


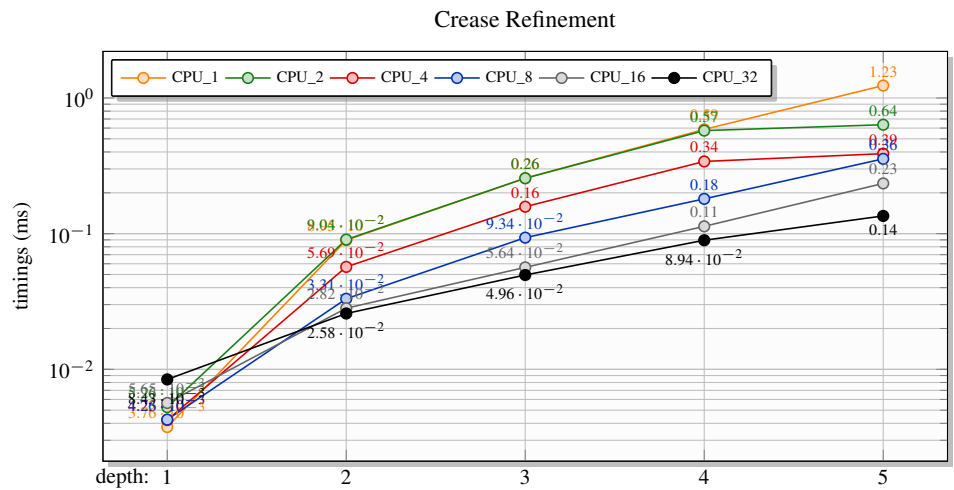
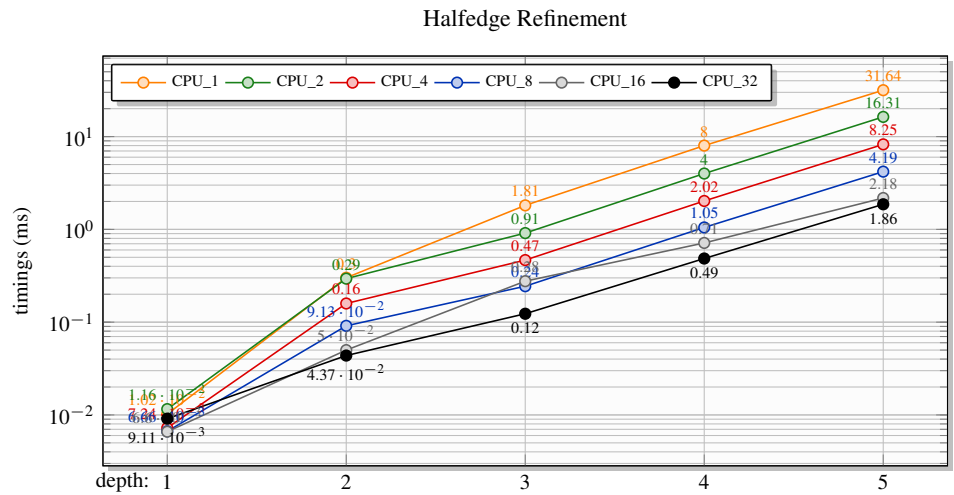
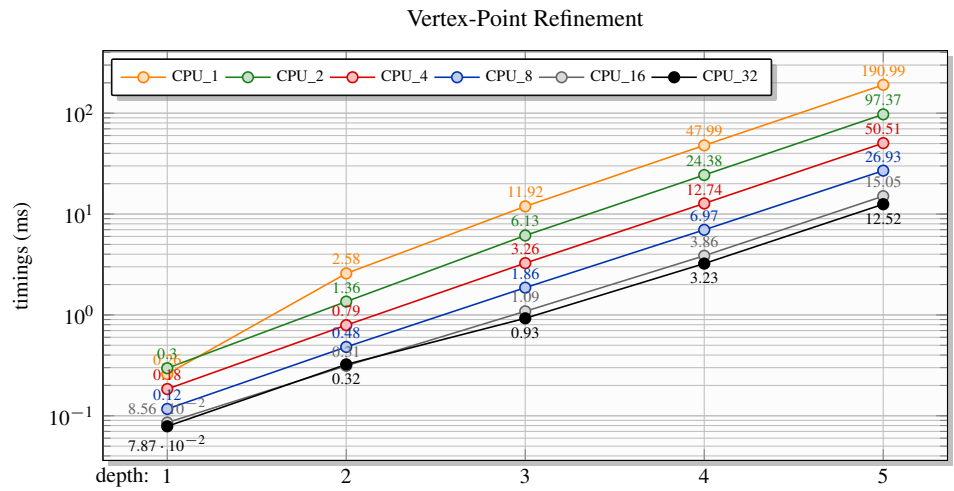
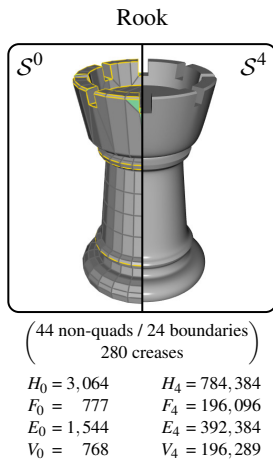
A Halfedge Refinement Rule for Parallel Catmull-Clark Subdivision Supplemental Material: CPU Performance Measurements

