# Supplemental of Analysis of Acceleration Structure parameters and Hybrid Autotuning for Raytracing

Killian Herveau, Philip Pfaffe, Martin Tillmann, Walter F. Tichy, Carsten Dachsbacher

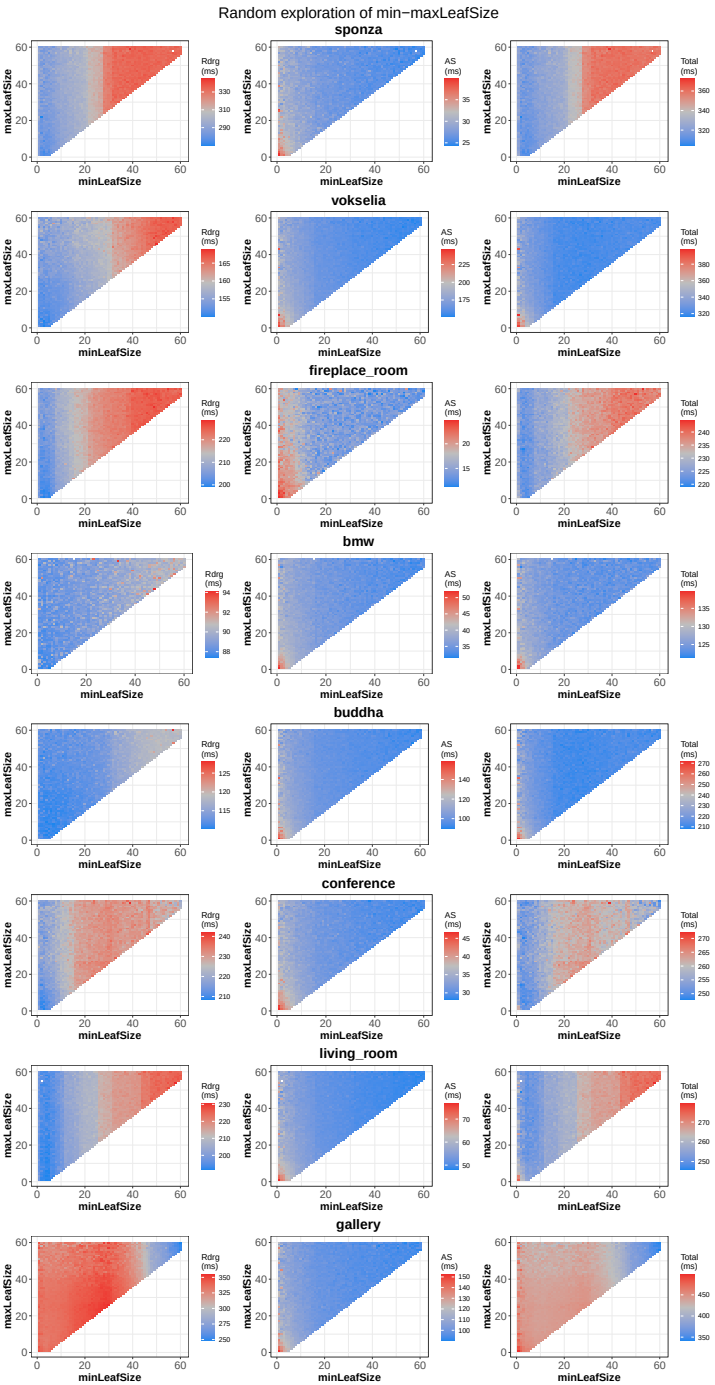



Figure 1: **Random exploration of `minLeafSize` and `maxLeafSize`** Scenes behave the same way: big leaves are good for AS build time, small for rendering time. Yet optimal tradeoff is different for every scene. Only gallery presents an unexpected behavior. This scene is made with scanned 3D data, which may have an odd structure

