

Fluorescence in Bidirectional Rendering

Master Thesis

Alisa Jung

An der Fakultät für Informatik
Institut für Visualisierung und Datenanalyse,
Lehrstuhl für Computergrafik

May 30, 2017

Reviewer:	Prof. Dr.-Ing. Carsten Dachsbacher
Second reviewer:	Prof. Dr. Hartmut Prautzsch
Third reviewer:	Prof. Ph.D. Steve Marschner
Advisor:	Dr. Johannes Schudeiske

Abstract

Fluorescence - the effect of a photon being absorbed at one wavelength and re-emitted at another - is present in many common materials such as clothes and paper. Yet there has been little research in rendering fluorescence, and previous projects are all based on a simple forward path tracer.

This thesis shows how fluorescence can be simulated by a bidirectional path tracer. We propose a simple model for fluorescent BRDFs that can be parameterized by physical measurements of fluorescent spectra, and demonstrate how this BRDF can be included in a bidirectional path tracer. The proposed algorithm samples vertices and wavelengths at the same time and connects vertices to produce a geometrically valid path. If the path contains fluorescent surfaces, the wavelengths of the camera and light sub-path will not match, and the algorithm resets wavelengths along the geometric path to generate a valid path with consistent wavelengths and a well defined measurement contribution for each wavelength combination.

We also present an alternative approach where camera and light sub-paths are sampled independent of wavelengths and then are evaluated for the full spectrum instead of one random wavelength per path segment.

Zusammenfassung

Fluoreszenz ist der Effekt, bei dem ein Photon absorbiert und dann zeitnah mit einer unabhängigen, meist längeren, Wellenlänge wieder emittiert wird. Fluoreszenz tritt in vielen alltäglichen Materialien, wie etwa Papier oder Textilien, auf. Dennoch gab es bisher nur wenig Forschung auf dem Gebiet, Fluoreszenz im Rahmen der photorealistischen Bildsynthese zu simulieren, und alle bisherigen Projekte basieren auf simplen Vorwärts-Pathtracer.

In dieser Arbeit wird untersucht, wie Fluoreszenz in einem spektralen Bidirektionalen Pathtracer umgesetzt werden kann. Wir führen ein einfaches BRDF-Modell für fluoreszierende Oberflächen ein, das mit real gemessenen oder manuell erstellten Spektren parameterisiert werden kann. Daraufhin zeigen wir, wie diese BRDF in einen bidirektionalen Pathtracer integriert werden kann. Der vorgestellte Algorithmus erzeugt zunächst einen Kamera- und einen Lichtpfad unabhängig voneinander, indem er Schritt für Schritt deren Vertices und die Wellenlängen auf den Segmenten dazwischen sammelt. Daraufhin werden die Teilpfade verbunden um einen geometrisch zusammenhängenden Pfad zu erzeugen. Falls der Pfad fluoreszierende Knoten enthält, passen jedoch die Wellenlängen des Kamera- und Lichtpfades nicht zusammen. In diesem Fall modifiziert der Algorithmus einige der Wellenlängen des Pfades, um einen gültigen Lichttransportpfad mit konsistenten Wellenlängen zu erhalten. Zudem präsentieren wir eine Alternative, bei der Kamera- und Lichtpfad wellenunabhängig erzeugt werden, und dann der verbundene Pfad für das komplette Spektrum evaluiert wird, anstatt für jedes Pfadsegment nur eine zufällige Wellenlänge zu betrachten.

Contents

1. Introduction	1
2. Fluorescence	3
2.1. Introduction	3
2.2. Physical Processes	4
2.2.1. Energy Levels: Jablonski Diagram	4
2.2.2. Absorption of a Photon: The Absorption Spectrum	5
2.2.3. Relaxation to the Ground State	7
2.2.3.1. Radiative Decay: The Emission Spectrum	7
2.2.3.2. Nonradiative Decay: The Quantum Yield	8
2.3. Measuring fluorescence	9
3. Previous Work on Fluorescence in Rendering	11
3.1. Fluorescent Light Sources	11
3.2. A Model for Fluorescence and Phosphorescence	12
3.3. Combined Rendering of Polarization and Fluorescence Effects	12
3.4. A Reflectance Model for Diffuse Fluorescent Surfaces	12
3.5. Simulation of Fluorescent Concentrators	13
3.6. Acquisition and Analysis of Bispectral Bidirectional Reflectance and Reradiation Distribution Functions	13
3.7. Bispectral Coding: compressive and high-quality acquisition of fluorescence and reflectance	14
3.8. Paper Reflectivity	14
3.9. Spectral Colour Prediction Model for a Transparent Fluorescent Ink on Paper	14
4. Spectral Bidirectional Path Tracing	15
4.1. Notation	15
4.2. Numerical Integration	17
4.2.1. The Rectangle Rule	17
4.2.2. Monte Carlo Integration	19
4.2.2.1. Importance Sampling	20
4.2.2.2. Multiple Importance Sampling	20
4.3. BRDFs	21
4.3.1. Bispectral Bidirectional Reflectance and Reradiation Distribution Function	22
4.4. Bispectral Rendering Equation	23
4.5. Spectral Measurement Equation	23
4.6. Bidirectional Path Tracing	25
4.6.1. Path Sampling	25
4.6.2. Forward Path tracing	27
4.6.3. BPT	27

4.7. Integration by Discretization of Wavelength Space	29
4.7.1. Scenes without fluorescence	31
4.7.2. Scenes with fluorescence	31
4.8. Conversion from Spectra to RGB	31
5. A diffuse and partly fluorescent BBRRDF	33
5.1. Basics	33
5.1.1. The Reradiation Matrix	33
5.1.2. Assumptions	34
5.1.3. Parameterization	34
5.1.4. Physically reasonable parameters	35
5.2. The BBRRDF	37
5.2.1. For $\lambda_{in} \neq \lambda_{out}$	37
5.2.2. For $\lambda_{in} = \lambda_{out}$	37
5.2.3. Full BBRRDF	37
5.2.4. Formulation as a reradiation matrix	38
5.2.5. Energy Conservation	38
5.3. Sampling new wavelengths	39
5.3.1. Probability of a fluorescent interaction	40
5.3.1.1. Light paths	40
5.3.1.2. Camera paths	40
5.3.2. Probability of the new wavelength	41
6. Fluorescence in a spectral bidirectional path tracer	42
6.1. Sampling a camera path	43
6.2. Sampling a light path	44
6.2.1. Light Source Importance Sampling	45
6.3. Resampling Wavelengths	46
6.3.1. Connecting Paths	46
6.3.1.1. Reset wavelengths on the light sub-path	47
6.3.1.2. Reset wavelengths on the camera sub-path	47
6.3.1.3. Reset all wavelengths	49
6.3.2. Connecting a light vertex to the camera	50
6.3.3. Special Case: 0 light vertices	50
6.3.4. Special Case: 1 light vertex	50
6.4. Throughput	51
6.4.1. Measurement contribution of a path	51
6.4.2. Probability of a complete connected path	52
6.4.2.1. Directional probabilities	52
6.4.2.2. Wavelength probabilities	52
6.4.2.3. Full pdf	56
6.4.2.4. Special cases: 0 camera / light vertices	56
6.4.2.5. Special case: 1 light vertex	56
6.4.3. MIS weights	56
6.4.4. Full throughput	58
6.5. Alternative: Evaluating paths for all wavelengths	59
7. Results	61
7.1. Scene: Room with fluorescent walls	61
7.1.1. Scene Analysis	62
7.1.2. Wavelength importance sampling vs. uniform wavelength sampling .	65

7.2. Scene: Green Buddhas / Spheres	68
7.2.1. Fluorescence can produce high saturations	69
7.2.2. Comparison of different parameters	70
7.3. Additional Measurements	72
8. Conclusion	74
8.1. Limitations	74
8.2. Future Work	75
Bibliography	77
Appendices	82
A. Additional Equations	84
A.1. Separating directional and wavelength integrations in the measurement equation	84

1. Introduction

The goal of photorealistic image synthesis is to create images that look as realistic as possible. In order to achieve that we simulate the physical processes involved in light transport, so that the images do not only look realistic, but are actually based on real, physical behaviour of light. However, physically accurate simulations of light transport that include all possible processes are extremely complex and time consuming. Given the currently available computational power, renderers are restricted to a subset of physical properties and often ignore effects that introduce a big overhead or do not have a big impact on the resulting image.

One effect that is often ignored is fluorescence. Fluorescence causes photons to change their wavelengths when they interact with fluorescent particles. Therefore it can only be simulated in spectral renderers, which simulate light transport for all wavelengths instead of just RGB¹.

Even though fluorescence can be observed in many common materials such as paper or clothes, there has been only little research into fluorescence in the context of rendering before. One reason for that is that the required spectral rendering systems are much more time consuming than a simple RGB renderer. Another one is that measuring fluorescent material samples is much more complex than measuring non-fluorescent materials, which makes the evaluation of the models used to describe fluorescent surfaces more difficult. Aside from those challenges, fluorescent effects are easy to fake, e.g. by using spectra that are not physically plausible, or adding weak light sources to "fluorescent" surfaces. And if only fluorescent light sources are required, there is no need to actually simulate fluorescence.

All previous projects on rendering fluorescence seem to be based on a forward path tracer. While this is a simple approach, it is not very efficient. Bidirectional path tracing proved to be a better alternative to path tracing before, so the goal of this thesis is to make fluorescent surfaces work in a bidirectional path tracer. The central challenge of this originates in the different path creation techniques: Forward path tracing creates one path at a time, which makes it easy to change wavelengths as the path progresses. Bidirectional path tracing creates two paths at the same time and then connects them to form a full path. If one (or both) of those paths changes its wavelength along the way, these connections become invalid. A byproduct of this thesis is a simple, diffuse fluorescent BRDF model.

¹RGB rendering simulates light transport paths without considering the light's wavelength, and evaluates colors only for a red, green and blue channel instead of the full (visible) spectrum.

While it may not be the most accurate surface model possible, it includes all central aspects of fluorescence and can be used to show how fluorescent surfaces can be included in bidirectional path tracing.

We will start with the basics of fluorescence in chapter 2, and summarize some previous work on fluorescence in rendering in chapter 3, although as mentioned before, there has been little work done on fluorescence in the context of photorealistic rendering. Chapter 4 reviews the fundamentals of spectral bidirectional path tracing. Chapter 5 introduces a fluorescent BRDF model, which we use in chapter 6 to explore how fluorescence can work in a bidirectional path tracer. The results are discussed and summarized in chapters 7 and 8, where we also point out some limitations and possible areas of future work.

2. Fluorescence

In this chapter we present common examples of fluorescent materials and the physical processes that cause fluorescence. We discuss which properties of fluorescence are relevant when rendering fluorescence and introduce the fluorescent absorption spectrum, emission spectrum and quantum yield.

2.1. Introduction

Fluorescence is the effect where a molecule or atom absorbs a photon while entering an excited energy state, and then quickly (after about 10^{-8} seconds) relaxes back to its ground energy state while emitting a photon of an independent wavelength. A similar effect is phosphorescence, which usually has a much longer time span between photon absorption and emission (0.1 – 100 seconds) and involves additional energy states. Fluorescence and Phosphorescence are summarized under the term **Photoluminescence**, which describes emission of photons by a substance after being excited by a photon. In general, the emission of photons after excitation by any form of energy except heat is called **Luminescence**. Other examples of luminescence are chemoluminescence (excitation from a chemical reaction) or electroluminescence (excitation from electric current) [Nas01].

Fluorescent substances can be found in many common objects. Fluorescent lamps (see 3.1) have found their way into many buildings, thanks to their low energy consumption. Paper, laundry detergents and textiles are often treated with **fluorescent whitening agents**, which absorb light in the ultraviolet range and emit it as blue light, thus making the surface appear whiter and brighter [Nas01]. In paper, most of those agents are stilbene derivates; in textiles they can also be based on coumarin derivates or other chemicals [Mur96].

Similarly, fluorescent pigments can make safety vests appear in a brighter yellow or orange. This effect is visible during the day because of the sun's UV light (see figure 7.9c), and is also strong during sunset, when the sunlight contains a high portion of UV light. At night there is no more UV light available and the fluorescent effect diminishes, leaving the textile at a less noticeable, darker color.

Less common examples of fluorescent substances are tonic water, which contains quinine, one of the first fluorescent substances discovered, teeth, marks on bank notes, ink in text markers and colored paper. Fluorescent ink, paint and pigments are commercially available in multiple different colors all over the spectrum.

Fluorescent dyes are used in many biological and chemical applications, such as fluorescent labeling or cell imaging. One example is fluorescein, which is presented in figure 2.2.

Although those applications are probably not as frequent in typical scenes in rendering most research on fluorescence originates in biology, chemistry and medicine.

2.2. Physical Processes

This section covers all the physical processes we need to understand before we can include fluorescence in a renderer. We will describe the energy transitions involved in the fluorescent process, and the spectra and constants representing them. In order to render fluorescence in a spectral renderer we need to consider several things:

First we need to know which wavelengths can be absorbed by the fluorescent substance, and how likely a single photon is absorbed (or, if we have a large number of photons with the same wavelength, which fraction of them is absorbed for that specific wavelength). Second, not all absorbed photons will be re-emitted, so we need to know about the probability that a photon is re-emitted (or the fraction of re-emitted photons). Last, we need to know at which wavelengths and in which directions photons can be emitted, that is, we need to know about the distribution of all emitted wavelengths.

2.2.1. Energy Levels: Jablonski Diagram

Each molecule and atom can exist in one of several energy states. The number of possible states depends on the structure of the particle and its electrons.

In molecules each energy state consists of many closely spaced vibrational and rotational energy levels [fsm]. The energy differences between those states and between the levels within a state decrease with increasing energy [JM66]. The energy states and levels as well as the transitions between them are illustrated by the **Jablonski Diagram** [Lak06, 1.2].

Atoms do not have vibrational or rotational energy levels. Atoms that can fluoresce usually emit a photon with the same wavelength as the absorbed photon. This is called resonance fluorescence and can also occur in fluorescent molecules, although those offer a wide spectrum of emitted wavelengths. From now on we only consider fluorescent molecules, since resonance fluorescence alone is not as interesting or relevant for rendering more realistic images and would require a more complex BRDF model.

While nothing interesting is happening the molecules are usually in any vibrational level at the ground singlet¹ state S_0 . At room temperature, the thermal energy is not high enough to cause fluorescent molecules to exist in a higher energy state, and only few molecules occupy higher vibrational levels of the ground state, so most of the molecules will exist at the lowest vibrational level of the ground state S_0 [JM66]. In general, all transitions due to absorption of a photon will start from here.

The Jablonski Diagram also includes higher-energy singlet states S_1 and sometimes S_2 , the so-called "excited states". In those states the electrons have the same spin properties as in S_0 . In some cases an electron can change its spin during a transition and will end up at the triplet states T_0 .

The vibrational levels of each energy state are represented by several thin horizontal lines. The lowest and thicker line in each state represents its lowest vibrational level. The actual number of vibrational energy levels within one energy state is much higher than what could be depicted in such a diagram [JM66], so the diagram only contains a few symbolic lines. The rotational energy levels within the vibrational levels are omitted as well [fee].

¹The term **singlet** refers to the spins of electrons. All electrons either have spin $\frac{1}{2}$ or $-\frac{1}{2}$, and in excited states the excited electron is paired with another electron in the ground state. In a singlet state, the sum of all spins is 0, and the excited electron and its paired ground state electron have opposite spins. In a triplet state the excited electron has the same spin as its paired ground state electron and the sum of spins is not 0. [Lak06]

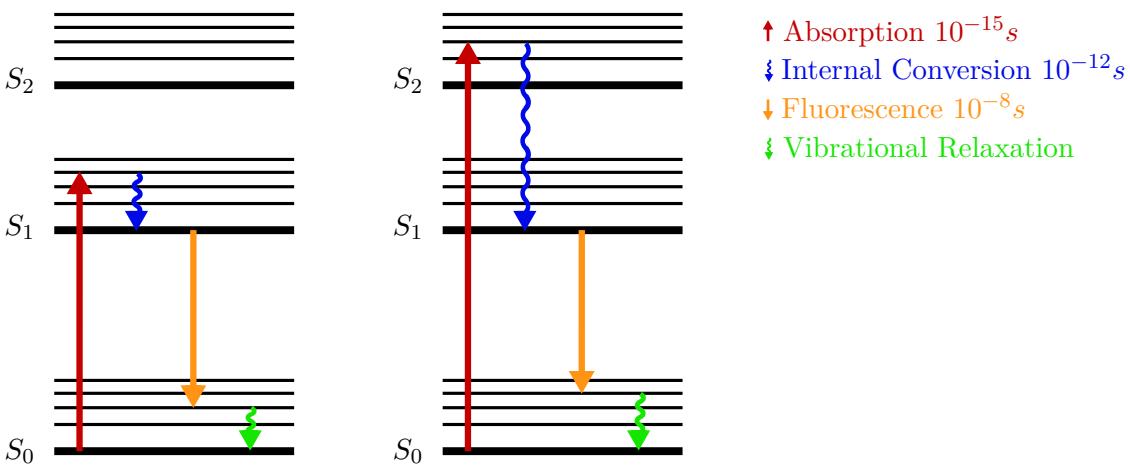


Figure 2.1.: Jablonski Energy Diagram with two examples for fluorescence. In the left diagram, the photon is absorbed to a higher vibrational level in S_1 , in the right case it is absorbed to a vibrational level in S_2 . In both cases the electron relaxes to the lowest level of S_1 before fluorescence can take place.

The arrows symbolize transitions between energy states and levels. Straight arrows represent radiative transitions (transitions involving the absorption or emission of photons), wiggly arrows represent nonradiative transitions.

2.2.2. Absorption of a Photon: The Absorption Spectrum

Whenever a photon "hits" a fluorescent molecule, it can be absorbed, depending on the photon's energy and the molecule's structure. Absorption takes about 10^{-15} seconds and can be considered instantaneous². Photons can only be absorbed if their energy matches the difference of the molecule's current energy level to another available energy level to which it can transition [fee]. In fluorescent molecules there is a broad range of energies that can be absorbed. We will discuss how this relates to the absorption spectrum below.

The range of wavelengths that can potentially cause fluorescence is not much bigger than the range of visible and ultraviolet wavelengths. If a photon's wavelength is below 200nm, its energy is high enough to ionize a molecule. This can cause the molecule to get destroyed or change its form in a way that loses its fluorescent properties. If, on the other hand, the wavelength is above roughly 1000nm, its energy is not high enough to excite any molecule to a level from which it could re-emit a photon. Additionally, fluorophores that can be excited with long wavelengths (above 700nm) need to have particularly close energy states so they can even be excited by such small energies. However, the smaller difference between the states also increases the likelihood of other, nonradiative processes that lose absorbed energy (see 2.2.3.2) and thus decreases the fluorescent intensity [SHE11].

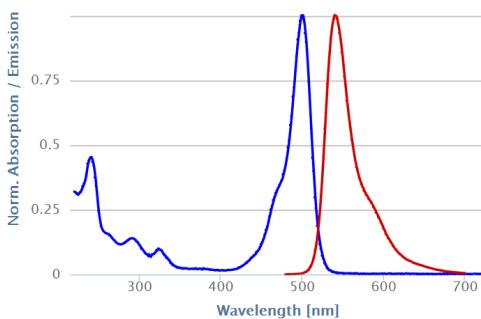
If a photon is not absorbed it simply passes through the molecule. Many fluorescent substances that absorb UV light are actually transparent to visible light (e.g. Quinine), and can be partly transparent to wavelengths where their absorption spectrum has smaller values.

If it is absorbed, an electron transitions to a higher vibrational level in one of the excited states. Typically the transition does not affect the electron's spin, so the excited state is a singlet state as well. No matter if this is a higher level in S_1 or S_2 (wavelengths longer than 200nm do not carry enough energy to enable transitions to higher states [SHE11]), the excited electron soon (10^{-12} seconds) relaxes to the lowest vibrational level of S_1 . This

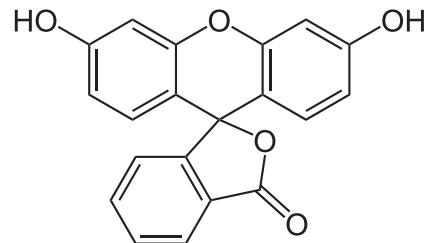
²In fact, electronic transitions are so fast that during the transition the nucleus does not have time to change its position. This is called the Franck Condon Principle.

is called **internal conversion** and is a nonradiative process, which means that no photon is emitted during the relaxation to S_1 . The energy difference is lost as heat or by collision with adjacent molecules. Since the timespan between absorption and fluorescent emission of a photon is in the order of 10^{-8} seconds, we can assume that the relaxation to the lowest vibrational energy level of S_1 is always completed before a photon can be emitted. Hence, all subsequent processes start at this lowest level on S_1 , and the emission spectrum (2.2.3.1) is generally independent of the absorbed wavelength [Lak06, 1.3.2].

Depending on the distribution of energy levels some wavelengths are more likely to be absorbed than others [fee]. This information can be measured by illuminating a fluorescent sample with varying wavelengths and measuring the emitted energy at one fixed wavelength (typically the emission maximum). Measuring the energy at only one fixed emitted wavelength is sufficient because the distribution of emitted wavelengths is independent of the absorbed wavelength - one does not need to measure the absorption spectrum for multiple emitted wavelengths; those spectra will all have the same shape and only differ by a scaling factor. The plot of relative emitted energy for each absorbed wavelength is called the **absorption spectrum**. One example is shown in figure 2.2a. Absorption spectra (as well as emission spectra, see 2.2.3.1) are available online for hundreds of different substances (e.g. by the TU Graz [tug] or commercial websites [the]). In most cases those spectra are not provided as absolute values, but scaled to 1, 100% or just arbitrary units. The probability for absorbing a photon of a certain wavelength is proportional to the value of the absorption spectrum at this wavelength.



(a) Absorption (blue) and Emission spectrum (red) of fluorescein, solved in EtOH [tug, substance 461].



(b) Structural formula of fluorescein.



(c) Fluorescein as powder.



(d) Fluorescein solved in water.

Source: <http://www.instructables.com/id/How-to-Make-Fluorescein-from-Highlighter-Markers/>

Figure 2.2.: Fluorescein is a common fluorescent chemical. It is available as a red powder and becomes fluorescent once it is solved in water. Its quantum yield depends on the solvent, but can get very high (95% or above).

2.2.3. Relaxation to the Ground State

Once the electron arrives at the lowest vibrational level of S_1 it stays there for some time before the energy is converted in one of several ways. We categorize the possible processes into radiative decay (turning the energy into a photon) and nonradiative decay (losing the energy without emitting a photon). [JM66] provides more in-depth explanations of the following effects on a molecular level.

2.2.3.1. Radiative Decay: The Emission Spectrum

One option for a molecule in the lowest vibrational level of S_1 is to transition to any vibrational level of the ground state and emit a photon. Usually the electron will return to a higher vibrational energy level within S_0 . The emitted photon's energy will be equal to the energy difference between S_1 and that vibrational level on S_0 . This is **fluorescence**. The fluorescent lifetime (the time the molecule spends in the lowest level of S_1) is in the order of 10^{-8} s, but note that - as with all transitions - the transition from S_1 to S_0 itself happens within 10^{-15} s.

The electron can return to any of the vibrational levels of S_0 with varying probabilities. As a result, emitted photons can have a variety of different wavelengths, and some wavelengths occur more often than others. The distribution of emitted photons is represented by the **emission spectrum**. See figure 2.2a for an example. The emission spectrum can be measured by exciting a fluorescent sample with one fixed wavelength (typically at the absorption maximum) and measuring the emitted energy for all wavelengths. Emission spectra are available online, and - just as the absorption spectra - are usually scaled to 1, 100% or just arbitrary units.

In most cases the emitted wavelength is longer (which means the photon has a lower energy) than the absorbed wavelength, due to the nonradiative loss of energy during the transition from a higher energy level to the lowest level of S_1 . This means that the emission spectrum covers longer wavelengths than the absorption spectrum, and its peak is at a longer wavelength. The **Stokes Shift** is the distance between the absorption and emission spectrum's maximum.

Since every fluorescent emission starts at the lowest level of S_1 the shape of the emission spectrum is independent of the absorbed wavelength. So while the emission spectrum's wavelength range is longer than the absorption spectrum's wavelength range, they often overlap and it is possible (although not likely) for a photon to be re-emitted at a shorter wavelength.

This overlap originates from so-called 0 – 0 - transitions: Those are transitions from the lowest vibrational energy level in S_0 to the lowest vibrational energy level in S_1 and back. This is the only case where both transitions (S_0 to S_1 and S_1 to S_0) involve the same amount of energy. All other absorption transitions require more, all other emission transitions less energy. Due to the multitude of rotational sub-levels, this transition actually covers a small range of wavelengths (or energies) instead of a single specific wavelength, causing the absorption and emission spectrum to overlap for that range of wavelengths [int].

Phosphorescence

Instead of returning to the ground state, the electron can also undergo a spin conversion and transition to the triplet state ("intersystem crossing"), and then relax to S_0 from there while emitting a photon. This is called **Phosphorescence**. Transitions from T_0 to S_0 have a much lower probability than transitions from S_1 to S_0 , so phosphorescence is much slower, usually between 10^{-2} s and 10^2 seconds, sometimes even longer.

An electron can also transition from T_0 back to S_1 and once again has all the options

described here. If it transitions to S_0 while emitting a photon now we have "delayed fluorescence", which we do not consider in our renderer.

Note that rendering phosphorescence is time dependent, whereas fluorescence can be considered instantaneous when rendering for "normal" camera models.

Fluorescence and Phosphorescence are diffuse: when a photon is emitted from fluorescence or phosphorescence it can be emitted into any direction [Lak06].

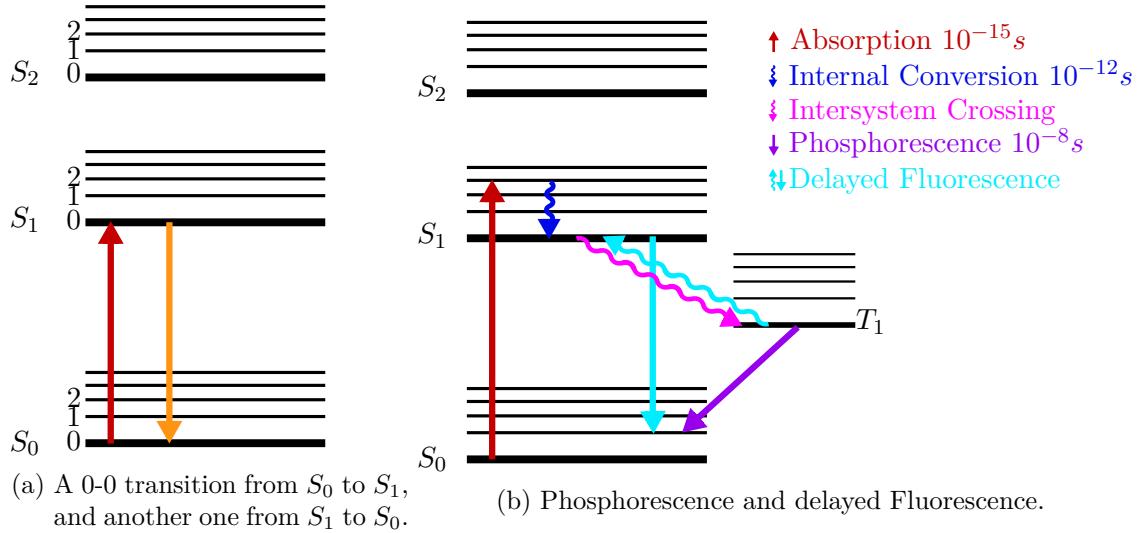


Figure 2.3.: Jablonski Diagrams with 0-0 transitions and intersystem crossing.

2.2.3.2. Nonradiative Decay: The Quantum Yield

While the molecule remains in an excited state it can also lose its excess energy without emitting a photon due to other effects. Such effects are summarized under the term **non-radiative decay**. For rendering, it is not important which of those effects is responsible for losing the energy, we just care about how much energy or how many photons will be re-emitted. This ratio is known as the **Quantum Yield** (or **quantum efficiency**):

$$\Phi = \frac{\text{emitted photons}}{\text{absorbed photons}}. \quad (2.1)$$

The quantum yield can also be described by the rates of radiative (fluorescent) decay k_r and nonradiative decay k_{nr} :

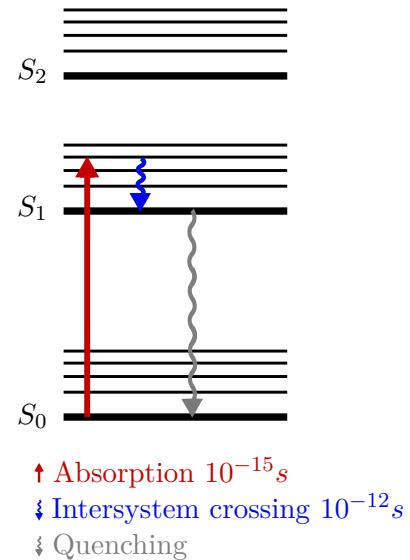
$$\Phi = \frac{k_r}{k_r + k_{nr}}. \quad (2.2)$$

For most substances the quantum yield is independent of the excitation wavelength (one common exception is Quinine), and does not influence the shape of the emission spectrum. It can however depend on other environmental factors, such as temperature, the solvent or other added substances.

Sources of nonradiative decay:

Quenching

Quenching decreases the intensity of fluorescence. It can happen in different forms. For example, an excited fluorescent molecule can collide with another particle, which does not need to be fluorescent, and is called a quencher. Upon contact the energy is directly transferred (meaning no radiation is emitted and then absorbed in the process) to the other molecule. The fluorescent molecule returns to its ground state without emitting a photon. Another example is static quenching: in this case, a fluorescent molecule that is currently not excited forms some non-fluorescent complex with another molecule. Thus the amount of fluorescent molecules decreases which in turn decreases the overall intensity of the fluorescent sample.[Lak06]



Photobleaching

Due to its higher energy an excited fluorescent molecule has a higher chance of participating in a chemical reaction. If it does, its fluorescent properties can be lost irreversibly.

Intersystem Crossing

As mentioned before, sometimes an excited electron can undergo a spin conversion and transition from S_1 to a triplet state T_0 . Not every electron in T_0 will eventually result in phosphorescence or delayed fluorescence, sometimes they will just relax to S_0 without emitting a photon.

Nonradiative relaxation (heat)

An excited electron can relax to S_0 without emitting a photon. In that case its excess energy is dissipated as heat.

2.3. Measuring fluorescence

Measuring fluorescent absorption and emission spectra requires a spectrofluorometer [Gui90] [Lak06]. A spectrofluorometer usually has a lamp which can produce all wavelengths, and monochromators to pick only one wavelength from the light source's emission spectrum or the fluorescent sample's emission spectrum. The emission spectrum can be measured by setting the monochromator between the light source and the sample to one wavelength, and measuring the emitted energy at all wavelengths by varying the wavelength that can pass through the monochromator between the sample and the camera. The absorption spectrum can be measured by fixing the wavelength that arrives at the camera, and varying the exciting wavelengths.

