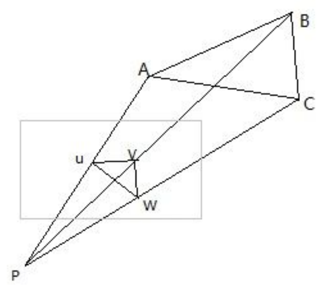
Project 4:P3P implementation

To do a P3P we will need 3 correspondences points to find the solutions and select one more point for solution justification. So four points are given in the 3D world coordinate system and we represented they and their 2D image coordinate system like below:

$$(A \leftrightarrow u, B \leftrightarrow v, C \leftrightarrow w, D \leftrightarrow z)$$

We will use A,B,C to solve P3P equation and D as the reference point to select the best result. We call the center of projection **P** where in the real world coordinate system P. Shown as the picture here(grey frame area represents the image space)



1.Simplify the equation for solving:

We can obtain 3 equation from the picture about:

$$\begin{cases} PA^2 + PB^2 - 2 * PA * PB * cos\alpha_{u,v} - AB^2 = 0 \\ PB^2 + PC^2 - 2 * PB * PC * cos\alpha_{v,w} - BC^2 = 0 \\ PA^2 + PC^2 - 2 * PA * PC * cos\alpha_{u,w} - AC^2 = 0 \end{cases}$$

According to Gao's paper, to solve this system, we need to first simplify this equation system. By dividing all equations by PC^2, and we give $y = \frac{PB}{PC}$, $x = \frac{PA}{PC}$, we can have:

$$\begin{cases} y^2 + 1 - 2ycos\alpha_{v,w} - \frac{BC^2}{PC^2} = 0 \\ x^2 + 1 - 2xcos\alpha_{u,w} - \frac{AC^2}{PC^2} = 0 \\ x^2 + y^2 - 2xycos\alpha_{u,v} - \frac{AB^2}{PC^2} = 0 \end{cases}$$

Then we give $v = \frac{AB^2}{PC^2}$, $av = \frac{BC^2}{PC^2}$, $bv = \frac{AC^2}{PC^2}$, this will transform the equations into

$$\begin{cases} y^2 + 1 - 2ycos\alpha_{v,w} - av = 0 \\ x^2 + 1 - 2xcos\alpha_{u,w} - bv = 0 \\ x^2 + y^2 - 2xycos\alpha_{u,v} - v = 0 \end{cases}$$

Where we have 2 coefficients:

$$\begin{cases} a = \frac{BC^2}{AB^2} \\ b = \frac{AC^2}{AB^2} \end{cases}$$

The last one can be written as $v = x^2 + y^2 - 2xycos\alpha_{u,v}$. By putting this equation back to 1 and 2 we obtain the two equations:

$$\begin{cases} (1-a)y^2 - ax^2 - cos\alpha_{v,w}y + 2acos\alpha_{u,v}xy + 1 = 0 \\ (1-b)x^2 - bx^2 - cos\alpha_{v,w}x + 2bcos\alpha_{u,v}xy + 1 = 0 \end{cases}$$

2. Normalize the data:

For a better result, we solve the problem by normalizing the data first, we remove unit from image point by applying the following equation:

$$\begin{cases} u'_x = \frac{u_x - c_x}{f_x} \\ u'_y = \frac{u_y - c_y}{f_y} \\ u'_z = 1 \end{cases}$$

We obtained f_x, f_y, c_x and c_y from camera calibration result. Then we normalize each point by equation:

$$\begin{cases} N_u = \sqrt{{u'_x}^2 + {u'_y}^2 + {u'_z}^2} \\ u''_x = \frac{u'_x}{N_u} \\ u''_y = \frac{u'_y}{N_u} \\ u''_z = \frac{u'_z}{N_u} \end{cases}$$

3. Solve the equation:

Second we need to find angles and distances that will be involved in the computing:

$$\begin{cases} cos\alpha_{u,v} = u_x * v_x + u_y * v_y + u_z * v_z \\ cos\alpha_{u,w} = u_x * w_x + u_y * w_y + u_z * w_z \\ cos\alpha_{v,w} = v_x * w_x + v_y * w_y + v_z * w_z \end{cases}$$

$$\begin{cases} AB = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2} \\ BC = \sqrt{(x_B - x_C)^2 + (y_B - y_C)^2 + (z_B - z_C)^2} \\ AC = \sqrt{(x_A - x_C)^2 + (y_A - y_C)^2 + (z_A - z_C)^2} \end{cases}$$

Then we solve the equation by zero decomposition based on Gao's paper and to obtain up to four solution, we obtain a quartic polynomial in x:

$$\begin{cases} a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 = 0 \\ b_1y - b_0 = 0 \end{cases}$$

