

Final Project: Your Topic

GROUP NUMBER: NAN
MEMBER 1: STUDENT 1
MEMBER 2: STUDENT 2
MEMBER 3: STUDENT 3

1 SECTION 1
2 SECTION 2

2.1 Residual Ratio Tracking

Ratio tracking and residual tracking, a way that is able to handle media with wavelength dependent extinction, can be combined to compute unbiased, low-variance transmittance estimates in general heterogeneous media. To be more specific, ratio tracking is to leverage the information discovered during the tracking more efficiently instead of deducing “just” a binary answer, finally provides a piecewise constant approximation to the transmittance function. Residual tracking, in the other hand, is complementary to delta tracking and ratio tracking and combines numerical estimation with an analytic approximation, yielding a piecewise exponential solution.

2.1.1 Ratio Tracking.

Definition. Ratio tracking is defined by following rules: 1. Replacing the Russian roulette by a weight that is equal to the probability of colliding with a fictitious particle. 2. Continuing the random walk until it reaches the end-point of the ray instead of probabilistically terminating the random walk at one of tentative collision points. 3. keeping track of the joint probability of colliding points with fictitious particles at all the preceding tentative collisions during the construction. By following the above rules, we can define the transmittance when reading d as:

$$T(d) = \prod_{i=1}^K \left(1 - \frac{\mu(x_i)}{\bar{\mu}}\right) \quad (1)$$

The multiplicand in the product represents the ratio of fictitious to all particles at collision point x_i .

Algorithm 1 is the pseudo-code of ratio tracking:

Unbiasedness of Ratio Tracking. The unbiasedness of Ratio Tracking can be demonstrated if it follows the following equation:

$$E[T] = \exp\left(-\int_0^d \mu(x)dx\right) = T(d) \quad (2)$$

Algorithm 1 Pseudocode of the ratio tracking estimator of transmittance along a ray with origin o , direction ω , and length d .

```

1: function RATIOTRACKINGESTIMATOR( $o, \omega, d$ )
2:    $t = 0$ 
3:    $T = 1$ 
4:   do:
5:      $\zeta = \text{rand}()$ 
6:      $t = t - \frac{\log(1-\zeta)}{\bar{\mu}}$ 
7:     if  $t \geq d$  : break
8:      $T = T * \left(1 - \frac{\mu(o+t*\omega)}{\bar{\mu}}\right)$ 
9:   while true
10:  return  $T$ 
11: end function

```

Now we try to prove that conclusion. Denoting τ_i realizations of T_i , the expected value of T can be written as:

$$E[T] = E\left[\sum_{i=0}^{\infty} T_i\right] = \sum_{i=0}^{\infty} E[T_i] = \sum_{i=0}^{\infty} \int_0^{\infty} \tau_i dP(\tau_i) \quad (3)$$

To list T_0 and T_1 as example, T_0 represents all realizations with the first tentative free flight distance S_1 already exceeding d . Since the value of T_0 equals to 1, the expected value reduces to the probability of S_1 taking on values greater than d :

$$E[T_0] = P(S_1 > d) = \int_d^{\infty} \tau_i dP(\tau_i) p_s(x) dx = \exp(-\bar{\mu}d) \quad (4)$$

In the case of T_1 , which represents events where $S_1 \leq d$ and $S_2 \geq -S_1$, then the expectancy of T_1 reads:

$$\begin{aligned}
E[T_1] &= \int_0^d \iota(x) P(S_2 > d - x) dP(S_1 \leq x) \\
&= \int_0^d \iota(x_1) P_s(x_1) \int_{d-x_1}^{\infty} p_s(x_2) dx_2 dx_1 \\
&= \int_0^d \iota(x_1) \bar{\mu} \exp(-\bar{\mu}x_1) \exp(-\bar{\mu}(d - x_1)) dx_1 \\
&= \bar{\mu} \exp(-\bar{\mu}d) \int_0^d \iota(x) dx
\end{aligned} \quad (5)$$

