LG2 Calibration Development

Owen Lu

Foreword

This document is an amalgamation of notes written during the development of the calibration algorithms and procedures for the LG2 from Aligned Vision. Previously, the LG2 required a laser tracker to perform coordinate transforms which is a tedious process that also requires expensive metrology equipment. Notably, their competitors instead developed a technique that requires multiple scans of the targets from different poses but does not require metrology to calibrate. The crowning achievement of my development in conjunction with Aligned Vision was to develop a way to calibrate a projector in a single position without using metrology to perform coordinate transforms between target space and projector space. This method has been shown to pass Boeing’s full projector certification at a 100% success rate with correctly assembled projectors, whereas the previous method only passed with around 40% success rate.

Central to the idea of using one position to estimate calibration parameters is the development of a good way to estimate pose with nominal estimates of parameters. Thus, it is a bit like the chicken and egg problem as you cannot calibrate if the pose is unknown, however, a good estimation of pose generally requires good calibration parameters. However, an ingenious method for camera calibration was developed by Zhengyou Zhang during his work at Microsoft Redmond that is able to simultaneously solve intrinsic parameters and poses without any initial guess. The only caveat is that at least 2 poses must be used. It is possible that in the future such a method could also be developed, but knowing the nominal mechanical configuration of the galvanometer system should generally be taken as a known, and a huge advantage at that.

The collection of notes details how the original mathematical model of the projector was built as well as its changes over time as better more general descriptions were implemented. The general aim of this document is mainly to record the thought process used to develop the methods and procedures used in the LG2 system today with the goal of creating the “best” calibration routine on the market. Best to me is loosely defined as the most robust with the fewest target scans possible. As such, this is not a formal scientific document and should be thought of as notes for reference and further study. Hopefully these notes will allow others to develop calibration further and give others a feel for the topic of study. Thus, sections of this document will be incomplete or in work indefinitely and revised when new techniques are discovered.

Table of Contents

[1. Laser Kinematics Derivation 13](#_Toc11764076)

[I. Introduction 13](#_Toc11764077)

[II. Coordinate frame definition 14](#_Toc11764078)

[III. Plane geometry 14](#_Toc11764079)

[IV. Ray Reflection transformation 18](#_Toc11764080)

[V. Finding Laser-Mirror intersection 19](#_Toc11764081)

[VI. Verification 21](#_Toc11764082)

[2. Ray Equation Solver 21](#_Toc11764083)

[I. Linearization 22](#_Toc11764084)

[II. Numerical calculation 24](#_Toc11764085)

[III. Alternative Method 26](#_Toc11764086)

[3. Description of Objective Functions 27](#_Toc11764087)

[I. Introduction 27](#_Toc11764088)

[II. Ray-Distance Method 27](#_Toc11764089)

[III. Ray Equation Method 29](#_Toc11764090)

[IV. DAC Error Method 30](#_Toc11764091)

[V. Conclusion 32](#_Toc11764092)

[5. Angle calibration functions of dual mirror laser galvanometer 33](#_Toc11764093)

[I. Introduction 33](#_Toc11764094)

[II. Forward kinematic assumption 34](#_Toc11764095)

[III. Linearity assumption 35](#_Toc11764096)

[IV. Error calculation 36](#_Toc11764097)

[V. Test results 37](#_Toc11764098)

[VI. Conclusion 39](#_Toc11764099)

[6. General description of projector geometry 41](#_Toc11764100)

[I. Introduction 41](#_Toc11764101)

[II. Unit vector generation 42](#_Toc11764102)

[III. Ray-plane intersection 43](#_Toc11764103)

[IV. ray reflection 44](#_Toc11764104)

[V. Serial ray generation 44](#_Toc11764105)

[VI. Objective Formulation 46](#_Toc11764106)

[VII. Initial conditions 47](#_Toc11764107)

[VIII. Test results 48](#_Toc11764108)

[7. DAC Correction with LOESS 49](#_Toc11764109)

[I. Introduction 49](#_Toc11764110)

[II. Procedure 51](#_Toc11764111)

[III. DAC Correction 52](#_Toc11764112)

[IV. LOESS Parameters 55](#_Toc11764113)

[8. Verification 57](#_Toc11764114)

[I. Current Calibration Method 57](#_Toc11764115)

[II. Theoretical basis of verification 59](#_Toc11764116)

[III. Elimination of metrology in calibration 60](#_Toc11764117)

[9. Virtual Image Planes for Dual Axis Galvanometer Pose Estimation 63](#_Toc11764118)

[I. Introduction 63](#_Toc11764119)

[II. EPNP via Gauss-Newton 68](#_Toc11764120)

[1. Define the control points 71](#_Toc11764121)

[2. Calculate the weights for each point 72](#_Toc11764122)

[3. Generate the constraint matrix 73](#_Toc11764123)

[4. Solve for the right singular values of 75](#_Toc11764124)

[5. Gauss-Newton algorithm to linear combinations 76](#_Toc11764125)

[6. Recover the camera control points 82](#_Toc11764126)

[7. Calculate the coordinates in camera space 83](#_Toc11764127)

[8. Solve the rigid body transformation problem 85](#_Toc11764128)

[IV. Rigid Transform Refinement 88](#_Toc11764129)

[V. Transform Generator 92](#_Toc11764130)

[I. Spherical Unit Vector Description 92](#_Toc11764131)

[1) Quaternion Description 92](#_Toc11764132)

[2) Rotation matrix description 93](#_Toc11764133)

[3) Transformation Matrix 94](#_Toc11764134)

[III. Conclusion 96](#_Toc11764135)

[10. ransac registration 97](#_Toc11764136)

[II. Introduction 97](#_Toc11764137)

[III. General Overview 98](#_Toc11764138)

[IV. Estimating number of trials 101](#_Toc11764139)

[V. Exhaustive computation 105](#_Toc11764140)

[VI. RANSAC Data Filter 106](#_Toc11764141)

# Laser Kinematics Derivation

*Abstract*—This document describes the method and assumptions used to obtain the kinematic equations for the laser projector system. Notably, coordinates are taken within the projector frame.

## Introduction

This project has been undertaken in order to confirm an understanding of the laser project system and eventually develop better calibration algorithms for the laser scanners. It has been historically noted that registration order of calibration points can create performance variations in the current LG projector. This is not ideal as the input data set should result in one solution describing the location and orientation of the system. Central to getting consistent results is the development of a mathematical model that represents the mechanical system with high fidelity.

