Univerza v Ljubljani, Fakulteta za elektrotehniko
Podiplomski magistrski študijski program 2. stopnje Elektrotehnika
Informacijsko komunikacijske tehnologije - Obdelava slik in videa (64238)

NAVODILA

Vaja 8: Houghova preslikava

Pripravili: Luka Škrlj, Gašper Podobnik & Tomaž Vrtovec

Navodila

Naslednje naloge se opravi med laboratorijsko vajo.

Postopki za razgradnjo slik, ki temeljijo na iskanju robov v dani sliki na podlagi odvodov, so pogosto nadgrajeni s postopki za povezovanje robov. Eden izmed učinkovitejših postopkov za globalno povezovanje robov, predvsem v obliki premic, pa temelji na *Houghovi preslikavi*.

1. Dana je arhivska datoteka vaja08.zip v ZIP formatu. Datoteke, ki se nahajajo v arhivu, shranite v izbrano mapo ter v Pythonu poženite uporabniški vmesnik vaja08.py (*desni klik*, *Run Python File in Terminal*). Uporabniški vmesnik omogoča nalaganje vhodne slike, določanje binarne slike robov, izris slike akumulatorja Houghove preslikave, iskanje lokalnih maksimumov v akumulatorju, izris pripadajočih premic in točk ter shranjevanje slik.

2. V skripti vaja08.py najdete deklaracijo funkcije za dvoparametrično (2P) Houghovo preslikavo v dvonimenzionalnem (2D) prostoru slike:

```
def houghTransform2D2P(iImage, stepR, stepF):
    """
    Houghova transformacija v 2D za 2 parametra.
    """

    raise NotImplementedError("Implement me")

    return oAcc, rangeR, rangeF
```

   kjer vhodni argument `iImage` $= f(x, y)$ predstavlja binarno sliko robov (za $f(x_i, y_i) \neq 0$ je (x_i, y_i) robna točka), `stepR` $= \Delta r = r_{n+1} - r_n$ korak spremembe parametra r, `stepF` $= \Delta\varphi = \varphi_{m+1} - \varphi_m$ pa korak spremembe parametra φ. Izhodni argument `oAcc` $= A(r_n, \varphi_m)$ predstavlja akumulator Houghove preslikave, `rangeR` $= [r_1, r_2, \ldots, r_N]$ vektor vseh diskretnih vrednosti parametra r, `rangeF` $= [\varphi_1, \varphi_2, \ldots, \varphi_M]$ pa vektor vseh diskretnih vrednosti parametra φ.

   Privzeto implementacijo dane funkcije nadomestite z delujočo implementacijo tako, da bo funkcija dejansko izvedla Houghovo preslikavo.

3. Preizkusite delovanje Houghove preslikave na sliki box1.png.

Gradivo

Naslednje naloge so neobvezne in namenjene boljšemu razumevanju vsebine.

1. Izrišite slike uspešnega iskanja najbolj izrazite premice ($n = 1$) na sliki box1.jpg in na sliki box2.jpg. Zapišite izbrane vrednosti parametrov iskanja robov (prag, standardni odklon), vrednost korakov Δr in $\Delta\varphi$ ter največjo končno vrednost akumulatorja. Kaj lahko na podlagi rezultatov sklepate o Houghovi preslikavi? Obrazložite odgovor.

2. Kaj se zgodi, če zmanjšamo koraka Δr in $\Delta\varphi$ na najnižjo vrednost ter iščemo večje število premic? Obrazložite odgovor.

3. Od česa je odvisna natančnost določanja premic s pomočjo Houghove preslikave? Obrazložite odgovor.

4. Navedite in obrazložite nekaj lastnosti Houghove preslikave.

5. Dana je slika circles-160x160-08bit.raw velikosti $X \times Y = 160 \times 160$ slikovnih elementov, ki je zapisana v obliki surovih podatkov (RAW) z 8 biti na slikovni element, velikost slikovnega elementa pa je enaka $\Delta x \times \Delta y = 1{,}0 \times 1{,}0\,\mathrm{mm}$. Naslednja naloga je neodvisna od danega grafičnega uporabniškega vmesnika:

   a) Določite sliko gradienta s pomočjo Sobelovih operatorjev. Izrišite amplitudne slike gradienta.

   b) Upragovite amplitudno sliko gradienta. Izrišite upragovljene slike. Zapišite tudi uporabljeno vrednost praga.

   c) Originalna slika predstavlja tri kroge različnih velikosti, pri čemer ima največji krog polmer $r = 39$ slikovnih elementov. Napišite funkcijo, ki bo na podlagi Houghove preslikave določila koordinate središča največjega kroga:

   ```
   def getCenterPoint(iImage, iRadius):
       # ...
       return oCenter, oAcc
   ```

   kjer vhodni argument `iImage` predstavlja binarno sliko robov, `iRadius` $= r$ pa polmer iskanega kroga. Izhodni argument `oCenter` $= [x_0, y_0]$ predstavlja koordinate središča kroga (x_0, y_0), `oAcc` pa akumulator Houghove preslikave. V pomoč pa sta vam lahko enačbi krožnice: $x = x_0 + r\cos\varphi$ in $y = y_0 + r\sin\varphi$.

   Izpišite koordinate središča iskanega kroga ter vrednost akumulatorja Houghove preslikave na teh koordinatah. Priložite tudi izris slike akumulatorja Houghove preslikave.

   Nalogo rešujte brez uporabe naprednih funkcij ter z uporabo funkcij in postopkov, ki smo jih do sedaj spoznali na laboratorijskih vajah.

6. V skladu z vprašanjem št. 5 napišite funkcijo, ki bo iskala kroge v sliki na podlagi Houghove preslikave:

   ```
   def getCenterPointandRadius(iImage):
       # ...
       return oCenter, oRadius, oAcc
   ```

   kjer vhodni argument `iImage` predstavlja binarno sliko robov, medtem ko izhodni argument `oCenter` $= [x_0, y_0]$ predstavlja koordinate središča kroga (x_0, y_0), `oRadius` $= r$ polmer kroga, `oAcc` pa akumulator Houghove preslikave.