

1. Elastic balls collision model: this Matlab code is available online, as it shows in fig.1, changes need to be added to make customized map and collision response.
2. RRT and RRT* algorithm: the RRT Matlab code is available online, but with low efficiency (better programmed code is also available but is too long to start with it). To get familiar with RRT, I started with the easy one, and added RRT* based on RRT code.
3. Algorithm efficiency: the original code is simple and clear, but was poor written. Ideal RRT is supposed to be $O(N \log N)$, but this one is way more than (N^2) . I optimized part of the brute force search with some KNN/range search algorithm, it is now between $O(N \log N)$ and $O(N^2)$. Next I'm going to use spatial binning. This is supposed to have much better performance.
4. Both RRT and RRT* have been applied to the customized map

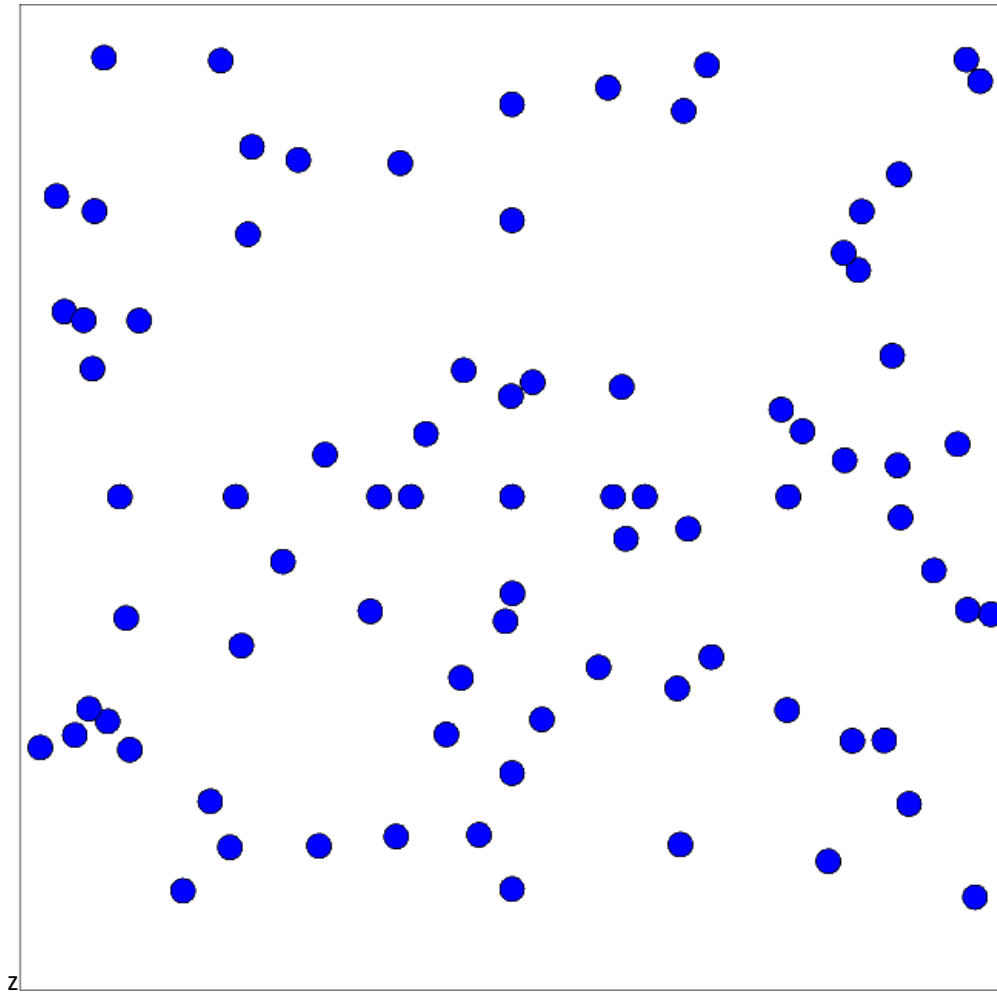