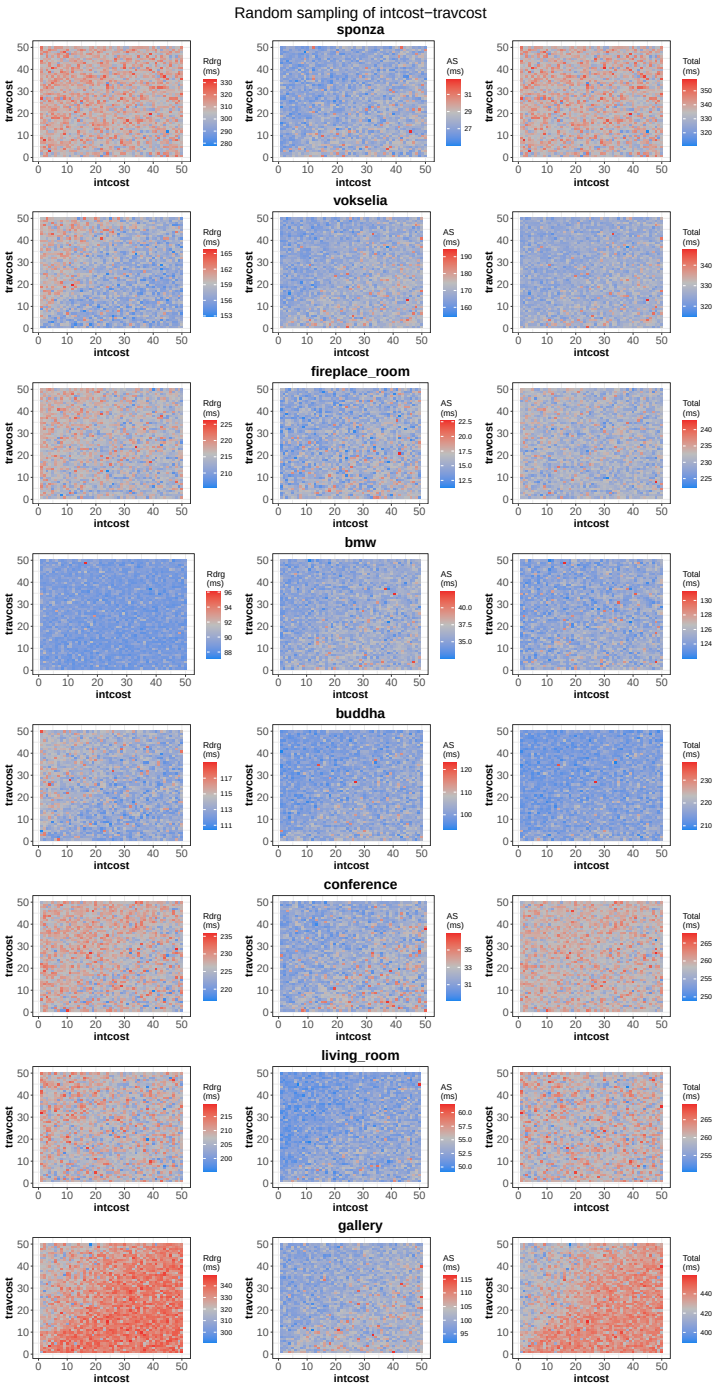



Figure 2: **Random exploration of `intcost` and `travcost`** Note: `quality` = 1. Most scenes show gradient of performance where the top left is better for build time (second column), and worse for rendering time (first column). Apart from this small effect, this shows that precisely finding a good couple is better than being in the right quadrant.

