

Uporaba računalniškega vida pri robotskem prijemanju

Robert Logar
ABB d.o.o.
Toplarniška 19, 1000 Ljubljana
robert.logar@si.abb.com

Robot grasping with computer vision

The article addresses a topic in robotics, which has been a subject of continuous interest during the last three decades, but has been evolving more rapidly during the last ten years. Vision guided robotics (visual servoing) and an application of such a system is presented and discussed.

1 Uvod

Velika večina instaliranih robotov deluje v proizvodnji, kjer lahko delovno okolje prilagodimo robotu. V aplikacijah, kjer se okolja in pozicioniranja objektov ne da tako natančno zagotoviti, se roboti niso razširili, razlog za to pa je v pomanjkanju senzorskih sposobnosti komercialnih robotov. Eden izmed uporabnih robotskih senzorjev je vid, ki omogoča brez kontaktno merjenje okolice.

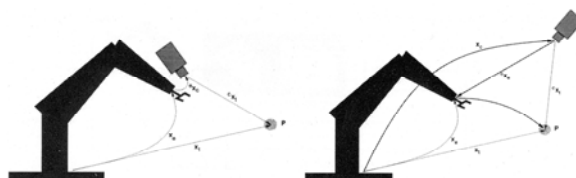
2 Vodenje z računalniškim vidom

Danes so na voljo robotski krmilniki različnih proizvajalcev s povsem integriranimi video sistemi, kjer sta operaciji vizualnega zaznavanja in manipulacije povezani odprtoznančno, saj kamera najprej »pogleda«, robot pa nato opravi gib. Natančnost vodenja je tako direktno pogojena z natančnostjo video sensorja in robotskega manipulatorja. Za izboljšanje natančnosti obeh sistemov se ponuja uporaba regulacijske zanke z video povratno informacijo, ki bi zmanjšala napako celotnega sistema. Če idejo pripeljemo do skrajnosti, lahko računalniški vid predstavlja povratno informacijo v regulaciji lege robotskega manipulatorja.

2.1 Postavitev kamere

Sistemi vodenja z računalniškim vidom ponavadi uporabljajo eno izmed naslednjih dveh postavitev kamere:

- pritrjeno na robotsko zapestje (orodje, prijemalo)
- fiksno postavljeno v delovnem prostoru



Slika 1: Različni postavitvi kamere

V prvem primeru, ki mu ponavadi pravijo »eye-in-hand« konfiguracija, je kamera pritrjena na zapestje robotskega manipulatorja. Tu obstaja poznana, ponavadi konstantna lega kamere glede na robotsko prijemalo.

V drugem primeru je kamera fiksno pritrjena v delovnem prostoru. V taki konfiguraciji je slika objekta, ki ga vidi kamera, neodvisna od gibanja robota, razen v primeru, ko je ciljni objekt robotsko prijemalo samo. Izpeljanka te konfiguracije, je primer, ko je kamera gibljiva, morda na drugem robotu ali pa se samo nagiba, tako, da opazuje vodnega robota s primernejšega zornega kota.

2.2 Arhitekture vodenja

Pri razvrstitvi sistemov vodenja z računalniškim vidom moramo najprej odgovoriti na naslednji vprašanji:

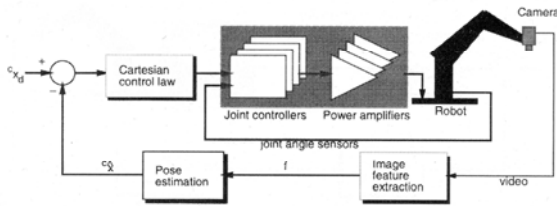
1. Je struktura vodenja hierarhična in video sistem zagotavlja nastavitvene točke kot vhode za regulator(je) sklepov ali video

sistem direktno računa vhode za regulator(je) sklepov?

2. Je signal napake definiran v 3D (prostor naloge) koordinatah ali direktno iz lastnosti zajete slike?

Če je sistem vodenja hierarhičen in video sistem nastavlja vhodne vrednosti za regulator sklepov, povratno informacijo pa predstavljajo položaji posameznih sklepov, potem tak sistem imenujemo dinamični »poglej in premakni« sistem (ang.: *dynamic look-and-move system*).

