Chapter 8

A Path Integral Formulation of Light Transport

In this chapter, we show how to transform the light transport problem into an integration problem. This *path integral formulation* expresses each measurement in the form of a simple integral (rather than as the solution to an integral equation or operator equation, as with the other formulations we have described). More precisely, each measurement I_j is written in the form

$$I_j \;=\; \int_\Omega f_j(\bar{x})\, d\mu(\bar{x})\,,$$

where Ω is the set of *transport paths* of all lengths, μ is a measure on this space of paths, and f_j is called the *measurement contribution function* (to be defined below).

The path integral model has several benefits. The main advantage is that by reducing light transport to an integration problem, it allows general-purpose integration methods to be applied. For example, we will show how light transport problems can be solved more robustly using *multiple importance sampling* (Chapter 9), an integration method that allows several different sampling strategies to be efficiently combined.

The path integral model also leads to new techniques for sampling paths. The problem with models based on integral equations is that they only describe scattering from one surface at a time. This leads to light transport algorithms that construct paths incrementally, by recursive sampling of the integral equation. The path integral model takes a more global view, which has led directly to techniques such as bidirectional path tracing (Chapter 10) and the Metropolis light transport algorithm (Chapter 11). These new techniques can only be properly understood within the path integral framework.

Finally, the path integral model is a useful tool for understanding the limitations of unbiased Monte Carlo algorithms. It provides a natural way to classify transport paths, and to identify those that cannot be sampled by certain kinds of techniques.

This chapter is organized as follows. First, we review the *three-point form* of the light transport equations, and show how to transform them into an integral over paths. We then discuss the advantages of the path integral model in more detail, and show how it can be used to construct unbiased Monte Carlo estimators. Finally, introduce the idea of *full-path regular expressions* (extending a notation of Heckbert [1990]), and discuss the limitations of path sampling approaches to light transport.

In Appendix 8.A, we describe several other ways that the path integral model can be formulated, by introducing new measures on the space of paths. These measures have natural physical interpretations whose meanings are described.

8.1 The three-point form of the transport equations

We show how to rewrite the transport equations to eliminate the directional variables $\omega_{\mathrm{i}}, \omega_{\mathrm{o}}$. This first step is to write the equilibrium radiance in the form $L(\mathbf{x} \to \mathbf{x}')$, where $\mathbf{x}, \mathbf{x}' \in \mathcal{M}$ are points on the scene surfaces. In terms of the function $L(\mathbf{x}, \omega)$ we have been using up until now, we define

$$L(\mathbf{x} \to \mathbf{x}') \;=\; L(\mathbf{x}, \omega)$$

where $\omega \;=\; \widehat{\mathbf{x}' - \mathbf{x}}$ is the unit-length vector pointing from $\mathbf{x}$ to $\mathbf{x}'$. (This representation of the ray space $\mathcal{R}$ was described in Section 4.1; recall that it has some redundancy, since $L(\mathbf{x} \to \mathbf{x}') = L(\mathbf{x} \to \mathbf{x}'')$ whenever $\mathbf{x}'$ and $\mathbf{x}''$ lie in the same direction from $\mathbf{x}$.)

Similarly, we write the BSDF as a function of the form

$$f_{\mathrm{s}}(\mathbf{x} \to \mathbf{x}' \to \mathbf{x}'') \;=\; f_{\mathrm{s}}(\mathbf{x}', \omega_{\mathrm{i}} \to \omega_{\mathrm{o}})\,,$$

where $\omega_{\mathrm{i}} = \widehat{\mathbf{x} - \mathbf{x}'}$ and $\omega_{\mathrm{o}} = \widehat{\mathbf{x}'' - \mathbf{x}'}$. The arrow notation $\mathbf{x} \to \mathbf{x}'$ symbolizes the direction

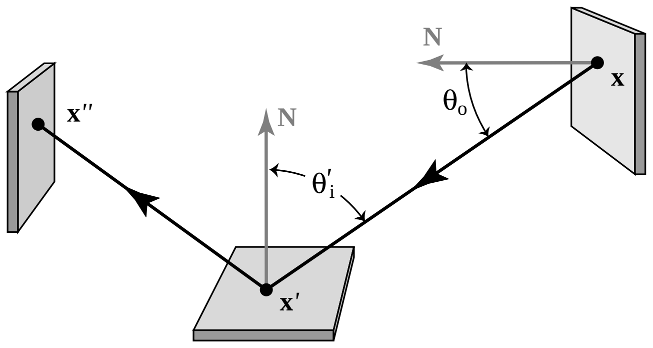

Figure 8.1: Geometry for the light transport equation in three-point form.

of light flow.

The *three-point form* of the light transport equation can now be written as

$$L(\mathbf{x}' \to \mathbf{x}'') \;=\; L_{\mathrm{e}}(\mathbf{x}' \to \mathbf{x}'') + \int_{\mathcal{M}} L(\mathbf{x} \to \mathbf{x}')\, f_{\mathrm{s}}(\mathbf{x} \to \mathbf{x}' \to \mathbf{x}'')\, G(\mathbf{x} \leftrightarrow \mathbf{x}')\, dA(\mathbf{x}) \tag{8.1}$$

(see Figure 8.1). This is simply a reformulation of the original version of the light transport equation (3.19) that we have already described. As before, $\mathcal{M}$ is the union of all scene surfaces, A is the area measure on $\mathcal{M}$, and L_{e} is the emitted radiance function. The function G represents the change of variables from the original integration measure $d\sigma^{\perp}$ to the new integration measure dA, which are related by

$$d\sigma^{\perp}_{\mathbf{x}'}(\omega_{\mathrm{i}}) \;=\; d\sigma^{\perp}_{\mathbf{x}'}(\widehat{\mathbf{x} - \mathbf{x}'}) \;=\; G(\mathbf{x} \leftrightarrow \mathbf{x}')\, dA(\mathbf{x})\,, \tag{8.2}$$

where

$$G(\mathbf{x} \leftrightarrow \mathbf{x}') \;=\; V(\mathbf{x} \leftrightarrow \mathbf{x}')\, \frac{|\cos(\theta_{\mathrm{o}})\, \cos(\theta'_{\mathrm{i}})|}{\|\mathbf{x} - \mathbf{x}'\|^2}\,. \tag{8.3}$$

Here θ_{o} and θ'_{i} are the angles between the segment $\mathbf{x} \leftrightarrow \mathbf{x}'$ and the surface normals at $\mathbf{x}$ and $\mathbf{x}'$ respectively, while $V(\mathbf{x} \leftrightarrow \mathbf{x}') = 1$ if $\mathbf{x}$ and $\mathbf{x}'$ are mutually visible and is zero otherwise.

We also use the change of variables (8.2) to rewrite the original measurement equation (3.18) as

$$I_j \;=\; \int_{\mathcal{M} \times \mathcal{M}} W_{\mathrm{e}}^{(j)}(\mathbf{x} \to \mathbf{x}')\, L(\mathbf{x} \to \mathbf{x}')\, G(\mathbf{x} \leftrightarrow \mathbf{x}')\, dA(\mathbf{x})\, dA(\mathbf{x}')\,, \tag{8.4}$$

where as usual, the notation $\mathbf{x} \to \mathbf{x}'$ indicates the direction of light flow. In particular, $W_{\rm e}^{(j)}(\mathbf{x} \to \mathbf{x}')$ represents the importance that is emitted from $\mathbf{x}'$ toward $\mathbf{x}$ (opposite to the arrow notation). This is, we define $W_{\rm e}^{(j)}(\mathbf{x} \to \mathbf{x}') \;=\; W_{\rm e}^{(j)}(\mathbf{x}', \omega)$, where $\omega = \widehat{\mathbf{x} - \mathbf{x}'}$.[1]

8.2 The path integral formulation

In this section, we first define the components of the path integral formulation: the integration domain, measure, and integrand. Next, we discuss the advantages of this formulation. Finally, we show how to use the path integral framework in Monte Carlo algorithms, and in particular how to calculate the probability densities with which paths are sampled.

