

Vaja 4: Robovi, gradient slike in morfološke operacije

Janez Perš, Marija Ivanovska

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

Povzetek

V okviru te vaje se bomo spoznali z nekaterimi uveljavljenimi metodami obdelave slik, ki temeljijo na izločanju robov, ter si ogledali postopke, ki jih pogosto uporabljamo na binarnih (dvobarvnih) slikah.

1 Izločanje robov (1 točka)

Pri tej in naslednji nalogi boste uporabljali sliko, ki si jo lahko prenesete iz naslednjega naslova:

<http://vision.fe.uni-lj.si/classes/RV/vaje/vaja4/hotel.jpg>

Na tej sliki boste poiskali robove na dva načina in sicer najprej z uporabo Sobelovega operatorja, nato pa še z uporabo Cannyjevega detektorja.

Sobelov operator sestavljata dve jedri filtra velikosti 3×3 , eno za vertikalne robove (S_x) in eno za horizontalne robove (S_y):

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Pri konvoluciji izbrane slike in enega izmed jeder dobimo numerično oceno gradientov intenzitete slikovnih elementov (odvode):

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * \text{Img}, \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \text{Img},$$

kjer je z Img označena matrika, ki predstavlja sliko, na kateri iščemo robove. Kombinacijo vertikalnih in horizontalnih robov dobimo z izračunom amplitude gradientov po formuli:

$$G = \sqrt{G_x^2 + G_y^2}$$

Ustvarite novo matlabovo skripto in definirajte matriki S_x in S_y . Sliko, ki ste si jo prenesli, pretvorite v sivinsko, nato pa z uporabo funkcije `imfilter` ali `conv2` izračunajte konvoluciji sivinske

slike za obe jedri, ter magnitudo gradientov. Rezultat naj bi bila sivinska slika, kjer imajo robovi svetlejšo barvo. Za izolacijo robov od ostanka slike eksperimentalno določite prag upravljanja, s katerim boste sivinsko sliko pretvorili v binarno tako, da imajo slikovni elementi robov vrednost 1.

Slaba strana uporabe odvodov je njihova občutljivost na motnje (šum na slikah), zato odstranjevanje šuma (na primer glajenje slike z uporabo Gaussovega filtra) znatno pripomore k izboljšani detekciji robov. To spoznanje je osnova Cannyjevega detektorja, ki sliko najprej zgladi, šele nato pa izračuna gradiente v x in y smeri, pri čemer poleg vertikalnih in horizontalnih robov upošteva tudi diagonalne, ki so pod kotom 45° ali 135° . Po izračunu gradientov se najdeni robovi dodatno poudarijo z odstranjevanjem točk, ki ne predstavljajo lokalnega maksimuma. Na koncu se rezultat-slika binarizira tako, da imajo slikovni elementi robov vrednost 1.

Preizkusite delovanje Cannyjevega detektorja na sivinski sliki hotela. Pri tem uporabite matlabovo funkcijo `edge` (funkcija je del Image processing toolbox-a). Ne pozabite navesti željeno metodo iskanja robov kot argument te funkcije (privzeta metoda implementira konvolucijo s Sobelovim operatorjem). Sliko, ki ste jo dobili z uporabo Cannyjevega detektorja prikažite ob sliki, ki ste jo dobili z uporabo Sobelovega operatorja in analizirajte razlike.

Namig: Za prikaz več slik znotraj istega okna lahko uporabite matlabovo funkcijo `subplot`.

2 Houghova transformacija (2.5 točk)

Na obeh binarnih slikah iz prejšnje naloge poiščite matematične zapise robov z uporabo Houghove transformacije. Algoritem Houghove transformacije sprogramirajte sami (ne uporabljajte vgrajenih matlabovih funkcij ali kode iz interneta). Kodo, ki jo boste napisali, boste namreč morali tudi podrobno razumeti in na zagovoru vaje to pokazati.

Houghovo transformacijo uporabljamo za izpeljavo značilnk na osnovi principa glasovanja. Oblike, ki nas zanimajo opišemo s pomočjo parametrov, ki oblikujejo Houghov prostor (v nadaljevanju akumulator). V primeru iskanja premic na primer iščemo oblike, ki ustrezajo matematičnemu opisu $y = kx + n$, zato bosta k in n parametra akumulatorja. Za diskretne vrednosti parametrov k in n definirane na omejenem, vnaprej določenem intervalu celih števil, bo Houghov prostor predstavljal množico z omejenim številom med seboj različnih premic. Vaša naloga je, da med vsemi možnimi premicami, ki so definirane z vsemi možnimi kombinacijami (k, n) najdete N najbolj izrazitih, ki se prilegajo robovom vaših slik. Upoštevajte $k \in [-2, 2]$ in $n \in [-200, 200]$, pri čemer oba intervala razdelite na 400 delov. Na tak način boste dobili matriko akumulatorja velikosti 401×401 . Vrednost vsakega elementa x_{ij} te matrike bo predstavljala število slikovnih točk robov binarne slike, ki pripadajo premici definirana s parom parametrov (k_i, n_j) . Premica (par (k_i, n_j)), ki bo imela najvišjo vrednost (največ glasov) bo predstavljala najbolj izrazit rob na binarni sliki. Algoritem testirajte na obeh slikah iz prejšnje naloge za različne vrednosti N . Ko boste našli vse (N) premice, jih izrišite čez originalno sliko.