V nasprotju pa direktno vodenje z računalniškim vidom (ang.: *direct visual servo*), povsem eliminira robotski krmilnik, le-tega nadomesti video krmilnik, ki računa vhode (kote, hitrosti, pospeške) za posamezne sklepe in tako uporablja samo vizualno zaznavanje za stabilizacijo robotskega mehanizma.



Slika 2: Dinamična »poglej-in-premakni« struktura na poziciji osnovanega vodenja.

Zaradi večih razlogov so skoraj vsi implementirani sistemi posvojili dinamični »poglej-in-premakni« pristop. Prva težava nastopi vsled dokaj nizkih frekvenc vzorčenja s strani video sistemov, kar dela regulacijo robotskega mehanizma s kompleksno, nelinearno dinamiko, izredno zahteven problem. Ob uporabi interne povratne informacije z visokimi frekvencami vzorčenja, pa se robotski manipulator s stališča video sistema obnaša kot sistem z idealno dinamiko. Naslednja prednost je ta, da princip »poglej-in-premakni« loči kinematične singularnosti mehanizma od video sistema, zaradi česar ga lahko le-ta obravnava kot idealni mehanizem za premikanje v kartezičnih koordinatah. Tako se načrtovanje sistema zelo poenostavi, saj novejši »resolved

rate« krmilniki poznajo posebne postopke za reševanje težav pri kinematičnih singularnostih.

Naslednja večja klasifikacija ločuje med vodenjem na osnovi pozicije in vodenjem na osnovi slike.

Pri vodenju na osnovi pozicije gre za iskanje določenih značilnosti slike, s pomočjo katerih ocenimo lego ciljnega objekta glede na kamero. Povratna informacija se računa tako, da se napaka ocenjene lege zmanjšuje.

Pri vodenju na osnovi slike se regulacijski vhodi (izhodi regulatorjev) računajo direktno iz slike. Tak pristop lahko zmanjša zakasnitve zaradi računanja, izloči potrebo po interpretaciji slike, obenem pa odpadejo tudi napake pri modeliranju kamere. Takšen pristop predstavlja precejšen izziv za načrtovalce vodenja, saj je sistem močno nelinearen in medsebojno odvisen (ang.: *coupled*).

Ena izmed tipičnih aplikacij vodenja z računalniškim vidom je pozicioniranje robotskega orodja (prijemala,..) relativno na ciljni objekt. Ena izmed možnih uporab je robotsko prijemanje, pri kateri večina avtorjev uporablja kamere pritrjene na robote. V večini primerov je algoritem vodenja izražen s pomočjo gibov v lego, ki je definirana s pomočjo slike ciljnega objekta. Položaj prijemala glede na predmet je tako definiran samo neposredno, s pomočjo kinematične povezave prijemalo-kamera. Napake v tem kinematičnem modelu vodijo do napak pri pozicioniranju prijemala, ki jih sistem ne more zaznati. V primeru, ko opazujemo tudi prijemalo, je možno take napake zaznati in jih učinkovito odpraviti. V splošnem ni nobenega zagotovila o natančnosti pozicioniranja sistema, razen v primeru, ko je mogoče opazovati kontrolne točke tako na predmetu, kot na prijemalu.

Ločili bomo torej nadaljnji dve vrsti sistemov:

- Sistemi, ki opazujejo tako ciljni objekt kot robotsko orodje. Imenovali jih bomo zaprtzančni sistemi končne točke ali ECL sistemi (ang.: *endpoint closed loop systems*)
- Sistemi, ki opazujejo samo ciljni objekt. Imenovali jih bomo odprtozančni sistemi končne točke ali EOL sistemi (ang.: *endpoint open loop systems*)

Ponavadi je mogoče pretvoriti EOL sistem v ECL sistem preprosto z dodajanjem opazovanih kontrolnih točk na robotskem orodju ali kakšnih drugih za nalogo pomembnih točk. Tako se teoretično gledano zdi, da so ECL sistemi v vsakem slučaju primernejši. Vendar pa temu ni vedno tako. Ker morajo ECL sistemi slediti tako orodje (prijemala) kot ciljni objekt, zahteva implementacija ECL vodenja reševanje precej kompleksnejših problemov na področju računalniškega vida, obenem pa se postavljajo omejitve zaradi zornih kotov, ki jim ni mogoče vedno zadostiti.