Recall that our goal is to express each measurement in the form

$$I_j \;=\; \int_\Omega f_j(\bar{x})\, d\mu(\bar{x})\,. \tag{8.5}$$

To do this, let Ω_k represent the paths of length k, i.e. the set of paths of the form

$$\bar{x} \;=\; \mathbf{x}_0\, \mathbf{x}_1\, \ldots\, \mathbf{x}_k\,,$$

where $1 \le k < \infty$ and $\mathbf{x}_i \in \mathcal{M}$ for each i. We define a measure μ_k on this set of paths, called the *area-product measure,* according to

$$\mu_k(D) \;=\; \int_D dA(\mathbf{x}_0)\, \cdots\, dA(\mathbf{x}_k)\,,$$

where $D \subset \Omega_k$ is a set of paths. Formally, μ_k is a product measure [Halmos 1950]; we could also have written its definition as

$$\begin{aligned} d\mu_k(\mathbf{x}_0 \ldots \mathbf{x}_k) \;&=\; dA(\mathbf{x}_0)\, \cdots\, dA(\mathbf{x}_k)\,, \\ \text{or} \qquad \mu_k \;&=\; \underbrace{A \times \cdots \times A}_{k \text{ times}}\,. \end{aligned}$$

[1]Notice that the visibility factor $V(\mathbf{x} \leftrightarrow \mathbf{x}')$ hidden in the function G is essential, since $L(\mathbf{x} \to \mathbf{x}')$ refers to the radiance leaving $\mathbf{x}$, while $W_{\rm e}^{(j)}(\mathbf{x} \to \mathbf{x}')$ applies to the radiance arriving at $\mathbf{x}'$. To put this another way, L and $W_{\rm e}^{(j)}$ are both exitant quantities, since $W_{\rm e}^{(j)}$ specifies the importance leaving $\mathbf{x}'$, rather than the importance arriving at $\mathbf{x}$.

Next, we define the *path space* Ω as

$$\Omega \;=\; \bigcup_{k=1}^{\infty} \Omega_k\,,$$

i.e. Ω represents the set of paths of all finite lengths. We extend the area-product measure μ to this space in the natural way, by letting

$$\mu(D) \;=\; \sum_{k=1}^{\infty} \mu_k(D \cap \Omega_k)\,. \tag{8.6}$$

That is, the measure of a set of paths is simply the sum of the measures of the paths of each length.[2]

To complete the definition of the path integral formulation (8.5), we must define the integrand f_j. To do this, we start with the measurement equation (8.4), and recursively expand the transport equation (8.1) to obtain

$$\begin{aligned}
I_j \;&=\; \sum_{k=1}^{\infty} \int_{\mathcal{M}^{k+1}} L_{\mathrm{e}}(\mathbf{x}_0 \!\to\! \mathbf{x}_1)\, G(\mathbf{x}_0 \!\leftrightarrow\! \mathbf{x}_1) \prod_{i=1}^{k-1} f_{\mathrm{s}}(\mathbf{x}_{i-1} \!\to\! \mathbf{x}_i \!\to\! \mathbf{x}_{i+1})\, G(\mathbf{x}_i \!\leftrightarrow\! \mathbf{x}_{i+1}) \\
&\qquad\qquad \cdot W_{\mathrm{e}}^{(j)}(\mathbf{x}_{k-1} \!\to\! \mathbf{x}_k)\; dA(\mathbf{x}_0) \cdots dA(\mathbf{x}_k) \qquad (8.7)\\
&=\; \int_{\mathcal{M}^2} L_{\mathrm{e}}(\mathbf{x}_0 \!\to\! \mathbf{x}_1)\, G(\mathbf{x}_0 \!\leftrightarrow\! \mathbf{x}_1)\, W_{\mathrm{e}}^{(j)}(\mathbf{x}_0 \!\to\! \mathbf{x}_1)\; dA(\mathbf{x}_0)\, dA(\mathbf{x}_1) \\
&+\; \int_{\mathcal{M}^3} L_{\mathrm{e}}(\mathbf{x}_0 \!\to\! \mathbf{x}_1)\, G(\mathbf{x}_0 \!\leftrightarrow\! \mathbf{x}_1)\, f_{\mathrm{s}}(\mathbf{x}_0 \!\to\! \mathbf{x}_1 \!\to\! \mathbf{x}_2)\, G(\mathbf{x}_1 \!\leftrightarrow\! \mathbf{x}_2) \\
&\qquad\qquad \cdot W_{\mathrm{e}}^{(j)}(\mathbf{x}_1 \!\to\! \mathbf{x}_2)\; dA(\mathbf{x}_0)\, dA(\mathbf{x}_1)\, dA(\mathbf{x}_2) \\
&+\; \cdots\,.
\end{aligned}$$

The integrand f_j is defined for each path length k separately, by extracting the appropriate term from the expansion (8.7). For example, given a path $\bar{x} = \mathbf{x}_0\mathbf{x}_1\mathbf{x}_2\mathbf{x}_3$, we have

$$\begin{aligned}
f_j(\bar{x}) \;&=\; L_{\mathrm{e}}(\mathbf{x}_0 \!\to\! \mathbf{x}_1)\, G(\mathbf{x}_0 \!\leftrightarrow\! \mathbf{x}_1)\, f_{\mathrm{s}}(\mathbf{x}_0 \!\to\! \mathbf{x}_1 \!\to\! \mathbf{x}_2) \\
&\qquad \cdot G(\mathbf{x}_1 \!\leftrightarrow\! \mathbf{x}_2)\, f_{\mathrm{s}}(\mathbf{x}_1 \!\to\! \mathbf{x}_2 \!\to\! \mathbf{x}_3)\, G(\mathbf{x}_2 \!\leftrightarrow\! \mathbf{x}_3)\, W_{\mathrm{e}}^{(j)}(\mathbf{x}_2 \!\to\! \mathbf{x}_3)
\end{aligned}$$

(see Figure 8.2). This function f_j is called the *measurement contribution function*.

[2]This measure on paths is similar to that of Spanier & Gelbard [1969, p. 85]. However, in our case the path space Ω does not include any infinite-length paths. This makes it easy to verify that (8.6) is in fact a measure, directly from the axioms [Halmos 1950].

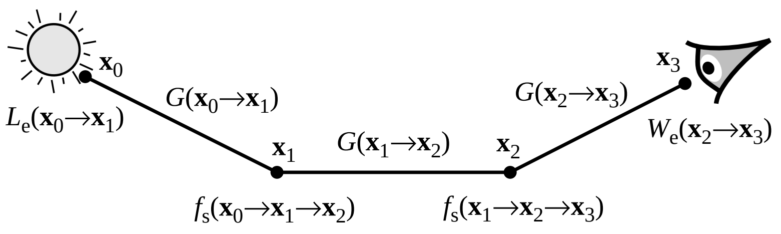
$\mathbf{x}_0$
$L_e(\mathbf{x}_0\to\mathbf{x}_1)$
$G(\mathbf{x}_0\to\mathbf{x}_1)$
$\mathbf{x}_1$
$G(\mathbf{x}_1\to\mathbf{x}_2)$
$\mathbf{x}_2$
$G(\mathbf{x}_2\to\mathbf{x}_3)$
$\mathbf{x}_3$
$W_e(\mathbf{x}_2\to\mathbf{x}_3)$
$f_s(\mathbf{x}_0\to\mathbf{x}_1\to\mathbf{x}_2)$
$f_s(\mathbf{x}_1\to\mathbf{x}_2\to\mathbf{x}_3)$

Figure 8.2: The measurement contribution function f_j is a product of many factors (shown for a path of length 3).

We have now defined all the terms of path integral model (8.5): the integration domain, integrand, and measure. There is nothing particularly complicated about this transformation; we have just expanded and rearranged the transport equations. The most significant aspect is that we have removed the sum over different path lengths, and replaced it with a single integral over an abstract measure space of paths.

8.2.1 Advantages of the path integral formulation

The path integral formulation has several advantages. First, the expression for each measurement has the form of an integral (as opposed to some other mathematical object). This allows us to derive new rendering algorithms by applying general-purpose integration techniques, such as multiple importance sampling (Chapter 9).