## Coordinate frame definition

The construction of the laser projector allows good choices for the axes to be made in order to simplify calculations. The two mirror axes are assumed orthogonal to one another from one another.

We select the first stage mirror to have a rotational axis along the Z-axis while the second stage mirror has axis along the X-axis. Thus, the first stage mirror rotates about the origin.

## Plane geometry

The construction of the system is such that an easy choice of a coordinate frame can be made.

Two rotational axes of the mirrors are orthogonal, to one another. The laser is also parallel to .

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

As a simplification, the first plane is assumed to have no thickness, normal and rotates about the line which is parallel to the z-axis.

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

In Figure 1 this virtual plane represents a mirror plane of zero thickness and will be corrected for later in the model. The normal to the plane can be written in terms of .

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

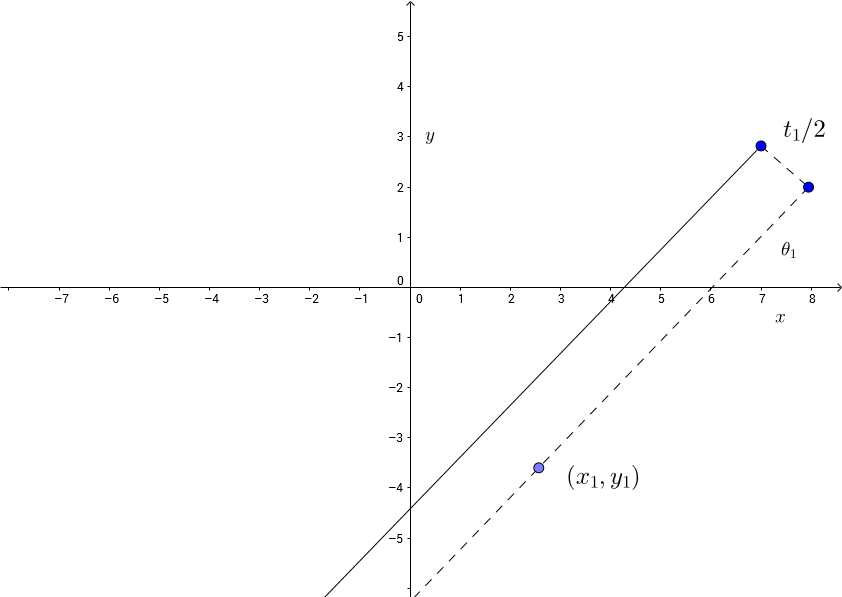


Figure 1 – Defined plane geometry in plane view

Therefore, the mirror plane can be defined by a dot product expression.

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

Now that the expression for the zero thickness plane is developed, an offset plane can also easily formulated. This offset plane will be shifted by half the mirror thickness .

This can be done easily by substituting for . Distributive law can then be applied to the dot product.

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

Similarly, the same method can be applied to describe the second mirror plane, with normal rotating about parallel to the x-axis. The diagram would be exactly the same as Figure 1 but with replacing the ordinate and replacing the abscissa.

|  |  |  |
| --- | --- | --- |
|  |  | (6) |
|  |  | (7) |
|  |  | (8) |

## Ray Reflection transformation

In order to find the ray emanating from the second mirror surface, the laser vector must undergo two reflection transformations.

To find reflected vector to input vector reflected across plane of unit normal , we can use the below equation.

|  |  |  |
| --- | --- | --- |
|  |  | (9) |

Let denote the ray vector after the first planar reflection

|  |  |  |
| --- | --- | --- |
|  |  | (10) |

Let denote the ray vector after the second planar reflection

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | (11) | (1) |

At this point since can be calculated, we only need point the point where the laser intersects the second plane.

## Finding Laser-Mirror intersection

Since the laser originates on the x-axis we can generate a line .

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | (12) | (2) |

Substitution of allows us to solve

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | (13) | (3) |

Therefore, the point which is the intersection of the laser and the first mirror plane is given below.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | (14) | (4) |

Then the line used to find the intersection of the laser line and the second mirror is using line and the second mirror plane.

|  |  |  |
| --- | --- | --- |
|  |  | (15) |

Substitute and solve for

|  |  |  |
| --- | --- | --- |
|  |  | (16) |
|  |  | (17) |
|  |  | (18) |

The last point on the plane can be solved by using line .

|  |  |  |
| --- | --- | --- |
|  |  | (19) |

Solving for such that allows for the final point to be found.

Since has no components, we simply use the equation

|  |  |  |
| --- | --- | --- |
|  |  | (20) |
|  |  | (21) |

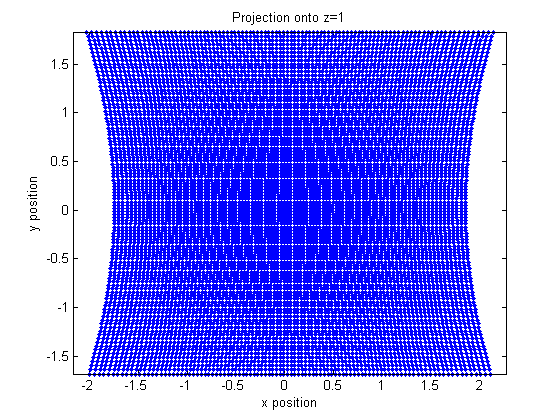


Figure 2 – Projection points using

Therefore, the final equation describing a projection on plane can be written below:

|  |  |  |
| --- | --- | --- |
|  |  | (22) |

## Verification

Using Solidworks 3D sketches and work planes, the modeled system can be constructed to verify the calculations. Calculations using parameters matched by the Solidworks model have been verified to be exact matches.

