Guiding Light Trees for Many-Light Direct Illumination: Supplemental

Eric Hamann, Alisa Jung and Carsten Dachsbacher

Karlsruhe Institute of Technology, Department of Informatics, Germany

Algorithm 1: Overview

```
Build light BVH over Emitters
spatialTree = Initialize Spatial Tree
for cell in spatialTree do
    cell.QuadTree = Initialize Quadtree
    cell.GLT1 = Initialize GLT
    cell.GLT2 = Initialize GLT
end
while Rendering do
    iteration = 0
    for i = 0 ... 2^iteration do
        for all pixels do
            (x_0, ..., x_L) = Sample path
            cell = spatialTree.LocateLeaf(x_{L-1})
            node = cell.GLT1.LocateLeaf(x_L.LightSource)
            node.Sum += L_e(x_L → x_{L-1}) G(x_L, x_{L-1}) / p(x_L)
        end
    end
    Swap GLT1 and GLT2 for each cell
    Refine SD tree and GLTs
    Clear image
    iteration++
end
```

Algorithm 2: Initialize GLT

```
// Initialize a weighted tree cut through the light BVH
GLT.Root.Weight = 1
GLT.Root.LeftChild.Weight = 0.5
GLT.Root.RightChild.Weight = 0.5
GLT.Root.LeftChild.Sum = 0
GLT.Root.RightChild.Sum = 0
```

Algorithm 3: Guided Next Event Estimation

```
(x_0, ..., x_{L-1}) = Sample path prefix
rand = Random Number in [0, 1)
cell = spatialTree.Locate(x_{L-1})
cutNode = cell.GLT2.Root
while cutNode not a Leaf in the GLT do
    if rand < cutNode.LeftChild.Weight) then
        rand /= cutNode.LeftChild.Weight
        cutNode = cutNode.LeftChild
    else
        rand /= cutNode.RightChild.Weight
        cutNode = cutNode.RightChild
    end
end
node = Get light BVH node corresponding to cutNode
while node not a leaf in the Light BVH do
    if rand < node.LeftChild.Weight then
        rand /= node.LeftChild.Weight
        node = node.LeftChild
    else
        rand /= node.RightChild.Weight
        node = node.RightChild
    end
end
x_L = Sample vertex on node.LightSource
return (x_0, ..., x_L)
```