

Vaja 1: Pravila in uvod v obdelavo slik z Matlabom

Janez Persš

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

Povzetek

Ta vaja je namenjena spoznavanju pravil, ki bodo veljala na laboratorijskih vajah iz predmeta Računalniški vid, utrjevanju vaših spretnosti pri delu z Matlabom ter spoznavanju osnovnih konceptov, ki jih srečate pri obdelavi slik. Spoznali boste, kako so v Matlabu predstavljene slike, ter se naučili naslavljanja posamezne slikovne elemente. Naučili se boste prebrati sliko ali videoposnetek iz diska v Matlabov delovni prostor (workspace), ter ju po obdelavi shraniti nazaj na disk. Izvedli boste nekaj preprostih operacij na nivoju slikovnih elementov.

1 Uvod

Vaje pri predmetu Računalniški vid bodo potekale s pomočjo interaktivnega razvojnega okolja Matlab. Pričakujemo, da ste v toku dosedanjega študija sam Matlab že spoznali, in da obvladate vsaj osnovno interaktivno delo z njim (enostavni in kompleksnejši izračuni) ter da znate napisati in pognati tudi krajši m-program. Ta vaja je namenjena tudi temu, da ocenite, ali je vaše poznavanje osnov zadostno. Zato je vaja sorazmerno enostavna. Kljub temu morate rešiti vse naloge in vajo zagovarjati na predpisan način. V primeru, da ugotovite, da vaše znanje Matlaba šepa, vam priporočamo, da posežete po ustrezni literaturi - na primer po knjigi [1].

1.1 Način izvedbe vaj

(Velja za vse vaje). Vsaka vaja bo sestavljena iz treh delov. V prvem delu vsake od vaj (prva ura prvega termina) bo asistent na kratko predstavil vsebino vaje, ki je na vrsti, ter cilje, ki jih morate doseči. Za tem sledi samostojno delo, ki se nadaljuje še naslednji teden, opravite ga pa lahko tudi doma. Tretji termin vsake od vaj bo posvečen zagovoru vaje ter predstavitvi naslednje vaje. Vaša obveznost je, da zagovarjate vajo najkasneje v tem terminu, če želite "kandidirati" za ocene v razponu med 6 in 10. Če iz kakršnegakoli razloga vaje ne zagovarjate, ali pa ocena ni pozitivna, imate še možnost popravnega izpita v naslednjem terminu (ko poteka že samostojno delo za novo vajo), vendar je takrat najvišja možna ocena 7. Če katerekoli od vaj ne zagovarjate (ali pa je zagovor ocenjen negativno), jo lahko zagovarjate na koncu semestra, vendar lahko tako zagovarjate *največ eno vajo!* Če vam do konca semestra ostane več kot ena vaja za zagovor, laboratorijskih vaj niste opravili.

Torej, vsak drugi teden začnemo z novo vajo, potem pa imate na voljo še dva termina, da to vajo zagovarjate brez kazenskega znižanja ocene. Vaje bodo torej potekale nekako takole (primer ne prikazuje celotnega semestra):

termin	1. vaja	2. vaja	3.vaja ...
1. termin	uvod		
2. termin	sam. delo		
3. termin	zagovor	uvod	
4. termin	(zagovor za max. 7)	sam. delo	
5. termin		zagovor	uvod
6. termin		(zagovor za max. 7)	sam. delo
7. termin			zagovor
8. termin			(zagovor za max. 7)

...

Primer ocenjevanja: Prvo vajo zagovarjate v drugem terminu. Vaše znanje je zadovoljivo, dobite oceno 8. Drugo vajo zagovarjate v šestem terminu. Zagovor je odličen, naloga perfektno rešena, vendar boste dobili zaradi zamude pri zagovoru oceno 7. Tretjo vajo zagovarjate v sedmem terminu, vendar asistent ni zadovoljen z nivojem vašega znanja, zato lahko a) zagovarjate v 8. terminu za maksimalno oceno 7, b) tvegate in si zagovor prihranite za konec semestra, ko boste sicer lahko dobili tudi oceno 10, vendar ne smete zamuditi nobenega zagovora več.

1.2 Potek samostojnega dela in potek zagovora

Zaradi prostorskih omejitev boste delali v skupinah po 2 študenta. Posamezne naloge boste rešili tako, da najprej (če želite) interaktivno (v Matlabovi komandni vrstici) preizkusite posamezne korake ter preverite delovanje posameznih funkcij. Potem ustvarite novo datoteko s končnico .m (lahko je v obliki skripte ali funkcije) in v njo sestavite zaporedje korakov, ki predstavljajo rešitev naloge. Posamezne funkcionalne celote razmejite s pomočjo klica funkcije `pause`, ki bo ustavila izvajanje, dokler ne pritisnete tipke za nadaljevanje. Kot rezultat vašega dela priznamo samo izdelek v obliki .m datoteke, interaktivno delo ne šteje.

Delo v parih pomeni, da si lahko sodelavca za istim računalnikom delita isto programsko kodo pri zagovoru. Med zagovorom bo asistent postavil nekaj vprašanj, ki se bodo dotikale tiste naloge, še posebej bo pa pozoren na to, ali znate pravilno pojasniti delovanje .m kode, ki ste jo predstavili kot rezultat vašega dela. Glede na nivo znanja, ki ste ga prikazali skozi napisano kodo *in* odgovore na vprašanja, boste dobili oceno. Tudi če si dva študenta delita isto kodo, je lahko ocena različna, če je nivo znanja na zagovoru različen.

