8

The Beam Radiance Estimate

"Scattering is easier than gathering."

—Irish Proverb

THE volumetric photon mapping [Jensen and Christensen, 1998] technique described in the previous chapter can efficiently simulate scattering in participating media without making simplifying assumptions about the properties of the medium being rendered. Similar to the volumetric radiance caching technique developed in Chapter 5, photon mapping handles isotropic, anisotropic, homogeneous, and heterogeneous media of arbitrary albedo. Photon mapping gains efficiency by reusing a small collection of photons to estimate in-scattered radiance at all locations in the scene using density estimation.

Just like in volumetric radiance caching, a ray marching process is used to integrate the contribution of radiance directly seen by the camera. In volumetric photon mapping, however, the radiance estimate, which is evaluated at each step during ray marching, requires costly range queries within the photon map. Minimizing the number of queries is desirable; however, if not enough sample points are used, the result is likely to be noisy. On the other hand, increasing the number of sample points is very costly and can slow down rendering significantly. Even with a fixed number of samples, finding an optimal distribution of sample points along the ray is difficult. Moreover, the ray marching formulation is suboptimal, firstly because it may gather the same photons more than once if the spherical neighborhoods overlap and, secondly, because it can lead to noise if the step size is too large and photons are omitted (see Figure 8.1).

In this chapter we develop a novel radiance estimate technique for participating me-

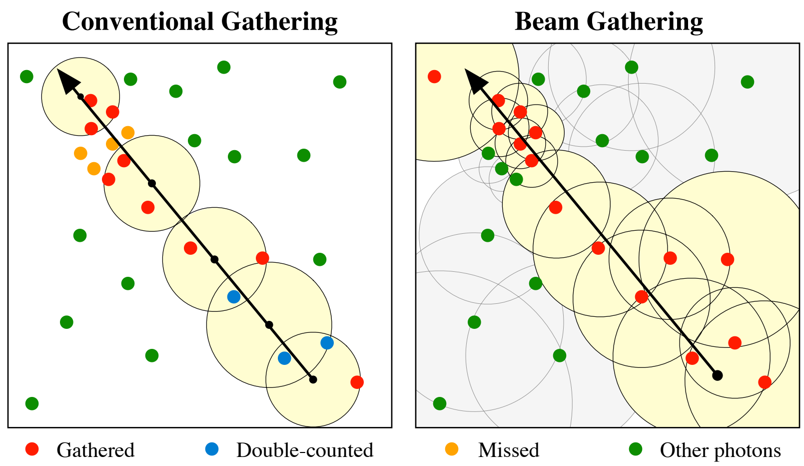

Figure 8.1: Conventional gathering (left) searches for photons in a sphere around numerous samples along the ray. This is inefficient because it can double-count photons if the searches overlap (blue) and it can miss important photons (orange) if the step-size becomes too large. The method described in this chapter (right) assigns a radius to each photon and performs a single range search to find all photons along the length of the entire ray.

dia, which eliminates this problem. To accomplish this, we use a theoretical reformulation of volumetric photon mapping. The technique developed in this chapter replaces the multiple point-queries performed during ray marching with a single beam-query, which explicitly gathers all photons along the length of an entire ray. These photons are used to estimate the accumulated in-scattered radiance arriving from a particular direction and need to be gathered only once per ray. This method handles both fixed and adaptive kernels, is significantly faster than conventional volumetric photon mapping, and produces images with less noise.

8.1 Contributions

In this chapter, we propose a novel approach for computing the contribution of in-scattered radiance. We gather photons along viewing rays and analytically compute their contributions, without point sampling. We present the following contributions:

- We combine the theory from Veach [1997] and Pauly et al. [2000] to derive a reformulation of volumetric photon mapping as a solution to the measurement equation. This theory allows for arbitrary *measurements* of radiance to be computed within participating media, where a measurement is simply an integral of the radiance multiplied with a weighting function.

- Using this new theory, we present an improved radiance estimate for volumetric photon mapping based on "beam gathering." This technique eliminates the need for stepping through the medium to find photons. Instead, it gathers all photons along a ray. We show how to efficiently implement this new gathering technique for both fixed and adaptive smoothing kernels and demonstrate that our method produces images with less noise than conventional photon mapping.

The rest of this chapter is organized as follows. In Section 8.2, we reformulate volumetric photon mapping in terms of the measurement equation and show how the photon map can be used to estimate *any* measurement of radiance within the scene. In Section 8.3, we present our new beam radiance estimate using this theory and describe the data structures needed to evaluate it efficiently. Finally, we show comparisons of our approach to conventional photon mapping in Section 8.4 and discuss avenues of future work in Section 8.5.

8.2 Reformulation of Volumetric Photon Mapping

Our technique queries once for photons along the length of an entire ray, instead of multiple times near points along the ray (see Figure 8.1). More formally, whereas regular photon mapping estimates L_i at discrete points using Equation 7.10, our main contribution is to directly estimate

$$\int_0^s T_r(\mathbf{x}\leftrightarrow\mathbf{x}_t)\,\sigma_s(\mathbf{x}_t)\,L_i(\mathbf{x}_t\rightarrow\vec{\omega})\,dt \tag{8.1}$$

along rays.

Though the explanation of photon mapping from the previous chapter is appealing at an intuitive level, it does not rigorously present the algorithm as a numerical solution to the RTE. Furthermore, this explanation is heavily tied to the geometric interpretation of gathering photons within a disc (on surfaces) or within a sphere (in participating media). In order to avoid these limitations and use the photon map to estimate general radiometric quantities in the volume, such as Equation 8.1, we use a more flexible derivation of particle tracing methods presented by Veach [1997]. We extend this derivation to handle participating media by combining it with the generalized path integral formulation of the radiative transport equation [Pauly et al., 2000] (Section 8.2.1) and show how to represent particle tracing algorithms like volumetric photon mapping in terms of the measurement equation (Sections 8.2.2 and 8.2.3). Finally, we show how to use the same photon maps to estimate more general quantities of radiance (Section 8.2.4).

