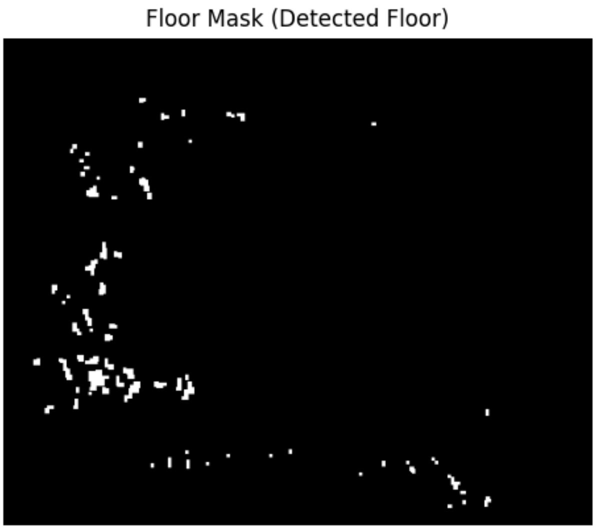
# Individual Part

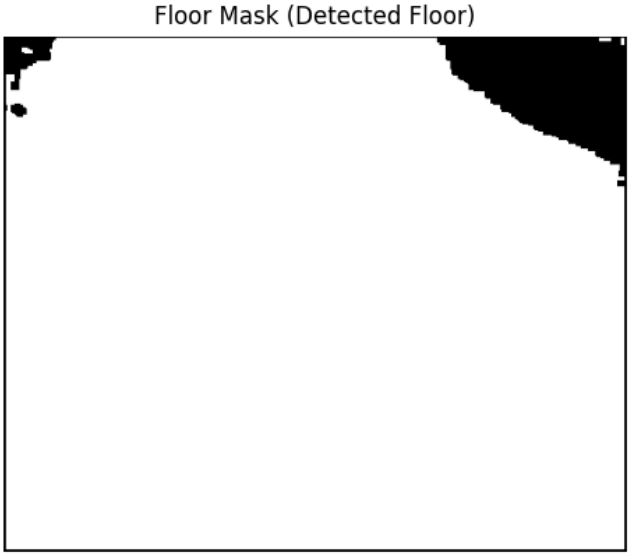
## MLESAC vs. RANSAC on Floor Detection

RANSAC is very sensitive to ε as being zero-cost makes RANSAC blind to how well an inlier fits. It means all inliers are treated equally as long as ε is selected properly (in this case 0.01), regardless of they are 0.0001 meter away or 0.009 meter away.  So in this case, if we choose a very low ε such as 0.0001, we exclude almost every “true” plane point. On the other hand, a large ε means a lot of inliers.

**On example1kinect.mat:**



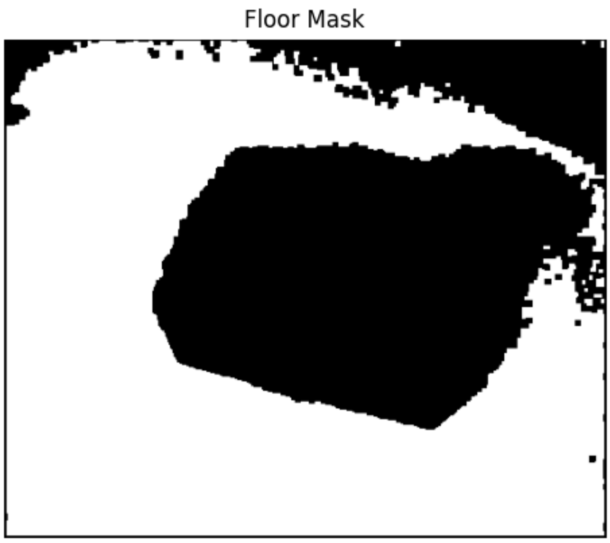

ε = 0.0001



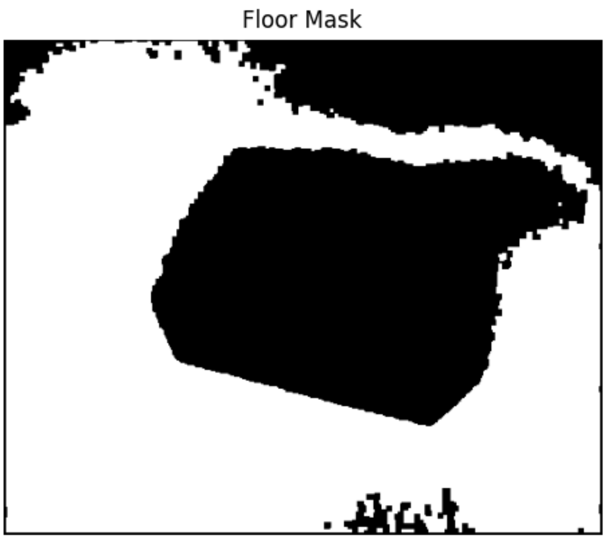

ε = 1

MLESAC on the other hand, is stable across entire ε range no matter how tiny or large it is. The reason is that inliers pay their actual distance and outliers pay a fixed price regardless of how far they are.