J. Dupuy and K. Vanhoey 

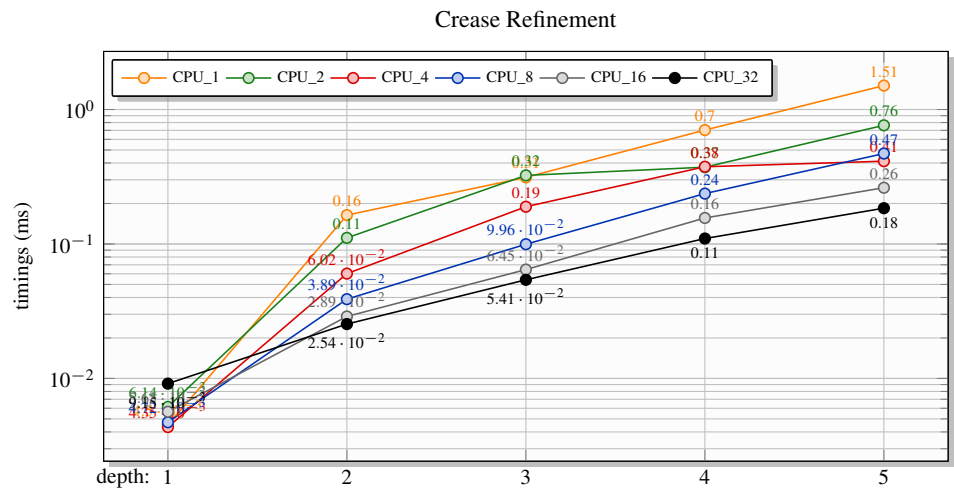
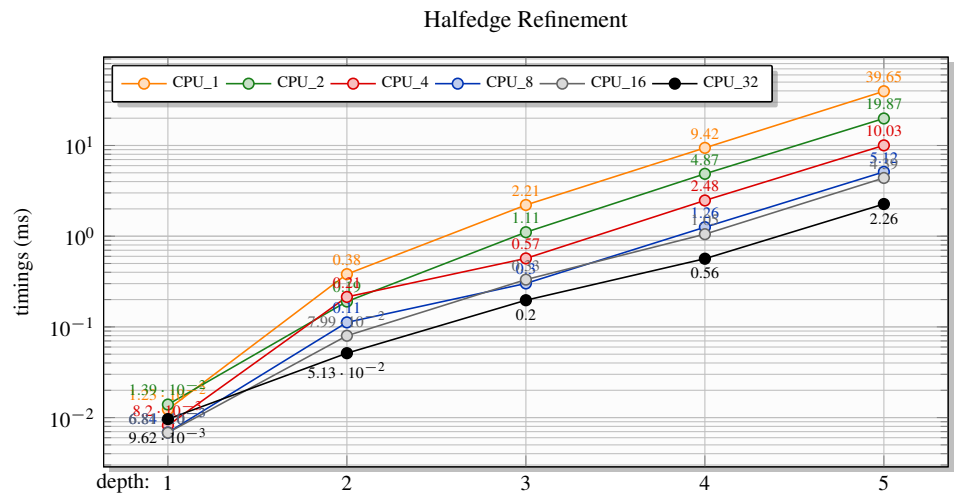
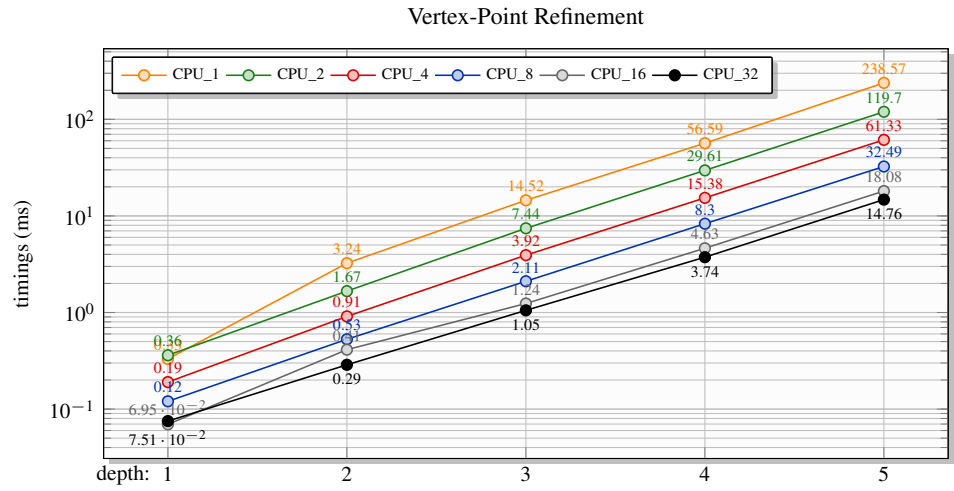
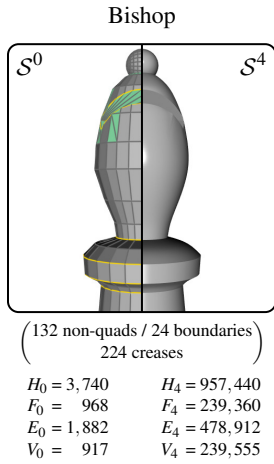
Unity Technologies