Second, the path integral model has a much simpler structure: a single expression defines the value of each measurement. In contrast, the integral equation approach requires two equations (the light transport and measurement equations), one of which is defined recursively. With the path integral approach, there are no adjoint equations, no intermediate quantities such as light or importance, and no need to choose between these alternatives. Measurements are defined and computed directly, by organizing the calculations around a *geometric* primitive (the path), rather than radiometric quantities.

By dealing with whole paths rather than rays, the path integral framework also provides a more explicit and complete description of light transport. Each path specifies the emission, scattering, and measurement events along a complete photon trajectory. On the other hand, integral equations describe the scattering events in isolation, by specifying the interaction of light with each surface separately.

This has practical consequences for sampling paths: the natural strategy for solving an integral equation is to sample the equation recursively, leading to paths that are built starting entirely from the lens, or entirely from a light source (depending on whether the light transport equation or its adjoint is sampled). With the path integral approach, on the other hand, it is possible to construct paths in arbitrary ways, e.g. by starting with a vertex in the middle, and building the path outwards in both directions. This leads directly to sampling strategies such as bidirectional path tracing (Chapter 10), and the Metropolis algorithm (Chapter 11).

Furthermore, the path integral approach gives a convenient framework for computing probability densities on paths (as described in the next section). This allows us to easily compare the probabilities with which a given path is sampled by different techniques. This is an essential prerequisite for the use of the multiple importance sampling and Metropolis techniques.

8.2.2 Applying the path integral formulation

In this section, we explain how the path integral framework can be used in Monte Carlo algorithms. We first show how measurements can be estimated, by randomly generating transport paths $\bar{X}$, and computing an estimate of the form $f_j(\bar{X})/p(\bar{X})$. This requires the evaluation of the probability density $p(\bar{X})$ with which each path was sampled. We consider how to do this within the framework of *local path sampling*, which is general enough to describe virtually all unbiased path sampling algorithms that are used in practice.

Our goal is to estimate the path integral

$$I_j \;=\; \int_\Omega f_j(\bar{x})\, d\mu(\bar{x})$$

for each measurement I_j. To do this, the natural Monte Carlo strategy is to first sample a random path $\bar{X}$ according to some chosen density function p, and then compute an estimate of the form

$$I_j \;\approx\; \frac{f_j(\bar{X})}{p(\bar{X})}\,. \tag{8.8}$$

This is an unbiased estimate of the measurement I_j, since its expected value is

$$\begin{aligned} E\left[\frac{f_j(\bar{X})}{p(\bar{X})}\right] &= \int_\Omega \frac{f_j(\bar{x})}{p(\bar{x})}\, p(\bar{x})\, d\mu(\bar{x}) \\ &= \int_\Omega f_j(\bar{x})\, d\mu(\bar{x}) \\ &= I_j\,, \end{aligned} \tag{8.9}$$

where we have assumed that p is measured with respect to the area-product measure μ, in order for the first line of this equation to hold.

To apply this strategy, we must be able to evaluate the functions f_j and p for the given path $\bar{X}$. An explicit formula for the measurement contribution function f_j has already been given; thus, the main question is how to evaluate the probability density $p(\bar{X})$. Obviously, this depends not only on the particular path $\bar{X}$, but also on how this path was generated. For example, one way to generate paths is with ordinary path tracing: the vertex $\mathbf{x}_k$ is chosen on the lens, and subsequent vertices $\mathbf{x}_{k-1}$, . . ., $\mathbf{x}_1$ are generated by following random bounces backward, until eventually we connect the path to a random vertex $\mathbf{x}_0$ on a light source. The probability $p(\bar{X})$ depends on all of the random choices made during this process, as we will discuss in more detail below.

8.2.2.1 Local path sampling

We will concentrate on a particular family of methods for generating paths, called *local path sampling algorithms*. These methods generate vertices one at a time, based on local information at existing vertices (such as the BSDF). There are three basic mechanisms that can be used to construct paths in this framework:

- A vertex can be chosen according some *a priori* distribution over the scene surfaces. For example, this can be used to sample a vertex on a light source, with a probability density proportional to the radiant exitance (i.e. the power per unit area emitted over the light source). Similarly, this technique can be used to sample the initial vertex on a finite-aperture lens. It can also be used to sample intermediate vertices along the path, e.g. to sample a vertex on a window between two adjacent rooms.

- The second method for generating a vertex is to sample a direction according to a locally defined probability distribution at an existing vertex $\mathbf{x}$, and then cast a ray to find the first surface intersection $\mathbf{x}'$ (which becomes the new vertex). For example, this is what happens when the BSDF at an existing vertex is sampled (or an approximation to the BSDF). This mechanism can also used to sample a direction for emission, once a vertex on a light source has been chosen.
- The third mechanism for path sampling is to connect two existing vertices, by checking the visibility between them. In effect, this step verifies the existence of an *edge* between two vertices, rather than generating a new vertex.

By combining these three simple techniques, it is possible to sample paths in a great variety of ways. Subpaths can be built up starting from the light sources, the lens, or from an arbitrary scene surface. These subpaths can then be joined together to create a full path from a light source to the lens. This local sampling framework is general enough to accommodate virtually all path sampling techniques that are used in practice.[3]

8.2.2.2 Computing the path probabilities

In this section, we describe how to compute the probability density $p(\bar{x})$ for sampling a given path $\bar{x}$. As mentioned above (equation (8.9)), we wish to compute the probability density with respect to the area-product measure μ, that is:

$$p(\bar{x}) \;=\; \frac{dP}{d\mu}(\bar{x})\,.$$

Given a path $\bar{x} \;=\; \mathbf{x}_0 \ldots \mathbf{x}_k$, this expands to

$$\begin{aligned} p(\bar{x}) \;&=\; \frac{dP}{d\mu}(\mathbf{x}_0 \ldots \mathbf{x}_k) \\ &=\; \prod_{i=0}^{k} \frac{dP}{dA}(\mathbf{x}_i)\,. \end{aligned}$$

[3]As an example of a non-local sampling technique, suppose that the location of a new vertex is computed by solving an algebraic equation involving two or more existing vertices. For example, this could be used to determine the point $\mathbf{y}$ on a curved mirror that reflects light from a given vertex $\mathbf{x}$ to another vertex $\mathbf{x}'$. This is not allowed in the local path sampling framework, since the position of $\mathbf{y}$ depends on more than one existing vertex. This type of non-local sampling will be discussed further in Section 8.3.4.

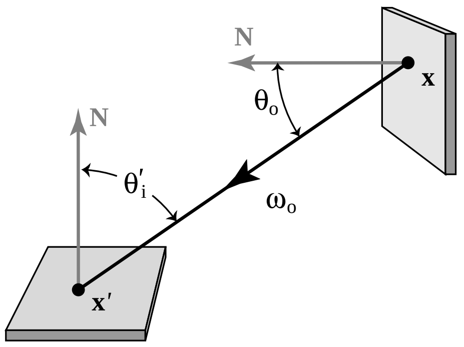

Figure 8.3: Geometry for converting between area and directional probabilities.

Thus to evaluate $p(\bar{X})$, we must compute the probability per unit area (dP/dA) with which each vertex $\mathbf{x}_i$ was generated, and multiply them together.

We now consider how to compute the probability for sampling a given vertex. According to the local path sampling model, each vertex $\mathbf{x}_i$ can be generated according to one of two methods: either $\mathbf{x}_i$ is sampled from a distribution over the scene surfaces (in which the probability density $dP/dA(\mathbf{x}_i)$ can be computed directly), or else it is generated by casting a ray from an existing vertex, in a randomly chosen direction.