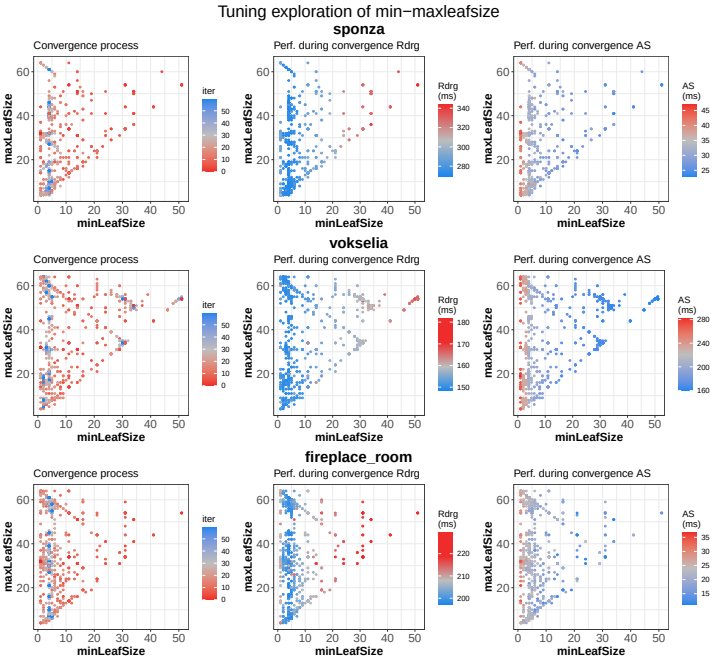



Figure 3: **Performance improvement during tuning (optimization: Rendering time)** `min/maxLeafSize` left: iteration number, center: rendering time, right: AS build time. small to medium scenes prefer small leafSize. `vokselia`'s optimum is 25 to 30 when optimizing for total time. Coherent with the distribution obtained in Figure 1

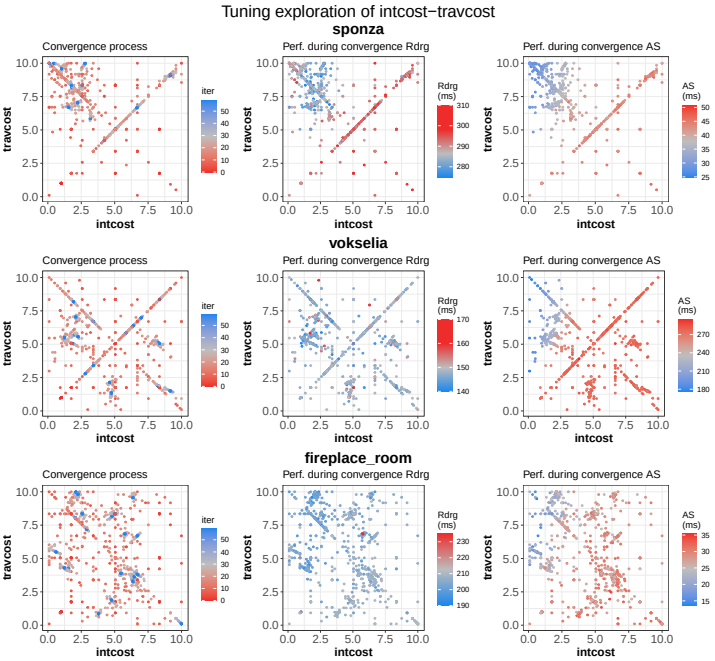



Figure 4: **Performance improvement during tuning (optimization: Rendering time)** `intcost`, `travcost` Optima are bundled around the top left corner: Low `intcost` and high `travcost` make the tree structure shallower when no other parameter controls the tree building. Straight lines show the exploration path of Nelder-Mead.

