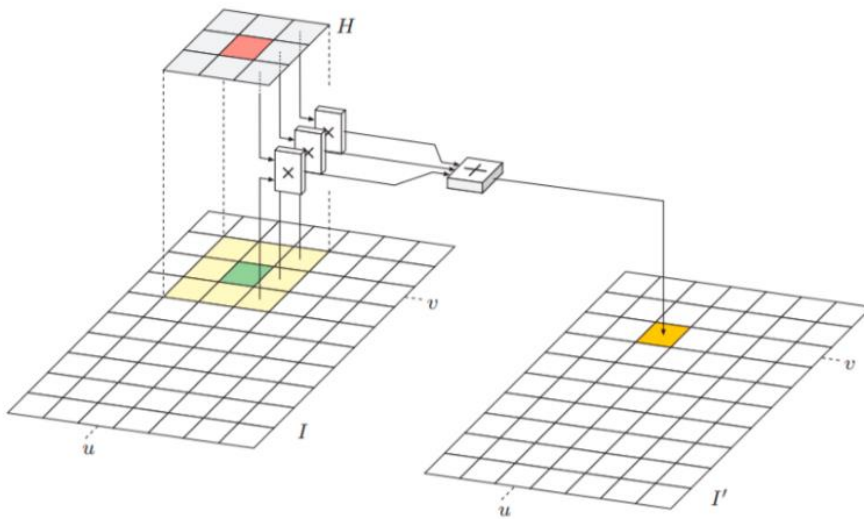


Lab9 Inne metody filtrowania obrazów, filtry konwolucyjne

Ogólna zasada: oglądamy obraz przez okno określonego wymiaru (kwadratowe o nieparzystej długości boku). Na podstawie wartości pikseli widocznych przez okno (i ewentualnie innych podanych wartości) wyznaczamy wartość piksela środkowego.

Ważne! Całe okno musi mieścić się w obrazie, więc punkty na brzegu obrazu nie „dostaną” nowych wartości.



Kwadratowe otoczenie K punktu (m, n) o boku k można wyznaczyć jak na ćwiczeniach 7.

Filtry Minimum, Maximum, Mediany, Średniej: Niech K oznacza otoczenie danego punktu (m, n) . Wtedy punktowi (m, n) dajemy wartość $p'(m, n)$, odpowiednio.

$$\begin{aligned} p'(m, n) &\leftarrow \min \{p(i, j): (i, j) \in K\} \\ p'(m, n) &\leftarrow \max \{p(i, j): (i, j) \in K\} \\ p'(m, n) &\leftarrow \text{median} \{p(i, j): (i, j) \in K\} \\ p'(m, n) &\leftarrow \text{mean} \{p(i, j): (i, j) \in K\} \end{aligned}$$