8.2.1 Generalized Path Integral Formulation

We use the path integral formulation of the RTE, which arises by recursively expanding the right hand side of Equation 4.27. Instead of expressing the radiance equilibrium *recursively*, the resulting path integral formulation is a *sum* over light-carrying paths of different lengths. In order to do this, we define the path space, the corresponding differential measure, and generalized radiometric terms using notation inspired by Pauly et al. [2000].

A light path $\bar{\mathbf{x}}_k^l$ is a set of $k+1$ vertices $\mathbf{x}_i$. Each path is classified according to its *path characteristic* $l \in \mathbb{N}$, which determines for each vertex whether it is in the volume or on a surface. This allows us to integrate over different measures for scattering events at surfaces and within the volume. We define the path characteristic l of a path $\bar{\mathbf{x}}_k^l$ such that the i$^{\text{th}}$ bit of l, $b_i(l)$, equals 1 if vertex $\mathbf{x}_i$ is on a surface and $b_i(l) = 0$ if it is in the volume. The space of all paths of length k with characteristic l is therefore:

$$\mathbf{X}_k^l = \left\{ \bar{\mathbf{x}}_k^l = \mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_k \;\middle|\; \mathbf{x}_i \in \begin{cases} \mathcal{V}, & \text{if } b_i(l) = 0 \\ \mathcal{A}, & \text{if } b_i(l) = 1 \end{cases} \right\}, \tag{8.2}$$

for $1 \le k < \infty$ and $0 \le l < 2^{k+1}$, and where $\mathcal{V}$ and $\mathcal{A}$ are the media volume and surface area, respectively (see Figure 8.2 for an illustration of this notation). We define the corresponding differential measure at a path vertex $\mathbf{x}_i$ as:

$$d\mu^l(\mathbf{x}_i) = \begin{cases} d\mathcal{V}(\mathbf{x}_i), & \text{if } \mathbf{x}_i \in \mathcal{V}, \text{ i.e. when } b_i(l) = 0 \\ d\mathcal{A}(\mathbf{x}_i), & \text{if } \mathbf{x}_i \in \mathcal{A}, \text{ i.e. when } b_i(l) = 1 \end{cases}. \tag{8.3}$$

Additionally, in order to express Equation 4.27 in terms of paths we need to transform the integration domain from solid angle (Ω and $\Omega_{4\pi}$), to area or volume ($\mathcal{A}$ and $\mathcal{V}$) depending on the type of scattering event. This transformation is achieved using the generalized geometry term:

$$\hat{G}(\mathbf{x}\leftrightarrow\mathbf{y}) = \frac{V(\mathbf{x}\leftrightarrow\mathbf{y})D_{\mathbf{x}}(\mathbf{y})D_{\mathbf{y}}(\mathbf{x})\sigma(\mathbf{x})T_r(\mathbf{x}\leftrightarrow\mathbf{y})}{\left\|\mathbf{x}-\mathbf{y}\right\|^2} \tag{8.4}$$

where

$$D_{\mathbf{x}}(\mathbf{y}) = \begin{cases} 1, & \text{if } \mathbf{x} \in \mathcal{V} \\ \vec{\mathbf{n}}(\mathbf{x}) \cdot \vec{\omega}_{\mathbf{xy}}, & \text{if } \mathbf{x} \in \mathcal{A} \end{cases}. \tag{8.5}$$

We define $\vec{\omega}_{\mathbf{xy}}$ to be the unit direction vector from $\mathbf{x}$ to $\mathbf{y}$, and $D_{\mathbf{y}}(\mathbf{x})$ is defined symmetrically to $D_{\mathbf{x}}(\mathbf{y})$ for both cases. The visibility function, V, is defined in Equation 2.19[1]. The $\sigma(\mathbf{x})$ function returns the scattering coefficient $\sigma_s(\mathbf{x})$ if $\mathbf{x} \in \mathcal{V}$ and 1 otherwise. Note that Equation 8.4 simplifies to the regular geometry term if no participating media is present.

Similarly, we generalize scattering events and emitted radiance. We define $\hat{f}$ to be the generalized scattering function combining the phase function and the surface BRDF

$$\hat{f}(\mathbf{x}_i) = \begin{cases} p(\mathbf{x}_{i+1}\rightarrow\mathbf{x}_i\rightarrow\mathbf{x}_{i-1}), & \text{if } \mathbf{x}_i \in \mathcal{V} \\ f_r(\mathbf{x}_{i+1}\rightarrow\mathbf{x}_i\rightarrow\mathbf{x}_{i-1}), & \text{if } \mathbf{x}_i \in \mathcal{A} \end{cases}, \tag{8.6}$$

and $\hat{L}_e$ is the generalized emitted radiance

$$\hat{L}_e(\mathbf{x}_i\rightarrow\mathbf{x}_{i-1}) = \begin{cases} \frac{\sigma_a(\mathbf{x}_i)}{\sigma_s(\mathbf{x}_i)} L_e(\mathbf{x}_i\rightarrow\mathbf{x}_{i-1}), & \mathbf{x}_i \in \mathcal{V} \\ L_e(\mathbf{x}_i\rightarrow\mathbf{x}_{i-1}), & \mathbf{x}_i \in \mathcal{A} \end{cases}, \tag{8.7}$$

where we multiply by $\frac{\sigma_a}{\sigma_s}$ because the emitted radiance in a volume needs to be multiplied by the absorption, not the scattering, coefficient.

Given this notation, the generalized path integral formulation expresses the outgoing radiance at $\mathbf{x}_1$ towards $\mathbf{x}_0$ as a sum over all possible light paths arriving at $\mathbf{x}_1$. This includes light paths of all lengths k, as well as all possible characteristics l for each length

$$L(\mathbf{x}_1\rightarrow\mathbf{x}_0) = \sum_{k=1}^{\infty}\sum_{l=0}^{2^{k+1}-1} \bar{L}(\bar{\mathbf{x}}_k^l). \tag{8.8}$$

[1]Note that unlike G, to simplify notation we include the visibility function in $\hat{G}$.

