



Robotika: Kalibrace

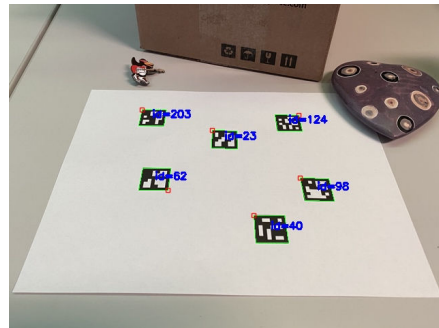
Vladimír Petřík

vladimir.petrík@cvut.cz

13.10.2025

Co víme¹

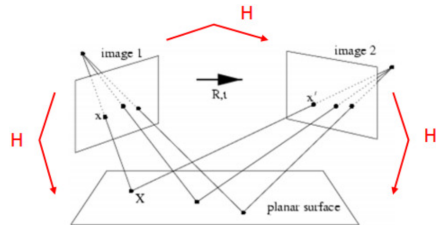
- ▶ Jak detekovat objekty v obraze
 - ▶ Šachovnice
 - ▶ Aruco markery
- ▶ Jak odhadnout homografii
 - ▶ Mapuje body mezi rovinami
 - ▶ Odhaduje se z korespondencí
 - ▶ Robustní - nevyžaduje odhad hloubky
- ▶ Kalibrace vnitřních parametrů kamery
 - ▶ Použijeme obrázky šachovnice
 - ▶ Odhadneme matici kamery K
- ▶ Odhad polohy
 - ▶ Odhaduje polohu kamery k objektu
 - ▶ Používá algoritmus PnP
 - ▶ Vyžaduje K a korespondence



¹Obrázky z docs.opencv.org

Co víme¹

- ▶ Jak detekovat objekty v obraze
 - ▶ Šachovnice
 - ▶ Aruco markery
- ▶ Jak odhadnout homografii
 - ▶ Mapuje body mezi rovinami
 - ▶ Odhaduje se z korespondencí
 - ▶ Robustní - nevyžaduje odhad hloubky
- ▶ Kalibrace vnitřních parametrů kamery
 - ▶ Použijeme obrázky šachovnice
 - ▶ Odhadneme matici kamery K
- ▶ Odhad polohy
 - ▶ Odhaduje polohu kamery k objektu
 - ▶ Používá algoritmus PnP
 - ▶ Vyžaduje K a korespondence

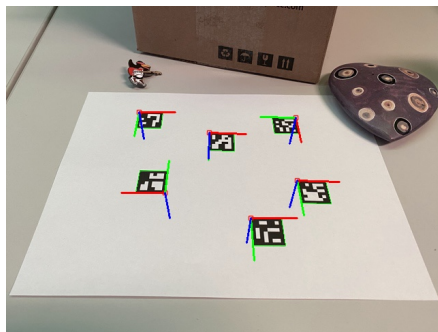
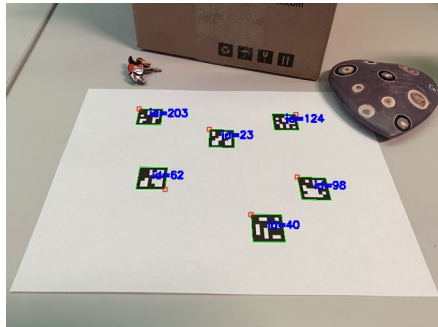


¹Obrázky z docs.opencv.org

Co víme¹

- ▶ Jak detekovat objekty v obraze
 - ▶ Šachovnice
 - ▶ Aruco markery
- ▶ Jak odhadnout homografii
 - ▶ Mapuje body mezi rovinami
 - ▶ Odhaduje se z korespondencí
 - ▶ Robustní - nevyžaduje odhad hloubky
- ▶ Kalibrace vnitřních parametrů kamery
 - ▶ Použijeme obrázky šachovnice
 - ▶ Odhadneme matici kamery K
- ▶ Odhad polohy
 - ▶ Odhaduje polohu kamery k objektu
 - ▶ Používá algoritmus PnP
 - ▶ Vyžaduje K a korespondence

¹Obrázky z docs.opencv.org



Kde je robot?