The path from raw measured data to spectra we can use for rendering contains additional steps, because any measurement instrument usually introduces some error. For example,

the light source used while measuring the absorption spectrum may emit varying intensities over different wavelengths, which will affect the measured absorption. Similarly, the sensor used for measuring the emission spectrum may have different sensitivities to different wavelengths, thus affecting the measured emission. In addition the monochromators may let some wavelengths pass through more efficiently than others. In many fluorescence applications, e.g. if one is only interested in detecting fluorescent substances with the same instrument, those errors can be ignored, but in the case of rendering we need spectra that are corrected for the responses of the instruments.

Since the shape of the absorption and emission spectra are generally independent of each other, it is sufficient to use only one excitation wavelength when measuring the emission spectrum, and to only measure the emitted energy at one wavelength when measuring the absorption spectrum.

There are several ways to measure the quantum yield. The direct way, which does not rely on any previous knowledge, is complex, but has been applied for several substance. The indirect method determines the quantum yield of an unkwnon compound relative to a reference compound (a substance with a known quantum yield). Since the quantum yield depends on multiple environmental factors, such as temperature, the solvent or the presence of other chemicals, one fluorescent compound can have different quantum yields depending on the circumstances.

Most fluorescent applications concern themselves with fluorescent substances in solvents. As a result, most research around measuring fluorescence focuses on the fluroescent spectra and the quantum yield. However, if we want to render fluorescent surfaces, we would also like to know about additional reflectance properties of the surface structure. The process that causes fluorescence itself is diffuse, but real-life materials rarely consist of fluorescent molecules only, so other non-fluorescent molecules as well as the overall surface structure may result in a surface that is not purely diffuse. So in addition to measuring the absorption and emission spectra it might be interesting to measure how the emitted intensities vary over different incident and exitant directions.

Measuring all those things at the same time is much more complex than measuring fluorescence or a non-fluorescent spectral BRDF on their own, which is one of the reasons fluorescence is still rare in photorealistic image synthesis. Still, some work has been done in this aera. For example, [HHA⁺10] explores how to measure a fully data-driven bispectral BRDF from a fluorescent surface sample, and [SBCD14] uses the redundancies of fluorescence and reflectance to reduce the complexity of measuring such surfaces. In contrast to the data-driven approaches, [WWLP06] derives a simple analytical model, which only requires measurements of the fluorescent spectra, for fluorescent surfaces with a non-fluorescent, specular component, although their model needs to be improved further before it can match real surfaces.

Since the focus of this thesis is on including fluorescence in a bidirectional path tracer, we ignored angle dependencies as well as media, and restricted our fluorescent materials to perfectly diffuse solid opaque surfaces.

3. Previous Work on Fluorescence in Rendering

Although fluorescence appears in many commonly used materials, so far there has been only little research on fluorescence in the context of photorealistic rendering. Projects on fluorescence in the context of rendering cover a more extensive version of the rendering equation that includes fluorescence, studies of the qualitative effects that a fluorescent BRDF model should have, and measuring fully data-driven fluorescent BRDFs.

However, none of those projects used bidirectional path tracing, and in most examples the fluorescent materials were isolated in simple scenes - The focus of all previous work was either on replicating realistic fluorescent behaviour or on measuring the spectral properties of fluorescent surfaces or scenes, and not on efficiently rendering those effects. This thesis builds on previously developed models and aims to render fluorescence more efficiently by adapting a bidirectional path tracer for fluorescence.

Some work that might be relevant to rendering fluorescence, although it has not been done with rendering in mind, are papers on fluorescence of paper and ink. Those projects mainly focus on predicting the appearance and overall reflectivity of fluorescent paper and only consider local models, but many of their results could potentially be used to develop more realistic fluorescent surface models for paper-like materials that can then be applied in photorealistic rendering.

3.1. Fluorescent Light Sources

Fluorescent light sources have become common since they are more energy conserving than incandescent lamps¹. They use electricity to excite a gas (mercury vapor) which in return emits UV light. This light is absorbed by a fluorescent phosphor coating which emits visible light [flu][Nas01]. Scenes with fluorescent light sources, but no other fluorescent surfaces, can actually be rendered without specifically simulating fluorescence. Instead, the fluorescent emission spectrum of the coating material can be used directly as the emission spectrum of an area light source in the scene. A more accurate model would not only require the simulation of fluorescence, but also using an emitting medium instead of an area light source, but would probably result in the same illumination, since visible light is only emitted by the planar coating of the light.

¹incandescent lamps produce light by heating a wire enough to make it glow.

3.2. A Model for Fluorescence and Phosphorescence

[Gla95] derives an extended version of the rendering equation, the **full radiance equation**, which includes fluorescence. He noted that the re-emission of a photon due to fluorescence has no directional character, in other words, that fluorescent interactions are always diffuse.

In their renderer, fluorescent surfaces are parameterized by a reradiation matrix. Its entries represent the transfer of energy from one wavelength to another. Their implementation relies on the assumption that light is always re-emitted at longer wavelengths, which allowed their raytracer to render in wavelength-independent passes, ordered from short to long wavelengths, and accumulate energies for longer wavelengths. This might have been due to limitations around the time the paper was published. The error from this assumption is probably quite small, since in most cases the absorption and emission spectra only overlap very little, and may have been worth the more convenient code. The (bidirectional) path tracers we use work differently - each pixel and sample per pixel uses its own individual random wavelength - so we are not restricted by any such assumption.

[Gla95] does not mention how the matrix' entries relate to the physical processes responsible for fluorescence. Instead they were modeled empirically.

3.3. Combined Rendering of Polarization and Fluorescence Effects

[WTP01] presents a framework that can handle both polarization and fluorescence at the same time. As in [Gla95], fluorescent materials are represented by the reradiation matrix. They note that the manual design of those matrices is hard, and instead use measured data provided by Labsphere Inc.

They also assume that light is always re-emitted at longer wavelengths, which is a generalization that does not hold with many fluorescent substances.

3.4. A Reflectance Model for Diffuse Fluorescent Surfaces

[WWLP06] investigated the qualitative properties of diffuse fluorescent surfaces. As mentioned before, fluorescent interactions themselves are diffuse. However, most real-life objects, including fluorescent paint and dyes, consist of different types of molecules, not all of which are fluorescent. Plus, most ordinary objects are not perfectly diffuse, but also have a specular component that increases with a flatter angle of incident and exitant light.

They observed the reflection of light off a stock orange cardboard. They noticed that light that does not interact with fluorescent pigments but is merely reflected by the surface will keep its original wavelength. The amount of light that did not change its wavelength increased with smaller (flatter) incident angles.

Based on the fact that fluorescent interactions are diffuse, they decided to split their BRDF model into a purely diffuse component, which is parameterized by the reradiation matrix and includes fluorescence, and a non-fluorescent semi-glossy specular component.

Since no measurements of their cardboard samples were available, they focused on achieving qualitatively similar results instead of a complete correct model. They investigated a Phong-Lambert-Model and a layered microfacet model. The Phong-Lambert-BRDF did not reproduce the observed properties, partly due to the round cosine lobe of the phong model. The images rendered with the microfacet layer BRDF corresponded well to the

photographs taken of the real fluorescent cardboard. They concluded that while the microfacet layer BRDF qualitatively resembled their observations, the hard split between diffuse and specular might be too simple to fully and accurately model all properties of fluorescent surfaces, and that more work needs to be done on fine-tuning their parameters and matching paint samples with rendered images.

Our fluorescent BRDF (see 5) is a purely diffuse BRDF without a specular non-fluorescent component. We restricted our model to be purely diffuse because the goal of this work was to make fluorescence work in a bidirectional path tracer, so we did not have enough time to implement or develop a more realistic model.

3.5. Simulation of Fluorescent Concentrators

Fluorescent concentrators are used in solar cells. They are filled with a fluorescent dye or other material that absorbs sunlight and emits it at longer wavelengths. Due to the internal structure and refractive indices of the concentrator, most of the longer-wavelength light is trapped inside and can be converted to energy by underlying solar cells.

[BHD⁺08] use photon tracing to predict the efficiency of different setups for fluorescent concentrators. Contrary to the other projects presented here, their model used a fluorescent medium instead of a surface, since their goal was to simulate how photons move through the concentrator. They include reflection and refraction at the concentrator's surface as well as scattering and fluorescence inside. The fluorescent medium is parameterized by the fluorescent material's absorption and emission spectrum and quantum yield. The path segment lengths inside the fluorescent medium are determined using the Lambert-Beer-Law, which describes the distance between fluorescent interactions in a solvent or gas, depending on the wavelength. Thus the resulting simulation is very realistic and yields good predictions of the different spectra found at real concentrators.

[BHD⁺08] also uses Monte Carlo integration to integrate over wavelengths, as we do in chapters 6.2 - 6.4. In particular, inside the fluorescent medium, they first use the Lambert-Beer law to sample the length of the next path segment. If the segment ends inside the medium, the photon can be absorbed by the fluorescent dye or the non-fluorescent PMMA (acrylic glass) of which the concentrator is made. If the photon is absorbed by a dye, the quantum yield is used as the probability of re-emitting the photon; in case of re-emission, the new wavelength is sampled from the emission spectrum. This resembles our procedure: whenever a fluorescent surface is hit, we first decide whether or not to have a fluorescent interaction, and then sample a new wavelength from the absorption or emission spectrum, depending on whether we are tracing a path from the camera or from the light source.

3.6. Acquisition and Analysis of Bispectral Bidirectional Reflectance and Reradiation Distribution Functions

[HHA⁺10] demonstrates how to measure reflectance and reradiation values from fluorescent surface samples.

One important concept of this work is the **bispectral bidirectional reflectance and reradiation distribution function** (bispectral BRDF, see 4.3.1). It is an extension of the BRDF that includes incident and exitant wavelengths ("bispectral"), as well as the fact that photons can be reflected at their own wavelength or absorbed and re-emitted at an independent wavelength ("reradiation").

A special focus lies on capturing the reradiation matrix for different incident and exitant angles. The result is a completely data-driven surface model (as opposed to the simplified analytical BBRRDF used by [WWLP06] and the model we introduce in chapter 5).

3.7. Bispectral Coding: compressive and high-quality acquisition of fluorescence and reflectance

[SBCD14] is concerned with measuring fluorescentce in scenes with a focus on using sparse coded measurements and later reconstructing the full reradiation matrix. They note that the rows within the fluorescent reradiation matrix (a matrix whose entries represent the light transported from incident to exitant wavelengths) are linear dependent, which allows them to formulate the fluorescent component of the full reradiation as a low-rank matrix. In addition, they note that while the non-fluorescent reflectance has some overlap with the fluorescent component in the reradiation matrix, it does not modify wavelengths and is therefore sparse as well. Those redundancies within the matrix allow them to use sparse measurements and reconstructing the reradiation matrices in high resolutions, which reduces the complexity of measuring fluorescent surfaces.

While they only applied their reconstruction algorithm for computer vision tasks for images of full scenes (photorealistic relighting and segmentation), it yields more insights into the behaviour of common fluorescent surfaces. In addition, it might be possible to apply their approach to efficiently capture (approximate) fluorescent BBRRDFs.

3.8. Paper Reflectivity

Several people investigated the fluorescence of paper [CAE10], [CE09]. However, their works were focused on predicting and modeling the overall reflectivity of paper and not directly in the context of rendering.

[Mur96] explains the chemical properties of fluorescent whitening agents used to dye paper, and how those dyes can be applied to paper. While this work has nothing to do with rendering, it presents a good explanation on how fluorescent whitening agents work on paper and provides some valuable background information.

3.9. Spectral Colour Prediction Model for a Transparent Fluorescent Ink on Paper

[EDH98] researches the fluorescence of ink printed on paper. It is not presented in the context of rendering; instead, they aim to predict the spectrum of fluorescent inks. They measure the full reradiation matrix of the fluorescent ink using two monochromators. One nice feature of this paper is that they actually base the reradiation matrix on the fluorescent absorption and emission spectrum, the quantum yield and the fluorescent extinction coefficient, which can be found in media and is modeled by a concentration constant in our fluorescent surface model. However, they ignore light being re-emitted at a shorter wavelength, which should be considered in a complete model.

Their model consists of a transparent coating on a thin absorbing ink layer (thus allowing them to ignore the nonlinearity of fluorescence in fluids with high concentrations of fluorescent molecules), and includes light going down through the layer, up the layer and interactions within it. As a result, they were able to predict the reflectivity of fluorescent ink on paper, which could serve as a basis for developing more realistic BRDFs of paper and ink that can be used in photorealistic rendering.

4. Spectral Bidirectional Path Tracing

This chapter provides a short review of bidirectional path tracing with a special focus on how to include wavelengths to the equations originally proposed by [Vea98]. We start with the mathematical basics for Monte Carlo integration and multiple importance sampling, introduce the bispectral BRDF (which is basically a BRDF that also depends on the incident and exitant wavelength), the Bispectral Rendering Equation and finally spectral Bidirectional Path Tracing.

The goal of this chapter is to revise bidirectional path tracing in non-spectral rendering as well as spectral rendering without fluorescence, and to introduce our notation. Chapter 6 will go into more detail on how a bidirectional path tracer can be extended to handle fluorescent surfaces.

4.1. Notation

Ω	Hemisphere over a surface point; Set of all directions
$\omega \in \Omega$	Direction
θ	Angle between ω and the surface normal
\mathcal{M}	Surface area of a scene; set of all surface points x
dA	Surface area measure
$d\omega$	Solid angle measure
$d\omega^\perp$	Projected solid angle measure
$V(x \leftrightarrow x')$	Visibility term. Either 1 or 0.
$G(x \leftrightarrow x')$	Geometry term. Includes V.
Λ	Set of all wavelengths
$\lambda_{in}, \lambda_{out}$	Incident and exitant wavelength
$\omega_{in}, \omega_{out}$	Incident and exitant direction

We will use the incident wavelength λ_{in} and angle ω_{in} for light arriving from the light source (maybe after some interactions with other surfaces between the light source and the current vertex), and the exitant wavelength λ_{out} and angle ω_{out} for light leaving a surface. So when sampling vertices on a camera path, we know the exitant wavelength and direction and want to sample new incident wavelengths and directions; on light paths we know the incident wavelength and direction and want to sample new exitant ones.

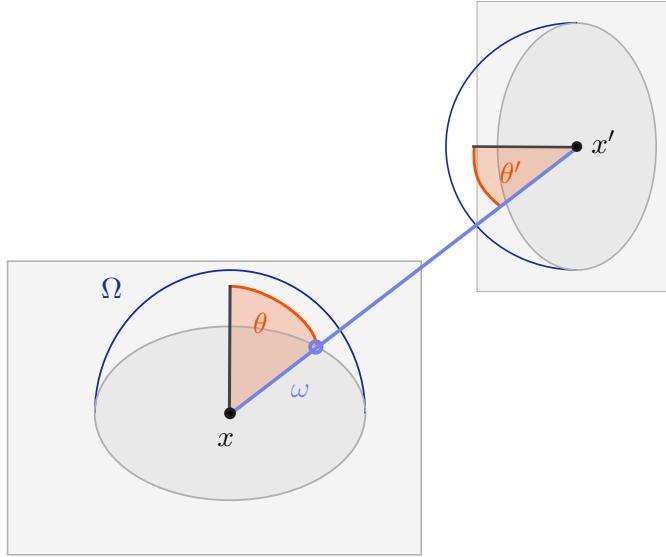


Figure 4.1.: Parameters of the geometry term.

For image synthesis we only need to simulate wavelengths that are either visible or might be converted to visible wavelengths by fluorescent substances. In our renderer we set $\Lambda = [280, 780]\text{nm}$ with visible light in $[380, 780]\text{nm}$ and ultraviolet light in $[280, 380]\text{nm}$. Note that Λ is one-dimensional, whereas Ω and \mathcal{M} are two-dimensional. The exact start and end of the visible wavelength range are contested, but $[280, 780]$ results in a nice number of 100 bins with a thin bin width of 5nm for which we are storing spectrum values.

It is also important to use a conversion from spectra/wavelengths to RGB (or other output values) that considers all simulated wavelengths (or at least those that are considered visible). For example, some conversion methods cut off everything above 700nm, and if they are applied, there is no need to simulate wavelengths longer than 700nm. We used the colormatching functions for the CIE 1931 standard observer (see 4.8), which consider wavelengths up to 780 nm, although the contribution of such long wavelengths is rather small (figure 4.9).

The **geometry term** is defined as

$$G(x \leftrightarrow x') = \frac{\cos\theta \cdot \cos\theta'}{\|x - x'\|} \cdot V(x \leftrightarrow x'), \quad (4.1)$$

where ω is the direction from a point x to another point x' , θ is the angle between ω and the surface normal at x , and θ' is the angle between $-\omega$ and the surface normal at x' . The parameters are depicted in figure 4.1. V is the visibility term and is either 1 or 0, depending on whether or not x and x' are visible from each other. The geometry term can be used to convert between the different measures as follows:

$$d\omega^\perp = \cos\theta \cdot d\omega = G(x \leftrightarrow x') \cdot dA. \quad (4.2)$$

4.2. Numerical Integration

Rendering photorealistic images requires integrating a high-dimensional equation (see 4.30). Solving this equation analytically is practically impossible, or at least extremely time consuming. Instead we employ numerical methods to find an approximate solution, which becomes more accurate with an increasing number of samples. This section will introduce the two numerical methods which we will use later on. First are the rectangle and the midpoint rule, which are perhaps the simplest methods of numerical integration. We will also discuss their limitations and introduce Monte Carlo integration as an alternative.

4.2.1. The Rectangle Rule

The rectangle rule is a numerical integration technique for closed integrals of a one-dimensional function $f : \mathbb{R} \rightarrow \mathbb{R}$ over an interval $(a, b) \subset \mathbb{R}$:

$$\int_a^b f(x)dx. \quad (4.3)$$

It approximates the integral by subdividing its area into N rectangles and adding up their areas. All rectangles have the same width h . The width can be derived from the interval width divided by the number of rectangles N :

$$h = \frac{b - a}{N}. \quad (4.4)$$

The height of each rectangle is the function's value at its left side, i.e. the n -th rectangle's height is $f(x_n)$ with

$$x_n = a + nh \text{ for } n = 0, \dots, N - 1. \quad (4.5)$$

The full rectangle rule is

$$\int_a^b f(x)dx \approx h \sum_{n=0}^{N-1} f(x_n). \quad (4.6)$$

Figure 4.2 shows an example of what the rectangle rule looks like for increasing values of N (or, correspondingly, a decreasing step width h).

The main advantage of this integration technique is that it is easy to understand and implement. It requires a fixed number of steps (or a fixed step width) before it can be applied, which makes it harder to use iteratively. We will apply the rectangle rule in a scenario where we know which number of steps we need, and where we already have the function's values available for all x_n . However, this can only be expected to work well if we have some prior knowledge about f , and if we know that evaluating f for the fixed number of samples will be fast enough and yield a low enough error without requiring further samples.

In its basic form the rectangle rule is only defined for one-dimensional intervals. It can be adapted to work on a two-dimensional, rectangle-shaped integration domain, but gets much more complex and converges much slower with an increasing number of dimensions. So for a larger number of dimensions and more complex integration domains we will use Monte Carlo integration (4.2.2) instead.

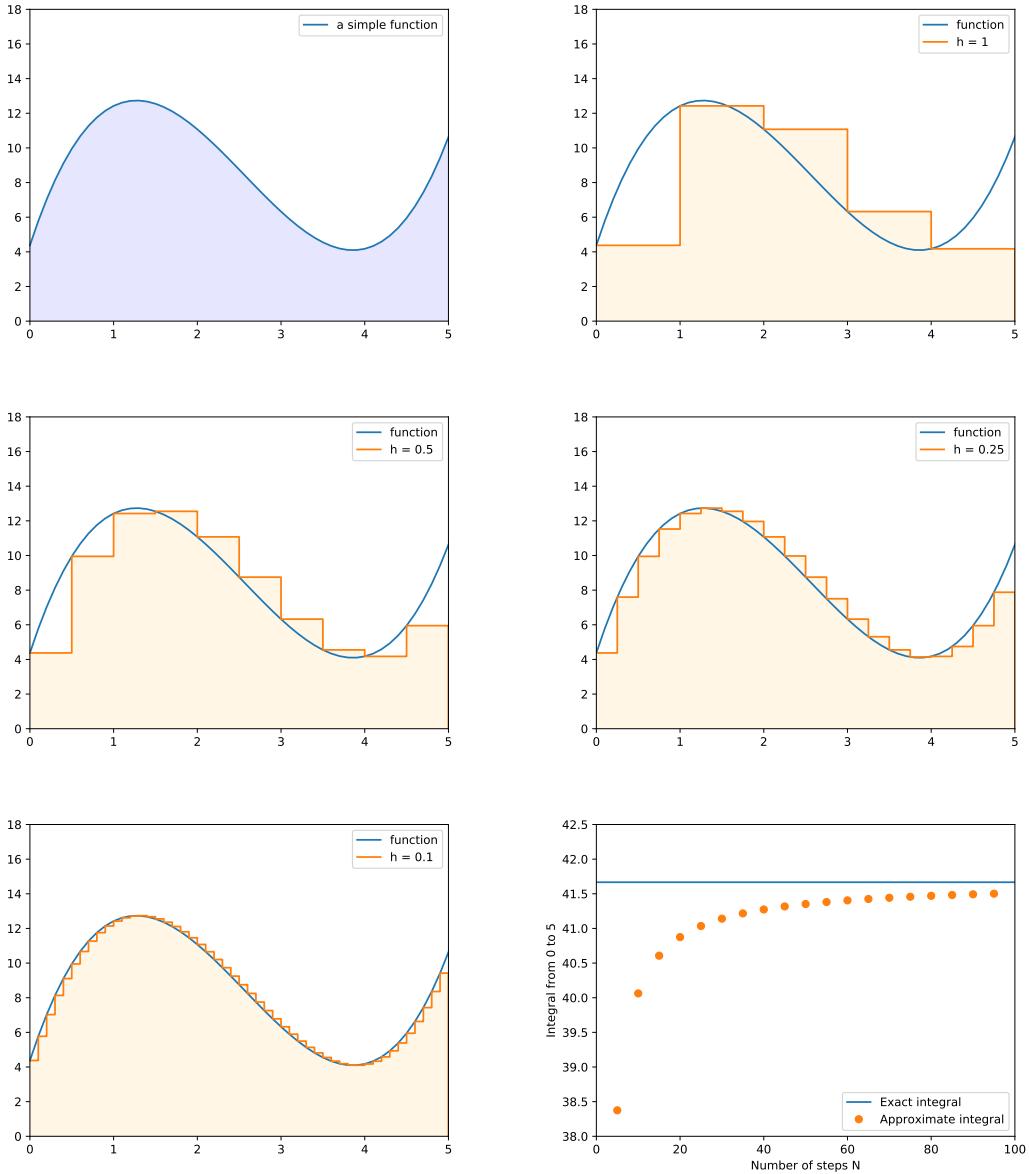


Figure 4.2.: The rectangle rule applied to a simple function with decreasing step width h (corresponding to an increasing number of steps N). The integral of the step functions (the orange area) approaches the exact integral value (the blue area) of the actual function. Each rectangle's height corresponds to the function's value at its left edge.

The Midpoint Rule

The midpoint rule (figure 4.3) is a variation of the rectangle rule. The midpoint rule also subdivides the integral into N rectangles. The only difference is that the rectangle's height is evaluated at the center x'_n of the rectangle instead of at its left side:

$$x'_n = a + nh + \frac{h}{2}. \quad (4.7)$$

Other than that, it has the same advantages and disadvantages as the rectangle rule.

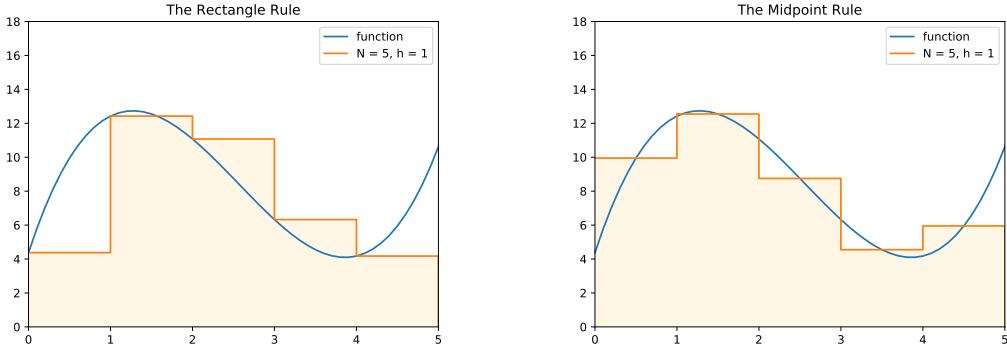


Figure 4.3.: Rectangle Rule vs. Midpoint Rule for same step width

In general, when f is a differentiable function, the midpoint rule is slightly more precise than the rectangle rule. In our particular use case, we do not have a normal function, but measurement values at discrete points, so whether we apply the rectangle or the midpoint rule is simply a question of interpretation (see 4.7).

4.2.2. Monte Carlo Integration

Monte Carlo integration [Vea98] is a numerical integration technique which is particularly well-suited for high-dimensional integrals. The measurement equation 4.30 contains such an integral, and path tracing algorithms (see 4.6) use Monte Carlo integration to solve it.

Assume we have an integral

$$I = \int_{\Omega} f(x) d\mu(x)$$

where Ω is a domain of arbitrary dimensionality, $\mu(x)$ is a measure which is defined over elements of that domain and $f : \Omega \rightarrow \mathbb{R}$.

Given a probability density function (**pdf**) p which we can use to draw random samples X from all of Ω , the **Monte Carlo estimator** F_N approximates the value of I by evaluating the integrand for a finite number N of random samples X_i :

$$F_N = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)}. \quad (4.8)$$

One main advantage of Monte Carlo integration over other integration methods is that it converges to an unbiased solution with a rate of $\mathcal{O}(\frac{1}{\sqrt{N}})$ - independent of the integral's dimensionality.

Opposed to the rectangle and midpoint rule, the Monte Carlo estimator is directly applicable to arbitrarily shaped and dimensional integration domains. And since the function is evaluated at independently selected samples instead of fixed steps, it is easy to iteratively increase the number of samples until the estimator reaches the desired accuracy.

4.2.2.1. Importance Sampling

Importance Sampling is a variance reduction technique for Monte Carlo integration. The idea is to choose a "good" pdf for drawing the random samples for which the integrand is evaluated. The optimal choice would be

$$p^*(x) = c \cdot f(x) \text{ with } c = \frac{1}{\int_{\Omega} f(x) d\mu(x)}. \quad (4.9)$$

With that pdf, the estimator's value would be

$$F_N = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p^*(X_i)} = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{c \cdot f(X_i)} = \int_{\Omega} f(x) d\mu(x) = I \quad (4.10)$$

for any number of random samples, so its variance would be 0. So even a single random sample drawn from p^* would give us the exact value of I . The problem is that in order to get p^* we would need to know the value of I beforehand. So the next best thing is to choose a pdf that at least approximates the integral. When we integrate the rendering equation later on, we do that by picking a pdf that is proportional to some factors within the integrand, which can be evaluated analytically (see 4.6 and 6).

4.2.2.2. Multiple Importance Sampling

Depending on the integrand, there might be several distinct potentially good pdfs. We will also refer to these pdfs as "sampling techniques". Multiple importance sampling (**MIS**) uses a set of weight functions to combine samples from several pdfs in order to make Monte Carlo integration more robust.

[Vea98] introduced two different approaches to MIS: the one-sample model and the multi-sample model. In the one-sample model, one pdf is chosen from all available pdfs, and one sample is drawn from this pdf, for which the integrand is evaluated and then weighted. In the multi-sample model, a number of samples are drawn from each of the pdfs, and all samples are weighted together. Our bidirectional path tracer only uses the multi-sample model, so we do not discuss the one-sample model any further.

The multi-sample model

Say we have n pdfs p_1, \dots, p_n defined over Ω . The multi-sample model also requires us to decide in advance on the number n_i of samples we want to generate from each p_i . Let X_{ij} refer to the j -th sample taken from pdf p_i ($j = 1, \dots, n_i$). Last we need a set of weight functions w_1, \dots, w_n which need to fulfill the following requirements:

$$\sum_{i=1}^n w_i(x) = 1 \text{ whenever } f(x) \neq 0 \quad (4.11)$$

and

$$w_i(x) = 0 \text{ whenever } p_i(x) = 0. \quad (4.12)$$

This also means that for every $X \in \Omega$ with $f(X) > 0$ there has to be at least one pdf such that $p_i(X) > 0$. In other words, it must be possible to sample every X with $f(X) > 0$ from at least one of the n sampling techniques.

All those samples are combined by the **multi-sample estimator** F :

$$F = \sum_{i=1}^n \frac{1}{n_i} \sum_{j=1}^{n_i} w_i(X_{ij}) \frac{f(x_{ij})}{p_i(X_{ij})}. \quad (4.13)$$

In our bidirectional path tracer we used the **balance heuristic** as the weight function:

$$w_i(x) = \frac{n_i p_i(x)}{\sum_{j=1}^n n_j p_j(x)}, \quad (4.14)$$

which is generally a good combination strategy. Other options for the weight functions as well as an analysis of their variance are presented in [Vea98].

4.3. BRDFs

The **bidirectional reflectance distribution function** f_r describes how surfaces reflect light. The definition of the BRDF depends on the Irradiance E

$$E = \frac{d\Phi}{dA}, \quad (4.15)$$

which is the incident flux $d\Phi$ over surface area dA , and the Radiance L

$$L = \frac{d^2\Phi}{dA \cdot \cos\theta \cdot d\omega}, \quad (4.16)$$

which is the flux $d\phi$ per surface area dA per projected solid angle $\cos\theta \cdot d\omega$.

Flux is measured in Watt (W), so the Irradiance and Radiance are measured in Wm^{-2} and $Wm^{-2}sr^{-1}$ respectively.

The BRDF is the ratio of exitant radiance $L_r(x, \omega_{out})$ (leaving a point x in direction ω_{out}) to the incident irradiance $E_i(x, \omega_{in})$ (for a single direction ω_{in}):

$$f_r(\omega_{in}, x, \omega_{out}) = \frac{dL_r(x, \omega_{out})}{dE_i(x, \omega_{in})} = \frac{dL_r(x, \omega_{out})}{L_i(x, \omega_{in}) \cos\theta_{in} d\omega_{in}}. \quad (4.17)$$

This ratio is constant - the exitant radiance is proportional to the incident irradiance, independent of the intensity.

A BRDF is **physically plausible** if it is

not negative: $f_r(\omega_i, x, \omega_o) \geq 0$,

reciprocal: $f_r(\omega_i, x, \omega_o) = f(\omega_o, x, \omega_i)$ and

energy conserving: $\forall x, \omega_i : \int_{\Omega} f_r(\omega_i, x, \omega_o) \cos\theta_o d\omega_o \leq 1$.

