

Razpoznavanje in lokalizacija objektov in robota v prostoru

Erik Pleško

*Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška 25, 1000 Ljubljana, Slovenija
E-pošta: ep9060@student.uni-lj.si*

Povzetek. Pri seminarju za predmet robotski vid sem se ukvarjal z problemom lokalizacije objektov in robota v prostoru z uporabo kamere na robotu. Implementiral sem lokalizacijo ArUco markerjev, s katerimi lahko označimo željene objekte ali jih uporabimo kot referenčne točke za lokalizacijo robota. Ukvarjal sem se z implementacijo vizualne odometrije kot dopolnilo, ko podatek o markerjih ni na voljo.

Ključne besede: openCV, ArUco, lokalizacija v prostoru, vizualna odometrija

Localization of objects and a robot in 3D space

As a seminar for subject Robotic vision I addressed a problem of localization of objects and a robot in 3D space using a camera mounted on the robot. I implemented detection of ArUco markers that can then be used to mark desired objects or reference points for the robot. I enhanced the solution with visual odometry that could be used in case there is no markers detectable by the camera.

1 UVOD

1.1 Naloga

Cilj seminarske naloge je pozicionirati robota Kuka YouBot v prostoru z uporabo kamere nameščene na robotu. Pri tem si bom pomagal z ArUco markerji ter vizualno odometrijo.

V začetku semestra se je fakulteta zaradi pandemije zaprla in seminar sem zato izvedel brez uporabe robota. Zato sem se ukvarjal le z pozicioniranjem kamere v prostoru. Pri delu sem porabil kamero prenosnega računalnika in kamero na mobilnem telefonu.

1.2 Uporaba programskega jezika Python in knjižnice OpenCV

Za opravljanje naloge sem uporabil programski jezik Python. Pri delu sem si pomagal z funkcijami iz knjižnice OpenCV. Za delo z ArUco markerji sem uporabil modul ArUco [1]. Za kalibracijo kamere in 3D rekonstrukcijo pa predvsem funkcije opisane v [2].

2 METODOLOGIJA

S kalibrirano kamero lahko zaznamo ArUco marker in določimo njegovo lego glede na kamero. To nam polnoma zadostuje za pozicioniranje kamere v prostoru. Da pa bi sistem naredili nekoliko bolj robusten, poskusim pozicioniranju dodati tudi vizualno odometrijo.

Korespondenčne točke, ki jih bomo za to uporabljali bomo dobili s pomočjo Lukas-Kanade sledenja izbranim točkam.

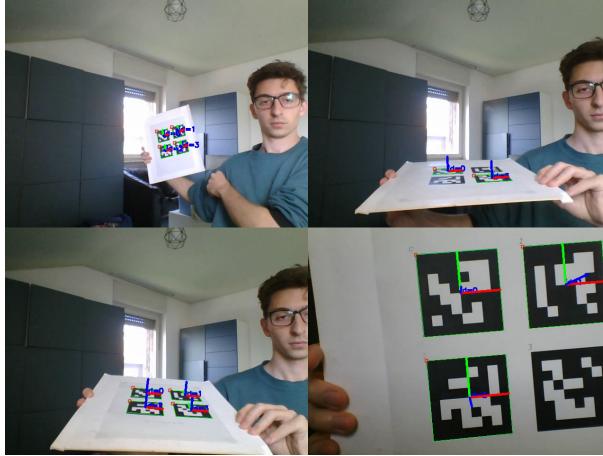
2.1 Kalibracija kamere

Z uporabo funkcij iz knjižnice OpenCV sem napisal skripto za kalibracijo kamere. Na kratko opisem postopek. Kalibracijo sem izvajal s šahovnico. Na zajetih slikah ali videu sem s funkcijo findChessboardCorners poiskal točke šahovnice ter takšne slike shranil. Pred iskanjem šahovnice sem z izračunom variance laplace-vega operatorja preveril, da slika ni preveč zamegljena zaradi premikanja. Iz zaznanih točk ogljišč sem s funkcijo calibrateCamera pridobil podatke kalibracije ter jih shranil v datoteko, ki jo ob vsakem uporabljanju kamere naložim v program.

