

Skalowanie Obrazów

Krzysztof Zając

18.11.2025

1 Interpolacja Funkcji

Zadanie polega na generacji dodatkowych punktów dla zadanej funkcji, (np. $\sin(x)$), tak aby nowe punkty były możliwie blisko oryginalnej funkcji. Zaczynając od zadanej liczby punktów (np. $N = 100$) na konkretnym odcinku (dla funkcji sinus, np. na $[-\pi, \pi]$) należy wygenerować funkcję interpolowaną mającą 2, 4 oraz 10 razy więcej punktów.

1. Wykonać interpolację 3 funkcji (podane poniżej) za pomocą jedno-wymiarowej konwolucji korzystając z różnych jąder konwolucji (podane poniżej, wybrać dowolne 3 z 5)
2. Zbadać jakość konwolucji za pomocą kryterium MSE .
3. Zdawać wpływ liczby oraz rozmieszczenia punktów na jakość konwolucji.
4. *Dodatkowe* Zbadać wpływ rozmieszczenia punktów na jakość interpolacji (punkty równomiernie rozłożone na osi x oraz losowane z rozkładu, na przykład normalnego)
5. *Dodatkowe* Porównać interpolację o dużej liczbie punktów z sekwencją interpolacji tworzących mniej punktów, na przykład interpolację tworzącą 16 razy więcej punktów zaimplementować jako sekwencję 4 interpolacji, tworzących 2 razy więcej punktów, tak że wynik każdej interpolacji jest wejściem kolejnej.

Przykładowe Funkcje do Interpolacji

$$f_1(x) = \sin(x) \quad (1)$$

$$f_2(x) = \sin(x^{-1}) \quad (2)$$

$$f_3(x) = \operatorname{sgn}(\sin(8x)) \quad (3)$$

Kryterium

Korzystając ze znajomości funkcji oryginalnej, należy sprawdzić różnicę w funkcji interpolowanej oraz oryginalnej, za pomocą metryki MSE (ang. *mean squared error*):

$$MSE(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^{N-1} (y_i - \hat{y}_i)^2, \quad (4)$$

gdzie N oznacza liczbę próbek, y funkcję oryginalną, a \hat{y} funkcję po dokonaniu interpolacji.

Funkcje Jądrowe:

$$h_1(x) = \begin{cases} 1 & t \in [0, 1) \\ 0 & t \notin [0, 1) \end{cases} \quad (5)$$

$$h_2(x) = \begin{cases} 1 & t \in [-\frac{1}{2}, \frac{1}{2}) \\ 0 & t \notin [-\frac{1}{2}, \frac{1}{2}) \end{cases} \quad (6)$$

$$h_3(x) = \begin{cases} 1 - |t| & t \in [-1, 1] \\ 0 & t \notin [-1, 1] \end{cases} \quad (7)$$

$$h_4(x) = \frac{\sin(x)}{x} \quad (8)$$

Dodatkowe jądro konwolucji opisane jest w artykule *Cubic Convolution Interpolation for Digital Image Processing* (link w źródłach). Do jądra $h_4(x)$ można dodać ucinanie, poprzez ograniczenie dziedziny funkcji do przedziału $[-a, a]$.

2 Skalowanie Obrazów

Zadanie polega na implementacji algorytmu skalowania obrazów cyfrowych na podstawie interpolacji funkcji oraz operacji splotu dyskretnego. Dla dowolnego obrazu w skali szarości należy zaimplementować operację zmniejszania korzystając ze splotu z jądrem uśredniającym. Następnie należy zaimplementować operację powiększania obrazu za pomocą 2 interpolacji funkcji, korzystając z rozwiązań poprzedniego ćwiczenia. Po pomniejszeniu oraz powiększeniu obrazu, sprawdzić jakość algorytmów subiektywnie oraz obiektywnie, za pomocą metryki *MSE* korzystając z poprzedniego ćwiczenia.

Proponowane rozszerzenia zadania

- Zaimplementować algorytm pomniejszania obrazu inaczej niż poprzez średnią z obszaru (np. *max pooling*).
- Zaimplementować operację zmniejszania oraz zwiększania o niecałkowitą wielokrotność obrazu.
- Porównać skalowanie obrazu z sekwencją mniejszych (podobnie jak w ćwiczeniu z interpolacją funkcji)
- Rozszerzyć implementację na kolorowe obrazy

Jądro Uśredniające

Jądro uśredniające może zostać użyte do zmniejszania obrazu. Przy zastosowaniu splotu wynikowy obraz będzie zawierał średnią wartości pikseli z danego rejonu obrazu, w zależności od rozmiaru jądra.

$$K_m = \frac{1}{N} \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix} \quad (9)$$

Zasady

Powyższa instrukcja dotyczy dwóch zajęć laboratoryjnych (część dotycząca obrazów w osobnym pliku), z których należy przygotować jedno sprawozdanie, z terminem oddania tydzień po drugich zajęciach. Do sprawozdania należy dołączyć kod (poprzez link do repozytorium np. [github](#)) oraz dokument tekstowy z opisem rozwiązania oraz wnioskami (w formacie PDF lub DOCX).

Proponowana ocena za poprawne wykonanie zadań bez poleceń dodatkowy to 4.0, żeby uzyskać wyższą ocenę, należy wykonać odpowiednie rozszerzenia ćwiczeń. Kilka proponowanych rozszerzeń jest uwzględnionych w instrukcji, każde z nich należy traktować jako bonus wynoszący 0.5 oceny.

Dodatkowe Źródła

- [Cubic Convolution Interpolation for Digital Image Processing](#)
- [sklearn.metrics User Guide](#)
- [scipy.interpolate](#)
- [The Algorithms](#)
- [Bilinear Interpolation and Convolution Kernels](#)
- [Wikipedia - Kernel](#)