

Practical Path Guiding for Efficient Light-Transport Simulation

Thomas Müller^{1,2} Markus Gross^{1,2} Jan Novák²

¹ETH Zürich
²Disney Research

1. Pseudocode

In this supplementary material, we provide pseudocode for the importance sampling and construction procedures for SD-trees.

Algorithm 1 Sample from the quadtree

```
1: procedure SAMPLEQUADTREE(node)
2:   if ISLEAF(node) then
3:     return UNIFORMRANDOMPOSITIONIN(node)
4:   else
5:     child ← SAMPLECHILDBYENERGY
6:     return SAMPLEQUADTREE(child)
7:   end if
8: end procedure
```

Algorithm 2 Compute PDF of sampling $\vec{\omega}$ from a quadtree

```
1: procedure PDFQUADTREE(node,  $\vec{\omega}$ )
2:   if ISLEAF(node) then
3:     return 1/4 $\pi$ 
4:   else
5:     child ← GETCHILD( $\vec{\omega}$ )
6:      $\alpha$  ← 4 · FLUX(child) / FLUX(node)
7:     return  $\alpha$  · PDFQUADTREE(child)
8:   end if
9: end procedure
```

Algorithm 3 Refine the binary tree post training iteration

```
1: procedure REFINEBINARYTREE( $k$ )
2:   for all leaf  $\in$  leafNodes do
3:     if NUMSAMPLES(leaf) >  $c \cdot \sqrt{2^k}$  then
4:       SUBDIVIDE(leaf)
5:     end if
6:   end for
7: end procedure
8:
9: procedure SUBDIVIDE(leaf)
10:  numSamples ← NUMSAMPLES(leaf)
11:  children ← SPLITALTERNATINGAXIS(leaf)
12:  SETNUMSAMPLES(children, numSamples / 2)
13: end procedure
```

Algorithm 4 Refine a quadtree post training iteration

```
1: procedure REFINEQUADTREE(quadtree)
2:    $\Phi$  ← FLUX(quadtree)
3:   for all node  $\in$  quadtree do
4:     if FLUX(node)/ $\Phi$   $\leq$   $\rho$  then
5:       PRUNECILDREN(node)
6:     else if ISLEAF(node) then
7:       SUBDIVIDE(node)
8:     end if
9:   end for
10: end procedure
11:
12: procedure SUBDIVIDE(leaf)
13:   $\Phi_n$  ← FLUX(leaf)
14:  children ← SPLITQUADTREE(leaf)
15:  SETFLUX(children,  $\Phi_n/4$ )
16: end procedure
```