Then we can succinctly express the expected value of T_k , where represents realizations that satisfy $C_k \leq d$ and $C_k + 1 > d - C_k$, as:

$$\begin{aligned}
E[T_k] &= \int_0^d \iota(x_1) P_s(x_1) \int_0^{d-x_1} \iota(x_1 + x_2) P_s(x_2) \dots \\
&\quad \dots \int_0^{d-\sum_{j=1}^{k-1} x_j} \iota\left(\sum_{j=1}^k x_j\right) p_s(x_k) \\
&\quad \times \int_{d-\sum_{j=1}^{k-1} x_j}^\infty p_s(x_{k+1}) dx_{k+1} dx_k \dots dx_2 dx_1 \\
&= \int_0^d \iota(x_1) \int_0^{d-x_1} \iota(x_1 + x_2) \dots \int_0^{d-\sum_{j=1}^{k-1} x_j} \iota\left(\sum_{j=1}^k x_j\right) \\
&\quad \times \int_{d-\sum_{j=1}^{k-1} x_j}^\infty \bar{\mu}^k \exp(-\bar{\mu}(x_1 + \dots x_k + (d - \sum_{j=1}^k x_j))) \\
&\quad dx_{k+1} dx_k \dots dx_2 dx_1 \\
&= \bar{\mu}^k \exp(-\bar{\mu}d) \int_0^d \iota(x_1) \int_0^{d-x_1} \iota(x_1 + x_2) \dots \\
&\quad \dots \int_0^{d-\sum_{j=1}^{k-1} x_j} \iota\left(\sum_{j=1}^k x_j\right) dx_k \dots dx_2 dx_1
\end{aligned} \tag{6}$$

Then we replace z_i for $\sum_{j=0}^i x_j$ to rewrite the equation as:

$$\begin{aligned}
E[T_k] &= \bar{\mu}^k \exp(-\bar{\mu}d) \int_0^d \iota(z_1) \int_0^{d-z_1} \iota(z_2) \dots \\
&\quad \dots \int_0^{d-z_{k-1}} \iota(z_k) dz_k \dots dz_2 dz_1
\end{aligned} \tag{7}$$

The multiple integrals integrate $\prod_{j=1}^k \iota(z_j)$ over a k-dimensional simplex, which can be written concisely as:

$$E[T_k] = \bar{\mu}^k \exp(-\bar{\mu}d) \frac{(\int_0^d \iota(x) dx)^k}{k!} \tag{8}$$

Finally, we express the expected value of T as following:]

$$\begin{aligned}
E[T] &= \sum_{k=0}^\infty E[T_k] \\
&= \sum_{k=0}^\infty \bar{\mu}^k \exp(-\bar{\mu}d) \frac{(\int_0^d \iota(x) dx)^k}{k!} \\
&= \exp(-\bar{\mu}d) \sum_{k=0}^\infty \frac{(\bar{\mu} \int_0^d \iota(x) dx)^k}{k!} \\
&= \exp(-\bar{\mu}d) \exp(\bar{\mu} \int_0^d 1 - \frac{\mu(x)}{\bar{\mu}} dx) \\
&= \exp(-\int_0^d \bar{\mu} dx) \exp(\int_0^d \bar{\mu} - \mu(x) dx) \\
&= \exp(-\int_0^d \mu(x) dx)
\end{aligned} \tag{9}$$

Though ratio tracking is unbiased, its weakness is also obvious: the expected number of steps required to reach the endpoint of the

ray is high, which directly depends on the value of the majorant extinction coefficient. That's why we introduce residual tracking.

2.1.2 Residual Tracking.

Definition. Before introducing the definition of residual tracking, we first analyze the transmission function, where $\mu_c(x)$ is the extinction coefficient which is a simplified version of the original $\mu(x)$ that enables expressing the optical thickness up to distance d in closed form $\tau_c(d)$:]

$$\begin{aligned}
T(d) &= \exp(-\int_0^d \mu(x) dx) \\
&= \exp(-\int_0^d \mu_c(x) + \mu(x) - \mu_c(x) dx) \\
&= \exp(-\tau_c(d)) \exp(-\int_0^d \mu(x) - \mu_c(x) dx)
\end{aligned} \tag{10}$$

