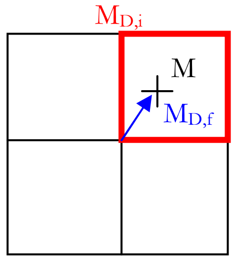
Appendix

We include here the details on how we remove the *frac* instruction from the tree lookup.

A complete tree would produce at depth D a 3D grid of resolution $N^D \times N^D \times N^D$. We call this grid the *depth D grid.* At depth D, the point M lies in a cell of this grid. The integer coordinates of this cell are $M_{D,i} = floor(M \cdot N^D)$. The local coordinate of M within this cell are $M_{D,f} = frac(M \cdot N^D)$.



It follows $M = \dfrac{M_{D,i} + M_{D,f}}{N^D}$

The lookup coordinates within the indirection pool are computed as

$$P = \frac{I_D + M_{D,f}}{S} = \frac{I_D + frac(M \cdot N^D)}{S}$$

Using the fact that $M = \dfrac{M_{D,i} + M_{D,f}}{N^D}$ we can rewrite P as $P = \dfrac{I_D + M \cdot N^D - M_{D,i}}{S}$

Note that $M_{D,i}$ is a constant within the node visited at depth D. It corresponds to the coordinates of the node within the grid of depth D. We call these coordinates G_D.

We rewrite G_D as $G_D = kS + Q$, where k is an integer and $Q < S$.

We now obtain $P = \dfrac{M \cdot N^D + I_D - Q - kS}{S} = \dfrac{M \cdot N^D + I_D - Q}{S} - k$ (1)

We define $\Delta_D = I_D - Q$

If we bind the indirection pool texture in repeat mode (GL_REPEAT), we can add any integer to P without changing the result. Therefore the term $-k$ in equation (1) can be ignored.

Finally, we have $P = \dfrac{M \cdot N^D + \Delta_D}{S}$ (2)

Instead of directly storing the node indices I_D we actually store Δ_D and use equation (2). This removes the *frac* operation. However, storing Δ_D could be a problem if it can take arbitrary large integer values. Fortunately, since $I_D < S$ and $Q < S$, it comes $-S < \Delta_D < S$. Moreover, if Δ_D is less than 0, we can use $S + \Delta_D$ instead without changing the result of the lookup (once again thanks to the repeat mode). Therefore we only have to store values in the range $[0, S[$.