Tárgymanipuláció kulcspontok detektálásán alapuló képfeldolgozással

Diplomaterv I. beszámoló

Ayhan Dániel
LIIAE1

Témavezető:

Dr. Kiss Bálint

Budapest, 2017. május 2.

Tartalomjegyzék

1. Bevezetés, feladatkitűzés

Az automatizált/robotizált megoldások egyre inkább elterjednek a mindennapi életünkben. Ennek egyik részterülete az olyan robotok kutatása, amelyek az ember otthonában képesek különböző feladatokat végrehajtani, például tárgyakat lokalizálni, megfogni, és áthelyezni. Egy alkalmazása lehet ennek, amikor a robot mozgáskorlátozott, vagy idős embereknek tárgyakat visz oda.

Tetszőleges tárgyak megtalálásához az egyik elterjedt módszer a vizuális információ felhasználása, képfeldolgozás használata.

A feladatom egy tetszőleges tárgy lokalizálására alkalmas algoritmus fejlesztése volt képfeldolgozás segítségével. Lokalizálás alatt a tárgy pontos pozícióját és orientációját, azaz *helyzetét* értem. A tárgyról előzetesen bármilyen információnk lehet, célszerű erre az adathalmazra is képfeldolgozással szert tenni. Feladat volt továbbá, hogy a tanszéki robotlaborban található robotkarral az adott tárgyat meg is fogjam, vagyis a képfeldolgozás információit felhasználó tárgymanipuláló algoritmust fejlesszek.

A tárgymanipulációhoz a robot és a kamera közötti transzformáció meghatározására, vagyis a kéz-szem kalibráció elvégzésére volt szükség.

A 3D pozíció meghatározására szolgáló algoritmussal már korábban foglalkoztam, ebben a beszámolóban a robot-PC interfész megvalósításáról, és a kéz-szem kalibrációról van szó.

2. A robotirányító rendszer felépítése

A robotlaborban a kommunikációs lánc a következő elemekből épül fel:

1. PC
2. PLC (Q03UDVCPU)
3. Robotvezérlő modul (Q172DRCPU)
4. Robotkar (Melfa RV-2F-Q)

A kommunikáció a következő módon valósul meg. A PC és PLC között a kommunikáció Modbus/TCP kapcsolaton keresztül történik, a PLC a belső hálózaton DHCP-vel kiosztott statikus IP-n elérhető. A Modbus/TCP egy Ethernet-alapú kommunikációs protokoll, amelyben a szerver elérhetővé teszi a hozzá csatlakozott kliensek számára a regisztereit, azok szabadon írhatók. A PLC és a robotvezérlő modul megosztott memórián keresztül kommunikálnak. A robotvezérlő közvetlenül irányítja a robotkart, azzal egy komplett berendezést alkot, a köztük lévő kommunikáció megoldása nem a felhasználó feladata.

A robot megfogójára egy Logitech c525 típusú webkamera van szerelve, ez szolgáltatja a képfeldolgozás számára a vizuális információt.

3. A PLC és robotprogram

A PLC-n két, a működéshez szükséges program fut. Az egyik egy Modbus szerver, ami a PC-vel való kommunikációt kezeli. Ez a Mitsubishi cég (és a programot korábban felfedező és használó hallgatók) jóvoltából már rendelkezésre állt számomra.

A másik program a robot felé történő kommunikációért felelős. Minthogy a PLC csak összekötő szerepet játszik a robot és a PC között, ezért ennek a programnak a feladatköre arra korlátozódik, hogy a Modbuson keresztül kapott értékeket továbbítsa a megosztott memóriába. Ezt a programot Menyhárt Balázs MSc hallgató készítette, én minimális módosításokkal az ő programját használom.

A roboton futó program kiolvassa a megosztott memóriából a megfelelő értékeket, és ennek megfelelően mozgatja a robotkart. Ennek a megvalósítása úgy történik, hogy egy-egy regiszterben kerül átadásra a pozíció-orientáció 6 paramétere. A robotprogram elvégzi a megfelelő korrekciókat (szög-radián átváltás), majd a megadott helyzetbe irányítja a megfogót. Ez ki van még egészítve egy utasításszámlálóval, amelynek értékét a PC-n futó programban vizsgálva megtudhatjuk, hogy az adott utasítást a robot elvégezte-e már.

