Abstract:

Contents

1 Introduction

2 Process Pipeline

The whole process of fault estimation can be seen as a pipeline. We start at the most basic starting block, the camera streams and end at the most complex block, the fault estimation. The pipeline is shown in Figure ??.

images/process_pipeline.png

Fig. 1: Process pipeline

The pipeline is divided into BLANK parts. Firstly, we preprocess the data. This is done by aligning the pointclouds and the RGB streams. There are multiple ways to align the pointclouds, but we found that, due to the high noise level of the depth data, the best way to align the pointclouds is to do it manually. Since we scale the point cloud into meters according to the meter per unit ratio provided by the depth camera, manual alignment is a matter of measuring the real world distances along the axis and then translating the pointcloud accordingly. Rotation on the other hand is a lot harder to do manually, so we use the NDT algorithm to align the pointclouds, with the measured distances as an initial guess.

The NDT algorithm, Normal Distribution Transform algorithm, is a non-linear optimization algorithm, which finds the best rotation and translation to align the pointclouds[NDT]. The more data we can provide to the NDT algorithm, the better the alignment will be. Therefore, if we provide the algorithm with the measured distances along the axis, it will be able to find the best rotation without changing the transaltion.

The fitness score is provided after the alignment and is a measure of how well the pointclouds are aligned. The fitness score is calculated by comparing the aligned pointclouds to the original pointclouds. The fitness score represents the mean squared distance from each point in the aligned pointcloud to its closest point in static pointcloud. Hence, the lower the fitness score, the better the alignment. Depending on the overlap of the pointclouds the fitness score can vary a lot. If the pointclouds are not overlapping significantly, the fitness score will be higher since the closest point is still far away and will be a bad indicator for a good alignment. Therefore, to validate the alignment of the pointclouds we also use the overlap score. The overlap score is calculated by comparing the aligned pointclouds to the original pointclouds. The overlap score represents the percentage of points in the aligned pointcloud that are within a certain distance of a point in the static pointcloud. With both metrics we can validate the alignment of the pointclouds.

The rotation and translation of the pointclouds are stored in a file, so that we can use them to align the pointclouds in the future.

The second part is the data acquisition, where we process the streams of multiple cameras and store the camera intrinsics, the RGB stream and the Depth stream. Using the intrinsics we can recreate the pointcloud based on the depth data at any point in the future.

The third part is the data population. To achieve the highest framerate we calculate the skeleton based on the recorded data and add it to the dataset in a seperate step. The RGB stream is used to create the dataset for the skeleton detection and the depth stream is used to create the pointclouds for the calculation of global skeleton points. We store both the local 2D coordinates in accordance to the image, as well as the global 3D coordinates based on the aligned pointclouds.

The fifth part is the fault estimation. This part is responsible for estimating the faults.