4.3.1. Bispectral Bidirectional Reflectance and Reradiation Distribution Function

For real materials the ratio of reflected to incident light depends on the light's wavelength λ . Spectral renderers take that fact into account and use the spectral BRDF

$$f_r(\omega_{in}, x, \lambda, \omega_{out}) = \frac{dL_r(x, \omega_{out}, \lambda)}{dE_i(x, \omega_{in}, \lambda)}. \quad (4.19)$$

A physically plausible spectral BRDF has to fulfill the properties in eq. 4.18 for each λ separately. Note that both the non-spectral BRDF and spectral BRDF have the same unit ($\frac{1}{sr}$).

Fluorescent surfaces are able to change the wavelength of incident light. Their reflectance properties can be described by a **BBRRDF** - the "bispectral bidirectional reflectance and reradiation distribution function". It depends on two wavelength parameters λ_{in} and λ_{out} :

$$f_r(\omega_{in}, \lambda_{in}, x, \lambda_{out}, \omega_{out}) = \frac{d^2 L_r(x, \omega_{out}, \lambda_{out})}{L_i(x, \omega_{in}, \lambda_{in}) \cos\theta_{out} d\omega_{out} d\lambda_{in}}. \quad (4.20)$$

A physically plausible BBRRDF has to be not negative and Helmholtz reciprocal (in terms of swapping ω_{in} and ω_{out} as in eq. 4.18) for each λ separately. For energy conservation¹ we have to consider energy arriving from all wavelengths, and the BBRRDF has to fulfill the following condition:

$$\forall x, \omega_{in}, \lambda_{in} : \int_{\Omega} \int_{\Lambda} f_r(\omega_{in}, \lambda_{in}, x, \lambda_{out}, \omega_{out}) \cos\theta_{out} d\lambda_{out} d\omega_{out} \leq 1. \quad (4.21)$$

Note that the BBRRDF has a different unit ($\frac{1}{sr \cdot nm}$) and is not symmetrical in terms of wavelengths: $f_r(\lambda_{in}, \lambda_{out}) \neq f_r(\lambda_{out}, \lambda_{in})$. A more extensive derivation of the BBRRDF can be found in [Hul10], who also describes how they can be measured from fluorescent surface samples.

Considering that light arriving at some point x' from direction ω_{in} has to come from some other point x'' , and light leaving x' in direction ω_{out} is generally going to hit some other point x , we will also use a different notation for the same bispectral BRRDF:

$$f_r(x'' \xrightarrow{\lambda_{in}} x' \xrightarrow{\lambda_{out}} x) := f_r(\omega_{in}, \lambda_{in}, x, \lambda_{out}, \omega_{out}). \quad (4.22)$$

¹If we wanted to be as precise as possible, at this point we would have to start thinking about whether we are talking about energy or photon count. Photons of long wavelengths actually carry less energy than photons of short wavelengths. Since radiance and irradiance are measured in units of energy (including area and direction), given the definition in eq. 4.20, eq. 4.21 only guarantees conservation of overall energy and does not consider the number of photons involved. A fluorescent BBRRDF that, like real fluorescent materials, mostly converts short wavelength photons to long wavelength photons, would be able to 'create' additional photons at longer exitant wavelengths for short incident wavelengths without violating energy conservation. This does not happen with real fluorescent molecules. The BBRRDF proposed in 5 will avoid this by norming the area of its emission spectrum parameter to 1. This will result in a small loss of energy - which coincides with the physical process of fluorescence - while allowing us to keep all photons if desired.

4.4. Bispectral Rendering Equation

In its simplest version the rendering equation ([Kaj86]) describes how the light L_o leaving a surface point x in one direction ω_o is composed of the light L_e that is emitted in that direction and all incident light L_i that is reflected towards ω_o from all incident directions ω_i :

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega} L_i(x, \omega_i) f_r(\omega_i, x, \omega_o) \cos\theta_i d\omega_i. \quad (4.23)$$

For spectral rendering, we need to consider that $L_o(\omega_o)$ might have different intensities for different wavelengths λ_o . For fluorescent surfaces we further need to consider that light is not only reflected with its current wavelength, but can also change its wavelength at a fluorescent surface. Such surfaces can be represented by a BBRRDF (see 4.3.1). So in addition to integrating over all incident directions we also need to integrate over all incident wavelengths, which results in the **bispectral rendering equation** (see [Hul10]):

$$L_o(x, \omega_o, \lambda_o) = L_e(x, \omega_o, \lambda_o) + \int_{\Omega} \int_{\Lambda} L_i(x, \omega_i, \lambda_i) f_r(\omega_i, \lambda_i, x, \lambda_o, \omega_o) \cos\theta_i d\lambda_i d\omega_i. \quad (4.24)$$

Ω is the set of incident directions measured over solid angle, and Λ is the set of all wavelengths.

Considering that light arriving at some point x' from direction ω_i has to come from some other point x'' , and light leaving x' in direction ω_{out} is generally going to hit some other point x , we can also integrate over light arriving from all surface points \mathcal{M} instead of light arriving from all directions Ω . This change of the integration domain requires changing the integration measure from $d\omega$ to dA (see equation 4.2). The result is the **three-point form** of the bispectral rendering equation:

$$L(x' \xrightarrow{\lambda} x) = L_e(x' \xrightarrow{\lambda} x) + \int_{\mathcal{M}} \int_{\Lambda} L(x'' \xrightarrow{\lambda'} x') f_r(x'' \xrightarrow{\lambda'} x' \xrightarrow{\lambda} x) G(x'' \leftrightarrow x') d\lambda' dA(x''). \quad (4.25)$$

One advantage of this notation is that it no longer contains any directions. This means that we no longer have to distinguish between emitted light at x towards x' , and incident light at x' from x . We can thus substitute the $L(x'' \xrightarrow{\lambda'} x')$ term inside the integral with the rendering equation for $L(x'' \xrightarrow{\lambda'} x')$ and recursively expand the integrand (see eq. 4.27 and 4.38).

4.5. Spectral Measurement Equation

When rendering photorealistic images, we are interested in measuring the intensitiy of individual pixel values for a hypothetical camera. The intensity I of a pixel j is described by the **measurement equation**

$$I_j = \int_{\mathcal{M} \times \Lambda \times \mathcal{M}} W_e^{(j)}(x' \xrightarrow{\lambda} x) L(x' \xrightarrow{\lambda} x) G(x' \leftrightarrow x) dA(x') d\lambda dA(x). \quad (4.26)$$

$W_e^{(j)}$ is the sensor responsivity function. It describes the response of a hypothetical pixel sensor to incident light, depending on where on the sensor the light arrives, from which

direction and with which wavelength. W is defined for all scene surface points and should be 0 for all points that are not part of the sensor; in that case the integral over the first \mathcal{M} measured by $dA(x)$ corresponds to integrating over the sensor's surface area.

The integration over Λ and the other \mathcal{M} corresponds to collecting incident light of all wavelengths λ from all surface points x' in the scene. Note that since G contains a visibility term the integrand becomes 0 for all x' that are not visible from x .

In this form the measurement equation contains a recursive term $L(x' \xrightarrow{\lambda} x)$. This term can be replaced by the rendering equation 4.25, which in turn introduces another $L(x'' \xrightarrow{\lambda'} x')$:

$$\begin{aligned}
I_j &= \int_{\mathcal{M} \times \Lambda \times \mathcal{M}} W_e^{(j)}(x' \xrightarrow{\lambda} x) L(x' \xrightarrow{\lambda} x) G(x' \leftrightarrow x) dA(x') d\lambda dA(x) \\
&= \int_{\mathcal{M} \times \Lambda \times \mathcal{M}} W_e^{(j)}(x' \xrightarrow{\lambda} x) \\
&\quad \left(L_e(x' \xrightarrow{\lambda} x) + \int_{\Lambda \times \mathcal{M}} L(x'' \xrightarrow{\lambda'} x') f_r(x'' \xrightarrow{\lambda'} x' \xrightarrow{\lambda} x) G(x'' \leftrightarrow x') dA(x'') d\lambda' \right) \\
&\quad G(x' \leftrightarrow x) dA(x') d\lambda dA(x) \\
&= \int_{\mathcal{M} \times \Lambda \times \mathcal{M}} W_e^{(j)}(x' \xrightarrow{\lambda} x) L_e(x' \xrightarrow{\lambda} x) G(x' \leftrightarrow x) dA(x') d\lambda dA(x) \\
&+ \int_{(\mathcal{M} \times \Lambda)^2 \times \mathcal{M}} W_e^{(j)}(x' \xrightarrow{\lambda} x) \\
&\quad L(x'' \xrightarrow{\lambda'} x') f_r(x'' \xrightarrow{\lambda'} x' \xrightarrow{\lambda} x) G(x'' \leftrightarrow x') dA(x'') d\lambda' \\
&\quad G(x' \leftrightarrow x) dA(x') d\lambda dA(x).
\end{aligned} \tag{4.27}$$

If we keep substituting the L term with the rendering equation, we can write the measurement equation as an infinite sum:

$$\begin{aligned}
I_j &= \sum_{k=2}^{\infty} \int_{(\mathcal{M} \times \Lambda)^{k-1} \times \mathcal{M}} L_e(x_{k-1} \xrightarrow{\lambda_{k-2}} x_{k-2}) G(x_{k-1} \leftrightarrow x_{k-2}) \\
&\quad \cdot \prod_{i=1}^{k-2} \left(f_r(x_{i+1} \xrightarrow{\lambda_i} x_i \xrightarrow{\lambda_{i-1}} x_{i-1}) G(x_i \leftrightarrow x_{i-1}) \right) \\
&\quad \cdot W_e^{(j)}(x_1 \xrightarrow{\lambda_0} x_0) dA(x_{k-1}) d\lambda_{k-2} dA(x_{k-2}) \cdots d\lambda_0 dA(x_0).
\end{aligned} \tag{4.28}$$

In this form there are no more recursive terms, the integrals have finite dimensions, and their integrands are defined for each path length $k - 1$. A path of length $k - 1$ has k vertices $x_0 \dots x_{k-1}$ and $k - 1$ wavelengths transported between adjacent vertices. λ_i is the wavelength transported from x_{i+1} to x_i . We write such a path as

$$\mathbf{x}\lambda = x_0\lambda_0x_1\dots\lambda_{k-2}x_{k-1}.$$

The integrand in eq. 4.38 is defined for each path $\mathbf{x}\lambda$ with at least 2 vertices. It is called the **measurement contribution function** f_j :

$$\begin{aligned} f_j(\mathbf{x}\lambda) = & L_e(x_{k-1} \xrightarrow{\lambda_{k-2}} x_{k-2}) G(x_{k-1} \leftrightarrow x_{k-2}) \\ & \cdot \prod_{i=1}^{k-2} f_r(x_{i+1} \xrightarrow{\lambda_i} x_i \xrightarrow{\lambda_{i-1}} x_{i-1}) G(x_i \leftrightarrow x_{i-1}) \\ & \cdot W_e^{(j)}(x_1 \xrightarrow{\lambda_0} x_0). \end{aligned} \quad (4.29)$$

Now we can rewrite the measurement equation in a path integral formulation:

$$I_j = \int_{\Omega_\Lambda} f_j(\mathbf{x}\lambda) d\mu_\Lambda(\mathbf{x}\lambda). \quad (4.30)$$

with

$$\Omega_\Lambda = \bigcup_{k=2}^{\infty} (\mathcal{M} \times \Lambda)^{k-1} \times \mathcal{M} \quad (4.31)$$

as the **path-wavelength-space** (the union of all sets of paths with $k = 2, \dots$ vertices), and

$$\mu_\Lambda(\mathbf{x}\lambda) = dA(x_0)d\lambda_0dA(x_1)\cdots d\lambda_{k-2}dA(x_{k-1}) \quad (4.32)$$

as the **area-wavelength-product measure** defined on each set of paths with k vertices.

A more extensive derivation of the path integral formulation of the measurement equation (but without wavelength dependencies) can be found in [Vea98].

A note on the sensor sensitivity function

In our renderer, and probably in many other renderers, we want the renderer to produce an image. This image does not store the full measured spectrum at each pixel, but instead stores a much smaller set of values that can be displayed by a monitor. In other words, at some point the measured spectra or wavelengths have to be converted to some color space, e.g. XYZ or RGB. This means that each pixel measures three intensities, and the sensor sensitivity includes the conversion of wavelengths to an X, Y or Z value (or R, G or B value, or whatever color space is used). This conversion usually includes an integration over the measured spectrum. We can solve this numerically, e.g. with Monte Carlo integration (sampling a random wavelength per path, see 4.6.1 / path segment, see 5.3), or with the rectangle or midpoint rule (evaluating a path for a set of equidistant wavelengths 4.7).

4.6. Bidirectional Path Tracing

4.6.1. Path Sampling

We can integrate the measurement equation in the path integral form by using Monte Carlo integration and drawing samples from the path-wavelength-space. A path $x_0\lambda_0\dots x_{k-1}$ will only contribute to the Monte Carlo estimator if x_0 is part of the current pixel, x_{k-1} emits light and all x_i and x_{i+1} are visible to each other. However, one cannot simply sample a complete path in one sampling step. Instead, we use **local path sampling**,

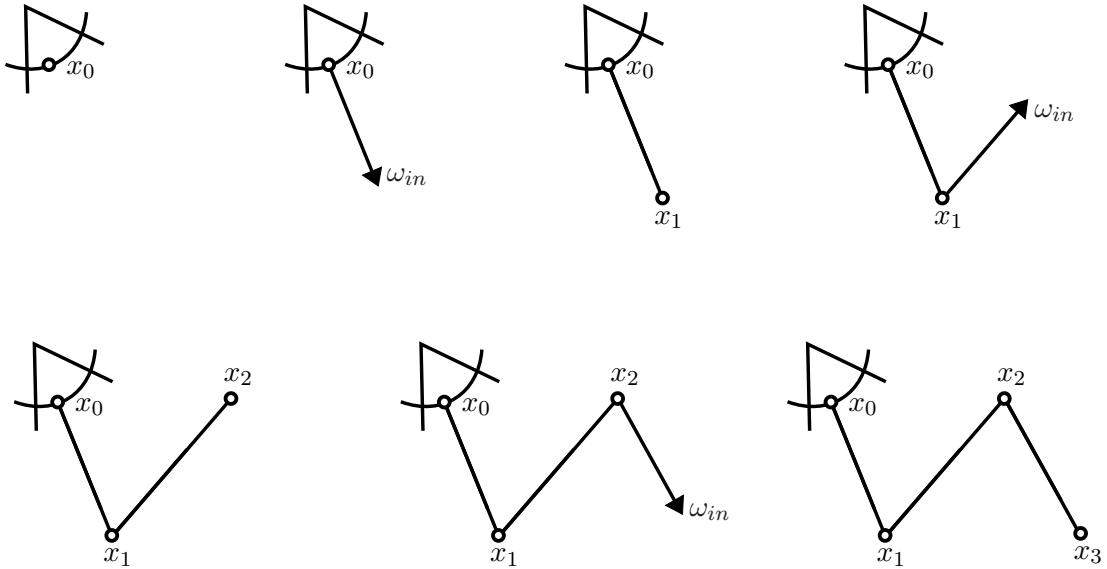


Figure 4.4.: Sampling a camera path (without wavelengths).

which means that we sample a start vertex and then consecutively sample new vertices and wavelengths step by step.

An individual vertex can be sampled from some distribution measured over surface area (e.g. sampling a vertex on a camera pixel, or sampling a vertex from a light source), or by starting at a given vertex and tracing a ray into a random direction, and letting the next vertex be the ray's first intersection with a surface in the scene. Additionally the end vertices of two sub-paths can be connected to form a longer path without introducing new vertices.

Since we know that the first vertex has to be on the camera, paths are usually started by sampling a vertex on the current pixel. Next we sample a direction in which the path should be continued. Tracing a ray along this direction through the scene and determining its first intersection with a surface gives us the next vertex. If this vertex is not perfectly transparent, we then need to sample a new direction for our path (e.g. importance sampling a new direction from its BRDF), trace a ray in that direction, compute the next vertex and so on. Figure 4.4 shows an example.

For spectral path tracing we also need to sample an initial wavelength that is transported along a path. This can be done while the first vertex is sampled. For spectral path tracing with fluorescence we might also need to sample new wavelengths whenever we add a fluorescent vertex to the path. Chapter 5.3 discusses how to sample new wavelengths and chapter 6 explores how we can guarantee that the resulting paths have consistent wavelength combinations and that we are able to sample all possible wavelength combinations.

For some surface types the (probability of the) new direction sampled at a vertex may depend on the wavelength. For other surfaces, the direction and wavelength are independent. Additionally, when a scene contains participating media, the segment lengths inside a medium before hitting the next particle may depend on the wavelength.

This thesis only considers surfaces where the sampling of directions is independent of wavelengths, and we ignore participating media.

4.6.2. Forward Path tracing

Forward path tracing only samples paths starting at the camera. Whenever a path hits a light source, its contribution is added to the estimator. Since randomly hitting a light source is rather unlikely and images produced by this technique alone are very noisy, forward path tracing is usually extended with next event estimation (**NEE**). This means that every time a vertex is added to the path, a random light vertex is sampled from all the light sources in the scene. We can then create a complete light transport path by connecting the new vertex to the light vertex (figure 4.5), and add its contribution to the estimator.

If a path tracer uses NEE it now has two strategies for producing a given path - randomly sampling all the way to the light source, or sampling up until the last vertex before the light source, and explicitly connecting to a light vertex. Samples from those two strategies are combined using multiple importance sampling.

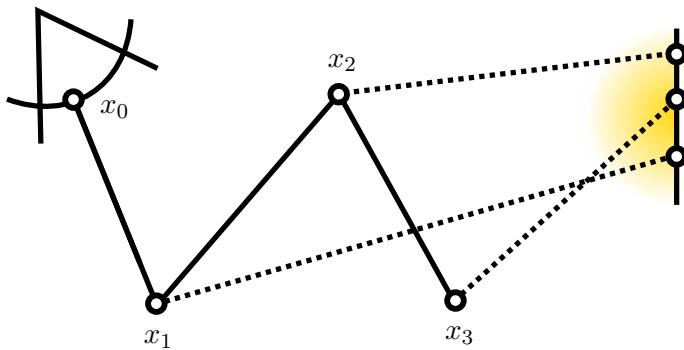


Figure 4.5.: Next Event Estimation: Each vertex on the camera path is connected to a random vertex on a light source.

4.6.3. BPT

The idea of bidirectional path tracing ([Vea98]) is to construct two sub-paths - one starting at the camera, and another one starting at the light source - and then connect them. The advantage of this is that if we already have a camera and light sub-path with s and t vertices, not only can we connect their end vertices to construct a full path with $s + t$ vertices, we can also construct a multitude of shorter paths by connecting each vertex of the camera path to each vertex of the light path (figure 4.6). NEE is a special case of this, where each vertex of the camera path is connected to a light path with one vertex.

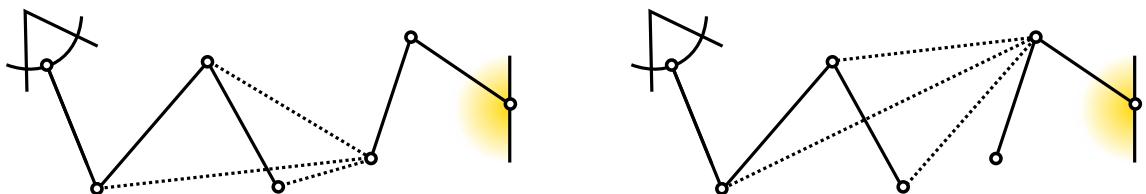


Figure 4.6.: Possible connections (dotted lines) between vertices of a camera and light sub-path (continuous lines). This figure does not show connections to the first light path vertex, since those are usually done in the form of NEE, with an independent light vertex for each camera vertex (figure 4.5). It also excludes direct connections from the camera to light path vertices, as well as paths randomly hitting the camera or light source.

As a result, the average time spent on constructing a path is shorter than if each path was sampled individually. Depending on the scene, paths of the same length may be sampled differently well by different light/camera sub-path length combinations. Combining all those techniques results in a much more robust algorithm than relying on one technique to sample all paths.

We now have $k+1$ techniques to sample a path with k vertices - anything from sampling a camera path with k vertices that implicitly hits the light source, to sampling a light path with k vertices that implicitly hits the camera², or sampling a camera path with s vertices and connecting it to a light path with $k-s$ vertices (figure 4.7). So whenever we actually sample a path with one of those techniques, we need to use multiple importance sampling to weight its contribution.

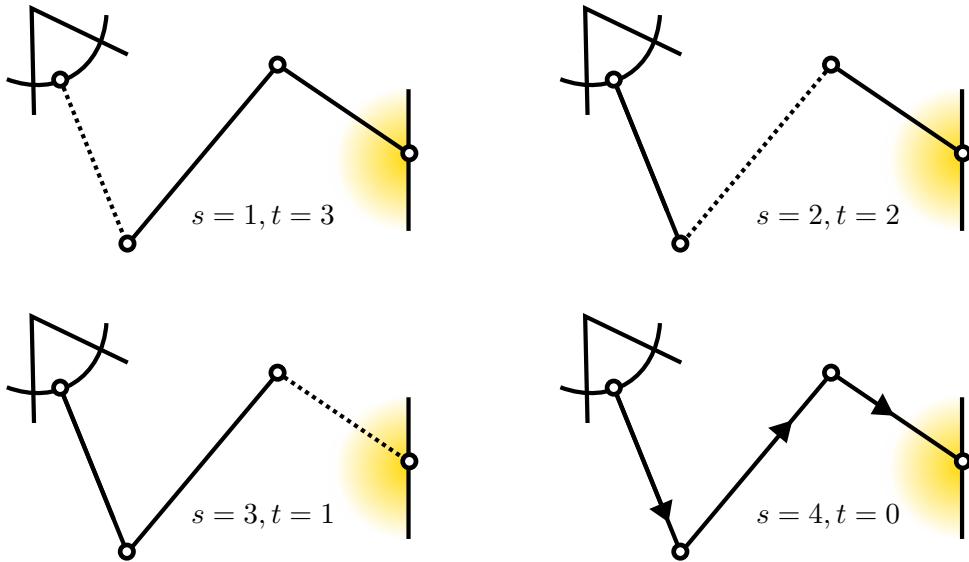


Figure 4.7.: Different strategies for sampling the same path with 4 vertices. In a full BPT this would also include implicitly sampling the camera on a light path ($s = 0, t = 4$); our renderer uses a pinhole camera and can therefore not sample such paths.

Let $\mathbf{x}_{s,t}\lambda$ refer to a path $\mathbf{x}\lambda = x_0\lambda_0 \dots x_{s+t-1}$ that was created by connecting x_{s-1} to x_s , and $p(\mathbf{x}_{s,t}\lambda)$ be the probability of sampling the path $\mathbf{x}\lambda$ by connecting vertices x_{s-1} of a camera path to x_s on a light path. The MIS weight of such a path is

$$w_{s,t}(\mathbf{x}\lambda) = \frac{p(\mathbf{x}_{s,t}\lambda)}{\sum_{i=0}^{s+t} p(\mathbf{x}_{i,s+t-i}\lambda)}. \quad (4.33)$$

In a non-spectral renderer the pdf of sampling the camera and light sub-path simply depends on the probability of the initial vertices on camera and light source, and the probabilities of sampling subsequent vertices. The probabilities for the sub-paths can be accumulated while the paths are sampled. The probability for the full path is the product

²Those two extreme cases are only possible if the scene contains area light sources and the camera has a finite surface in the scene.

of the sub-path probabilities:

$$\begin{aligned} p(\mathbf{x}_{s,t}) &= p_{cam}(x_0 x_1 \dots x_{s-1}) p_{light}(x_{s+t-1} \dots x_s) \\ p_{cam}(x_0 x_1 \dots x_{s-1}) &= \begin{cases} 1 & s = 0 \\ p_A(x_0) \cdot \prod_{i=0}^{s-2} p_A(x_i \rightarrow x_{i+1}) & s > 0 \end{cases} \\ p_{light}(x_{s+t-1} \dots x_s) &= \begin{cases} 1 & t = 0 \\ p_A(x_{s+t-1}) \cdot \prod_{i=s+1}^{s+t-1} p_A(x_i \rightarrow x_{i-1}) & t > 0 \end{cases} \end{aligned} \quad (4.34)$$

where $p_A(x)$ is the probability of sampling x over surface area, and $p_A(x \rightarrow x')$ is the probability of sampling x' from x over surface area.

In a spectral renderer without fluorescence, the pdf of the path also includes one additional probability for the wavelength transported along the path. In a simple implementation where this wavelength is sampled uniformly from all visible wavelengths this probability will be the same for all wavelengths and can be ignored when computing MIS weights. In a spectral renderer with fluorescence, the pdf of the path includes the probabilities for all the wavelengths along the sub-paths. This probability may vary for different sub-path lengths i and $k - i$. Pdfs for paths with fluorescence are presented in chapter 6.

4.7. Integration by Discretization of Wavelength Space

In the previous section we showed how to integrate the measurement equation using Monte Carlo integration. In particular, we sampled new vertices step by step to sample a path that can be used in the Monte Carlo estimator. For spectral rendering we also used Monte Carlo integration to sample the path's initial wavelength, in order to integrate the spectrum arriving at the camera. We can also use Monte Carlo integration to sample additional wavelengths at fluorescent surfaces, as discussed in chapter 5.3.

If the BRDFs, the sensor sensitivity and light sources used in the scene have independent behaviour for varying directions and varying wavelengths, i.e. if there exist a separate directional component $f_\omega(x \rightarrow x' \rightarrow x'')$ and wavelength component $f_\lambda(\lambda_{in}, x, \lambda_{out})$ such that

$$f_r(x \xrightarrow{\lambda_{in}} x' \xrightarrow{\lambda_{out}} x'') = f_\omega(x \rightarrow x' \rightarrow x'') \cdot f_\lambda(\lambda_{in}, x, \lambda_{out}) \quad (4.35)$$

and if there exist $W_\omega^{(j)}$ and $W_\lambda^{(j)}$ such that

$$W_e^{(j)}(x' \xrightarrow{\lambda} x) = W_\omega^{(j)}(x' \rightarrow x) \cdot W_\lambda^{(j)}(\lambda) \quad (4.36)$$

and if for each light source there exist L_ω and L_λ such that

$$L_e(x \xrightarrow{\lambda'} x') = L_\omega(x \rightarrow x') \cdot L_\lambda(\lambda'), \quad (4.37)$$

we can separate the integration over wavelengths from the integration over vertices and rewrite the measurement equation 4.30 as

$$\begin{aligned}
I_j &= \sum_{k=2}^{\infty} \int_{(\mathcal{M} \times \Lambda)^{k-1} \times \mathcal{M}} L_e(x_{k-1} \xrightarrow{\lambda_{k-2}} x_{k-2}) G(x_{k-1} \leftrightarrow x_{k-2}) \\
&\quad \cdot \prod_{i=1}^{k-2} \left(f_r(x_{i+1} \xrightarrow{\lambda_i} x_i \xrightarrow{\lambda_{i-1}} x_{i-1}) G(x_i \leftrightarrow x_{i-1}) \right) \\
&\quad \cdot W_e^{(j)}(x_1 \xrightarrow{\lambda_0} x_0) dA(x_{k-1}) d\lambda_{k-2} dA(x_{k-2}) \cdots d\lambda_0 dA(x_0) \\
&= \sum_{k=2}^{\infty} \int_{\mathcal{M}^k} \left(\int_{\Lambda^{k-1}} L_\lambda(x_{k-1}, \lambda_{k-2}) \cdot \prod_{i=1}^{k-2} f_\lambda(\lambda_i, x_i, \lambda_{i-1}) \cdot W_\lambda^{(j)} d\lambda_{k-2} \cdots d\lambda_0 \right) \\
&\quad \cdot L_\omega(x_{k-1} \rightarrow x_{k-2}) G(x_{k-1} \leftrightarrow x_{k-2}) \\
&\quad \cdot \prod_{i=1}^{k-2} (f_\omega(x_{i+1} \rightarrow x_i \rightarrow x_{i-1}) G(x_i \leftrightarrow x_{i-1})) \\
&\quad \cdot W_\omega^{(j)}(x_1 \rightarrow x_0) dA(x_{k-1}) \cdots dA(x_0).
\end{aligned} \tag{4.38}$$

A more detailed derivation of this equation is given in appendix A.1. In this formulation, the integration over vertices and the integration over wavelengths are now separate. We can also rewrite I_j in a special path integral formulation, where we use the alternative path-wavelength-space

$$\Omega'_\Lambda = \bigcup_{k=2}^{\infty} \mathcal{M}^k \times \Lambda^{k-1} \tag{4.39}$$

and alternative area-wavelength-product measure

$$\mu'_\Lambda = d\lambda_0 \cdots d\lambda_{k-2} \cdot dA(x_0) \cdots dA(x_{k-1}), \tag{4.40}$$

which results in the following equation:

$$I_j = \int_{\Omega'_\Lambda} f_j(\mathbf{x}\boldsymbol{\lambda}) d\mu'_\Lambda(\mathbf{x}\boldsymbol{\lambda}). \tag{4.41}$$

This allows us to integrate path space and wavelength space separately, and thus use different integration methods that are well suited for the individual integrals. For path space we still use Monte Carlo integration, but for wavelength space we can also use the rectangle method. We expect it to work well because the individual spectra are one-dimensional and the spectra's values are already provided at a fixed step width, since we store spectra as a vector (or array, or a similar data structure) of values representing its intensity in 5nm-bins. In the end, we are still integrating over all combinations of vertices and wavelengths, but since we only allow constrained BRDFs and light sources we can sample a geometric path (vertices only, no wavelengths) first, where we ignore wavelengths, and then integrate over all possible wavelength combinations on that path. It would still be possible to sample the geometric path first and then sample a random wavelength for each path length, thus using Monte Carlo integration to integrate over paths and wavelengths, but as we show in chapter 7, at least for simple test scenes the rectangle rule yields much smoother results than Monte Carlo integration when we need to integrate spectra.

4.7.1. Scenes without fluorescence

In scenes without fluorescence, we only need to integrate Λ once, since the wavelength can never change along a path. In this case, the measurement equation reduces to

$$\begin{aligned} I_j = & \sum_{k=2}^{\infty} \int_{\mathcal{M}^k} \left(\int_{\Lambda} L_{\lambda}(x_{k-1}, \lambda) \cdot \prod_{i=1}^{k-2} f_{\lambda}(\lambda, x_i, \lambda) \cdot W_{\lambda}^{(j)} d\lambda \right) \\ & \cdot L_{\omega}(x_{k-1} \rightarrow x_{k-2}) G(x_{k-1} \leftrightarrow x_{k-2}) \\ & \cdot \prod_{i=1}^{k-2} (f_{\omega}(x_{i+1} \rightarrow x_i \rightarrow x_{i-1}) G(x_i \leftrightarrow x_{i-1})) \\ & \cdot W_{\omega}^{(j)}(x_1 \rightarrow x_0) dA(x_{k-1}) \cdots dA(x_0). \end{aligned} \quad (4.42)$$

So after sampling a geometric path \mathbf{x} we can integrate its contributing spectrum by accumulating the BRDF values separately for each wavelength bin. We can achieve that with element-wise multiplication of the light source's emission spectrum at x_{k-1} to all the stored spectrum values of the BRDFs f_{λ} along the path, and finally multiplying the accumulated spectrum with the sensor sensitivity function. For pseudocode refer to algorithm 6.

