Progressive Visibility Caching for Fast Indirect Illumination

Justus Ulbrich, Jan Novák, Hauke Rehfeld and Carsten Dachsbacher
Karlsruhe Institute of Technology

In this supplementary material we provide a pseudocode for refining the visibility cache.

```plaintext
pre-compute
    shoot initial CRs

while improving
    target-CR = find CR with highest importance
    moved-CR = find CR with lowest importance

    // move moved-CR
    create candidates around target-CR
    select candidate that has furthest distance to all other CRs
    move moved-CR to candidate

    recompute visibility for moved-CR

    // update correlation for pairs
    update-list = []
    foreach neighbor of moved-CR as n
        update-list += (moved-CR, index(n))

    foreach CR moved-CR was neighbor of as c
        update-list += (c, index(moved-CR))

    // find new neighbors
    foreach update-list as (cr, i)
        cr[i] = find new neighbor
        parallel brute force search with w()

    foreach CR moved-CR was not neighbor of as c
        furthest-w = w(c, furthest-CR of c)
        if w(c, moved-CR) < furthest-w
            replace furthest-CR with moved-CR
            update-list += (c, index(moved-CR))

    // recompute correlation
    foreach update-list as (src-CR, index)
        dst-CR = src-CR[index]
        foreach sample direction as dir
            p0 = point seen in dir from src-CR
            dstDir = direction from dst-CR to p0
            p1 = point seen in dstDir from dst-CR
            dist = distance from p0 to p1
            maxDist = max{dist from src-CR to p0, dist from dst-CR to p1}

            t = PREDEFINED
            t /= sqrt(dot(normal at p0, dir))
            t /= sqrt(dot(normal at p1, dstDir))

            if dist / maxDist > t
                num-misses += 1

            correlation = 1 - (num-misses / num-samples)
```

Listing 1: Refinement Algorithm. Cache record is abbreviated as CR, w() is the weighting function from Section 3.1

© The Eurographics Association 2013.