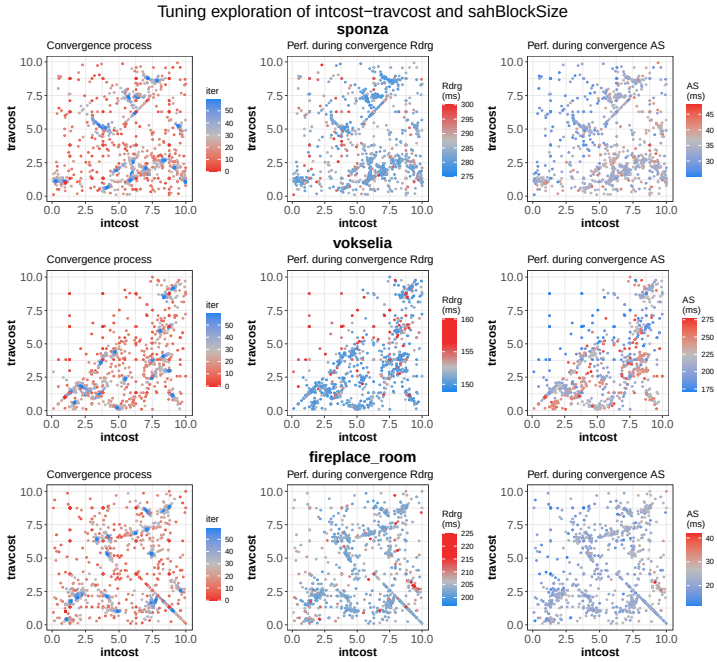



Figure 5: **Performance improvement during tuning (optimization: Rendering time)** `intcost`, `travcost` **and** `sahBlockSize`. `sahBlockSize`'s value is mostly the hardware-optimal value, which is why it is not relevant to visualize it. Yet, optima are scattered significantly more than on Figure 4 but still organized in clusters. This showcases what parameter interdependency can do to the parameter space.

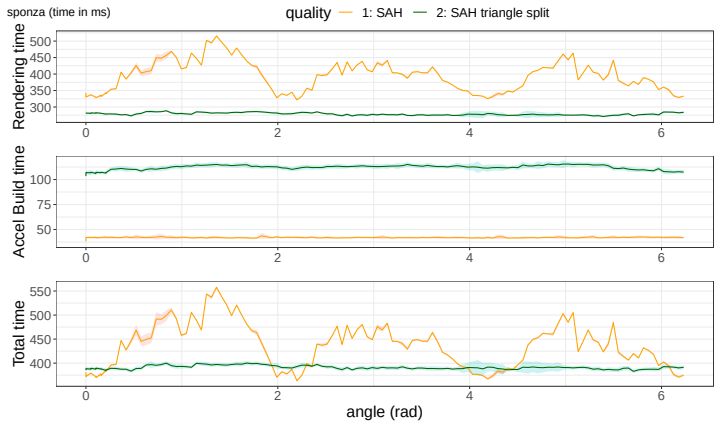



Figure 6: **Evolution of performance during cylinder rotation experiment.** `sponza`. Triangle splitting is the better choice most of the time here. Rendering time is heavily influenced by the intersections, the worst performance is 1.5 times the one of the start.

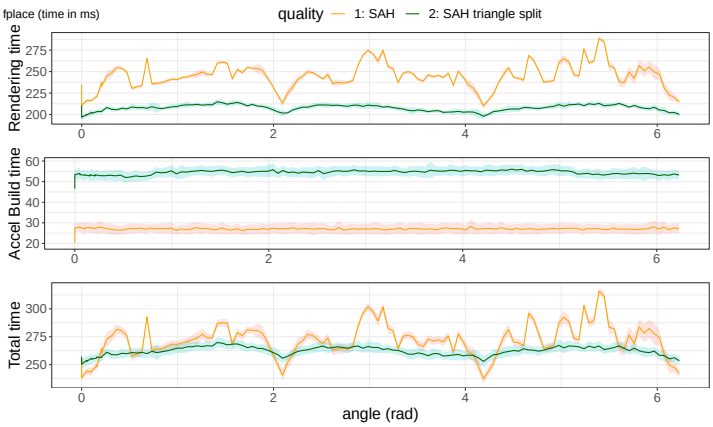
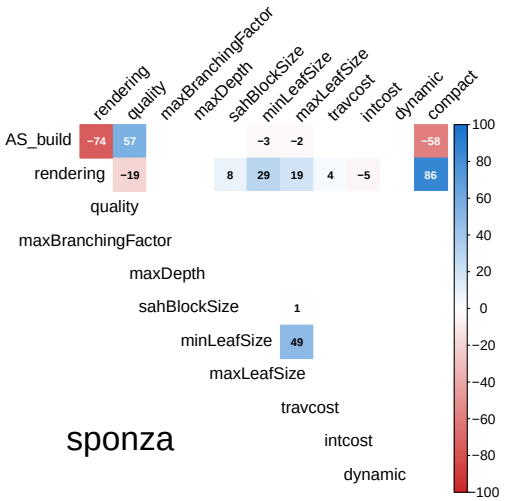