4.7.2. Scenes with fluorescence

In scenes with fluorescence we may need to have more than one integration over Λ per path. As with non-fluorescent paths, we start with the emission spectrum of the light source and walk towards the camera. At each vertex where the BRDF is not fluorescent, we multiply BRDF's reflectance spectrum to the accumulated spectrum. Whenever we find a vertex with a fluorescent BRDF we need to integrate the accumulated spectrum to compute the absorbed energy. In order to do that we first multiply the accumulated spectrum with the BRDF's absorption spectrum and concentration parameter, integrate the accumulated spectrum with the rectangle rule, and use that integral and the BRDF's quantum yield to scale the BRDF's emission spectrum which becomes the new accumulated spectrum. For pseudocode refer to 6.

So in scenes with fluorescent surfaces we have two different kinds of integrations over Λ : The integration at the camera corresponds to applying the sensor sensitivity to the incident spectrum, since in the end each sensor can only output one value (the integral over the incident spectrum scaled by the sensor sensitivity function), and the integration at each fluorescent surface, which collects all incident light and redistributes it according to the emission spectrum.

4.8. Conversion from Spectra to RGB

In order to display an image that was rendered by a spectral renderer on a normal monitor, the accumulated spectral path contributions need to be converted to some RGB format. Our implementation for the conversion from wavelengths and spectra to RGB values heavily relies on code from [MSHD15]. First, each path's wavelength or spectrum is converted to XYZ using the color matching functions for the CIE 1931 standard observer [idl04] (figure 4.8). When the image is written to a file, the accumulated XYZ values are converted to sRGB using the conversion matrix from [pyt] (equation 4.43). This matrix contains negative entries, so it is possible to get negative red, green or blue values. This happens whenever the color falls outside the sRGB gamut, which is often the case due to the high saturation of fluorescent surfaces (e.g. figure 7.10b).

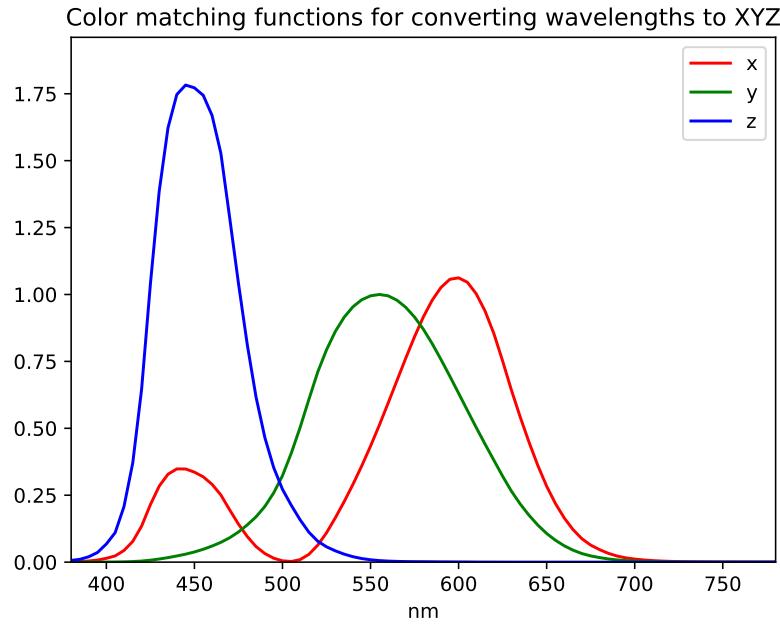


Figure 4.8.: Color matching functions for converting wavelengths to XYZ values.

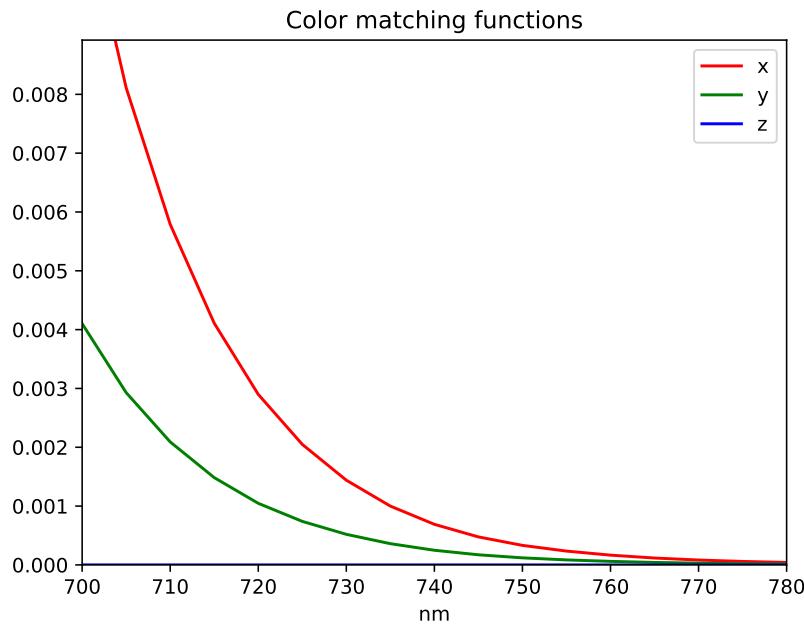


Figure 4.9.: Color matching functions zoomed in on [700, 780] nm.

$$M_{xyz \rightarrow rgb} = \begin{pmatrix} 3.24 & -1.54 & -0.50 \\ -0.97 & 1.88 & 0.04 \\ 0.06 & -0.20 & 1.06 \end{pmatrix} \quad (4.43)$$

Approximate matrix for converting a color from XYZ to the sRGB colorspace.

5. A diffuse and partly fluorescent BBRDF

In this chapter we present an analytical model for a perfectly diffuse BBRDF that consists of a purely fluorescent component and a non-fluorescent component, both of which are perfectly diffuse. Since this thesis is focused on how we need to adapt bidirectional light transport so it can handle wavelength changes, we did not investigate more complex surface models. The approach presented in 6.3.1.1 will not be able to deal with dispersion or any surface where the directional sampling depends on the wavelength. However, it should be possible to include simple extensions of the perfectly diffuse BBRDF, such as specular components or mirrors, without much additional effort.

For example, [WWLP06] investigated the qualitative behaviour of specular reflections on a mostly diffuse fluorescent surface and proposed two models that combine a fluorescent (or a combination of a fluorescent and non-fluorescent) perfectly diffuse component with a non-fluorescent glossy component.

5.1. Basics

5.1.1. The Reradiation Matrix

Some previous works on fluorescence in rendering used the reradiation matrix to model fluorescent surfaces [Gla95] [WTP01][WWLP06] [Hul10]. Its entries represent the fraction of energy transported from λ_{in} to λ_{out} .

The entries on the diagonal correspond to the non-fluorescent reflectance of a surface. The fluorescent energy transport is mostly on one side of the diagonal, because most emitted photons have a longer wavelength than absorbed photons.

The values on the diagonal are typically much bigger than the values representing fluorescence [Don54]. This is partly because of the conservation of energy: non-fluorescent surfaces can reflect close to 100% of incoming light at the same wavelength. Fluorescent surfaces distribute the reflected energy across a whole bandwidth of wavelengths. So instead of one reflectance value being less than one, the whole integral over a row or column (depending on how we order the matrix) corresponding to a single absorption wavelength has to be (less than) one. Note that for fluorescent materials the reflectivity - which is the integral over a row or column representing an emission wavelength - can be greater than one (for an example see figure 7.11b).

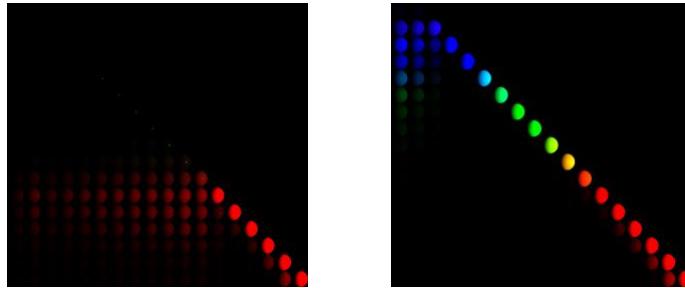


Figure 5.1.: Reradiation matrices for a fluorescent red paint (left) and paper (right). The images are taken from [HHA⁺10], who measured and visualized the matrices for several materials. The intensity represents how much light is reflected/reemitted from absorbed wavelengths (columns) to emitted wavelengths (rows).

However, acquiring measurements for such matrices is complex and time consuming, and since not many people have measured full reradiation matrices for fluorescent BBRRDFs before, we did not have any measurements available to use in our renderer. So instead, we had to design our own BBRRDF, for which we decided to stick to a simple diffuse model.

This model might not be completely in line with realistic measurements, but we hope that it provides an intuitive approximation. We did not have time to evaluate the quality of our BBRRDF compared to fluorescent measurements. We will also propose a method to construct a plausible reradiation matrix, based on an absorption, emission and reflectance spectrum used to parameterize our BBRRDF.

5.1.2. Assumptions

Even if our BBRRDF may not be the most realistic fluorescent BBRRDF possible, we still tried to base as much as possible on physically measurable properties. However, we also wanted to create a general model, so we ignored some behaviour that is only observed with some fluorescent substances, as well as effects that only occur for certain solvents, concentrations or other factors.

For example, in reality some fluorescent molecules change their behaviour when they are clumped together. We assume that all molecules are distributed homogeneously in our fluorescent surfaces, that they are mixed homogeneously with non-fluorescent molecules and do not interact with each other or change over time. We also ignore phosphorescence, photobleaching (reactions of fluorescent molecules caused by their higher reactivity while they are in an excited state that irreversibly change the molecule, thus destroying its fluorescent properties), and substances whose quantum yield depends on the absorbed wavelength (e.g. quinine).

5.1.3. Parameterization

The BBRRDF is based on a diffuse lambert BRDF. This means that light is reflected into all directions of the upper hemisphere equally. We will not allow transmission, transparency or media; our BBRRDF is for opaque surfaces only.

Our BBRRDF models surfaces as a combination of a fluorescent component and a non-fluorescent component.

The non-fluorescent component is parameterized by a single reflectance spectrum $r(\lambda)$. It corresponds to the diagonal of the reradiation matrix and does not alter wavelengths; the reflectance spectrum simply describes how much light is reflected for each wavelength.

The fluorescent component is parameterized by an absorption spectrum $a(\lambda)$, an emission spectrum $e(\lambda)$, the scalar quantum yield Φ , and a scalar concentration parameter c . This parameter correlates to the concentration of the fluorescent component. It can be used to control how many photons are absorbed, and how many photons should interact with the non-fluorescent component instead.

So for the fluorescent component, three of the four parameters can simply be derived from physical measurements. The fourth parameter c is responsible for scaling the absorption spectrum to influence how many photons are actually absorbed. The reason we introduced this parameter is that most available absorption spectra are usually scaled to one, or arbitrary units. However, if we directly use the value of the absorption spectrum to determine the amount of absorbed photons, this would mean that for the maximum absorption wavelength any photon of that wavelength will be absorbed by a fluorescent particle, and no light of that wavelength will be reflected. Since this might not be desirable nor realistic for every single material, we introduced a scalar parameter c . It can be used to control the amount of fluorescence in the material, and the probability for the absorption of a photon will still be proportional to the absorption spectrum, but scaled down by c . If a material actually should absorb all light at the fluorescent component's absorption maximum wavelength, this can be achieved by setting $c = 1$.

5.1.4. Physically reasonable parameters

A renderer can usually handle arbitrary values for the reflectance, absorption and emission spectra as well as Φ and c . So we are responsible for choosing input parameters that produce physically plausible behaviour. This section presents some constraints that guarantee a certain degree of realism, even if the spectra used for the BBRRDF are not based on measurements but designed by hand.

First, c and Φ should be in $(0, 1)$. $c = 0$ is a special case in which the BBRRDF's behaviour is that of a simple, non-fluorescent lambert BRDF. $c = 1$ simply means that at the absorption maximum all photons are absorbed, but will otherwise not have any unrealistic impact on the BBRRDF's behaviour.

$\Phi = 0$ is probably not realistic, although in some cases Φ can become pretty small. However, this case would simply mean that the material absorbs a lot of photons without re-emitting any at a different wavelength, so the same behaviour could also be modeled by a non-fluorescent BRDF with a lower reflectance spectrum, which would make fluorescent BBRRDFs with $\Phi = 0$ behave like a non-fluorescent material. We could not find any measurements where the quantum yield was equal to one, but in many cases the quantum yield can at least get very close to 1 (e.g. 0.97 for fluorescein in EtOH [tug]), so we did not want to introduce a hard limit for Φ here that is less than 1.

Next, the absorption spectrum should never be bigger than 1:

$$a(\lambda) \leq 1 \forall \lambda \in \Lambda. \quad (5.1)$$

The absorption spectrum determines how likely it is that a photon of a certain wavelength is absorbed, or in other words, how many photons out of a whole bundle of photons are absorbed for each wavelength. If the absorption spectrum was bigger than 1, the material would be able to absorb more photons than available. Since we scale down the absorption efficiency with c later, it makes sense to scale the absorption spectrum so that its maximum intensity equals 1.

Similarly, the non-fluorescent reflectance spectrum should never be bigger than 1 as well, since that would mean that for some wavelengths more light is reflected than is absorbed.

With fluorescent materials that are illuminated with a whole bandwidth of wavelengths, it is indeed possible that for some wavelengths more light is leaving the material than it originally absorbed, but in that case at least some of that light was absorbed at another wavelength and then re-emitted by fluorescent molecules. The non-fluorescent component alone cannot achieve that.

Last, the emission spectrum's area should be normed to 1:

$$\int_{\Lambda} e(\lambda) d\lambda = 1. \quad (5.2)$$

The emission spectrum represents the distribution of emitted wavelengths. Our goal is to use the emission spectrum to compute how much light is re-emitted at a certain wavelength. Since fluorescent materials emit light at a whole bandwidth of wavelengths, we need to make sure that the material does not emit more photons than it originally absorbed, therefore we need to scale down the emission spectrum such that its integral is 1.

If the emission spectrum is mostly at longer wavelengths than the absorption spectrum, this criterium also guarantees that no additional energy is introduced. Since light at longer wavelengths carries less energy than light at shorter wavelengths, this means that in fact on average some energy is lost. This corresponds to the non-radiative relaxation from a higher energy level in any excited state to the lowest energy level in the first excited state, which always happens before fluorescence.

The absorption and emission spectra that can be found online are often scaled to 1; in that case the absorption spectrum can be used directly, while the emission spectrum has to be scaled down by its area before it can be used as a parameter for our BBRRDF.

Unquantifiable properties

In chapter 2 we noted that while the absorption and emission spectra can overlap, light is mostly re-emitted at longer wavelengths. This means that most of the emission spectrum's area is at longer wavelengths than the absorption spectrum's area, and the emission spectrum's peak is at a longer wavelength than the absorption spectrum's peak. The distance between those peaks is called the **Stokes shift** [Lak06]. The stokes shift can range from a few nm to over 100nm, depending on the substance, so we do not want to set a limiting range here. However, for realistic behaviour the emission spectrum and its peak should generally be at longer wavelengths than the absorption spectrum.

If the emission and absorption spectra are plotted over wavenumber, the emission spectrum often is a mirror image of the absorption spectrum¹. This happens because of the similarity of the distribution of energy levels in the ground state and first excited state. Since there are many exceptions to the mirror image rule, we do not include it as a requirement for physically reasonable behaviour, but simply mention it as a guideline if one wants to design fluorescence spectra manually. For example, if the molecule can also be excited to the second excited state, the absorption spectrum includes additional peaks corresponding to those absorptions. These peaks are not mirrored by the emission spectrum, since all emissions start at the first excited state, irrespectively of the previous excited state. In simpler cases, there is usually one peak both in the absorption spectrum and the emission spectrum.

We show in 5.2.5 that for the proposed BBRRDF model for any fixed absorbed wavelength the integrated reflected energy over all exitant wavelengths is ≤ 1 as long as the parameters are physically reasonable.

¹The fluorescence spectra found online are often provided in intensity over wavelength. In that case the emission and absorption spectra look like distorted or stretched mirror images of each other.

5.2. The BBRRDF

We split the BBRRDF's behaviour in two cases. First we define the BBRRDF for purely fluorescent interactions, where the incident wavelength λ_{in} and the exitant wavelength λ_{out} are different. Then we look at interactions where $\lambda_{in} = \lambda_{out}$, in which case the BBRRDF can both reflect the incident light, or absorb it and re-emit it at the same wavelength.

5.2.1. For $\lambda_{in} \neq \lambda_{out}$

$$f(\lambda_{in}, \omega_{in}, x, \lambda_{out}, \omega_{out}) = c \cdot a(\lambda_{in}) \cdot \Phi \cdot e(\lambda_{out}) \cdot \frac{1}{\pi} \quad (5.3)$$

This equation describes fluorescent behaviour: The absorption spectrum a , scaled by the scalar c , describes how much of the incident light with wavelength λ_{in} is absorbed by the fluorescent component. The quantum yield Φ indicates how much of the absorbed energy is re-emitted at all, and the emission spectrum e tells us how much of the re-emitted energy is emitted at the new wavelength λ_{out} . The factor $\frac{1}{\pi}$ comes from the diffuse lambert BRDF and makes sure that light is distributed evenly over all projected solid angles of the upper hemisphere.

5.2.2. For $\lambda_{in} = \lambda_{out}$

$$\begin{aligned} f(\lambda, \omega_{in}, x, \omega_{out}) = & [r(\lambda) \cdot (1 - c \cdot a(\lambda)) \\ & + c \cdot a(\lambda) \cdot \Phi \cdot e(\lambda)] \cdot \frac{1}{\pi} \end{aligned} \quad (5.4)$$

Note that we allow the fluorescent component to re-emit absorbed light at the same wavelength, at least wherever the absorption and emission spectrum overlap. The second line in equation 5.4 models that behaviour.

No matter if this behaviour is included or not, the fluorescent material can still absorb light at λ ; this light is no longer available to be reflected by the non-fluorescent component. Therefore we scale the non-fluorescent reflectance $r(\lambda)$ by $1 - c \cdot a(\lambda)$, the relative amount of light that is not absorbed by the fluorescent component and still available for reflecting.

5.2.3. Full BBRRDF

Now we can combine equations 5.3 and 5.4 to formulate the full BBRRDF. The fluorescent component $c \cdot a(\lambda_{in}) \cdot \Phi \cdot e(\lambda_{out})$ is part of both cases, so we can simply copy it to the full BBRRDF. The non-fluorescent component only contributes to the BBRRDF if $\lambda_{in} = \lambda_{out}$, which we can achieve by adding a $\delta_{in,out}$ term to the non-fluorescent component. As a result, we get the following BBRRDF:

$$\begin{aligned} f(\lambda_{in}, \omega_{in}, x, \lambda_{out}, \omega_{out}) = & [\delta_{in,out} \cdot r(\lambda_{in}) \cdot (1 - c \cdot a(\lambda_{in})) \\ & + c \cdot a(\lambda_{in}) \cdot \Phi \cdot e(\lambda_{out})] \cdot \frac{1}{\pi}. \end{aligned} \quad (5.5)$$

This BBRRDF is reciprocal in terms of directions $\omega_{in}, \omega_{out}$, but it is **not** reciprocal in terms of wavelengths: $f(\lambda_{in}, \lambda_{out}) \neq f(\lambda_{out}, \lambda_{in})$. It is therefore important that we pay close attention to which wavelength is incident and which wavelength is leaving the surface, especially if we are dealing with paths constructed from the camera as well as the light source later on.

5.2.4. Formulation as a reradiation matrix

We can also convert the BBRRDF from equation 5.5 into the form of a reradiation matrix M . First, we need to decide on a set of discrete wavelengths $\lambda_0, \dots, \lambda_n$ that should correspond to the rows and columns of the matrix. \mathbf{a}, \mathbf{e} and \mathbf{r} are vectors representing the spectra evaluated at the λ_i :

$$\mathbf{a} = \begin{pmatrix} a(\lambda_0) \\ \vdots \\ a(\lambda_n) \end{pmatrix}, \mathbf{e} = \begin{pmatrix} e(\lambda_0) \\ \vdots \\ e(\lambda_n) \end{pmatrix}, \mathbf{r} = \begin{pmatrix} r(\lambda_0) \\ \vdots \\ r(\lambda_n) \end{pmatrix}. \quad (5.6)$$

Next we define the reradiation matrix of the non-fluorescent component $R = (r_{i,j})$ with

$$r_{i,j} = \begin{cases} r(\lambda_i) & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

We can thus construct a full reradiation matrix, where the rows and columns represent absorption and emission wavelengths $\lambda_0, \dots, \lambda_i$ respectively:

$$M = R \cdot (\mathbb{I} - c \cdot a) + c \cdot \Phi \cdot a \cdot e^T. \quad (5.7)$$

With this particular construction $R \cdot (\mathbb{I} - c \cdot a)$ is a diagonal matrix, while $c \cdot \Phi \cdot a \cdot e^T$ has rank one. In other words, the rows in the fluorescent component of the matrix are multiples of each other. This is also the case for real reradiation matrices [SBCD14]. Therefore storing the parameters as three spectra and two scalars instead of storing the full matrix saves a considerable amount of storage, and makes it easier to model and tweak the BBRRDF's parameters instead of modifying a full matrix.

If the absorption and emission spectrum do not overlap, the matrix will be triangular, and the diagonal will simply be the reflectance spectrum.

5.2.5. Energy Conservation

If all parameters were chosen according to 5.1.4, we can show that the integral over all emitted wavelengths for one fixed incident wavelength never exceeds 1. In terms of the reradiation matrix, this would mean that the sum of weighted values of each row (corresponding to one fixed incident wavelength) never exceeds 1, if the values are weighted with the difference between the wavelength represented by the current column and the exitant wavelength represented by the next column.

In particular, we need $\int_{\Lambda} e(\lambda) d\lambda = 1$ and $0 \leq r, a, \Phi, c \leq 1$. Thus we can show that the BBRRDF does not introduce additional energy for any absorbed wavelength λ_i as follows:

$$\begin{aligned} & \int_{\Omega} \int_{\Lambda} f(\lambda_{in}, \omega_{in}, \lambda_{out}, \omega_{out}) d\lambda_{out} d\omega_{out}^{\perp} \\ &= \int_{\Omega} \int_{\Lambda} [\delta_{in,out} \cdot r(\lambda_{in}) \cdot (1 - c \cdot a(\lambda_{in})) + c \cdot a(\lambda_{in}) \cdot \Phi \cdot e(\lambda_{out})] \cdot \frac{1}{\pi} d\lambda_{out} d\omega_{out}^{\perp} \\ &= \int_{\Lambda} \delta_{in,out} \cdot r(\lambda_{in}) \cdot (1 - c \cdot a(\lambda_{in})) + c \cdot a(\lambda_{in}) \cdot \Phi \cdot e(\lambda_{out}) d\lambda_{out} \\ &= r(\lambda_{in}) \cdot (1 - c \cdot a(\lambda_{in})) + c \cdot a(\lambda_{in}) \cdot \Phi \cdot \int_{\Lambda} e(\lambda_{out}) d\lambda_{out} \\ &= r(\lambda_{in}) \cdot (1 - c \cdot a(\lambda_{in})) + c \cdot a(\lambda_{in}) \cdot \Phi \cdot 1 \\ &\leq 1 \cdot (1 - c \cdot a(\lambda_{in})) + c \cdot a(\lambda_{in}) \cdot 1 \cdot 1 \\ &= 1 - c \cdot a(\lambda_{in}) + a(\lambda_{in}) \cdot c \\ &= 1. \end{aligned} \quad (5.8)$$

5.3. Sampling new wavelengths

If we use some form of path tracing and Monte Carlo integration to integrate over Ω_Λ , we might need to sample a new wavelength whenever the current path hits a fluorescent surface. This can be done by sampling wavelengths uniformly, or by importance sampling new wavelengths from the BBRRDF. We show in 7.1.2 that wavelength importance sampling introduces little to no overhead while increasing the image quality.

When importance sampling new wavelengths from the BBRRDF we need to note several things. First, the BBRRDF is not reciprocal in terms of wavelengths, so we need to use a different pdf depending on whether we are on a path constructed from the camera, or a path constructed from the light source. Second, since the BBRRDF can also reflect light without altering its wavelength, we also need to be able to produce paths that do not change their wavelength at every fluorescent surface. In other words, we need to be able to sample whether we even want to have a fluorescent interaction at the current BBRRDF before we actually sample a new wavelength, and if we do not have a fluorescent interaction we need to keep the path's current wavelength for the next path segment.

We start by considering a single photon's path, and all the processes that can happen when it hits a material that is modeled by our BBRRDF: Whenever a photon arrives at the surface with λ_{in} it can get absorbed by a fluorescent particle with probability $c \cdot a(\lambda_{in})$. If it does get absorbed, it gets re-emitted with probability Φ , and the new wavelength is a random variable with distribution e . If it does not get absorbed by the fluorescent component in the first place, it is reflected by the non-fluorescent component with probability $(1 - c) \cdot r$. If the photon is not re-emitted or reflected, it is absorbed by the material and its path stops here.

However, in path tracing we usually consider paths that carry the energy of multiple photons with the same wavelength instead of single photons. So instead of re-emitting, reflecting or absorbing the complete path with a certain probability, we continue each path but weight its energy. With this view, we can replace the probability of absorption of a single photon with an expected value of the fraction of photons that are absorbed (and treat the subsequent events accordingly). So if a whole bundle of photons arrive at a surface with λ_{in} , a fraction $c \cdot a(\lambda_{in})$ of photons gets absorbed by the fluorescent material, a fraction $\Phi \cdot e(\lambda_{out})$ of those absorbed photons gets re-emitted at different wavelengths λ_{out} , and a fraction $(1 - c) \cdot r(\lambda_{in})$ of the remaining photons gets reflected by the non-fluorescent component.

If we want to use Monte Carlo integration for integrating Ω_Λ , we always carry one wavelength per path segment. So if a path hits a fluorescent BBRRDF, we cannot simply split it up into multiple paths with different new wavelengths. Instead, we need to importance sample one new wavelength from the BBRRDF and weight the path's contribution with the probability of sampling that wavelength. Since the BBRRDF is a continuous function with a δ -peak at the incident wavelength, we split the sampling in two steps. First, we decide whether or not we even want to change the wavelength, and second, if we do have a fluorescent interaction, we sample a new wavelength from the emission spectrum (on light paths) or absorption spectrum (on camera paths). If we do not have a fluorescent interaction, the new wavelength is the same as the old wavelength.

Note that for camera paths, whenever we hit a fluorescent surface, we already know the exitant wavelength λ_{out} and need to sample a new incident wavelength λ_{in} , while on light paths we know the incident wavelength λ_{in} and need to sample a new exitant wavelength λ_{out} .

5.3.1. Probability of a fluorescent interaction

5.3.1.1. Light paths

Whenever we hit a fluorescent surface on a light path, we start with a fixed incident wavelength λ_{in} and want to decide if we want to sample a new wavelength λ_{out} or keep the incident wavelength as the exitant wavelength. Looking back to the overall energy reflected over all wavelengths from eq. 5.8, we can split the total reflected energy for λ_{in}

$$\text{total reflected energy} = r(\lambda_{in}) \cdot (1 - c \cdot a(\lambda_{in})) + c \cdot a(\lambda_{in}) \cdot \Phi \cdot 1 \quad (5.9)$$

into a non-fluorescent part

$$r(\lambda_{in}) \cdot (1 - c \cdot a(\lambda_{in}))$$

and a fluorescent part

$$c \cdot a(\lambda_{in}) \cdot \Phi.$$

We chose to construct the probability for having a fluorescent interaction from those two components:

$$\begin{aligned} P_l(\text{fluorescent interaction} | \lambda_{in}) &= \frac{\text{reflected fluorescent energy}}{\text{total reflected energy}} \\ &= \frac{c \cdot a(\lambda_{in}) \cdot \Phi}{c \cdot a(\lambda_{in}) \cdot \Phi + r(\lambda_{in}) \cdot (1 - c \cdot a(\lambda_{in}))}. \end{aligned} \quad (5.10)$$

The l in P_l indicates that we use this probability while constructing a light path. Since this is a binary choice and not a continuous pdf based on a BRDF, we call it P instead of p .

5.3.1.2. Camera paths

The same approach can be applied backwards to get the probability of a fluorescent interaction on a camera path. For a fixed λ_{out} , we can calculate how much of the energy leaving with wavelength λ_{out} is simply reflected, and how much was re-emitted from a fluorescent interactions after absorbing all incident wavelengths, by integrating over all **incident** wavelengths λ_{in} :

$$\begin{aligned} &\int_{\Omega} \int_{\Lambda} f(\lambda_{in}, \omega_{in}, \lambda_{out}, \omega_{in}) d\lambda_{in} d\omega_{in}^{\perp} \\ &= \int_{\Omega} \int_{\Lambda} [\delta_{in,out} \cdot r(\lambda_{in}) \cdot (1 - c \cdot a(\lambda_{in})) + c \cdot a(\lambda_{in}) \cdot \Phi \cdot e(\lambda_{out})] \cdot \frac{1}{\pi} d\lambda_{in} d\omega_{in}^{\perp} \\ &= \int_{\Lambda} \delta_{in,out} \cdot r(\lambda_{in}) \cdot (1 - c \cdot a(\lambda_{in})) + c \cdot a(\lambda_{in}) \cdot \Phi \cdot e(\lambda_{out}) d\lambda_{in} \\ &= r(\lambda_{out}) \cdot (1 - c \cdot a(\lambda_{out})) + c \cdot \int_{\Lambda} a(\lambda_{in}) d\lambda_{in} \cdot \Phi \cdot e(\lambda_{out}). \end{aligned} \quad (5.11)$$

Our probability for having a fluorescent interaction on the camera path is the fraction of reflected light to reflected and emitted light for a fixed exitant wavelength λ_{out} :

$$P_c(\text{fluorescent interaction} | \lambda_{out}) = \frac{c \cdot \int_{\Lambda} a(\lambda_{in}) d\lambda_{in} \cdot \Phi \cdot e(\lambda_{out})}{c \cdot \int_{\Lambda} a(\lambda_{in}) d\lambda_{in} \cdot \Phi \cdot e(\lambda_{out}) + r(\lambda_{out}) \cdot (1 - c \cdot a(\lambda_{out}))}. \quad (5.12)$$

Note that on camera paths the absorption spectrum is evaluated for the **emitted** wavelength λ_{out} . In the context of tracing a camera path the emitted wavelength at the current vertex is given - the emitted wavelength at a vertex x_i is the incident wavelength from the previous vertex x_{i-1} , if x_0 is on the camera. If we want to know how much of the light arriving at the same wavelength is reflected towards this λ_{out} , we need to scale $r(\lambda_{out})$ with $1 - ca(\lambda_{out})$ to account for light that arrives at the vertex with λ_{out} but is absorbed by the fluorescent component and can therefore not be reflected at the same wavelength by the non-fluorescent component.