This document provides exhaustive performance measurements of our CPU implementation. The measurements were performed on an AMD Ryzen Threadripper 3960X CPU with 24-cores. We compiled our program with options `-march-native` and `-Os`. Our program relies on OpenMP to spawn threads, and we report performance measurements for threads counts 1, 2, 4, 8, 16, and 32. Each number corresponds to the median timing over a set of 50 runs. The goal of this document is to convey the key information that the performances of our method scale linearly with the number of threads from 1 to 16 threads. For 32 threads we observe less significant speed-ups, which is due to the fact that the CPU has 24 cores and obviously becomes less efficient at distributing tasks.

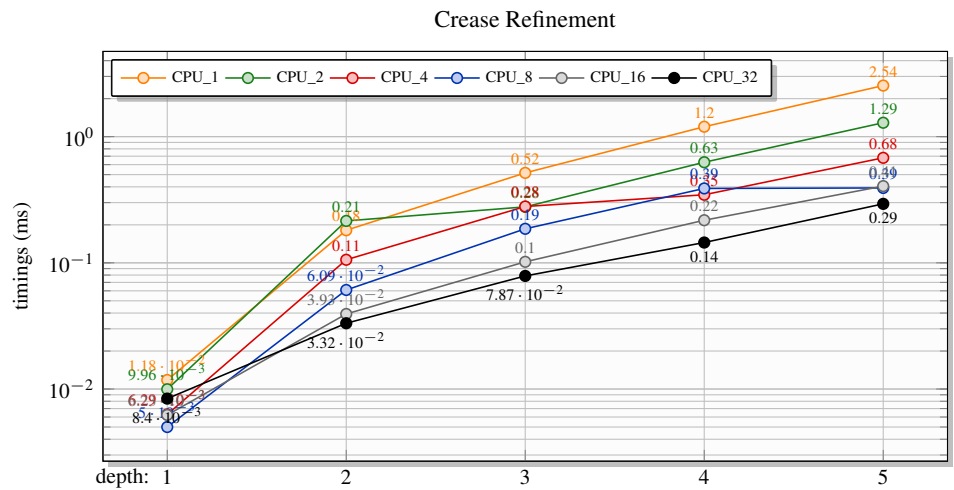
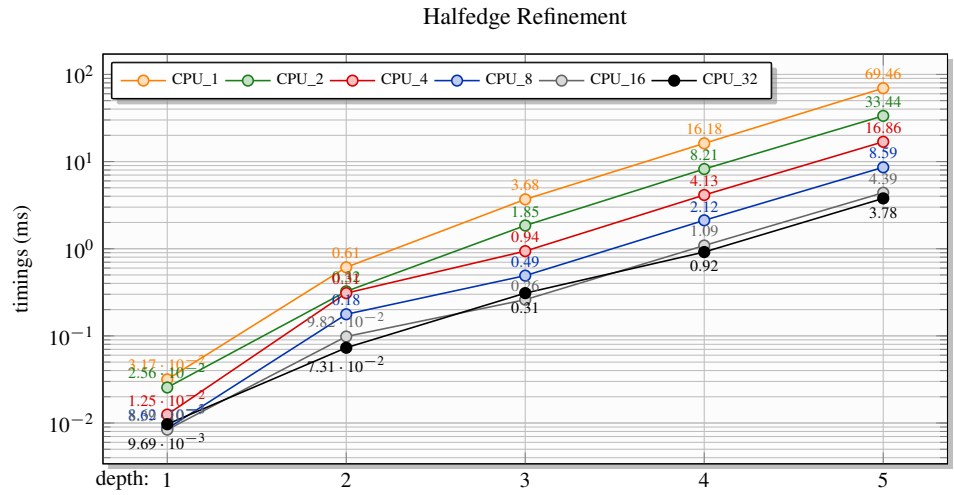
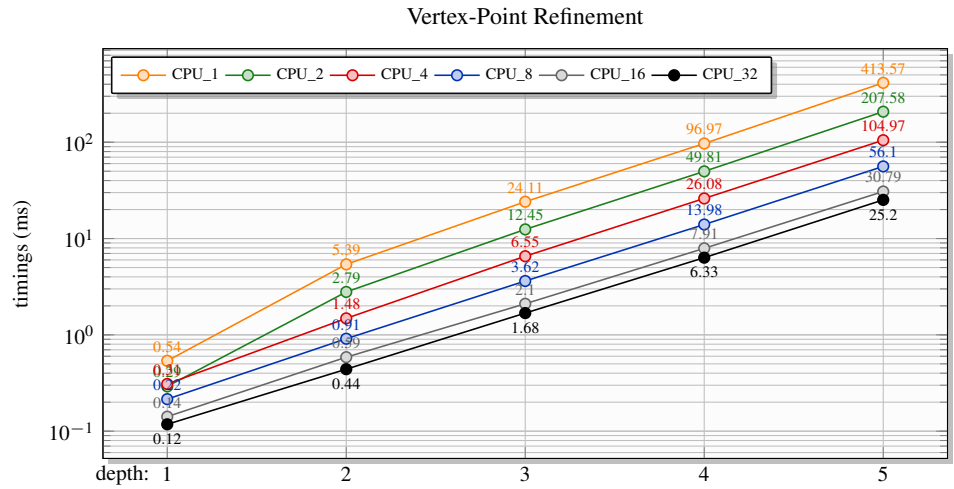
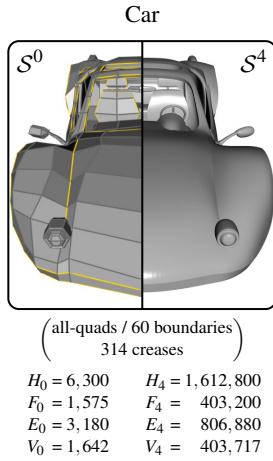
1. Rook