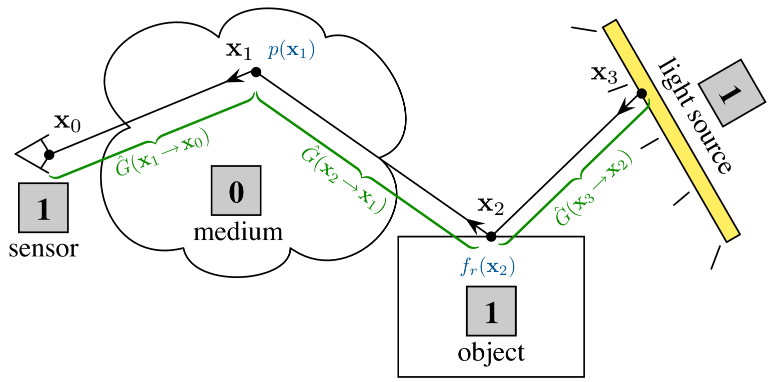

Figure 8.2: An example path, $\bar{\mathbf{x}}_3^{13}$, with length $k = 3$. The path characteristic, $l = 1101_b = 13$, concatenates the type of scattering event from each path vertex. The radiance transported along the path $\bar{L}(\bar{\mathbf{x}}_3^{13})$ is the emitted radiance at the light multiplied by a series of scattering events (blue) and geometry terms (green).

$\bar{L}(\bar{\mathbf{x}}_k^l)$ measures the amount of radiance transported along a path $\bar{\mathbf{x}}_k^l$ and is defined as

$$\bar{L}(\bar{\mathbf{x}}_k^l) = \int_{\mu^l(\mathbf{x}_k)} \cdots \int_{\mu^l(\mathbf{x}_2)} \hat{L}_e(\mathbf{x}_k \to \mathbf{x}_{k-1}) \left(\prod_{j=1}^{k-1} \hat{f}(\mathbf{x}_j) \hat{G}(\mathbf{x}_{j+1} \leftrightarrow \mathbf{x}_j) \right) d\mu^l(\mathbf{x}_2) \cdots d\mu^l(\mathbf{x}_k). \tag{8.9}$$

The radiance transported along an example path is shown in Figure 8.2.

8.2.2 The Measurement Equation

Many global illumination algorithms can be described in terms of the *measurement equation*. The measurement equation describes an abstract measurement of incident radiance taken over some set of rays in a scene:

$$I = \langle W_e, L \rangle = \int_{\mathcal{V}} \int_{\Omega_{4\pi}} W_e(\mathbf{x} \to \vec{\omega})\, L(\mathbf{x} \leftarrow \vec{\omega})\, d\vec{\omega}\, d\mathcal{V}(\mathbf{x}). \tag{8.10}$$

The importance function W_e represents an abstract measuring sensor and is defined over the whole ray space $\mathcal{V} \times \Omega_{4\pi}$ (though typically W_e is non-zero for only a small subset of this domain).

Path tracing, for instance, measures the contribution of radiance arriving over the area of a pixel. Radiosity algorithms integrate the contribution of radiance over basis functions defined on the scene geometry. Both of these approaches can be described using Equation 8.10 with an appropriate importance function.

In his dissertation, Veach [1997] showed how particle tracing methods for surface illumination can also be expressed as a solution to the measurement equation by using the path integral form of the rendering equation. We extend this idea and use the generalized path integral formulation to describe volumetric photon tracing in the same way.

8.2.3 Volumetric Photon Tracing

Photon tracing methods can be thought of as a way of generating samples from the scene's equilibrium radiance distribution and then using this single collection of samples to render the entire image. The photon tracing stage generates N weighted sample rays, or photons, $(\alpha_i, \mathbf{x}_i, \vec{\omega}_i)$, where each $(\mathbf{x}_i, \vec{\omega}_i)$ is a ray and α_i is a corresponding weight. Our goal is to use these samples to take *unbiased* estimates of the radiance as a weighted sum,

$$E\left[\frac{1}{N}\sum_{i=1}^{N} W_e(\mathbf{x}_i \to \vec{\omega}_i)\,\alpha_i\right] = \langle W_e, L\rangle, \tag{8.11}$$

for an arbitrary importance function W_e. We must therefore determine the proper distribution of samples for Equation 8.11 to hold.

We start by manipulating the measurement equation on the right-hand side. To express the measurement equation using paths, we expand Equation 8.10 in terms of the *outgoing* radiance and convert to area form, introducing an additional geometry term,

$$\langle W_e, L\rangle = \int_{\mu^l(\mathbf{x}_1)} \int_{\mu^l(\mathbf{x}_0)} W_e(\mathbf{x}_0 \to \mathbf{x}_1)\, L(\mathbf{x}_1 \to \mathbf{x}_0)\, \hat{G}(\mathbf{x}_1 \leftrightarrow \mathbf{x}_0)\; d\mu^l(\mathbf{x}_0)\; d\mu^l(\mathbf{x}_1). \tag{8.12}$$

By combining with Equations 8.8 and 8.9 and moving the summations outside the integrals, this becomes

$$\sum_{k=1}^{\infty} \sum_{l=0}^{2^{k+1}-1} \left[\int_{\mu^l(\mathbf{x}_k)} \cdots \int_{\mu^l(\mathbf{x}_0)} W_e(\mathbf{x}_0 \to \mathbf{x}_1)\, \hat{L}_e(\mathbf{x}_k \to \mathbf{x}_{k-1}) \left(\prod_{j=1}^{k-1} \hat{f}(\mathbf{x}_j)\, \hat{G}(\mathbf{x}_{j+1} \leftrightarrow \mathbf{x}_j) \right) \hat{G}(\mathbf{x}_1 \leftrightarrow \mathbf{x}_0)\, d\mu^l(\mathbf{x}_0) \cdots d\mu^l(\mathbf{x}_k) \right].$$

The Monte Carlo estimator for this expression using N samples is

$$E\left[\frac{1}{N}\sum_{i=1}^{N} W_e(\mathbf{x}_{i,0} \to \mathbf{x}_{i,1}) R_i\right], \tag{8.13}$$

where each R_i is a random-walk path of length k_i generated using Russian roulette,

$$R_i = \frac{\hat{L}_e(\mathbf{x}_{i,k_i} \to \mathbf{x}_{i,k_{i-1}})}{pdf(\mathbf{x}_{i,k_i}, \mathbf{x}_{i,k_i-1})} \prod_{j=1}^{k_i-1} \left(\frac{1}{q_{i,j}} \frac{\hat{f}(\mathbf{x}_{i,j})\, \hat{G}(\mathbf{x}_{i,j+1} \leftrightarrow \mathbf{x}_{i,j})}{pdf(\mathbf{x}_{i,j-1})} \right) \hat{G}(\mathbf{x}_{i,1} \leftrightarrow \mathbf{x}_{i,0}), \tag{8.14}$$

and $q_{i,j}$ was the probability of terminating R_i at the j^{th} vertex. Comparing Equation 8.13 with 8.11 we see that in order to satisfy the requirements we need to set $\alpha_i = R_i$.