For BRDFs without fluorescence, the probability of a fluorescent interaction is 0, both on camera and light paths.

For fluorescent BBRRDFs, whenever we are on a camera path and $e(\lambda_{out}) = 0$ (or whenever we are on a light path and $a(\lambda_{in}) = 0$), the probability of a fluorescent interaction is 0 as well; when implementing this the code should detect this case and return 0 directly instead of evaluating equations 5.10 or 5.12 to avoid division by 0 if r happens to be 0 as well.

Finally, note that the integral over the emission spectrum is 1, while the integral over the absorption spectrum is usually much bigger, which is why the integral over the emission spectrum does not appear in the above probabilities. The integral of the absorption spectrum can be precomputed and does not need to be evaluated every time a fluorescent surface is hit.

5.3.2. Probability of the new wavelength

In case of a fluorescent interaction on the light path (camera path), we sample the wavelength for the next segment from the emission (absorption) spectrum of the current vertex. Since the emission spectrum's area should already be normed to 1, we can directly use it as a probability density function from which we can sample a new wavelength. The absorption spectra should be scaled to 1, so we have to norm it before we can use it as a pdf. This can be done in advance, so sampling new wavelengths from the absorption spectrum should not introduce any overhead compared to sampling wavelengths from the emission spectrum.

Otherwise we keep the wavelength of the current segment for the next segment with probability 1. The full probability for the wavelength of the next segment is either the product of the probability for a fluorescent interaction and the pdf evaluated for the new wavelength, or simply one minus the probability for a fluorescent interaction if the wavelength is not modified.

Chapter 6.4 will explore in more detail how the probabilities of sampling individual wavelengths contribute to the probability of the full path, and how to define wavelength probabilities on non-fluorescent surfaces to achieve a unified equation for computing path probabilities.

6. Fluorescence in a spectral bidirectional path tracer

Bidirectional path tracing samples a sub-path starting at the camera and another sub-path starting at a light source independently and then tries to connect their vertices to create multiple complete light transport paths. In a spectral render where Monte Carlo integration is used to integrate the full path-wavelength-space, i.e. where the contributions of different wavelength combinations for a path are evaluated with Monte Carlo integration, the connected path is only valid if the incident wavelength of the last camera vertex and the emitted/reflected wavelength at the last light vertex are the same.

In spectral rendering without fluorescent surfaces (i.e. without vertices where the wavelength of a path can change) this is achieved by starting the camera and light sub-path with the same wavelength.

However, if we do this in scenes with fluorescent surfaces, we will not be able to sample all possible complete light transport paths, and as soon as one of the sub-paths changes its wavelength, connections between the sub-paths become invalid.

So in order to sample all possible paths as all combinations of vertices and wavelengths, the vertices on the camera and on the light source must be allowed to have different incident and exitant wavelengths respectively.

Note that for simulating a "normal" camera the available wavelengths for the camera vertex can be limited to visible wavelengths. The available wavelengths for the light vertex can also include invisible light from the ultraviolet region, if the light source emits UV light.

We will use the following notation for a complete light transport path with k vertices

$$\mathbf{x}\lambda = x_0\lambda_0x_1\dots\lambda_{k-2}x_{k-1}$$

where x_0 is on the camera, x_{k-1} is on a light source, λ_i refers to the wavelength transported on segment i from x_{i+1} to x_i . On a valid path, the wavelength λ_i on segment i between x_i and x_{i+1} is the same wavelength as the exitant wavelength at x_{i+1} and the incident wavelength at x_i .

6.1. Sampling a camera path

Each camera path starts with a vertex on the camera, and a wavelength sampled uniformly from the visible spectrum. This first vertex and the initial path direction are sampled just as with normal path tracing, and we trace a ray along the initial direction until it intersects another surface. This intersection is the path's next vertex.

If that vertex is not fluorescent, we simply sample a new direction (e.g. by importance sampling a local direction from the BRDF, or by performing NEE) and continue the path without changing its wavelength. For convenience, we will later define the probability of a fluorescent interaction at a non-fluorescent vertex as 0.

Whenever the path hits a vertex x with a fluorescent BBRRDF, depending on its parameters we decide whether we want a fluorescent or non-fluorescent interaction (see 5.3.1) with $P_c(\text{fluorescent interaction at } x|x.\lambda_{out})$. In case of a fluorescent interaction the incident wavelength for x is then sampled from the absorption spectrum and the path is continued into a new direction with that new wavelength. In case of a non-fluorescent interaction we set the incident wavelength to be the same as the exitant wavelength and continue the path into a new direction with that old wavelength.

If we find another vertex x' , the wavelength transported along the segment from x' to x as well as the exitant wavelength at x' are set to be the same as the incident wavelength at x . If we do not find another vertex (e.g. if the path leaves the scene, or if the path is terminated due to russian roulette), we still need to have sampled the expected incident wavelength at x , so we can connect x to light path vertices later on (e.g. figure 6.1: If the path is terminated at x_3 , we still need to set the incident wavelength λ_3 at x_3 , even if there is no incident direction at x_3).

Camera paths are terminated once their accumulated contribution becomes too small or they leave the scene. Note that the path's contribution also depends on the wavelengths of the path's segments - a camera path with contribution 0 might have had a bigger contribution for a different combination of wavelengths.

Algorithm 1 Sampling a Camera path (vertices and wavelengths)

```

1:  $x \leftarrow \text{sampleVertexOnPixel}()$ 
2:  $x.\lambda_{in} \leftarrow \text{sampleWavelengthUniformly}()$ 
3: while true do
4:    $x_{old} \leftarrow x$ 
5:    $\omega_{in} \leftarrow \text{sampleDirection}(x_{old})$ 
6:   if  $\text{ray}(x_{old}, \omega_{in})$  leaves scene then
7:     break
8:   end if
9:    $x \leftarrow \text{rayIntersect}(x_{old}, \omega_{in})$ 
10:   $x.\lambda_{out} \leftarrow x_{old}.\lambda_{in}$ 
11:  if  $\text{random01}() < P_c(\text{fluorescent interaction at } x|x.\lambda_{out})$  then
12:     $x.\lambda_{in} \leftarrow x.\text{absorptionSpectrum.sampleWavelength}()$ 
13:  else
14:     $x.\lambda_{in} \leftarrow x.\lambda_{out}$ 
15:  end if
16:  if  $\text{path.accumContribution} == 0$  then
17:    break
18:  end if
19: end while

```

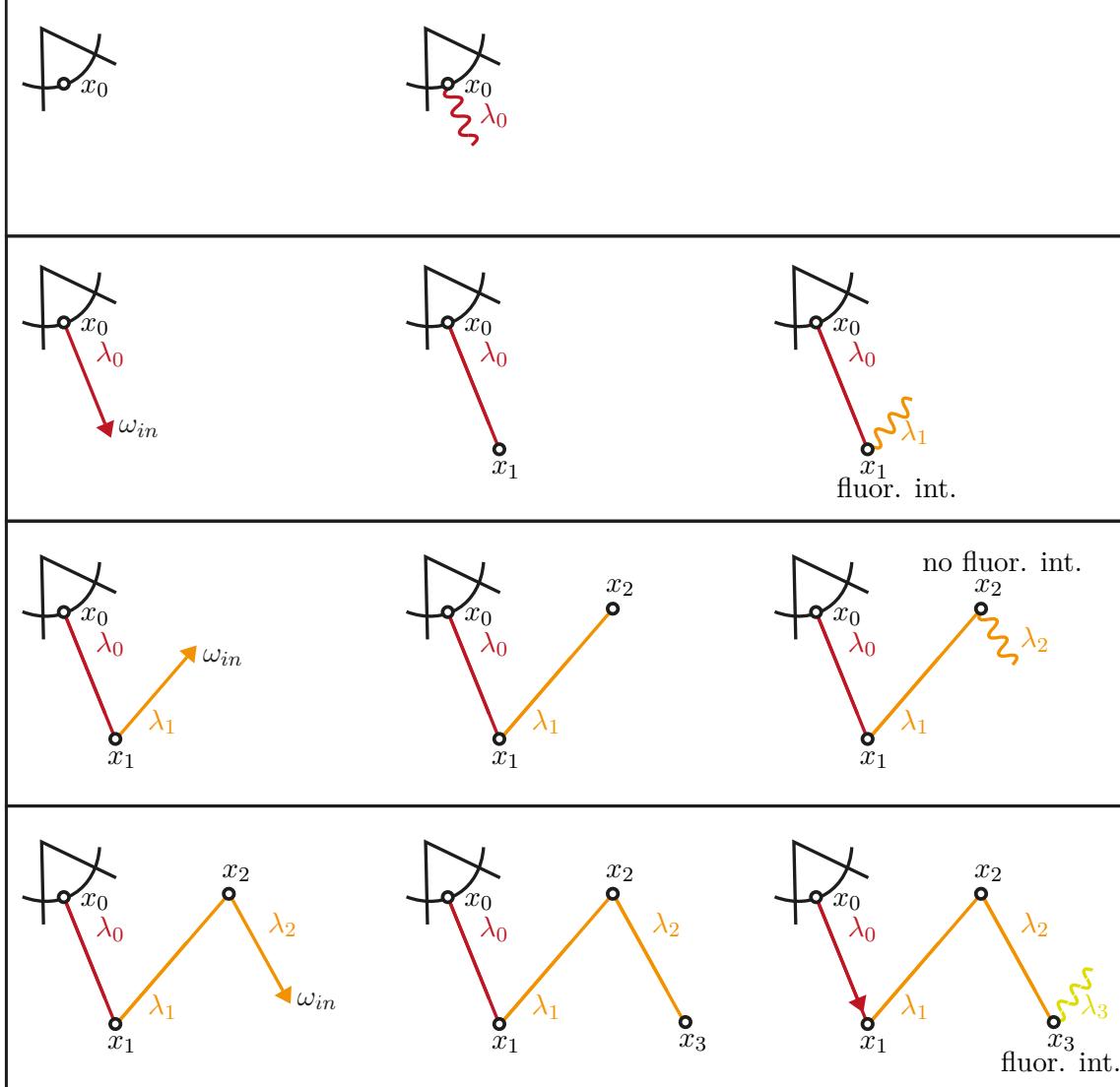


Figure 6.1.: Sampling a camera path.

Row 1: Sample point on camera and initial wavelength.

Rows 2-4: Sample 2nd, third, fourth point and their incident wavelengths. Note that even if the path is cancelled at x_3 , we still need to sample an incident wavelength λ_3 at x_3 , so we can connect to x_3 later.

6.2. Sampling a light path

Each light path starts with a vertex on a light source. Several options for sampling the initial vertex and wavelength are discussed in 6.2.1. Our renderer works with lambert area emitters, but the proposed algorithm should also be able to deal with other types of light sources, as long as the directional sampling doesn't depend on wavelengths (or, in other words, for every light source and any given direction the distribution of emitted wavelengths is the same). Point lights would require some additional attention, since they can never be implicitly hit by a camera path, but this would only influence the MIS weights for combining full light transport paths and should not affect the correctness of the renderer.

The light path is then traced through the scene. Whenever it hits a hybrid fluorescent surface, we first sample if we want a fluorescent interaction (see eq. 5.10), and if we do we sample a new emitted wavelength from the emission spectrum. In case of a non-fluorescent

interaction, as well as at any non-fluorescent surface, the exitant wavelength is the same as the incident wavelength.

Using those strategies for sampling the camera and light path separately, the probability of any pair of vertices from the camera and light path having the same incident and exitant wavelength is 0, so we need to modify some of those wavelengths later. We will discuss how to do this in section 6.3.1.1. Since our modification algorithm will only change wavelengths on the light path, and the contribution of a path depends on its wavelength combinations, we will not stop light paths if their throughput becomes zero - with the modification proposed later, it is possible that a light path that has a throughput of 0 while it is sampled will still produce a non-zero contribution for a different combination of wavelengths later on.

Algorithm 3 Example algorithm for sampling a light path. Note that light paths are not terminated when their throughput becomes 0, and that there exist multiple options for sampling the initial wavelength.

```

1:  $x \leftarrow \text{sampleVertexOnLightSource}()$ 
2:  $x.\lambda_{out} \leftarrow \text{sampleWavelengthFromLight}(x)$ 
3: while true do
4:    $x_{old} \leftarrow x$ 
5:    $\omega_{out} \leftarrow \text{sampleDirection}(x_{old})$ 
6:    $x \leftarrow \text{raytrace}(x_{old}, \omega_{out})$ 
7:    $x.\lambda_{in} \leftarrow x_{old}.\lambda_{out}$ 
8:   if  $\text{random01}() < P_l$  (fluorescent interaction at  $x|x.\lambda_{in}$ ) then
9:      $x.\lambda_{out} \leftarrow x.\text{emissionSpectrum.sampleWavelength}()$ 
10:   else
11:      $x.\lambda_{out} \leftarrow x.\lambda_{in}$ 
12:   end if
13: end while
```

6.2.1. Light Source Importance Sampling

We identified several options for starting a light path:

1. Sample a light source depending on its overall energy, sample a vertex x on the light source, and sample λ_{light} uniformly
2. Sample a wavelength λ_{light} and light source at the same time from the energy distribution of all light sources over all wavelengths
3. First sample λ_{light} uniformly, then sample a light source depending on its energy emitted at λ_{light}
4. First sample a light source depending on its overall energy, then sample λ_{light} from that light source's emission spectrum

The first option is the simplest one, but is also expected to produce the most noise.

The second and third option do not work well with the reset algorithm proposed later: Whenever the light sub-path wavelengths are reset all the way to the light source, the wavelength emitted at the light source in the final path will not be the wavelength that was responsible for sampling that light source in the first place. That vertex's pdf would no longer be valid, since its sampling depended on its original wavelength, which we no longer know about.

We chose the last option, because it fits with the overall algorithm and offers at least some degree of importance sampling, as opposed to the first option.

6.3. Resampling Wavelengths

The bidirectional path tracer first constructs a camera and light sub-path. Their initial wavelengths and potential new wavelengths at fluorescent surfaces along the paths are sampled independent of each other. Chapter 5.3 presented two strategies for sampling new wavelengths from fluorescent BBRRDFs, one for camera sub-paths and one for light sub-paths.

This section will discuss how to connect vertices from those sub-paths. We present two versions of the same idea, where some of the wavelengths of the full path are modified, and shortly discuss why we did not reset all wavelengths of a path. We will also explore special cases of sub-paths with 1 or 0 vertices - paths that are explicitly connected to or implicitly hit the camera or a light source.

6.3.1. Connecting Paths

If we connect an arbitrary camera vertex to an arbitrary light vertex, the incident wavelength required at the camera vertex will be different from the exitant wavelength required at the light vertex (see figure 6.2). In order to get a valid light transport path, we need to modify (some of) the wavelengths of the path. We will do this by starting at the connecting segment and then either walking along the light path, modifying the wavelengths of its segments to be consistent with the camera path, or walking along the camera path, modifying its wavelengths to be consistent with the light path.

Those two options only work if there are no surfaces or media in the scene where the probabilities used for sampling directions and vertices do not depend on the path's local wavelength - as soon as we reset some of the wavelengths along the path, the information on the previous wavelength is lost, so the pdf used for sampling new vertices has to be valid for any wavelength. Similarly we do not support wavelength-dependent refraction, where light of each wavelength is scattered into different directions. For example, if a path contains a smooth dielectric and the incident / exitant direction at the dielectric was chosen according to the path's original wavelength while it was constructed, the path's throughput will be 0 for any other wavelength interacting with the smooth dielectric for the path's given directions.

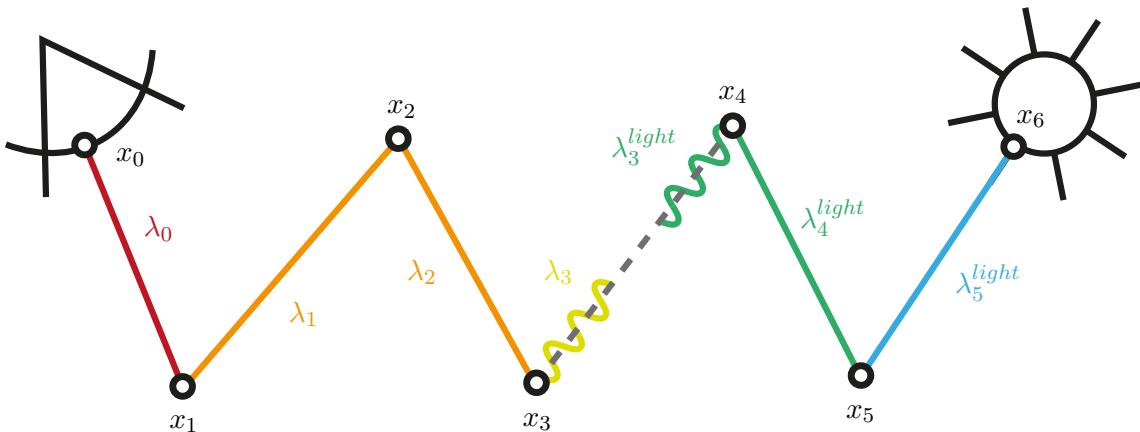


Figure 6.2.: A camera and light sub-path. The wavelengths of the segments are represented by different colors. When the sub-paths are sampled independently, none of the wavelengths will match, in this case $\lambda_3 \neq \lambda_3^{light}$, or in other terms, $x_3.\lambda_{in} \neq x_4.\lambda_{out}$.

6.3.1.1. Reset wavelengths on the light sub-path

Say we have a camera path with s vertices and a light path with t vertices ($s, t > 1$). In order to connect their end vertices and create a full path with k vertices, we first set the wavelength λ_{s-1} of the connecting segment to be the required incident wavelength of the last camera vertex x_{s-1} . If x_{s-1} is not fluorescent, or fluorescent but has a non-fluorescent interaction, this wavelength is the same as λ_{s-2} on the previous segment $s - 2$. If x_{s-1} is fluorescent and has a fluorescent interaction, the required incident wavelength was sampled from the absorption spectrum at x_{s-1} while the camera path was created.

Then we walk along the light path, starting at the connecting vertex x_s towards the light vertex x_{k-1} . At every non-fluorescent vertex we reset the wavelength of the following segment with probability 1. At every potentially fluorescent vertex we first sample whether or not to have a fluorescent interaction at that vertex as if we were constructing a camera path. In case of a non-fluorescent interaction we reset the wavelength of the following segment (with probability $1 - P_c(\text{fluor. interaction at } x_i | \lambda_{i-1})$) and continue the walk along the light path. In case of a fluorescent interaction we stop the walk. The wavelengths of all following segments remain the wavelengths that were originally sampled while the light path was created. We will call this the **forward reset algorithm**.

Let s' denote the index of the vertex where the walk is stopped. With the forward reset algorithm $s' \geq s$; if we sample a fluorescent interaction at x_s the walk is stopped right there, $s' = s$ and no wavelengths on the light path are reset. If we do not encounter any fluorescent surfaces along the walk, or if we only sample non-fluorescent interactions, we reset the wavelengths of all segments until we find the light source and s' will be $k - 1$. In that case, we simply need to evaluate the light source for this new wavelength. This means that this approach will also work for scenes without any fluorescent surfaces. Since the wavelengths for light path segments might be sampled multiple times there will be a slight overheads, but measurements have shown it to be negelactable (see 7.1.2).

Algorithm 4 Reset wavelengths along a light path

```

1: procedure RESET WAVELENGTHS FORWARD( $s, t$ )
2:    $\lambda_{s-1} \leftarrow \lambda_{s-2}$ 
3:   for  $i \leftarrow s$  to  $k - 2$  do
4:     if  $x_i$  is not fluorescent then
5:        $\lambda_i \leftarrow \lambda_{i-1}$ 
6:     else
7:       if  $\text{random01}() < P_c(\text{fluor. interaction at } x_i | \lambda_{i-1})$  then
8:          $s' \leftarrow s$ 
9:         break
10:      else
11:         $\lambda_i \leftarrow \lambda_{i-1}$ 
12:      end if
13:    end if
14:   end for
15: end procedure

```

6.3.1.2. Reset wavelengths on the camera sub-path

The same approach also works in reverse, as resetting wavelengths on the camera path. We will call this the **backward reset algorithm**. As a prerequisite, we can no longer terminate camera paths once their throughput becomes 0, but instead we can cancel light paths with a throughput of 0.

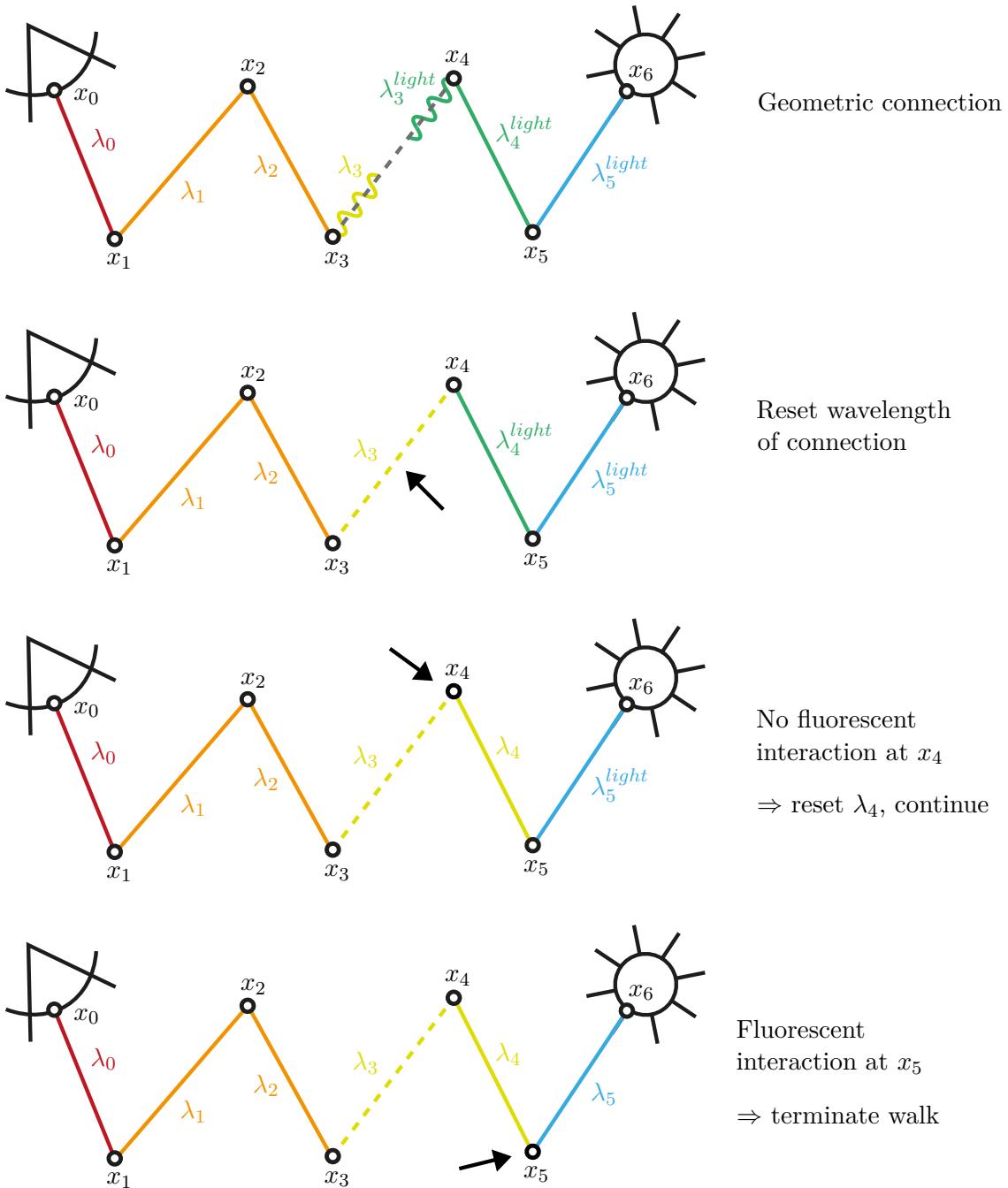


Figure 6.3.: Example of the forward reset algorithm.

When we connect a camera and light vertex, the connecting segment's wavelength is set to be the emitted/reflected wavelength at the last light vertex, x_s . If x_s does not have a fluorescent interaction, this will be the wavelength λ_s of the next segment or, if the path consists of only one light vertex, a wavelength that was sampled from the light source's emission spectrum. If it does, that wavelength is the one that was sampled from the vertex's emission spectrum while the light path was created.

The walk along the camera path works analogously to the walk along the light path described before. If no fluorescent surfaces are found, or no fluorescent interactions are sampled, the walk stops at the camera. If fluorescent surfaces are found, we sample if we

want to have a fluorescent interaction, but this time using P_l as if we were constructing a light path.

Algorithm 5 Reset wavelengths along a camera path

```

1: procedure RESET WAVELENGTHS BACKWARD( $s, t$ )
2:    $\lambda_{s-1} \leftarrow \lambda_s$ 
3:   for  $i \leftarrow s - 1$  to 1 do
4:     if  $x_i$  is not fluorescent then
5:        $\lambda_{i-1} \leftarrow \lambda_i$ 
6:     else
7:       if  $random01() < P_l(\text{fluor. interaction at } x_i | \lambda_i)$  then
8:          $s' \leftarrow s$ 
9:         break
10:      else
11:         $\lambda_{i-1} \leftarrow \lambda_i$ 
12:      end if
13:    end if
14:   end for
15: end procedure

```

In theory, at least with the simple pinhole camera model we used, this might be slightly more efficient than the forward reset algorithm, since the initial light path wavelength can be importance sampled from the light source, while the initial camera path wavelength is sampled uniformly from the visible spectrum. So for short paths, or paths without any fluorescent interaction, the final path will only carry one wavelength from light to camera, and importance sampling this wavelength might offer a lower variance than uniform sampling.

However, given our codebase, resetting wavelengths along the camera path turned out to be very inefficient. In the path tracing code explicit connections to the light source were done while the path was created, so resetting the wavelengths along the camera path every time lead to some significant overhead. The expected effort to make this more efficient did not fit into the given time frame for this thesis, so this approach was not pursued any further. In addition, if more complex camera models are used there might be other strategies for importance sampling the camera path's initial wavelength as well, which could improve the quality of resetting wavelengths along the light path.

6.3.1.3. Reset all wavelengths

With the two techniques described above, all segment wavelengths on the path (except for the camera path's initial wavelength) are importance sampled from either an emission spectrum on the light side of the segment, or from an absorption spectrum on the camera side of the segment. However, this wavelength contributes to the throughput in the BRDFs on both ends of the segment. So if we reset wavelengths after the path is completed, a better wavelength importance sampling option might be to sample the wavelength from a combination of the absorption spectrum on one end and the emission spectrum at the other end of a segment.

We did not pursue this approach for two reasons:

First, not every path consists only of fluorescent surfaces, and even if it does not all of those surfaces are guaranteed to have a fluorescent interaction every time. So we would also need to sample products of multiple adjacent non-fluorescent spectra between the next fluorescent emission and absorption spectra, and we would need to sample all

combinations of having fluorescent and non-fluorescent combinations, which would become too complicated. As the measurements in 7.7b indicate, multiplying spectra creates some overhead, so sampling them and all their combinations would lead to even more overhead and less quality than just evaluating the full spectrum at once.

Second, in order to sample adjacent absorption and emission spectra, one would have to multiply those spectra together. In our implementation, this meant an element-wise multiplication of two vectors representing wavelength bins of the spectra. This would result in at least as much overhead as just evaluating the path for all wavelength bins, which is discussed in 6.5.

6.3.2. Connecting a light vertex to the camera

Due to time constraints paths with 1 or 0 camera vertices were not implemented.

If such paths were included, instead of resetting the wavelengths of the light path one can also use the last light vertex's wavelength as the camera vertex's wavelength and not reset any wavelengths on the light path. Not only is this simpler, it might also lead to a lower variance for simple camera models: Remember that all the light path segments' wavelengths were importance sampled, either from the light source emission spectrum at the beginning, or fluorescent emission spectra along the way. If the forward reset algorithm is applied here, the first wavelength of the final path would be sampled uniformly among all visible wavelengths, which is probably worse than the importance sampled last wavelength of the light path. However, this has to be considered when computing the MIS weights.

6.3.3. Special Case: 0 light vertices

In scenes with area light sources it is possible for the camera path to hit a light source randomly. Since all the wavelengths of the full path were sampled while creating the camera path, the final path will have consistent wavelengths. If we use the forward reset algorithm, there is no need to reset anything, so we can skip to computing the path's throughput and MIS weight (see 6.4). If we use the backward reset algorithm, we need to sample a new wavelength from the light source and start a walk along the camera path at the light vertex, resetting the camera path's wavelengths to the new wavelength until we sample a fluorescent interaction.

6.3.4. Special Case: 1 light vertex

In this case we explicitly connect a camera path with an arbitrary number of vertices to the light source (NEE). In order to decrease correlations between different connected paths, instead of using the first vertex of the same light sub-path for connections with every camera path vertex, each camera vertex is connected to a new light vertex on a random light source.

If we use the forward reset algorithm, in order to have a consistent MIS weight computation this light vertex is sampled from the overall energy distribution of all light sources, as discussed in option 4 in 6.2.1.

However, in this special case of 1 light vertex we already know which wavelength needs to be emitted at the light source vertex. A possible improvement might be to sample the light source depending on its intensity for the required wavelength instead of its overall energy (as we did for constructing full light paths, where the light source is sampled independently of wavelengths). This will make the MIS weight computation more complicated, because the probability of sampling the light vertex in a 1-vertex-light-sub-path will be different than the probability of sampling the same vertex in a multiple-vertex-light-sub-path.

For the backward reset algorithm, the light vertex should never be sampled depending on the last camera path vertex's incident wavelength. Instead, when the light vertex is sampled, we also need to sample a new wavelength from the chosen light source's emission spectrum anyway. This wavelength is used to replace the last camera vertex's incident wavelength, and for the backward reset algorithm starting at the last camera vertex.

6.4. Throughput

This section explores the throughput for a full light transport path $\mathbf{x}\lambda$, where each segment between two vertices transports one wavelength and the vertices and wavelengths were sampled randomly from some pdf. We will use the notation $\mathbf{x}_{s,t}\lambda$ for a path $\mathbf{x}\lambda$ that was sampled by connecting a camera sub-path with s vertices to a light sub-path with t vertices.