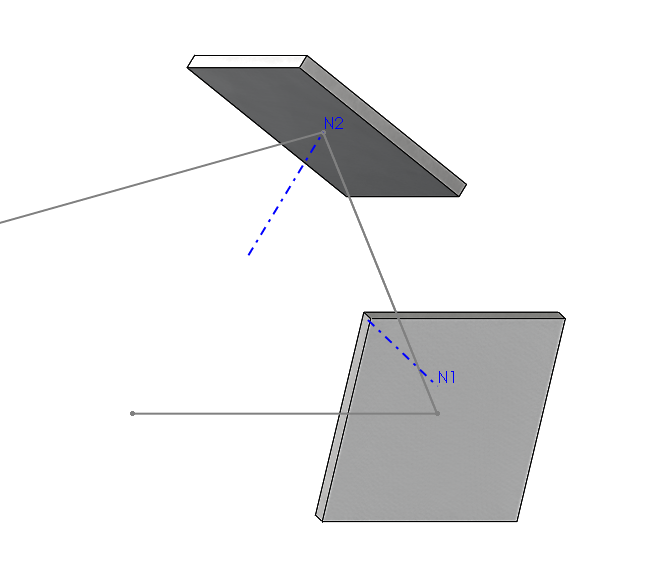


Figure 3 – Ray reflection solid model for verification

# Ray Equation Solver

*Abstract*— In order to generate DAC commands for the projector three equations are solved for two unknowns. In this implementation, Newton’s method is used.

## Linearization

The projector model generates the point on the second mirror from which the ray exits the projector as well as the direction of the ray.

This gives rise to three equations when solved generate a theoretical intersection of the ray and the point. Below it is represented by a vector equation.

|  |  |
| --- | --- |
|  | (23) |
|  | (24) |

Both and are both functions of the projector geometry and the DAC angles. We wish to find the DAC commands that best satisfies Eq. (23).

Linearizing with respect to the DAC angles allows us to calculate the solution very quickly. In practice, numerically differentiating using a forward difference with a step of 1 DAC count is sufficient. In short, we wish to find the derivative matrix to calculate the solution of Equation 4.

|  |  |
| --- | --- |
|  | (25) |
|  | (26) |
|  | (27) |

## Numerical calculation

To calculate the derivative matrix with respect to we simplify evaluate the function three times.

Function evaluations:

|  |  |
| --- | --- |
|  | (28) |
|  | (29) |
|  | (30) |

Then, the overdetermined system is solved for an angular increment.

Let

|  |  |  |  |
| --- | --- | --- | --- |
|  | | (31) | |
|  |  | | (32) |

Solve

|  |  |  |  |
| --- | --- | --- | --- |
|  | | (33) | |
|  |  | | (34) |

Iterate

|  |  |  |  |
| --- | --- | --- | --- |
|  | | (35) | |
|  |  | | (36) |

In practice, this takes 1 iteration to converge for the nominal projector model parameters within 1 DAC count. However, 2 or 3 iterations is recommended.

## Alternative Method

In order to obtain a solution in 1 iteration, the norm constraint can be relaxed. Thus, we add variable to Eq. (23). The derivative matrix can then be written as a 3 by 3 square matrix which can be solved explicitly and converges quickly.

|  |  |  |
| --- | --- | --- |
|  |  | 37) |

|  |  |  |
| --- | --- | --- |
|  |  | 38) |

# Description of Objective Functions

*Abstract*— From discussion with AV, the accuracy of the inverse algorithm to compute DAC commands from 3D coordinates is of critical importance. Thus, ensuring that the objective is created properly to result in good inverse performance and precision is necessary. However, calculation speed is also an important consideration as implementations may become impractical if this important element is neglected. The methods in this paper are all in different regions of the speed-precision spectrum.

## Introduction

The underlying problem of generating an objective that uses errors in DAC commands upon inversion of coordinates is the speed. The methods that are described all have different accuracy and speed characteristics when used as an objective and are used in different stages in the solving process.

## Ray-Distance Method

This was the first objective used when building the projector model. The perpendicular distance between a modeled ray and a point was used to quantify the model performance. The DAC inputs would be used to generate the ray, and the point is measured so the distance can be computed.

The main advantage of this technique is that computing the resultant ray from an input-ray and two reflections are extremely fast. However, using the resultant model to generate DAC commands was not consistent. The root of the problem results from the fact that the ray position and orientation cannot be measured via the current procedure. Thus, simply because a mathematical ray is close to a point, does not mean the inverse function generates very accurate DAC commands

|  |  |  |
| --- | --- | --- |
|  |  | (39) |

## Ray Equation Method

In order to generate two DAC commands with XYZ coordinates, three equations with two unknowns must be solved. This is summarized by the vector equation in Eq. (40) which is equivalent to Eq. (41).

|  |  |
| --- | --- |
|  | (40) |
|  | (41) |

Instead of varying DAC commands, model commands are varied in attempts to solve the equation. The computed vector norm of during optimization measures how well the model fits the data. The number also represents the Euclidean distance between the model point and the measured point.

Thus, minimizing the sum of squared norms with respect to projector parameters results in better DAC agreement between the data and model.

|  |  |
| --- | --- |
|  | (42) |

## DAC Error Method

This is the most direct method of optimization with respect to the most important objective. This objective uses the projector model parameters to invert positions to DAC commands and compares the sum of squared DAC errors. Using a linearization of the projector model with respect to the DAC commands the overdetermined system can be solved.

Thus we solve the vector equation in Eq. 5.

|  |  |  |
| --- | --- | --- |
|  |  | (43) |

The solution is represented by .

|  |  |
| --- | --- |
|  | (44) |

Let the DAC commands from the calibration data be represented by

|  |  |
| --- | --- |
|  | (45) |

Thus, we select the parameters to solve Eq. 8.

|  |  |
| --- | --- |
|  | (46) |

## Conclusion

The three objectives all have the inherent tradeoff between accuracy and speed. The Ray-Distance method is the fastest and the least accurate, while the DAC Error Method is the slowest and most accurate. In practice, it is helpful to start with the lower accuracy methods and quickly switch tohigher accuracy methods. This results in good error performance overall and reasonable solve time.

# Angle calibration functions of dual mirror laser galvanometer

*Abstract*— It was thought that galvanometer angles within AGS laser projects were known by measurement however, it has been revealed that this is not the case. Thus, although the galvanometer relationship between digital command rotation is approximately linear, the angle must be inferred via calculation and rays generated using a mathematical model and linear regression. This document outlines how such calibration functions can be inferred in the absence of direct measurement.

## Introduction

The problem, which has yet to be proven in a mathematical sense, is summarized in this basic statement:

Knowing the projector geometry you can solve for the angle pair that generates a ray through some point . Knowing the angle pair that generates through some point you can solve for the geometry of the projector. Knowing neither the geometry nor the angle pair, there are infinite solutions for both the angle pair and projector geometry that produces that passes through .

