

## Vaja 5: Sivinske preslikave slik

Pripravila: Gašper Podobnik & Tomaž Vrtovec

### Navodila

Sivinske preslikave slik  $\mathcal{T} : \mathbb{R} \rightarrow \mathbb{R}$  v splošnem preslikajo vrednosti vseh slikovnih elementov (sivinske vrednosti) iz dinamičnega območja  $[0 \dots L_f - 1]$  originalne oz. vhodne slike  $f = f(x, y, z, \dots)$  v dinamično območje  $[0 \dots L_g - 1]$  izhodne oz. preslikane slike  $g = g(x, y, z, \dots)$ .

1. Dana je sivinska slika `image-512x512-16bit.raw` velikosti  $X \times Y = 512 \times 512$  slikovnih elementov, ki je zapisana v obliki surovih podatkov (RAW) z 11 biti na slikovni element, ki predstavljajo predznačena cela števila v obliki 16-bitnega zapisa (`'int16'`).

Naložite in prikažite dano sliko.

2. Napišite funkcijo za *splošno linearno preslikavo* sivinskih vrednosti:

```
def scaleImage(iImage, iA, iB):  
    # ...  
    # your code goes here  
    # ...  
    return oImage
```

kjer vhodni argument `iImage` predstavlja vhodno sliko  $f$ , `iA` in `iB` parametra splošne linearne preslikave  $a$  in  $b$ , izhodni argument `oImage` pa predstavlja izhodno sliko  $g$ :

$$g = a f + b.$$

Prikažite sliko, pridobljeno s splošno linearno preslikavo sivinskih vrednosti dane slike pri vrednostih parametrov  $a = -0,125$  in  $b = 256$ .

3. Napišite funkcijo za *linearno oknenje* sivinskih vrednosti:

```
def windowImage(iImage, iC, iW):  
    # ...  
    # your code goes here  
    # ...  
    return oImage
```

kjer vhodni argument `iImage` predstavlja vhodno sliko  $f$ , `iC` in `iW` parametra linearnega oknenja  $c$  in  $w$ , izhodni argument `oImage` pa predstavlja izhodno sliko  $g$ :

$$g = \begin{cases} 0 & \text{pri } f < c - \frac{w}{2}, \\ \frac{L_g - 1}{w} \left( f - \left( c - \frac{w}{2} \right) \right) & \text{pri } c - \frac{w}{2} \leq f \leq c + \frac{w}{2}, \\ L_g - 1 & \text{pri } f > c + \frac{w}{2}, \end{cases}$$

kjer je  $L_g = 2^8$  zgornja meja dinamičnega območja sivinskih vrednosti izhodne slike (spodnja meja naj bo 0).

Prikažite sliko, pridobljeno z linearnim oknenjem sivinskih vrednosti slike iz točke 2 pri vrednostih parametrov  $c = 1000$  in  $w = 500$ .

4. Napišite funkcijo za *odsekoma linearno preslikavo* na podlagi kontrolnih točk:

```
def sectionalScaleImage(iImage, iS, oS):  
    # ...  
    # your code goes here  
    # ...  
    return oImage
```

kjer vhodni argument `iImage` predstavlja vhodno sliko  $f$ , `iS` in `oS` pa vektorja sivinskih vrednosti v vhodni in izhodni sliki, kjer  $[iS(i), oS(i)] = \mathbf{s}_i = (s_{f,i}, s_{g,i})$ ;  $i = 0, 1, \dots, N-1$  predstavlja vsako izmed  $N$  kontrolnih točk preslikave. Izhodni argument `oImage` predstavlja izhodno sliko  $g$ :

$$g_{[s_{g,i}, s_{g,i+1}]} = \frac{s_{g,i+1} - s_{g,i}}{s_{f,i+1} - s_{f,i}} \left( s - s_{f,i} \right) + s_{g,i}; \quad s \in [s_{f,i}, s_{f,i+2}]; \quad \forall i = 0, 1, \dots, N-2,$$

kjer  $g_{[s_{g,i}, s_{g,i+1}]}$  označuje izhodno sliko  $g$  na intervalu sivinskih vrednosti  $[s_{g,i}, s_{g,i+1}]$ ,  $[s_{f,i}, s_{f,i+1}]$  pa interval sivinskih vrednosti vhodne slike  $f$ .

Prikažite sliko, pridobljeno z odsekoma linearno preslikavo sivinskih vrednosti slike iz točke 3 pri kontrolnih točkah  $\mathbf{s}_1 = (0, 85)$ ,  $\mathbf{s}_2 = (85, 0)$ ,  $\mathbf{s}_3 = (170, 255)$  in  $\mathbf{s}_4 = (255, 170)$ .

5. Napišite funkcijo za *nelinearno gama preslikavo* sivinskih vrednosti:

```
def gammaImage(iImage, iG):  
    # ...  
    # your code goes here  
    # ...  
    return oImage
```

kjer vhodni argument `iImage` predstavlja vhodno sliko  $f$ , `iG` je parameter gama preslikave  $\gamma$ , izhodni argument `oImage` pa predstavlja izhodno sliko  $g$ :

$$g = (L - 1)^{1-\gamma} f^\gamma,$$

kjer je  $L = 2^8$  zgornja meja dinamičnega območja sivinskih vrednosti vhodne in izhodne slike (spodnja meja naj bo 0).

