

Vaja 2: Sledenje objektov

Janez Pers

Laboratorij za strojno inteligenco
Fakulteta za elektrotehniko, Univerza v Ljubljani
e-mail: janez.pers@fe.uni-lj.si

Povzetek

Pri tej nalogi boste spoznali problematiko *sledenja objektov*. Izraz *sledenje* se sicer uporablja v večih kontekstih, v tej nalogi pa se boste posvetili problematiki povezovanja detekcij objekta v trajektorijo gibanja.

1 Sledenje objektov

Sledenje je izraz, ki se uporablja tako v kontekstu celotne analize videoposnetkov, vključno z obdelavo slike, detekcijo objekta na sliki s pomočjo različnih metod (odvisnih predvsem od aplikacije, oziroma od značilnosti problema, ki ga rešujemo) in povezovanjem zaznanih pozicij v trajektorije gibanja, kot tudi v ožjem kontekstu, kjer nas zanima samo zadnji del - povezovanje že razpoložljivih detekcij v čim bolj smiselne trajektorije, ki čim bolj verno odražajo dejansko gibanje objekta. Problem povezovanja detekcij v trajektorije morda izgleda preprost, vendar pa se razmere zelo zapletejo, ko poskušamo slediti večim objektom. Če pa so objekti celo vizualno zelo podobni, pa postane sledenje zelo težek in zaenkrat v splošnem nerešljiv problem. Pri tej vaji boste v Matlabu preizkusili algoritma sledenja z optimalnim pripisovanjem detekcij, z in brez Kalmanovega filtra.

2 Baza podatkov za to vajo

Uporabili boste bazo CVBASE06 Dataset, ki jo dobite na naslednjem naslovu:

<http://vision.fe.uni-lj.si/cvbase06/downloads.html>

Obvezno preberite priložena navodila. Baza ima vključen lepo dokumentiran vmesnik za uporabo v Matlabu, ki ga preizkusite. Uporabljali boste trajektorije igralcev v **rokometu**, ki so že vključene v samo bazo, tako da se vam *z detekcijo ni treba ukvarjati*. Naloga je torej namenjena izključno problemu sledenja, torej pridobivanja trajektorij iz detekcij. Lahko pa obenem razmislite, kakšne metode bi uporabili za detekcijo objektov – lani smo pri predmetu Računalniški vid delali vajo iz detekcije objekta glede na prednaučen vzorec barve. Ali bi lahko za detekcijo uporabljali tudi katere od metod, ki ste jih spoznali letos?

Podatke o pozicijah igralcev pripravite za uporabo na naslednji način:

- Iz baze z uporabo ustreznih funkcij preberite trajektorije vseh igralcev za celoten čas, za katerega so na voljo. Prikažite si jih na ekranu s pomočjo funkcij, ki so del baze, poskusite pa jih narisati tudi sami s pomočjo funkcije `plot`.

- Te trajektorije bodo predstavljale zlati standard, glede na katerega boste vrednotili rezultate uporabljenih algoritmov.
- Da bomo lahko trajektorije uporabili kot vir realističnih podatkov, jih boste anonimizirali (torej, zanemarili dejstvo, da veste, kateremu igralcu pripadajo)
- Kot zadnji korak priprave pa jim boste primešali Gaussov šum z znano kovariančno matriko. Za to si oglejte primer iz dokumentacije za funkcijo iz Matlaba `randn`. Vsak eksperiment boste ponovili trikrat, z različno intenziteto šuma. Poigrajte se s funkcijo `randn` in pogledajte, kakšen šum vam generirajo različne kovariančne matrike.

3 Vrednotenje rezultatov

Izhod vaših metod sledenja bodo trajektorije, ki bi naj pripadale določenim osebam (igralcem). Na vsaki sliki – to je, ko govorimo o sledenju, v vsakem časovnem trenutku – ovrednotite skupno napako vaše metode kot povprečje Evklidskih razdalj v prostoru x, y med pravo pozicijo vsakega od igralcev (ki ste jo prebrali iz baze podatkov še preden ste podatke anonimizirali in pošumili) in ocenjeno pozicijo, ki vam jo da preizkušana metoda. Napaka bo zaradi zamenjav identitet verjetno rasla skozi čas, zato prikažite graf napake, izračunajte pa tudi povprečje napak od začetka do konca trajektorij.

4 Naloga 1: Optimalno prirejanje (5 točk)

Iz pošumljenih detekcij poskusite generirati trajektorije tako, da začnete z znano pozicijo vsakega od igralcev na prvi sliki iz posnetka (to je v prvem časovnem trenutku), potem pa na vsaki naslednji sliki (vsakem naslednjem časovnem trenutku) določenemu igralcu pripišete tisto anonimno pozicijo, ki mu je najbližje. Pri tem uporabite algoritem za optimalno prirejanje – tak algoritem išče optimalno porazdelitev pozicij med igralce na način, da je strošek tega prirejanja čimmanjši. Strošek prirejanja naj je v vašem primeru vsota Evklidskih razdalj med pozicijami igralcev v prejšnjem časovnem trenutku in pozicijami v tem časovnem trenutku. Za optimalno prirejanje uporabite Munkresov algoritem [1], znan tudi pod imenom "Hungarian algorithm", "Kuhn-Munkres algorithm". Najprej si preberite članek o tej metodi, nato pa metodo implementirajte z uporabo funkcije `munkres.m`.

5 Naloga 2: Kalmanov filter (5 točk)

Rezultate 1. naloge, naknadno filtrirajte s pomočjo Kalmanovega filtra v dveh dimenzijah, zasnovane na predpostavki konstantne hitrosti. V okviru predavanj vam je bil takšen filter predstavljen, vi ga le uporabite. Ne pozabite na to, da je Matlab odlično orodje za linearno algebro, zato uporabite čimveč njegovih zmožnosti matričnega računanja. Filtriranje seveda izvedete za vsakega igralca posebej.

Literatura

- [1] H. W. Kuhn. The hungarian method for the assignment problem. Naval Research Logistics (NRL), 52(1):7–21, 2005.