The first exponential term, referred to as the control transmittance $T_c(d)$, represents a coarse approximation of $T(d)$, which is computed using the simplified extinction coefficient μ_c . The second exponential then serves as a correction that accounts for the difference between the control and the actual transmittance. We denote this exponential as the residual transmittance T_r and refer to the integrand as the residual extinction coefficient $\mu_r(x) = \mu(x) - \mu_c(x)$. There is also one thing worth noticing-- in certain situations-- depending on the value of $\mu_c(x)$ --the residual extinction may be negative. Algorithm 2 is the pseudo-code of residual ratio tracking:

Algorithm 2 Pseudocode of the residual ratio tracking estimator for sampling transmittance along a ray with origin o , direction ω , and length d .

```

1: function RESIDUALRATIOTRACKINGESTIMATOR( $o, \omega, d$ )
2:    $t = 0$ 
3:    $T_c = \exp(-\mu_c * d)$ 
4:    $T_r = 1$ 
5:   do:
6:      $\zeta = \text{rand}()$ 
7:      $t = t - \frac{\log(1-\zeta)}{\bar{\mu}_r}$ 
8:     if  $t \geq d$  : break
9:      $T_r = T_r * (1 - \frac{\mu(o+t*\omega) - \mu_c}{\bar{\mu}_r})$ 
10:  while true
11:  return  $T_c * T_r$ 
12: end function

```

Expression. According to the equation mentioned before, we can obtain the residual ratio tracking estimator as below:]

$$\langle T_r(d) \rangle_{RR} = \prod_{i=1}^K (1 - \frac{\mu_r(x_i)}{\bar{\mu}_r}) = \prod_{i=1}^K (1 - \frac{\mu(x_i) - \mu_c}{\bar{\mu}_r}) \tag{11}$$

As our goal is to minimize the tracking cost by prolonging the tentative free-flight distances, it yields to finding the smallest $\bar{\mu}_r$ that still avoids negative multiplicands, which would prevent convergence of the above estimator. To ensure that $\mu_r(x)/\bar{\mu}_r \leq 1$, we set

$\bar{\mu}_r = \max(|\mu_r(x)|; 0 \leq x \leq d)$, i.e. the maximum absolute difference between $\mu(x)$ and μ_c along the ray.

For different values of μ_c , the visualization of them is totally different, shown in the following figure, where the bright regions correspond to cases when the residual transmittance takes on values higher than 1 to correct for the overly low control transmittance.

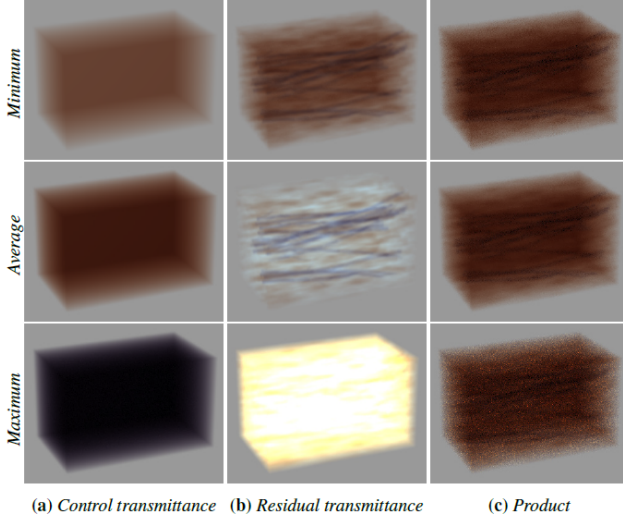


Fig. 1. Residual ratio tracking in a simple procedural volume with the control transmittance (a) computed using the minimum (top), the average (middle), and the maximum (bottom) extinction coefficient in the volume. The residual transmittance (b) used to compute the product (c) was estimated using 4 trackings only to emphasize the noise typical for each control extinction

- if $\mu_c = \mu_{min} = \min(\mu(x)); 0 \leq x \leq d$, i.e. the control extinction is underestimating, the control transmittance systematically overestimates the real transmittance
- if $\mu_c = \mu_{avg} = \text{avg}(\mu(x)); 0 \leq x \leq d$, the control extinction along the ray matches the real extinction on average. The residual tracking thus corrects only for the local over- and underestimation of the control transmittance w.r.t. the real transmittance along the ray.
- if $\mu_c = \mu_{max} = \max(\mu(x)); 0 \leq x \leq d$, the control transmittance systematically underestimates the real transmittance and the residual tracking thus needs to produce values greater than 1 to scale the control up.

