

The intersection of the planes can be non-robust when the planes are near parallel. Here are the tricks we use in our implementation:

- We first test if an intersection exists for every pair of planes. Given N planes, a naive pairwise intersection algorithm has a complexity of $O(N^2)$. We then collect plane triplets such that every pair in the plane triplet intersect. This is achieved by testing each plane against the known intersecting pairs, which also has a complexity of $O(N^2)$.
- The 3D vertices of the final faces are obtained by computing the intersections of the plane triplets. To cope with limited floating point precision, each vertex is identified by the indices (in an increasing order) of the three planes from which it is computed. By doing so, two vertices with almost identical positions can be distinguished. This turned out to be quite robust in handling very close and near parallel planes.

For more details, please refer to our implementation (i.e., [hypothesis_generator.h](#) and [hypothesis_generator.cpp](#)).