4. A PC program

A PC program Python nyelven íródott. Képfeldolgozásra az OpenCV függvénykönyvtárat, a Modbus kommunikáció megvalósításához a PyModbusTCP könyvtárat használtam. A program feladata a robot felső szintű vezérlése, azaz célpozíciójának megadása, illetve a kamera kezelése. A legmagasabb szintű funkciója a kéz-szem kalibráció automatikus elvégzése. Az automatikus működésen túl meg lehet adni manuálisan is pozíciót, ahova a robotot mozgatni szeretnénk, valamint manuálisan is lehet képet készíteni a kamerával.

5. A gépi látás alapfogalmai

A kalibrációs algoritmus áttekintése előtt ismerkedjünk meg a gépi látásban, képfeldolgozásban használatos fogalmakkal. A kamerát úgynevezett pinhole-kameramodellel írjuk le. Ez a modell azt feltételezi, hogy a kép keletkezése olyan módon történik, hogy a háromdimenziós teret perspektivikusan vetítjük a képsíkra.

A kamerához definiálható egy kamera koordináta-rendszer. Ennek a koordináta-rendszernek az origójának a vetítés középpontját tekintjük, a z tengelye merőleges a vetítés síkjára. A kép koordináta-rendszer síkbeli koordináta-rendszer, a síkra vetített pontokat értelmezzük ebben. x és y tengelyei a kamera koordináta-rendszer tengelyeivel egyirányúak. Amennyiben homogén koordinátákkal jellemezzük a tér pontjait, mind a perspektív vetítés, mind a világ koordináta-rendszer és a kamera koordináta-rendszer közötti eltolás, forgatás mátrixszal kifejezhető:

$$w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{23} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{1}$$

A vetítés mátrixát kameramátrixnak hívjuk, a paramétereit pedig a kamera belső paramétereinek nevezzük. Mivel ezek a kamera rendeltetésszerű használata során nem változnak jelentősen, a kamera szerves részei, egyszeri kalibrálással meghatározhatók. A koordináta-rendszerek közti transzformáció paraméterei a külső paraméterek. Az összes paraméter meghatározható kellően nagy számú pontpár segítségével. A pontpárok alatt a világ koordináta-rendszerben ismert pozíciójú pontokat és azok kép koordináta-rendszerben megadott vetületét értjük. Az OpenCV-nek van olyan példakódja, amely elvégzi a kalibrációt, és kiszámítja a kérdéses mátrixokat.

A valóságban a pinhole kameramodell sajnos nem állja meg a helyét, a képet többféle torzítás terheli például abból adódóan, hogy nem vetítést alkalmazunk, hanem lencserendszert, valamint, hogy a detektoron elhelyezkedő pixelek gyártástechnológiája nem tökéletes. Ezeket a paramétereket most elhanyagoljuk, így a fenti összefüggésben a γ paraméter 0 lesz.

6. Külső paraméterek meghatározása, PnP probléma

A kéz-szem kalibráció során szükség van a kamera külső paramétereinek ismeretére. Azt a feladatot, amely ismert térbeli helyzetű tárgypontok és a hozzájuk tartozó képi pontok alapján próbálja meghatározni a kamera külső paramétereit PnP, azaz Perspective-n-Point problémának nevezzük.

Ennek egyik verziója a P3P algoritmus, amely csak 4 pontpárból számítja a paramétereket. Három pont kell ahhoz, hogy véges sok egyértelműen megoldást kapjunk (négy ponttal már túlhatározott lenne a feladat). A negyedik pontot az algoritmus arra használja fel, hogy a véges sok megoldás közül kiválassza a helyeset. Ezt úgy teszi meg, hogy a lehetséges négyféle megoldás közül azt választja, amelyikkel a negyedik pontot a képre vetítve a legkisebb hibát kapjuk a megadott képi ponthoz képest. Ezt aztán tetszőleges optimumkereséssel tovább lehet finomítani, pl. Levenberg-Marquardt eljárással. Ezt az eljárást használhatjuk akkor is, ha több, mint négy pont áll rendelkezése.