To calculate the density in the latter case, let $\mathbf{x}$ be the existing vertex, and let $\mathbf{x}' = \mathbf{x}_i$ be the new vertex. We assume that $\mathbf{x}'$ was generated by casting a ray from $\mathbf{x}$ in the direction ω_o, where

$$\omega_\mathrm{o} \;=\; \widehat{\mathbf{x}' - \mathbf{x}}$$

(see Figure 8.3). We are also given the probability density $p(\omega_\mathrm{o})$ with which ω_o was chosen (measured with respect to solid angle). To compute the density $p(\mathbf{x}')$ with respect to surface area, we must express it in terms of the given density $p(\omega_\mathrm{o})$. These two densities are related by

$$\begin{aligned} \frac{dP}{dA}(\mathbf{x}') &= \frac{dP}{d\sigma}(\omega_\mathrm{o})\,\frac{d\sigma(\omega_\mathrm{o})}{dA(\mathbf{x}')} \\ \Longrightarrow \qquad p(\mathbf{x}') &= p(\omega_\mathrm{o})\left(\frac{|\cos(\theta'_\mathrm{i})|}{\|\mathbf{x}-\mathbf{x}'\|^2}\right) \end{aligned} \tag{8.10}$$

(see Figure 8.3). The parenthesized expression is the solid angle subtended at $\mathbf{x}$ per unit of surface area at $\mathbf{x}'$.

Using these rules, it is straightforward to compute the probability density $p(\bar{x})$ for the whole path. We simply consider the vertices in the order that they were generated, and multiply together the densities dP/dA for each vertex (converting from directional to area probabilities as necessary). There are few restrictions on how the paths are generated: starting from the lens (as with path tracing), starting from the lights (as with particle tracing), or a combination of both (as with bidirectional path tracing). Paths can also be constructed starting from the middle, by sampling vertices according to predefined distributions over the scene surfaces: this could be useful in difficult geometric settings, e.g. to generate transport paths that pass through a known small portal.

In the path integral framework, all of these possibilities are handled in the same way. They are viewed as different sampling strategies for the measurement equation (8.5), leading to different probability distributions on the space of paths. They are unified under one simple equation, namely the estimate $f_j(\bar{X})/p(\bar{X})$.

Densities with respect to projected solid angle. In many cases, it is more natural and convenient to represent directional distributions as densities with respect to projected solid angle $\sigma^\perp$ (rather than ordinary solid angle σ). We summarize the equations here for future reference.

Given an existing vertex $\mathbf{x}$ (Figure 8.3), let $p(\omega_\mathrm{o})$ and $p^\perp(\omega_\mathrm{o})$ be the probability densities with respect to ordinary and projected solid angle respectively for sampling the given direction ω_o. These two densities are related by

$$\begin{aligned} & \frac{dP}{d\sigma^\perp}(\omega_\mathrm{o}) &=& \frac{dP}{d\sigma}(\omega_\mathrm{o})\,\frac{d\sigma(\omega_\mathrm{o})}{d\sigma^\perp(\omega_\mathrm{o})} \\ \Longrightarrow \quad & p^\perp(\omega_\mathrm{o}) &=& p(\omega_\mathrm{o})\,\frac{1}{\cos(\theta_\mathrm{o})}\,, \end{aligned} \tag{8.11}$$

where we have used the relationship

$$d\sigma^\perp(\omega_\mathrm{o}) \;=\; |\omega_\mathrm{o}\cdot\mathbf{N}(\mathbf{x})|\, d\sigma(\omega_\mathrm{o})\,.$$

Putting this together with equation (8.10), we can convert between densities with respect to projected solid angle and densities with respect to surface area using

$$\begin{aligned} p(\mathbf{x}') &= p^{\perp}(\omega_o) \frac{|\cos(\theta_o)\ \cos(\theta_i')|}{\|\mathbf{x} - \mathbf{x}'\|^2} \\ &= p^{\perp}(\widehat{\mathbf{x}' - \mathbf{x}})\, G(\mathbf{x} \leftrightarrow \mathbf{x}')\,, \end{aligned}$$

where G is the geometric factor (8.2).[4] Notice that this conversion factor is symmetric, unlike the conversion factor (8.10) for densities with respect to ordinary solid angle.

8.3 The limitations of path sampling

Although algorithms based on path sampling tend to be simple and general, they do have limits. For example, if point light sources and perfect mirrors are allowed, then there are some types of transport paths that cannot be sampled at all. Images computed by path sampling algorithms will be missing the contributions made by these paths. As a typical example of this problem, consider a scene where a point light source reflects off a mirror, creating caustics on a diffuse surface. Although algorithms such as bidirectional path tracing are capable of rendering these caustics when viewed directly, they will fail if the caustics are viewed indirectly through a second mirror. (The indirectly viewed caustics will simply be missing from the image.)

More generally, there are some light transport problems that are provably difficult for any algorithm. In this regard, it has been shown that some ray tracing problems are *undecidable*, i.e. they cannot be solved on a Turing machine [Reif et al. 1994]. These examples are not physically realizable, since they rely on perfect mirrors and infinite geometric precision. However, we can expect that as the geometry and materials of the input scene approach a provably difficult configuration, any light transport algorithm will perform very badly.

Our goals in this section are more practical. We are mainly concerned with the limitations of *local* path sampling algorithms, as described in Section 8.2.2.1. For this type of algorithm, problems are caused not only by mirrors and point sources, but also by refraction,

[4]Note that the visibility term $V(\mathbf{x} \leftrightarrow \mathbf{x}')$ hidden in G is required only when the visibility between $\mathbf{x}$ and $\mathbf{x}'$ is not known.

perfectly anisotropic surfaces, parallel light sources, pinhole lenses, and orthogonal viewing projections. Our goal is to determine which combinations of these features can cause local path sampling algorithms to fail.

We start by reviewing Heckbert's regular expression notation for paths. Next, we show how to extend this notation to describe the properties of light sources and sensors, in order to allow features such as point light sources and orthographic lenses to be represented in a compact and consistent way. We then give a criterion for determining which types of paths cannot be generated by local path sampling. Finally, we consider some ways to lift this restriction using non-local sampling methods.

8.3.1 Heckbert's regular expression notation for paths

Heckbert [1990] introduced a useful notation for classifying paths by means of regular expressions. Originally, it was used to describe the capabilities of multi-pass global illumination algorithms, e.g. algorithms that combine radiosity and ray tracing. In this context, it was assumed that all BSDF's can be written as a linear combination of an ideal diffuse component and an ideal specular component. For example, a typical surface might reflect 50% of the incident light diffusely, reflect 10% in a mirror-like fashion, and absorb the rest.

Paths are then described using regular expressions of the form[5]

$$L\,(S|D)^*\,E\,.$$

Each symbol represents one vertex of a path: L denotes the first vertex of the path, which lies on a light source, while E denotes the last vertex (the camera position or "eye"). The remaining vertices are classified as S or D, according to whether the light was reflected by the specular or diffuse component of the surface respectively. Note that the symbols S and D represent the type of the *scattering event* at each vertex, not the type of the surface, since the surface itself is allowed to be a combination of specular and diffuse.

[5]In regular expressions, X^+ denotes one or more occurrences of X, X^* denotes zero or more occurrences of X, $X|Y$ denotes a choice between X or Y, ϵ denotes the empty string, and parentheses are used for grouping.

Definitions for general materials. This notation is easily extended to scenes with general materials, by redefining the symbols S and D appropriately. We show how to make these definitions rigorously, by relating them to the BSDF.

Let $\bar{x} = \mathbf{x}_0 \ldots \mathbf{x}_k$ be a path, and consider the scattering event at a vertex $\mathbf{x}_i$ (where $0 < i < k$). For general materials, we let the symbol D represent any scattering event where the BSDF is finite, i.e. where

$$f_{\mathrm{s}}(\mathbf{x}_{i-1} \to \mathbf{x}_i \to \mathbf{x}_{i+1}) < \infty .$$

All other scattering events (where the BSDF is not finite) are denoted by the symbol S. This category includes not only pure specular reflection and refraction, where light is scattered in a zero-dimensional set of directions, but also pure anisotropic scattering, where light is scattered in a one-dimensional set of directions (similar to the reflection properties of brushed aluminum). These possibilities will be discussed in more detail below.

8.3.2 Full-path regular expressions

Heckbert's notation describes only the scattering events along a path. We show how to extend these regular expressions in a natural way, to describe the properties of light sources and sensors as well.