Connection to Conventional Photon Tracing. Though derived in a different fashion, Equation 8.14 is exactly how conventional photon mapping distributes photons within the scene. For instance, for a diffuse area light, photons are emitted using a cosine distribution with the power of the light source. In Equation 8.14, photons are emitted with the radiance of the light source divided by the pdf of choosing a position and direction on the light. These quantities are equivalent. Hence the particles generated above represent differential flux. The correspondence between the photon powers [Jensen and Christensen, 1998] and the sample weights is $\Delta\Phi_i = \frac{\alpha_i}{N}$.

8.2.4 Radiance Estimation Using the Measurement Equation

The main advantage of the reformulation in Section 8.2.3 is that it naturally accommodates computation of any measurement of radiance within the scene simply by using an appropriately defined importance function W_e. In this section, we first show how the conventional estimate for in-scattered radiance can be expressed as a measurement. We then go one step further and show how to derive a beam radiance estimate which approximates Equation 8.1 along rays directly.

Conventional Radiance Estimate. The conventional radiance estimate approximates the value of the in-scattered radiance L_i at fixed points within the scene. To express this using the theory from Section 8.2, we need to transform Equation 4.13 into the measurement equation. Since the measurement equation is an integral over all of ray space ($\mathcal{V} \times \Omega_{4\pi}$), we artificially expand L_i to also integrate over the volume

$$L_i(\mathbf{x}_t \to \vec{\omega}) = \int_{\Omega_{4\pi}} p(\mathbf{x}_t, \vec{\omega} \leftrightarrow \vec{\omega}_t) L(\mathbf{x}_t \leftarrow \vec{\omega}_t) d\vec{\omega}_t \tag{8.15}$$
$$= \int_{\mathcal{V}} \int_{\Omega_{4\pi}} \delta(\|\mathbf{x}' - \mathbf{x}_t\|)\, p(\mathbf{x}', \vec{\omega} \leftrightarrow \vec{\omega}_t)\, L(\mathbf{x}_t \leftarrow \vec{\omega}_t) d\vec{\omega}_t d\mathcal{V}(\mathbf{x}').$$

In order to keep the expressions equivalent when we add the integration over volume, we also introduce a Dirac delta function δ.

The bottom row of the above equation is now the measurement equation, where $W_e = \delta(\|\mathbf{x}' - \mathbf{x}_t\|)\, p(\mathbf{x}', \vec{\omega} \leftrightarrow \vec{\omega}_t)$. Hence we can compute an *unbiased* estimate using the photon map by evaluating Equation 8.11 with this importance function. However, in order to obtain a useful estimate of radiance at all points in the scene, a normalized kernel function is used in place of the delta function. This is where bias is introduced in the photon mapping method. Another interpretation is that by replacing the delta function with a kernel, photon mapping computes an *unbiased* estimate of *blurred* radiance. Jensen and Christensen [1998] use a constant three-dimensional kernel with a radius based on the k^{th} nearest neighbor. This results in the following radiance estimate by applying Equation 8.11

$$L_i(\mathbf{x}_t \to \vec{\omega}) \approx \int_{\mathcal{V}} \int_{\Omega_{4\pi}} K_t(\|\mathbf{x}' - \mathbf{x}_t\|)\, p(\mathbf{x}', \vec{\omega} \leftrightarrow \vec{\omega}_t)\, L(\mathbf{x}_t \leftarrow \vec{\omega}_t) d\vec{\omega}_t d\mathcal{V}(\mathbf{x}'), \tag{8.16}$$
$$\approx \frac{1}{N} \sum_{i=1}^{N} K_t(\|\mathbf{x}_i - \mathbf{x}_t\|)\, p(\mathbf{x}_i, \vec{\omega} \leftrightarrow \vec{\omega}_i) \alpha_i \tag{8.17}$$

where the kernel K_t is defined as

$$K_t(r) = \begin{cases} \frac{3}{4\pi d_k(\mathbf{x}_t)^3} & \text{if } r \in [0, d_k(\mathbf{x}_t)] \\ 0 & \text{otherwise} \end{cases}, \tag{8.18}$$

and $d_k(\mathbf{x}_t)$ is the distance from $\mathbf{x}_t$ to the k^{th} nearest photon. Note that this is equivalent to the conventional volumetric radiance estimate in Equation 7.10.

Beam Radiance Estimate. A similar procedure can be used to derive an estimate for the accumulated in-scattered radiance along an entire ray. To accomplish this, we first expand out L_i

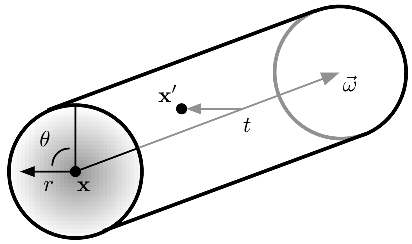

Figure 8.3: In the beam radiance estimate, $\mathbf{x}'$ is parametrized in cylindrical coordinates, (t, θ, r), about the ray $(\mathbf{x}, \vec{\omega})$. An unbiased estimate would only consider points directly on the ray, while a biased version uses a kernel (shown in grey) to blur the radiance within a beam.