2.2 AruCo markerji

Za iskanje markerjev na zajeti sliki sem uporabil funkcijo detectMarkers. Funkcija vrne ogljišča detektiranih markerjev, ki jih nadalje uporabljam. Markerje na sliki narišemo s funkcijo drawDetectedMarkers. S funkcijo estimatePoseSingleMarkers pa iz najdenih ogljišč določimo lego markerjev v koordinatnem sistemu kamere. Kordinatni sistem je postavljen tako: os x kaže od leve proti desni, os y kaže navzdol, os z pa v sliko. S funkcijo drawAxis lahko vrišemo koordinatne sisteme vsakega markerja. Barve rdeča, zelena in modra (RGB) označujejo osi x, y in z. Na sliki 1 prikažem nekaj zaznanih markerjev v različnih legah.

Za približen preizkus, kako natančno zaznamo lego markerjev, sem izbral koordinatni sistem markerja 0 kot globalni koordinatni sistem, nato pa preračunal kje se ostali markerji nahajajo glede na globalni sistem. Rezultate za nekaj leg markerjev prikazujem na sliki 2. Markerje sem natisnil na list papirja na znani razdalji ter list nalepil na kos trdega kartona, zato predpostavljam, da ležijo v ravni. Stranica markerja meri 70 mm,



Slika 1: Detekcija ArUco markerjev. Vsak detektirani marker je označen z obrisom, ki povezuje ogljišča, koordinatnim sistemom postavljenim v središče markerja ter z id oznako markerja. Levi sliki prikazujeta uspešno zaznane vse markerje. Desni sliki prikazujeta primera, ko nekateri markerji niso uspešno zaznani. Do tega pride, če del markerja ni viden ali če je marker viden pod premajnim kotom.

razdalje med njihovimi središči pa so 91 mm v x smeri in 90 mm v z smeri.

2.3 Lucas-Kanade sledenje

Preden začnemo sledenju točk, s funkcijo goodFeaturesToTrack določimo nabor točk, ki jim bomo sledili. Točke, ki jim ne sledimo uspešno zavrzemo. Primer korespondenčnih točk pridobljenih na takšen način je prikazan na sliki 3.

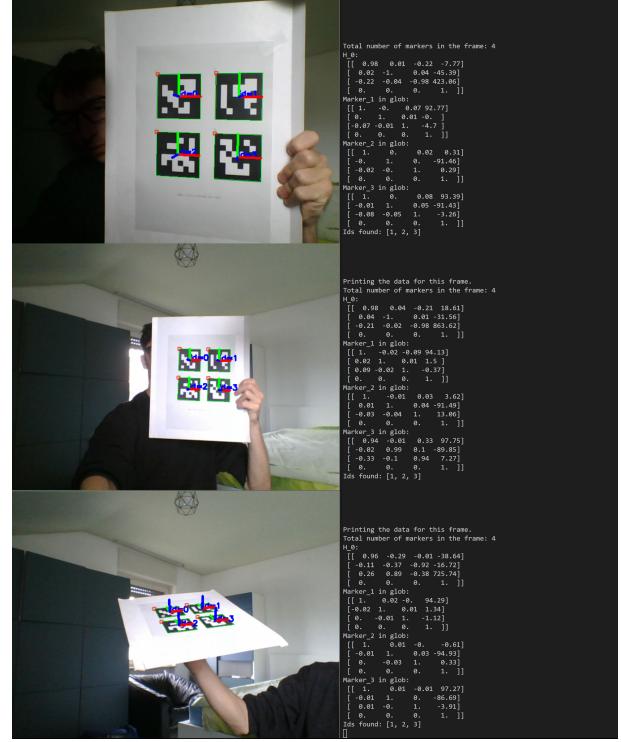
2.4 Vizualna odometrija

V tem podoglavlju bom opisal delo na temo vizualne odometrije. Na dveh zajetih slikah poznamo korespondenčne točke iz katerih želimo določiti relativni premik kamere med zajemom slik. ArUco markerje v tem poglavju uporabljam kot referenčno meritev lege kamere za oceno delovanja odometrije.

2.4.1 Ocena lega na podlagi umetno ustvarjenih točk:
Za preizkus delovanja in razumevanje uporabljenih funkcij sem najprej generiral približno 100 3D točk ter jih s funkcijo projectPoints projiciral na 2D sliko kamere. Iste točke sem projiciral na sliki dveh kamer, ki sta razmaknjeni v prostoru, nato pa poskusil iz točk na slikah dobiti informacijo relativni legi kamer. Postavitev prikažem na sliki 2.4.1.