Each light source is classified according to a two-letter combination, of the form $(S|D)\,(S|D)$. The first letter represents the surface area of the light source: D denotes a finite-area source, while S denotes a source with zero area (e.g. a point or linear source). The second letter represents the directional properties of the emission: D denotes emission over a finite solid angle, while S denotes emission over a set of angles with measure zero. Thus, a point light source that radiates light in all directions would be denoted by the regular expression LSD. Note that unlike Heckbert's notation, the symbol L does not represent a real vertex; it is simply a placeholder that indicates the ordering of vertices (i.e. the fact that the first vertex is on a light source rather than a sensor).

Similarly, to represent the properties of the sensor we use a suffix of the form

$$(S|D)\,(S|D)\,E\,.$$

LDD	a diffusely emitting sphere
LDS	sunlight shining through a window, where the window itself is modeled as the light source
LSD	a point spotlight
LSS	a laser beam
DDE	a finite-aperture lens
SDE	an orthographic projection lens (where the image plane located within the scene, rather than at infinity)
DSE	a pinhole lens
SSE	an idealized spot meter (which measures radiance along a single given ray)

Table 8.1: Examples of regular expressions that approximate various kinds of real light sources and sensors (e.g. by treating the sun as a point at infinity, etc.)

The first letter represents the directional sensitivity of the sensor, i.e. whether it is sensitive to light over a finite solid angle (D), or to light that arrives from a set of directions with measure zero (S). The second letter represents the surface area of the sensor, with the same conventions used for the first letter of the light source classification.

Table 8.1 gives some examples of light sources and lens models which are good approximations to the various letter combinations (e.g. if we treat the sun as a point source at infinity).

Combining this notation for light sources and sensors with Heckbert's notation for scattering events, an entire path is thus described by a regular expression such as

$$L\,D\,D\,S^*\,D\,D\,E\,.$$

This example represents a path that starts on an ordinary area light source, is scattered by zero or more specular surfaces, and terminates at an ordinary finite-aperture lens. This extended notation is called a *full-path regular expression*.

The main advantage of full-path expressions is that they give a compact way to describe the paths generated by specific sampling strategies. For this purpose, it is essential to specify the properties of the light source and sensor, since some strategies do not work for sources or sensors with zero area, or those that emit or measure light over a zero solid angle. (For example, "pure" path tracing cannot handle point light sources, since they will never be intersected by a path that is randomly generated starting from the lens.) We will make extensive use of full-path expressions to describe the sampling strategies of bidirectional path tracing and Metropolis light transport, and also to investigate the limitations of local path sampling.

Formal definitions of the full-path notation. Full-path regular expressions can be defined more rigorously in the following way. First, we show how to split the emitted radiance function L_{e} into a product of two factors $L_{\mathrm{e}}^{(0)}$ and $L_{\mathrm{e}}^{(1)}$, which represent the spatial and directional components of the emission respectively. The factor $L_{\mathrm{e}}^{(0)}$ is defined by

$$L_{\mathrm{e}}^{(0)}(\mathbf{x}) \;=\; \int_{\mathcal{S}^2} L_{\mathrm{e}}(\mathbf{x}, \omega)\, d\sigma^{\perp}(\omega)\,, \tag{8.12}$$

and represents the *radiant exitance* (emitted power per unit area) associated with a point $\mathbf{x}$ on a light source. The second factor $L_{\mathrm{e}}^{(1)}$ is given by

$$L_{\mathrm{e}}^{(1)}(\mathbf{x}, \omega) \;=\; L_{\mathrm{e}}(\mathbf{x}, \omega) / L_{\mathrm{e}}^{(0)}(\mathbf{x})\,, \tag{8.13}$$

and represents the directional distribution of the emitted radiance at $\mathbf{x}$. These factors correspond to the fact that sampling for emission is naturally subdivided into two steps, consisting of first choosing a point on a light source, and then a direction for the emitted ray. Notice that by definition,

$$\int_{\mathcal{S}^2} L_{\mathrm{e}}^{(1)}(\mathbf{x}, \omega)\, d\sigma^{\perp}(\omega) \;=\; 1\,,$$

so that $L_{\mathrm{e}}^{(1)}$ is simply the probability density function for ω, for a given choice of $\mathbf{x}$.

With these definitions, the light source notation LXY has the following meaning:

$$X \;=\; \begin{cases} D & \text{if } L_{\mathrm{e}}^{(0)}(\mathbf{x}_0) < \infty \\ S & \text{otherwise}\,, \end{cases}$$

$$Y \;=\; \begin{cases} D & \text{if } L_{\mathrm{e}}^{(1)}(\mathbf{x}_0 \!\to\! \mathbf{x}_1) < \infty \\ S & \text{otherwise}\,. \end{cases}$$

Likewise, we can rigorously define the meaning of the notation YXE for sensors. This is done by splitting the emitted importance function W_e into a product of two factors $W_\mathrm{e}^{(0)}$ and $W_\mathrm{e}^{(1)}$, and making a definition similar to the one for LXY.

Thus far, we have only distinguished between light that is emitted or scattered in a two-dimensional set of directions (D), vs. all other cases (S). It is sometimes useful to classify the S vertices further, according to whether light is scattered in a zero- or one-dimensional set of directions (S_0 vs. S_1). This extended notation is discussed in Appendix 8.B, and can be used to describe the properties of light sources, sensors, and materials more precisely.

Note that Langer & Zucker [1997] have independently proposed a classification system for light sources that is similar to the one described here. However, they do not attempt to give a general definition of their classification scheme, they do not develop any notation for it, and they do not consider the classification of sensors or scattering events.

8.3.2.1 Interpreting sources and sensors as scattering events

The definitions above are somewhat cumbersome to use, because sources and sensors are treated as special cases. In other words, the first two $(S|D)$ symbols and the last two $(S|D)$ symbols of each path cannot be handled in the same way as the rest, since they represent emission and measurement rather than scattering. It would be easier to reason about these regular expressions if the S and D symbols had a consistent meaning.

In this section, we show how the S and D symbols describing light sources and sensors can be interpreted as "scattering events" in a natural way. To do this, we introduce an imaginary vertex at each end of the path, and extend the definition of the BSDF to describe light transport to and from these imaginary vertices. With these changes, all of the symbols in a full-path regular expression have a consistent interpretation, so that the special cases associated with sources and sensors can be avoided.

The conversion from emission to scattering is described in two steps. We first consider the directional component of the emission, and then the spatial component.

Scattering events at $\mathbf{x}_0$ and $\mathbf{x}_k$. We show how the directional components of the emission functions L_e and W_e can be interpreted as scattering at the vertices $\mathbf{x}_0$ and $\mathbf{x}_k$. To do this, we introduce two imaginary vertices Ψ_L and Ψ_W, which become the new path endpoints.

A complete path thus has the form

$$\Psi_L \, \mathbf{x}_0 \, \mathbf{x}_1 \; \ldots \; \mathbf{x}_k \, \Psi_W \,,$$

where the vertices Ψ_L and Ψ_W always occur at positions $\mathbf{x}_{-1}$ and $\mathbf{x}_{k+1}$ respectively.

We regard the vertex Ψ_L as the source of all light, while Ψ_W is the source of all importance. That is, rather than allowing surfaces to emit light directly, we assume that emission occurs only at the vertex Ψ_L. Light is emitted along imaginary rays of the form $\Psi_L \to \mathbf{x}$, and is then scattered at $\mathbf{x}$ into physical rays of the form $\mathbf{x} \to \mathbf{x}'$. This process is defined so that we obtain the same results as the original emission function L_e. Similarly, all sensor measurements are made at the point Ψ_W. This corresponds to the following symbolic definitions:

$$\begin{aligned}
L_\text{e}(\Psi_L \to \mathbf{x}) &= L_\text{e}^{(0)}(\mathbf{x})\,, \\
f_\text{s}(\Psi_L \to \mathbf{x} \to \mathbf{x}') &= L_\text{e}^{(1)}(\mathbf{x} \to \mathbf{x}')\,, \\
f_\text{s}(\mathbf{x}' \to \mathbf{x} \to \Psi_W) &= W_\text{e}^{(1)}(\mathbf{x}' \to \mathbf{x})\,, \\
W_\text{e}(\mathbf{x} \to \Psi_W) &= W_\text{e}^{(0)}(\mathbf{x})\,,
\end{aligned}$$

where $L_\text{e}^{(i)}$ and $W_\text{e}^{(i)}$ are the spatial and directional components of emission (8.12, 8.13).

