

## Vaja 3: Interpolacija in decimacija slik

Priprava: Gašper Podobnik &amp; Tomaž Vrtovec

### Navodila

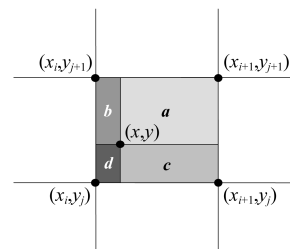
S postopkom *interpolacije slik* določamo sivinsko vrednost v izbrani točki slike na podlagi znanih sivinskih vrednosti na urejeni diskretni mreži slikovnih elementov oz. vzorčnih točk. Interpolacija ničtega reda priredi izbrani točki  $(x, y)$  sivinsko vrednost njej najbližje točke na diskretni mreži vzorčnih točk (interpolacija po principu najbližjega sosedu). Interpolacija prvega reda priredi izbrani točki  $(x, y)$  sivinsko vrednost v obliki utežene vsote sivinskih vrednosti štirih njej sosednjih točk na diskretni mreži vzorčnih točk (bilinearna interpolacija):

$$f(x, y) = a \cdot f(x_i, y_j) + b \cdot f(x_{i+1}, y_j) + c \cdot f(x_i, y_{j+1}) + d \cdot f(x_{i+1}, y_{j+1}),$$

kjer je  $f(x, y)$  sivinska vrednost v točki  $(x, y)$ . Utež posamezne vzorčne točke je sorazmerna deležu ploščine pravokotnika, ki ga določata točka  $(x, y)$ , v kateri računamo sivinsko vrednost, in točka, ki je njej diagonalno nasprotna:

$$a = \frac{(x_{i+1} - x)(y_{j+1} - y)}{(x_{i+1} - x_i)(y_{j+1} - y_j)}, \quad b = \frac{(x - x_i)(y_{j+1} - y)}{(x_{i+1} - x_i)(y_{j+1} - y_j)},$$

$$c = \frac{(x_{i+1} - x)(y - y_j)}{(x_{i+1} - x_i)(y_{j+1} - y_j)}, \quad d = \frac{(x - x_i)(y - y_j)}{(x_{i+1} - x_i)(y_{j+1} - y_j)}.$$



1. Dana je sivinska slika `pumpkin-200x152-08bit.raw` velikosti  $X \times Y = 200 \times 152$  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 interpolacijo ničtega oz. prvega reda:

```
def interpolateImage(iImage, iSize, iOrder):
    # ...
    # your code goes here
    # ...
    return oImage
```

kjer vhodni argument `iImage` predstavlja interpolacijsko sliko, `iSize=[X,Y]` vektor velikosti interpolirane slike, `iOrder` pa red interpolacije (ničti=0, prvi=1), medtem ko izhodni argument `oImage` predstavlja interpolirano sliko.

Prikažite interpolirano sliko, pridobljeno z interpolacijo ničtega oz. prvega reda, pri čemer je velikost interpolirane slike dvakrat večja od dane slike.

### Vprašanja

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

1. Določite interpolacijsko sliko kot območje velikosti  $X \times Y = 65 \times 50$  slikovnih elementov v dani sliki, pri čemer se prvi slikovni element območja nahaja na lokaciji  $(x, y) = (75, 30)$  slikovnih elementov v dani sliki (to je v 75-tem stolpcu in 30-ti vrstici slike).

Priložite sliko dobljenega interpolacijskega območja. Priložite tudi sliko histograma interpolacijskega območja ter zapišite minimalno, maksimalno in povprečno sivinsko vrednost območja.

2. Kaj so prednosti in kaj slabosti interpolacije ničtega reda?

Priložite sliko interpoliranega območja velikosti  $X \times Y = 600 \times 300$  slikovnih elementov, pridobljenega z interpolacijo ničtega reda interpolacijskega območja. Priložite tudi sliko histograma interpoliranega območja ter zapišite minimalno, maksimalno in povprečno sivinsko vrednost interpoliranega območja.

3. Kaj so prednosti in slabosti interpolacije prvega reda?

Priložite sliko interpoliranega območja velikosti  $X \times Y = 600 \times 300$  slikovnih elementov, pridobljenega z interpolacijo prvega reda interpolacijskega območja. Priložite tudi sliko histograma interpoliranega območja ter zapišite minimalno, maksimalno in povprečno sivinsko vrednost interpoliranega območja.