in Equation 8.1 and then artificially inflate the resulting expression to integrate over the whole volume:

$$\int_0^s \int_{\Omega_{4\pi}} T_r(\mathbf{x} \leftrightarrow \mathbf{x}_t)\, \sigma_s(\mathbf{x}_t)\, p(\mathbf{x}_t, \vec{\omega} \leftrightarrow \vec{\omega}_t)\, L(\mathbf{x}_t \leftarrow \vec{\omega}_t) d\vec{\omega}_t\, dt = \tag{8.19}$$

$$\int_{\mathbb{R}} \int_0^{2\pi} \int_{\mathbb{R}} \int_{\Omega_{4\pi}} \delta(r)\, (H(t) - H(t-s))\, T_r(\mathbf{x} \leftrightarrow \mathbf{x}')\, \sigma_s(\mathbf{x}')\, p(\mathbf{x}', \vec{\omega} \leftrightarrow \vec{\omega}_t)\, L(\mathbf{x}' \leftarrow \vec{\omega}_t) d\vec{\omega}_t dr d\theta dt. \tag{8.20}$$

$\mathbb{R}$ is the set of real numbers and $\mathbf{x}'$ is expressed in cylindrical coordinates, (t, θ, r), about $(\mathbf{x}, \vec{\omega})$, where r is the radius to the ray (see Figure 8.3). We have added a Dirac delta function δ as before, and the Heaviside step functions ($H(x) = 1$ when $x > 0$ and 0 otherwise) constrain the computation to $t \in (0, s)$. Equation 8.20 is now equivalent to the measurement equation, where the integral over volume has been converted into cylindrical coordinates and where $W_e = \delta(r)(H(t) - H(t-s))\, T_r(\mathbf{x} \leftrightarrow \mathbf{x}') \sigma_s(\mathbf{x}')\, p(\mathbf{x}', \vec{\omega} \leftrightarrow \vec{\omega}_t)$.

Since the probability of photons landing exactly on the ray $(\mathbf{x}, \vec{\omega})$ is zero, we introduce bias by blurring the radiance and replacing the delta and step functions with a smooth kernel, K. This integral can then be estimated with the measurement equation using the photons as

$$\int_{\mathbb{R}} \int_0^{2\pi} \int_{\mathbb{R}} \int_{\Omega_{4\pi}} K(t, \theta, r)\, T_r(\mathbf{x} \leftrightarrow \mathbf{x}')\, \sigma_s(\mathbf{x}')\, p(\mathbf{x}', \vec{\omega} \leftrightarrow \vec{\omega}_t)\, L(\mathbf{x}' \leftarrow \vec{\omega}_t) d\vec{\omega}_t dr d\theta dt =$$
$$\frac{1}{N} \sum_{i=1}^{N} K(t_i, \theta_i, r_i)\, T_r(\mathbf{x} \leftrightarrow \mathbf{x}_i)\, \sigma_s(\mathbf{x}_i)\, p(\mathbf{x}_i, \vec{\omega} \leftrightarrow \vec{\omega}_i)\, \alpha_i, \tag{8.21}$$

where $(t_i, \theta_i, r_i) = \mathbf{x}_i$ are the cylindrical coordinates of photon i about the ray.

The blurring in the conventional radiance estimate is spherical, and so the kernel needs

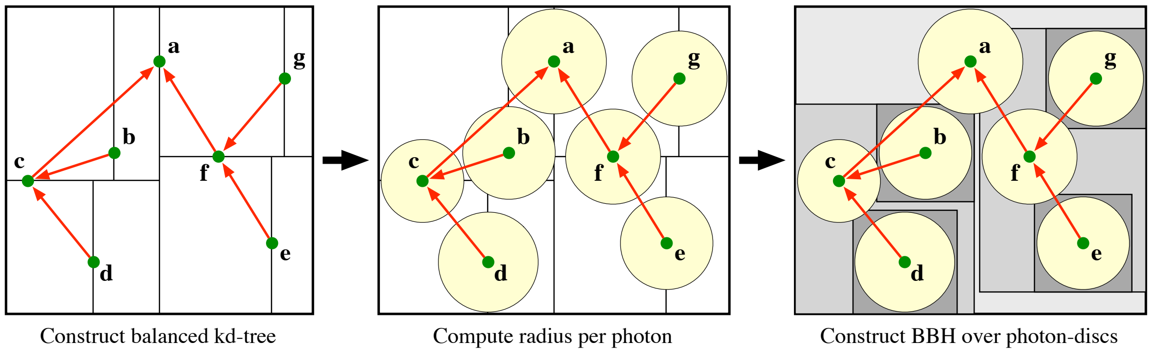

Figure 8.4: After photons have been distributed in the scene, our algorithm constructs a balanced kd-tree (left). We assign a valid radius to each photon by querying in the kd-tree (middle). Finally, we rapidly construct a bounding-box hierarchy over the photon-discs (right) by reusing the same hierarchical structure (shown in red) as the kd-tree.

to be normalized for 3D. However, with the beam radiance estimate, we blur in two dimensions (perpendicular to the ray) since the radiance we are computing already includes the integration along the ray itself. Therefore, the kernel in the beam estimate is normalized for 2D.

8.2.5 Kernel Radiance Estimation

For both the conventional and beam radiance estimates, the characteristics of the bias and blur are determined by the smoothing function chosen. Several options exist for applying a smoothing kernel to the photon map data.

The *kernel method* uses a fixed-radius smoothing kernel and results in a uniform blur of radiance within the scene. In practice, using a fixed-width circular kernel implies that in order to evaluate the beam radiance estimate (Equation 8.21) using the photon map (Equation 8.11) we only need to consider photons which are located within a fixed-radius cylinder about the ray $(\mathbf{x}, \vec{\omega})$. Alternatively, the equivalent dual interpretation considers each photon as the center of an oriented disc *facing the ray* and all *photon-discs* that intersect the ray need to be found.

If the density of photons varies significantly it can be difficult to choose a single radius that works well for all regions of the scene. This can be solved by allowing the size and shape of the blurring kernel to vary spatially. In conventional photon mapping, the k^{th} *nearest neighbor method* (k-NN) is used to adapt the kernel width to the local density. Generalizing point-based k-NN to a visually comparable range search along rays is challenging. However, spatial variation can easily be applied to the dual *photon-discs* interpretation using the *variable kernel method* (VK) [Breiman et al., 1977]. A smoothing kernel is "attached" to each photon, and the radius of the kernel is allowed to vary from one photon to another, based on the local density. In contrast to k-NN estimation, where the kernel widths vary based on the distance from the *evaluation* location to the data points, in the VK method the kernel widths only depend on the data points themselves. In order to facilitate this, the method relies on a *pilot estimate* of the local density at each data point in order to assign the kernel widths. We review these and other density estimation methods in more detail in Appendix C. This is the approach we take.