Unbiasedness of Residual Ratio Tracking. The unbiasedness of Residual Ratio Tracking can be demonstrated if it follows the following equation:

$$E[T] = \exp\left(-\int_0^d \mu(x) - \mu_c dx\right) = \frac{T(d)}{\exp(-\mu_c d)} \quad (12)$$

Now we try to prove that conclusion. As the proof is very similar, we can directly mimic the above proof:

$$\begin{aligned} E[T] &= \exp(-\bar{\mu}_r d) \sum_{k=0}^{\infty} \frac{(\bar{\mu}_r \int_0^d \mu(x) dx)^k}{k!} \\ &= \exp(-\bar{\mu}_r d) \exp(\bar{\mu}_r \int_0^d 1 - \frac{\mu(x) - \mu_c}{\bar{\mu}_r} dx) \\ &= \exp\left(-\int_0^d \bar{\mu}_r \exp\left(\int_0^d \bar{\mu}_r - (\mu(x) - \mu_c) dx\right)\right) \\ &= \exp\left(-\int_0^d \mu dx(x) - \mu_c dx\right) \end{aligned} \quad (13)$$

2.2 Adaptive and unbiased sampling using kd-tree based space partitioning

As it's a time-consuming process to find a new scattering event in inhomogeneous participating media, we use an adaptive and unbiased sampling technique using kd-tree based space partitioning. During preprocessing, it partitions the analytical space (the bounding box of the medium) into sub-spaces (partitions, presented in kd-tree) according to the spatial variation of the mean free path in the medium. During rendering, the locations of the scattering events are determined adaptively using the kd-tree.

Here is the pseudo-code of WoodcockTracking and kdTreeFreePathSampling:

Algorithm 3 Pseudocode of WoodcockTracking Algorithm

Require: $x_0, \vec{\omega}$: The ray starting at x_0 in direction $\vec{\omega}$
 1: k_M : The majorant extinction coefficient
 2: (d_{min}, d_{max}) : The interval of the ray to evaluate
Ensure: The free path d to the next scattering event
 3: $d \leftarrow d_{min} - \ln(1 - \text{rand}()) / k_M$
 4: **while** $d \leq d_{max} \wedge k(x_0 + d\vec{\omega}) / k_M < \text{rand}()$ **do**
 5: $d \leftarrow d - \ln(1 - \text{rand}()) / k_M$
 6: **end while**
 7: **return** d

2.2.1 Unbiased Sampling and Space Partitioning.

Strategy. Suppose the ray starting at x_0 in direction $\vec{\omega}$ passes through two adjacent partitions i and j , with $k_M^{(i)}$ and $k_M^{(j)}$ being their majorant extinction coefficients. Let the distances from x_0 to the intersections with those partitions be s , q , and t . First, to sample the free path in the interval $(s, t]$, we first deal with the $(s, q]$ by using WoodcockTracking Algorithm, and if the free path exceeds q , we rewind the free path back to q and proceed to interval $(q, t]$ and use that algorithm again. By rewinding the free path back to q , we can ensure the unbiasedness of the sampling.