However, what these statements do not include is the assumption that the DAC angles can be converted to angles perfectly through a linear relationship. This is the most important overarching assumption in this model.

## Forward kinematic assumption

Previously, laser kinematic equations have been derived to describe the ray output of the second mirror. Thus, assuming some projector geometry the ray is a function of only two variables . The geometry is the exact geometry used as a reference in the projector design drawings and is referred to as nominal dimensions.

The ray is described with the equations below.

|  |  |
| --- | --- |
|  | (47) |
|  | (48) |
|  | (49) |

The ray in galvanometer coordinates, defined as frame B is given below.

|  |  |  |
| --- | --- | --- |
|  |  | (50) |

Thus, solving the system of 3 equations below will generate the inferred angle pair, with corresponding ray length .

|  |  |  |
| --- | --- | --- |
|  |  | (51) |

## Linearity assumption

In order to convert the DAC readings to mirror angles, a simple linear formula is assumed where mirror angle.

|  |  |  |
| --- | --- | --- |
|  |  | (52) |
|  |  | (53) |

Constants are all constructed by solving the system for and curve fitting the relationship vs. DAC readings.

## Error calculation

The error in the angles is calculated with the formula below.

|  |  |  |
| --- | --- | --- |
|  |  | (54) |
|  |  | (55) |

Ideally, random error would be the cause for disagreements. However, projector geometry is not perfect, and the main factor, which is mirror misalignment causes and to be functions of both DAC readings.

|  |  |  |
| --- | --- | --- |
|  |  | (56) |
|  |  | (57) |

This phenomenon must be further studied as understanding these plots in relation to the mirror misalignment may yield vastly more accurate theoretical results. Possible changes in the experimental procedure may be necessary to determine such relationships.

## Test results

AGS has provided 10ft calibration data along with their own algorithm error report. In general, the projector is calibrated at one distance and the error is reported as the RMS difference between the point in space the laser intersection on a corresponding z-plane. It has been discussed to instead use the ray-point perpendicular distance instead of constraining error to be in-plane, as this is an arbitrary constraint that inflates the error.

However, for the purpose of comparison, the same AGS error algorithm is used.

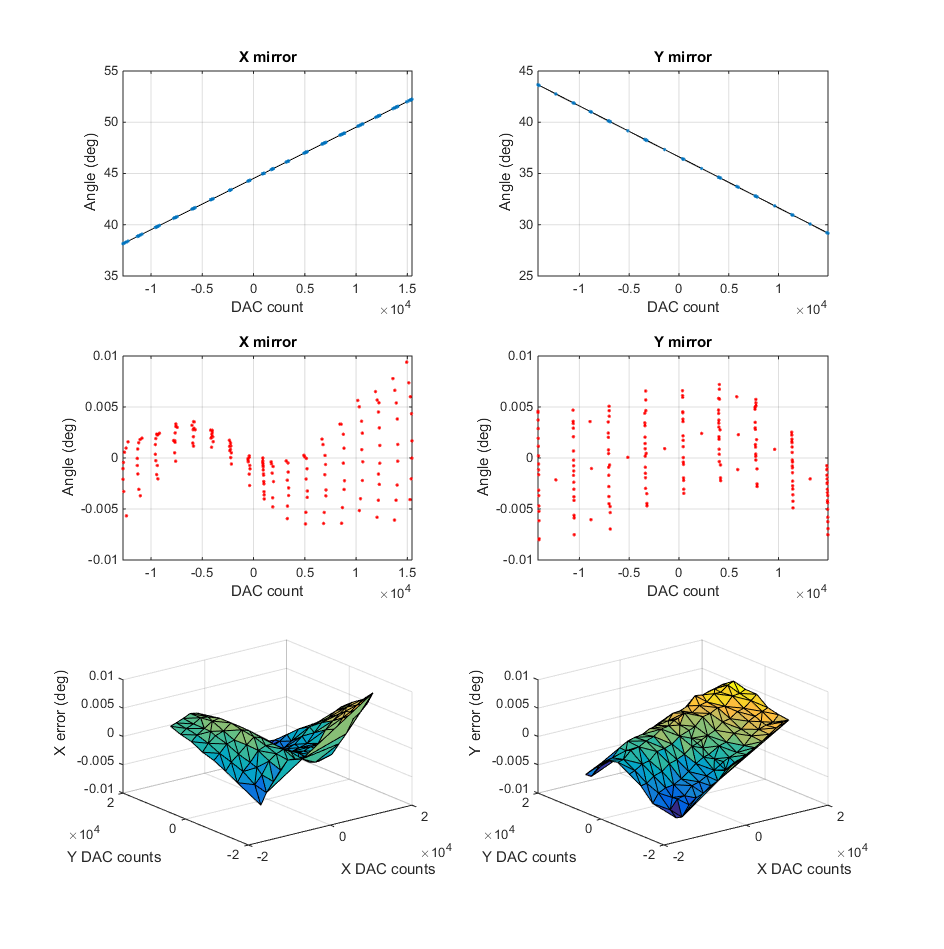


Figure 4 – (top) Linear curve fit to explicit solves;

(middle) Error between fit and solved angles

(bottom) Error between fit and solved angles vs. DAC readings

Using a LOESS quadratic fit in two variables, a fit was constructed to the error in order to correct the angles. The overall performance of the algorithm to predict the location of the laser ray outperformed the AGS algorithm by 37% for RMS error and 30% standard deviation.

## Conclusion

Currently, the method to solve angles and use curve fits to correct the error can generate good results at set distances. However, this method noticeably has lower accuracy when using the same projector at different distances. In theory, once the projector model parameters are defined, different distances should not affect the error plots. Furthermore, the error plots look smooth and have obvious grouping with meridian data. This implies that systematic error that is unmodelled must be added. More study in different methods and important model parameters to be added will be conducted to improve understanding of the system. The wish is to be able to construct a mathematical projector model that works well at multiple distances, resulting in more consistent performance.

