RANSAC Method to Registration

Owen Lu

Electroimpact Inc.

owenl@electroimpact.com

*Abstract*— In order to improve the overall robustness of the galvanometer registration algorithm, a RANSAC method has been employed. Using random selection of points based the probability of finding a good set of points can be calculated and iterated for a reasonable transform. Then the points considered inliers can be used for a registration solve.

# Introduction

Central to this implementation is a calibrated projector model from the EI Python application. Using knowing XYZ points in projector space with their corresponding DAC angles, the projector model is created. This is important since dependent on the calibration the threshold for “outliers” is defined differently.

# General Overview

Central to the RANSAC method is the use of multiple random samples of data points in order to minimize the probability that the subset includes outliers. Registration based on inlier data types is then assumed to give reasonable results.

The sample of data should be the minimum number of points to calculate the rotation matrix and translation vector which is 4.

Suppose is the probability that the subset of 4 points contains at least one outlier. Then we say that is the probability of failure since calculating a transform based on a subset with bad points will occur. It follows that the probability of failure of trials is which decays quickly to zero since .

It is then clear that for some number the probability of failure is sufficient to satisfy . We want to find the minimum value to reduce the probability of failure below threshold which is arbitrarily defined.

First assume that we have a calibrated projector model. The algorithm is then outlined.

RANSAC Transform Solver:

1. Define inlier threshold in DAC counts
2. Sample 4 random data points composed of DAC value pairs and corresponding XYZ points
3. Calculate the transform
4. Calculate the DAC angles using the calibrated model for XYZ points
5. Store the number of inliers using the criterion in step 1
6. If the number of inliers has increased, store the current as the best transform
7. Loop through steps 2-6 times
8. Using all points within the random sample and inlier list, calculate transform .
   1. This transform is calculated using all the inlier data

# Estimating number of trials

Let be the number of trials and be the number of outliers in the set of registration points. Some basic properties of this algorithm are listed:

1. As grows, the probability of failure decreases
2. As grows, more computational time is necessary
3. As grows relative to the probability of failure increases for a given

Thus, can be increased to offset the effect of increased likelihood of outliers at the cost of computational time.

Let be the probability of randomly sampling 4 inlier points in a set of . We can compute with Eq. 1.

|  |  |  |
| --- | --- | --- |
|  |  | () |

We define as the probability of failure we are willing to accept. Then we must find the minimum value that satisfies the inequality in Eq. 2.

|  |  |  |
| --- | --- | --- |
|  |  | () |

Note that computing can cause overflow issues so it makes to compute the ratio of factorials instead. A function shown in Eq. 3 is useful to avoid unsigned-integer overflow issues and can be easily constructed using a while loop.

|  |  |  |
| --- | --- | --- |
|  |  | () |

Then we can just evaluate two times to find without worry of overflow since should be small.

|  |  |  |
| --- | --- | --- |
|  |  | () |

Example Calculation

Suppose that we want a registration set of 7 to be robust even in the presence of 2 outliers.

Let

Substitution into Eq. 1 yields and can be calculated

Assume we want a better than 1/1,000 chance of success in the presence of 2 outliers.

Thus, we should run the RANSAC loop 45 times if we make this assumption. Interestingly if we exhaustively choose all sets of 4, there are only 35 combinations. We see that in the case of small sets of registration points, it is sometimes more efficient to compute exhaustively all registration subsets. However, computing 45 iterations of the RANSAC algorithm happens in approximately 5 seconds, which is a small fraction of the total time spent on this process.

# Exhaustive computation

In the case of small registration sets, it becomes worthwhile to exhaustively compute all the combinations of registration targets. The reason is that the number of trials can be similar to the number of combinations. Furthermore, computing all combinations means that if there is a set of 4 points that are considered inliers, the algorithm will work properly as it will be one of the combinations. Exhaustive computation becomes infeasible quickly as the number of registration targets rise, however, in most use cases with it is feasible.

# RANSAC Data Filter

A further step can be done to get a more accurate transform. After finding a reasonable transform using RANSAC, we can filter all outliers out of the data set and compute the best fit transform on all inlier data sets. The hope is that by using more points the random errors will cancel to give a more accurate transform.