3 Aplikacija

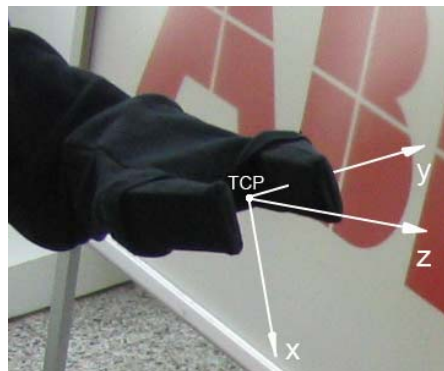
Naloga robota je bila s pomočjo računalniškega vida zaznati premikajoči predmet na drči, ga slediti in pobrati z drče. V celici je bil vključen robot ABB IRB 1400, FIS sistem računalniškega vida FDS research (osebni računalnik z digitalizatorjem slike in ustrezno programsko opremo, CCD kamera, digitalno vhodno-izhodna enota), prijemala Schunk PPG-F 80/20, drča ter ciljni objekt.

3.1 Postavitev koordinatnih sistemov

Vsi relativni gibi glede na ciljni objekt so se izvajali v koordinatnem sistemu prijemala. Poleg tega sta bila definirana še dva koordinatna sistema: koordinatni sistem drče ter sistema računalniškega vida.

Koordinatna sistema drče in računalniškega vida sta bila med seboj premaknjena. Da bi robot lahko uspešno sledil objekt, je bilo potrebno določiti translacijski zamik med njima oziroma uskladiti koordinatna sistema obeh objektov. Za popolno uskladitev je bilo

potrebno odpraviti še pojav paralakse oziroma navidezne večje oddaljenosti od resnične, ko je objekt izmaknjen iz optične osi kamere.



Slika 3: Koordinatni sistem prijemala in TCP

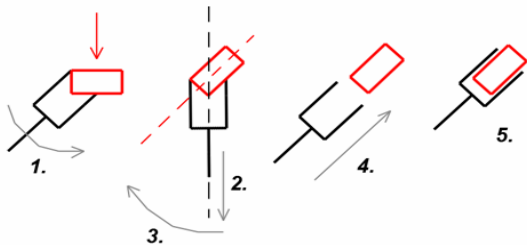
3.2 Gibanje robota (sledenje objekta)

Na robotskem krmilniku sta se izvajala dve opravili: glavno opravilo (*task main*), v katerem so se izvajale rutine za usklajevanje koordinatnih sistemov, izpis na zaslon, gibanje robota in opravilo SerialComm, ki je skrbelo za komunikacijo z »vision« sistemom in računanje novih točk iz prejetih koordinat. Večopravnost (ang.: *multitasking*) je bila uporabljena zato, da se je pridobilo na času, saj se je računanje novih točk na ta način lahko izvajalo sočasno z gibanjem robota.

Sledenje objekta je potekalo v glavnem opravilu v WHILE zanki, ki se je zaključila, ko je objekt prišel po X osi dovolj blizu prijemala. Izstopu iz zanke je sledil gib prijemala na zadnje povzročene koordinate (x , y , γ). Jedro zanke je bilo sestavljeno iz dveh gibov, ker robot za gladko gibanje potrebuje dve vnaprej znani točki, saj si s pomočjo dinamičnega modela vnaprej izračunava tokove motorjev. Sicer je v tem primeru robot vedno »za korak zadaj« vendar se to ni izkazalo kot zelo moteče. Robot si je torej računal točki v trenutnem prehodu zanke za naslednji prehod, dokler ni objekt prišel dovolj blizu prijemala. Sledil je izhod iz zanke in prijemanje.