Prikažite sliko, pridobljeno z gama preslikavo sivinskih vrednosti slike iz točke 3 pri vrednosti parametra  $\gamma = 0,5$ .

## Vprašanja

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

1. Zapišite najmanjšo in največjo slikovno (sivinsko) vrednost posameznih slik iz točk 1–5 v navodilih.
2. Napišite funkcijo za *upragovanje* sivinskih vrednosti:

```
def thresholdImage(iImage, iT):  
    # ...  
    # your code goes here  
    # ...  
    return oImage
```

kjer vhodni argument `iImage` predstavlja vhodno sliko  $f$ , `t` je vrednost praga  $t$ , izhodni argument `oImage` pa predstavlja izhodno sliko  $g$ :

$$g(x, y) = \begin{cases} 0; & f(x, y) \leq t, \\ L_g - 1; & f(x, y) > t, \end{cases}$$

kjer je  $L_g = 2^8$  zgornja meja dinamičnega območja sivinskih vrednosti izhodne slike (spodnja meja naj bo 0).

Priložite programsko kodo funkcije `thresholdImage()` in sliko, pridobljeno z upragovanjem sivinskih vrednosti linearno oknene slike (iz točke 3 navodil) pri vrednosti parametra  $t = 127$ .

3. Priložite sliko, ki prikazuje potek števila slikovnih elementov uprakovane slike s sivinsko vrednostjo  $s_g = 0$  v odvisnosti od parametra  $t$ , pri čemer  $t$  spreminjate čez celotno dinamično območje sivinskih vrednosti linearno oknene slike (iz točke 3 navodil) po koraku 1. Kako se imenuje prikazan potek? Priložite tudi pripadajočo programsko kodo.
4. Napišite funkcijo za *odsekoma nelinearno preslikavo* na podlagi kontrolnih točk:

```
def nonLinearSectionalScaleImage(iImage, iS, oS):
    # ...
    # your code goes here
    # ...
    return oImage
```

kjer vhodni argument `iImage` predstavlja vhodno sliko  $f$ , `iS` in `oS` pa vektorja sivinskih vrednosti v vhodni in izhodni sliki, kjer  $[iS(i), oS(i)] = s_i = (s_{f,i}, s_{g,i})$ ;  $i = 0, 1, \dots, N-1$  predstavlja vsako izmed  $N = 2M + 1$  kontrolnih točk preslikave ( $M$  je naravno število). Izhodni argument `oImage` predstavlja izhodno sliko  $g$ :

$$g_{[s_{g,i}, s_{g,i+2}]} = q_i(s); \quad s \in [s_{f,i}, s_{f,i+2}]; \quad \forall i = 0, 2, 4, \dots, N-3,$$

kjer  $g_{[s_{g,i}, s_{g,i+2}]}$  označuje izhodno sliko  $g$  na intervalu sivinskih vrednosti  $[s_{g,i}, s_{g,i+2}]$ ,  $q_i(s) = A_i s^2 + B_i s + C_i$  pa predstavlja kvadratno funkcijo na intervalu sivinskih vrednosti  $[s_{f,i}, s_{f,i+2}]$  vhodne slike  $f$ , ki poteka skozi kontrolne točke  $(s_{f,i}, s_{g,i})$ ,  $(s_{f,i+1}, s_{g,i+1})$  in  $(s_{f,i+2}, s_{g,i+2})$ .

Priložite programsko kodo funkcije `nonLinearSectionalScaleImage()` in sliko, pridobljeno z odsekoma nelinearno preslikavo sivinskih vrednosti dane slike pri kontrolnih točkah  $s_1 = (0, 0)$ ,  $s_2 = (40, 255)$ ,  $s_3 = (80, 80)$ ,  $s_4 = (127, 20)$ ,  $s_5 = (167, 167)$ ,  $s_6 = (207, 240)$  in  $s_7 = (255, 255)$ . Zapišite tudi koeficiente  $A_i$ ,  $B_i$  in  $C_i$  dobljenih kvadratnih funkcij  $q_i$ ;  $i = 1, 3, 5$ .

## Dodatek

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

Implementirajte splošno nelinearno preslikavo na podlagi  $N = 2M + 1$  kontrolnih točk ( $M$  je naravno število), pri čemer kontrolne točke medsebojno povežete s kubičnimi zlepkami, tj. odsekoma zveznimi polinomi tretjega reda:

$$c_i(s) = A_i + B_i s + C_i s^2 + D_i s^3,$$

kjer so  $A_i$ ,  $B_i$ ,  $C_i$  in  $D_i$  koeficienti  $i$ -tega zlepka za  $\forall i = 0, 2, 4, \dots, N - 3$ . Drugi odvod vsakega polinoma lahko v končnih točkah nastavimo na nič ter tako podamo robni pogoj, ki dopolni sistem  $N - 2$  enačb in vodi to t.i. enostavnejše rešitve v obliki “naravnih” kubičnih zlepkov.