**On example1kinect.mat:**




ε = 0.0001




ε = 1

Advantage of this approach can be its **imperviousness** to ε. One disadvantage can be the extra parameter γ that has to be tuned.

## Preemptive RANSAC

As opposed to normal RANSAC, this one has a predictable budget, meaning you decide up-front how much work you are going to do. We choose M and B, the total number of random models and the size of each data-batch for pruning, respectively. Preemptive RANSAC gives you some sort on an overview on how much work we will have to do, which is suitable for systems where you cannot afford to go too long. On the other hand, tuning B and M could become a bit complex. In the implementation, the line:

```
keep_count = max(1, int(M * (2 ** (-int(evaluated_points / B)))))
```

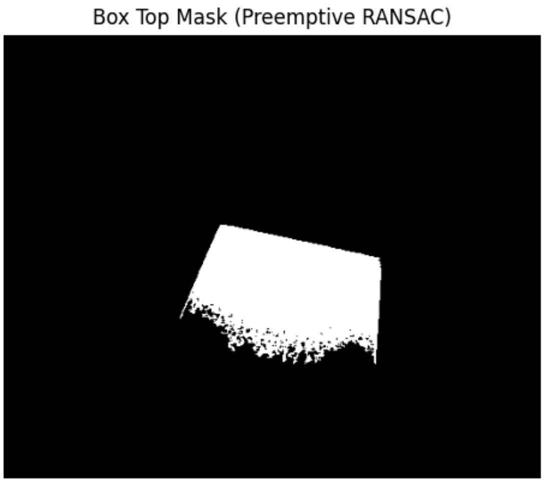
controls how many hypotheses are kept after each batch.

$$f(i) = \lfloor M \cdot 2^{-\lfloor \frac{i}{B} \rfloor} \rfloor,$$

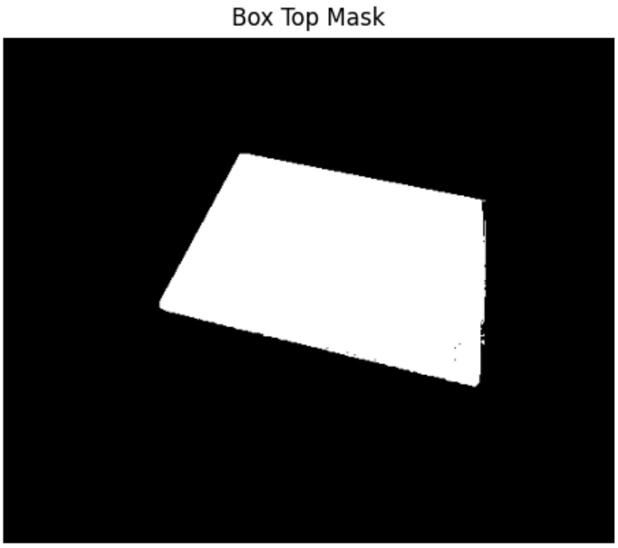
It means that after each batch of B evaluated points, the number of surviving models is halved.

## How effective are M and B?

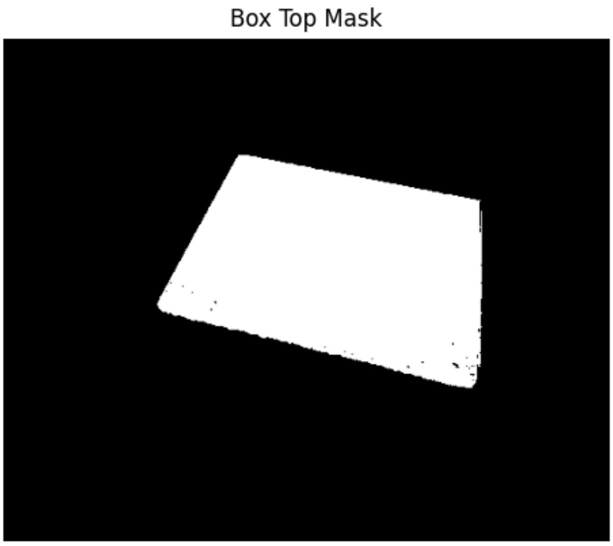
Small B means that the model is very aggressive when it comes to pruning. Bad models are pruned after little evidence, meaning after seeing a few points, it is dropped. It is faster but not reliable. In case of M, if M is a small value, this means that we may never sample a clean all-inlier triple. Therefore, the final plane is basically guesswork.




M = 10, B = 10




M = 50, B = 50




M = 1000, B = 1000

## In conclusion:

- **Under-budgeting M: N**o good hypothesis ever makes it through, so our estimated plane will wander randomly from run to run.
- **Over-pruning with too-small B: E**ven a correct model can be pruned if the first batch of B points happens to be mostly outliers (or noisy inliers).
- **Too-large M or B: Y**ou still get the same correct fit, but you pay more in compute without further accuracy improvements.