**Registration Method Overview**

Method of Virtual Cameras:

The “Registration” procedure used in Aligned Vision’s products is identical to the well-known problem of camera pose estimation in computer vision. If one obtains an accurate camera model, the pose can be solved accurately knowing points and their corresponding pixel coordinates where . We leverage this knowledge by using a calibrated projector model in order to generate virtual pixels. This allows the use of any known Perspective-N-Point (PnP) methods to generate the correct transform. An added step is necessary to convert the coordinates systems since both Aligned Vision and Electroimpact use different coordinate systems to describe the origin of the projector. In practice, this rough estimation is good enough to transform points to within 1 inch of their actual locations in projector space, allowing easy refinement after the initial estimate is given.

The steps are given below:

1. Convert the model vector outputs into pixels by projection **(vecs2pixel)**
2. Using corresponding XYZ points with the virtual pixels, solve the PnP problem. **(pnpProj)**
3. Add additional rotation to bring coordinates into Align-Vision coordinate convention (Applied as a constant rotation matrix)
4. The output of this process is a rotation matrix and translation which can be applied to points in the tool coordinates.

Model Based Transform Refinement:

It is assumed that the projector has been calibrated already before this process begins. It is assumed that given are the vector positions of the targets in the arbitrary tool coordinate system. After registration, the DAC pairs are found corresponding to the tool coordinates .

At this point the model is able to estimate the position of the points within projector space using the method of virtual cameras.

Since DAC pairs are present, the model can also predict where the points should be approximately located in projector space. The discrepancy between the two sets of points is the error. At this point, the solution to the rigid transform problem can be applied. The solution, which reduces least squares can be calculated via SVD and the output. This process can be repeated multiple times to refine the transform.

The steps are given below:

1. Use from the previous virtual camera method to bring tool points into projector space.
2. Use the DAC pairs to create rays from the model
3. Choose the points along the rays that are equidistant from the model standing point as the corresponding transform point. **(estimatePts)**
4. Solve the rigid rotation problem. **(tfSolve)**
   1. Iterate 2,3,4 until the mean error between the set of points is small
5. Exit the loop, now have been updated.

**Function inputs and outputs (MATLAB)**

**U = vecs2pixel(u)**

U is an nx2 array of pixel coordinates

u is an nx3 array of unit vectors

**[R,t] = pnpProj(X,U)**

X is an nx3 array of points

U is an nx2 array of pixel coordinates

R is the rotation matrix (to be applied to column vectors)

t is the translation vector (to be applied to column vectors)

**Xp = estimatePts(DACS,Xest,proj,correctionBool)**

DACS is an nx2 array of dac angles

Xest is an nx3 array of coordinates that represent the registration targets estimated in projector coordinates

Proj is an object containing all the parameters of the projector model, including curve fits

correctionBool is a boolean which specifies whether a curve fit correction is to be applied or not

X is an nx3 array representing the points generated by the model for the dac angles input

**[R,t] = tfSolve(X,Xp)**

X is an nx3 array representing the world coordinates of the registration targets

Xp is an nx3 array representing the model predicted points

R is the rotation matrix (to be applied to column vectors)

t is the translation vector (to be applied to column vectors)