Scattering events at Ψ_L and Ψ_W. We now show how the spatial components of emission can be interpreted as scattering at the imaginary vertices Ψ_L and Ψ_W. To do this, we assume that the emitted light is initially concentrated on the single imaginary ray $\Psi_L \to \Psi_L$. This light is scattered at Ψ_L, to obtain a distribution along rays of the form $\Psi_L \to \mathbf{x}$. We then proceed as before (with a second scattering step at $\mathbf{x}$), to obtain emission along physical rays $\mathbf{x} \to \mathbf{x}'$. Similarly, measurements are handled by scattering light from rays of the form $\mathbf{x} \to \Psi_W$ into the single ray $\Psi_W \to \Psi_W$, where the actual measurement takes place.

This idea corresponds to the following symbolic definitions. First we define Φ_L and Φ_W to represent the total power and the total importance emitted over all surfaces of the scene:

$$\begin{aligned}
\Phi_L &= \int_{\mathcal{M}} L_\text{e}^{(0)}(\mathbf{x}) \, dA(\mathbf{x})\,, \\
\Phi_W &= \int_{\mathcal{M}} W_\text{e}^{(0)}(\mathbf{x}) \, dA(\mathbf{x})\,.
\end{aligned}$$

Next, we change the emission functions so that light and importance are emitted on a single imaginary ray:

$$\begin{aligned} L_{\mathrm{e}}(\Psi_L \to \Psi_L) &= \Phi_L \,, \\ W_{\mathrm{e}}(\Psi_W \to \Psi_W) &= \Phi_W \,. \end{aligned}$$

Finally, we extend the BSDF to scatter this light and importance along rays of the form $\Psi_L \to \mathbf{x}$ and $\mathbf{x} \to \Psi_W$ respectively:

$$\begin{aligned} f_{\mathrm{s}}(\Psi_L \to \Psi_L \to \mathbf{x}) &= L_{\mathrm{e}}^{(0)}(\mathbf{x}) \,/\, \Phi_L \,, \\ f_{\mathrm{s}}(\mathbf{x} \to \Psi_W \to \Psi_W) &= W_{\mathrm{e}}^{(0)}(\mathbf{x}) \,/\, \Phi_W \,. \end{aligned}$$

Notice that these BSDF's are normalized to integrate to one, so that there is a natural correspondence with scattering.

With these conventions, every S and D symbol corresponds to a unique scattering event at some vertex of the full path $\mathbf{x}_{-1} \ldots \mathbf{x}_{k+1}$. Furthermore, these symbols have a consistent meaning. Given any vertex $\mathbf{x}_i$ of a path, the symbol D means that the BSDF at that vertex is finite (so that energy is spread over a two-dimensional set of adjacent vertices), while S means that the BSDF is not finite (in which case power is distributed to a zero- or one-dimensional set of adjacent vertices). This consistency will be useful as we study the limitations of local path sampling below.

8.3.3 The limitations of local path sampling

In this section, we show that local sampling strategies can only generate paths that contain the substring DD. Any path that does not contain this substring cannot be sampled, and the contributions of these paths will be missing from any computed images. Examples of paths that cannot be sampled are shown in Table 8.2.

We start by consider specular vertices, and the constraints that they impose on path sampling. Next, we show that paths can be sampled by local sampling strategies if and only if they contain the substring DD. Finally, we discuss the significance of these results.

Lemma 8.1. *Let $\bar{x}$ be any path generated by a local sampling algorithm, for which the measurement contribution function $f_j(\bar{x})$ is non-zero. If this path contains a specular vertex $\mathbf{x}_i$,*

$L\,S\,D\,S\,D\,S\,E$	a point light source reflected in a mirror, viewed with a pinhole lens
$L\,D\,S\,S\,D\,S\,D\,E$	caustics from a parallel light source, viewed with an orthographic lens
$L\,S\,D\,S\,D\,S\,D\,S\,E$	caustics from a point light source, viewed indirectly through a mirror with a pinhole lens

Table 8.2: Examples of path types that cannot be generated by local sampling algorithms.

then one of the adjacent vertices $\mathbf{x}_{i+1}$ *or* $\mathbf{x}_{i-1}$ *was necessarily generated by sampling the BSDF at* $\mathbf{x}_i$*.*

Proof. For any fixed positions of $\mathbf{x}_i$ and $\mathbf{x}_{i-1}$, consider the positions of $\mathbf{x}_{i+1}$ for which

$$f_{\mathrm{s}}(\mathbf{x}_{i-1} \to \mathbf{x}_i \to \mathbf{x}_{i+1}) \;=\; \infty\,,$$

i.e. for which $\mathbf{x}_i$ is a specular vertex. By definition, the possible locations of $\mathbf{x}_{i+1}$ form a set of area measure zero, since they subtend a zero solid angle at $\mathbf{x}_i$. Similarly, if we fix the positions of $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$, the possible locations of $\mathbf{x}_{i-1}$ for which $\mathbf{x}_i$ is a specular vertex form a set of measure zero.

Thus, if the vertices $\mathbf{x}_{i-1}$ and $\mathbf{x}_{i+1}$ are generated independently by the local sampling algorithm, then $\mathbf{x}_i$ has type D with probability one. Thus if $\mathbf{x}_i$ has type S, then one of these two vertices must be generated by sampling the BSDF at $\mathbf{x}_i$ (since this is the only other alternative that is allowed within the framework of local path sampling). ■

It is easy to extend this result to the case where several specular vertices are adjacent.

Corollary 8.2. *Let* $\bar{x}$ *be a path as described above, and suppose that* $\bar{x}$ *contains a subpath* $\mathbf{x}_i \ldots \mathbf{x}_j$ *of the form* DS^+D*. Then one of the endpoints* $\mathbf{x}_i$ *or* $\mathbf{x}_j$ *must be generated by sampling the BSDF of the adjacent* S*-vertex (that is, either* $\mathbf{x}_{i+1}$ *or* $\mathbf{x}_{j-1}$*).* ■

We are now ready to consider the sampling of full paths.

Theorem 8.3. *Let $\bar{x}$ be a path generated by a local sampling algorithm for which the measurement contribution function is non-zero. Then $\bar{x}$ necessarily has the form*

$$L\,(S|D)^*\,D\,D\,(S|D)^*\,E\,,$$

i.e. it must contain the substring DD. Furthermore, it is possible to generate any path of this form using local sampling strategies.

Proof. If $\bar{x}$ does not contain the substring DD, then it has the form

$$L\,(D|\epsilon)\,S^+\,(DS^+)^*\,(D|\epsilon)\,E\,.$$

This path has n specular substrings of the form S^+, but only $n-1$ vertices of type D separating them.[6] Thus according to the corollary above, one of these D vertices must be generated by sampling the BSDF of *both* adjacent specular vertices (which is not possible). In effect, there are not enough D vertices to allow this path to be sampled by local techniques.

Conversely, let $\bar{x}$ be a path that contains an edge $\mathbf{x}_i\mathbf{x}_{i+1}$ of the form DD. Then this path can be generated by at least one local sampling strategy: namely, by generating the subpath $\mathbf{x}_0 \ldots \mathbf{x}_i$ starting from a light source, and the subpath $\mathbf{x}_{i+1} \ldots \mathbf{x}_k$ starting from the lens. ∎

Thus, the DD condition is necessary and sufficient for local path sampling. Of course, specific algorithms may have more restrictive requirements. With ordinary path tracing, for example, all vertices are generated starting from the camera lens, except for the vertex $\mathbf{x}_0$ which is chosen directly on the surface of a light source. This implies that ordinary path tracing can only sample paths of the form

$$L\,(S|D)\,D\,D\,(S|D)^+\,E\,.$$

These results are significant for two reasons. First, it is very common for graphics systems to support point light sources and perfect mirrors, even though these are mathematical idealizations that do not physically exist. If scenes are modeled that use these primitives, then some lighting effects will simply be missing from the computed images. Second, even

[6]The symbol following L and the symbol preceding E do not count, because they are not sampled: they represent the fixed, imaginary vertices Ψ_L and Ψ_W.

if we disallow these features (e.g. by disallowing point and parallel light sources, so that every path starts with the prefix LDD), we should expect that path sampling algorithms will perform badly as the scene model approaches a difficult configuration. In this case, the contributions from the difficult paths will not be missing; however, they will be sampled with high variance, leading to noisy regions in resulting images.