4. Kaj dosežemo z interpolacijami višjih redov, npr. z interpolacijo tretjega reda?

5. Pri prikazovanju slik vzpostavite dimenzijsko sorazmerje med interpolacijskim in interpoliranim območjem (tj. območji morata biti enakih fizičnih dimenzij), in sicer tako, da spremenite obstoječo funkcijo `displayImage` z dodatnima vhodnima argumentoma `iGridX` in `iGridY`:

```
def displayImage(iImage, iTitle, iGridX, iGridY):  
    # ...  
    # your code goes here  
    # ...
```

ki predstavljata vektorja položajev slikovnih elementov v  $x$  in  $y$  smeri, ki omogočata izračun parametra `extent` v funkciji `imshow` iz modula `matplotlib.pyplot` ([https://matplotlib.org/stable/api/\\_as\\_gen/matplotlib.pyplot.imshow.html](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.imshow.html)).

Priložite slike interpoliranih območij, pridobljenih z interpolacijo ničtega oz. prvega reda interpolacijskega območja, ki sta v dimenzijskem sorazmerju z interpolacijskim območjem. Priložite tudi programsko kodo spremenjene funkcije `displayImage`.

## Dodatek

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

S postopkom *decimacije slik* zmanjšamo število slikovnih elementov slike tako, da jo vzorčimo z nižjo frekvenco. Večstopenjsko decimacijo lahko predstavimo kot zaporedje decimacij s faktorjem 2 v obliki piramide. Filtrirano in z 2 decimirano sliko na nivoju  $2n$  dobimo kot konvolucijo slike s predhodnega nivoja  $n$  z jedrom  $c(i, j)$  digitalnega filtra velikosti  $N \times N$ ;  $N = 2M + 1$ :

$$2^n f(x, y) = \sum_{i=-M}^{i=M} \sum_{j=-M}^{j=M} c(i, j) \cdot 2^n f(2x - i, 2y - j).$$

Jedro  $c(i, j)$  digitalnega filtra, ki zmanjša frekvenčno širino, ima v naslednjo obliko:

$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$
$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$
$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$

$M = 1$

$\frac{1}{400}$	$\frac{1}{80}$	$\frac{1}{50}$	$\frac{1}{80}$	$\frac{1}{400}$
$\frac{1}{80}$	$\frac{1}{16}$	$\frac{1}{10}$	$\frac{1}{16}$	$\frac{1}{80}$
$\frac{1}{50}$	$\frac{1}{10}$	$\frac{4}{25}$	$\frac{1}{10}$	$\frac{1}{50}$
$\frac{1}{80}$	$\frac{1}{16}$	$\frac{1}{10}$	$\frac{1}{16}$	$\frac{1}{80}$
$\frac{1}{400}$	$\frac{1}{80}$	$\frac{1}{50}$	$\frac{1}{80}$	$\frac{1}{400}$

$M = 2$

1. Napišite funkcijo za decimacijo slike:

```
def decimateImage(iImage, iKernel, iLevel):
    # ...
    # your code goes here
    # ...
    return oImage
```

kjer vhodni argument `iImage` predstavlja decimacijsko sliko, `iKernel` jedro  $c(i, j)$  digitalnega filtra velikosti  $N \times N$ , `iLevel` pa je celoštevilčni nivo piramidne decimacije, medtem ko izhodni argument `oImage` predstavlja decimirano sliko.

Decimirajte dano sliko z danima jedroma  $c(i, j)$  digitalnega filtra velikosti  $M = 1$  in  $M = 2$  ter prikažite rezultate na nivoju piramidne decimacije `iLevel = 2`.

2. Kaj so prednosti in kaj slabosti decimacije?
3. S kakšnim faktorjem upada velikost slike pri decimaciji? S kakšnim faktorjem upada število podatkov, potrebnih za zapis slike?
4. Kakšne pogoje morajo izpolnjevati koeficienti jedra  $c(i, j)$  digitalnega filtra, ki ga uporabljate pri decimaciji slike?
5. Če bi želeli pri decimaciji preprosto vzorčiti vsak drugi slikovni element brez zmanjševanja frekvenčne širine slike, kakšna bi bila v takem primeru oblika in kakšne bi bile vrednosti jedra  $c(i, j)$  digitalnega filtra?