8.3 Algorithm

In order to use the dual interpretation to evaluate the beam radiance estimate (Equation 8.21), we need an efficient way of locating all photon-discs that intersect an arbitrary ray. Additionally, to use variable width kernels we need to efficiently compute a radius for each photon in the photon map. At a high level, our volumetric photon mapping technique performs the five steps in Algorithm 8.1. Steps 1 and 2 are identical to conventional photon mapping, while 3–5 are unique to our approach and are explained in more detail below.

Photon Radius Computation. We augment the traditional photon map by associating a radius with each photon. For fixed width kernels the radius is a constant for all photons and does not need to be explicitly stored. For variable width kernels using the VK method, we perform a density estimation at each photon to assign a radius. At each photon we compute the local density by estimating the distance to the k^{th} nearest photon and use this as the photon-disc's radius. This pilot estimate is performed using the photon map kd-tree. The value k plays the same role as in the conventional radiance estimate: it controls the amount of blur.

As an optimization, we only search for the nearest $n \ll k$ photons and estimate the necessary radius for k photons. By assuming locally uniform photon density, if $d_n(\mathbf{x}_i)$ is the distance to the n^{th} photon from photon i, we estimate the radius as $r_i = d_n(\mathbf{x}_i)\sqrt[3]{\frac{k}{n}}$. The n parameter controls the sensitivity of the computed radius to the local variation in density. Very small values of n, $n < 5$, can produce noisy radii, which change drastically between neighboring

Algorithm 8.1: Beam photon mapping.

1 Shoot photons from light sources.
2 Construct a balanced kd-tree for the photons.
3 Assign a radius for each photon.
4 Construct a bounding-box hierarchy over the photon-discs.
5 Use the photon BBH to render the image.

photons, while large values are more expensive to compute. In practice, we have found that $n = \sqrt{k}$ works well as a default value, and this value was used for all our scenes, significantly accelerating the preprocessing step.

Bounding Box Hierarchy Construction. In order to efficiently locate all photons-discs which intersect a ray, we construct a bounding box hierarchy. Heuristics for constructing efficient BBHs have been extensively studied within the context of ray tracing [Wald et al., 2007]. However, the performance characteristics of our ray intersections are different than for regular ray tracing since we are interested in all intersections, not just the first intersection along a ray. Furthermore, the best heuristics tend to induce long construction times. We employ a rapid construction scheme by exploiting the information in the photon map kd-tree and reusing that hierarchy for our BBH.

For each photon in the photon map, we compute the bounding box of all descendant photon-discs. The bounding box of each node encloses the node's photon radius and the bounding boxes of its two child nodes. The computation starts at the leaves and propagates upwards through the tree. This procedure results in a balanced median-split-style BBH, but unlike traditional BBHs our hierarchy contains photons at interior nodes, not just at the leaves. Figure 8.4 illustrates the relationship between the kd-tree and the BBH. The BBH can be constructed by passing the root of the photon map kd-tree to Algorithm 8.2.

Given a balanced kd-tree, this linear-time construction procedure is extremely fast and produces BBHs which can be efficiently traversed for nearby photons during rendering. After the BBH is constructed the photon map kd-tree is no longer used and can be freed from memory. Using a BBH and a per-photon radius, an additional seven floating-point values need to be stored, increasing the size of each photon from 20 bytes to 46 bytes.

Algorithm 8.2: CONSTRUCTBBH(p)

Data: p is a node in a balanced photon map.
Result: The subtree at p contains a valid BBH.

```
begin
    B ⇐ BoundingBox(p.position, p.radius);
    if p.leftChild then
        B ⇐ B ∪ ConstructBBH(p.leftChild);
    end
    if p.rightChild then
        B ⇐ B ∪ ConstructBBH(p.rightChild);
    end
    p.bbox ⇐ B;
    return B
end
```

The Beam Radiance Estimate. During rendering we estimate the accumulated in-scattered radiance (Equation 8.1) along viewing rays by locating all photons whose bounding spheres intersect the ray. These photons are found using a standard ray-BBH intersection traversal. The contribution from each photon $(\alpha_i, \mathbf{x}_i, \vec{\omega}_i)$ is accumulated using Equation 8.21; however, with the variable kernel method, a kernel K_i is defined per photon. This leads to the following radiance estimate

$$\frac{1}{N}\sum_{i=1}^{N} K_i(\mathbf{x}, \vec{\omega}, s, \mathbf{x}_i, r_i)\, T_r(\mathbf{x} \leftrightarrow \mathbf{x}'_i)\, \sigma_s(\mathbf{x}'_i)\, p(\mathbf{x}_i, \vec{\omega} \leftrightarrow \vec{\omega}_i)\, \alpha_i, \tag{8.22}$$

Table 8.1: Rendering parameters and timings, in seconds, (s), and minutes, (m), for all example scenes. Statistics relating to the photon tracing preprocess are shown in the first set of columns. We compare our beam radiance estimate method (B) to conventional photon mapping (C) with both a fixed width kernel and an adaptive width kernel. The r column represents the fixed-width kernel radius, while r_+ is the maximum search radius and the number of nearest neighbors is k.

