale Markov decision problems with KL control cost and its application to crowdsourcing. In *International Conference on Machine Learning*, 1053–1062 (2015).

[28] Tabibian, B., Gómez, V., De, A., Schölkopf, B. & Gomez Rodriguez, M. On the design of consequential ranking algorithms. In Peters, J. & Sontag, D. (eds.) *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, vol. 124 of *Proceedings of Machine Learning Research*, 171–180 (PMLR, Virtual, 2020).

[29] Greger, G., Shirley, P., Hubbard, P. & Greenberg, D. The irradiance volume. *Computer Graphics and Applications, IEEE* **18**, 32–43 (1998).

[30] Shirley, P. & Chiu, K. Notes on adaptive quadrature on the hemisphere technical report number 411 (1994).

Appendix A

Radiometry and Light Transport

In photorealistic rendering we often integrate the radiometric quantities explained in Chapter 2.1. I will present an explanation of the integrals that we will evaluate and some tricks to simplify the task.

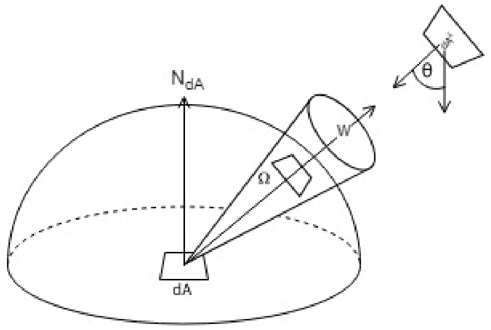


Figure 8: The unit projected area $dA^{\perp}$ is perpendicular to the direction w in which the solid angle Ω is centered.

A.1 Radiometry further concepts

A.1.1 Computation of energy Q

Energy $Q = hc/\lambda$,is computed using the velocity of light c, the plank constant h, the wavelength of the photon λ.

A.1.2 Solid angles

A.2 Projected area

In Equation 2.3 we introduced the differential projected area ($dA^{\perp}$). This differential is obtained by applying a cosine term to the differential area dA. This term stems from the way we measure radiance. We measure radiance counting the photons passing through a small surface ($dA^{\perp}$) perpendicular to direction w. The direction w is the one where the solid angle is centered. An intuitive idea about this is the fact that the flux arriving to a large surface is distributed diffusely so we have to take into account the larger area. Figure 8 shows the concept of projected area. From these we state the following relation:

- Differential projected area: $dA^{\perp} = cos\theta dA$.

The angle θ is the angle between the surface normal n and the direction w of the solid angle.(Figure 8)

A.2.1 Integrating over the hemisphere

In realistic rendering, we often work on the hemisphere centered around a point. In this project we mostly integrate the functions over the hemisphere. All the possible directions from a surface point is what we want to represent with the hemisphere. In fact an hemisphere is a 2D space formed by all the directions with origin at the center of the hemisphere. To parameterize this directions we use hemispherical coordinates.

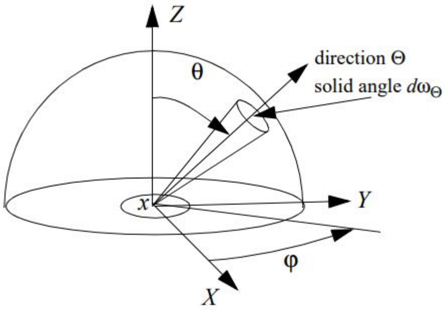


Figure 9: Hemispherical coordinates. Direction $\Theta = (\varphi, \theta)$

In hemispherical coordinates a direction is defined by two angles, see Figure 9. The range of values for the angles θ and φ are:

$$\varphi \in [0, 2\pi],$$

$$\theta \in [0, \pi/2].$$

The angle φ represents the azimuth, and is measured respect to the axis placed in the tangent plane of point x. Notice that φ moves around a complete circumference, this is why its range of values goes from 0 to 2π. Angle θ represents the elevation respect the normal vector at surface point x, this can be represented with just a fourth part of circumference values. Now that we have defined directions on the hemisphere, we can obtain a 3D point adding a distance r along the direction Ω. Using trigonometry we can transform between Cartesian(XYZ) and Spherical coordinates easily:

$$x = r\cos\varphi\sin\theta, y = r\sin\varphi\cos\theta, z = r\cos\theta. \tag{A.1}$$

In rendering algorithms, we often integrate functions that are defined over the directions of a surface point. This means that the integral evaluations are expressed per unit hemisphere, so the radius r is equal to 1. When $r = 1$ we are , in fact, defining directions or points exactly on the hemisphere .

There is a remarkable difference between spherical coordinates and Cartesian. For a concrete differential solid angle $d\Theta$, its corresponding area on the hemisphere is larger near the horizon and smaller at the poles. The differential adds a $\sin\theta$ factor to take the area differences into account. Following this, we define a differential solid angle as:

$$dw_\Theta = \sin\theta d\theta d\varphi.$$

Then the integration for a function $f(\Theta) = f(\theta, \varphi)$ over an hemisphere is expressed:

$$\int_\Omega f(\Theta)dw_\Theta = \int_0^{2\pi}\int_0^{\pi/2} f(\theta, \varphi)\sin\theta d\theta d\varphi. \tag{A.2}$$

Appendix B

Monte Carlo

B.1 Inversion Method, sampling a PDF.

Consists in sampling according to a given PDF using the *inverse cumulative distribution function* of $p(x)$. This is done by evaluating the CDF^{-1} of $p(x)$. This method requires to compute analytically the CDF and its inverse. The process to compute a sample y from a PDF follows the steps below:

1. Compute the CDF P(x) integrating the PDF : $P(Y) = \int_{-\infty}^{Y} p(x)dx$.
2. Compute the inverse CDF $P^{-1}(Y)$.
3. Sample a uniformly distributed random variable X.
4. Finally, compute sample y evaluating $P^-1(Y)$ with X sample: $y = P^-1(X)$.

This method is used to generate multidimensional samples for evaluating the rendering integral. And the random variable that it is used for this is the canonical uniform random variable ξ, a continuous uniformly distributed random variable.