Safe Navigation and Exploration in Environments Modeled as Gaussian Mixture Model

Tomasz Niedzialkowski

Professor : Alcherio Martinoli

Assistant(s) : Lucas Wälti

While conventional representations, such as voxel grids, are straightforward to manipulate, they are fundamentally limited by their finite resolution. As the extent of a mapped environment grows, these grids do not scale well in terms of memory footprint. Gaussian Mixture Models (GMMs) offer a powerful alternative by compressing this information, enabling efficient data storage and manipulation. Although GMMs have not yet achieved the widespread adoption in robotics seen by voxel maps or newer techniques like Gaussian Splatting, previous works—such as the GIRA framework—have effectively demonstrated their potential.

Building upon GIRA's fitting module, I developed a comprehensive system for sequential GMM fusion and autonomous exploration. The core of this system is a fusion engine that constructs the map incrementally. As new measurements arrive, Gaussians fitted to the recent point cloud are compared against the existing stored model using the Kullback-Leibler divergence metric. This allows the system to intelligently merge best-matching components while appending distinct, unmatched features to the global map.

To enable autonomous data collection without reverting to classical voxel conversions, I implemented an exploration logic driven by map uncertainty. Drawing inspiration from the RT-GuIDE project, I utilized heuristics to estimate the confidence of the map. I adopted a displacement metric—where the stability of a Gaussian's position over consecutive updates indicates certainty—and introduced a "fusion count" metric. The latter operates on the premise that the more frequently a Gaussian is fused with incoming data, the higher the confidence in its existence.

Although the project prioritizes continuous GMM representation, I introduced sparse discretization to structure the decision-making process. Two discretization approaches were implemented: a "simple" method using fixed map boundaries, and an "extended" method where cells are dynamically fitted to recent Gaussians to capture local details. These grids anchor a graph-based navigation system. Viewpoint candidates are generated in a circular pattern around the grid centers and connected to a graph of historical odometry nodes. Using Dijkstra's algorithm, the system selects the viewpoint that offers the best utility-to-distance ratio. The selected target is then processed by a trajectory generation module to ensure the drone executes a safe, collision-free path.

Benchmarking revealed that GMM fitting is computationally demanding, taking nearly 4s on a CPU versus 0.06s on a GPU, highlighting the critical need for hardware acceleration or employing more optimized fitting algorithm. Comparative tests of the uncertainty heuristics showed no significant performance difference between using displacement or fusion count. However, the analysis of exploration strategies demonstrated that the extended graph-based approach was significantly more robust to parameter changes and scalable to larger environments than the simpler greedy approach.

Although the project successfully met its set requirements, there is potential for further development. A key improvement would be the implementation of graph loop closure to enhance long-term navigation efficiency. Additionally, the system could be extended by replacing the current synthetic simulation odometry with a localization module based entirely on GMM registration, creating a fully self-contained GMM navigation stack.