Naloge lahko rešujete tudi doma (to velja predvsem za vmesni termin, šamostojno delo). V tem primeru vas prosimo, da na vaje prinesete (prenosni) računalnik, da ne boste s svojim zagovorom pri delu ovirali študenta, s katerim ste sicer dodeljeni za isti računalnik.

Iz vsega omenjenega sledi, da je vaša minimalna obveznost prisotnost na približno polovici vaj (če naloge rešujete doma), v okviru te prisotnosti pa morate zagovarjati vse vaje.

2 Naloga 1.1: Branje in pisanje slik ter video datotek

Najprej iz strežnika prenesite k sebi dve sliki in en video posnetek. To boste storili z uporabo orodja `wget`, lahko pa uporabite tudi brskalnik Firefox:

```
!wget http://vision.fe.uni-lj.si/classes/RV/vaje/vaja1/lena.tiff
!wget http://vision.fe.uni-lj.si/classes/RV/vaje/vaja1/baboon.tiff
!wget http://vision.fe.uni-lj.si/classes/RV/vaje/vaja1/Hollywood2-t00427-rgb.avi
```

Še razlaga: klicaj pomeni v Matlabu klic zunanjega programa.

Sivinske slike so v Matlabu zapisane kot (2D) matrike slikovnih elementov, barvne slike pa kot (3D) polja slikovnih elementov. Za primer lahko poskusimo "ročno" sestaviti naslednjo sivinsko sliko dimenzije 3x3 slikovnih elementov:

```
A=[20,100,255;255,0,80;50,100,150]; image(uint8(A));
```

Vrstice slike ustrezajo vrsticam matrike, stolpci slike pa stolpcem matrike. Funkcija `image()` prikaže sliko v Matlabovem oknu, pretvorba v 8 bitni nepredznačeni format je pa potrebna zato, da zna Matlab sliko pravilno skalirati. Po izvršitvi gornje vrstice boste zagledali sliko, katere sredinski element (2,2) je popolnoma črn, saj mu je pripisana vrednost 0. Zgornji desni element ter element (2,1) pa bosta popolnoma bele barve, saj jima je pripisana vrednost 255.

Barvne slike so zapisane kot 3D polja, kjer prvi dve dimenziji (vrstica in stolpec) ustrezata vrstici oziroma stolpcu slike, tretja dimenzija (v tej dimenziji ima slika vedno točno tri elemente) pa predstavlja tri barvne ravnine, torej rdečo, zeleno in modro.

Sliko preberemo iz diska s pomočjo funkcije `imread`. Na primer, sliko Lene preberete iz diska in prikažete na naslednji način:

```
A=imread('lena.tiff');
image(A);
```

Če želite izvedeti o tej funkciji kaj več, lahko dobite obsežno dokumentacijo s pomočjo ukaza `doc imread`, ali pa malo manj obsežno s pomočjo klica `help imread`. V nadaljevanju vaj bomo zato navajali le imena funkcij ki jih boste uporabljali, na vas pa je, da s pomočjo dokumentacije ugotovite način, kako jih uporabiti.

Za branje video datotek uporabite funkcijo `aviread`, ki pa ne vrne neposredno RGB slike, ampak strukturo, ki vsebuje dve komponenti, `cdata` in `colormap`. RGB slika je zapisana v komponenti `cdata`.

Naloga. Naložite sliko `lena.tiff` in izpišite vrednosti vseh treh (R,G,B) kanalov slikovnega elementa v vrstici 350 in stolpcu 100. Podobno naredite še za sliko `image(A)baboon.tiff`, in za slike z indeksi od 50 do 60 v video posnetku `Hollywood2-t00427-rgb.avi`.

3 Naloga 1.2: Zapisovanje slik in posnetkov, ter enostavna aritmetika

V prejšnji nalogi ste delali z barvnimi slikami. Recimo da je transformacija iz RGB slike v sivinsko sliko definirana z naslednjo formulo: $Y = 0.299R + 0.587G + 0.114B$,

pri čemer je Y sivinska vrednost slikovnega elementa. Upoštevajte, da funkcija `image` pravilno prikaže le RGB slike. Kako boste dobljeno sivinsko sliko pretvorili nazaj v RGB zapis, ki se bo

prikazal s pomočjo funkcije `image(uint8(A))`?

Naloga. Sliki "lena" in "baboon" pretvorite v sivinski sliki, ju razširite nazaj na 3 kanale in shranite na disk s pomočjo funkcije `imwrite`, ter prikažite s pomočjo funkcije `image`. Podobno operacijo izvedite s posnetkom `Hollywood2-t00427-rgb.avi`, rezultat pa shranite v posnetek z imenom `gray.avi`. Obdelati morate vse slike posnetka! Za zapisovanje video posnetka uporabite funkciji `avifile` in `addframe`.

Naloga. Sliki in posnetek iz prejšnje točke obdelajte in shranite na disk na naslednji način. Najprej izračunajte sivinsko vrednost vsakega slikovnega elementa, potem pa generirajte barvo pikselov na obdelani sliki po naslednjem pravilu:

$$barva = \begin{cases} zelena & Y < 100, \\ oranžna & 100 < Y < 200, \\ rdeča & Y > 200, \end{cases} \quad (1)$$

Literatura

[1] Timothy A. Davis. *Matlab Primer*. CRC Press, eighth edition, 2011. Uporabite Google.