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

NAVODILA

Vaja 7: Prostorsko filtriranje slik

Pripravila: Gašper Podobnik & Tomaž Vrtovec

Navodila

Prostorsko filtriranje slike $f(x,y)$ je definirano kot lokalna preslikava $\mathcal{T}$ sivinskih vrednosti vhodne oz. originalne slike, na podlagi katere dobimo izhodno oz. filtrirano sliko $g(x,y) = \mathcal{T}(f(x,y))$. Med najbolj običajne vrste prostorskega filtriranja sodijo *filtriranje z jedrom*, *statistično filtriranje* in *morfološko filtriranje*.

1. Dana je dvodimenzionalna (2D) sivinska slika cameraman-256x256-08bit.raw velikosti $X \times Y = 256 \times 256$ 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 prostorsko filtriranje 2D slike:

```
def spatialFiltering(iType, iImage, iFilter, iStatFunc=None,
    iMorphOp=None):
    # ...
    # your code goes here
    # ...
    return oImage
```

   kjer vhodni argument `iType` predstavlja vrsto filtriranja (`'kernel'` za filtriranje z jedrom, `'statistical'` za statistično filtriranje in `'morphological'` za morfološko filtriranje), `iImage` vhodno sliko, preostale vhodne argumente glede na vrsto filtriranja, in sicer:

   - *za filtriranje z jedrom*: jedro filtra (ang. kernel) `iFilter` v obliki matrike velikosti $N \times M$ ($N = 2n+1$; $M = 2m+1$; $n, m \in \mathbb{N}$), ki vsebuje koeficiente filtra $w(i,j)$, pri čemer ima središčni koeficient koordinate $(i,j) = (0,0)$, npr. za $N \times M = 3 \times 3$:

Uteženo povprečenje ($\frac{1}{16} \times$):

1	2	1
2	4	2
1	2	1

Gaussovo povprečenje ($\sigma = 0{,}5$):

0,01	0,08	0,01
0,08	0,64	0,08
0,01	0,08	0,01

Sobelov operator za x smer:

−1	0	1
−2	0	2
−1	0	1

Sobelov operator za y smer:

−1	−2	−1
0	0	0
1	2	1

Laplaceov operator:

1	1	1
1	−8	1
1	1	1

   - *za statistično filtriranje*: poljubna matrika `iFilter` velikosti $N \times M$ ($N = 2n+1$; $M = 2m+1$; $n, m \in \mathbb{N}$), ki določa prostorsko domeno filtra, ter Pythonova funkcija `iStatFunc`, ki določa statistično operacijo $\mathcal{S}$ (npr. `np.median` za mediano);
   - *za morfološko filtriranje*: matrika `iFilter` velikosti velikosti $N \times M$ ($N = 2n+1$; $M = 2m+1$; $n, m \in \mathbb{N}$), ki vsebuje koeficiente $b(i,j) \in \{0,1\}$, npr.:

$N \times M = 3 \times 3$

1	1	1
1	1	1
1	1	1

$N \times M = 3 \times 1$

1
1
1

$N \times M = 5 \times 5$

0	0	1	0	0
0	1	1	1	0
1	1	1	1	1
0	1	1	1	0
0	0	1	0	0

ter vrsto operacije `iMorphOp` (`'erosion'` za morfološko erozijo, `'dilation'` za morfološko dilacijo).

Izhodni argument `oImage` predstavlja filtrirano sliko $g(x, y)$, ki jo glede na vrsto filtriranja pridobimo kot:

– *za filtriranje z jedrom*: korelacijo jedra filtra $w(i, j)$ in vhodne slike $f(x, y)$:

$$g(x, y) = \sum_{i=-m}^{m} \sum_{j=-n}^{n} w(i, j) f(x + i, y + j);$$

– *za statistično filtriranje*: izvedbo statistične operacije $\mathcal{S}$ nad vhodno sliko $f(x, y)$:

$$g(x, y) = \underset{\substack{i=-m\ldots m \\ j=-n\ldots n}}{\mathcal{S}} \Big\{ f(x + i, y + j) \Big\};$$

– *za morfološko filtriranje*: morfološko erozijo ($\ominus$) ali dilacijo ($\oplus$) vhodne slike $f(x, y)$ s strukturnim elementom $b(i, j)$:

$$g(x, y) = (f \ominus b)(x, y) = \underset{\substack{i=-m\ldots m \\ j=-n\ldots n}}{min} \Big\{ b(i, j)\Big|_{b(i,j)\neq 0} \; f(x + i, y + j) \Big\},$$

$$g(x, y) = (f \oplus b)(x, y) = \underset{\substack{i=-m\ldots m \\ j=-n\ldots n}}{max} \Big\{ b(i, j)\Big|_{b(i,j)\neq 0} \; f(x + i, y + j) \Big\}.$$

Filtrirajte dano sliko, pri čemer uporabite naslednje filtre iz navodil:

– *za filtriranje z jedrom*: glajenje s podanim jedrom uteženega povprečja velikosti 3×3;

– *za statistično filtriranje*: prostorska domena 3×3 ter statistična operacija mediane;

– *za morfološko filtriranje*: erozija s podanim strukturnim elementom velikosti 5×5.

