Omdat een slam algoritme te complex is om zelf te maken, is er onderzoek gedaan naar verschillende libraries. De libraries worden hieronder kort toegelicht.

## Google Cartographer

De eerste library die veel belovend leek was Google cartographer (<https://github.com/googlecartographer/cartographer>). Hierin waren twee verschillende versies. Zowel met ROS als zonder ROS. Er is alleen gekeken naar de cartographer zonder ROS, omdat ROS niet werd gebruikt op de Manifold. Het was niet gelukt om dit te compileren op de Manifold zelf. Later zijn we dit nogmaals gaan proberen binnen een VM op Linux. Hier lukte dat wel, maar vanwege de gebrekkige documentatie is hier snel vanaf gezien.

## Gmapping (ROS)

Later is er gekeken naar gmapping (<http://wiki.ros.org/gmapping>). Omdat het op dit moment in het project belangrijker was om een werkende demo te krijgen is er allen nog geëxperimenteerd in een VM. Hierop is ROS geïnstalleerd, omdat gmapping dat nodig had. Na veel verschillende dingen te hebben uitgeprobeerd is het niet gelukt om dit werkend te krijgen. Dit lag aan de ROS support van de sweep. Het lukte niet om de informatie van de sweep door te sturen naar ROS, hierdoor kon gmapping dus niet de informatie gebruiken en werd er niks weergegeven. Hier is niet meer gelukt om dit werkend te krijgen.

## MRPT

Als laatste hebben is er gekeken naar MRPT. Dit was eerder ook al geprobeerd, maar toen was het niet gelukt dit te compileren op de Manifold. Toen er later gebruik werd gemaakt van de VM hebben we dit nogmaals geprobeerd. Ditmaal lukte het wel om MRPT te installeren en kond er mee ontwikkeld worden. De documentatie van MRPT was ook niet heel uitgebreid. Zo miste er bijvoorbeeld wat voorbeelden en moest alle informatie worden gehaald uit de API referentie website (<http://reference.mrpt.org/stable/>). Uiteindelijk is er door de project groep een kleine demo geschreven die obstakels in de omgeving kan detecteren. Nu is het aan de volgende groep om deze informatie op te slaan in MRPT en vervolgens bewegingen te verwerken binnen de opgebouwde kaart.