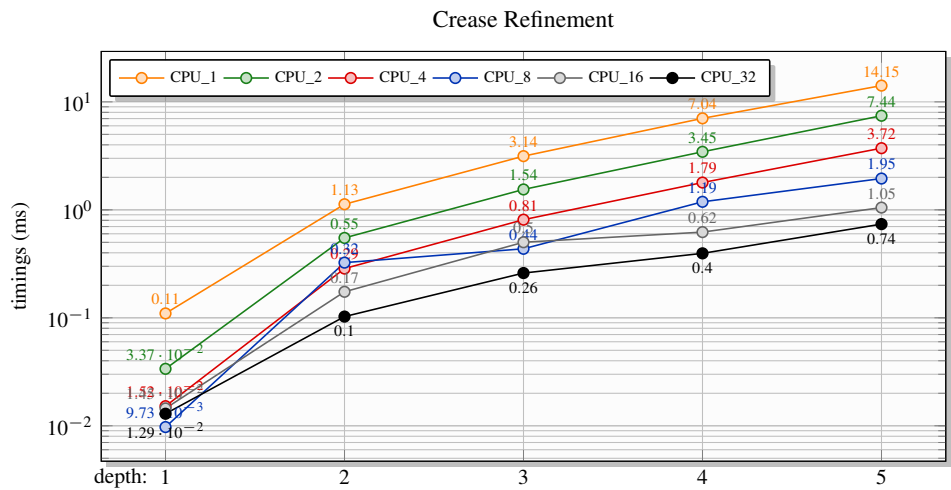
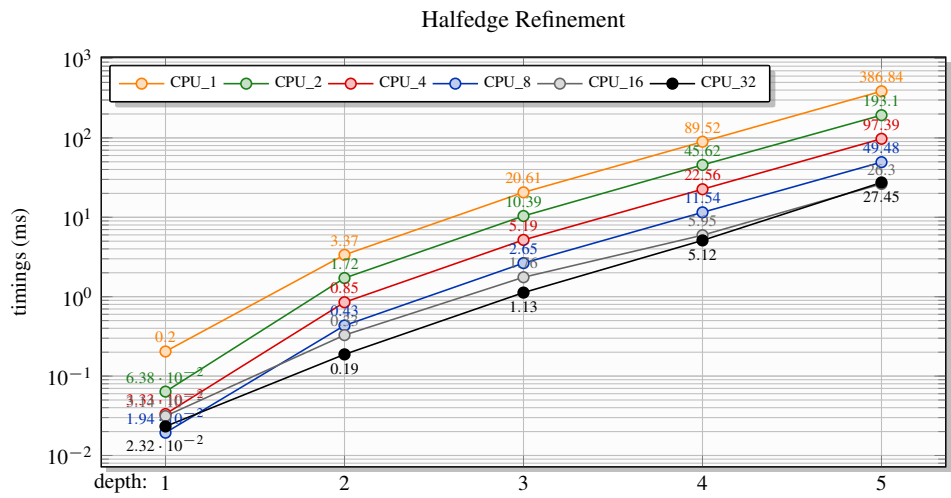
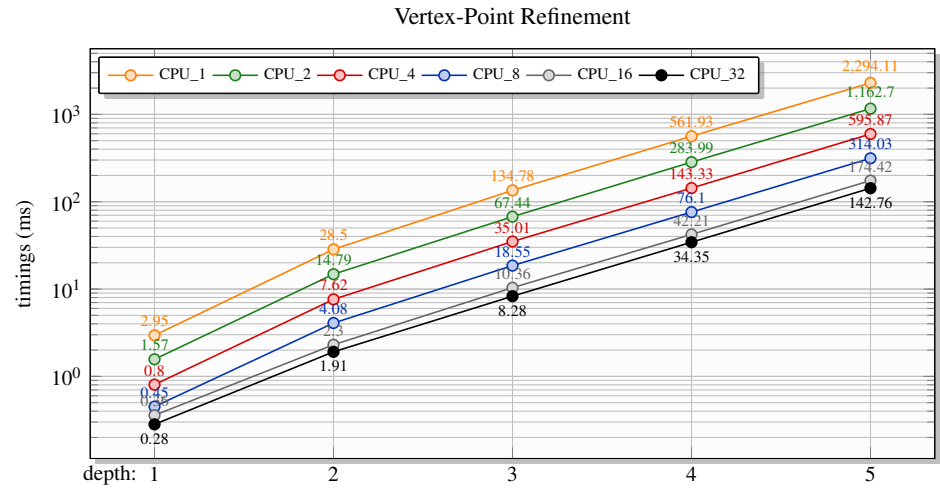
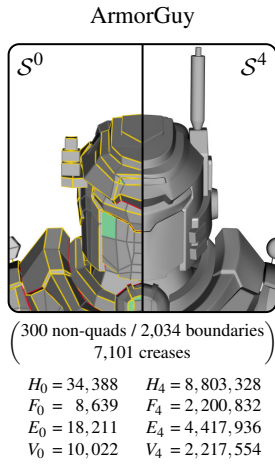
2. Bishop



