A APPENDIX

A.1 Hausdorff Distance-Aware Progressive Polyhedron Simplification

Algorithm 1: Progressive Polyhedron Simplification

```
Input  : Polyhedron P
Output : Multi-LOD Representation P'
P' ← P;
for i^th facet f'_i in P' and f_i in P do
    f'_i.assoc_facets ← {f_i} ;
while Lowest LOD not reached do
    for each vertex v in P' do
        set v as unprocessed;
    while P' has unprocessed vertex v do
        V_c ← all vertices in P' connected to v;
        for each v' in V_c + v do
            set v' as processed;
        f' ← facet constructed with vertices in V_c;
        f'.interior ← v.is_protruding;
        f'.exterior ← v.is_recessing;
        for each facet f around vertex v do
            f'.interior ← f'.interior&f.interior;
            f'.exterior ← f'.exterior&f.exterior;
            for each facet f_a in f.assoc_facets do
                f'.assoc_facets ← f'.assoc_facets + f_a;
                f_a.assoc_facets ← f_a.assoc_facets + f';
            remove f from P';
        if f'.interior = false && f'.exterior = false then
            determine state of f' evaluating f'.assoc_facets;
        add f' to P';
        f'.removed_vertex ← v;
    for each facet f' in P' do
        for each point p in sample(f', ε) do
            d ← distance(p, f'.assoc_facets) + ε/√2;
            f'.hdist ← max(f'.hdist, d) ;
            P'.hdist ← max(P'.hdist, d);
    for each facet f in P do
        for each point p in sample(f, ε) do
            f' ← closest facet in f.assoc_facets to p ;
            d ← distance(p, f') + ε/√2;
            f'.phdist ← max(f'.phdist, d);
            P'.phdist ← max(P'.phdist, d);
    for each facet f' in P' do
        if f'.removed_vertex ≠ null then
            encoding f'.removed_vertex;
        encoding f'.interior?0 : ⌈(f'.hdist*255)/(P'.hdist)⌉;
        encoding f'.exterior?0 : ⌈(f'.phdist*255)/(P'.phdist)⌉;
encoding the remaining primitives on P' as LOD 0;
```

A.2 Performance Analysis of Computing with Facet-Association and BVH-tree

Consider the calculation of Hausdorff distances and proxy Hausdorff distances for the facets of a low-LOD polyhedron P' in relation to the original polyhedron P. During the Hausdorff distance calculation, we construct a BVH tree comprising $|P|$ facets, and we sample $SR \times |P'|$ points from the surface of P'. Each BVH tree lookup operation consumes time equivalent to $t \times \log |P|$, with t representing a constant coefficient associated with tree lookups in a given execution environment. Consequently, the time required for calculating the Hausdorff distance is proportional to $SR \times |P'| \times t \times \log |P|$. Similarly, the time spent on computing the proxy Hausdorff distance is proportional to $SR \times |P| \times t \times \log |P'|$. Importantly, it is evident that $|P'| < |P|$, resulting in the inequality $SR \times |P'| \times t \times \log |P| < SR \times |P| \times t \times \log |P'|$. Therefore, calculating the proxy Hausdorff distance demands more time than computing the Hausdorff distance. Additionally, as more rounds of simplification are performed, the value of $|P'|$ decreases, resulting in reduced computation time for low-LOD polyhedrons.

With the FA method, each sampled point on facet f merely needs to compute its distance from the facets associated with f. The time required for calculating the Hausdorff distance and proxy Hausdorff distance can be estimated as $SR \times |P'| \times t \times r' \times |P|$ and $SR \times |P| \times t \times r \times |P'|$, respectively. Here, r' represents the average portion of facets in P associated with one facet in P', and r represents the average portion of facets in P' associated with one facet in P. Remarkably, as the facets in low-LOD polyhedrons and the original polyhedrons are associated region by region, it can be approximated that r is roughly equal to r'. Consequently, calculating the Hausdorff and proxy Hausdorff distances consumes approximately the same amount of time. Moreover, as the simplification process progresses, the values of both r and r' increase, while the value of $|P'|$ decreases. Consequently, the computation time does not exhibit a strictly decreasing trend with varying LODs.