3.3 Prijemanje

Robot je bil do sedaj po z osi k.s. prijemala za 80 mm odmaknjen od točke, na kateri se je nahajal objekt (da je lahko le ta tudi pod kotom zdrsnil v prijemalo). Sledil je gib točno na mesto objekta in objekt bi sedaj že moral biti v prijemalu. V to seveda nisem mogel biti prepričan, zato je robot še enkrat preveril razliko med kotom prijemala in kotom objekta. Če je bila ta manjša od 15° , je prijemalo zaprl, v nasprotnem primeru pa se je izvršila rutina, ki je poskušala objekt spraviti v prijemalo.



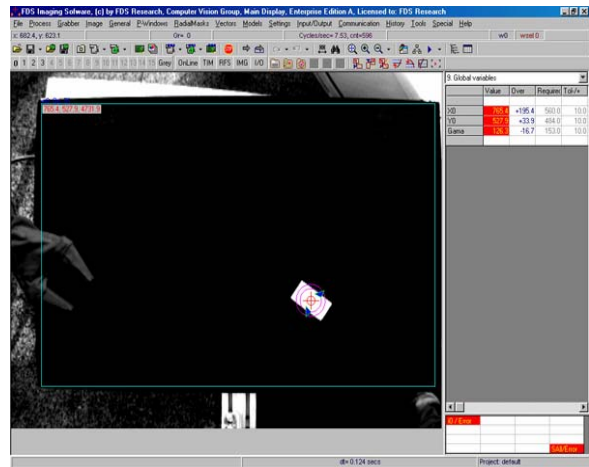
Slika 4: Popravljenje orientacije objekta

V rutini poravnave je robot najprej postavil prijemalo na kot $\gamma = 0$ in se odmaknil po osi z k.s. prijemala. Nato je vnovič zahteval koordinate in se, še vedno odmaknjen po osi z, postavil na nove koordinate. V naslednjem koraku se je zopet približal objektu in ponovno preveril ujemanje kota prijemala in objekta. Če je bila razlika še vedno prevelika, je postopek ponovil (največ petkrat), v nasprotnem primeru pa je prijemalo zaprl in se dvignil 40 cm nad točko prijemanja, kjer je preveril če je bil prijem uspešen (mikrostikala na prstih prijemala). Če je bilo temu tako je objekt odložil na določeno mesto, drugače pa je čakal na ukaz za ponoven start lovljenja.

3.4 Aplikacija računalniškega vida

Algoritem je koordinate predmeta določil tako, da je izračunal težišče svetlih točk (točk z vrednostjo nad določeno mejo) v aktivnem oknu. Na dobljeno težišče je postavil radialno plavajočo masko in poiskal presečišče robov predmeta z robovi maske. S pomočjo dobljenih

točk (težišča in dveh presečišč) ni bilo težko določiti kota γ .



Slika 5: Zaslonski sistem računalniškega vida

Zaradi principa delovanja, je vsak svetel predmet, ki se je pojavil v obravnavanem oknu, pomenil napačne koordinate objekta, saj se je težišče svetlih točk premaknilo drugam. Poleg ciljnega objekta se je v aktivnem oknu vedno pojavilo še prijemalo, ki sem ga v ta namen oblekel v rokavico iz črnega blaga.

3.5 Testiranje in rezultati

Že med prvimi poskusi zaganjanja aplikacije sem opazil, da so zakasnitve v sistemu kar velike. Večino le-teh pa je prispevala robotska stran. Prva prioriteta robotskega krmilnika je seveda gibanje robota in vse ostale naloge (prekinitve, druga opravila) so deležne manjše pozornosti. Tako lahko na primer med računsko zahtevnimi gibi od zahteve po prekinitvi pa do začetka izvajanja prekinitvene rutine preteče do 100 ms! Potem so tu še zakasnitve med postavljanjem digitalnih izhodov. Sledil je odgovor sistema računalniškega vida, ki je tudi potreboval nekaj časa. Na koncu je robotski program izvedel še potrebne pretvorbe in vpise. Tako se je na koncu izkazalo, da je od premika predmeta do ustreznega giba robota preteklo okrog 0,5s.