	Photon Tracing Preprocess				**Fixed Radius Render**				**Adaptive Radius Render**				
Scene	N	Shoot (s)	Balance (s)	Radius (s)	r	Δt	C (m)	B (m)	r_+	k	Δt	C (m)	B (m)
Cornell	0.4M	1.50	0.30	2.0	0.4	0.40	3:19	3:00	0.6	1.5K	0.80	4:03	3:35
Stage	1M	3.25	0.76	5.0	0.3	0.20	4:21	4:15	0.5	0.5K	0.70	6:38	6:22
Cars	2M	19.0	1.50	2.0	0.4	1.25	1:30	1:30	0.5	1K	1.25	2:02	1:53
Lighthouse	1M	2.83	0.78	6.0	0.4	0.25	1:07	0:59	0.5	0.4K	1.00	1:12	1:05

where $\mathbf{x}'_i = \mathbf{x} + t_i\vec{\omega}$ is the projection of the photon location $\mathbf{x}_i$ onto the ray's direction $\vec{\omega}$, and $t_i = (\mathbf{x}_i - \mathbf{x}) \cdot \vec{\omega}$. We define the kernel as

$$K_i(\mathbf{x}, \vec{\omega}, s, \mathbf{x}_i, r_i) = \begin{cases} r_i^{-2} K_2\left(\frac{d_i}{r_i}\right) & \text{if } d_i \in [0, r_i] \\ 0 & \text{otherwise} \end{cases}, \tag{8.23}$$

where r_i is the pre-computed radius for photon i. We use Silverman's two-dimensional biweight kernel [Silverman, 1986] along the ray, $K_2(x) = 3\pi^{-1}(1-x^2)^2$, where d_i is the shortest distance from photon i to the ray. We chose this kernel because it is smooth, efficient to evaluate, and has local support. Equation 8.22 is the beam radiance estimate, and it replaces the ray marching computation from conventional photon mapping (second term in Equation 7.9).

Heterogeneous Media. For homogeneous media, the transmission terms, $T_r(\mathbf{x} \leftrightarrow \mathbf{x}'_i)$, can be computed explicitly for each photon during gathering. Beam gathering in heterogeneous media can also be handled efficiently by first marching along the ray and saving the transmission values in a 1D lookup table. Then, during gathering, each photon's transmission is computed by interpolating within the lookup table. This preprocess needs to be performed independently for each ray, just prior to gathering, so the lookup table can be reused. Furthermore, if single scattering is simulated separately by directly sampling light sources, the lookup table can be populated during this marching step.

8.4 Results

We compared our beam gathering technique against conventional volumetric photon mapping using ray marching. In order to isolate just the performance of the photon gathering methods, we use the photon map for both single and multiple scattering. We compared results on four test scenes: Cars, Lighthouse, Stage, and a Cornell box. For each scene we compare using a fixed gathering radius for both types of estimates, and we also compare the conventional estimate using k-NN to the beam estimate using the VK method. The images were all rendered with a maximum dimension of 1024 pixels with up to four samples per pixels on an Intel Core 2 Duo 2.4 GHz machine using one core.

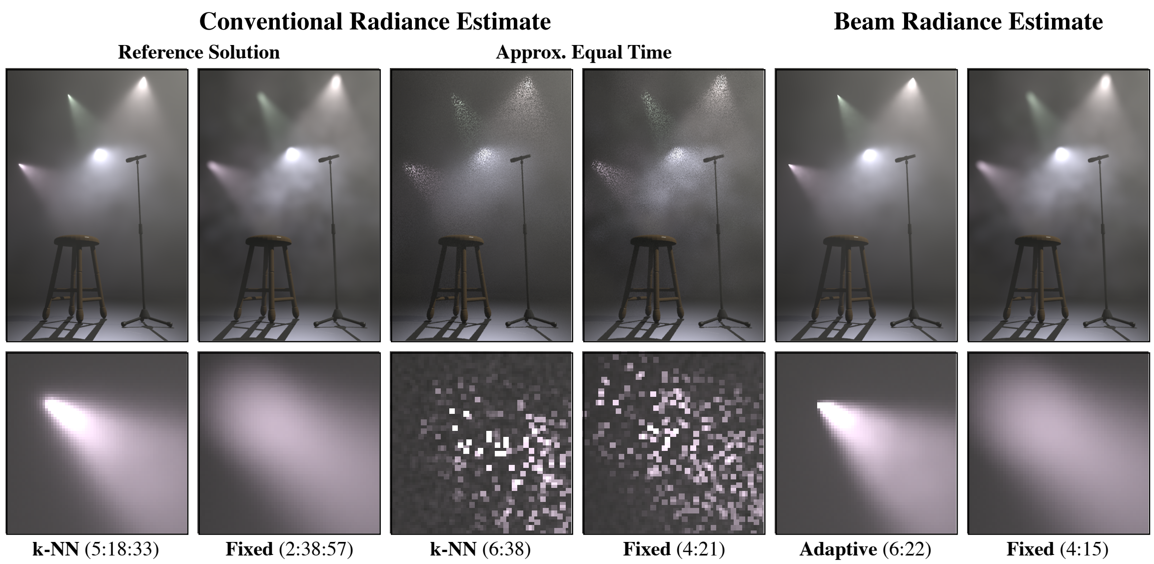

Figure 8.5: A comparison between the convention radiance estimate and our beam radiance estimate on the Stage scene with render times provided as (hours:minutes:seconds). Our method (right) produces images with much less noise than an equal time rendering using conventional volumetric photon mapping (middle) for both a fixed radius and an adaptive radius gathering approach. Our method does not require stepping but matches the quality of conventional photon mapping if a very small step size is used (left).

In our experimental setup, we first choose suitable gathering parameters (search radius and number of nearest neighbors k) and render the scenes using our method. We then use the same parameters using conventional photon mapping but tune the minimum step-size Δt to obtain approximately equal render times. Note that Δt is the *minimum* step-size and that exponential stepping is used to sample the ray according to transmission. Finally, we render a high-quality result with conventional photon mapping using a very small step size as a "reference."

We show visual comparisons of the methods in Figures 8.5 and 8.6. All images of each scene are rendered using the *same photon map*. The only differences in quality and performance are due to the gathering method used. We used the Henyey-Greenstein phase function for all

Table 8.2: Medium scattering properties and photon tracing statistics for the four example scenes. N is the total number of photons stored.

	Medium Parameters				**Photon Tracing**		
Scene	σ_s	σ_a	g	N	Shoot (s)	Balance (s)	Radius (s)
Cornell	0.225	0.225	0.00	0.4M	1.50	0.30	2.0
Stage	0.225	0.225	0.75	1M	3.25	0.76	5.0
Cars	0.06	0.015	0.00	2M	19.0	1.50	2.0
Lighthouse	0.24	0.010	0.75	1M	2.83	0.78	6.0

scenes. The render times, gathering parameters and timings for constructing the photon maps are listed in Table 8.1. We report the scattering parameters for all the example scenes in Table 8.2.

