

From my understanding and research, it seems that the best solution would be using Pose Graph Optimization to solve the problem at hand which stated we don't have the room dimensions. In Pose Graph Optimization we can let the robot explore its environment (the room) in cycle and we would use a sensor such as a lidar sensor and we would also need some way to know the robot's approximate deviation from its original position through odometry sensors such as IMU sensors or motor encoders. We can then approximately determine the room's dimensions by mapping out our readings. But of course, throughout the robot's cycle of moving around the room in a cycle there would be some noise and inaccurate readings introduced. This can be countered by having "constraints" between the robot's many positions throughout the cycle. Those constraints would act upon how much we trust our sensor's readings. These constraints would dynamically alter some of our readings to a more accurate value. What we would finally need to do would be to have a different type of constraint that connects between the robot's start position of the cycle and its ending position which can usually be determined by the robot reencountering similar features at said position that the robot encountered at the start of the cycle. And, again, depending on our sensors' accuracy we would pull the whole cycle and log of positions closer together. This process of pulling our cycle closer together minimizes errors that may have happened along the way. We could also raise our readings' accuracy by repeating this cycle multiple times.