# General description of projector geometry

*Abstract*— Using nominal projector dimensions and solving for mirror angles, a preliminary curve fit can be created to map both DAC angles to real angles. This allows optimization to begin near a feasible region. Ideally, all parameters that describe the geometry of the projector could be solved on in order to get the real dimensions of the projector. However, measurement error and redundancy in the effects of parameters on the objective function may prevent this from being possible. This document describes the main parameters that will be solved on to generate projector structures to be used for ray angle calculations.

## Introduction

In order to account for errors in construction, a more general model is necessary to account for variations. Fortunately, although the explicit equations to generate the output ray are very long expressions, the core of the problem is simple. There are two mirrors, described as planes in space, with a ray input that is reflected off both mirrors. It is known that any deviation from the mirror normal from the ideal orientation can greatly affect the ray output. This is known as mirror runout. Thus, the optimization method is used to solve 8 scalar values. The first 4 describe the linear calibration constants to convert DAC to angles, the last 4 describe the initial configuration of the normal. All other dimensions that describe the projector are assumed to be nominal.

## Unit vector generation

To generate unit vectors spherical coordinates are used. This allows the optimization to be done without constraints on vector length.

|  |  |  |
| --- | --- | --- |
|  |  | (58) |
|  |  | (59) |
|  |  | (60) |

Therefore, two angles can describe any unit vector in 3 space.

## Ray-plane intersection

Laser ray is defined by a point and a vector for some interval of . For some interval the lfinite line describes the light ray in space.

|  |  |  |
| --- | --- | --- |
|  |  | (61) |

The mirror plane is defined also with a point and vector by the below equation.

|  |  |  |
| --- | --- | --- |
|  |  | (62) |

The intersection between the line and the plane is then given by **Error! Reference source not found.**.

|  |  |  |
| --- | --- | --- |
|  |  | (63) |
|  |  | (64) |

## ray reflection

The final equation to generate the ray output from a mirror is calculated using the ray orientation and the normal vector. The calculation is to reflect across shown in Eq. (65).

|  |  |  |
| --- | --- | --- |
|  |  | (65) |

## Serial ray generation

Using a ray and a plane, the output ray can be generated. This process can be repeated from multiple mirrors. In this case, there are only two mirrors defined by . If the ray input is defined by then the following calculations will generate the output of the projector.

The output of , is defined by the ray-plane intersection and ray reflection equations.

|  |  |  |
| --- | --- | --- |
|  |  | (66) |
|  |  | (67) |
|  |  | (68) |

Applying the same equations to using generates in Eq. (71)

|  |  |  |
| --- | --- | --- |
|  |  | (69) |
|  |  | (70) |
|  |  | (71) |

## Objective Formulation

The objective used is the average perpendicular distance from the point in space to the corresponding ray generated by the project with corresponding angle pairs.

The perpendicular distance between and line is given by .

|  |  |  |
| --- | --- | --- |
|  |  | (72) |

The objective is then the average of over all data points. Currently, the first optimization algorithm takes into account only 8 variables describing the projector. Until dimensions can be verified, these will be the only ones used.

|  |  |  |
| --- | --- | --- |
|  |  | (73) |

## Initial conditions

Using initial conditions generated from the “Explicit method to solving angle calibration functions to dual mirror galvanometers" the seed values for can always be generated consistently. The selection of corresponds with the nominal configuration of the normal vector in the ideal construction of the projector.

|  |  |  |
| --- | --- | --- |
|  |  | (74) |
|  |  | (75) |

Therefore generates and generates .

## Test results

Using MATLAB’s built-in function fminsearch to perform optimization given the data sets, the objective value was roughly 1mm. This is a relatively ssmall error assuming a linear mapping between DAC commands and angles and is promising for future development.

# DAC Correction with LOESS

*Abstract*—In order to increase the accuracy of the inverse projector model, a LOESS curve fit is employed in the current implementation. However, in principle, any curve fit method can be used to correct the DACS.

## Introduction

After projector calibration has been done to solve for constants that give better accuracy via the inverse DAC calculation curve fitting is employed to correct the DAC values further. For each galvanometer axis, a curve fit is employed.

Let:

be the x-axis DAC value via inverse calculation

be the x-axis DAC value after correction

be the y-axis DAC value via inverse calculation

be the y-axis DAC value after correction

|  |  |  |
| --- | --- | --- |
|  |  | (76) |
|  |  | (77) |

The goal is to fit the functions and using the LOESS fit in 2 dimensions.

It can be easier to think of and as the error between the calculated values versus the measured values.

Let:

be the measured error in the x-axis

be the measured x-axis DAC command

be the measured error in the y-axis

be the measured y-axis DAC command

|  |  |  |
| --- | --- | --- |
|  |  | (78) |
|  |  | (79) |

The 2D curve fit is performed on with respect to for a calibrated projector.

## Procedure

In order to perform this correction, calibration data must be present. The basic steps are listed:

1. Using known target positions, find corresponding DAC angles
2. Calibrate the projector parameters
3. Perform an inverse calculation to calculate for each target
4. Calculate the difference between the measured and inverted values using Eq. (78) and Eq. (79)
5. Curve fit with respect to

## DAC Correction

After the curve fits have been completed correction is applied to the DAC via addition using Eq. (76) and Eq. (77) after inversion. This is a straight forward calculation however, the transform solving correction is affected in a less straight forward manner.

The underlying problem is that first, the position and rotation of the projector is unknown relative to the targets, therefore an inversion to calculate DAC pairs for each position is not possible with good accuracy.

If we simply use the measured DAC pairs the corresponding output rays will be incorrect since curve fit correction is applied to the DAC value directly. The problem is illustrated in a hypothetical procedure following:

1. Suppose the point is known in projector space
2. An inverse of to obtain is calculated
3. Curve fit correction to obtain allows more accurate projection to point in the forward sense.
4. Use in the forward model to calculate the ray
   1. In the model, the pair generates the perfect ray through
   2. The pair will generate a ray that does not pass through in general, which will add error to the calculated transform

In order to use the best rays to solve the transform, must be recovered from . This means that rays must be generated by adjusting the DACs by subtracting the curve fit functions from the measured values.

Therefore, we estimate the model DACs by using the following equations during transform estimation.

|  |  |  |
| --- | --- | --- |
|  |  | (80) |
|  |  | (81) |

## LOESS Parameters

During trials using LOESS style curve fits, there are two main parameters. Which are the polynomial and the span. It was found that a quadratic polynomial combined with a span of 12% gave the best results.