3. Napišite funkcijo za spreminjanje prostorske domene slike 2D slike:

```
def changeSpatialDomain(iType, iImage, iX, iY, iMode=None, iBgr=0):
    # ...
    # your code goes here
    # ...
    return oImage
```

kjer vhodni argument `iType` predstavlja vrsto spreminjanja prostorske domene (`'enlarge'` za razširitev, `'reduce'` za zmanjšanje), `iX` in `iY` pa število slikovnih elementov, ki se spremenijo (dodajo ali odvzamejo) od vsake x oz. y koordinatne smeri. Ostala dva vhodna argumenta (`iMode` in `iBgr`) boste rabili pri domači nalogi, zdaj pa sta nastavljena na privzeti vrednosti. Izhodni argument `oImage` predstavlja izhodno sliko z razširjeno ali zmanjšano prostorsko domeno. Klic funkcije `changeSpatialDomain()` opravite znotraj funkcije `spatialFiltering()`, pri čemer razširitev oz. zmanjšanje prostorske domene prilagodite velikosti filtra.

Ponovno filtrirajte dano sliko s filtri iz točke 2 navodil ter opazujte v rezultate z vidika spreminjanja prostorske domene slike.

Vprašanja

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

1. Priložite izrise slik, ki jih pridobite s filtriranjem dane slike v točki 2 iz navodil (filtriranje z jedrom: glajenje s podanim jedrom uteženega povprečja velikosti 3×3; statistično filtriranje: prostorska domena 3×3 ter statistična operacija mediane; morfološko filtriranje: erozija s strukturnim elementom velikosti 5×5).

2. Napišite funkcijo za izračun koeficientov jedra filtra za uteženo povprečenje:

```
def weightedAverageFilter(iM, iN, iValue):
    # ...
    # your code goes here
    # ...
    return oFilter
```

kjer vhodna argumenta `iM` $= M$ in `iN` $= N$ predstavljata poljubno velikost filtra $N \times M$ ($N = 2n + 1$; $M = 2m + 1$; $n, m \in \mathbb{N}$), `iValue` pa osnovno utež filtra (npr. 2 v primeru iz navodil), medtem ko izhodni argument `oFilter` predstavlja matriko jedra filtra.

Priložite programsko kodo funkcije `weightedAverageFilter()`. Kako se imenuje filter, za katerega izberemo `iValue` $= 1$?

3. Filtrirajte dano sliko z Sobelovima operatorjema, ki sta podana v navodilih. Izračunajte tudi amplitudni in fazni odziv dobljenega gradientnega vektorskega polja.

Priložite pripadajočo programsko kodo, izrise odzivov na filtriranje s Sobelovima operatorjema ter izrise amplitudnega in faznega odziva gradientnega vektorskega polja, pri čemer za prikazovanje odzivov izrabite celotno dinamično območje sivinskih vrednosti ($0 \ldots 255$).

4. Izostrite dano sliko s pomočjo t. i. maskiranja neostrih področij, pri čemer za pridobivanje maske uporabite glajenje z Gaussovim povprečenjem (jedro filtra velikosti 3×3 je podano v navodilih), za stopnjo ostrenja pa izberete $c = 2$.

   Priložite pripadajočo programsko kodo, izris maske neostrih področij z izrabljenim celotnim dinamičnim območjem sivinskih vrednosti ($0 \ldots 255$) ter izris pridobljene izostrene slike.

5. Dopolnite funkcijo za spreminjanje prostorske domene slike `changeSpatialDomain()` tako, da bo omogočala več načinov spreminjanja:

```
def changeSpatialDomain(iType, iImage, iX, iY, iMode=None, iBgr=0):
    # ...
    # your code goes here
    # ...
    return oImage
```

   kjer dodaten vhodni argument `iMode` predstavlja način spreminjanja prostorske domene, in sicer:

   - *s konstantno sivinsko vrednostjo*: `'constant'`, pri je čemer poljubna konstantna sivinska vrednost podana v vhodnem argumentu `iBgr`;
   - *z ekstrapolacijo sivinskih vrednosti*: `'extrapolation'`;
   - *z zrcaljenjem sivinskih vrednosti*: `'reflection'`;
   - *s periodičnim ponavljanjem sivinskih vrednosti*: `'period'`.

   Priložite programsko kodo dopolnjene funkcije `changeSpatialDomain()` ter izrise dane slike z razširjeno prostorsko domeno, pri čemer uporabite vsakega od zgoraj navedenih načinov razširitve tako, da sliko v vsako x koordinatno smer razširite za 128 slikovnih elementov, v vsako y koordinatno smer pa za 384 slikovnih elementov (pri razširitvi s konstantno sivinsko vrednostjo izberite vrednost 127).

6. Primerjajte rezultate, ki jih pridobite z različnimi vrstami filtriranja nad vhodno sliko, pri čemer vhodni sliki na različen način (glej vprašanje 5) spremenite prostorsko domeno.

   Kako vpliva način razširitve prostorske domene slike na rezultate filtriranja? Utemeljite odgovor.

Dodatek

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

Napišite funkcijo za izračun koeficientov jedra filtra za Gaussovo povprečenje:

```
def gaussianFilter(iM, iN, iStd):
    # ...
    # your code goes here
    # ...
    return oFilter
```

kjer vhodna argumenta `iM` $= M$ in `iN` $= N$ predstavljata poljubno velikost filtra $N \times M$ ($N = 2n + 1$; $M = 2m + 1$; $n, m \in \mathbb{N}$), `iStd` pa standardni odklon filtra, medtem ko izhodni argument `oFilter` predstavlja matriko jedra filtra.