

Zwiększanie rozdzielczości bitmapy

Sławomir Górawski
Aleksander Czeszejko-Sochacki

15 stycznia 2018

1 Wstęp

Zarówno podczas powiększania obrazu, jak i zmiany jego rozdzielczości pojawia się następujący problem: jaki odcień powinny mieć nowe piksele? Muszą być one, rzecz jasna, związane z odcieniami pikseli z obrazu przed przetworzeniem. Do ich ustalenia stosuje się rozmaite metody numeryczne. Pewnym z nich przyjrzymy się poniżej.

2 Cel wykonanych doświadczeń

Aby móc określić, który sposób przekształcania obrazu jest najlepszy, musimy znaleźć pewne wskaźniki jakości zmiany rozdzielczości. Zamiast tego prze-testujemy trzy metody i ocenimy je dwójako:

- Porównamy wizualnie, która metoda najlepiej odwziedla obraz sprzed przekształcenia. Porównamy ostrość krawędzi i stopień rozmycia obrazu.
- Porównamy czasy działania wszystkich metod.
- Sprawdzimy, czy kolejność skalowania (czy najpierw w pionie, czy w poziomie) ma znaczenie.

3 Opis użytych metod

Dane: bitmapa o 256 odcieniach szarości o rozmiarze $N_x \times N_y$

Szukane: bitmapa o 256 odcieniach szarości o rozmiarze $M_x \times M_y$

przy czym $N_x < M_x$ i $N_y < M_y$.

Przyjmijmy, że rozszerzanie obrazu odbywa się niezależnie w pionie i w poziomie, to znaczy najpierw rozszerzamy obraz w jednym, potem w drugim kierunku. Dodatkowo, użyte metody ograniczają się do rozszerzania obrazów o rozmiarze $M \times 1$, tzn. wektora $[p_1, p_2, \dots, p_M]$. Cały wejściowy obraz można potraktować jako sumę takich wektorów i każdy z nich niezależnie przeskalować.

Poniżej przyjmujemy $K(p_i)$ - wartość odcienia w punkcie p_i .
Użyte metody:

1. *metoda najbliższego sąsiada*, wartość odcienia piksela p_i :

$$K(p_i) = K(\text{round}(p_i))(i = 1, 2, \dots, N)$$

co oznacza, że każdemu nowemu pikselowi przyporządkowujemy odcień starego piksela o najbliższym indeksie.

2. *metoda bazująca na funkcji sklejaną pierwszego stopnia*, konstruujemy taką funkcję sklejaną pierwszego stopnia S , że:

$$S(t_i) = K(t_i)(i = 1, 2, \dots, M)$$

a następnie przyjmujemy:

$$K(p_i) = S(p_i)$$

3. *metoda bazująca na naturalnej funkcji sklejaną trzeciego stopnia*, konstruujemy taką naturalną funkcję sklejaną trzeciego stopnia Z , że:

$$Z(t_i) = K(t_i)(i = 1, 2, \dots, M)$$

a następnie przyjmujemy:

$$K(p_i) = Z(p_i)$$

Wszystkie wymienione metody zostały zaimplementowane w pliku `program.jl`,
w następujących funkcjach: `expand_using_nearest_neighbours`, `expand_using_linear_spline`,
`expand_using_cubic_spline`

4 Testy

W pliku `program.ipynb` znajdują się obrazy, dla których powyższe metody zostały przetestowane:

Nazwa	Rozmiar domyślny	Rozmiar po zmianie rozdzielczości
"Cameraman"	200×200	400×400