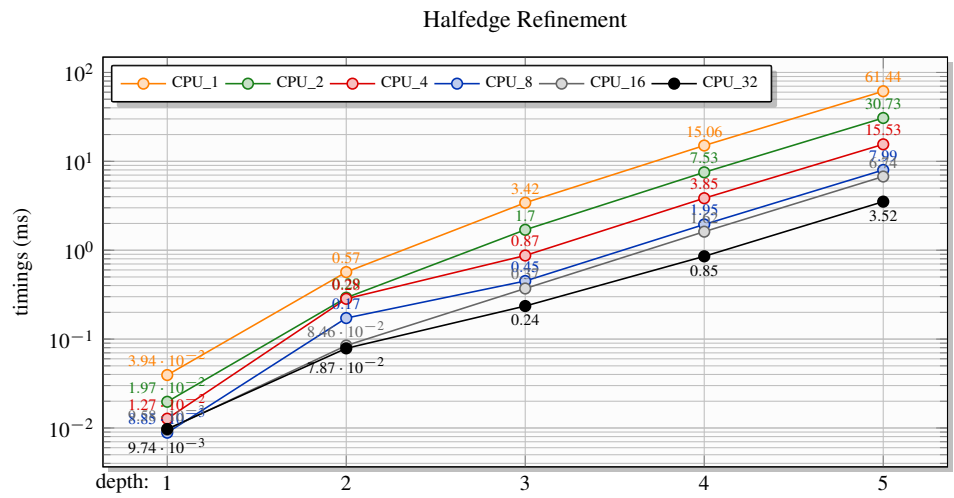
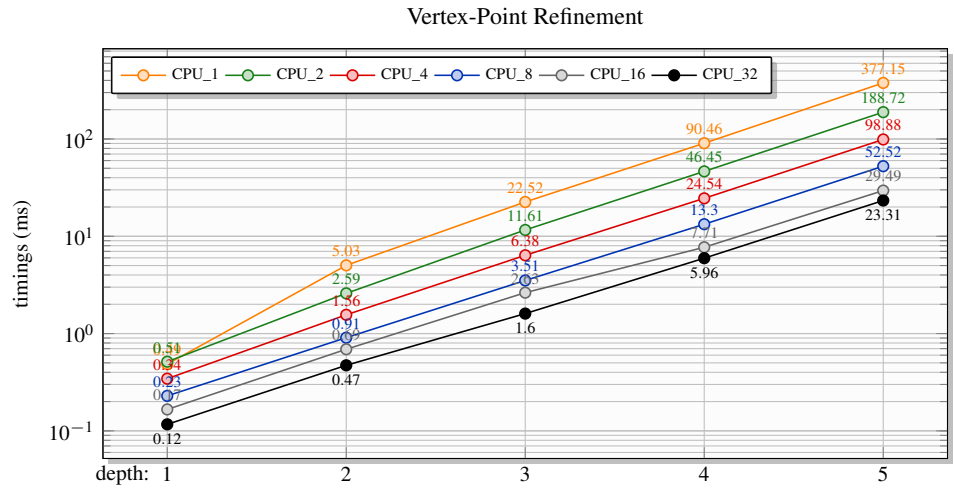
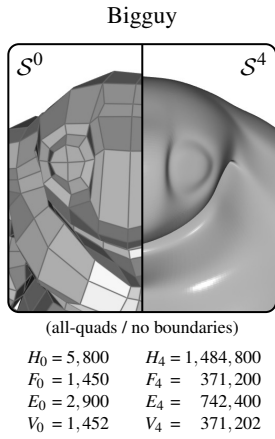
3. Car