8.3.4 Approaches to non-local sampling

We outline several approaches for handling paths that cannot be sampled locally. The easiest solution is to not allow these paths in the first place, by placing mild restrictions on the scene model. For example, any of the following strategies are sufficient:

- Allow only (ordinary) area light sources, so that all paths start with LDD.
- Allow only finite-aperture lenses, so that all paths end with DDE.
- Do not allow perfectly specular surfaces.

These strategies ensure that path sampling algorithms will produce unbiased results, although there can still be high variance in limiting cases as discussed above.

A second approach is to use a more sophisticated path sampling strategy. We first introduce some new terminology.

Chains and chain separators. Given a path, we divide its edges into a sequence of *chains* as follows. A vertex is called a *chain separator* if it has type D, or if it is one of the special vertices Ψ_L or Ψ_W. A *chain* is now defined to be a maximal subpath bounded by chain separators (not including the symbols L and E, which do not correspond to any vertex). For example, the path

$$L\,D\,S\,D\,D\,S\,S\,D\,S\,E$$

consists of four chains. The first chain is DSD, consisting of the imaginary edge from Ψ_L to $\mathbf{x}_0$, and the real edge from $\mathbf{x}_0$ to $\mathbf{x}_1$. The second chain is DD (the edge $\mathbf{x}_1\mathbf{x}_2$), the third is $DSSD$ (three edges connecting $\mathbf{x}_2$ to $\mathbf{x}_5$), and the last chain is DS, an imaginary edge from $\mathbf{x}_5$ to Ψ_W. Notice that each chain separator vertex is shared between two chains (except for the special vertices Ψ_L and Ψ_W).

Connectors. We can extend the class of paths that can be sampled by implementing methods that generate *connecting chains*. That is, given two vertices $\mathbf{x}$ and $\mathbf{x}'$ of type D, we would like to generate a chain of zero or more specular vertices that connect them. Strategies that do this are called *connectors*. The simplest connector consists of joining the two vertices with an edge, by checking the visibility between them. This yields a chain of the form DD.

Another simple form of connector can be used with planar mirrors, by computing the point $\mathbf{y}$ on the mirror that reflects light from $\mathbf{x}$ to $\mathbf{x}'$. If such a point $\mathbf{y}$ does not exist, or if either of the segments $\mathbf{x}\mathbf{y}$ or $\mathbf{y}\mathbf{x}'$ is occluded, then the connection attempt fails. Otherwise, we have generated a connecting chain of the form DSD. This is similar to the idea of "virtual worlds" and "virtual light sources" used in radiosity and elsewhere [Rushmeier 1986, Wallace et al. 1987, Ward 1994].

Connectors can also be used to handle parallel light sources (LDS) and orthogonal viewing projections (SDE) in a simple way. For example, a connecting chain between a real vertex $\mathbf{x}$ and the imaginary vertex Ψ_L can be generated by projecting $\mathbf{x}$ onto the surface of the light source along the direction of emission.

The general case is closely related to the problem of computing illumination from curved reflectors [Mitchell & Hanrahan 1992]. The connecting chains problem can be equivalently stated as follows: given a point source at $\mathbf{x}$, what is the irradiance received at $\mathbf{x}'$ over specular paths? Light flows from $\mathbf{x}$ to $\mathbf{x}'$ along paths of stationary optical length, also known as *Fermat paths*. In general, there are a countable set of such paths, and they can be found by solving an optimization problem [Mitchell & Hanrahan 1992]. Once a path has been found, the irradiance received at $\mathbf{x}'$ along that path can be determined by keeping track of the shape of the wavefront as light is reflected, refracted, and propagated, and computing the Gaussian curvature of the wavefront at $\mathbf{x}'$.

In our case, we seek an algorithm that can either generate all such paths (in which case their contributions are summed), or one that can generate a single path at random (in which case there must be a non-zero probability of generating each candidate path, and this probability must be explicitly computable). This would make it possible to generate paths of any type in an unbiased Monte Carlo algorithm.

Although it seems unlikely that the general case will ever be practical, these ideas are still useful for handling planar mirrors, short sequences of such mirrors, or simple curved surfaces. With more sophisticated geometric search techniques, it may eventually be possible to handle moderately large numbers of specular surfaces in this way with reasonable efficiency.

Appendix 8.A Other measures on path space

We describe several new measures on the path space μ. These include the *measurement contribution measure*, the *power throughput measure*, the *scattering throughput measure*, and the *geometric throughput measure*. Each of these measures has a natural physical significance, which is described. We also show that it is possible to base the path integral framework on any of these measures (rather than using the area-product measure μ). To avoid confusion, we will use the symbol μ^{a} for the area-product measure throughout this appendix.

The measurement contribution measure. The most important of these new measures is the *measurement contribution measure*, defined by

$$\mu_j^{\mathrm{m}}(D) \;=\; \int_D f_j(\bar{x})\, \mu^{\mathrm{a}}(\bar{x})\,. \tag{8.14}$$

This equation combines f_j and μ^{a} into a single measure μ_j^{m}, with the following physical significance: $\mu_j^{\mathrm{m}}(D)$ represents the portion of measurement I_j that is due to light flowing on the given set of paths D. In particular, the value of I_j itself is given by

$$I_j \;=\; \mu_j^{\mathrm{m}}(\Omega)\,,$$

i.e. I_j is the measure of the whole path space. The units of $\mu_j^{\mathrm{m}}(D)$ are [S] (the unit of sensor response).

This measure μ_j^{m} is actually the fundamental component of our path integral framework. It is more basic than the measurement contribution function f_j, since f_j implicitly depends on the measure used for integration (i.e. the area-product measure μ^{a}). By choosing different integration measures (e.g. the ones we define below), we can obtain any number of different but equivalent "measurement contribution functions". In contrast, the meaning of μ_j^{m} does not depend on details such as these.

The main reason for working with the function f_j (rather than the measure μ_j^{m}) is so that Monte Carlo estimators can be written as a ratio of functions, rather than as Radon-Nikodym derivatives. For example, the estimator $f_j(\bar{X})/p(\bar{X})$ corresponds to the Radon-Nikodym derivative

$$\frac{d\mu_j^{\mathrm{m}}}{dP}(\bar{X})\,.$$

Although this may be an improvement from the standpoint of purism (since it avoids any reference to the auxiliary measure μ^{a}), it is undesirable from a practical standpoint. It makes use of the Radon-Nikodym derivative (which is unfamiliar to many in graphics), and leaves us with a rather abstract expression with no clear recipe for computing its value. This is why we have emphasized the formulation of Section 8.2, where μ_j^{m} is split into a function f_j and a measure μ^{a}, and where the measure is made as simple as possible.

The power throughput measure. We now consider another interesting measure called the *power throughput measure* (μ^{p}), which is obtained from the previous measure by omitting the importance function $W_{\mathrm{e}}^{(j)}$. Explicitly, it is defined for paths of length k by

$$\begin{aligned} \mu_k^{\mathrm{p}}(D) \;=\; \int_D & L_{\mathrm{e}}(\mathbf{x}_0 \to \mathbf{x}_1)\, G(\mathbf{x}_0 \leftrightarrow \mathbf{x}_1)\, f_{\mathrm{s}}(\mathbf{x}_0 \to \mathbf{x}_1 \to \mathbf{x}_2)\, G(\mathbf{x}_1 \leftrightarrow \mathbf{x}_2) \cdots \\ & \cdots\; f_{\mathrm{s}}(\mathbf{x}_{k-2} \to \mathbf{x}_{k-1} \to \mathbf{x}_k)\, G(\mathbf{x}_{k-1} \leftrightarrow \mathbf{x}_k)\; dA(\mathbf{x}_0) \cdots dA(\mathbf{x}_k)\,, \end{aligned} \tag{8.15}$$

where $D \subset \Omega_k$, and then extended to a measure μ^{p} over the whole path space by the same technique we used for the area-product measure (8.6).