Vsem zakasnitvam navkljub pa je sistem presenetljivo dobro opravljal svojo nalogo. Upam si trditi, da je bilo uspešnih več kot 60%

poskusov, med katerimi je bila večina takih, ko smo namensko oteževali lovljenje s »težkimi« koti predmeta. Če je robot uspel spraviti objekt v prijemalo že v prvem poskusu, je nalogo brez težav dokončal. V nasprotnem primeru pa je s postavljenimi strategijami popravljanja kar uspešno reševal situacijo. Težko je podati oceno, vendar mislim, da je bila polovica popravljanj uspešnih.

4 Sklep

V izdelani aplikaciji je še precej prostora za izboljšave. Prva in najpomembnejša bi bila vpeljati v sistem nekaj predikcije. Ker koordinate z »vision« sistema vedno nekaj zaostajajo za dejanskim stanjem, bi iz parih povzorčenih koordinat (dveh ali treh) lahko izračunali krivuljo (vsaj premico ali parabolo) skozi dane točke. Robot bi tako lahko točko prijema določil prej in se tja tudi postavil. S tem bi v veliki meri rešili težave, ki nastopijo, ko se predmet ne giblje približno vzporedno z x osjo koordinatnega sistema drče. Naslednja stvar je prijemanje ali bolje rečeno popravljanje neuspešnih poskusov. Predvsem bi bilo zanimivo vdelati nekaj porivanja in premikanja objekta tudi z zunanjimi stranmi prstov in si ga tako pripraviti za enostaven prijem.

Današnje raziskave na področju vodenja z računalniškim vidom potekajo v smeri doseganja čim večje robustnosti za uporabo v industriji skupaj z izboljševanjem zmogljivosti vodenja v dejanskem času. Razvijalci poskušajo integrirati dobre lastnosti obeh pristopov. Vodenje na osnovi slike (*IBVS – image based visual servoing*) je zanimivo, ker ne zahteva rekonstrukcije lege objektov iz slike, medtem ko vodenje na osnovi pozicije (*PBVS*) omogoča optimalno generiranje trajektorij, a v splošnem zahteva računanje lege ciljnega objekta. Razvoj integriranih sistemov pogosto zahteva uporabo namenske opreme in učinkovitih algoritmov za doseg delovanja v dejanskem času, ki poleg tega zahteva uporabo relativno preprostih značilnosti slike (linije), saj bi bilo razpoznavanje kompleksnejših vzorcev račun-

sko predrago. Vendar, kljub uporabi preprostih značilnosti, je frekvenca vzorčenja v najboljšem primeru 25 Hz (ponavadi omejitev kamer). Nezanemarljive so tudi ostale zakasnitve v sistemih, zato je za enkrat edini način za učinkovito vodenje natančno modeliranje procesov in uporaba dobljenih modelov za prediktivno vodenje.

5 Summary

Today's research in the field of computer vision tends towards more robust systems for industry use, together with higher performance in real time. The researchers try to combine the good points of image based visual servoing (*IBVS*) and position based visual servoing (*PBVS*). *IBVS* does not require reconstruction of objects from the picture, but on the other hand, *PBVS* enables optimum path. One of the main problems in today's use of computer vision are the delays in the system. This is why we have to provide exact models of the process and use them for feed-forward control.

The application itself has quite some room for improvement. The first and the most important would be to apply a feed-forward loop, which would calculate the appropriate path. The next obvious improvement would be in the field of grasping in case of mismatch between the orientation of the object and the gripper. An idea would be to push the object with the outer side of the fingers to achieve a more appropriate orientation of the object.

6. Literatura

- [1] R. Logar, Robotsko prijemanje z računalniškim vidom, Univerzitetno diplomsko delo, Ljubljana, 2004
- [2] T. Bajd, Osnove Robotike, Ljubljana, 2001
- [3] The International Journal of Robotics Research, October – November 2003, Special Issue on Visual Servoing. Peter I. Corke. *Editorial*, chapter 1, pp. 887-888.
- [4] ABB Automation Technologies, Robotics. *Product Specification IRB 1400*, 3HAC 9376-1 / Rev 3, M2000.