Relativno lego med kamerama sem določil preko esencialne matrike, ki sem jo dobil s funkcijo findEssentialMat. Nato pa sem lego določil s funkcijo recoverPose.

S pogledom na sliko 2.4.1 se zdi da ocenjevanje lege vsaj na precej idealnih podatkih o točkah deluje popolno. A za simulacijo situacije, ki jo bomo srečali v realnem primeru sem program še nekoliko nadgradil. Na sliki 5 prikazujem podobno simulacijo. Simuliram uporabo



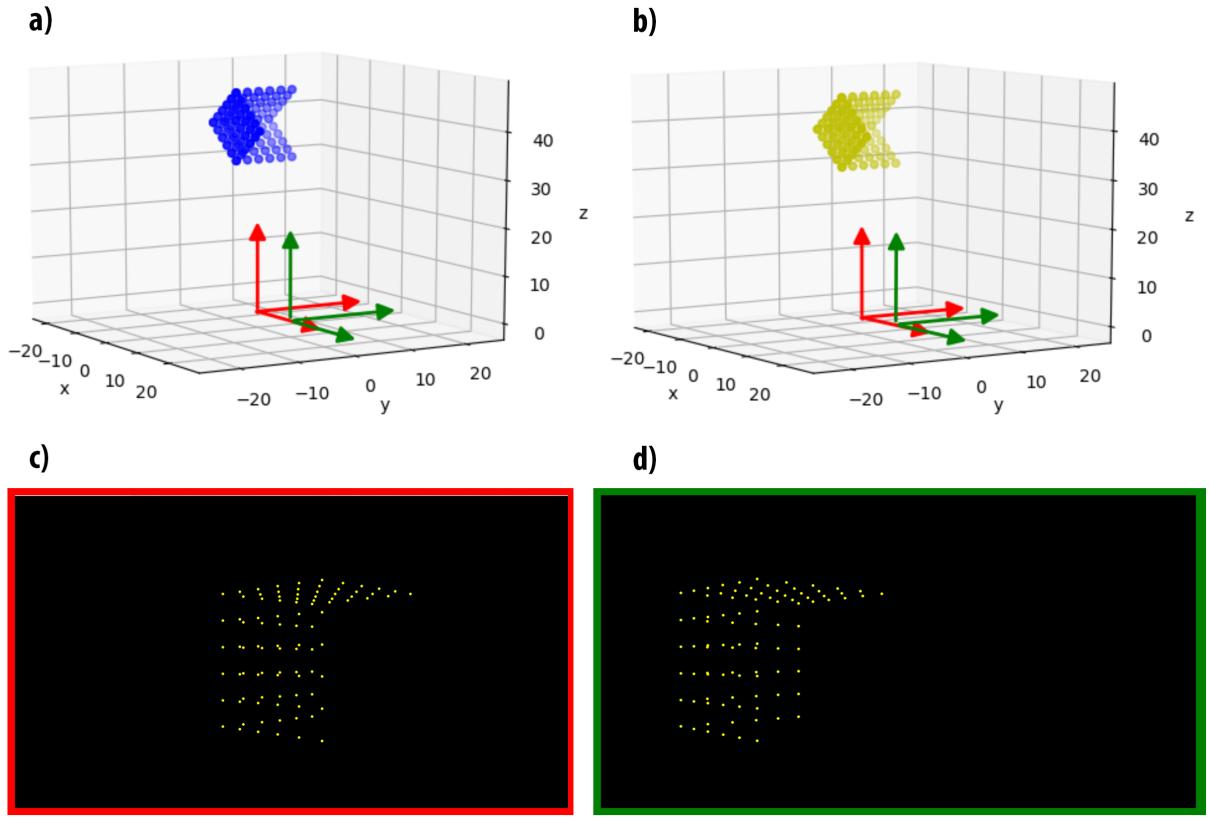
Slika 2: Nekaj leg markerjev ter relativna lega markerjev v koordinatnem sistemu markerja z id 0 (levi zgornji marker). Ob vsaki sliki navedem 4 homogene transformacijske matrike. Najprej lega markerja 0 v sistemu kamere, nato pa lega ostalih markerjev v globalnem sistemu (sistem markerja 0). Rotacijski del matrike bi moral biti identiteta, pozicijski vektor pa približno 90 (enote so milimetri) v ustreznih x in y smereh ter 0 v smeri z.