Fig. 1 elastic ball collision model

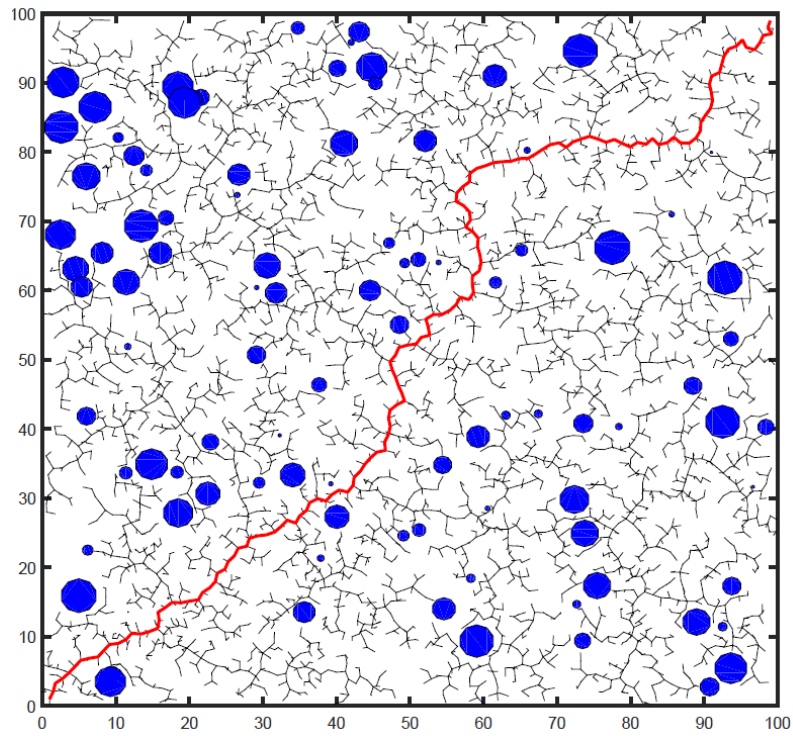


Fig.2 RRT with efficiency b/w $O(N \log N)$ and $O(N^2)$

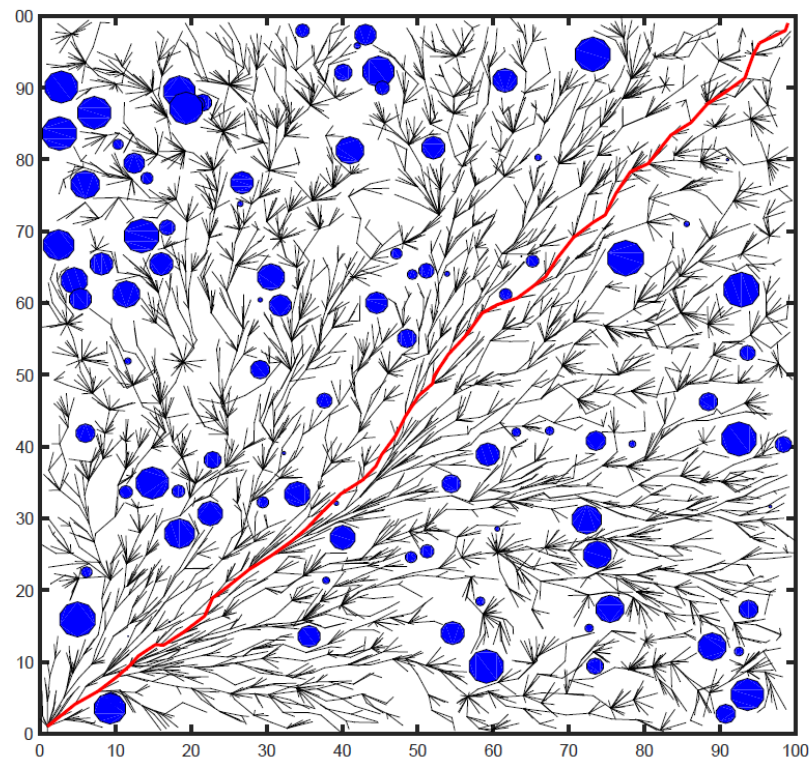


Fig. 3 RRT* with the same random map and random nodes, but more smooth route

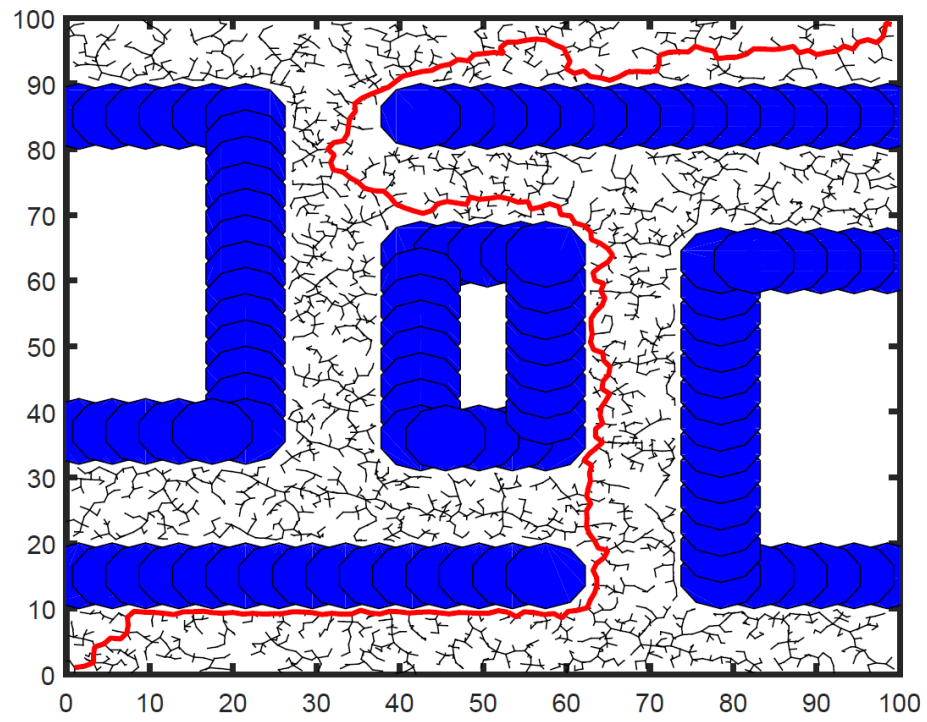


Fig. 4 RRT with customized map, about 3500 nodes and 36 s CPU time

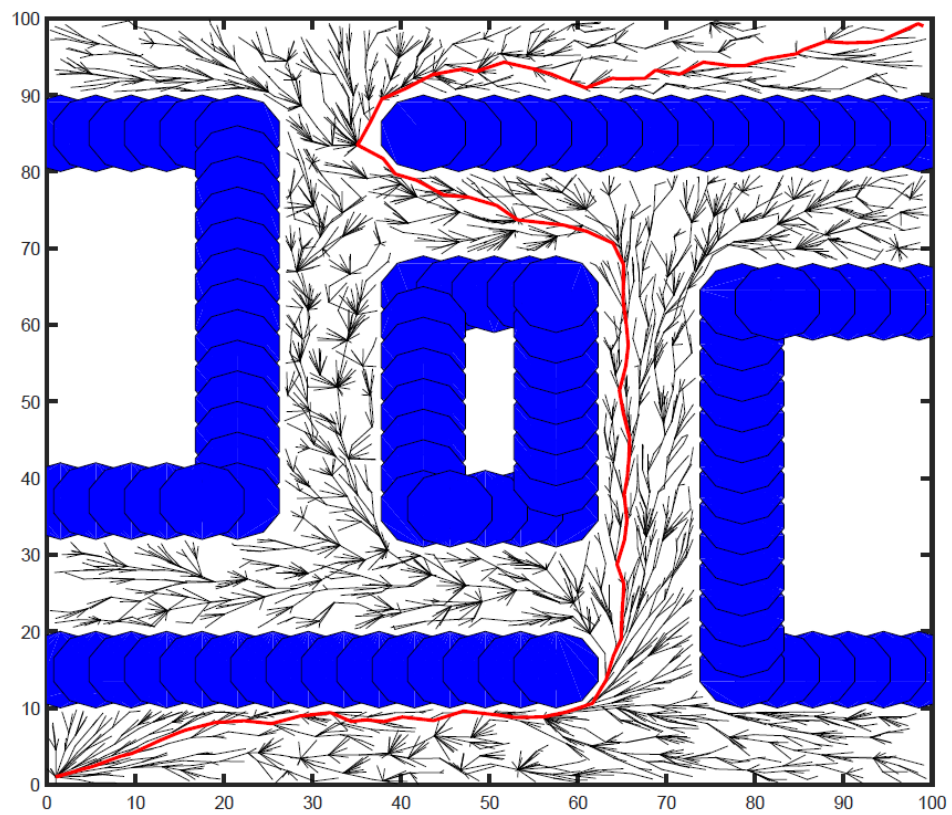


Fig. 5 RRT* with customized map, about 3500 nodes, 120s CPU time