7. Kéz-szem kalibráció

A kéz-szem kalibráció során a robot megfogójának és a kamerának a relatív helyzetét kívánjuk meghatározni.

Az ehhez használt módszer a belső paraméterek kalibrálásánál alkalmazotthoz hasonló, egy kalibrációs objektumról képeket készítünk, amelyen megkeressük a tárgy ismert pozíciójú pontjait. Az így kapott összefüggésekből számítjuk először a kamera külső paramétereit, majd az egyéb ismert paraméterek felhasználásával a kamera relatív helyzetét.

A robotikában különböző koordináta-rendszereket használunk. A kalibráció során a következő koordináta-rendszereket használom:

- A robot bázis-koordinátarendszere, amelyik a nulladik, álló szegmenséhez van rögzítve. Betűjele: R (Robot)
- A megfogó vagy szerszám koordináta-rendszere, amely a robot utolsó szegmensének végén található. Betűjele: T (Tool)
- A kamera koordináta-rendszere. Ennek a koordináta-rendszernek az origóját a pinhole modell alapján a vetítési pontba képzeljük el, a z tengelye egybeesik az optikai tengellyel és a kamera nézési irányába mutat. A valóságban az optikai tengely és a lencse síkjának metszéspontjában található az origó. Betűjele: C (Camera)
- A tárgy-koordinátarendszer. Ez az a koordináta-rendszer, amiben a kalibráló objektum pontjainak koordinátáit értelmezzük. Nem kifejezetten lényeges, hogy hova tesszük, de az előbb említett koordinátákat numerikusan ismernünk kell ebben a koordináta-rendszerben. Betűjele: O(Object)

A B koordináta-rendszerből az A koordináta-rendszerbe való transzformáció a következőképpen írható fel homogén koordinátákban:

$$\mathbf{T_{AB}} = \left[\begin{array}{c|c} \mathbf{R_{AB}} & \mathbf{t_{AB}} \\ \hline \mathbf{0^T} & 1 \end{array}\right] \tag{2}$$

Itt $\mathbf{R_{AB}}$ 3x3-as rotációs mátrix, $\mathbf{t_{AB}}$ pedig oszlopvektor. A $\mathbf{t_{AB}}$ vektor az A koordináta-rendszer origójából a B koordináta-rendszer origójába mutat (A-ban felírva), míg $\mathbf{R_{AB}}$ oszlopai a B koordináta-rendszer egységvektorai szintén A-ban felírva. A jelöléstechnikából következik, hogy $\mathbf{T_{AB}}^{-1} = \mathbf{T_{BA}}$

A fent ismertetett koordináta-rendszerek közötti transzformációs lánc felírható:

$$\mathbf{T_{CO}T_{OR}T_{RT}T_{TC}} = \mathbf{I} \tag{3}$$

Az egyes transzformációknak speciális tulajdonságaik vannak, amit kalibráláskor ki lehet (sőt kell) használni. A kamera a végberendezésre szilárdan van rögzítve, nem mozdul el. A robot is rögzítve van a gépalaphoz, amihez képest a kalibráló tárgy nem mozog a kalibráció során. Ez azt jelenti, hogy a $\mathbf{T_{OR}}$ és $\mathbf{T_{TC}}$ mátrixok nem változnak, ha a robotkart mozgatjuk. Ezen felül ismert a robotkar pozíciója a bázis-koordinátarendszerben, hiszen a robot irányítórendszere megoldja a direkt geometriai feladatot, valamint a PnP algoritmus megadja a kamera helyzetét a tárgy-koordinátarendszerben, így $\mathbf{T_{CO}}$ és $\mathbf{T_{RT}}$ mátrixok ismertek mindenkor.

A kalibráció két lépésben történik. Az első lépésben a rotációs mátrixokat számítjuk ki, a másodikban a transzlációs vektorokat.