The throughput of a path is its measurement contribution f_j (see 6.4.1) weighted with the probability p of sampling the path (see 6.4.2) and its MIS weight $w_{s,t}$ (see 6.4.3) for combining paths that were sampled with different strategies. Note that the measurement contribution function f_j is independent of the strategy used to sample the path, i.e. it is independent of the lengths s and t of the camera and light sub-paths, while the path's probability and MIS weight depend on s and t . After sampling a path with a certain probability, we add its throughput to the multi-sample estimator, which we use to estimate the measured intensity at pixel j .

$$\text{throughput}(\mathbf{x}_{s,t}\lambda) = w_{s,t}(\mathbf{x}\lambda) \cdot \frac{f_j(\mathbf{x}\lambda)}{p(\mathbf{x}_{s,t}\lambda)}. \quad (6.1)$$

Note that this equation does not include division by the number of samples, which can be done after the throughputs of a sufficient number of paths were added up. We decided not to include this division here so that eq. 6.1 can be used to compute the throughput of any individual path without considering how many paths were sampled before, or are to be sampled in total.

6.4.1. Measurement contribution of a path

The measurement contribution of a path can be computed similarly to non-fluorescent path tracing. The only difference is that the BRDFs along the path may in fact be BBRRDFs and thus depend on the incident and exitant wavelength, so we need to include wavelengths into a full representation of a path. The measurement contribution of a path is independent of the strategy (lengths s, t of the camera and light sub-path, strategy used to reset wavelengths on some segments) that was used to sample the path.

If we have a path $\mathbf{x}\lambda = x_0\lambda_0 \dots x_{k-2}\lambda_{k-2}x_{k-1}$ with x_0 on the camera, x_{k-1} on a light source, and λ_i as the wavelength transported from x_{i+1} to x_i , the measurement contribution function f_j for energy arriving at x_0 on a pixel j for wavelength λ_0 is

$$\begin{aligned} f_j(\mathbf{x}\lambda) &= L_e(x_{k-1} \xrightarrow{\lambda_{k-2}} x_{k-2}) G(x_{k-1} \leftrightarrow x_{k-2}) \\ &\quad \cdot \prod_{i=1}^{k-2} f_r(x_{i+1} \xrightarrow{\lambda_i} x_i \xrightarrow{\lambda_{i-1}} x_{i-1}) G(x_i \leftrightarrow x_{i-1}) \\ &\quad \cdot W_e^{(j)}(x_1 \xrightarrow{\lambda_0} x_0). \end{aligned} \quad (6.2)$$

The measurement contribution function was introduced in more detail in chapter 4.5.

6.4.2. Probability of a complete connected path

The probability of sampling the complete path $\mathbf{x}\lambda$ depends on the length of the camera and light sub-path as well as the strategies used to sample the wavelengths. This section will discuss these probabilities assuming the wavelengths were reset along the light path (described in 6.3.1.1).

The probabilities for resetting wavelengths on the camera path can be computed in the same manner, which we will discuss at the end of this section.

If the scene fulfills all conditions we required for the forward reset algorithm or for integrating the wavelength space with the rectangle rule (see 4.7), we can split the probability of sampling the path into a directional probability p_A and wavelength probability p_Λ . Note that this does **not** work if the scene contains materials or media where the scattering direction or path segment lengths varies for different wavelengths.

For paths that are evaluated for all wavelength bins (discussed in 6.3.1.3), we only need the directional probabilities p_A from this section. The wavelength probabilities p_Λ can be ignored, since for that approach no wavelengths need to be sampled.

6.4.2.1. Directional probabilities

The directional probability p_A for sampling the vertices of a path is measured over surface area can be computed as in a non-spectral bidirectional path tracer. If both sub-paths have at least one vertex, the probability of sampling x_0 and x_{k-1} does not depend on any other vertices, since those are the first vertices of their respective paths. In that case, the directional probability p_A of sampling the vertices \mathbf{x} of the full path $\mathbf{x}\lambda$ with s vertices on a camera sub-path and t vertices on a light sub-path is

$$p_A(\mathbf{x}_{s,t}) = p_A(x_0) \cdot \prod_{i=1}^s p_A(x_{i-1} \rightarrow x_i) \cdot \prod_{i=s+2}^{k-1} p_A(x_i \rightarrow x_{i-1}) \cdot p_A(x_{k-1}). \quad (6.3)$$

The last factor $p_A(x_{k-1})$ is not part of p_A for paths with $t = 0$ (no light vertex, this happens whenever the camera path randomly hits a light source; this is not possible with point light sources). $p_A(x_0)$ is not part of p_A when $s = 0$. This is only possible if light paths can randomly hit the camera, which our code does not support.

For some BRDFs $p_A(x_i \rightarrow x_{i+1})$ may also depend on x_{i-1} (unless $i = 0$), and $p_A(x_i \rightarrow x_{i-1})$ may depend on x_{i+1} (unless $i = k - 1$).

6.4.2.2. Wavelength probabilities

First, we use $p_c(\lambda_0|x_0)$ as the pdf for sampling the initial wavelength of a camera path starting at x_0 , and $p_l(\lambda_{k-2}|x_{k-1})$ as the pdf for sampling the initial wavelength of a light path starting on a light source x_{k-1} . The concrete values of these pdfs depend on the choice of sampling strategy for initial path wavelengths. In our implementation we sample initial camera wavelengths uniformly

$$p_c(\lambda_0|x_0) = \begin{cases} \frac{1}{400} & \lambda_0 \in [380, 780] \\ 0 & \text{otherwise,} \end{cases} \quad (6.4)$$

and the initial light path wavelength from the light source's emission spectrum e

$$p_l(\lambda_{k-2}|x_{k-1}) = \frac{e(\lambda_{k-2})}{\int_\Lambda e(\lambda)d\lambda}. \quad (6.5)$$

Next, we define the probabilities for individual segment wavelengths λ_i for all segments i within the path ($0 < i < k - 2$), depending on whether we consider them on a camera or light sub-path:

$$p_c(\lambda_i|x_i, \lambda_{i-1}) = \begin{cases} P_c(\text{fluorescent interaction at } x_i|\lambda_{i-1}) \cdot p_{x_i.abs.}(\lambda_i) & \lambda_i \neq \lambda_{i-1} \\ 1 - P_c(\text{fluorescent interaction at } x_i|\lambda_{i-1}) & \lambda_i = \lambda_{i-1} \end{cases} \quad (6.6)$$

and

$$p_l(\lambda_i|x_{i+1}, \lambda_{i+1}) = \begin{cases} P_l(\text{fluorescent interaction at } x_{i+1}|\lambda_{i+1}) \cdot p_{x_{i+1}.em.}(\lambda_i) & \lambda_i \neq \lambda_{i+1} \\ 1 - P_l(\text{fluorescent interaction at } x_{i+1}|\lambda_{i+1}) & \lambda_i = \lambda_{i+1}. \end{cases} \quad (6.7)$$

$p_{x_i.abs.}$ and $p_{x_{i+1}.em.}$ are pdfs that are proportional to the absorption spectrum at x_i and the emission spectrum at x_{i+1} respectively, since we sample new wavelengths from those spectra whenever there is a fluorescent interaction.

For convenience we define $P_x(\text{fluorescent interaction at } x_i) := 0$ if x_i is not fluorescent. In those cases we get

$$p_c(\lambda_i|x_i, \lambda_{i-1}) = p_l(\lambda_{i-1}|x_i, \lambda_i) = \begin{cases} 0 & \lambda_i \neq \lambda_{i-1} \\ 1 & \lambda_i = \lambda_{i-1}. \end{cases} \quad (6.8)$$

However, whenever we override some of the wavelengths along the light path, the probabilities for the original wavelengths of those segments will no longer be valid. For each segment i that gets assigned a new wavelength λ_i , we need to consider two new probabilities: The probability of actually changing the wavelength at vertex x_i ¹ (the probability of a fluorescent interaction, see 5.3.1), and the probability of the new incident wavelength itself. If we do not change all the wavelengths on the light sub-path, we also need to include the probability for keeping the wavelength (not having a fluorescent interaction) at the vertex where we stop the reset algorithm.

Let $x_{s'}$ be the vertex where the forward reset algorithm stops. This means that either $x_{s'}$ is on a light source, or it is the first vertex on the light sub-path with a fluorescent interaction. In the latter case the incident wavelength $\lambda_{s'}$ at $x_{s'}$ is the wavelength that was originally sampled while creating the light path, while the emitted wavelength $\lambda_{s'-1}$ was assigned during the reset algorithm.²

All vertices from x_s to $x_{s'-1}$ will have a non-fluorescent interaction, or not be fluorescent at all. The probability for the wavelengths on the segments between x_s and $x_{s'}$ are the probabilities of not having a fluorescent interaction at the vertex on the camera side of the segment. For non-fluorescent vertices the probability of not having a fluorescent interaction is 1.

If $x_{s'}$ is not on the light source, we need to include the probability of having a fluorescent interaction at $x_{s'}$ coming from the camera side. In that case, the wavelength probabilities of all segments after $x_{s'}$ are the probabilities from the original light path.

¹When the wavelengths are reset along the camera path instead of the light path, the vertex responsible for changing the wavelength λ_i of segment i is x_{i+1}

²For the MIS weights we also need to evaluate probabilities for different \hat{s}, \hat{t} combinations. For those, we pretend that the forward reset algorithm started its walk along the light path at $x_{\hat{s}}$, and \hat{s}' is the index of the first vertex with a fluorescent interaction

For $s' < k - 1$ the probability p_Λ for sampling all wavelengths on a complete path using the forward reset algorithm is

$$\begin{aligned}
 p_\Lambda(\mathbf{x}_{s,t}, \boldsymbol{\lambda}) = & p_c(\lambda_0) \cdot \prod_{i=1}^{s-1} p_c(\lambda_i | x_i, \lambda_{i-1}) \\
 & \cdot \prod_{i=s}^{s'-1} (1 - P_c(\text{fluorescent interaction at } x_i | \lambda_{i-1})) \\
 & \cdot P_c(\text{fluorescent interaction at } x_{s'} | \lambda_{s'-1}) \\
 & \cdot \prod_{i=s'}^{k-3} p_l(\lambda_i | x_{i+1}, \lambda_{i+1}) \\
 & \cdot p_l(\lambda_{k-2}).
 \end{aligned} \tag{6.9}$$

The first line in equation 6.9 contains the probabilities for the wavelengths on the camera sub-paths. These wavelengths were sampled while the camera path was constructed and are never modified in the forward reset algorithm. In our implementation, λ_0 is sampled uniformly, while all subsequent new λ_i are sampled from a fluorescent absorption spectrum.

The second line covers all wavelengths that were reset along the light sub-path. Since we only reset wavelengths until we sample a fluorescent interaction, the corresponding segments all carry the same wavelength, and for each segment on the light path with that wavelength we need to include the probability for not having a fluorescent interaction at the vertex x_i on the camera side of segment i .

The third line is the probability for terminating the forward reset algorithm at $x_{s'}$. We only terminate the algorithm if the reset algorithm continues until the light source (which we excluded with $s' < k - 1$) or if we sample a fluorescent interaction, given the wavelength $\lambda_{s'-1}$ on the camera side of vertex $x_{s'}$.

The fourth line collects the probabilities of the remaining wavelengths along the light sub-path. Those wavelengths were sampled while the light path was constructed. The last line represents the probability of sampling the initial wavelength on the light path. In our implementation, the wavelengths along the light path were importance sampled from emission spectra, and the light path's initial wavelength is importance sampled from the light source's emission spectrum.

If $s' = k - 1$ (which means that the forward reset algorithm replaces all wavelengths of the light path and terminates at the light source), or if $t < 2$ (which means that we have 0 or 1 light vertices, in which case we do not apply the forward reset algorithm at all), the last three lines of equation 6.9 can be omitted and the probability for sampling all wavelengths along the path reduces to

$$\begin{aligned}
 p_\Lambda(\mathbf{x}_{s,t}, \boldsymbol{\lambda}) = & p_c(\lambda_0) \cdot \prod_{i=1}^{s-1} p_c(\lambda_i | x_i, \lambda_{i-1}) \\
 & \cdot \prod_{i=s}^{k-2} (1 - P_c(\text{fluorescent interaction at } x_i | \lambda_{i-1})).
 \end{aligned} \tag{6.10}$$

Wavelength probabilities for the backward reset algorithm

Instead of resetting wavelengths along a light path we can also reset wavelengths along the camera path and set the connecting segment's wavelength λ_{s-1} to be the exitant wavelength at the adjacent light path vertex x_s . Let $x_{s'}$ be the vertex where the backward reset algorithm was terminated. If $s' > 0$ the probability for sampling all wavelengths on a path with the backward reset algorithm is

$$\begin{aligned}
 p_{\Lambda}^{back}(\mathbf{x}_{s,t}, \boldsymbol{\lambda}) = & p_c(\lambda_0) \cdot \prod_{i=1}^{s'-1} p_c(\lambda_i | x_i, \lambda_{i-1}) \\
 & \cdot P_l(\text{fluorescent interaction at } x_{s'} | \lambda_{s'}) \\
 & \cdot \prod_{i=s'+1}^{s-2} (1 - P_l(\text{fluorescent interaction at } x_i | \lambda_i)) \\
 & \cdot \prod_{i=s-1}^{k-3} p_l(\lambda_i | x_{i+1}, \lambda_{i+1}) \\
 & \cdot p_l(\lambda_{k-2}).
 \end{aligned} \tag{6.11}$$

The main difference here is that in line 2 we use $P_l(\text{fluorescent interaction at } x_{s'} | \lambda_{s'})$ instead of $P_c(\text{fluorescent interaction at } x_{s'} | \lambda_{s'-1})$ as a criterium for terminating the reset algorithm.

After applying any of the two reset algorithms we know that the wavelengths on the segments between s and s' (or s' and s with the backward reset algorithm) are the same as the connecting segment's wavelength $\lambda_{s'-1}$. Therefore we can use a more general formulation for p_{Λ} and p_{Λ}^{back} where we can see more clearly that the only difference in computing the wavelength probabilities for either strategy is the termination criterium at $x_{s'}$:

$$\begin{aligned}
 p_{\Lambda}(\mathbf{x}_{s,t}, \boldsymbol{\lambda}) = & p_c(\lambda_0) \cdot \prod_{i=1}^{s'-1} p_c(\lambda_i | x_i, \lambda_{i-1}) \\
 & \cdot P_c(\text{fluorescent interaction at } x_{s'} | \lambda_{s'-1}) \\
 & \cdot \prod_{i=s'}^{k-3} p_l(\lambda_i | x_{i+1}, \lambda_{i+1}) \cdot p_l(\lambda_{k-2})
 \end{aligned} \tag{6.12}$$

$$\begin{aligned}
 p_{\Lambda}^{back}(\mathbf{x}_{s,t}, \boldsymbol{\lambda}) = & p_c(\lambda_0) \cdot \prod_{i=1}^{s'-1} p_c(\lambda_i | x_i, \lambda_{i-1}) \\
 & \cdot P_l(\text{fluorescent interaction at } x_{s'} | \lambda_{s'}) \\
 & \cdot \prod_{i=s'}^{k-3} p_l(\lambda_i | x_{i+1}, \lambda_{i+1}) \cdot p_l(\lambda_{k-2}).
 \end{aligned} \tag{6.13}$$

Finally, if we apply the backward algorithm and $s' = 0$, or if we want to apply the backward reset algorithm but $s < 2$, the first two lines in equation 6.13 (corresponding to the first three lines in equation 6.11) can be omitted, just as we did for the forward reset algorithm in equation 6.10.

6.4.2.3. Full pdf

This thesis only considers surfaces where the directional probabilities are independent of the path's wavelengths. Therefore the full probability of sampling a path and all its wavelengths is the product of the individual probabilities defined before:

$$p(\mathbf{x}_{s,t}, \boldsymbol{\lambda}) = p_A(\mathbf{x}_{s,t}) \cdot p_{\Lambda}(\mathbf{x}_{s,t}, \boldsymbol{\lambda}). \quad (6.14)$$

6.4.2.4. Special cases: 0 camera / light vertices

If the scene contains area light sources, it is possible that a camera path randomly hits a light source. Similarly, if the camera model occupies some area, it is possible that a light path implicitly hits a pixel. In that case, the terms $p_l(\lambda_{k-1})$ or $p_c(\lambda_0)$ have to be omitted, since the emitted wavelength at the light source was in fact sampled while constructing a camera path, or the incident wavelength expected at the camera was sampled while constructing a light path.

6.4.2.5. Special case: 1 light vertex

If we explicitly construct a full path by connecting a camera sub-path to a vertex on a light source (next event estimation, see 4.6.2), we know which wavelength we need the light source to emit in order to create a valid complete path with matching wavelengths. Since we already know the wavelength we need, we can also importance sample the light source depending on its intensity at the required wavelength. This will make the MIS weights more complicated, as we discussed in 6.3.4, and was therefore not implemented, but might improve the overall image quality.

For light sub-paths with more than one vertex importance sampling the light path's start vertex depending on a certain wavelengths is not promising for multiple reasons:

First, it is possible that along the camera path the wavelength changes at fluorescent surfaces. So if we wanted to importance sample the light source of the light path to which we will connect the different camera vertices, we might have to sample a new light path for each camera vertex, which would work against the efficiency of bidirectional path tracing.

Second, even if we importance sample the light source for a certain wavelength, there is no guarantee that subsequent vertices on the light path will still emit / reflect that same wavelength. If they are fluorescent and subsequent wavelengths are different, we can still apply the forward reset algorithm and get a correct result, but there would be no reason to expect that importance sampling the light source for a seemingly random wavelength will be better than just sampling a light source depending on its overall energy.

Third, if there is fluorescence on the light sub-path, the forward reset algorithm might walk all the way to the light source and change the wavelength emitted at the light source, which would defeat the whole purpose of importance sampling the light source for a certain wavelength.

6.4.3. MIS weights

A bidirectional path tracer can sample a path with k vertices in $k + 1$ different ways, with $s = 0, \dots, k$ camera vertices and $t = k - s$ light vertices. If the bidirectional path tracer samples wavelengths along the camera and light sub-paths while they are created, and then resets some of them with either the forward or backward reset algorithm, there are still only $k + 1$ ways to sample a path with k vertices **and** the $k - 1$ wavelengths transported from vertex to vertex: Given a path $\mathbf{x}_{s,t}, \boldsymbol{\lambda} = x_0 \lambda_0 x_1 \dots \lambda_{k-2} x_{k-1}$ that was created by

connecting a camera sub-path with s vertices to a light sub-path with t vertices, we can reconstruct the vertex $x_{s'}$ where the forward reset algorithm terminated by starting at x_s and walking along the light path until we find a vertex with different incident and exitant wavelength. Therefore, given a list of vertices and wavelengths and the s, t used to sample them, there is only one way to sample the wavelengths of that path.

We use the multi-sample model introduced by [Vea98] to combine the techniques for sampling a path with k vertices with the following weighing function:

$$w_{s,t}(\mathbf{x}\lambda) = \frac{p(\mathbf{x}_{s,t}\lambda)}{\sum_{i=2}^{k-1} p(\mathbf{x}_{i,k-i}\lambda)}. \quad (6.15)$$

Our implementation only includes paths with at least 2 camera vertices, so the sum in the denominator needs to start for $i = 2$. If paths with 1 or 0 camera vertices can be sampled as well, the sum needs to start at 1 or 0.

The forward and backward product in the directional probability p_A end at adjacent vertices x_{s-1} and x_s at the connecting segment. In contrast, the forward and backward products in the wavelength probability p_Λ end before the same vertex, which will be somewhere on the light sub-path after the connecting segment for the forward reset algorithm, and somewhere on the camera sub-path before the connecting segment for the backward reset algorithm.

Implementation details

The directional probabilities can be computed and accumulated for all segments on both the camera and light sub-path while the paths are created. Those values can be used in the MIS weight without further modification, because they remain unchanged even if some wavelengths are modified.

The wavelengths probabilities can also be computed and accumulated before the paths are connected, but some of those values will no longer be valid after the walk along the light or camera sub-path resets some wavelengths, depending on which reset algorithm is used. If the forward reset algorithm is applied, it might still be useful to compute the original values for the light sub-path, because especially if we connect to a longer light sub-path and if the reset algorithm stops before reaching the light source, the wavelength probabilities between $x_{s'}$ and the light source are still valid. The same thing applies to long camera sub-paths if the backward reset algorithm is applied.

The weight function $w_{s,t}$ can also be written using probabilities measured over projected solid angle p_{σ^\perp} . By substituting

$$p_A(x_i \rightarrow x_j) = p_{\sigma^\perp}(x_i \rightarrow x_j) G(x_i \leftrightarrow x_j) \quad (6.16)$$

we can write $w_{s,t}$ as

$$w_{s,t}(\mathbf{x}\lambda) = \frac{p_{\sigma^\perp,\Lambda}(\mathbf{x}_{s,t}\lambda)/G(x_{s-1} \leftrightarrow x_s)}{\sum_{i=2}^k p_{\sigma^\perp,\Lambda}(\mathbf{x}_{i,k-i}\lambda)/G(x_{i-1} \leftrightarrow x_i)} \quad (6.17)$$

with $G(x_{k-1} \leftrightarrow x_k) := 1$ for convenience, since the probability for camera paths implicitly finding the light source contains a geometry term for every segment, while the probability for connected paths include all geometry terms except the one for the connecting segment.

This results in fewer multiplications with geometric terms, which can get rather small ($\sim 10^{-6}$ in our test scenes).

6.4.4. Full throughput

We just presented the three components we need to compute a path's throughput as its weighted contribution to the multi-sample estimator.

Similar to 6.17 we can cancel out some of the geometric terms in $\frac{f_j}{p_{A,\Lambda}}$ using eq. 6.16. As a result, we can ignore some multiplications by the geometry term. For paths that were created by connecting to a light path or vertex ($t > 0$) we get

$$\begin{aligned} \text{throughput}(\mathbf{x}_{s,t}\boldsymbol{\lambda}) &= w_{s,t}(\mathbf{x}\boldsymbol{\lambda}) \cdot \frac{f_j(\mathbf{x}\boldsymbol{\lambda})}{p(\mathbf{x}_{s,t}\boldsymbol{\lambda})} \\ &= w_{s,t}(\mathbf{x}\boldsymbol{\lambda}) L_e(x_{k-1} \xrightarrow{\lambda_{k-2}} x_{k-2}) \cdot \frac{G(x_{s-1} \leftrightarrow x_s) \prod_{i=1}^{k-2} f_r(x_{i+1} \xrightarrow{\lambda_i} x_i \xrightarrow{\lambda_{i-1}} x_{i-1}) \cdot W_e(x_1 \rightarrow x_0)}{p_{\sigma^\perp}(\mathbf{x}_{s,t}) p_\Lambda(\mathbf{x}_{s,t}\boldsymbol{\lambda}) \cdot p_A(x_0)}, \end{aligned} \quad (6.18)$$

and for paths that were sampled as camera paths randomly finding a light source ($s > 0$) we get

$$\begin{aligned} \text{throughput}(\mathbf{x}_{k,0}\boldsymbol{\lambda}) &= w_{k,0}(\mathbf{x}\boldsymbol{\lambda}) \cdot \frac{f_j(\mathbf{x}\boldsymbol{\lambda})}{p(\mathbf{x}_{k,0}\boldsymbol{\lambda})} \\ &= w_{k,0}(\mathbf{x}\boldsymbol{\lambda}) L_e(x_{k-1} \xrightarrow{\lambda_{k-2}} x_{k-2}) \cdot \frac{\prod_{i=1}^{k-2} f_r(x_{i+1} \xrightarrow{\lambda_i} x_i \xrightarrow{\lambda_{i-1}} x_{i-1}) \cdot W_e(x_1 \rightarrow x_0)}{p_{\sigma^\perp}(\mathbf{x}_{k,0}) p_\Lambda(\mathbf{x}_{k,0}\boldsymbol{\lambda})}. \end{aligned} \quad (6.19)$$

6.5. Alternative: Evaluating paths for all wavelengths

Instead of sampling a new wavelength for every fluorescent surface this section shows how to evaluate paths for the full spectrum. Since the visible spectrum consists of an infinite number of wavelengths, actually evaluating every single wavelength is theoretically impossible. However, spectral reflectance data and absorption/emission spectra are typically measured for a finite number of wavelength bins, ranging in width from 0.1 nm to 10 or even more nm. If such spectra are used in a rendering system, and we assume that the spectrum represented by an array of bins is piecewise constant, the full spectral throughput can be approximated with the rectangle or midpoint rule by evaluating each spectrum for a finite number of wavelength bins.

The main advantage over the reset algorithms proposed earlier is that no wavelengths need to be sampled. The path probability and MIS weights only contain directional probabilities, and for scenes with narrow spectra we have a higher chance of sampling paths with a non-zero contribution.

The disadvantage is that it is very similar to RGB rendering, except that instead of evaluating for a red, green and blue channel the path is evaluated for a bigger number of spectral bins. So while this approach can simulate fluorescence, it is not able to render effects such as wavelength dependent refraction or media with wavelength dependent scattering coefficients. The limitations of this approach are discussed and formalized in 4.7.

Algorithm 6 evaluates the spectral component of the throughput for a path with k vertices using the rectangle rule, where r_i, c_i, a_i, Φ_i and e_i refer to the non-fluorescent reflectance spectrum r , fluorescent concentration parameter c , fluorescent absorption spectrum a , quantum yield Φ and fluorescent emission spectrum e at vertex x_i . It illustrates how the energy distribution over wavelengths changes for fluorescent surfaces. The pseudocode does not include other necessary multiplications, such as factors due to the BRDF (e.g. $\frac{1}{\pi}$ for lambert BRDFs), any geometry terms or pdfs; its purpose is to show how we can integrate the spectral component of the throughput with the rectangle rule.

For piecewise constant spectra, encoded as an array or vector of wavelength bins, the multiplications are element-wise multiplications of the bin values. The integration is a summation of all the bin values weighted by the bin widths.

We initialize the accumulated spectrum t at the light source with the light source's emission spectrum. For every non-fluorescent surface, we multiply the accumulated spectrum with the surface's non-fluorescent reflectance spectrum r . When we hit a fluorescent surface, we need to consider both the non-fluorescent reflected light, and the light that is absorbed and re-emitted by the spectral component. In order to compute the absorbed light we need to integrate the accumulated spectrum scaled with the absorption spectrum and concentration parameter. We use the rectangle rule for this integration and simply add up all the bin values of the accumulated spectra, multiplied by the bin widths.

Due to the integral in line 4 we need to compute the throughput starting at the light source and following the path towards the camera. It is no longer possible to simply accumulate the throughput for a camera path while the path is sampled, and then multiply it to the accumulated throughput of a light sub-path or a light source's emission spectrum - at least not without some additional considerations to account for the integrals. For our implementation it seemed more convenient to simply compute the full throughput from light to camera after the sub-paths are connected.

Algorithm 6 Compute full spectral energy carried along a path

```

1:  $t(\lambda) \leftarrow x_{k-1}.\text{emission}(\lambda)$ 
2: for  $i = k - 2$  to 1 do
3:   if  $x_i$  is fluorescent then
4:      $t(\lambda) \leftarrow (1 - c_i \cdot a_i(\lambda) \cdot t(\lambda)) \cdot r_i(\lambda) + \int_{\Lambda} c_i \cdot a_i(\lambda) \cdot t(\lambda) d\lambda \cdot \Phi_i \cdot e_i(\lambda)$ 
5:     ...
6:   else
7:      $t(\lambda) \leftarrow t(\lambda) \cdot r_i(\lambda)$ 
8:     ...
9:   end if
10: end for

```

7. Results

In this chapter we present some images containing fluorescent surfaces. The surfaces are modeled with the diffuse BBRRDF described in 5. Since our goal was to have a bidirectional path tracer render fluorescence, we did not use any other complex surface models. All light sources are diffuse lambert area emitters. The other surfaces are diffuse lambert BRDFs with a reflectance spectrum. Note that the non-fluorescent surfaces can also be interpreted as a special case of fluorescent surfaces with $c = 0$, but for a better distinction we will simply refer to them as non-fluorescent. There are no participating media or textures, no animations, depth of field, or other lens effects; i.e. our camera model is a pinhole camera.

7.1. Scene: Room with fluorescent walls

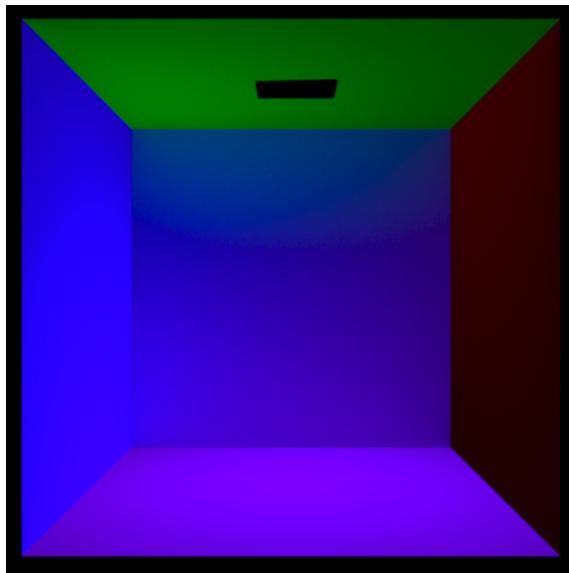


Figure 7.1.: Empty box with UV light source and fluorescent walls, ceiling and floor.

This scene (figure 7.1) shows how multiple fluorescent surfaces can interact with each other. The spectra in this scene (figure 7.2) were designed manually so that the emission spectrum of one material would overlap with the absorption spectrum of the next material. The rectangle shape of the spectra is not realistic, but was helpful with debugging as well

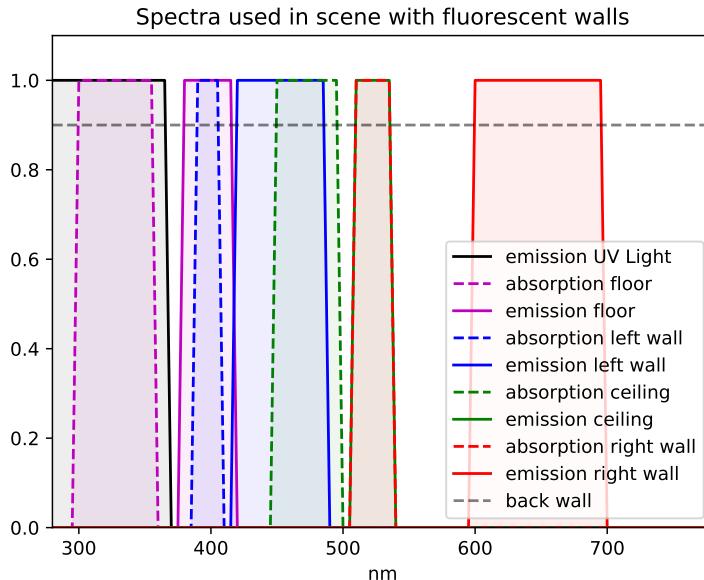


Figure 7.2.: Absorption and emission spectra of the left and right wall, ceiling and floor, as well as the light source's emission spectrum and back wall's reflectance spectrum.

as making sure only certain spectra overlap. The same overlapping effect can still occur with real materials, although it would be less intense and harder to control. We also made sure that the width of the rectangles as well as the distances between corresponding absorption and emission spectra are not too far off realistic spectra.