# Verification

*Abstract*—Using the current projector model and LOESS curve fits to for correction, DACs can be calculated to 1-2 count precision. When comparing multiple data sets taken at different distances, however, there is a large systematic error that is observed when using one data set to calibrate and the other data set to verify. This large error is almost completely eliminated by adding 3 rotational degrees of freedom to the projector axes selected by an optimization engine. This paper describes the experimental procedure that will be used to generate the data at multiple depths using only one metrology based transform from wall space to projector space. Included are also the calibration method, the theoretical basis for the experiment and procedure to obtain data for the verification.

## Current Calibration Method

Currently, the method used to generate projector parameters from a single data set is described in the list below:

1. From the nominal projector, ray equations are solved for the theoretical positions of the mirrors
2. A least-squares linear curve fit is used to map DAC commands to the theoretical angles
3. Optimization for a rotational transform is solved to minimize errors
4. A pair of LOESS based curve fits are used to map theoretical DAC commands to observed commands

After these steps, the parameters of the projector are fully defined. In the current model, 29 parameters describe the projector and 6 describe a coordinate transform of the projector relative to the frame that the 3D coordinates are measured in.

At this point, on a new data set, the coordinate transform that minimizes the squared DAC error is used. This is very similar to the certification procedure output currently implemented. However, all 3D coordinates are used to generate only a rotational transform.

## Theoretical basis of verification

Currently, it is known that estimating the rotational transform of the projector is prone to random error due to the distance between the machined targets compared to the precision of the laser tracker. Thus, if multiple data sets taken at different distances are used, there is a rotational transform characterized by quaternion **for each data set**. This means that for each new data set, 3 new degrees of freedom are introduced that greatly impact the performance of the model.

However, if the projector were to remain stationary and targets at different nominal distances were scanned there would be a single transform for the projector, as it has not moved. Thus, we would expect that calibrating on two distances of targets, similar RMS error would result across both sets of targets. Currently, this is not true, and additional rotational transforms are used to compensate for this error. The validity of this method has not been assessed in detail so far to my knowledge.

## Elimination of metrology in calibration

It was previously brought to my attention that one competitor uses a technique calibrate the projector that is not based on metrology. Therefore, they must inherently solve for the 6 transform variables that define on each set to locate the projector in space for calibration.

Example:

Suppose 6 sets of data are used to calibrate the projector, then an extra 36 degrees of freedom to describe the 6 different positions/rotations of the projector.

The method I have used up to this point is very similar. For each data set, I have introduced instead 3 new variables describing purely the rotational transform of the projector.

On paper, if I use the 10FT data to calibrate the projector, and solve a transform on the 20FT, I can get 1-2 DAC count error. However, there is no way to verify whether this transform is physically correct, only that the DAC commands match very well. In practice, this quaternion results in a rotation matrix very close to the identity, which is reassuring but not proof that it is correct.

I believe (without direct proof yet) that this method would tend to overstate the accuracy when the problem is to generate DAC commands from 3D coordinates. I will have to verify this later possibly with this experiment. **However, in principle, I am sure that with some tweaks a method that can be used to generate projector calibration without metrology is definitely within reach.**

# Virtual Image Planes for Dual Axis Galvanometer Pose Estimation

*Abstract*— A fundamental problem in computer vision is camera calibration and pose estimation. Due to the importance of the field of study, pose estimation is now a well-understood topic and thus we look to leverage the same algorithms by converting the projector into a pinhole camera. After the conversion, the exact same techniques can be used to generate initial estimates of the pose, shortcutting years of theoretical development. A model based refinement technique can then be used to increase accuracy.

## Introduction

Central to the model of the Dual-axis Galvanometer Model (DGM) is that rays are generated by mirror reflection about two planes in space. In principle, the output ray, defined by a point and unit vector, can be calculated in different mirror configurations. A simple form of the DGM assumes that rays emanate from a single point. This creates a situation that is identical to the pinhole camera. Since camera pixel coordinates are generated by projection to a plane, all that is left to do is to calculate the intersection of generated rays and a plane to obtain virtual pixels.

**Electroimpact DGM EPnP implementation**

1. Nominal projector model is generated
2. DAC pairs corresponding to XYZ coordinates are simulated in the model to generate rays
3. Rays are projected onto plane in a pinhole camera coordinate system with
4. Projected coordinates are assumed to be pixels
5. EPnP algorithm is used in combination with Gauss-Newton (GN) Optimization
6. Rotation matrix and translation vector are obtained and converted back into AV coordinate system.

The second step to the method is a DGM reconstruction of all measured points combined with a Singular Value Decomposition (SVD) based rigid transform solver.

**Electroimpact DGM Rigid Transform Refinement**

1. Using transform from EPnP based estimation, points are brought into projector space
2. DAC xy pairs generate rays corresponding to XYZ positions
3. Calculate the nearest point on the ray to the XYZ coordinate point, or the point estimated by distance
4. Solve the rigid transform problem
5. Update the transformed points
6. Loop 4-5 until a mean error tolerance is achieved
7. Return

A side note is the Efficient Perspective-n-Point Pose Estimation (EPnP) is a method that requires a calibrated camera model in order to work and can generate transforms without necessarily using points assumed to be coplanar. This fact is extremely important AV’s use case, since calibration verification is done roughly in plane, while registration in tool coordinates can have arbitrary arrangements and an arbitrary number of targets. Due to the convention used to generate the virtual image plane, assuming a plane on , the camera intrinsic matrix, which is defined as a 3 by 3 matrix is the identity matrix.

In simplified versions of the DGM projection rays emanate from the origin in AV conventions. This is the same as a pinhole camera model. However, in pinhole camera models the Z-axis is along the direction of projection. Overall the AV coordinate convention is rotated from the standard intrinsic camera coordinate convention which makes coordinate conversion easy.

Figure 5 is shown to explain the idea of ray projection onto an image plane, creating the virtual camera pixels. The only difference is that the rays are not centered at the origin in the general DGM.

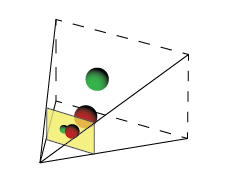


Figure 5 - XYZ projection on to the image plane

## EPNP via Gauss-Newton

The algorithm is broken down into 8 detailed steps.