Figure 7: **Evolution of performance during cylinder rotation experiment.** `fireplace_room`. Evolution of AS build is very stable for this scene. The scene is too small for the cylinder to make a significant difference



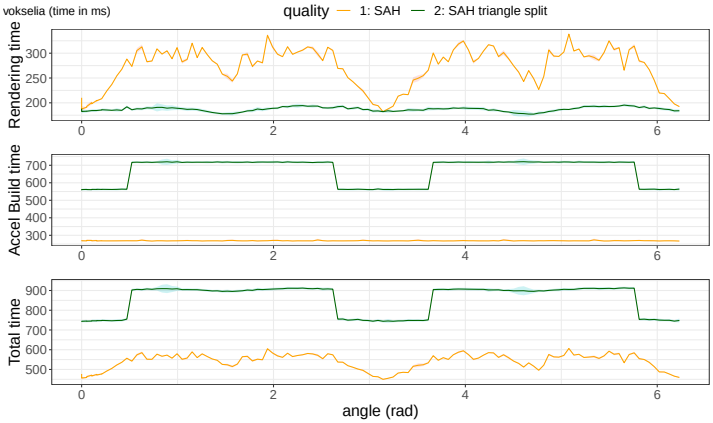
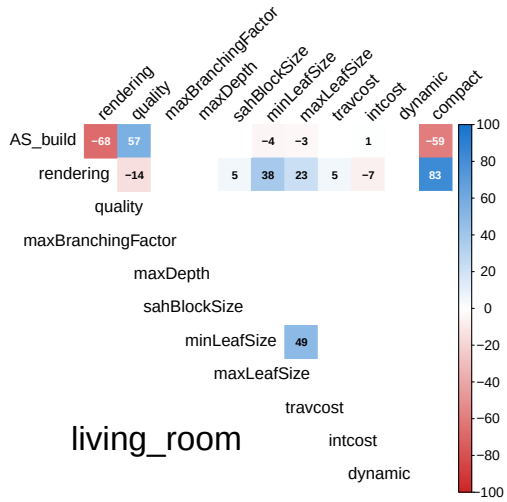




Figure 8: **Evolution of performance during cylinder rotation experiment.** `vokselia`. Sudden drop of performance at 0.5 rad, triangle split is not worth the cost for this configuration. This drop in performance is due to the cylinder intersecting a large amount of nodes, causing a lot of subsequent triangle splits. Performance in rendering time is far superior with splitting



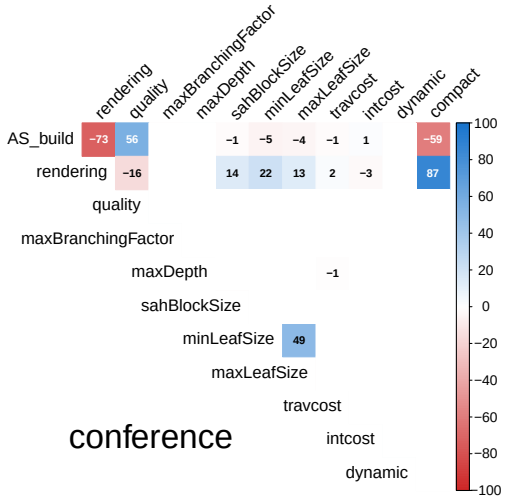




Figure 9: **Correlation matrix between parameters and timings** All rows are shown for completeness. Conference is a very hard scene but seem less sensitive to minLeafSize. min/maxLeafSize are correlated because min is always smaller than max

