

Vaja 1: Točkovni detektorji in vizualni deskriptorji

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

Ta vaja je namenjena spoznavanju s točkovnimi detektorji in deskriptorji, ter prikazu potencialnih aplikacij takšnih algoritmov.

1 Uvod

Pri tej vaji uporabljajte programsko orodje Matlab in orodjarno (toolbox) `vlfeat`, ki vsebuje implementacije mnogih koristnih detektorjev in deskriptorjev, ima pa tudi vmesnik za Matlab. Za namestitev orodjarne sledite navodilom na spletni strani:

<https://www.vlfeat.org/install-matlab.html>

Uporabite že prevedeno verzijo `vlfeat`.

2 Naloga 1: Poravnava slik z detektorji in deskriptorji (2,5 točk)

K sebi prenesite video, posnet z zelo nemirno kamero:

<http://vision.fe.uni-lj.si/Classes/ST/shaking.avi>

S pomočjo značilnih točk in deskriptorjev po metodi SIFT ugotovite, kakšne so geometrijske preslikave iz slike v sliko. Čeprav to ne drži popolnoma, predpostavite, da je prizor planaren. Modelirate torej afino transformacijo, ki izvede translacijo in rotacijo slik (učinek tresenja kamere). Z inverznimi preslikavami nato preoblikujete slike tako, da se posnetek "umiri".

Začnite z detekcijo značilnih točk SIFT na prvi sliki posnetka I_1 . V teh točkah izračunajte vrednosti deskriptorja SIFT. V ta namen uporabite funkcijo `vl_sift`:

<https://www.vlfeat.org/overview/sift.html>

Za detektirane točke na sliki I_1 s primerjavo SIFT deskriptorjev poiščite njihove korespondenčne točke na vseh naslednjih slikah $\{I_2, I_3, \dots, I_n\}$.

Ko boste imeli vse pare korespondenčnih točk, za vse slike $I_i, i = 2, 3, \dots, n$ izračunajte njihovo transformacijsko matriko, ki sliko I_1 preslika v I_i . V ta namen lahko uporabite Matlabovo funkcijo `estimateGeometricTransform`, ki z metodo RANSAC oceni parametre transformacije. Za aplikacijo transformacije uporabite funkcijo `imwarp`.

3 Naloga 2: Deskriptorji, uporabljeni za razpoznavanje objektov

Z namenom avtomatskega razpoznavanja objektov različnih kategorij, preizkusite tri načine vizualnega opisa slike, in sicer:

- metodo "vreče besed" (angl. bag of words), ki uporablja SIFT deskriptorje,
- opis z deskriptorjem HOG [1],
- opis s kovariančnim deskriptorjem [2].

3.1 Podatkovna baza

Uporabljajte bazo Caltech 101, ki jo dobite na naslednjem naslovu:

<https://data.caltech.edu/records/mzrjq-6wc02>

Baza med ostalim vsebuje tudi oznake objektov (očrtane pravokotnike, angl. bounding box). Deskriptorje računajte samo na delu slike, ki ga omejuje pravokotnik. Na razpolago imate tudi `show_annotation.m` datoteko, ki vam oznake objektov prikaže. Izberite **dve kategoriji objektov**, ki se vam zdita že vizualno očitno različni, tako da sebi in metodi ne boste brez potrebe oteževali dela. Pri izbiri kategorij pa vendarle pazite na to, da ne boste izbrali kategorij s premajhnim številom slik.

Bazo pripravite za uporabo na naslednji način:

- Slike iz vsake kategorije razdelite na pol. Prvo polovico slik uporabljajte za učenje, drugo pa za testiranje naučenega algoritma. Nikoli ne smete testirati na slikah, ki ste jih uporabili za učenje!
- Učno množico slik razdelite še na polovico. Prvo polovico (torej eno četrtno vseh slik) uporabite za dejansko učenje, drugo polovico pa za določanje parametrov vaših algoritmov.

Tako boste večino dela naredili na prvi polovici slik, na drugi polovici pa boste zagnali vašo metodo samo zato, da dobite končne rezultate.

Da bo delo enostavnejše in hitrejše, med razvojem algoritmov uporabljajte deskriptorje slik, ki ste jih vnaprej izračunali in shranili na disk (v `.mat` datoteko).