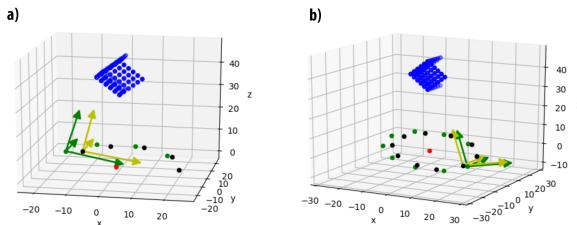
Slika 3: Na slikah označene korespondenčne točke pridobljene s sledenjem začetno izbranim točkam. (desno) Začetna slika. (levo) slika po 10 sekundah videa (približno 250 sličic).

ene kamere, ki se giblje po prostoru in zajame 10 slik. Vsako sliko primerjam s prvo in izračunal relativno lego kamere glede na njen začetno lego.

Na sliki 5 vidimo, da ocena relativne lege kamere že na praktično idealnih točkah ni zelo dobra. Razloga za to ne poznam. Sem pa opazil, da projekcija 3D točk v ravnino slike ni v vseh primerih delovala, kot sem pričakoval, a sem se odločil, da zaradi časovne omejenosti seminarja nadaljujem z preizkusom postopka na realnih posnetkih.

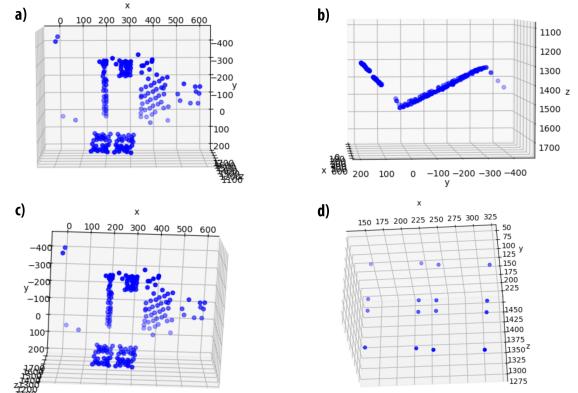


Slika 4: Umetno ustvarjene točke v obliki dela kocke. Rdeč in zelen koordinatni sistem prikazujejo postavitev kamer. a) Začetna postavitev kamer ter točk v 3D prostoru. b) Rekonstruirana postavitev kamer ter točk. Skalo sem umeril z izračunom razdalje med enim parom sosednjih dveh točk. c) Slika točk na rdeči kameri. d) Slika točk na zeleni kameri.



Slika 5: Simulacija premikanja kamere po prostoru. Modro so umetno določene točke. Zelen koordinatni sistem prikazuje trenutno referenčno lego kamere glede na katero so izračunane projekcije točk na sliko kamere. Zelene točke prikazujejo pretekle lege referenčne kamere. Rumen koordinatni sistem prikazuje ocenjeno lego kamere, črne točke pa pretekle ocenjene lege kamere. a) in b) 3D prikaz ob različnih časih.

2.4.2 Ocena lega realne kamere: Pri oceni relativne lege kamere sem uporabil korespondenčne točke pridobljene z uporabo Lucas-Kanade postopka opisanega v 2.3. Scena in sliki iz kamere sta vidni na sliki 7 a) in b). Uporabil sem dve sliki in iz njiju določil relativno lego kamere ter 3D rekonstrukcijo korespondenčnih točk. 3D rekonstrukcija je prikazana na sliki 6.



Slika 6: Rekonstrukcija 3D točk iz slike 7. a) - c) Rekonstruirane točke iz različnih pogledov. d) Rekonstruirane točke oglišč markerjev

3 REZULTATI

Pozicioniranje kamere z ArUco je bil najboljši način, ki mi je bil na voljo. Zato sem na ta način dobljeni podatek uporabil kot referenco pri vrednotenju vizualne odometrije. Podatke o markerjih sem uporabil tudi za

določanje skale. Uporabil sem 3D rekonstrukcijo točk ogljič markerjev, prikazano na sliki 6 c), ter izračunal dolžine stranic markerjev. Njihovo povprečje sem uporabil za skaliranje. Brez ustreznega skaliranja imamo samo podatek o smeri premika kamere.