Physically, $\mu^{\mathrm{p}}(D)$ represents the power that is carried by a set of paths D (units: $[\mathrm{W}]$). A nice property of this measure is that it is independent of any sensor: there is only one measure for the whole scene, rather than one per sensor (as with μ_k^{m}). It can still be used to evaluate measurements, however, using the relationship

$$I_j \;=\; \int_\Omega W_{\mathrm{e}}^{(j)}(\mathbf{x}_{k-1} \to \mathbf{x}_k)\, d\mu^{\mathrm{p}}(\bar{x})\,.$$

This equation shows that I_j can be split into a function and a measure in more than one way. In this case, we have moved almost all the factors of f_j into the integration measure, leaving only $W_{\mathrm{e}}^{(j)}$ as the “measurement contribution function”.

The scattering throughput measure. Next, we discuss the *scattering throughput measure* μ^{s}. The value $\mu^{\mathrm{s}}(D)$ represents the power-carrying capacity of a set of paths D, in the following sense: if a uniform radiance L_{e} is emitted along the first segment of each path in D, then the power carried by these paths and received by surfaces at the path endpoints will be

$$L_{\mathrm{e}}\, \mu^{\mathrm{s}}(D)\,.$$

The definition of μ^{s} is identical to the previous measure (8.15), except that the emitted radiance function L_{e} is omitted (as well as the importance function $W_{\mathrm{e}}^{(j)}$). A nice property of this measure is that it depends only on the scene geometry and materials, not on the light sources or sensors. The units of $\mu^{\mathrm{s}}(D)$ are $[\mathrm{m}^2 \cdot \mathrm{sr}]$.

The geometric throughput measure. Finally, we consider the *geometric throughput measure* μ^{g}, which measures the geometric "size" of a set of paths. To do this, we start with the expression for the scattering throughput μ^{s}, and set all of the BSDF factors to the constant value

$$f_{\mathrm{s}}(\mathbf{x}_{i-1} \to \mathbf{x}_i \to \mathbf{x}_{i+1}) \;=\; \frac{1}{2\pi} \,.$$

Physically, this corresponds to a scene where the surfaces scatter light in all directions uniformly; the value $1/(2\pi)$ ensures that f_{s} is energy-preserving (see Section 6.3).[7] With this modification to the scattering throughput measure μ^{s}, any differences in the power-carrying capacity of different path sets are due entirely to their geometry.

Explicitly, the geometric throughput measure μ^{g} is defined at each path length k by

$$\mu_k^{\mathrm{g}}(D) \;=\; \left(\frac{1}{2\pi}\right)^{k-1} \int_D G(\mathbf{x}_0 \leftrightarrow \mathbf{x}_1) \,\cdots\, G(\mathbf{x}_{k-1} \leftrightarrow \mathbf{x}_k)\; dA(\mathbf{x}_0) \cdots dA(\mathbf{x}_k) \,, \tag{8.16}$$

and extended to a measure μ^{g} over the whole path space as before. The term *geometric throughput measure* is particularly appropriate for μ^{g}, since it is a natural extension of the throughput measure μ defined on the space of rays (see Section 4.1): these two measures are identical for paths of length one. The units of μ^{g} are the same as the previous measure, namely $[\mathrm{m}^2 \cdot \mathrm{sr}]$.

Notice that μ^{g} has several properties that we should expect of a geometric measure on paths. First, it does not encode any preference for directional scattering at surfaces (since this is a property of materials rather than geometry). Second, in general the measure μ^{g} is not finite, even for scenes with finite surface area.[8] This corresponds to the fact that there is no geometric reason for light energy to diminish as it propagates over long paths.

In fact, by comparing the scattering and geometric throughput measures, it is possible to determine whether the power-carrying capacity of a given set of paths is limited primarily by materials or geometry. A suitable quantitative measure of this is the ratio

$$\mu^{\mathrm{s}}(D) \,/\, \mu^{\mathrm{g}}(D) \,.$$

[7]This type of surface has the same radiance when viewed from all directions, on both sides of the surface. In an environment where only reflection is allowed, i.e. where all surfaces are one-sided, the BRDF would be $f_{\mathrm{r}} = 1/\pi$ instead.

[8]If the scene has finite area, then $\mu_k^{\mathrm{g}}(\Omega_k)$ will be finite for each path length k. However, when we take the union Ω over all path lengths, the resulting space has infinite geometric measure.

The area-product measure. Finally, we return to the area-product measure μ^{a}. The chief advantage of this measure is that it is simple. This makes it easy to compute the probabilities of various sampling techniques with respect to this measure, so that we may compare them. Like the geometric throughput measure μ^{g}, the area-product measure is in general not finite.

Appendix 8.B Subclassification of specular vertices

Specular vertices can be subclassified into two categories, according to whether light is scattered into a zero- or one-dimensional set of directions. We distinguish between these possibilities with the symbols S_0 and S_1. This notation allows the properties of sources, sensors, and materials to be specified more precisely.

We first consider light sources, which are represented by a string of the form LXY. The first symbol X represents the physical extent of the light source, so that S_0 denotes a point source, while S_1 denotes a linear, ring, or other one-dimensional source. The second symbol Y represents the set of directions over which light is emitted. The symbol S_0 denotes emission in a discrete set of directions, while S_1 denotes emission into a plane or other one-dimensional set. A similar classification applies to sensors, which are represented by a string of the form YXE. Several examples are given in Table 8.3.

For scattering events, S_0 denotes a surface that scatters light from an incoming direction $\omega_{\rm i}$ into a discrete of directions (e.g. a mirror or a window). The symbol S_1 denotes a surface such as an ideal anisotropic reflector, where light from an incoming direction $\omega_{\rm i}$ is scattered into a one-dimensional set of outgoing rays.

For example, the full-path regular expression

$$L\, S_1\, D\, S_0^*\, S_0\, D\, E$$

represents a path where light is emitted from a linear source, bounces off zero or more mirrors, and then is measured by a camera with an orthographic lens.

Formal definitions of S_0, S_1, and D. For completeness, we give formal definitions of these symbols. Consider a scattering event at a vertex $\mathbf{x}_i$. As we have already mentioned, this vertex has type D is the BSDF at $\mathbf{x}_i$ is finite:

$$f_{\rm s}(\mathbf{x}_i, \omega_{\rm i} \!\to\! \omega_{\rm o}) \;<\; \infty\,,$$

where $\omega_{\rm i}$ and $\omega_{\rm o}$ are the directions toward $\mathbf{x}_{i-1}$ and $\mathbf{x}_{i+1}$ respectively.

The scattering event at $\mathbf{x}_i$ is defined to be S_0 whenever the BSDF behaves locally like a two-dimensional Dirac distribution (as was used to define the BSDF for mirror reflection in

LS_0D	a uniform point source, point spotlight, etc.
LS_0S_1	emission from a point into a planar fan or sheet
LS_0S_0	an idealized laser beam
LS_1D	a typical linear or ring source
LS_1S_1	an area light source in “flatland” [Heckbert 1990]
LDS_0	sunshine through a window
DS_0E	a typical pinhole lens model
DS_1E	a pinhole lens with motion blur due to movement of the camera (in a static scene)
S_0DE	an orthographic viewing projection
S_0S_0E	an idealized spot meter

Table 8.3: Examples of regular expressions for light sources and sensors, where the specular components have been subclassified into zero- and one-dimensional components.

Section 5.2.1.2). More precisely, this happens when there is a constant $\epsilon > 0$ such that

$$\int_D f_{\mathrm{s}}(\mathbf{x}_i, \omega \to \omega_{\mathrm{o}}) \, d\sigma^{\perp}(\omega) \;\geq\; \epsilon$$

for every open set $D \subset \mathcal{S}^2$ that contains ω_{i}.

Finally, a vertex is defined to be S_1 if it is not S_0 or D. It is straightforward to extend these definitions to the classification of light sources and sensors, using the functions $L_{\mathrm{e}}^{(i)}$ and $W_{\mathrm{e}}^{(i)}$ defined in Section 8.3.2.