In all cases, our method produces significantly higher-quality images than conventional photon mapping. This is because querying once for all photons along a ray is more efficient than repeatedly querying for photons near numerous samples along the ray. Not surprisingly, we see that the reduced blurring of the adaptive kernel gathering methods is essential for scenes like the Stage and Lighthouse, where concentrated beams of light are visible. However, at the same render time, this advantage is difficult to discern in the conventional photon mapping images.

Though the k-NN and VK methods both adapt the width of the kernel to the local photon density, they are distinct approaches which result in similar, but not identical, density estimates. This is what accounts for the small differences in blurring between our adaptive results and the k-NN "reference" images. However, as our results show, the same value of k produces visually comparable renderings using the two methods.

8.5 Summary and Discussion

In this chapter, we showed how volumetric photon mapping can be expressed as a solution to the measurement equation. This formulation showed that *any* measurement of radiance within participating media can be estimated using the photon map. We applied this formulation by using the photon map to directly estimate accumulated in-scattered radiance along rays. This approach was implemented using an efficient beam gathering method, which can be used for both fixed and adaptive width kernels. The resulting algorithm produces images with significantly less noise than conventional volumetric photon mapping using the same render time.

8.6 Acknowledgements

This chapter is, in part, a reproduction of the material published in Wojciech Jarosz, Matthias Zwicker, and Henrik Wann Jensen. "The Beam Radiance Estimate for Volumetric Photon Mapping." In *Computer Graphics Forum (Proceedings of Eurographics 2008)*, 27(2):557–566, 2008; as well as the extended technical report Wojciech Jarosz, Matthias Zwicker, and Henrik

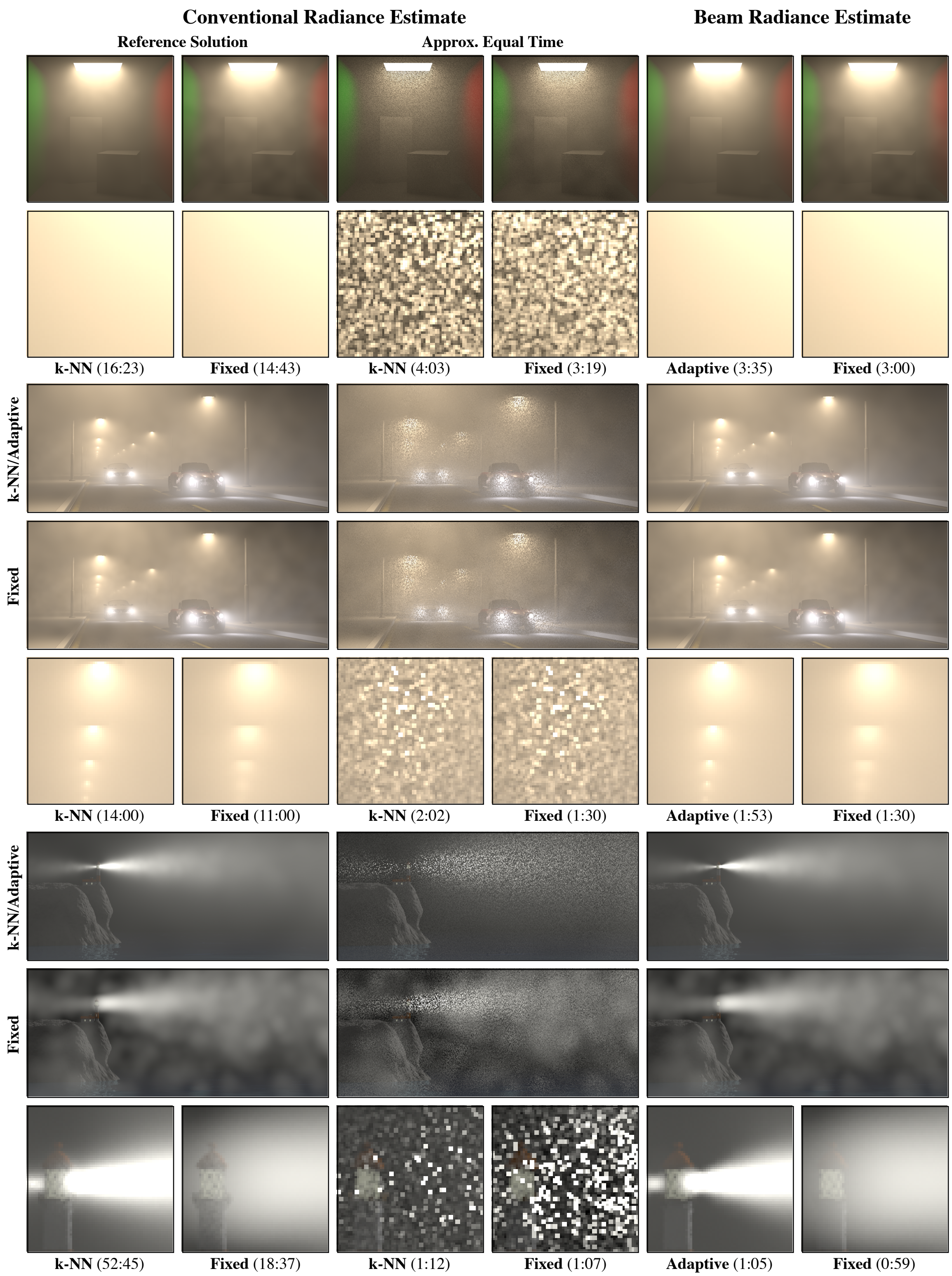

Figure 8.6: Visual comparison for the Cornell box, Cars, and Lighthouse scenes.

Wann Jensen. “The Beam Radiance Estimate for Volumetric Photon Mapping.” Technical Report CS2008-0914, University of California, San Diego, 2008. The dissertation author was the primary investigator and author of both papers. This work was supported in part by NSF grant CPA 0701992.