1. Define control points in world coordinates for the homogeneous barycentric system
2. Calculate the weights based on the control points for each data points
3. For each of the data points, generate 2 constraints on the control points described in the camera coordinates. Concatenate all equations and form such that where is a 12 vector containing the 4 control points and their components within the camera system.
4. Solve for the right singular vectors of to describe the general solution space of .
5. Use the Gauss-Newton optimization method to generate the best linear combination which are the right singular vectors associated with the 4 smallest singular values of . Let be this best linear combination from a least squares optimization.
6. Recover the control points in camera space from .
7. Use control points in camera space and the previously calculated weights in step 2 to find all points in camera space.
8. Solve the rigid body rotation problem which defines the pose of the camera.

In general, since the model of the system is a linear model we attempt to describe as much of the model as possible as matrix operations.

First, the control points which are arbitrary are selected. Note they must be non-coplanar. We store these inside the matrix describing the control points in the world frame.

### Define the control points

We simply pick the orthogonal basis vectors of the rectangular world system and add the origin. The choice is arbitrary, but this choice is well conditioned which results in numerical stability.

|  |  |  |
| --- | --- | --- |
|  |  | 82) |

### Calculate the weights for each point

Let be the points in the world coordinate system where the in is a vector describing the coordinates.

To solve the weights of the barycentric system we can augment the matrices and by adding a row of ones.

Let:

The augmented be

The augmented be

|  |  |  |
| --- | --- | --- |
|  |  | 83) |

Where is a matrix where each column contains a vector of the control points weights for the point.

|  |  |  |
| --- | --- | --- |
|  |  | 84) |

Since is square we can simply invert it.

|  |  |  |
| --- | --- | --- |
|  |  | 85) |

At this point, has been calculated.

### Generate the constraint matrix

Matrix is derived from concatenating 2 constraint equations from each of the points. We assume that for the point

|  |  |  |
| --- | --- | --- |
|  |  | 86) |

It is assumed that matrix describing the camera is skew-less.

|  |  |  |
| --- | --- | --- |
|  |  | 87) |

This gives rise to two constraints, formed from the pixel equations. This means that

|  |  |  |
| --- | --- | --- |
|  |  | 88) |
|  |  | 89) |

Where

The resultant vector is a matrix which is solved for its right singular values. This is the same as solving for in the following equation.

|  |  |  |
| --- | --- | --- |
|  |  | 90) |

### Solve for the right singular values of

In practice, finding the 4 eigenvectors of that are associated with the smallest eigenvalues is more numerically stable than finding the null space of since noise can make it such that the null space is empty for example.

The maximum number of null vectors is 4 which is the reason we find the 4 eigenvectors associated with the smallest eigenvalues. This basically means that the vectors are sent very close to the origin. Roughly satisfying our above equation.

At this point, we have four vectors .

Assume that the corresponding eigenvalues have the following relationship: .

### Gauss-Newton algorithm to linear combinations

Therefore, is the best single null vector approximation, to . It follows that is the second-best approximation and so forth.

Thus, it makes sense to attempt to create a solution which is a linear combination of .

We wish to solve the weights that gives such that the norms between corresponding points in world space and camera space are equal. This creates an equivalent scale between the two coordinate systems and also refines their directions.

In practice, it is helpful to use the 2-norm squared and compare the values since the expressions are simpler.

For any pair of points in camera space, the norms squared should be equal to that of the pair of points in world space.

|  |  |  |
| --- | --- | --- |
|  |  | 91) |

Since we have defined every point as a weighted sum of control points, there are only 4 points that we need to consider. Since there are only 6 ways to pair 2 points within a set of 4, we have only 6 equations to be summed for overall squared error.

We define each the 6 differences as such

Let

|  |  |  |
| --- | --- | --- |
|  |  | 92) |
|  |  | 93) |
|  |  | 94) |
|  |  | 95) |
|  |  | 96) |
|  |  | 97) |
|  |  | 98) |

We notice that squaring the norm squared yields quartic equations, which creates negative solutions. For example, if was the vector of weights that minimized , then would also be a solution to the minimum.

In practice, this means that the rotation and translation are negative, which results in the points within the camera system having a negative coordinate. This is impossible due to camera coordinate conventions, therefore, the points are then multiplied by and the proper transform is found. This problem is avoided if the 2-norm squared residuals are used instead of the 2-norm squared-squared residuals. However, the derivative computation is elegantly expressed using the expression from earlier which is helpful in using Gauss-Newton methods.

The problem is then state below:

Minimize with respect to

Since Gauss-Newton requires the derivatives of the residuals to be computed for each data point

Let

|  |  |  |
| --- | --- | --- |
|  |  | 99) |

We formulate the Jacobian, which holds the derivative of with respect to in the columns, evaluated at in its row. is the vector containing all . This is basically a derivative matrix calculated for each data point. Note that superscript denotes the iteration count. This means that we must initialize .

Let

|  |  |  |
| --- | --- | --- |
|  |  | 100) |
|  |  | 101) |
|  |  | 102) |
|  |  | 103) |
|  |  | 104) |
|  |  | 105) |

This results in a general method to solve for regardless of the number of approximate singular vectors . For example, if we are only considering using vector to solve the system. Then is initialized and .

Calculating the linear combination to find is then the linear combination with the values

|  |  |  |
| --- | --- | --- |
|  |  | 106) |

Then the transformed control points in camera space can be calculated.

### Recover the camera control points

Let

|  |  |  |
| --- | --- | --- |
|  |  | 107) |

Then the transformed control points can be recovered in camera space.

|  |  |  |
| --- | --- | --- |
|  |  | 108) |
|  |  | 109) |
|  |  | 110) |
|  |  | 111) |
|  |  | 112) |
|  |  | 113) |

### Calculate the coordinates in camera space

At this point since the weights in can be used to calculate all the points **.**

|  |  |  |
| --- | --- | --- |
|  |  | 114) |

Although scales should be very close at this point, we compute the scale factor to equate the norms again.

Let

|  |  |  |
| --- | --- | --- |
|  |  | 115) |
|  |  | 116) |

Then we compute such that the squared error in norms

|  |  |  |
| --- | --- | --- |
|  |  | 117) |

Taking the derivative of with respect to

|  |  |  |
| --- | --- | --- |
|  |  | 118) |

Setting the derivative to zero allows us to express in closed form by using dot products, or equivalently, inner products.

|  |  |  |
| --- | --- | --- |
|  |  | 119) |

At this point, the scales are the closest possible using a single scaling factor with respect to a squared residual error.

