

3. Domača naloga

Robotika in računalniško zaznavanje, 29. december 2010

Preprosta rekonstrukcija redke disparitete

Napišite funkcijo `disparity`, ki za podani (sivinski) sliki leve in desne kamere vrne seznam stabilnih točk ter ocenjeno dispariteto za te točke. Postopek naj bo tak:

- Na eni izmed slik zaznajte stabilne točke z uporabo Harrisovega detektorja robov.
- Za točke, oziroma za regije okoli točk) v drugi sliki poiščite točko, ki se z njo najbolj ujema. Pri tem uporabite eno izmed metrik podobnosti regij v sivinskih slikah (priporočena je normalizirana prečna korelacija) ter za iskalno področje predpostavite omejitve, ki izvirajo iz narave problema.
- Izračunajte dispariteto v slikovnih enotah (razliko med položajem obeh slikovnih elementov).

Glava funkcije naj izgleda tako: `function [x, y, d] = disparity(left, right, N)`, kjer sta `left` in `right` vhodni matriki, ki predstavljata par sivinskih slik, `N` pa velikost regije, na podlagi katere se vrši primerjava točk. Priporočam, da si za testiranje vizualizacije omislite tudi nek način vizualizacije disparitetne informacije.

V datoteki, ki je priložena temu opisu najdete testni par slik (zajamete seveda lahko tudi svoje) ter implementacijo Harrisovega detektorja robov (ki sicer ni ravno hitra, vendar je vsaj popolna). Primer klica funkcije `harris` je naslednji:

```
[cnr, y, x] = harris(right, 2, 10, 3);
```

kjer sta `y` in `x` vektorja koordinat točk. Za več informacij o pomenu posameznih parametrov si lahko pogledate tudi pomoč za funkcijo (`> help harris`), same vrednosti pa lahko določite tudi z eksperimentiranjem.

Splošna navodila

- Nalogo morate oddati preko Učilnice najkasneje do **12. januarja 2011**.
- Pošljite zip datoteko, katere ime je vaša vpisna številka. Arhiv naj vsebuje Octave/Matlab združljive datoteke, ki imajo ime enako imenu ustrezne funkcije.
- V funkcijah lahko uporabite samo standardne funkcije iz paketov `image` in `statistics`. Če uporabite še kakšen drug *standarden* paket, mi to omenite v komentarjih ali v priloženi tekstovni datoteki. Seveda to ne pomeni, da lahko uporabite samo ovijete (morebiti obstoječo) funkcijo, ki implementira celoten zahtevani algoritem.

- **Koda mora biti dobro dokumentirana!** Za vsako *netrivialno* vrstico kode v komentarju povejte kakšen je njen pomen v algoritmu.
- Za morebitna vprašanja me lahko kontaktirate na luka.cehovin@fri.uni-lj.si ali na forumu na Učilnici.
- Pravočasno oddane naloge bodo trenutno na hitro pregledane glede izpolnjevanja osnovnih zahtev, navedenih zgoraj. Konec semestra bom vse zbrane naloge še malo bolj podrobno pregledal in tiste, pri katerih bo obstajal sum prepisovanja (ter nekaj naključnih izbrancev) povabil na obvezen zagovor.