7.1.1. Scene Analysis

Wall Gradients

Since the light source's UV light can only interact with the back wall and the floor, the left wall is only illuminated by the floor, the ceiling is only illuminated by the left wall, and the right wall is only illuminated by the ceiling¹. As a result, the lower part of the left wall is brighter than its top part (and the left part of the ceiling is brighter than its right side, and so on). This effect is hard to notice in the images in 7.6, but is more obvious in a decent image viewer² and is clearly visible in scanline plots of pixel rows or columns: Figures 7.3a, 7.3b and 7.3c show how the side walls and the ceiling are illuminated by the previous surface. In contrast, the floor is illuminated by the light source above (figure 7.3d) and therefore has an almost symmetrical gradient. The dotted cyan line represents the values of the same pixel row backwards and reveals a slight offset, due to the fact that the light source is not exactly in the center of the scene.

¹To be more precise, each side wall and ceiling are also illuminated by light that is reflected off the back wall after being emitted by the previous surface.

²The output of our renderer are .exr files, so it is easier to evaluate the different walls separately with different exposures etc. The images displayed here have been converted to .png, so some of the detail was lost.

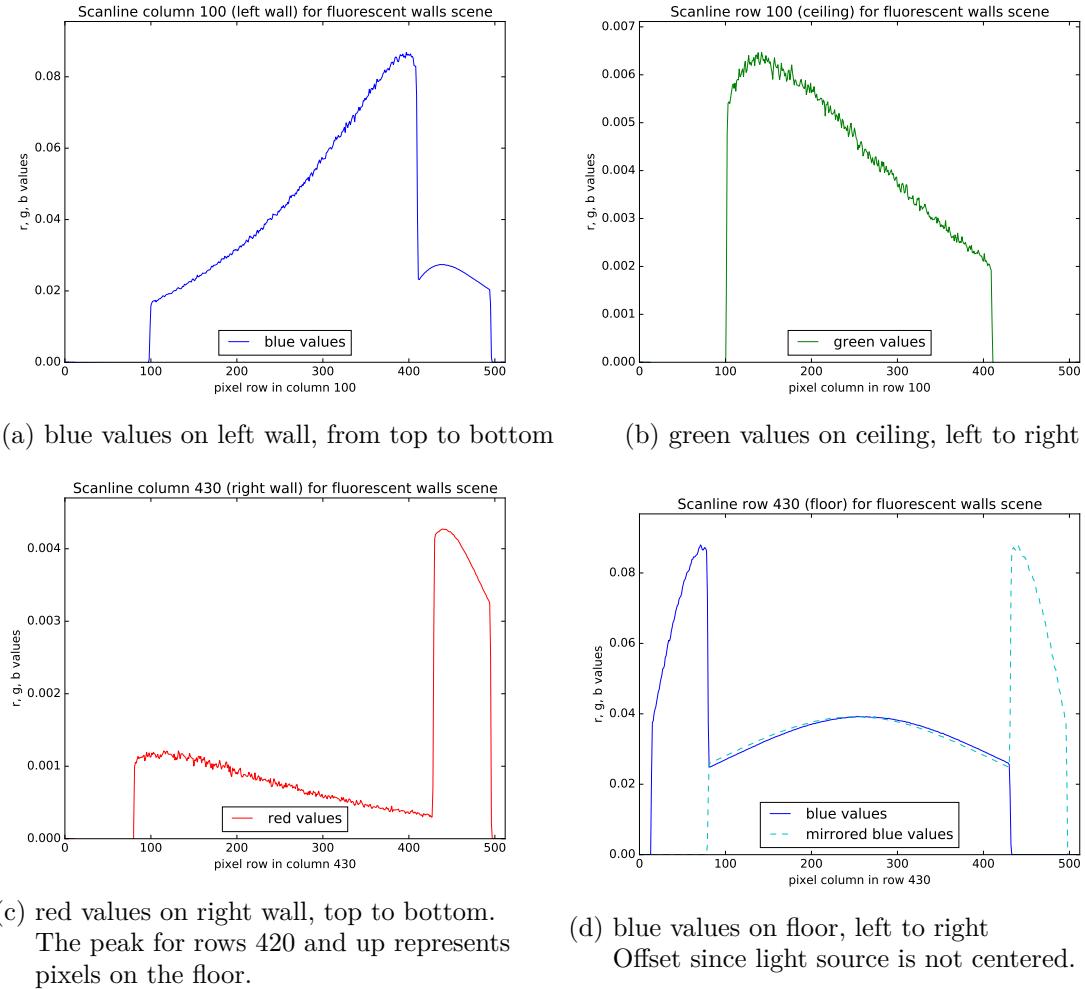


Figure 7.3.: Scanlines /-rows for the fluorescent walls scene.

Path lengths

One interesting feature of those artificial spectra is that we can tell exactly which number of fluorescent interactions we need to have on paths in distinct regions of the image, and how many vertices we can have on those paths. Figure 7.4 contains images that were restricted to one path length.

For example, the paths contributing to the pixels on the floor have either 3 or 4 vertices (figure 7.5, and figures 7.4a, 7.4b) : They start at the light source, may optionally reflect off the back wall, then hit the floor and finally the camera. If they include any other wall before hitting the floor for the last time, the UV light is lost completely. The paths need to have exactly one fluorescent interaction on the floor, where the light source's UV light is absorbed and re-emitted as visible light with short wavelengths (blue/purple).

Similarly, the paths on the left wall have 4 (fig. 7.4b), 5 (fig. 7.4c) or 6 (fig. 7.4d) vertices (light source - (optional back wall) - floor - (optional back wall) - left wall - camera) and need to have two fluorescent interactions: One on the floor, and one on the left wall, where the floor's light is absorbed and then re-emitted at slightly longer wavelengths.

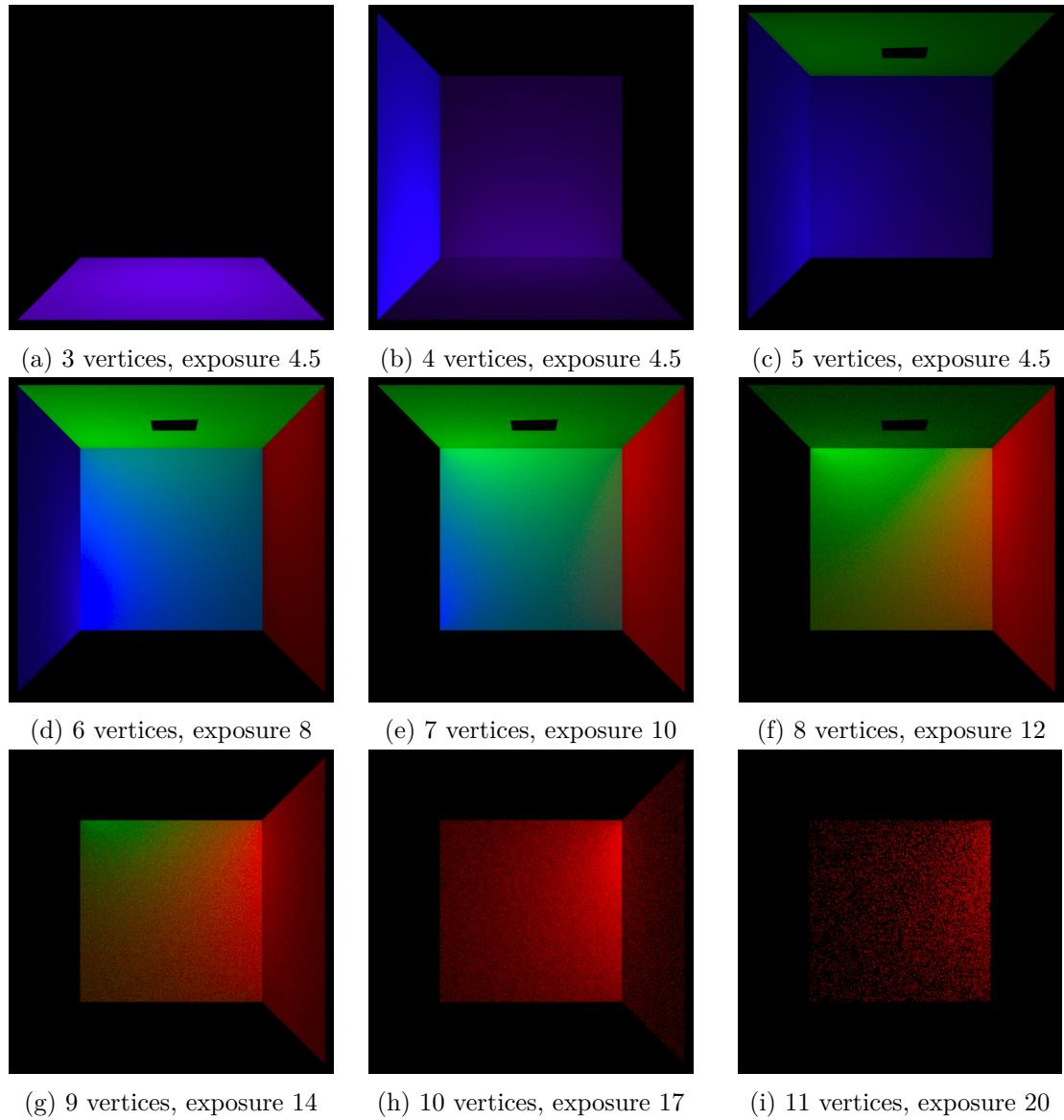


Figure 7.4.: Contributions of different path lengths. Rendered with full spectral evaluation and 10,000 samples per pixel. The exposure was increased for longer paths.

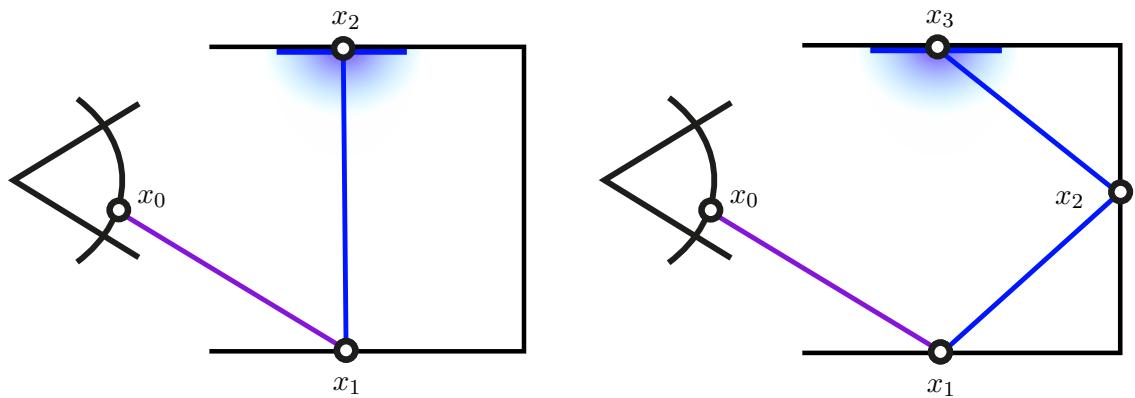


Figure 7.5.: All possible paths contributing to the pixels on the floor.

7.1.2. Wavelength importance sampling vs. uniform wavelength sampling

We presented two distinct strategies for evaluating paths in our bidirectional path tracer: Either sample new wavelengths whenever we hit a fluorescent surface (Monte Carlo integration of the spectrum, see 6.4), or evaluate the full spectral throughput by piecewise linear integration of the accumulated spectrum (rectangle rule, see 4.7, 6.5). In the first case we can sample new wavelengths uniformly, or use importance sampling to potentially increase the contribution of the sampled wavelengths. For importance sampling we chose a pdf that was proportional to the emission (absorption) spectrum on light (camera) paths. Figure 7.6 shows the advantage of importance sampling new wavelengths over sampling wavelengths uniformly.

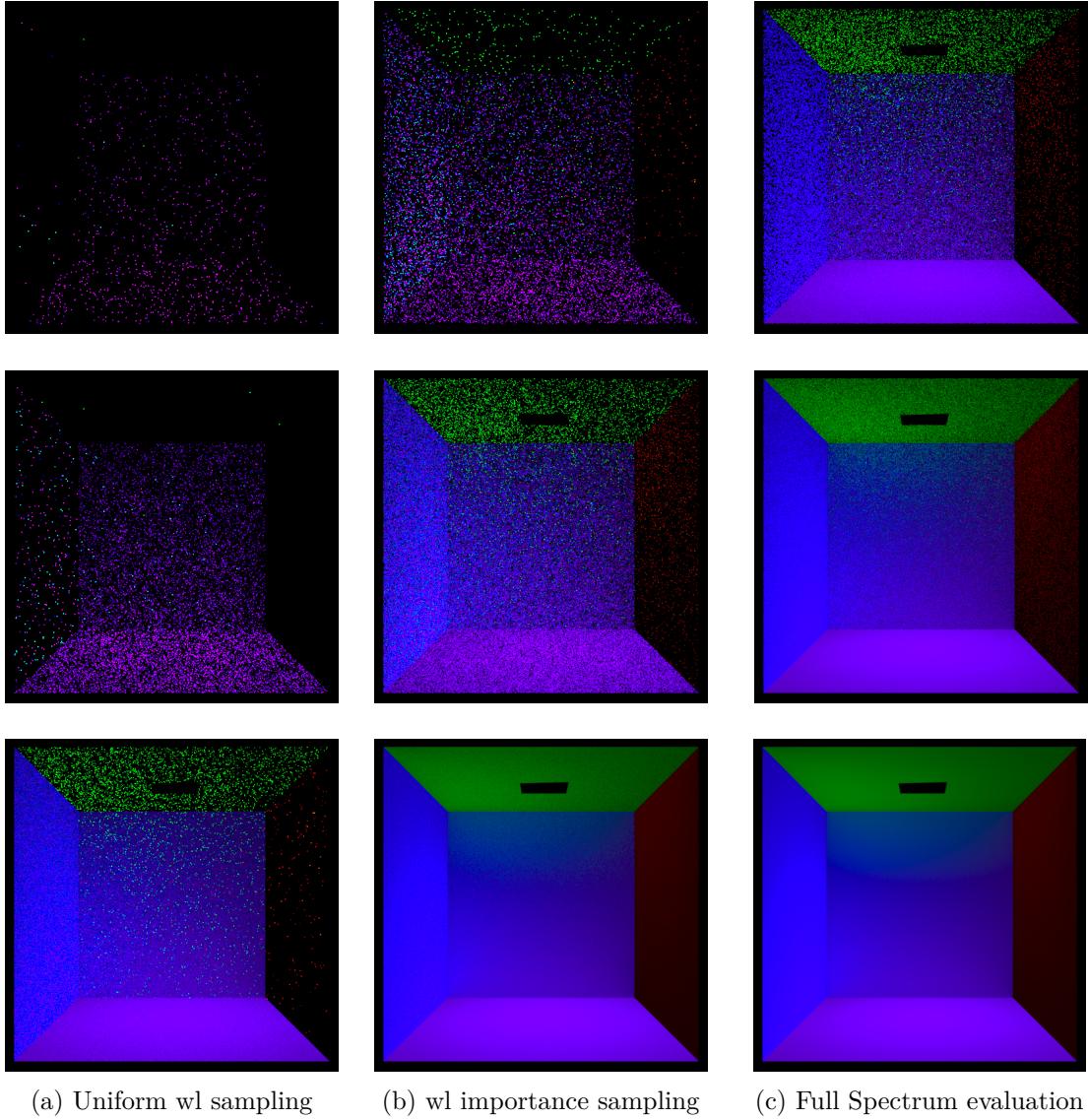


Figure 7.6.: Top: 1 sample per pixel. Middle: 16 samples. Bottom: 10,000 samples.

On the floor, we have paths that have exactly one fluorescent interaction: The paths start at the light source, may or may not hit the back wall, interact with the floor and hit the camera³. For those paths, wavelength importance sampling is better than uniform wavelength sampling, although uniform wavelength sampling will also converge to a smooth

³Note that paths that do not have a fluorescent interaction with the floor will have a 0 contribution for all visible light and will therefore not contribute to the image, since the light source only emits UV light

floor eventually. Note that even if the spectra used in this scene are artificial, for this comparison using real spectra wouldn't make much of a difference: While real absorption and emission spectra look more like a smooth curve and less like a rectangle, they are still pretty narrow, covering a bandwidth of only a few dozen nanometers. Therefore importance sampling wavelengths from real spectra on paths with one fluorescent interaction will have a similar advantage over uniform wavelength sampling.

On the wall, where paths that contribute to the image need to have two fluorescent interactions, the advantage of wavelength importance sampling increases dramatically, and gets even bigger on the ceiling (3 fluorescent interactions) and right wall (4 fluorescent interactions). Figure 7.6 shows that with uniform wavelength sampling, image areas requiring paths with more than 1 fluorescent interaction converge extremely slowly.

Figure 7.7a shows the computation time per sample per pixel for uniform wavelength sampling and wavelength importance sampling. At first glance it may seem like sampling a wavelength from a spectrum is more time consuming than sampling a wavelength uniformly. However, note that when wavelengths are sampled uniformly, the probability that the new wavelength lies outside the current absorption/emission spectrum is relatively high, which means that many paths are terminated early on because their contribution is 0. Figure 7.8 compares the distribution of path lengths. Figure 7.8b indicates that uniform wavelength sampling terminates paths earlier, and figure 7.8c shows that the absolute number of contributing paths is much higher for wavelength importance sampling, even for paths with few vertices. Considering this, wavelength importance sampling introduces almost no overhead, but converges much faster than uniform wavelength sampling.

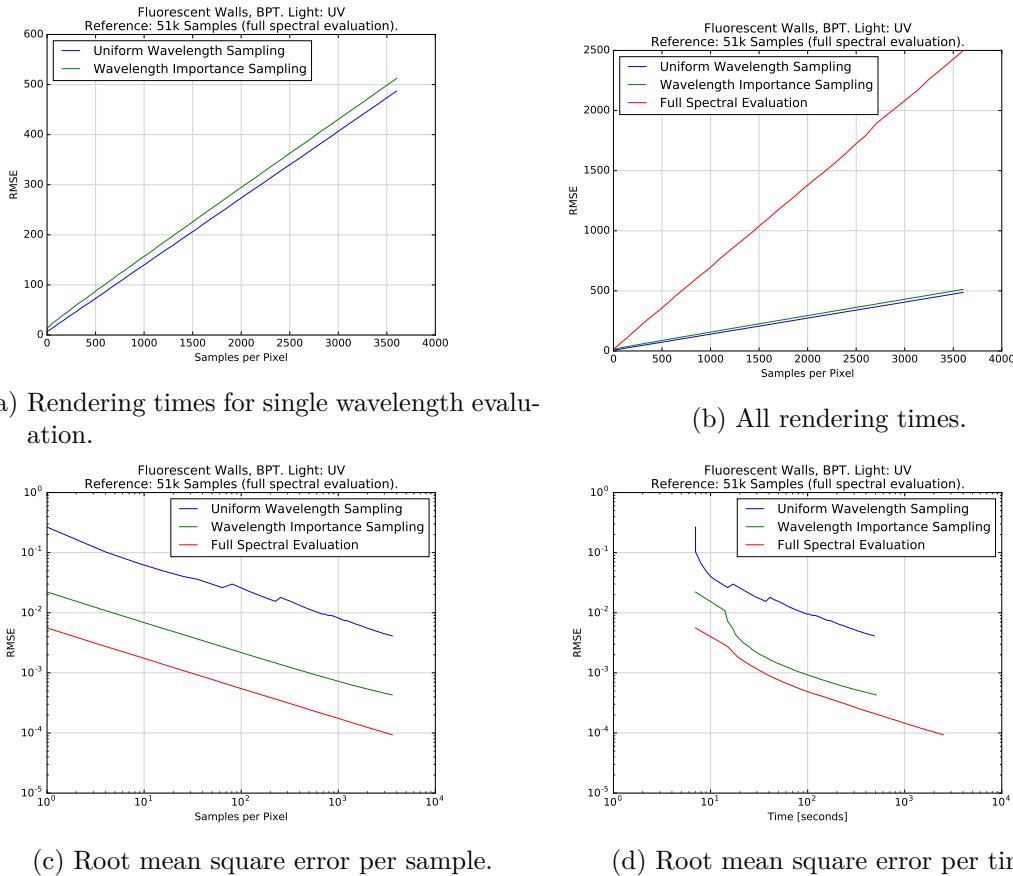
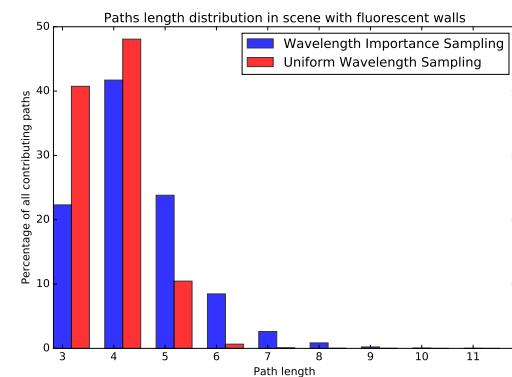
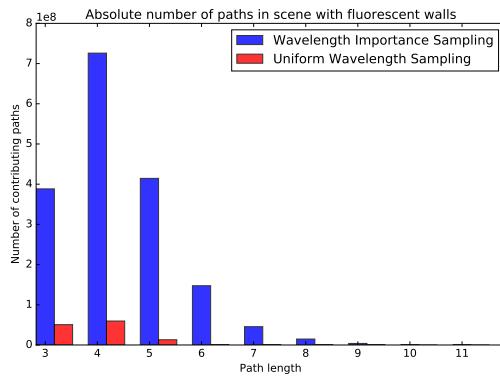


Figure 7.7.: Error and time measurements for scene with fluorescent walls.

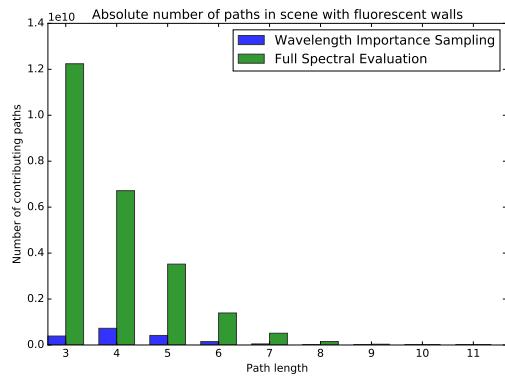
Technique	paths
Uniform sampling	1,239,504
Importance sampling	17,407,150
Full evaluation	245,663,774

(a) Absolute number of contributing paths with 1024 samples per pixel



(b) Distribution of contributing path lengths

(c) Absolute number of contributing path lengths for 1024 samples per pixel.



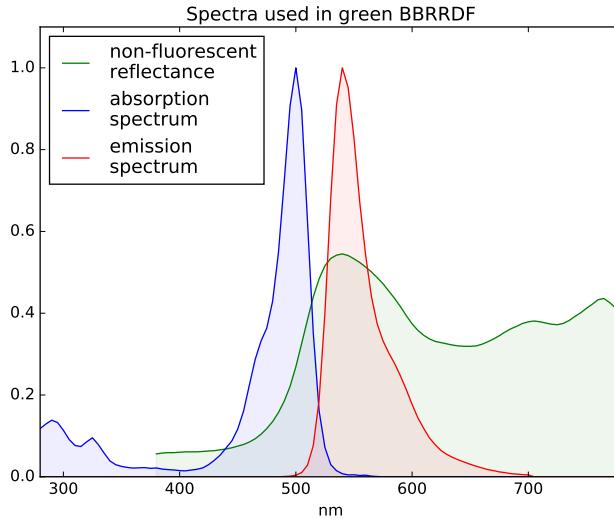
(d) Absolute number of contributing path lengths for 1024 samples per pixel.

Figure 7.8.: Depending on which technique is used, we get a different absolute number of contributing paths (a, c and d) as well as a different distribution of path lengths (b).

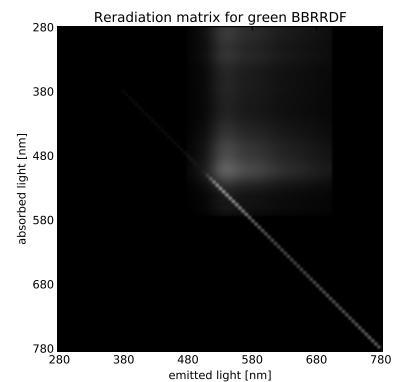
In this scene, evaluating the full spectrum for each path takes about 5 times as long as evaluating the paths for random wavelengths (see 7.7b). Figure 7.8d indicates that this overhead is not exclusively caused by the additional evaluation time for a single path, but also by the larger number of contributing paths that need to be evaluated. And as figure 7.7d shows, at least in this scene the overhead is worth it - images rendered with full spectral evaluation have a lower error not only with the same sample count, but also with equal computation time.

7.2. Scene: Green Buddhas / Spheres

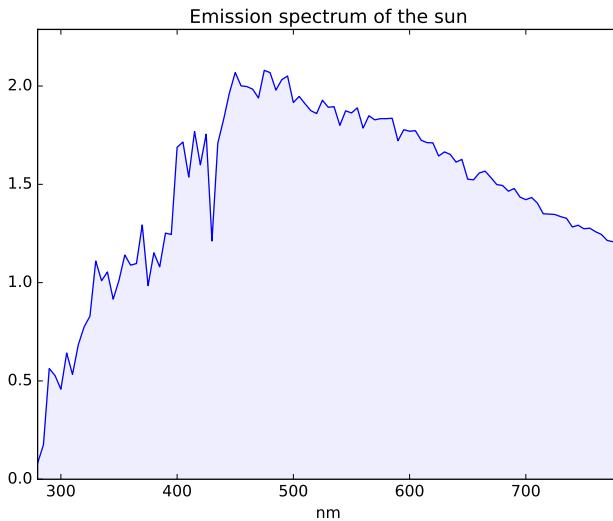
This scene contains three different materials: The light source on the ceiling is a lambert emitter with the sun's emission spectrum (see figure 7.9c). The buddhas or spheres are fluorescent, the spectra used in their BBRRDF are displayed in figure 7.9a. In this example we used real spectra: The fluorescent component of the BBRRDF is supposed to be fluorescein, and the non-fluorescent component is supposed to be the yellow-green color checker square. The Walls have a non-fluorescent diffuse BRDF with a constant spectrum of 0.8 and also reflect ultraviolet light.



(a) The non-fluorescent reflectance spectrum is that of the yellow-green color checker square. The absorption and emission spectrum belong to fluorescein ([tug, Substance 461], fig. 2.2) and overlap slightly.



(b) Corresponding reradiation matrix for the spectra in 7.9a. Fluorescent entries were scaled up by taking their 4th root.



(c) "(solar spectrum at top of atmosphere) at mean Earth-Sun distance." [ast]



(d) The yellow-green color checker square with RGB = (157,188,64) [col]

7.2.1. Fluorescence can produce high saturations

In this example we used $c = \Phi = 1$ for the left buddha's fluorescent BBRRDF to produce the highest possible fluorescent effect. The right buddha has a regular, non-fluorescent diffuse BRDF with the same reflectance spectrum (yellow-green color checker spectrum, figure 7.9d). Even in the converted .png image displayed in figure 7.10a, the left buddha appears brighter and more saturated. Figure 7.10b shows the intensity of the blue channel of figure 7.10a. In this image, the fluorescent buddha appears completely black. Before the image was converted to .png to be displayed here, the blue values of the fluorescent buddha were actually negative. This means that the saturation of the fluorescent material is too high for the sRGB gamut we used for our final images and can not be displayed correctly in this color space (figure 7.11a).

Figure 7.11b presents the reflectivity of the fluorescent BBRRDF if it is illuminated with a constant light source. For wavelengths around 570nm, the BBRRDF both reflects and emits so much light, that the overall exitant light is more than the incident light for that wavelength, which cannot happen with non-fluorescent surfaces.

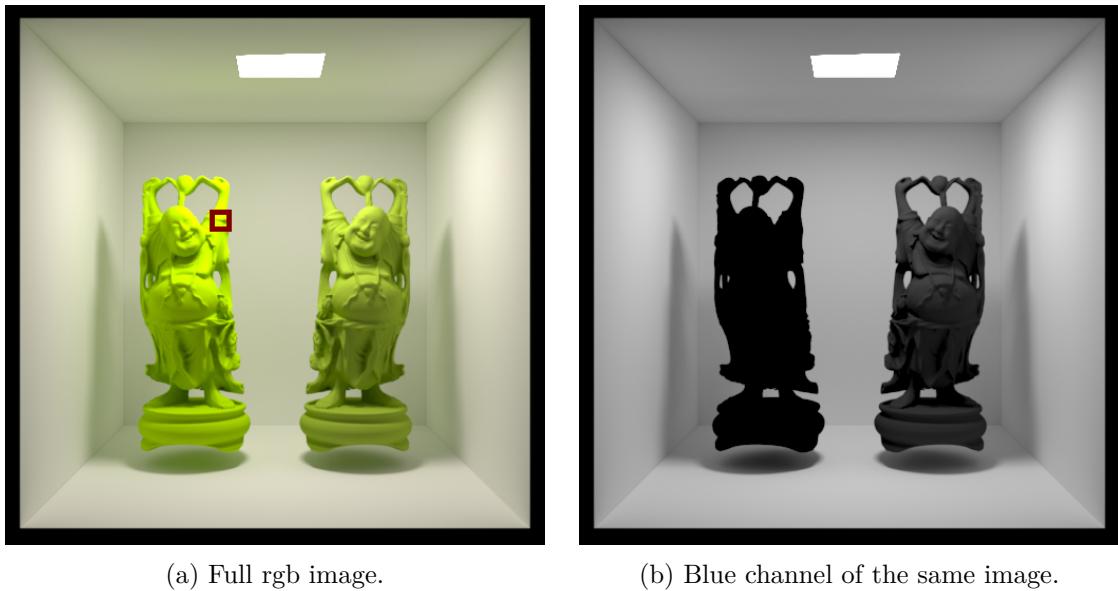
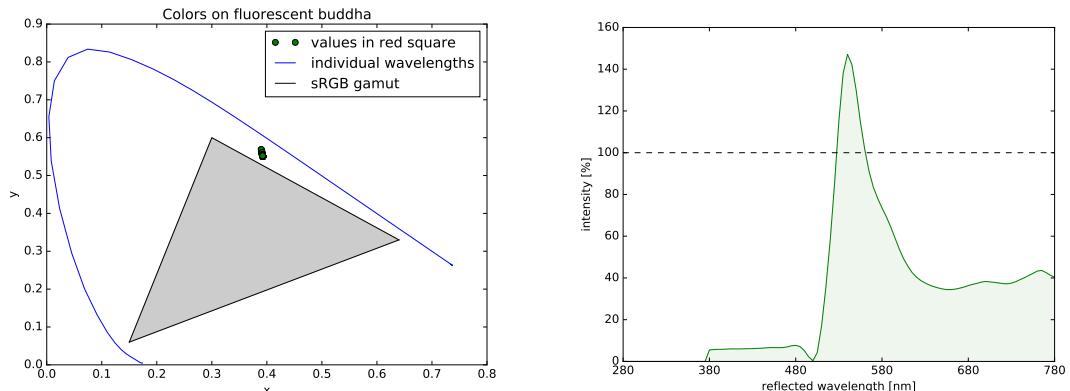


Figure 7.10.: The left buddha is fluorescent, the right buddha is not. Both buddhas have the same non-fluorescent reflectance spectrum.