At this point, we check if the values contained in are negative. If they are, then they are all negative, so we multiply by to get the correct points.

### Solve the rigid body transformation problem

Now we have two matrices with the 3D points in camera space and world space. The last step is to solve the rigid rotation problem. Start by calculating the centroids.

|  |  |  |
| --- | --- | --- |
|  |  | 120) |
|  |  | 121) |

The by abuse of notation matrix subtracted by the centroid denotes a point by point subtraction.

|  |  |  |
| --- | --- | --- |
|  |  | 122) |
|  |  | 123) |

Let

|  |  |  |
| --- | --- | --- |
|  |  | 124) |

Compute the Singular Value Decomposition (SVD) of

|  |  |  |
| --- | --- | --- |
|  |  | 125) |

Finally, the rotation and translation of the camera with respect to the reference coordinate frame is calculated.

|  |  |  |
| --- | --- | --- |
|  |  | 126) |
|  |  | 127) |

In practice, the process is completed 4 times to solve 4 different transforms using different numbers of null vectors. For example, we compute the linear weights for 1 vector, 2 vectors, 3 vectors, and finally 4 vectors. We always keep the smallest null vectors first and add the next smallest to compute the transformed control points. Finally, we compare the reprojection error of all 4 sets in order to pick out the best transform.

## Rigid Transform Refinement

The equation which must be solved for each point, which currently is used iteratively as the inverse DAC calculation.

|  |  |  |
| --- | --- | --- |
|  |  | 128) |

Since is a unit vector

|  |  |  |
| --- | --- | --- |
|  |  | 129) |

Assuming that points in the projector space undergo a rigid transform from tool coordinates allows us to substitute.

|  |  |  |
| --- | --- | --- |
|  |  | 130) |

This equation should hold true for all registration points. The method to refine transform estimate is an iterative method that uses a model estimate to estimate the left side, to generate a point could which can be solved exactly for **.**

Assume that is estimated by the EPnP method. Then the registration point is below.

|  |  |  |
| --- | --- | --- |
|  |  | 131) |

Estimate

|  |  |  |
| --- | --- | --- |
|  |  | 132) |

Calculate the estimated point in projector space

|  |  |  |
| --- | --- | --- |
|  |  | 133) |

Solve the rigid rotation problem that minimizes the least squares distances. This is a problem with a known solution via SVD and also has known solutions for linear weights.

|  |  |  |
| --- | --- | --- |
|  |  | 134) |

Adjust the points in projector space.

|  |  |  |
| --- | --- | --- |
|  |  | 135) |

Recalculate for all points and loop until a tolerance is reached on the vector norm of the mean error.

|  |  |  |
| --- | --- | --- |
|  |  | 136) |

## Transform Generator

*Abstract*— In order to generate a transformation matrix efficiently, a method is devised using three angular parameters and three translation parameters. The method simply combines an angular parameter with a unit vector generated by using standard spherical coordinates which generates a description of axis-angle rotation. This is then concatenated with a translation vector and can be easily applied to a series of augmented vectors.

### Spherical Unit Vector Description

The projector model is made up of 5 main variables. These variables describe the input ray, the two galvanometer axes, and the two mirror normal vectors. Thus, all five variables have a unit vector contained within them. To generate these unit vectors without constraints on the magnitude or further normalization calculations, spherical coordinates are used.

|  |  |
| --- | --- |
|  | (137) |

### Quaternion Description

In general, an axis angle rotation can be described with a unit vector   combined with the magnitude of rotation by using quaternions. Since can be described in spherical coordinates with the radius the quaternion can be written as a function of 3 angles .

|  |  |
| --- | --- |
|  | (138) |

### Rotation matrix description

Ultimately, the quaternion is used to generate a rotation matrix.

|  |  |
| --- | --- |
|  | (139) |

The rotation can be applied to vector such that is the product of the rotation matrix and original vector . When writing the vectors as row vectors the rotation matrix which is a 3 3 must be applied via right multiplication. Note that most standard rotation matrices are left multiplied to a column vector.

|  |  |
| --- | --- |
|  | (140) |
|  | (141) |
|  | (142) |

### Transformation Matrix

A translation described by a row vector can be added to the rotation matrix to describe a new transformed point. We will denote the newly transformed vector which also combines translation.

|  |  |
| --- | --- |
|  | (143) |

However, if an augmented vector is used in the transform we can write a new matrix when multiplied to generates .

|  |  |
| --- | --- |
|  | (144) |

We can check that matrix multiplication is still well defined since is 1 4 and is 4 3.

|  |  |
| --- | --- |
|  | (145) |

The reason that this notation is used is that we can augment multiple rows each containing a vector and apply the same transformation in single matrix multiplication.

|  |  |
| --- | --- |
|  | (146) |
|  | (147) |
|  | (148) |

To check the dimensionality we see that **P** is a 4 matrix and is a 4 3 matrix. Therefore, is 3 as expected.

## Conclusion

This hybrid method is used in order to estimate the pose of the projector by using at least 4 known target positions and their corresponding DAC angles. Notably, the sweet spot in turns of accuracy and number of targets seems to be around 10-12 from initial testing. When more than 4 known target positions are used, redundancy in principle allows for severe outliers to be removed from the data set. This is immediately possible with a Random Sample Consensus approach (RANSAC), although unlikely to be computationally efficient. Furthermore, this method has the potential to allow for tracker-less calibration as transforms are not measured. Currently, preliminary testing shows that tracker-less calibration on one scan of 296 points at 10ft is empirically accurate in the 10-15ft range, but to provide good results in the 20ft distance in the bottom left corner. Further theoretical development that may prove fruitful should be done on generating constraint equations on projector parameters for each added view.

# ransac registration

*Abstract*— In order to improve the overall robustness of the galvanometer registration algorithm, a RANSAC method has been employed. Using a random selection of points the probability of finding a good set of points can be calculated and iterated for a high chance of finding correct transforms. 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 the 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.

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

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.

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

Note that computing can cause overflow issues so it makes sense 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.

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

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

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

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 rises, 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.