Úklid pracovního prostoru

Aleš Trna, Minh Hoang Tran

14/12/2023

***Popis*— Předmětem této práce je popis řešení závěrečné semestrální úlohy z předmětu Robotika.**

***Klíčové pojmy*— Robot, manipulátor, počítačové vidění**

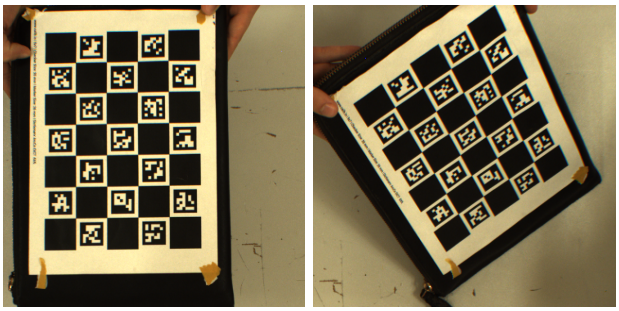
I. Zadání úlohy

V pracovním prostoru robota CRS97 jsou rozmístěny kostky o konstantní velikosti označené značkami typu *Aruco*.Robot je umístěn v kleci, která vymezuje jeho pracovní prostor. Ke kleci je na konstantním místě připevněna kamera, která zabírá stále stejnou scénu, ale ze zadání neznáme její přesnou pozici. Naším úkolem je roztřídit Kostky s *Aruco* značkami tak, aby kostky označené stejnými značkami byly uloženy na stejném místě.

II. Vidění

A. Kalibrace kamery

Abychom mohli detekovat, kde se kostky nacházejí, musíme provést kalibraci kamery. Kalibrací kamery zjistíme její distorzi, kterou poté budeme moci kompenzovat, abychom dokázali jednoznačně a přesně identifikovat polohu *Aruco* značky vůči kameře.
Princip kalibrace kamery spočívá v nafocení vzorků o známé velikosti a tvaru a porovnávání zkreslení výsledného obrázku s realitou. Pro jednoduchost se při této úloze obvykle používá Šacovnice o konstantním rozměru čtverce. Pro toto konkrétní řešení byla použita šachovnice s *Aruco* značkami na bílých polích o jiném, fixním rozměru díky které byla získána přesnější data. Pro co nejpřesnější odhad distorze a parametrů kamery je třeba nafotit co nejvíce vzorků z co nejvíce poloh vůči kameře.



Obrázek 1: Plochá a nakloněná *Charuco* deska

Na nafocených vzorcích byla detekována *Charuco* deska pomocí funkce *interpolateCornersCharuco* z knihovny *cv2*, díky které byla získána Jednoznačná poloha rohů šachovnice a *Aruco* značek. Tato data byla použita pro funkci *CalibrateCameraCharuco* ze stejné knihovny. Výsledkem byla matice kamery

$$\mathbb{C} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{1}$$

Kde:

- f_x, f_y jsou ohniskové vzdálenosti v konkrétních osách
- c_x, c_y jsou optické středy

Zároveň byly získány distorční koeficienty. Díky kterým můžeme kompenzovat distorzi kamery.

B. Hand to Eye kalibrace

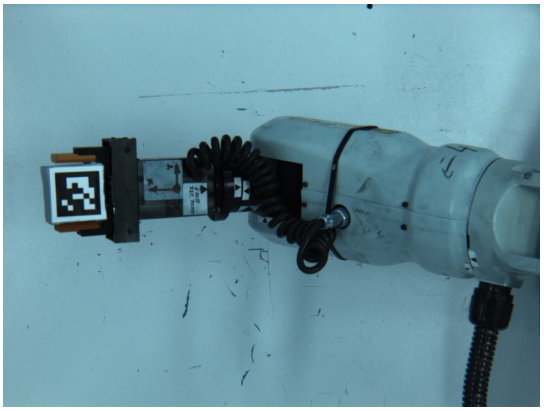
Ze zadání úlohy není jasné, kde přesně se kamera vůči robotovi nachází. Je několik způsobů, jak to zjistit. Nejjednodušší je změřit polohu kamery vůči pevně zvolenému bodu. Tato metoda je ale pro naše použití příliš nepřesná. Dalším způsobem je vytvoření homografické matice, která by nám mapovala scénu kamery do roviny. Předmětem této práce je však řešení nerovninného problému, proto tento způsob není vhodný. Protože máme již zkalibrovaný obraz z kamery dokážeme díky funkcím knihovny *cv2* zjistit polohu objektů. Díky senzorům v robotu známe přesnou polohu chapadla vůči základně robota. V našem konkrétním případě není kamera připevněna na chapadle robota, nýbrž na konstrukci v konstantní poloze vůči robotu. Řešíme tedy ”Hand to Eye” problém. Na jeho řešení můžeme využít opět knihovnu *cv2*, konkrétně funkci *CalibrateRobotWorldHandEye*. Do chapadla robota vložíme objekt o známých rozměrech, u nějž můžeme zjistit jeho polohů vůči kameře. V našem případě využijeme kostku s *Aruco* značkou. S robotem budeme jezdit v zorném poli kamery a zaznamenávat scénu pomocí kamery a k ní korespondující polohu chapadla vůči základně robota. Poté na fotkách detekujeme polohu *Aruco* značek a data vložíme do zmíněné funkce. Řešíme tedy rovnici

$$\mathbb{A}_i\mathbb{X} = \mathbb{Y}\mathbb{B}_i \tag{2}$$

Kde:

- $\mathbb{A}_i$ je i-tá naměřená transformace chapadla vůči bázi
- $\mathbb{X}$ je konstantní transformace značky vůči chapadlu robota

- $\mathbb{B}_i$ je i-tá odhadnutá transformace značky vůči kameře
- $\mathbb{Y}$ je konstantní transformace kamery vůči bázi robota.



Obrázek 2: Kostka s *Aruco* značkou uchopená (s konstantní transformací) v chapadle robota

Hledáme tedy transformaci $\mathbb{Y}$, pro její výpočet použijeme výše zmíněnou funkci.

C. Korekce transformace

Po zjištění polohy kamery vůči základně robota můžeme zjistit polohu kostek vůči robotu. využijeme k tomu rovnici

$$T_{B\to K} = T_{B\to C} \cdot T_{C\to K} \tag{3}$$

Kde:

- $T_{B\to K}$ je transformace kostky vůči bázi robota
- $T_{B\to C}$ je transformace kamery vůči bázi robota
- $T_{C\to K}$ je transformace kostky vůči kameře
- Násobení tranformací odpovídá aplikování levé tranformace na transformaci pravou

Transformaci $T_{C\to K}$ získáme pomocí *cv2.aruco* funkcí *detectMarkers* a *estimatePoseSingleMarkers*. Jako prvotní test přesnosti kalibrace porovnáme změřenou a spočtenou polohu kostek vůči bázi robota. V našem případě jsme zjistili, že robot

III. Návrh řešení

TODO:{ Program je navrhnut jako stavový automat s následujícími stavy:

A. Inicializační stav

B. Třídění podle vrstev

}