Proof of Unbiasedness. To prove unbiasedness, we need to show that we can obtain the free path d with probability:

$$\begin{aligned} P(x' = x_0 + d\vec{\omega} \wedge s < d \leq t) &= e^{-\tau(x_0 + s\vec{\omega}, x')} k(x') \\ P(d > t) &= e^{-\tau(x_0 + s\vec{\omega}, x_0 + t\vec{\omega})} \end{aligned} \quad (14)$$

1:4 • Group Number: NAN
 Member 1: Student 1
 Member 2: Student 2
 Member 3: Student 3

Algorithm 4 Pseudocode of kdTreeFreePathSampling Algorithm

Require: $x_0, \vec{\omega}$: The ray starting at x_0 in direction $\vec{\omega}$
Ensure: The free path d to the next scattering event

```

1: while true do
2:    $kdTreeNode\ p \leftarrow findNextLeafNode()$ 
3:   if  $p = nil$  then
4:     return INFINITY
5:   end if
6:    $(d_{min}, d_{max}) \leftarrow intersectionBetweenRayAndNode(p)$ 
7:    $k_M \leftarrow majorantExtinctionCoefficientStoredIn(p)$ 
8:    $d \leftarrow WoodcockTracking(x_0, \vec{\omega}, k_M, d_{min}, d_{max})$ 
9:    $d_{isect} \leftarrow INFINITY$ 
10:  if  $intersectionWithObjectSurfaces(x_0, \vec{\omega}, p)$  then
11:     $d_{isect} \leftarrow distanceToTheIntersection()$ 
12:  end if
13:  if  $d \geq d_{isect}$  then
14:    return IntersectionEvent
15:  end if
16:  if  $d < d_{max}$  then
17:    return  $d$ 
18:  end if
19: end while

```

than k_M for the entire space. These observations would give us a stopping criterion for the partitioning process, as well as a decision criterion for the partitioning location q .

First, we obtain d_1 during the sampling for the first interval $(s, q]$ with the probability:

$$\begin{aligned}
 P(x' = x_0 + d_1 \vec{\omega} \wedge s < d_1 \leq q) &= e^{-\tau(x_0 + s \vec{\omega}, x')} k(x') \\
 P_1(d_1 > q) &= e^{-\tau(x_0 + s \vec{\omega}, x_0 + q \vec{\omega})}
 \end{aligned} \tag{15}$$

Similarly, we obtain d_2 during the sampling for the second interval $(q, t]$

$$\begin{aligned}
 P(x' = x_0 + d_2 \vec{\omega} \wedge q < d_2 \leq t) &= e^{-\tau(x_0 + q \vec{\omega}, x')} k(x') \\
 P_1(d_2 > t) &= e^{-\tau(x_0 + q \vec{\omega}, x_0 + t \vec{\omega})}
 \end{aligned} \tag{16}$$

Since the sampling for the second interval is executed if and only if d_1 exceeds q during the sampling for the first interval, we obtain

$$\begin{aligned}
 P(x' = x_0 + d \vec{\omega} \wedge s < d \leq q) &= P_1(x' = x_0 + d \vec{\omega} \wedge s < d \leq q) \\
 &= e^{-\tau(x_0 + s \vec{\omega}, x')} k(x') \\
 P(x' = x_0 + d \vec{\omega} \wedge q < d \leq t) &= P_1(d_1 > q) P_2(x' = x_0 + d \vec{\omega} \wedge q < d \leq t) \\
 &= e^{-\tau(x_0 + s \vec{\omega}, x_0 + q \vec{\omega})} e^{-\tau(x_0 + q \vec{\omega}, x')} k(x') \\
 &= e^{-\tau(x_0 + s \vec{\omega}, x')} k(x') \\
 P(d > t) &= P_1(d > q) P_2(d > t) = e^{-\tau(x_0 + s \vec{\omega}, x_0 + q \vec{\omega})} e^{-\tau(x_0 + q \vec{\omega}, x_0 + t \vec{\omega})} \\
 &= e^{-\tau(x_0 + s \vec{\omega}, x_0 + t \vec{\omega})}
 \end{aligned} \tag{17}$$

2.2.2 Unbiased Sampling and Space Partitioning. The performance of the sampling technique shown in previous section highly depends on the partitioning. If we partition a nearly homogeneous interval, an additional iteration is required to rewind the free path to q when proceeding to the adjacent interval, and this would be a waste compared to the case with no partitioning. On the other hand, we can have more benefit, if $k_M^{(i)}$ in a partition is much smaller