

Zastosowania Konwolucji

Krzysztof Zając

26.11.2024

Zastosowania Konwolucji

Konwolucje mogą być wykorzystane do różnych zastosowań w przetwarzaniu obrazów, w zależności od jądra. W zadaniu należy zaimplementować konwolucję dyskretną do następujących zagadnień: wykrywanie krawędzi, rozmywanie, wyost్రzanie oraz usuwanie szumu.

Wykrywanie krawędzi

W przetwarzaniu obrazów do wykrywania krawędzi wykorzystywane są gradienty jasności obrazów, duża zmiana jasności prawdopodobnie jest krawędzią na obrazie. Do wykrywania krawędzi zastosować można operatory Sobela dla osi X oraz Y , oznaczone S_x i S_y lub operator Laplace, oznaczony L . Operatory Sobela są przybliżeniem pochodnej obrazu w osi X lub Y , natomiast operator Laplace jest przybliżeniem gradientu.

Uwaga: Operatory są czułe na kierunek obliczania pochodnej lub gradientu, zmiana jasności w stronę kolorów jasnych będzie inna niż w stronę kolorów ciemnych, ponadto mogą zmienić zakres obrazu, tak, że piskele będą miały wartości mniejsze niż 0 lub większe niż 1.

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad (1)$$

$$S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2)$$

$$L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (3)$$

Rozmywanie

Do rozmycia obrazu należy zastosować jądro uśredniające, podobnie jak przy skalowaniu, stosując krok konwolucji równy 1. Jądro może być także ważone rozkładem normalnym, wtedy określa się je jako rozmycie Gaussowskie. Przykładowe jądro Gaussowskie oznaczone jest jako G .

$$G = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 1 & 4 & 1 \\ 1 & 2 & 1 \end{bmatrix} \quad (4)$$

Wyost్రzanie

Wyost్రzanie obrazu jest operacją w pewnym sensie zestawioną z rozmycia oraz detekcji krawędzi. Rozmywanie, to inaczej lokalne uśrednianie, czyli w matematycznym sensie usuwanie wysokich częstotliwości z obrazu, a wykrywanie krawędzi to usuwanie niskich częstotliwości. Wyost్రzanie polega na podbiciu wysokich częstotliwości bez usuwania niskich, przykładowe jądro do wyost్రzania jest oznaczone poniżej jako W .

$$W = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad (5)$$

Demozaikowanie

Demozaikowanie to proces konstrukcji obrazu RGB z pomiarów natężenia światła (np. czujnikiem CMOS) oraz jednego z czujników koloru CFA (*ang.* color filter array). W fotografii cyfrowej każdy piksel zawiera sensor jednego koloru, wzór ułożenia poszczególnych kolorów nazywany jest mozaiką (stąd nazwa demozaikowanie). Zazwyczaj każdy producent aparatów stosuje inny wzór ułożenia kolorów, przykładowe wzory są podane poniżej. Celem ćwiczenia jest rekonstrukcja wybranych przykładowych obrazów z ich mozaiki do pełnego obrazu w kolorze oraz subiektywna ocena jakości, w zależności od użytej metody demozaikowania. Mozaiki 5 przykładowych obrazów są załączone w pliku ZIP.

W celu implementacji można wzorować się na przykładzie <https://github.com/kzajac97/machine-vision/tree/main/classes/demosaicking> oraz źródłach.

Filtry Kolorów

2 filtry kolorów opisane są poniżej: maska Bayera (od nazwiska autora maski, oznaczona B) oraz maska Fuji (wykorzystywana w aparatach firmy Fuji X-Trans, oznaczona jako F). Wartości R , G oraz B oznaczają obecność pomiaru natężenia kolorów czerwonego, zielonego oraz niebieskiego.

$$B = \begin{bmatrix} G & R \\ B & G \end{bmatrix} \quad (6)$$

$$F = \begin{bmatrix} G & B & R & G & B & R \\ R & G & G & B & G & G \\ B & G & G & R & G & G \\ G & B & R & G & B & R \\ R & G & G & B & G & G \\ B & G & G & R & G & G \end{bmatrix} \quad (7)$$

Metody Demozaikowania

Konwolucja

Demozaikowanie można także zaimplementować za pomocą konwolucji 2D, z odpowiednim doбором przesunięć każdego jądra konwolucji, podobnie jak zmniejszanie obrazu. Należy dobrać maskę konwolucji tak, aby ”przepisywała” lub uśredniała wartości koloru z sąsiedztwa danego piksela. Konwolucję należy wykonać dla całego obrazu, ustawiając ”krok” wynoszący jeden. Jądro konwolucji dla danego piksela ma wzmocnienie, które równe jest sumie jego wartości (np. jądra 2 na 2 z wartościami 1, będzie mieć wzmocnienie 4). Wzmocnienie może powodować wartości obrazu poza zakresem możliwym dla danego typu danych (zwykle wartości obraz są reprezentowane jako liczby rzeczywiste od 0 do 1). Dla demozaikowania, wzmocnienie powinno zależeć od mozaiki i być proporcjonalne do ilości pikseli w danym kolorze (dla maski Bayera 2 na 2, kolory czerwony i niebieski powinny mieć wzmocnienie równe 4, a zielony równe 2).

Uwaga: Niektóre biblioteki numeryczne dodają sztuczne piksele do obrazu (*ang.* padding), tak aby jego rozmiar pozostał niezmienny przez konwolucję 2D, może to powodować błędy przy brzegach obrazu. Aby temu zapobiec należy wykonywać wypełnianie pikselami o wartości zero.

Zasady

Zasady oceniania sprawozdania, aby uzyskać daną ocenę, należy wykonać wszystkie wymagania przypisane do niej. Wymagania bonusowe, są brane pod uwagę powyżej oceny 4.0.

- Wymagania minimum (3.0) - Zaimplementować 3 podane zastosowania konwolucji, każde za pomocą jednego filtra.
- Wymagania minimum (3.0) - wykonać demozaikowanie obrazów korzystając z konwolucji 2D dla filtra Bayera
- Wymagania domyślne (4.0) - Zaimplementować 3 podane zastosowania, w wybranych porównując kilka filtrów, dla każdego z nich
- Bonus (+0.5) - Zaimplementować dodatkowe filtry krawędzi, znane w literaturze, takie jak operator Sobela-Feldmana, operator Scharra lub Operator Prewitta.
- Bonus (+0.5) - Zaimplementować detektor Sobela, złożony z sumy detekcji w osiach X oraz Y .
- Bonus (+0.5) - Zaimplementować demozaikowanie dla filtra Fuji.

Dodatkowe Źródła

- <https://en.wikipedia.org/wiki/Demosaicing>
- https://en.wikipedia.org/wiki/Bayer_filter
- <http://nagykrisztian.com/store/hirakawa.pdf>
- <https://ui.adsabs.harvard.edu/abs/2006JEL....15a3003C/abstract>
- <https://paperswithcode.com/task/demosaicking>
- <https://www.semanticscholar.org/paper/Color-filter-array-recovery-using-a-threshold-based-Chang-Cheung/361502aad08b474fa7a399532608f855651a9cc4>
- https://en.wikipedia.org/wiki/Sobel_operator
- <https://pl.wikipedia.org/wiki/Sobel>
- <https://pl.wikipedia.org/wiki/Prewitt>
- [https://en.wikipedia.org/wiki/Kernel_\(image_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing))
- <https://blog.demofox.org/2022/02/26/image-sharpening-convolution-kernels/>