7.1. A rotációk becslése

A (3) egyenletet átalakítva kapjuk, hogy:

$$\mathbf{T_{OC}} = \mathbf{T_{OR}T_{RT}T_{TC}} \tag{4}$$

Ha az egész egyenletet beszorozzuk jobbról a nullvektorral:

$$\mathbf{T_{OC}} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{T_{OR}} \mathbf{T_{RT}} \mathbf{T_{TC}} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{5}$$

És elvégezzük a megfelelő szorzásokat:

$$\begin{bmatrix} \mathbf{t_{OC}} \\ 1 \end{bmatrix} = \mathbf{T_{OR}} \begin{bmatrix} \mathbf{R_{RT}} \mathbf{t_{TC}} + \mathbf{t_{RT}} \\ 1 \end{bmatrix} \tag{6}$$

Ebben az egyenletben számunkra ismeretlen a $\mathbf{T_{OR}}$ mátrix és a $\mathbf{t_{TC}}$ vektor. A kalibráció során készítünk N darab képet, amelyekhez meghatározzuk a külső paramétereket, vagyis a $\mathbf{T_{OC}}$ mátrixot. Ezeket a képeket úgy készítjük el, hogy az $\mathbf{R_{RT}}$ mátrix állandó, vagyis két kép készítése között nem forgatjuk a kamerát, csak eltoljuk. Ekkor kiegészíthető a fenti egyenlet egy indexszel:

$$\begin{bmatrix} \mathbf{v_{OCi}} \\ 1 \end{bmatrix} = \mathbf{T_{OR}} \begin{bmatrix} \mathbf{R_{RT}} \mathbf{t_{TC}} + \mathbf{t_{RTi}} \\ 1 \end{bmatrix} \quad i = 1..N \tag{7}$$

Azok az ismeretlenek illetve paraméterek, amik nem kaptak indexet, azok állandóak a képek készítése során. Vegyük a fenti N darab egyenlet átlagát:

$$\begin{bmatrix} \frac{\sum \mathbf{t_{OCi}}}{N} \\ 1 \end{bmatrix} = \mathbf{T_{OR}} \begin{bmatrix} \mathbf{R_{RT}} \mathbf{t_{TC}} + \frac{\sum \mathbf{t_{RTi}}}{N} \\ 1 \end{bmatrix} \tag{8}$$

Ha kivonjuk a két egyenletet egymásból, a következőt kapjuk:

$$\begin{bmatrix} \mathbf{t_{OCi}} - \frac{\sum \mathbf{t_{OCi}}}{N} \\ 0 \end{bmatrix} = \mathbf{T_{OR}} \begin{bmatrix} \mathbf{t_{RTi}} + \frac{\sum \mathbf{t_{RTi}}}{N} \\ 0 \end{bmatrix} \tag{9}$$

Bevezetjük, hogy:

$$\overline{\mathbf{t_{OCi}}} \coloneqq \mathbf{t_{OCi}} - \frac{\sum \mathbf{t_{OCi}}}{N}$$
$$\overline{\mathbf{t_{RTi}}} \coloneqq \mathbf{t_{RTi}} - \frac{\sum \mathbf{t_{RTi}}}{N}$$

Szemléletesen a felülvont változó az adott vektornak az összes vektor tömegközéppontjához képesti relatív helyzetét mutatja meg. Fontos látni, hogy ezek kiszámíthatók, ismerjük őket. (9) írható a következő alakban is:

$$\begin{bmatrix} \overline{\mathbf{t_{OCi}}} \\ 0 \end{bmatrix} = \left[\begin{array}{c|c} \mathbf{R_{OR}} & \mathbf{t_{OR}} \\ \hline \mathbf{0^T} & 1 \end{array}\right] \begin{bmatrix} \overline{\mathbf{t_{RTi}}} \\ 0 \end{bmatrix} \tag{10}$$

Ebből pedig:

$$\overline{\mathbf{t_{OCi}}} = \mathbf{R_{OR}} \overline{\mathbf{t_{RTi}}} \tag{11}$$

$\mathbf{R_{OR}}$ kiszámítása tehát ekvivalens az kapott vektorpárok közti optimális rotáció kiszámításával. Optimális alatt azt a rotációt értem, amelyikkel a vektorpárok megfelelő tagját elfordítva a pár másik tagjához képesti legkisebb négyzetes hibaösszeget kapjuk.

