Tech-Paper: Volume Rendering Integral

Sebastian Weiß

December 2020

1 The Volume Rendering Integral

In our work, we assume a low-albedo emission plus absorption model for volume rendering [Max95]. Let $\tau : [0,1] \rightarrow \mathbb{R}_0^+$ be the absorption due to a given density, and $C : [0,1] \rightarrow \mathbb{R}_0^+$ the assigned self-emission, both specified via a transfer function. Then, the transparency of the line segment from $t = a$ to b is written as

$$T(a,b) = \exp\left(-\int_a^b \tau(V(s(t)))dt \right). \tag{1}$$

The transparency is 1 if the medium between a and b does not absorb any light and approaches zero for complete absorption. Then, the light intensity reaching the eye is

$$L(a,b) = \int_a^b g(V(s(t)))T(a,t)dt, \tag{2}$$

were $g(v) = \tau(v)C(v)$. Usually, the emission is not given as a single scalar intensity, but as an RGB tuple. In this case, Equation 2 becomes a vector quantity.

2 Splitting the volume integral at ray segment boundaries

The general volume integrals, Equation 1 and Equation 2, are impossible to evaluate directly for arbitrary functions τ and L_e. Therefore, these integrals are subdivided into smaller parts and integrated for each part separately. How to derive the equations for the parts and how to build them together are summarized here. The notation here is based on Max [Max95] and de Boer *et al.* [dBGHM97] and they serve as a summary and remainder on where the equations originate.

2.1 Absorption

First, we have a look at the transparency $T(a,b)$ based on the absorption coefficient τ. Given a subdivision of $[a,b]$ into N parts,

$$a = t_0 < t_1 < ... < t_N = b,$$

the integral in Equation 1 can be split into smaller parts as following:

$$\begin{aligned}
T(a,b) &= \exp\left(-\int_a^b \tau(s(t))dt\right) \\
&= \exp\left(-\sum_{i=0}^{N-1}\int_{t_i}^{t_{i+1}} \tau(s(t))dt\right) \\
&= \prod_{i=0}^{N-1} \underbrace{\exp\left(-\int_{t_i}^{t_{i+1}} \tau(s(t))dt\right)}_{=T(t_i,t_{i+1})=:T_i}.
\end{aligned} \tag{3}$$

Hence, the final transparency is given by $T = \prod_{i=0}^{N-1} T_i$ where 1 indicates fully transparent, 0 fully opaque. Alternatively, one is usually more interested in the opacity α with 1 being fully opaque and 0 being fully transparent. The relationship between α and T follows

$$\alpha := 1 - T \Leftrightarrow T = 1 - \alpha. \tag{4}$$

Substituting Equation 4 into Equation 3 gives rise to

$$\begin{aligned}
1 - \alpha = \prod_{i=0}^{N-1}(1-\alpha_i) \text{ with } \alpha_i = 1 - T_i \\
\Leftrightarrow \alpha = 1 - \prod_{i=0}^{N-1}(1-\alpha_i).
\end{aligned} \tag{5}$$

Let $\alpha^{(k)} := \prod_{i=0}^{k-1}(1-\alpha_i)$ be the evaluation of Equation 5 up to k terms. By induction we have:

$$\begin{aligned}
k = 0 &\ : \alpha^{(0)} = 0 \\
k \to k+1 &\ : \alpha^{(k+1)} = \alpha^{(k)} + (1-\alpha^{(k)})\alpha_k.
\end{aligned} \tag{6}$$

This leads to the well-known front-to-back algorithm:

Algorithm 1 Front-to-back algorithm for absorption

```
$\alpha = 0$
for $i = 0, ..., N-1$ do
    evaluate $\alpha_i = 1 - T_i$
    $\alpha = \alpha + (1-\alpha)\alpha_i$
    optional early-out if $\alpha$ gets close to 1
end for
```

The important aspect in this algorithm is to compute α_i. In the simplest form, the following approximation schemes, summarized by Max [Max95], are used:

$$\begin{aligned}\alpha_i &= 1 - \exp\left(-\int_{t_i}^{t_{i+1}} \tau(s(t))dt\right) && \text{approx. by left factor}\\ &\approx 1 - \exp(-\tau(s(t_i))(\underbrace{t_{i+1} - t_i}_{=:\Delta t_i})) && \text{Taylor}\\ &\approx 1 - \max(0,\, 1 - \Delta t_i\, \tau(s(t_i)))\\ &= \min(1,\, \Delta t_i\, \tau(s(t_i))) && .\end{aligned} \tag{7}$$

In special cases, e.g. hexahedral grids with tri-linear interpolation, analytical solutions for the transparency are possible.

2.2 Emission

Next, we analyze the emission term L, based on the emission coefficients L_e, again subdivided over intervals $t_0, ..., t_N$.

$$\begin{aligned}L(a,b) &= \int_a^b L_e(s(t))T(a,t)dt\\ &= \sum_{i=0}^{N-1} \int_{t_i}^{t_{i+1}} L_e(s(t))T(a,t)dt\\ &= \sum_{i=0}^{N-1} T(a,t_i) \int_{t_i}^{t_{i+1}} L_e(s(t))T(t_i,t)dt\\ &= \sum_{i=0}^{N-1} T(a,t_i) \underbrace{\int_{t_i}^{t_{i+1}} L_e(s(t))T(t_i,t)dt}_{=:L_i}\,.\end{aligned} \tag{8}$$

Hence the final emission is given by $L = \sum_{i=0}^{N-1} T(a,t_i)L_i$ where $T(a,t_i) = \prod_{j=0}^{i-1} T_j$, see Equation 3. Using the substitution $\alpha = 1 - T$, we arrive at

$$L = \sum_{i=0}^{N-1} (1 - \alpha^{(i)})L_i \tag{9}$$

which gives rise to the following extension of Algorithm 1, now incorporating emission:

Algorithm 2 Front-to-back algorithm for absorption and emission

```
$\alpha = 0, L = 0$
for $i = 0, ..., N-1$ do
    evaluate $\alpha_i = 1 - T_i$ and $L_i$
    $L = L + (1 - \alpha)L_i$
    $\alpha = \alpha + (1 - \alpha)\alpha_i$
    optional early-out if $\alpha$ gets close to 1
end for
```

Note that the emission here is presented as a scalar quantity, but the computations can be easily extended to vector quantities, e.g. RGB-colors or spectra.

Again, the crucial part of this algorithm is the computation of L_i, we will again present the simple approximation, commonly used in rendering here [Max95]:

$$\begin{aligned} L_i &= \int_{t_i}^{t_{i+1}} L_e(s(t))T(t_i,t)dt \qquad \text{approx. by left factor} \\ &\approx (L_e(s(t_i))\underbrace{T(t_i,t_i)}_{=1})(\underbrace{t_{i+1}-t_i}_{=:\Delta t_i}) \qquad (10) \\ &= L_e(s(t_i))\Delta t_i \end{aligned}$$

For more precise evaluation, quadrature schemes like Simpson's scheme have to be used.

References

[dBGHM97] Martijn W de Boer, Alexander Gröpl, Jürgen Hesser, and Reinhard Männer. Reducing artifacts in volume rendering by higher order integration. *IEEE Visualization'97 Late Breaking Hot Topics*, pages 1–4, 1997.

[Max95] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.