

Vaja 2: Histogram slike

Pripravila: Gašper Podobnik & Tomaž Vrtovec

Navodila

Metoda *izravnave histograma* je tehnika za izboljšavo kontrasta slike.

1. Dana je sivinska slika `valley-1024x683-08bit.raw` velikosti $X \times Y = 1024 \times 683$ slikovnih elementov, ki je zapisana v obliki surovih podatkov (RAW) z 8 biti na slikovni element.

Naložite in prikažite dano sliko.

2. Napišite funkcijo za izračun histograma, normaliziranega histograma in kumulativne porazdelitve verjetnosti sivinskih vrednosti slike:

```
def computeHistogram(iImage):  
    # ...  
    # your code goes here  
    # ...  
    return oHist, oProb, oCDF, oLevels
```

kjer vhodni argument `iImage` predstavlja sliko, medtem ko izhodni argument `oHist` predstavlja histogram slike, `oProb` normalizirani histogram slike, `oCDF` kumulativno porazdelitev verjetnosti sivinskih vrednosti slike, `oLevels` pa pripadajoči vektor dinamičnega območja sivinskih vrednosti (upoštevajte, da je spodnja meja dinamičnega območja sivinskih vrednosti enaka nič: $s_{\min} = 0$).

Napišite tudi funkcijo za prikaz izbranega histograma slike:

```
def displayHistogram(iHist, iLevels, iTitle):  
    # ...  
    # your code goes here  
    # ...
```

kjer vhodni argument `iHist` predstavlja histogram, normalizirani histogram ali kumulativno porazdelitev verjetnosti slike, `iLevels` pripadajoče dinamično območje sivinskih vrednosti slike, `iTitle` pa naslov prikaznega okna. Za prikazovanje histograma uporabite funkcijo `bar()`.

Izračunajte in prikažite histogram, normalizirani histogram in kumulativno porazdelitev verjetnosti sivinskih vrednosti dane slike.

3. Napišite funkcijo za določanje slike z izravnanim histogramom:

```
def equalizeHistogram(iImage):  
    # ...  
    # your code goes here  
    # ...  
    return oImage
```

kjer vhodni argument `iImage` predstavlja sliko, izhodni argument `oImage` pa sliko z izravnanim histogramom. Izravnavo histograma izvedete preko preslikave sivinskih vrednosti $s_i \rightarrow T(s_i)$, pri čemer je funkcija preslikave T določena kot:

$$T(s_i) = \left\lfloor CDF(s_i) \cdot s_{\max} \right\rfloor; \quad 0 \leq s_i \leq s_{\max},$$

kjer je CDF kumulativna porazdelitev verjetnosti sivinskih vrednosti, s_{\max} je zgornja meja dinamičnega območja sivinskih vrednosti, operator $\lfloor \cdot \rfloor$ pa predstavlja zaokroževanje navzdol na celo število.

Za dano sliko določite sliko z izravnanim histogramom ter jo prikažite. Izračunajte in prikažite tudi histogram, normalizirani histogram in kumulativno porazdelitev verjetnosti sivinskih vrednosti slike z izravnanim histogramom.

Vprašanja

Odgovore na sledeča vprašanja zapišite v poročilo, v katerega vstavite zahtevane izrise in programske kode.

1. Priložite izris dane slike ter izrise pripadajočega histograma, normaliziranega histograma ter kumulativne porazdelitve verjetnosti sivinskih vrednosti.

Kakšne lastnosti ima histogram dane slike?

2. Priložite izris slike z izravnanim histogramom ter izrise pripadajočega histograma, normaliziranega histograma ter kumulativne porazdelitve verjetnosti sivinskih vrednosti.

Kakšne lastnosti ima histogram in kakšne kumulativna porazdelitev verjetnosti sivinskih vrednosti slike z izravnanim histogramom?

3. Napišite funkcijo za izračun entropije slike brez uporabe naprednih funkcij iz Pythonovih knjižnic:

```
def computeEntropy(iImage):  
    # ...  
    # your code goes here  
    # ...  
    return oEntropy
```

kjer vhodni argument `iImage` predstavlja sliko, izhodni argument `oEntropy` pa entropijo slike.

Priložite programsko kodo funkcije ter zapišite vrednosti entropije za dano sliko ter za sliko z izravnanim histogramom.

Entropija katere slike je večja in zakaj? Odgovor ustrezno obrazložite.

Dodatek

Odgovore na sledeče probleme ni potrebno prilagati k poročilu, prispevajo pa naj k boljšemu razumevanju vsebine.

Napišite funkcijo, ki sliki doda aditiven Gaussov šum:

```
def addNoise(iImage, iStd):  
    # ...  
    # your code goes here  
    # ...  
    return oImage, oNoise
```

kjer vhodni argument `iImage` predstavlja sliko, `iStd` pa standardni odklon dodanega šuma (povprečna amplituda dodanega šuma je nič, modelirate ga s pomočjo funkcije `randn()`), ki jo najdete v knjižnici `numpy`, v modulu `random`, medtem ko izhodni argument `oImage` predstavlja sliko z dodanim šumom, `oNoise` pa matriko dodanega šuma.

Opazujte slike z dodanim šumom ter pripadajoče histograme pri dodajanju šuma z različnim standardnim odklonom (npr. 2, 5, 10, 25 sivinskih vrednosti). Na kaj morate biti pozorni pri prikazovanju slike šuma in pri računanju pripadajočega histograma?