Na sliki 7 c) prikazujem rezultat postopka.

a)



b)



c)

```
Cam1_inM0 (camera1 in M0 (global CS)):
[[ 0.995  0.049 -0.092 -58.793]
 [ 0.104 -0.516  0.85 -1150.898]
 [-0.006 -0.855 -0.518  820.349]
 [ 0.       0.       0.       1.   ]]

Cam2_inM0 (camera2 in M0 (global CS)):
[[ 1.      -0.001 -0.006 422.514]
 [ 0.005  -0.5    0.866 -1072.381]
 [-0.004 -0.866 -0.5    770.655]
 [ 0.       0.       0.       1.   ]]

*** H_markers (Cam2_inCam1 - camera2 in camera1): ***
[[ 0.995 -0.048  0.087 487.131]
 [ 0.05   0.999 -0.019 25.538]
 [-0.086  0.023  0.996 48.394]
 [ 0.       0.       0.       1.   ]]

H (estimated) after scale:
[[ 0.997 -0.039  0.068 515.604]
 [ 0.041  0.999 -0.019 1.808]
 [-0.067  0.022  0.997 75.539]
 [ 0.       0.       0.       1.   ]]

H - H_markers:
[[ 0.002  0.009 -0.018 28.473]
 [-0.009  0.       -0.       -23.729]
 [ 0.018 -0.001  0.001 27.145]
 [ 0.       0.       0.       1.   ]]
```

Slika 7: Z modro so označena ogljišča markerjev, z zeleno pa ostale korespondenčne točke. a) Slika kamere v začetnem položaju. 3D rekonstrukcija točk na sliki 6 je izražena v koordinatnem sistemu te kamere. b) Slika kamere v končnem položaju. c) Rezultat vizualne odometrije. H_{markers} podaja relativno lego 2. kamere glede na prvo določeno z markerji. H_e podaja ocenjeno lego z odometrijo. Na dnu pa imamo izračeno razliko.

Če kvalitativno ovrednotimo rezultat iz slike 7 c), lahko rečemo, da je ocena solidna. Pri premiku kamere za približno pol metra naredimo okoli 30 mm napake, kar je dobrih 5 %. Pri trenutni rešitvi je to primer najboljšega delovanja. Pri drugačnih izbirah slik in postavitev kamere rezultati niso vedno tako dobro. Odstopanja so večja, včasih je tudi 3D rekonstrukcija vidno popačena.

4 DISKUSIJA

Implementiral sem uporabo ArUco markerjev za določanje lege glede na markerje. To se lahko uporabi za določanje lege markerja pri fiksni kameri, določanja lege kamere pri fiksni ali poznani legi markerje ali pa relativno lego kamere glede na marker, v za šalo izmišljenem primeru, ko robot s kamero lovi bežeči marker.

Nato sem dodal uporabo Lucas-Kanade postopka za sledenje izbranim točkam in ga uporabil za določanje korespondenčnih točk na sceni, ko se kamera premika.

Točke sem uporabil za uresničenje preprostega in primitivnega načina določanja relativne lege kamere med dvema zajetima slikama. Trenutni postopek bi si želel širše preizkusiti in analizirati njegovo delovanje v ugodnih in slabih razmerah. Postopek pa se zdaleč ne poslužuje vseh informacij, ki so nam v zajetem videu na voljo in tako je tukaj še zelo veliko prostora za nadgradnjo.

4.1 Možne izboljšave

Najprej bi si želel delovanje postopkov bolje ovrednotiti. To velja za vrednotenje ocenjevanja lege ArUco markerjev in vizualne odometrije. Zanimiva bi bila uporaba robotske roke za premikanje kamere v prostoru.

Pri ocenjevanju lege z odometrijo bi si bilo zelo smiselnega pomagati z vso zajeto informacijo o prostoru. To pomeni, da bi vse zajete slike uporabil za rekonstrukcijo 3D prostora in določanja lege kamere v tistem trenutku.

LITERATURA

- [1] Dokumentacija openCV ArUco https://docs.opencv.org/4.2.0/d9/d6a/group_aruco.html
- [2] Dokumentacija openCV kalibracija in rekonstrukcija https://docs.opencv.org/4.2.0/d9/d0c/group_calib3d.html#ga69f2545a8b62a6b0fc2ee060dc30559d
- [3] R. Szelitzki: Computer Vision - Algorithms and Applications. Springer London (2011)

Pleško Erik je leta 2019 diplomiral iz fizike na Fakulteti za matematiko in fiziko na Univerzi v Ljubljani. Je študent prvega letnika robotike na Fakulteti za Elektrotehniko na Univerzi v Ljubljani.