- ▶ Uvažujeme homografii - jak získáme 3D body v souřadnicovém systému robota?
 1. Ručně změříme pozice Aruco markerů vzhledem k základně
 - ▶ Umístíme Aruco markery do roviny, která nás zajímá
 - ▶ Může být obtížné změřit přesně
 2. Ručně umístíme robota ke kalibračnímu objektu
 - ▶ Kombinujeme známé 3D body kalibračního objektu s FK
 3. Použijeme Aruco marker uchopený robotem
 - ▶ Použijeme FK k výpočtu pozice značky v souřadnicovém systému robota
 - ▶ Musíme se pohybovat v rovině, která nás zajímá
 - ▶ Nebudeme mít přesný odhad značky vzhledem k ruce
- ▶ Omezeno na mapování roviny na rovinu



Kalibrace polohy/orientace kamery

- ▶ Kalibrace oko-ruka
- ▶ Řešíme $A^i X = Y B^i$
 - ▶ Měření: $A^i, B^i \in SE(3)$
 - ▶ Odhadnuté parametry: $X, Y \in SE(3)$
- ▶ Kamera může být namontována na
 - ▶ Základně robota (kalibrace oko-ruka, eye-to-hand)
 - ▶ Chapadle (kalibrace oko-v-ruce, eye-in-hand)
- ▶ Kalibrace oko-ruka, eye-to-hand
 - ▶ $A^i = T_{RG}^i$
 - ▶ $B^i = T_{CT}^i$
 - ▶ $X = T_{GT}$
 - ▶ $Y = T_{RC}$
- ▶ Kalibrace oko-v-ruce, eye-in-hand
 - ▶ $A^i = T_{CT}^i$
 - ▶ $B^i = T_{GR}^i$
 - ▶ $X = T_{TR}$
 - ▶ $Y = T_{CG}$



Kód pro kalibraci $AX = YB$

```
def solve_AX_YB(a: list[SE3], b: list[SE3]) -> tuple[SE3, SE3]:  
    """Solve  $A^iX=YB^i$ , return  $X, Y$ """  
  
    rvec_a = [T.rotation.log() for T in a]  
    tvec_a = [T.translation for T in a]  
    rvec_b = [T.rotation.log() for T in b]  
    tvec_b = [T.translation for T in b]  
  
    Rx, tx, Ry, ty = cv2.calibrateRobotWorldHandEye(rvec_a, tvec_a, rvec_b, tvec_b)  
    return SE3(tx[:, 0], S03(Rx)), SE3(ty[:, 0], S03(Ry))
```



Vliv velikosti Aruco markerů

- ▶ Odhad hloubky malých Aruco markerů je obtížný
 - ▶ Velké chyby vedou k nepřesné kalibraci
 - ▶ Velké markery omezují pohyb
- ▶ Řešení: minimalizace reprojekčních chyb
 - ▶ Původní měření jsou v pixelech
 - ▶ Můžeme provést jemnou optimalizaci v prostoru pixelů
 - ▶ Reprojekční chyba:
 - ▶ $X^*, Y^* = \arg \min_{X, Y} (\sum_i \mathbf{u}_i - \pi(\mathbf{x}_i(X, Y)))$
 - ▶ $\mathbf{u}_i$ - detekovaný 2D bod, $\mathbf{x}_i$ - 3D bod v souřadnicovém systému kamery, π - projekce
 - ▶ $\mathbf{x}_i$ je spočteno pomocí FK a poloh X a Y
 - ▶ Počáteční odhad je z kalibrace oko-ruka



Vliv kinematiky robota

- ▶ Kinematika robota také není perfektní
 1. Různé konfigurace robota mají různé chyby
 2. Často chyby v offsetech kloubů
- ▶ Řešení #1: omezit se na podmnožinu konfigurací
 - ▶ Při řešení IK používat pouze podobné konfigurace
 - ▶ Pro kalibraci ale také pro provoz robota!
- ▶ Řešení #2: kalibrovat offsety kloubů
 - ▶ Stejně jako při minimalizaci reprojekční chyby
 - ▶ $X^*, Y^*, \theta_{\text{offset}}^* = \arg \min_{X, Y, \theta_{\text{offset}}^*} (\sum_i \mathbf{u}_i - \pi(\mathbf{x}_i(X, Y, \theta_{\text{offset}}^*)))$
 - ▶ FK je upraveno tak, aby zahrnovalo offsety kloubů
- ▶ Kalibrace není jednorázový úkol
 - ▶ Kalibrovat robota před každým důležitým úkolem
 - ▶ Kalibrovat robota po každé významné změně



- ▶ Kalibrace závisí na úkolu
 - ▶ Homografie
 - ▶ Vnitřní parametry kamery
 - ▶ Poloha kamery
 - ▶ Kinematika robota
- ▶ Kalibrace není jednorázový úkol
 - ▶ Dobře automatizovat

Cvičení

- ▶ Cvičení na KN
- ▶ Detekce obruče a homografie