3.2 Metoda klasifikacije objektov

Pri klasifikaciji objektov uporabite metodo najbližjega sosedu (angl. nearest neighbour classification). Metoda deluje tako, da nov, neznan vzorec primerja z vsemi učnimi vzorci iz vseh kategorij. Primerjanje v tem kontekstu pomeni izračun razdalje med deskriptorji vzorcev, ki jih primerjamo. Upoštevajte, da je treba za primerjavo značilnic, ki nam jih dajo različni tipi deskriptorjev, uporabiti razdalje, ki so najprimernejše za določen tip deskriptorja. V vašem primeru smo izbrali tri različne deskriptorje (SIFT, HOG in kovariančni deskriptor), za katere so optimalni trije različni tipi razdalj.

Pri primerjavi, neznanemu vzorcu pripišete tisto kategorijo, v katero sodi najbližji od znanih (učnih) vzorcev. Če je torej deskriptor neznanega vzorca F_x najbolj podoben deskriptorju F_y iz kategorije A (pri primerjavi deskriptorjev smo za ta par dobili najmanjšo razdaljo), lahko sklepamo, da je neznan vzorec ravno tako predstavnik kategorije A .

3.3 Evaluacija razpoznavalnikov

Uspešnost vaših algoritmov evaluirajte tako, da preštejete število pravilno razpoznanih in število napačno razpoznanih vzorcev. Pri tem uporabite standardne mere uspešnosti razpoznavanja: procent pravilno razpoznanih pozitivnih primerov (angl. true positive rate (TPR)) in procent napačno razpoznanih pozitivnih primerov (angl. false positive rate (FPR)).

Nekaj o teh merah in sploh o konceptu binarnega razpoznavanja si lahko preberete v članku [3], ali pa na Wikipedijini strani, ki povzema glavne poudarke tega članka:

http://en.wikipedia.org/wiki/Receiver_operating_characteristic

3.4 Algoritem 1: Vreča besed in SIFT deskriptorji (3 točke)

Na slikah izbranih kategorij najprej s pomočjo SIFT detektorja poiščite značilne točke. V značilnih točkah izračunajte še SIFT deskriptorje. S pomočjo vektorske kvantizacije po metodi k-tih povprečij (angl. k-means) ugotovite, kako se ti deskriptorji združujejo. V ta namen uporabite Matlabovo funkcijo `kmeans`. Število rojev n določite eksperimentalno tako, da je $n \in [10, 15]$. Vrednost parametra n optimizirajte z uporabo 2. polovice učne množice slik. Za vsako sliko posebej izračunajte histogram deskriptorjev. Število binov histograma naj bo enako številu rojev n . Histograme normalizirajte! Normalizirani histogram enega vzorca je vaš deskriptor, ki ga primerjate s histogrami ostalih vzorcev. Kot mero podobnosti uporabite razdaljo Bhattacharyya, ki je opisana na povezavi:

http://en.wikipedia.org/wiki/Bhattacharyya_distance. Pri tem si lahko pomagate s predpripravljenimi Matlabovimi funkcijami, kot je na primer `bhattacharyya.m`.

3.5 Algoritem 2: HOG deskriptor (2 točki)

Za razpoznavanje tokrat uporabite HOG deskriptor, ki ga najdete v `vlfeat` toolboxu. Deskriptorje primerjajte z Evklidsko razdaljo.

3.6 Algoritem 3: Kovariančni deskriptor (2,5 točk)

V Matlabu implementirajte kovariančni deskriptor. Pri tem sledite matematični definiciji deskriptorja, ki je podana v članku [2]. Slike filtrirajte z uporabo matlabove funkcije `conv2`. Tak deskriptor je možno implementirati v približno 10 vrsticah kode. Pazite na izračun razdalje med deskriptorji – čeprav izračun ni enostaven, pa je v Matlabu mogoče razdaljo izračunati v eni vrstici kode.

Literatura

- [1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893, 2005.
- [2] F. Tuzel, O. amd Porikli and P. Meer. Region covariance: A fast descriptor for detection and classification. In *European Conference on Computer Vision (ECCV)*, May 2006.
- [3] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861 – 874, 2006.