4. ArmorGuy

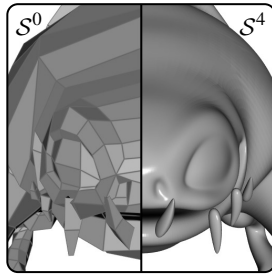


5. Bigguy



6. Monsterfrog

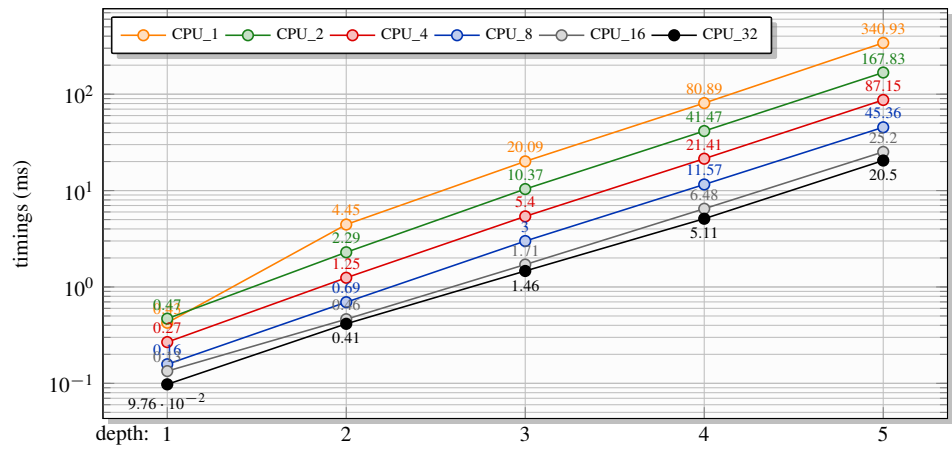
Monsterfrog



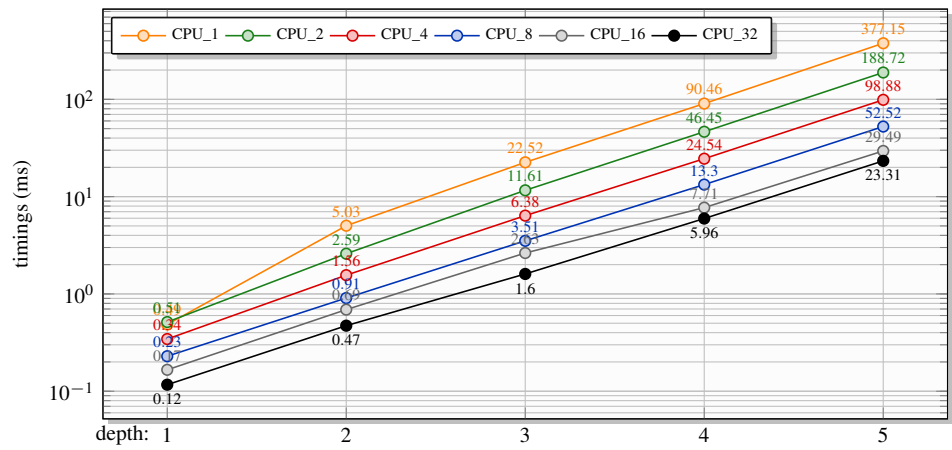
(all-quads / no boundaries)