7.2.2. Comparison of different parameters

In this section we keep the spectra from figure 7.9a, but vary the concentration parameter and quantum yield of the fluorescent material. We used spheres instead of buddhas, because they have smoother shadows, so hopefully the impact of the varying parameters is more obvious.

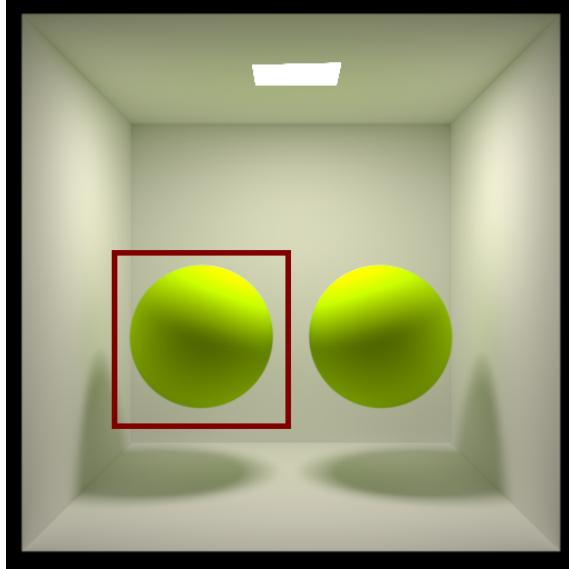


Figure 7.12.: Scene for which we are going to vary c and Φ . Both spheres are fluorescent, figure only shows the left sphere.

Figure 7.13 displays the same area of the image in figure 7.12 with different c, Φ parameters for the sphere's fluorescent surface. Note how the images in the top row, where $c = 0$, are all the same. This is to be expected - if $c = 0$, the BBRRDF's fluorescent component will never absorb any light, so varying the quantum yield, which only influences what happens to absorbed light, will not result in a different image.

Towards the lower right, the fluorescent quantum yield as well as the concentration parameter increase, thus giving the sphere a brighter, more saturated appearance.

On the bottom left $c = 1, \Phi = 0$ the sphere is actually darker than on the top left, since the fluorescent component absorbs some of the incident light without reemitting any, thus taking energy away from the light that can be reflected by the non-fluorescent component.

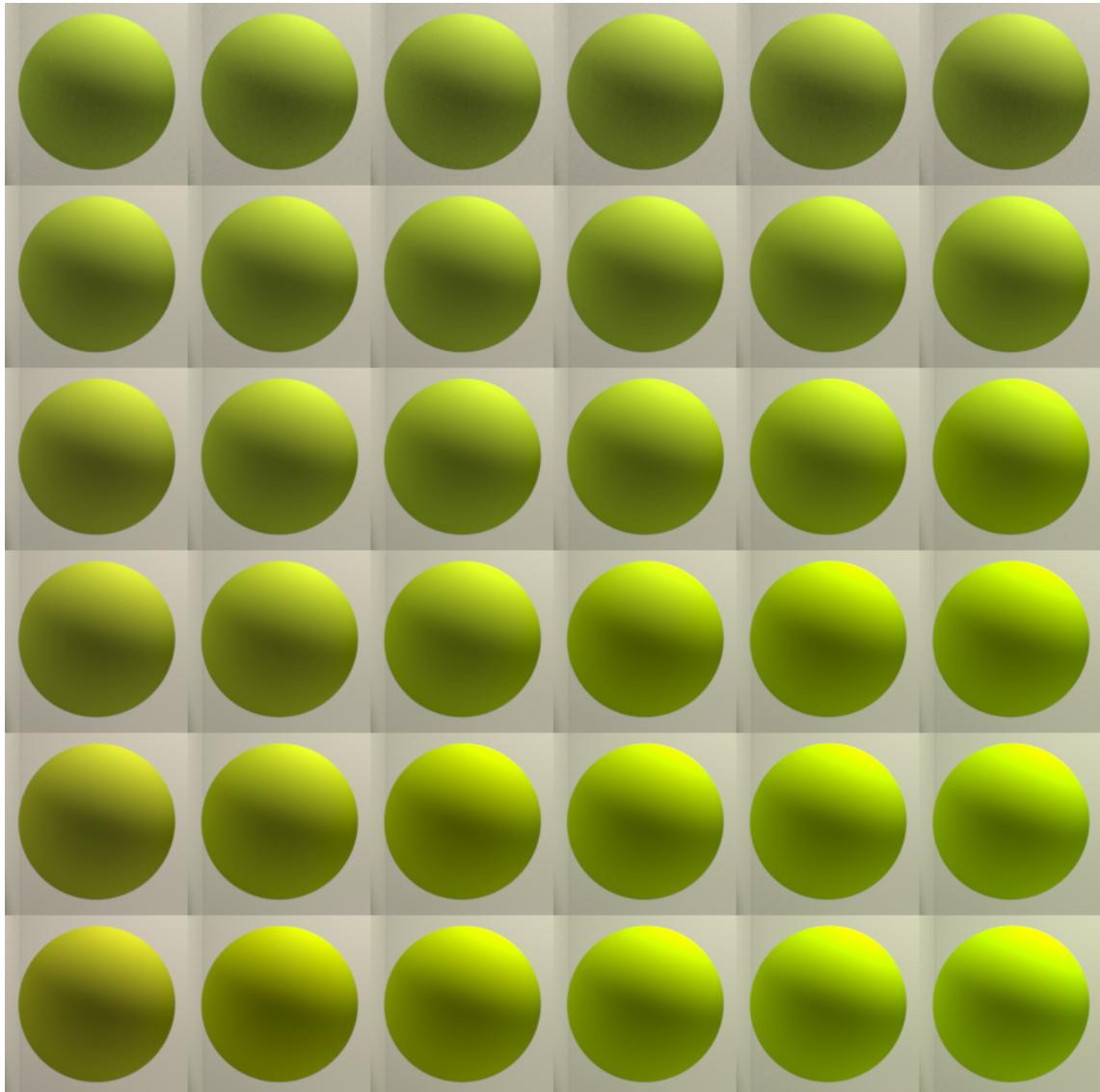


Figure 7.13.: From left to right: $\Phi = 0, 0.2, 0.4, 0.6, 0.8, 1.0$.
From top to bottom: $c = 0, 0.2, 0.4, 0.6, 0.8, 1.0$.

7.3. Additional Measurements

	Uniform	Importance
Time for sampling 10^9 wavelengths	16s	101s
No. of sampled wavelengths for 34 samples	15,956,202	15,575,460
No. of paths for 34 samples	130,082,442	135,654,750
Time for rendering 34 samples	5min 40s	5min 50s
Time for 34 samples, no fluorescence	5min 33s	5min 33s

Figure 7.14.: Measurements on scene with fluorescent spheres, comparing uniform wavelength sampling and wavelength importance sampling.

The first row in table 7.14 shows the time spent on sampling wavelengths with different strategies. The measurement was created by isolating the sampling code and drawing 10^9 samples, first uniformly in [280, 780], then from one varying spectrum with 101 values, representing 5nm-bins in [280, 780]. Note that this does not include the sampling of whether or not to have a fluorescent interaction, since this needs to be done in both cases.

The numbers show that per sampled wavelength, wavelength importance sampling is considerably more expensive than uniform wavelength sampling. However, we will now discuss why this number is still negligible compared to the overall computation time for an actual image. We chose to measure the following values from the scene with the fluorescent spheres (figure 7.12), since it contains both fluorescent and non-fluorescent surfaces and uses real spectra.

We also measured how many wavelengths were actually needed for rendering a scene. Row 2 in 7.14 shows that for a low number of samples this value is on the order of 10^7 , which is already two orders of magnitude less than what we measured before. Note that this number increases with a higher percentage of fluorescent surfaces in the scene, and can get as low as the number of sampled paths when there are no fluorescent surfaces.

Now, if we consider the time spent on rendering the same image, once with uniform wavelength sampling (340s), and once with wavelength importance sampling (350s), we can see that the time spent on sampling the required number of wavelengths is less than 1% of the overall rendering time for both techniques.

Figure 7.15 shows the number and distribution of path lengths for the same sample count. Considering this, the time overhead for rendering images with wavelength importance sampling is more likely to be caused by the evaluation of the higher number of contributing paths, and not by the very small additional time spent on sampling wavelengths.

Finally, the last row shows the computation time for the same scene and sample count, but without fluorescence (the BRDF of the spheres uses the same non-fluorescent reflectance spectrum as before, see figures 7.9a, 7.9d). When there is no fluorescence, both wavelength importance sampling and uniform wavelength sampling require the same time for rendering the scene, as is to be expected.

We also measured the time spent on sampling whether or not to have a fluorescent interaction. This code needs to be executed for both uniform wavelength sampling and wavelength importance sampling. We isolated the code and measured 10^9 iterations in the same manner we measured how long it takes to actually sample a new wavelength, with the spectra from figure 7.9a. The code for sampling whether to have a fluorescent interaction on light paths is faster (49s) than for camera paths (62s). However, both cases take longer than actually sampling a new wavelength, which contributes to the general overhead of rendering the same scene with fluorescence over rendering it without fluorescence.

In conclusion, adding fluorescent surfaces introduces multiple sources of overhead: Sampling whether or not to have a fluorescent interaction, actually sampling new wavelengths, and potentially longer / more contributing paths. In comparison to the rendering time with no fluorescent surfaces, this overhead is noticeable but relatively small.

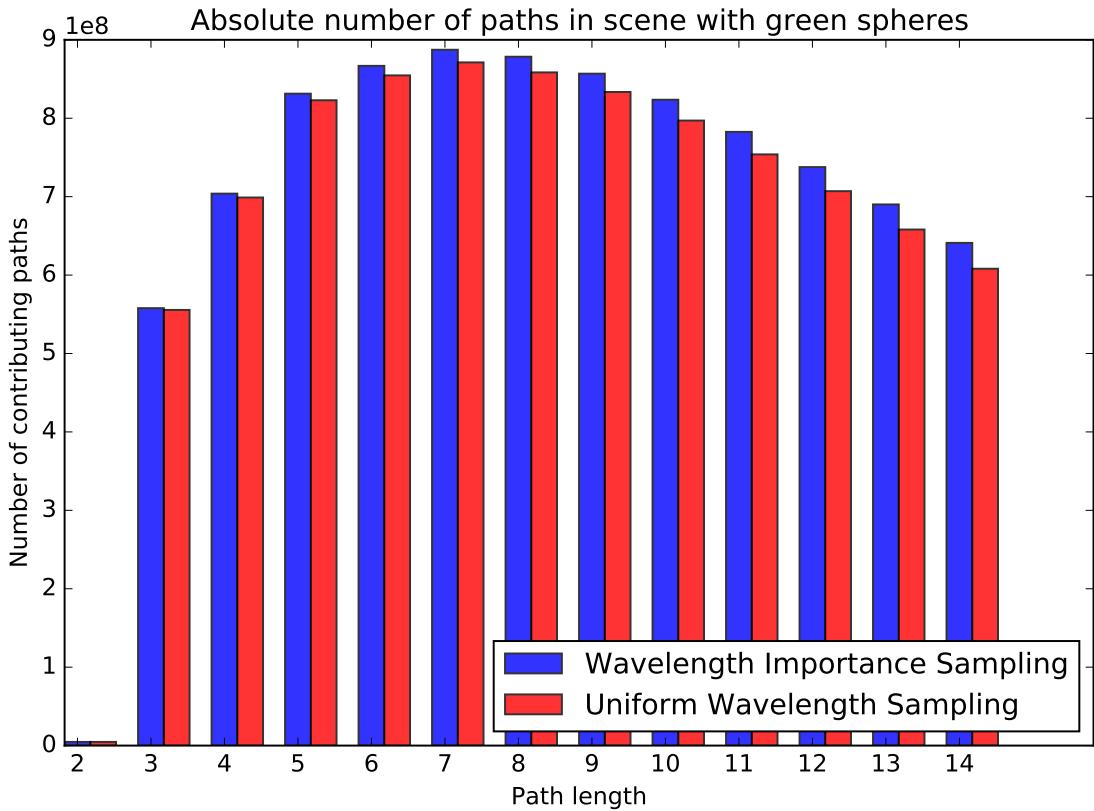


Figure 7.15.: Absolute number of contributing paths in the scene with green spheres. Uniform wavelength sampling and wavelength importance sampling have a similar distribution of path lengths, but wavelength importance sampling produces a slightly higher number of contributing paths.

8. Conclusion

The goal of this thesis was to adapt a spectral bidirectional path tracer so it can simulate fluorescent surfaces. In order to do that we first introduced a simple BBRRDF model for surfaces consisting of a fluorescent and non-fluorescent component, both of which are diffuse. The parameters for that model can be based on physical measurements of fluorescent substances and non-fluorescent reflectance data. We then showed that if a bidirectional path tracer treats such surfaces as a forward path tracer would, by changing the constructed path's wavelength whenever a fluorescent surface is hit, connections between camera and light sub-paths become invalid and their measurement contribution is not well-defined anymore.

As a solution we proposed two ideas. First was the forward reset algorithm, which starts by sampling a camera and light sub-path with independent wavelengths and changes a path's wavelength when it hits a fluorescent surface, and then modifies some of the wavelengths along the light path to make its wavelengths consistent with the camera path's wavelengths. As an alternative we introduced the backward reset algorithm, which works similarly but modifies wavelengths along the camera path.

The second approach was to not use Monte Carlo integration for evaluating the contribution of a path over all wavelengths, but instead apply the rectangle rule, a simple numerical integration method. We based this on the observation that usually the spectra used to describe surfaces in a scene are not provided as a continuous function, but as discrete values for a set of wavelength bins. The resulting algorithm samples the camera and light sub-path independent of each other and independent of a wavelength, and then evaluates the path's contribution for all wavelength bins.

We were thus able to simulate fluorescence in a bidirectional path tracer, however the resulting algorithms are not able to handle some other light transport effects, such as dispersion or some media.

8.1. Limitations

Both the reset algorithms and the rectangle rule do not support surface models or media where the probability for locally sampling a new direction depends on the incident / exitant wavelength. One reason for that is that if we evaluate a path that was constructed with one wavelength in mind for another wavelength its measurement contribution may become zero. Another problem is that we cannot simply compute a probability for constructing a

path with one wavelength, and then use that probability to weight the path's contribution for another wavelength or another set of wavelength combinations.

In case of the reset algorithms it might be possible to fix paths such that they still contribute when they carry other wavelengths by modifying their vertices as well. Due to the limited timeframe we did not explore how this might be done.

The BBRRDF we proposed only models the processes that are essential for fluorescence - reflecting photons at the same wavelength or absorbing and emitting them at another - but does not replicate more complicated behaviour. Real-life fluorescent materials can look very bright or saturated under certain lighting conditions; we were able to recreate that effect with our BBRRDF using measured spectral data, where the resulting color of our renderer was outside the sRGB color space. We did not include any specularity, textures or media and did not compare the qualitative appearance of our results to real-life measurements. Since our focus was on including fluorescence as the effect of changing wavelengths into a bidirectional path tracer, this model was sufficient for our case.

If the parameters are chosen with care the BBRRDF can still match several physically measurable properties of fluorescence, such as the absorption and emission spectrum, the quantum yield, and real-life materials that show fluorescent as well as non-fluorescent reflectance at the same time. In addition, the model supports more artistic freedom, and the correlation between the BBRRDF's parameters and physical appearance allows for manual design of artificial spectra that appear in a desired color and saturation.

8.2. Future Work

There are several areas in this thesis that can be improved further:

The BBRRDF

We proposed a basic diffuse BBRRDF that models fluorescent surfaces changing wavelengths, but does not support any other surface interactions. In addition, we did not evaluate the realism of our BBRRDF by comparing it to real measurements. One field of future work might be to compare the results of the BBRRDF to real measurements, and to improve the model further. In particular, it would be interesting to investigate the accuracy of our assumption for reconstructing the full reradiation matrix from the BBRRDF's parameters.

Previous works ([WWLP06]) tried to separate the specular and diffuse component of a fluorescent BBRRDF, this might be improved further by applying a layered-materials approach ([JdJM14]).

Evaluating the full spectrum

In our simple test scenes without any complicated geometry or BRDFs integrating wavelengths with the rectangle rule was more efficient than Monte Carlo integration. This was the case not only in fluorescent areas within the image, but for all surfaces. Our implementation used a resolution of 5nm for storing spectra. The speed of the rectangle rule is directly related to the number of bins that have to be evaluated; it might be interesting to see whether a lower resolution of 10, 20 or more nm starts introducing a noticeable or measurable error, or if the quality of the resulting image remains the same while achieving even better efficiency. Especially in simple scenes without any of the excluded effects, this offers many of the effects that can only be simulated by spectral rendering, while not producing the color noise and slow convergence that is typical for a spectral renderer.

Another disadvantage of our implementation of the rectangle rule is that light and camera

sub-paths were never terminated due to a low throughput, since we only compute the throughput of full paths after connecting the sub-paths. Our throughput computation could also be done iteratively when sampling the light sub-path and used to terminate it if its throughput becomes 0 for all wavelengths. But due to the integral and delta-function in our BBRRDF we did not explore how a meaningful throughput could be accumulated for camera sub-paths in a way that would allow us to terminate them once they will not transport any energy all the way to the camera.

Vertex Connection and Merging

Vertex Connection and Merging (VCM, [GKDS12] [HPJ12]) is a technique that combines bidirectional path tracing and photon mapping. With VCM all light sub-paths are created at once (e.g. one light sub-path per pixel or camera path) before the camera paths are created. Then, the n -th camera sub-path is connected to the n -th light path, and photon mapping is performed with the photons from all light paths. In this version every camera sub-path is connected to one predetermined light sub-path.

The main difference to pure bidirectional path tracing is that we now have a big set of light sub-paths to choose from, instead of a single light path per camera path. In spectral path tracing with fluorescent surfaces, this could be useful if all light paths are started with random wavelengths, changing their wavelength at fluorescent surfaces. Then, whenever a camera path needs to be connected to a light path with t vertices, instead of simply connecting to the t -th vertex of the n -th light path, one could pick a light path with a similar wavelength at the t -th vertex and connect there.

Thus there would be no need to modify wavelengths of a path after it was sampled, and effects like dispersion or media would not interfere with simulating fluorescence anymore. However, since we can never expect to have any light path with the exact same wavelength that we need at the camera path there might be some error. In addition, there would have to be some probability for choosing the light path to which to connect. The maximum number t of light vertices that shall be connected to a certain camera path can no longer be determined by the number of vertices of one specific light path, since with that approach a camera vertex might be connected to several distinct light sub-paths of different lengths. But since several effects that are impossible with our methods would be trivial with VCM, this would be an interesting area of future work.

Bibliography

- [ast] “Reference solar spectral irradiance: Astm g-173,” accessed: 2017/05/03. [Online]. Available: <http://rredc.nrel.gov/solar/spectra/am1.5/ASTMG173/ASTMG173.html>
- [BHD⁺08] M. Bendig, J. Hanika, H. Dammertz, J. C. Goldschmidt, M. Peters, and M. Weber, “Simulation of fluorescent concentrators,” in *2008 IEEE Symposium on Interactive Ray Tracing*, Aug 2008, pp. 93–98.
- [CAE10] L. G. Coppel, M. Andersson, and P. Edström, “Quantum efficiency of fluorescent dyes in different furnishes,” 2010.
- [CE09] L. G. Coppel and P. Edström, “Open source monte carlo simulation platform for particle level simulation of light scattering from generated paper structures,” 2009.
- [col] “Average measurements of all colorchecker charts,” accessed: 2017/05/16. [Online]. Available: http://xritephoto.com/documents/literature/en/ColorData-1p_EN.pdf
- [Don54] R. Donaldsen, “Spectrophotometry of fluorescent pigments,” pp. 210–214, 1954.
- [EDH98] P. Emmel, R. David, and R. D. Hersch, “Spectral colour prediction model for a transparent fluorescent ink on paper,” in *Proceedings of the 6 th IS&T/SID Color Imaging Conference*, 1998, pp. 17–20.
- [fee] “Fluorescence excitation and emission fundamentals,” accessed: 2017/02/02. [Online]. Available: <http://www.chem.uci.edu/~dmitryf/manuals/Fundamentals/Fluorescence%20Excitation%20and%20Emission%20Fundamentals.pdf>
- [flu] “The fluorescent lamp,” accessed 2017/03/28. [Online]. Available: <http://www.edisontechcenter.org/Fluorescent.html>
- [fsm] “Fluorescencespectroscopy,” accessed: 2017/05/05. [Online]. Available: https://www.mcgill.ca/biochemistry/files/biochemistry/404_silvius_09.pdf
- [GKDS12] I. Georgiev, J. Křivánek, T. Davidovič, and P. Slusallek, “Light transport simulation with vertex connection and merging,” *ACM Trans. Graph.*, vol. 31, no. 6, pp. 192:1–192:10, Nov. 2012. [Online]. Available: <http://doi.acm.org/10.1145/2366145.2366211>
- [Gla95] A. S. Glassner, *A Model for Fluorescence and Phosphorescence*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 60–70. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-87825-1_5
- [Gui90] G. Guilbault, *Practical Fluorescence, Second Edition*, ser. Modern Monographs in Analytical Chemistry. Taylor & Francis, 1990. [Online]. Available: <https://books.google.com/books?id=7eI3AVl4dDsC>

- [HHA⁺10] M. B. Hullin, J. Hanika, B. Ajdin, H.-P. Seidel, J. Kautz, and H. P. A. Lensch, “Acquisition and analysis of bispectral bidirectional reflectance and reradiation distribution functions,” in *ACM SIGGRAPH 2010 Papers*, ser. SIGGRAPH ’10. New York, NY, USA: ACM, 2010, pp. 97:1–97:7. [Online]. Available: <http://doi.acm.org/10.1145/1833349.1778834>
- [HPJ12] T. Hachisuka, J. Pantaleoni, and H. W. Jensen, “A path space extension for robust light transport simulation,” *ACM Trans. Graph.*, vol. 31, no. 6, pp. 191:1–191:10, Nov. 2012. [Online]. Available: <http://doi.acm.org/10.1145/2366145.2366210>
- [Hul10] M. Hullin, “Reconsidering light transport: Acquisition and display of real-world reflectance and geometry,” 2010. [Online]. Available: <https://diglib.eg.org/handle/10.2312/8243>
- [idl04] C. C. internationale de l’éclairage, “Technical report: Colorimetry,” 2004.
- [int] “An introduction to fluorescence spectroscopy.” [Online]. Available: <https://www.chem.uci.edu/~dmitryf/manuals/Fundamentals/Fluorescence%20Spectroscopy.pdf>
- [JdJM14] W. Jakob, E. d’Eon, O. Jakob, and S. Marschner, “A comprehensive framework for rendering layered materials,” *ACM Trans. Graph.*, vol. 33, no. 4, pp. 118:1–118:14, Jul. 2014. [Online]. Available: <http://doi.acm.org/10.1145/2601097.2601139>
- [JM66] H. H. Jaffe and A. L. Miller, “The fates of electronic excitation energy,” *Journal of chemical education*, vol. 43, no. 9, 1966.
- [Kaj86] J. T. Kajiya, “The rendering equation,” in *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’86. New York, NY, USA: ACM, 1986, pp. 143–150. [Online]. Available: <http://doi.acm.org/10.1145/15922.15902>
- [Lak06] J. R. Lakowicz, *Principles of Fluorescence Spectroscopy*, 3rd ed., J. R. Lakowicz, Ed. Springer, 2006.
- [MSHD15] J. Meng, F. Simon, J. Hanika, and C. Dachsbacher, “Physically meaningful rendering using tristimulus colours,” *Computer Graphics Forum (Proceedings of Eurographics Symposium on Rendering)*, vol. 34, no. 4, pp. 31–40, 2015.
- [Mur96] S. G. Murray, *Dyes and fluorescent whitening agents for paper*. Dordrecht: Springer Netherlands, 1996, pp. 161–193. [Online]. Available: http://dx.doi.org/10.1007/978-94-011-0605-4_10
- [Nas01] K. Nassau, *The physics and chemistry of color*, 2nd ed. John Wiley & Sons, Inc., 2001.
- [pyt] “Colour science for python.” [Online]. Available: <http://www.colour-science.org/>
- [SBCD14] J. Suo, L. Bian, F. Chen, and Q. Dai, “Bispectral coding: compressive and high-quality acquisition of fluorescence and reflectance,” *Opt. Express*, vol. 22, no. 2, pp. 1697–1712, Jan 2014. [Online]. Available: <http://www.opticsexpress.org/abstract.cfm?URI=oe-22-2-1697>
- [SHE11] M. Sauer, J. Hofkens, and J. Enderlein, *Handbook of Fluorescence Spectroscopy and Imaging*. WILEY-VCH Verlag, 2011.

- [the] “Fluorescence spectraviewer,” accessed: 2017/03/23. [Online]. Available: <https://www.thermofisher.com/de/de/home/life-science/cell-analysis/labeling-chemistry/fluorescence-spectraviewer.html>
- [tug] “Fluorophores.org database of fluorescent dyes, properties and applications,” accessed: 2017/03/23. [Online]. Available: <http://fluorophores.tugraz.at/>
- [Vea98] E. Veach, “Robust monte carlo methods for light transport simulation,” Stanford, CA, USA, 1998, aAI9837162.
- [WTP01] A. Wilkie, R. F. Tobler, and W. Purgathofer, *Combined Rendering of Polarization and Fluorescence Effects*. Vienna: Springer Vienna, 2001, pp. 197–204. [Online]. Available: http://dx.doi.org/10.1007/978-3-7091-6242-2_18
- [WWLP06] A. Wilkie, A. Weidlich, C. Larboulette, and W. Purgathofer, “A reflectance model for diffuse fluorescent surfaces,” in *Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia*, ser. GRAPHITE ’06. New York, NY, USA: ACM, 2006, pp. 321–331. [Online]. Available: <http://doi.acm.org/10.1145/1174429.1174484>

Erklärung

Ich versichere, dass ich die Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht und die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet habe. Die Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt und von dieser als Teil einer Prüfungsleistung angenommen.

Karlsruhe, den 30. Mai 2017


(Alisa Jung)

Appendices

A. Additional Equations

A.1. Separating directional and wavelength integrations in the measurement equation

If the directional and wavelength component of all light sources and BRDFs in the scene are independent of each other, we can separate the integration over wavelengths from the integration over path space in the measurement equation. In mathematical terms, each BRDF f_r , light source L , and the sensor sensitivity function $W_e^{(j)}$ need to have a directional component $f_\omega(x \rightarrow x' \rightarrow x'')$, L_ω , $W_\omega^{(j)}$ and wavelength component $f_\lambda(\lambda_{in}, x, \lambda_{out})$, L_λ , $W_\lambda^{(j)}$ such that

$$f_r(x \xrightarrow{\lambda_{in}} x' \xrightarrow{\lambda_{out}} x'') = f_\omega(x \rightarrow x' \rightarrow x'') \cdot f_\lambda(\lambda_{in}, x, \lambda_{out}) \quad (\text{A.1})$$

$$L_e(x \xrightarrow{\lambda'} x') = L_\omega(x \rightarrow x') \cdot L_\lambda(\lambda') \quad (\text{A.2})$$

$$W_e^{(j)}(x' \xrightarrow{\lambda} x) = W_\omega^{(j)}(x' \rightarrow x) \cdot W_\lambda^{(j)}(\lambda) \quad (\text{A.3})$$

If this is the case, the measurement equation can be rewritten as follows:

$$\begin{aligned}
I_j &= \sum_{k=2}^{\infty} \int_{(\mathcal{M} \times \Lambda)^{k-1} \times \mathcal{M}} L_e(x_{k-1} \xrightarrow{\lambda_{k-2}} x_{k-2}) G(x_{k-1} \leftrightarrow x_{k-2}) \\
&\quad \cdot \prod_{i=1}^{k-2} \left(f_r(x_{i+1} \xrightarrow{\lambda_i} x_i \xrightarrow{\lambda_{i-1}} x_{i-1}) G(x_i \leftrightarrow x_{i-1}) \right) \\
&\quad \cdot W_e^{(j)}(x_1 \xrightarrow{\lambda_0} x_0) dA(x_{k-1}) d\lambda_{k-2} dA(x_{k-2}) \cdots d\lambda_0 dA(x_0) \\
&= \sum_{k=2}^{\infty} \int_{(\mathcal{M} \times \Lambda)^{k-1} \times \mathcal{M}} L_e(x_{k-1} \xrightarrow{\lambda_{k-2}} x_{k-2}) G(x_{k-1} \leftrightarrow x_{k-2}) \\
&\quad \cdot \prod_{i=1}^{k-2} (f_\omega(x_{i+1} \rightarrow x_i \rightarrow x_{i-1}) f_\lambda(\lambda_i, x_i, \lambda_{i-1}) G(x_i \leftrightarrow x_{i-1})) \\
&\quad \cdot W_e^{(j)}(x_1 \xrightarrow{\lambda_0} x_0) dA(x_{k-1}) d\lambda_{k-2} dA(x_{k-2}) \cdots d\lambda_0 dA(x_0) \\
&= \sum_{k=2}^{\infty} \int_{\mathcal{M}^k} \int_{\Lambda^{k-1}} L_e(x_{k-1} \xrightarrow{\lambda_{k-2}} x_{k-2}) G(x_{k-1} \leftrightarrow x_{k-2}) \\
&\quad \cdot \prod_{i=1}^{k-2} (f_\omega(x_{i+1} \rightarrow x_i \rightarrow x_{i-1}) G(x_i \leftrightarrow x_{i-1})) \\
&\quad \cdot \prod_{i=1}^{k-2} f_\lambda(\lambda_i, x_i, \lambda_{i-1}) \\
&\quad \cdot W_\omega^{(j)}(x_1 \rightarrow x_0) \cdot W_\lambda^{(j)}(\lambda_0) d\lambda_{k-2} \cdots d\lambda_0 dA(x_{k-1}) \cdots dA(x_0) \\
&= \sum_{k=2}^{\infty} \int_{\mathcal{M}^k} \int_{\Lambda^{k-1}} L_\lambda(x_{k-1}, \lambda_{k-2}) \cdot \prod_{i=1}^{k-2} f_\lambda(\lambda_i, x_i, \lambda_{i-1}) W_\lambda^{(j)}(\lambda_0) d\lambda_{k-2} \cdots d\lambda_0 \\
&\quad \cdot L_\omega(x_{k-1} \rightarrow x_{k-2}) G(x_{k-1} \leftrightarrow x_{k-2}) \\
&\quad \cdot \prod_{i=1}^{k-2} (f_\omega(x_{i+1} \rightarrow x_i \rightarrow x_{i-1}) G(x_i \leftrightarrow x_{i-1})) \\
&\quad \cdot W_\omega^{(j)}(x_1 \rightarrow x_0) dA(x_{k-1}) \cdots dA(x_0)
\end{aligned} \tag{A.4}$$