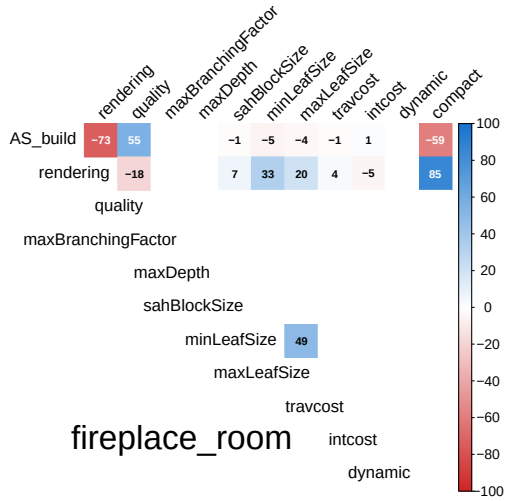



Figure 10: **Correlation matrix between parameters and timings** We see the opposite correlation between AS build and rendering regarding minLeafSize. This shows in random exploration too.

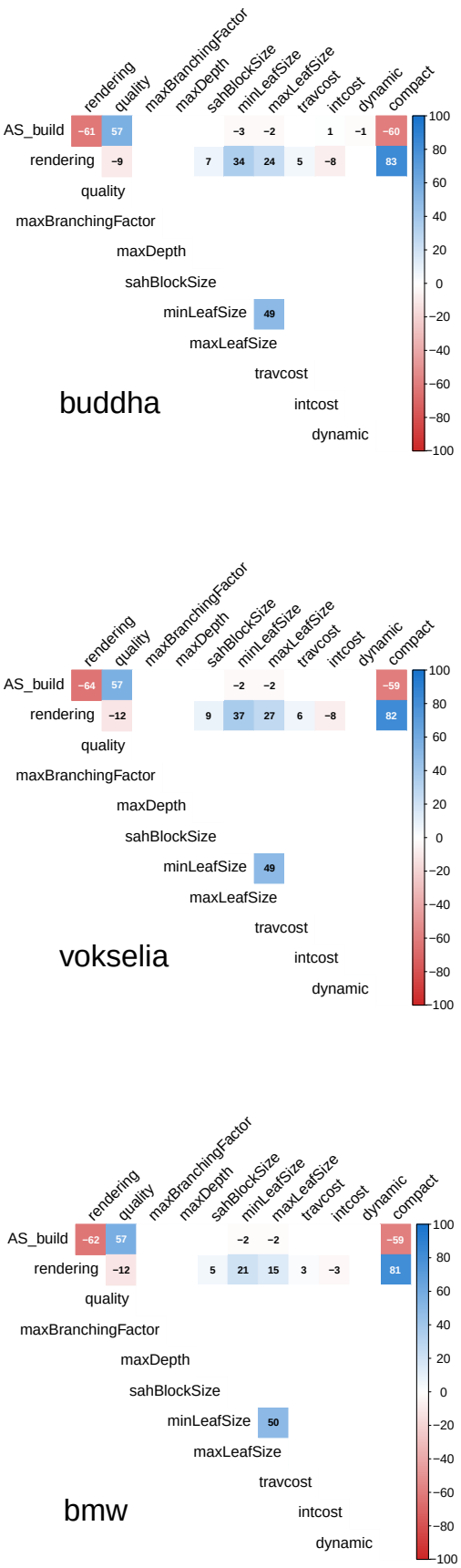



Figure 11: **Correlation matrix between parameters and timings** All scenes so far show correlation bewteen intcost/travcost and rendering time. This clearly hints that these parameters have a somewhat consistent effect. We discuss this more in the random exploration.