Method of Virtual Image Planes for Dual Axis Galvanometer Pose Estimation

Owen Lu

Electroimpact Inc.

owenl@electroimpact.com

*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 pose, shortcutting years of theoretical development0. A model based refinement technique can then be used to increase the 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 xy 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 xy 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 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 1 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.

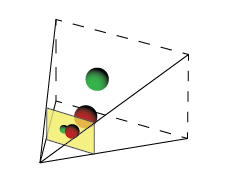


Figure - XYZ projection on to the image plane

# 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 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.

## 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

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

Since is square we can simply invert it.

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

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

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

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.

## 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.

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

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

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

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

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

## Recover the camera control points

Let

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

## Calculate the coordinates in camera space

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

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

Let

Then we compute such that the squared error in norms

Taking the derivative of with respect to lambda

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

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.

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

Let

Compute the Singular Value Decomposition (SVD) of

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

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.

Since is a unit vector

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

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 the by the EPnP method. Then the registration point is below.

Estimate

Calculate the estimated point in projector space

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.

Adjust the points in projector space.

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

# 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.