

Vaja 2: Kalibracija kamer, projektivna transformacija in radialna distorzija

Janez Persh

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

Povzetek

V okviru te vaje si bomo spoznali koncept kalibracije kamer. Najprej se boste spoznali z uveljavljenim orodjem za kalibracijo kamer (Camera Calibration Toolbox za Matlab), potem pa boste sami izvedli na poenostavljenem primeru izvedli dve bistveni komponenti kalibracije: korekcijo radialne distorzije in projektivno transformacijo.

1 Kalibracija kamer(e) s toolboxom (20%)

Pri tej vaji boste uporabljali Camera Calibration Toolbox, avtorja Jeana Yvesa Bougueta. Za vajo boste toolbox downloadali in instalirali sami, in sicer na naslednji način. Najprej ga prenesete iz naslednjega naslova:

http://www.vision.caltech.edu/bouguetj/calib_doc/

Na tej strani najdete obsežno dokumentacijo in navodila za uporabo. Povezavo za download najdete sami. Datoteko `toolbox_calib.zip` shranite v vaš domači direktorij (to je `/home/rv`, odprite terminalsko okno ter jo odpakirajte z ukazom `unzip toolbox_calib.zip`. Ker boste pri tej vaji imeli opraviti z množico slik, ustvarite nov direktorij z imenom "vaja02", in sicer takole (velja za linux) `mkdir vaja02`. Zaženite Matlab z ukazom `matlab &`, in ga zaenkrat pustite na miru. V terminalskem oknu se premaknite v vaš nov direktorij, namenjen današnjim vajam, `cd vaja02`, in vanj shranite še bazo slik, ki je kot primer dodana toolboxu. Dobite jo na naslednjem naslovu:

http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/calib_example.zip

ZIP datoteko odpakirate z ukazom (v terminalskem oknu!) `unzip calib_example.zip`. Če ste vse izvedli pravilno, se bo v direktoriju `vaja02` pojavilo 20 slik v formatu TIFF. Te slike kažejo kalibracijski objekt (planarno tarčo, pokrito z vzorcem šahovnice). Oglejte si jih.

Pri tej vaji bomo sledili primeru, ki je na strani toolboxa označen kot "First calibration example". Ustrezno stran odprite v Firefoxu in si jo na kratko oglejte, potem pa sledite navodilom. V okviru te naloge boste sledeč navodilom, izvedli nekaj osnovnih korakov. Delo bo od te točke naprej potekalo v Matlabu.

Najprej se v Matlabu prestavite v direktorij, kjer se nahajajo slike (vaja02): `cd vaja02`. Nato morate v Matlabovo pot dodati še pot do kalibracijskega toolboxa, in sicer z ukazom:

```
addpath ('/home/rv/toolbox_calib/').
```

Sedaj imate vse pripravljeno za delo. **Toolbox premore enostaven grafični vmesnik, ki ga priključite z ukazom `calib_gui`**. Po tem po vrsti izvedete naslednje korake, ob tem pa si pomagajte z navodili iz "First calibration example". Ko se pojavi vprašanje o tem, ali hočete delati s toolboxom v "standard mode" ali pa v "memory efficient mode", izberite "standard mode". S klikom na "image name" boste toolboxu podali koren imena slik, ki je v vašem primeru kar "Image" (pazite, delo s tem toolboxom je kombinacija klikov na grafične elemente, npr. slike, in podajanja parametrov v komandni vrstici).

Kliknite na "Read images" in toolbox bo prebral vseh 20 slik. Sedaj boste izvedli kalibracijo, vendar pa jo boste izvedli samo na 5 slikah izmed množice 20tih. Priporočljivo je, da izbirate 5 slik izmed prvih 16, pri čemer pa zaradi raznolikosti nalog ne dovolimo, da izberete pet zaporednih slik (npr. 1,2,3,4,5). Pričakujemo, da bo vsaka od skupin imela svoj izbor slik.

S klikom na "Extract grid corners" sprožite postopek iskanja značilnih točk na slikah, to je, vogalov kvadratov šahovnice. Sledite navodilom, in pazite, kako označujete smer področja, v katerem se nahajajo kvadrati. Ne ukvarjajte se posebej z radialno distorzijo, če naletite na sliko, kjer zaradi distorzije detekcija vogalov ni dobra, izberite drugo sliko in začnite postopek izločanja vogalov od začetka.