A Kabsch algortitmus pont ezt a problémát oldja meg, így én is ezt alkalmaztam. Az algoritmus a két vektorhalmaz közti koordinátánkénti kovariancia-mátrixot számítja ki, majd ennek SVD felbontásából határozza meg az optimális forgatást:

$$\mathbf{P} \coloneqq \begin{bmatrix} \vdots \\ \mathbf{t_{RTi}^T} \\ \vdots \end{bmatrix} \in \mathbb{R}^{N \times 3}, \quad \mathbf{Q} \coloneqq \begin{bmatrix} \vdots \\ \mathbf{t_{OCi}^T} \\ \vdots \end{bmatrix} \in \mathbb{R}^{N \times 3}, \quad i = 1..N \tag{12}$$

$$\mathbf{A} = \mathbf{P}^T\mathbf{Q} \tag{13}$$

$$\mathbf{U\Sigma V}^T = \mathbf{A} \tag{14}$$

$$\mathbf{R_{OR}} = \mathbf{U}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & det(\mathbf{UV}^T)\end{bmatrix}\mathbf{V}^T \tag{15}$$

Ezután $\mathbf{R_{TC}}$ számítása történik:

$$\mathbf{R_{TRi}R_{RO}R_{OCi}} = \mathbf{R_{TC}} \quad i = 1..N \tag{16}$$

A bal oldalon található rotációs mátrixok "átlagát" kellene kiszámítani, ezt is a Kabsch algoritmussal oldom meg. Az algoritmusnak pontpárokra van szüksége. Ebben az esetben $3N$ darab vektorpárunk lesz, minden átlagolandó rotációhoz 3-3 vektorpár. Egy ilyen párhármas egyik tagja a rotációs mátrix megfelelő oszlopvektora, a másik tagja pedig a neki megfelelő egységvektor ($[1\,0\,0]^T$, $[0\,1\,0]^T$, vagy $[0\,0\,1]^T$). Ekkor a (12)-ben definiált mátrixok a következő alakot öltik:

$$\mathbf{P} \coloneqq \begin{bmatrix}\vdots\\ \mathbf{R_{TRi}R_{RO}R_{OCi}}\\ \vdots\end{bmatrix}, \quad \mathbf{Q} \coloneqq \begin{bmatrix}\vdots\\ \mathbf{I}\\ \vdots\end{bmatrix}, \quad i = 1..N \tag{17}$$

ahol $\mathbf{I}$ a 3×3-as egységmátrix.

Így a rotációk átlagát véve a Kabsch algoritmussal megkaphatjuk az $\mathbf{R_{TC}}$ mátrixot.

7.2. A transzlációk becslése

Ha kifejtjük a (4) egyenletet (2) szerint:

$$\left[\begin{array}{c|c}\mathbf{R_{OC}} & \mathbf{t_{OC}}\\ \hline \mathbf{0^T} & 1\end{array}\right] = \left[\begin{array}{c|c}\mathbf{R_{OR}} & \mathbf{t_{OR}}\\ \hline \mathbf{0^T} & 1\end{array}\right]\left[\begin{array}{c|c}\mathbf{R_{RT}} & \mathbf{t_{RT}}\\ \hline \mathbf{0^T} & 1\end{array}\right]\left[\begin{array}{c|c}\mathbf{R_{TC}} & \mathbf{t_{TC}}\\ \hline \mathbf{0^T} & 1\end{array}\right] \tag{18}$$

Összeszorozva a jobb oldalt:

$$\left[\begin{array}{c|c}\mathbf{R_{OC}} & \mathbf{t_{OC}}\\ \hline \mathbf{0^T} & 1\end{array}\right] = \left[\begin{array}{c|c}\mathbf{R_{OR}R_{RT}R_{TC}} & \mathbf{R_{OR}R_{RT}t_{TC}} + \mathbf{R_{OR}t_{RT}} + \mathbf{t_{OR}}\\ \hline \mathbf{0^T} & 1\end{array}\right] \tag{19}$$

Ebben a fázisban a készített képek számát jelöljük M-mel. Ebből, ha bevezetjük az indexet a változó paraméterekhez , kapjuk, hogy:

$$\mathbf{t_{OCi}} = \mathbf{R_{OR}R_{RTi}t_{TC}} + \mathbf{R_{OR}t_{RTi}} + \mathbf{t_{OR}} \quad i = 1..M \tag{20}$$