And all coefficients a_i and b_i are written below($p = 2cos\alpha_{v,w}$, $q = 2cos\alpha_{u,w}$ and $r = 2cos\alpha_{u,v}$, $a = \dfrac{BC^2}{AB^2}$, $b = \dfrac{AC^2}{AB^2}$):

$$a_4 = a^2 + b^2 - 2a - 2b + 2(1 - r^2)ba + 1$$

$$a_3 = -2qa^2 - rpb^2 + 4qa + (2q + pr)b + (r^2q - 2q + rp)ab - 2q$$

$$a_2 = (2 + q^2)a^2 + (p^2 + r^2 - 2)b^2 - (4 + 2q^2)a - (pqr + p^2)b - (pqr + r^2)ab + q^2 + 2$$

$$a_1 = -2qa^2 - rpb^2 + 4qa + (pr + qp^2 - 2q)b + (rp + 2q)ab - 2q$$

$$a_0 = a^2 + b^2 - 2a + (2 - p^2)b - 2ab + 1$$

$$b_1 = b((p^2 - pqr + r^2)a + (p^2 - r^2)b - p^2 + pqr + r^2)^2$$

$$b_0 = ((1-a-b)x^2 + (a-1)qx - a + b + 1))((r^3(a^2 + b^2 - 2a - 2b + (2 - r^2)ab + 1)x^3 + r^2(p + pa^2 - 2rqab + 2rqb - 2rq - 2pa - 2pb + pr^2b + 4rqa + qr^3ab - 2rqa^2 + 2pab + pb^2 - r2pb^2)x^2 + (r^5(b^2 - ab) - r^4pqb + r^3(q^2 - 4a - 2q^2a + q^2a^2 + 2a^2 - 2b^2 + 2) + r^2(4pqa - 2pqab + 2pqb - 2pq - 2pqa^2) + r(p^2b^2 - 2p^2b + 2p^2ab - 2p^2a + p2 + p^2a^2))x + (2pr^2 - 2r^3q + p^3 - 2p^2qr + pq^2r^2)a^2 + (p^3 - 2pr^2)b^2 + (4qr^3 - 4pr^2 - 2p^3 + 4p^2qr - 2pq^2r^2)a + (-2qr^3 + pr^4 + 2p^2qr - 2p^3)b + (2p^3 + 2qr^3 - 2p^2qr)ab + pq^2r^2 - 2p^2qr + 2pr^2 + p^3 - 2r^3q)$$

We solve the two equation and obtained x, y values. Then we put back into the equation:

$$v = x^2 + y^2 - 2xycos\alpha_{u,v}$$

Then we obtain the value of v.

4. Find R, and evaluating the best solutions:

Then we can find up to 4 solutions for distance PA, PB, PC,as:

$$\begin{cases} PC = \frac{AB}{\sqrt{v}} \\ PA = x * PC \\ PB = y * PC \end{cases}$$

so we can find out the 3D correspondence points for point A, B, C:

$$\begin{cases} A = u * PA \\ B = v * PB \\ C = w * PC \end{cases}$$

Then we need to find the rotation matrix. First we need to normalize the data as we did in project 2 and 3. We will need two points set, 3D world points and 3D calculated points. We need to find the centroids of both data set.

$$Point = [x, y, z]$$

$$centroid = \frac{1}{N}\sum P^i$$

Then we do a single value decomposition,

$$H = \sum (P_A^i - centroid_A)(P_B^i - centroid_B)^T$$

Then we solve single value decomposition for H, and

$$R = VU^T$$

We will obtain up to four R and t and use the D point to find the different z. And then decide the best one by find the z who can minimize the geometry errors.

5.Result:

This is the image I use for calibration and I marked the 3D ground truth on the image.

1. We obtained P from project 4:

$$P = \begin{bmatrix} -0.0318831121161524l & -1.5679970513784O8 & 0.6217497784271375 & 373.1027780352189 \\ 1.301872661885767 & 0.0864034912993817 & 1.185585632832001 & 145.4679105125706 \\ -0.000532767714O322199 & -0.00007545562337819969 & 0.00220995958l449845 & 1 \end{bmatrix}$$

We can recover R and t:

$$R = \begin{bmatrix} 0.07965434747591044 & -0.9967122463097114 & -0.01482845185567472 \\ 0.9689118805545341 & 0.07392000179754633 & 0.2361050635935768 \\ -0.2342326891116249 & -0.03317425794949885 & 0.9716143864522784 \end{bmatrix}$$

$$t = \begin{bmatrix} 52.45071327625251 \\ -145.7984159098626 \\ 439.6525595345279 \end{bmatrix}$$

2. First we obtained the camera matrix to obtain the following:

$$f_x = 681.93745630416$$
$$f_y = 680.4544899305517$$
$$c_x = 291.7473867898793$$
$$c_y = 371.1215194342687$$

Then we have the following points in table below:

2D image corresponding points	3D world corresponding points
(96,120)	(0,0,0)
(409,101)	(210,0,0)
(99,554)	(0,-270,0)
(468,521)	(210,-270,0)

3. We put the data set into the P3P solve we built, I got 2 valid solution for the 4 degree equation, and here is two set of new 3D which is centered by the camera. And its R and t according to each transformation.