Ko ste označili točke na vseh petih izbranih slikah, lahko sprožite postopek kalibracije s klikom na "Calibration". Po nekaj računanja bo toolbox podal nekaj notranjih (intrinzičnih) kalibracijskih parametrov kamere. Pri sebi odgovorite na naslednja vprašanja (odgovore boste morali poznati pri zagovoru vaje):

- Zakaj sploh ocenjujemo optično središče (principal point)? Zakaj ni vedno kar v središču slike?
- V kakšnih enotah je podana goriščna razdalja, zakaj ima dve komponenti, in kako bi jo preračunali v milimetre?
- Kaj so elementi vektorja distorzije, ki jih izračuna toolbox?

Za ilustracijo si oglejte še različne grafične prikaze po končani kalibraciji, najmanj pa "Show extrinsic" in "Analyze error". Kaj vam pokažeta? Ti dve sliki tudi shranite na disk v formatu JPG, kot dokaz, da ste izvedli to nalogo. Ta naloga sicer ne zahteva nobenega programiranja. Preden zaprete vmesnik toolboxa, uporabite še opcijo "undistort images". Ta vam shrani slike, na katerih je popravljena radialna distorzija, v nove slikovne datoteke. Oglejte si jih in jih primerjajte z originalnimi slikami.

2 Projektivna transformacija (40%)

Napišite program (funkcijo) v Matlabu, ki bo korigirala popačenje, ki ga na sliki šahovnice povzroči projektivna transformacija (ta se pojavi takoj, ko optična os kamere ni popolnoma pravokotna na ravnino šahovnice). Predpostavimo, da slika nima radialnega popačenja. Zato boste funkciji podali slike, ki ste jih s toolboxom za kalibracijo kamer izvozili čisto na koncu, z "undistort images".

Program, ki ga boste napisali, naj deluje na naslednji način:

- Kot edini parameter sprejme sivinsko sliko šahovnice.
- Šahovnico prikaže uporabniku, ta pa označi nekaj značilnih točk (premislite, katerih, in koliko). Z uporabo ustreznih funkcij Matlabu izračuna parametre projektivne transformacije med prikazano sliko, in idealno sliko, na kateri ni opaziti projektivnega popačenja.
- S pomočjo teh parametrov izvede transformacijo slike tako, da projektivno popačenje izgine, ter sliko prikaže. Zatem prikaže še osnovno sliko, na kateri izriše oglišča vseh kvadratov, tudi tistih, ki jih uporabnik ni podal (namig – uporabite inverzno transformacijo).

Za izvedbo te naloge boste potrebovali funkcije, kot so `image`, `ginput`, `maketform`, `imtransform`... Poglejte si dokumentacijo za omenjene funkcije v Matlabu, in primere uporabe.

3 Radialna distorzija (40%)

Napišite funkcijo v Matlabu, ki bo ocenila parameter radialnega popačenja, ki ga modelira naslednja funkcija (preslikava iz nepopačenega radija v popačen radij):

$$r' = f \cdot \ln \left(\frac{r_l}{f} + \sqrt{1 + \frac{r_l^2}{f^2}} \right). \quad (1)$$

in njen inverz (preslikava iz popačenega radija v nepopačen radij):

$$r'_l = -\frac{f}{2} \frac{(e^{-\frac{2r'}{f}}) - 1}{e^{-\frac{r'}{f}}}. \quad (2)$$

Parameter popačenja je v tem primeru f , večji kot je, manjše je popačenje.

Oceno popačenja izvedete na naslednji način. Najprej sestavite funkcijo za označevanje točk na sliki, ki bo z ukazom `image` prikazala eno od izvornih slik, s katerimi ste začeli delo (z vključenim radialnim popačenjem – `Image15.tif`). Označite dve vertikalni vrstici oglišč kvadratov, eno ob robu slike, eno pa približno v središču slike. Podatke shranite. Skozi ta dva niza točk potegnite premico tako, da minimizirate odstopanje v smislu vsote kvadratov. To odstopanje ocenite – oba koraka vam izvede funkcija `fitline.m`, ki jo najdete tule:

<http://www.peterkovesi.com/matlabfns/Robust/fitline.m>

Ta funkcija rabi dodatne funkcije za svoje delovanje. Najdete jih na strani avtorja, posvečeni računalniškemu vidu:

<http://www.peterkovesi.com/matlabfns/>

To oceno izvedite tudi po korekciji s pomočjo formule (2). S pomočjo for zanke izvedite oceno skupnega odstopanja točk za obe premici (vsota odstopanj) za različne vrednosti parametra distorzije f in narišite graf. Parameter f naj teče od 400 do 800 z intervalom 20. Pri kateri vrednosti f bo popačenje najboljše korigirano?

4 Dodatna naloga (10%)

Napišite program, ki določi optimalno vrednost parametra f s pomočjo nelinearne optimizacije. Optimizacijo izvedite s pomočjo Matlabove funkcije `fminsearch`.