Figyeljük meg, hogy most már a megfogó orientációját, azaz az $\mathbf{R_{RTi}}$ mátrixot is változtatjuk a képek készítése során. A fenti egyenletben az ismeretlenek a $\mathbf{t_{TC}}$ és $\mathbf{t_{OR}}$ vektorok. Átrendezve az egyenletet:

$$\mathbf{R_{OR}}^{-1}\mathbf{t_{OCi}} - \mathbf{t_{RTi}} = \mathbf{R_{RTi}t_{TC}} + \mathbf{R_{OR}}^{-1}\mathbf{t_{OR}} \tag{21}$$

Bevezetve:

$$\begin{aligned}\mathbf{p_i} &\coloneqq \mathbf{R_{OR}}^{-1}\mathbf{t_{OCi}} - \mathbf{t_{RTi}}\\ \mathbf{A_i} &\coloneqq \mathbf{R_{RTi}}\\ \mathbf{B} &\coloneqq \mathbf{R_{OR}}^{-1}\end{aligned} \tag{22}$$

Kapjuk, hogy:

$$\mathbf{p_i} = \mathbf{A_i t_{TC}} + \mathbf{B t_{OR}} \tag{23}$$

Ha az összes M egyenletet egybefoglaljuk:

$$\mathbf{Cx} = \mathbf{k} \tag{24}$$

ahol:

$$\mathbf{x} \coloneqq \begin{bmatrix} \mathbf{t_{TC}} \\ \mathbf{t_{OR}} \end{bmatrix}, \quad \mathbf{C} \coloneqq \begin{bmatrix} \vdots & \\ \mathbf{A_i} & \mathbf{B} \\ \vdots & \end{bmatrix}, \quad \mathbf{k} \coloneqq \begin{bmatrix} \vdots \\ \mathbf{p_i} \\ \vdots \end{bmatrix} \tag{25}$$

$$\mathbf{x} \in \mathbb{R}^{6\times 1}, \quad \mathbf{C} \in \mathbb{R}^{3M\times 6}, \quad \mathbf{k} \in \mathbb{R}^{3M\times 1}, \quad i = 1..M$$

Innen a pszeudoinverz segítségével az optimális megoldás számolható:

$$\hat{\mathbf{x}} = \left(\mathbf{C}^T\mathbf{C}\right)^{-1}\mathbf{C}^T\mathbf{k} \tag{26}$$

8. Eredmények

A kalibráló algoritmust a **??**. ábrán látható sakktáblamintával futtattam le. Az első lépésben azonos rotáció mellett 10-10 mérést végeztem minden tengely mentén elmozdítva a kamerát. A második fázisban a sakktáblára nagyjából $\pm 15°$-os szögből néztem rá, két tengely mentén mozgatva a kamerát.

A rotáció meghatározásának pontossága igen jó volt, viszont a megfogó és a kamera közti eltolást nem tudta meghatározni pontosan. A PnP algoritmus az objektum kamerától való távolságára nem kifejezetten érzékeny, nagy hibát tud véteni, ezért a kalibráló algoritmus is rossz adatokból dolgozik. A kamera oldalirányú elmozdulását szépen érzékeli a kalibráció, a z irányú elmozdulást viszont több 100%-os hibával tudja csak meghatározni.

9. Továbbfejlesztési lehetőségek

A kalibrálásra több módszer is rendelkezésre áll, valószínűleg fejlettebbek, mint az általam implementált, lehet, hogy azokkal kisebb hiba is megvalósítható.

A most használt algoritmus is javítható lenne, ha a kamerahelyzet-meghatározásnál különböző bizonytalanságok figyelembe vételével kiegészítenénk az algoritmust.

A nagy kalibrálási hiba okozója a kis látószög is lehet, ezt növelve talán nagyobb pontosság érhető el.

10. Forrásjegyzék

1. Gao, Xiao-Shan; Hou, Xiao-Rong; Tang, Jianliang; Cheng, Hang-Fei (2003). "Complete Solution Classification for the Perspective-Three-Point Problem"
2. Kabsch, W. (1976). A solution for the best rotation to relate two sets of vectors. Acta Crystallographica Section A, 32(5), pp.922-923.