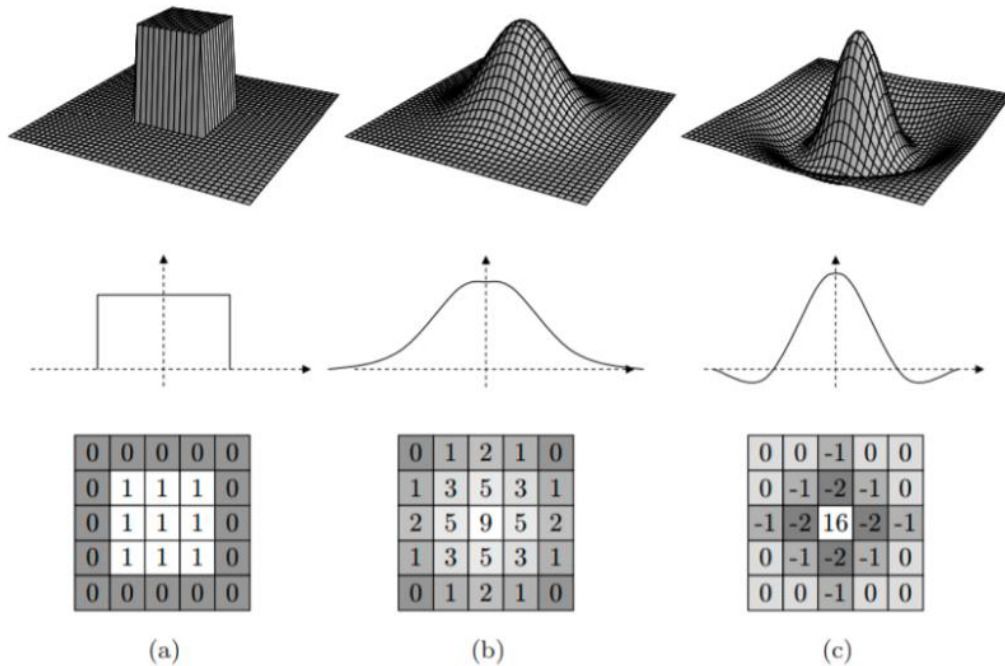
Filtry konwolucyjne

Do wyznaczenia filtru opartego na sumie ważonej, wagi przechowujemy w tablicy (lub innej strukturze danych) **H** o wymiarach takich jak okno i wykonujemy mnożenie odpowiednich wartości, sumujemy i ewentualnie dzielimy przez stałą (**scale**). Ten sposób nazywa się metodą konwolucji (ang. **Convolution**), a tablica z wagami nazwana jest jądrem (ang. **Kernel**)

Dla okna wymiaru k mamy:

$$p'(m, n) \leftarrow \sum_{a=0}^{k-1} \sum_{b=0}^{k-1} p(m-d+a, n-d+b) \cdot H(a, b), \text{ gdzie } d = \text{int}(k/2)$$

Przykłady:



Moduł ImageFilter

Filtry ze stałymi ustawieniami (1.-10) oraz z parametrami (11-18)

- | | |
|----------------------|-----------------|
| 1. BLUR | 10.EMBOSS |
| 2. DETAIL | 11.BoxBlur |
| 3. EDGE_ENHANCE | 12.GaussianBlur |
| 4. EDGE_ENHANCE_MORE | 13.UnsharpMask |
| 5. FIND_EDGES | 14.Kernel |
| 6. SHARPEN | 15.RankFilter |
| 7. SMOOTH | 16.MedianFilter |
| 8. SMOOTH_MORE | 17.MinFilter |
| 9. CONTOUR | 18.MaxFilter |

Po tych ćwiczeniach należy wiedzieć jakie parametry są stosowane przy różnych filtrach, stosowanie gotowych filtrów z powyższej listy, stosowanie filtru Kernel, filtr SOBEL

Zadanie dodatkowe dla chętnych: Wybierz [obraz.png](#) w trybie RGB (może być obraz z poprzednich ćwiczeń) i wczytaj jako [obraz](#)

1. Napisz funkcję [filtruj\(obraz, kernel, scale\)](#), która na podstawie podanej tablicy (lub listy) [kernel](#) wykonuje konwolucję (suma ważona) a następnie dzieli przez skalę [scale](#).
2. Filtr BLUR
 - a. Zastosuj filtr BLUR do swojego obrazu.
 - b. Pobierz informacje o filtrze BLUR, wstaw je jako parametry funkcji [filtruj](#). Zastosuj do obrazu.
 - c. Porównaj obrazy z a. i b.
3. Filtr CONTOUR
 - a. Zastosuj filtr CONTOUR do swojego obrazu.
 - b. Pobierz informacje o filtrze CONTOUR, wstaw je jako parametry funkcji [filtruj](#). Zastosuj do obrazu.
 - c. Porównaj obrazy z a. i b.
4. SOBEL, podobnie jak Emboss wyróżnia krawędzie. Przekonwertuj swój obraz na tryb 'L' ([obraz.convert\('L'\)](#)). Na tym obrazie:
 - a. Zastosuj filtr EMBOSS
 - b. Pobierz informacje o filtrze EMBOSS a następnie zmień zawartość listy [kernel](#).
 - i. SOBEL1: (-1, 0, 1, -2, 0, 2, -1, 0, 1). Zastosuj filtr
 - ii. SOBEL2: (-1, -2, -1, 0, 0, 0, 1, 2, 1). Zastosuj filtr
 - c. Na diagramie plt ([fig2.png](#)) umieść obraz otrzymany po konwersji na L oraz obrazy z punktów a. i b. Napisz jakie widzisz różnice między powyższymi obrazami.