$H_0 = 5,168$	$H_4 = 1,323,008$
$F_0 = 1,292$	$F_4 = 330,752$
$E_0 = 2,584$	$E_4 = 661,504$
$V_0 = 1,308$	$V_4 = 330,768$

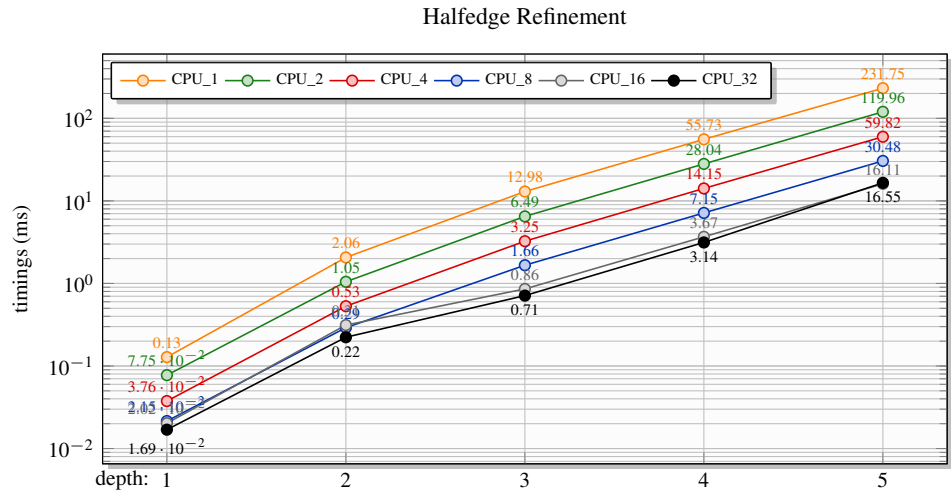
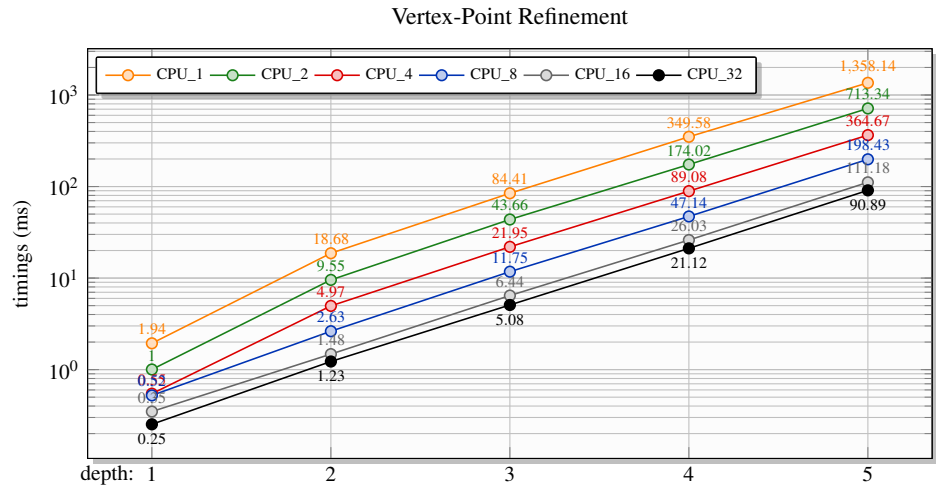
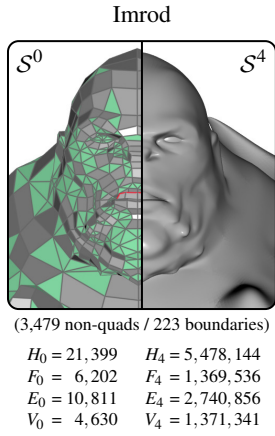
Vertex-Point Refinement



Vertex-Point Refinement



7. Imrod



8. T-Rex