Solution 1:

3D world corresponding points	3D camera corresponding points
(0,0,0)	(-109.788,-141.153,382.477)
(210,0,0)	(41.9953, -96.9578, 244.243)
(0,270,0)	(-125.536,120.319,447.683)

$$R = \begin{bmatrix} 0.7227793966407636 & 0.06202905183457325 & -0.6882894307775745 \\ 0.21045310610108 & -0.9684147802771939 & 0.1337247302225042 \\ -0.6582548396510438 & -0.2415061284298553 & -0.7130044572138388 \end{bmatrix}$$

$$T = \begin{bmatrix} -109.7883942166427 \\ -141.152915125413 \\ 382.4767191600081 \end{bmatrix}$$

Solution 2:

3D world corresponding points	3D camera corresponding points
(0,0,0)	(-81.5858,-104.893,284.226)
(210,0,0)	(71.3672,-164.771,415.069)
(0,270,0)	(-122.478,116.46,433.325)

$$R = \begin{bmatrix} 0.7283476069462869 & 0.1514517500142022 & 0.6682605261971206 \\ -0.2851313293228712 & -0.8198274413624576 & 0.4965713356886989 \\ 0.6230649151526582 & -0.5522185561955716 & 0.5539357162154186 \end{bmatrix}$$

$$T = \begin{bmatrix} -81.58581492855507 \\ -104.8933786867341 \\ 284.2256237234901 \end{bmatrix}$$

We put both solution compute with point Z and calculate the geometry error, we found Error2 is much smaller than the Error1. So we pick solution 2 as the best solution.

4. **Compare our result to the openCV result:**

Here is the best R we got:

$$R = \begin{bmatrix} 0.7283476069462869 & 0.1514517500142022 & 0.6682605261971206 \\ -0.2851313293228712 & -0.8198274413624576 & 0.4965713356886989 \\ 0.6230649151526582 & -0.5522185561955716 & 0.5539357162154186 \end{bmatrix}$$

Here is the CV_P3P result:

$$R = \begin{bmatrix} 0.7355221916923913 & 0.1449032134137023 & 0.6618233633458432 \\ -0.2807712022875447 & -0.8238316794842061 & 0.4924113075917723 \\ 0.6165830337383829 & -0.5480038560261155 & -0.5652618330343531 \end{bmatrix}$$

Here is the CV_INTERATIVE result:

$$R = \begin{bmatrix} 0.996506230598651 & -0.0801656881105766 & 0.0234263679540127 \\ -0.00735366109372494 & -0.9751599501261058 & -0.2089388391896278 \\ 0.03959418176867907 & 0.2064861593606075 & -0.9776480791995525 \end{bmatrix}$$

Here is the CV_EPNP result:

$$R = \begin{bmatrix} 0.9962888030175931 & -0.08153745139940349 & -0.02757290338468313 \\ -0.07240090936963658 & -0.9671090445588301 & 0.2438405303778612 \\ -0.04654813964296346 & -0.2409392868583222 & -0.09694232980200045 \end{bmatrix}$$

Here is the best T we got:

$$T = \begin{bmatrix} -81.58581492855507 \\ -104.8933786867341 \\ 284.2256237234901 \end{bmatrix}$$

Here is the T from CV_P3P:

$$T = \begin{bmatrix} -82.80315116569588 \\ -106.7502439467854 \\ 290.6138284515796 \end{bmatrix}$$

Here is the T from CV_interative

$$T = \begin{bmatrix} -129.6102262042158 \\ -162.1822985451835 \\ 444.5621716131024 \end{bmatrix}$$

Here is the T from CV_EPNP

$$T = \begin{bmatrix} -129.6148921097624 \\ -162.8050183326458 \\ 391.2180542528273 \end{bmatrix}$$

5. Discussion

I learned that the P3P is using Geometry to compute the coordinate that was centered by camera. To find this coordinate we need to solve triangle problem in order to find the distance within points. In conclusion, if we calibrate our camera well we could easily reconstruct a 3D scenario.

I followed the implementation that was used by openCV library, so if I use P3P method in openCV, I got a very close result. However, I think PNP will require more points compare to P3P to be accurate. The R and t from last project looks very different from this times result. I would assume that’s because of the points we select is different. And we have many a large error for project 3 result.