3 Pridobivanje ozadja posnetka (0.5 točk)

Pri tej nalogi boste uporabljali posnetek tekme squasha, ki si ga prenesete iz naslednjega naslova:

<http://vision.fe.uni-lj.si/classes/RV/vaje/vaja4/squash.avi>

Posnetek predvajajte in se spoznajte z naravo prizora. Na sliki sta dva igralca na modrem ozadju, en nosi belo majico, drug pa rdečo. Med igro sta večinoma oddaljena en od drugega, nekajkrat pa pride med njima do kontakta.

Iz posnetka izločite sliko praznega igrišča. Kot lahko vidite, igrišče na posnetku nikoli ni prazno zato boste mogli uporabiti znanje o prizoru. Igralca se namreč ves čas premikata, zato je verjetno, da bi statistika vsakega slikovnega elementa, ki bi ga opazovali skozi čas, pokazala, da je slikovni element veliko večino časa ene same barve – barve ozadja. Spomnite se predavanj in omembe nelinearnih metod filtriranja. Izračunajte sliko ozadja s pomočjo enega samega klica ene od statističnih funkcij, ki so že vgrajene v Matlab (pozor, sliko ozadja boste morali pretvoriti v 8 bitni celoštevilski format s pomočjo funkcije `uint8`, da jo boste lahko prikazali). Za izračun ozadja ne uporabite celega posnetka, ampak izberite samo vsako deseto ali dvajseto sliko sekvence, torej 80 ali 40 slik, ki jih zložite v 4-D matriko. Na tej 4D matriki uporabite izbrano statistično funkcijo. Razmislite, katera statistična funkcija je najbolj primerna za problem, ki ga rešujete.

4 Detekcija področij in morfološke operacije (1.5 točk)

Z uporabo ozadja, ki ste ga pri prejšnji nalogi izločili naredite enostaven algoritem sledenja igralcev iz posnetka `squash`”.

Za vsako sliko posnetka `squash` najprej izračunajte sliko razlike. Slika razlike je matrika evklidskih razdalj posameznih RGB slikovnih elementov slike posnetka in RGB slikovnih elementov ozadja. To sliko prikažite s pomočjo funkcij `imagesc` in `colorbar` (paleta za prikaz lahko nastavite na sivinske vrednosti s pomočjo klica `colormap gray`). Na prikazani sliki boste opazili, da so vrednosti slikovnih elementov, ki predstavljajo igralce, precej visoke, za razliko od vrednosti večine ostalih slikovnih točk.

Določite približen prag upravljanja slike razlike tako, da na binarni sliki ohranite oba igralca, in odstranite ozadje (za ta korak je morda boljše, če ne uporabite `colormap gray`, čeprav bo slika izgledala nenaravno). V idealnem primeru naj bi slikovni elementi igralcev imeli vrednost 1 (so bele barve), slikovni elementi ozadja pa vrednost 0 (so črne barve). Ker je na sliki razlike prisoten tudi šum, boste ugotovili, da ima en del slikovnih elementov ozadja belo barvo, kar nam otežuje lokalizacijo igralcev na sliki. Zato boste z uporabo morfoloških operacij poskusili izboljšati binarno sliko tako, da bo na njej čimmanj šuma, po drugi strani pa naj bosta igralca čimbolj vidna. V ta namen uporabite Matlabovo funkcijo `bwmorph` z ustrezno izbiro parametrov.

Po izboljšavi binarne slike s pomočjo funkcije `bwlabel` najdite poziciji (ali očrtana pravokotnika ali težišči) največjih zaključenih področij na sliki – če ste vse naredili prav, bosta to že poziciji igralcev v slikovnih koordinatah. Te pozicije vrišite na izvirno sliko.

Z opisanim postopkom obdelajte celoten posnetek in naredite novo `.avi` datoteko, kjer bodo čez igralca narisane njune pozicije.

5 Detekcija na podlagi barve (1.5 točk)

Igralcev na posnetku iz prejšnje naloge ne morete identificirati samo s pomočjo slike razlike, zato s pomočjo znanja, pridobljenega na prejšnji Vaji 3 metodo nadgradite tako, da bo na podlagi barve majice, ki jo nosi igralec, določila njegovo identiteto. Uporabite lahko poljubno metodo – ročno

določanje mej v RGB prostoru, ali pa barvni histogram. Posnetek, ki ga naredite s tako popravljeno metodo, naj ima eksplicitno označeno tudi identiteto igralca (recimo 1 in 2 ali A in B).